Spatio-Temporal Traffic Anomaly Detection for Urban Networks

Lin Zhu

A thesis submitted as fulfilment of the requirements for degree of
Doctor of Philosophy of Imperial College London

Centre for Transport Studies
Department of Civil and Environmental Engineering
Imperial College London, United Kingdom

Aug 2019

Abstract

Urban road networks are often affected by disruptions such as accidents and roadworks, giving rise to congestion and delays, which can, in turn, create a wide range of negative impacts to the economy, environment, safety and security. Accurate detection of the onset of traffic anomalies, specifically Recurrent Congestion (RC) and Nonrecurrent Congestion (NRC) in the traffic networks, is an important ITS function to facilitate proactive intervention measures to reduce the level of severity of congestion. A substantial body of literature is dedicated to models with varying levels of complexity that attempt to identify such anomalies. Given the complexity of the problem, however, very less effort is dedicated to the development of methods that attempt to detect traffic anomalies using spatio-temporal features. Driven both by the recent advances in deep learning techniques and the development of Traffic Incident Management Systems (TIMS), the aim of this research is to develop novel traffic anomaly detection models that can incorporate both spatial and temporal traffic information to detect traffic anomalies at a network level.

This thesis first reviews the state of the art in traffic anomaly detection techniques, including the existing methods and emerging machine learning and deep learning methods, before identifying the gaps in the current understanding of traffic anomaly and its detection. One of the problems in terms of adapting the deep learning models to traffic anomaly detection is the translation of time series traffic data from multiple locations to the format necessary for the deep learning model to learn the spatial and temporal features effectively. To address this challenging problem and build a systematic traffic anomaly detection method at a network level, this thesis proposes a methodological framework consisting of (a) the translation layer (which is designed to translate the time series traffic data from multiple locations over the road network into a desired format with spatial and temporal features), (b) detection methods and (c) localisation. This methodological framework is subsequently tested for early RC detection and NRC detection.

Three translation layers including connectivity matrix, geographical grid translation and spatial temporal translation are presented and evaluated for both RC and NRC detection. The early RC detection approach is a deep learning based method that combines Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM). The NRC detection, on the other hand, involves only the application of the CNN. The performance of the proposed approach is compared against other conventional congestion detection methods, using a comprehensive evaluation framework that includes metrics such as detection rates and false positive rates, and the sensitivity analysis of time windows as well as prediction horizons. The conventional congestion detection methods used for the comparison include Multilayer Perceptron, Random Forest and Gradient Boost Classifier, all of which are commonly used in the literature.

Real-world traffic data from the City of Bath are used for the comparative analysis of RC, while traffic data in conjunction with incident data extracted from Central London are used for NRC detection. The results show that while the connectivity matrix may be capable of extracting features of a small network, the increased sparsity in the matrix in a large network reduces its effectiveness in feature learning compared to geographical grid translation. The results also indicate that the proposed deep learning method demonstrates superior detection accuracy compared to alternative methods and that it can detect recurrent congestion as early as one hour ahead with acceptable accuracy. The proposed method is capable of being implemented within a real-world ITS system making use of traffic sensor data, thereby providing a practically useful

tool for road network managers to manage traffic proactively. In addition, the results demonstrate that a deep learning-based approach may improve the accuracy of incident detection and locate traffic anomalies precisely, especially in a large urban network. Finally, the framework is further tested for robustness in terms of network topology, sensor faults and missing data. The robustness analysis demonstrates that the proposed traffic anomaly detection approaches are transferable to different sizes of road networks, and that they are robust in the presence of sensor faults and missing data.

Acknowledgements

First and foremost, I would like to express my deep and sincere gratitude to my primary supervisors, Professor John Polak and Dr Aruna Sivakumar, for their invaluable guidance, assistance and financial support throughout my PhD research at Imperial College London. I also want to express my gratitude to Dr Rajesh Krishnan for his continuous help, patient guidance and valuable suggestions from the very first steps of this study. I have been privileged to have had the opportunity to work with them for the past four years.

I would like to thank Martin Wylie of Bath City Council for providing the traffic detector data used in this PhD research. I am also very grateful to Andy Emmonds and Ashley Turner from Transport for London for the provision of traffic flow and incident data used within the thesis.

I would also like to thank Dr Nicolo Daina, Dr Jianlin Luan, Dr Simon Hu, Dr Alireza Zolfaghari and Dr Esra Suel for useful discussion, especially during the early stage of my PhD. I extremely appreciate the help from Dr Jianlin Luan in respect to the use of SUMO simulation at the beginning of this PhD research, and his useful suggestions throughout my study. I would also like to thank Dr Fangce Guo for her help in multiple perspectives. I would like to thank Dr Wenlai Zhao from Tsinghua University and Teng Yu from the University of St Andrews for the useful discussion on artificial intelligence at the very early stage of my study. Furthermore, I must also thank Sarah Wills, Fionnuala Donovan and Maya Mistry for handling the paperwork and their administrative assistance.

My very grateful thanks are also extended to my colleagues and friends at the Centre for Transport Studies and Department of Civil and Environmental Engineering for being the best office mates and for the ideas we shared during the course of my PhD. In random order, special thanks to Chenyang Wu, Francesco Manca, Heather Hou, Shiming Xu, Dr Sebastian Maier, Dr Ke Han, Dr Daniel Horcher, Roger Teoh, Nan Zhang, Yang Yu, Dr Bianca Howard, Dr Elena Psyllou, Dr Zongxiang Wu, Dr Xiang Yun, Xin Meng and Zhan Lv. I am also deeply grateful for my friends outside the department for their invaluable moral support and company during the course of my study.

I am immensely grateful to my parents and my older sister for their never-ending love, patience and support in my life; without them, this journey would have been impossible. Last but not least, I would like to thank my husband for his love, sacrifices and his continuous support and encouragement.

Declaration

I hereby certify that all material in this thesis is my own work. Any quotation from, or description of the work of others is acknowledged by reference to the sources, whether published or unpublished.

Lin Zhu

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 The Context of Intelligent Transport Systems

Transport systems are vital for modern life since an efficient transport system can significantly enhance the efficiency of individual travel, reduce traffic congestion, minimise traffic incidents and improve safety. So-called Intelligent Transport Systems (ITS) represent the state-of the-art in respect to traffic techniques or systems that aim to provide improved traffic control and management and thus a better service for road users using technology (Alam et al., 2016). Generally, ITS encompasses a broad range of control, management, electrics, wireless communication and sensing technologies with the aim to improve factors, such as safety, reliability, efficiency and sustainability of transport systems (Singh and Gupta, 2015).

The concept of ITS was initially proposed in the United States, Japan and Germany in the late 1960s and early 1970s and involved the integration of route guidance systems (Mohan, 2009). Complex forms of ITS were not deployed on a practical scale until the mid-1980s, however, with technological advances, and in particular the massive improvement in computational capabilities, made these cheaper and more reliable (Mohan, 2009). In the recent past, massive research efforts have been devoted to this subject and its applications. Examples of successfully deployed ITS systems include Urban Traffic Control (UTC) Systems, Advance Traffic Management Systems (ATMS), Advanced Driver Assistance Systems (ADAS), Advanced Traveller Information Systems (ATIS) and Advanced Vehicle Control Safety Systems (AVCSS) (Shaheen and Finson, 2013).

Even though considerable progress has been made on all aspects of ITS, we are yet to see much

improvement in overall safety or congestion (Mohan, 2009; Alam et al., 2016). Traffic congestion might be alleviated more effectively with a more precise prediction of traffic events as well as more prompt detection of events that have happened. This may now be possible with existing and emerging sensor data and associated ITS technologies that monitor traffic conditions. The recent advances in Artificial Intelligence (AI), such as machine learning and deep learning (LeCun et al., 2015), is progressively revolutionising approaches to classification and prediction and hold considerable promise in respect to monitoring or mitigating safety and traffic congestion.

1.2 Research Background

Given the scope for improvement of current ITS systems and the potential of emerging AI techniques, it is vital to have a better understanding of traffic anomalies and detection methods in order to alleviate traffic congestion properly. This section will therefore present the background on traffic anomalies and briefly summarise the current literature on anomaly detection.

1.2.1 Traffic Anomalies

Urban road networks are often affected by disruptive anomalies, such as accidents, inclement weather and roadworks. With a restriction to road capacity, these anomalies give rise to congestion and traffic delays, which can in turn create a wide range of negative impacts to the economy, environment, safety and security. Traffic congestion can cause reduced service to road users, increase delays and pollution, and lead to potential safety hazards (Weisbrod et al., 2003). According to the urban utility report published by Texas Transportation Institute, congestion led to around 6.9 billion hours of delays for road users, and around 3.1 billion gallons of wasted fuel, resulting in total economic costs of $160 billion in the urban areas of United States in 2014 alone (Schrank and Lomax, 2015). The UK also suffers such economic losses due to traffic congestion, which is getting worse, with a predicted 64% increase in the annual impact from £13.1 billion to £21.4 billion between 2013 and 2030 based on a study from the Centre for Economics and Business Research (CEBR, 2014).

The causes of traffic anomalies and congestion delay have been widely investigated in the past few decades. In general, congestion has two categories, recurrent and non-recurrent. Recurrent

congestion refers to the situation when traffic demand exceeds the road capacity, while non-recurrent congestion is caused by unpredictable changes and unexpected occurrences such as traffic incidents, weather and special events (Lomax and Margiotta, 2003). Recurrent congestion and associated delays are significant contributors to the economic losses described above (Younes and Boukerche, 2015). Hence, it is valuable to predict RC as early as possible and alleviate traffic congestion where possible.

On the other hand, non-recurrent congestion is varied in terms of its spatial and temporal dimensions, and hence can result in more disruptive impacts on the economy and society. Ikhrata and Michell (1997) have shown that 50% of congestion on US highways is due to non-recurrent congestion. Other studies have shown that traffic incidents or anomalies are also one of the major factors behind the avoidable increase of travel time and costs in transportation networks by claiming that approximately 25% of congestion is caused by non-recurring incidents such as unexpected crashes, spilled debris and broken cars (Deniz and Celikoglu, 2011). A report representing 85 large metropolitan areas in the US from 1982 to 2003 concluded that non-recurrent congestion accounted for 60% of all congestion, with traffic incidents contributing up to approximately 25% of this (Systematics, 2005). Overall, RC is shown to be a major factor as well that contributes to congestion and the resultant adverse economic impacts in cities. This has led to an increasing research interest in traffic incident analysis and management.

1.2.2 Anomaly Detection

Given the impact of traffic anomalies on the urban networks, a lot of effort has been devoted to the precise detection of traffic anomalies and their proactive control, both in terms of developing various algorithms and in the application of new sensor technologies (Chandola et al., 2009). For example, ITS have been developed to use information extracted from detectors to assist in signal control and traffic management, and traffic anomaly detection can be a very important function in ITS applications such as ATMS and ATIS.

Detecting anomalies using collected data has been studied by statisticians since the 19th century (Edgeworth, 1887). Many automatic traffic anomaly detection algorithms have been applied to motorways, such as the Motorway Incident Detection and Automated Signalling (MIDAS) systems (Highways Agency, 2005). Most previous studies focus on corridor-based methods

that apply and calibrate detection algorithms on a specific corridor (Taylor, 2008). Relatively little research has been carried out to develop methods that use spatio-temporal data from urban road networks for anomaly detection (Taylor, 2008). The extension of the traditional corridor-based method into a network level has the limitation of linearly increasing complexity as well as computational costs.

In addition to the problem of anomaly detection at an urban network level, another challenge lies in the early prediction of recurrent congestion. Traditional anomaly detection technologies such as the MIDAS system have only been able to report current or post-event traffic anomaly detection. Early detection, or an early alarm, of potential recurrent congestion is especially useful for proactive traffic control and management. Meanwhile, it is vital to localise anomalies simultaneously occurring across an urban network.

With recent developments in machine learning, especially deep learning, new advanced detection techniques provide us with a potential opportunity to incorporate the historical traffic patterns and spatio-temporal congestion propagation at a whole network level. Thus, this research seeks to extend traffic anomaly detection, including both early recurrent detection and non-recurrent detection, from traditional corridor based methods to urban roadways network based methods by using technologies that are able to incorporate information both spatially and temporally. In addition, this research will also focus on methods that offer the ability for early prediction of RC at a whole network level.

1.3 Aims and Objectives

Given the motivation stated above, this research focuses on traffic anomaly detection, encompassing both recurrent congestion and non-recurrent congestion. The broad aim of this research is to develop novel traffic anomaly detection models that can incorporate both spatial and temporal traffic information to detect multiple traffic anomalies at the urban network level rather than one or several corridors. This spatial-temporal detection model is expected to be used for dynamic real-time use that will allow traffic managers to take prompt action to react to congestion and incidents, reduce the late response caused by the undetected incidents and improve the accuracy and reliability of urban road networks.

In order to achieve the aim, the main objectives are summarised as follows:

(1) Understand traffic anomalies, the types of anomalies and their corresponding distribution of impacts across an urban traffic network in order to identify the current gaps in the literature about traffic anomaly detection and thus the relevant research challenges.

(2) Develop a novel framework for multiple traffic incidents detection and early congestion prediction at an urban network level.

(3) Develop a novel spatio-temporal model that can detect traffic anomalies and predict congestion at an early stage. Explore the application of deep learning techniques to anomaly detection and make use of advanced artificial intelligence and machine learning to develop a method to map the anomaly detection problem to deep learning and evaluate them against baseline models.

(4) Propose an approach for the localisation of traffic anomalies detected in the network in association with the proposed spatio-temporal anomaly detection.

(5) Investigate the effect of network topology, sensor faults and missing data on the accuracy of the network level traffic anomaly detection and early prediction.

1.4 Thesis Outline

The rest of the thesis is organised as follows.

- **Chapter 2** presents a comprehensive review of existing work in the area of anomaly detection, including traffic anomaly detection.
- **Chapter 3** establishes the conceptual methodological framework consisting of translation layers, detection methods and localisation, according to the above aims and objectives, before presenting potential detection algorithms and performance evaluation methods.
- **Chapter 4** explores translation methods to map traffic data at a network level into a form suitable for input to deep learning methods.

- **Chapter 5** applies the proposed methodology to real-world traffic data and evaluates the accuracy of the proposed methods compared to conventional detection methods.
- **Chapter 6** presents a method to detect the localisation of the anomaly after applying the method to network wide data.
- **Chapter 7** examines the robustness and accuracy of the proposed method when applied to data from networks with different topologies and in the presence of missing sensor data.
- **Chapter 8** summarises the main findings and contributions from this research, and presents avenues for future research.

1.5 Contributions

A number of contributions have been made in this thesis towards both understanding the nature of traffic anomalies and improving the accuracy and reliability of traffic anomaly detection by incorporating the spatio-temporal features of the urban traffic network into machine learning methods. The key contributions in this thesis are summarised below.

- The concept of anomaly and its detection led to the understanding of traffic anomalies and a systematic development of a conceptual traffic anomaly detection framework for both RC and NRC. The traffic anomaly detection framework presented in this thesis is novel to the literature.
- Identification of three translation layers and their comprehensive comparison is quite new to the literature. The translation layer acts as an important first layer to transform the input traffic data into the desired format for use by deep learning methods. Existing applications of deep learning have mainly focused on the deep learning algorithms themselves, rarely studying the impact of different translation layer methods.
- This research proposed a novel network-level based detection model that enables it to learn spatio-temporal information using one model across an entire urban network rather than single corridors or links with the application of CNN and LSTM. The proposed detection methods were compared with three conventional detection methods and found to be superior.

- It was shown that it is possible to predict RC early, before the actual congestion occurs. This would allow traffic managers to take early corrective action and thereby improve the reliability of urban road networks.
- While deep learning based methods provide superior accuracy in NRC and early RC detection, there is no straight-forward mechanism to locate the incident on the network. Six different methods to determine the location of traffic anomalies were proposed and a random forest based localisation method was found to be best suited.
- Besides the conceptual and methodological contributions, this PhD research also provides practical advice based on different issues that may arise in real-time applications including the heterogeneity of data sources, disruption from the sensor faults and the transferability of the model between different traffic networks.

1.6 Publications

As part of this study, several scholarly articles were published as below. The link between each chapter and publications is summarised in Figure 1.1.

1. **Zhu, L**, Krishnan R, Sivakumar, A, Guo, F, Polak, J. Early Identification of Recurrent Congestion in Heterogeneous Urban Traffic, *The 22nd IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Auckland, New Zealand, 2019.
2. **Zhu, L**, Krishnan R, Guo, F, Polak, J, Sivakumar, A. Traffic Monitoring and Anomaly Detection based on Simulation of Luxembourg Road Network, *The 22nd IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Auckland, New Zealand, 2019.
3. **Zhu, L**, Guo, F, Krishnan R, Polak, J. A Deep Learning Approach for Traffic Incident Detection in Urban Networks, *The 21st IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Hawaii, USA, 2018.
4. **Zhu, L**, Guo, F, Krishnan R, Polak, J. The Use of Convolutional Neural Networks for Traffic Incident Detection at a Network Level, *Transportation Research Board 97th Annual Meeting*, No. 18-00321, Washington DC 2018.

5. **Zhu, L**, Guo, F, Polak, J, Krishnan, R. Spatial-Temporal Hybrid Deep Neural Networks for Early Congestion Detection. *7th Symposium of the European Association for Research in Transportation*, Athens, Greece 2018.

6. **Zhu, L**, Guo, F, Krishnan R, Polak, J. Automated Early Detection of Congestion on Urban Roads: A Deep Learning Approach, 50th Annual Conference of the Universities-Transport-Study-Group (UTSG). London, UK, 2018.

7. **Zhu, L**, Guo, F, Polak, J, Krishnan, R. Urban Link Travel Time Estimation Using Traffic States based Data Fusion, *Journal of IET Intelligent Transport Systems*, 2018.

8. **Zhu, L**, Guo, F, Krishnan R, Polak, J. Multi-sensor Fusion Based on the Data from Bus GPS, Mobile Phone and Loop Detectors in Travel Time Estimation. *Transportation Research Board 96th Annual Meeting*. No. 17-03472, Washington DC 2017.

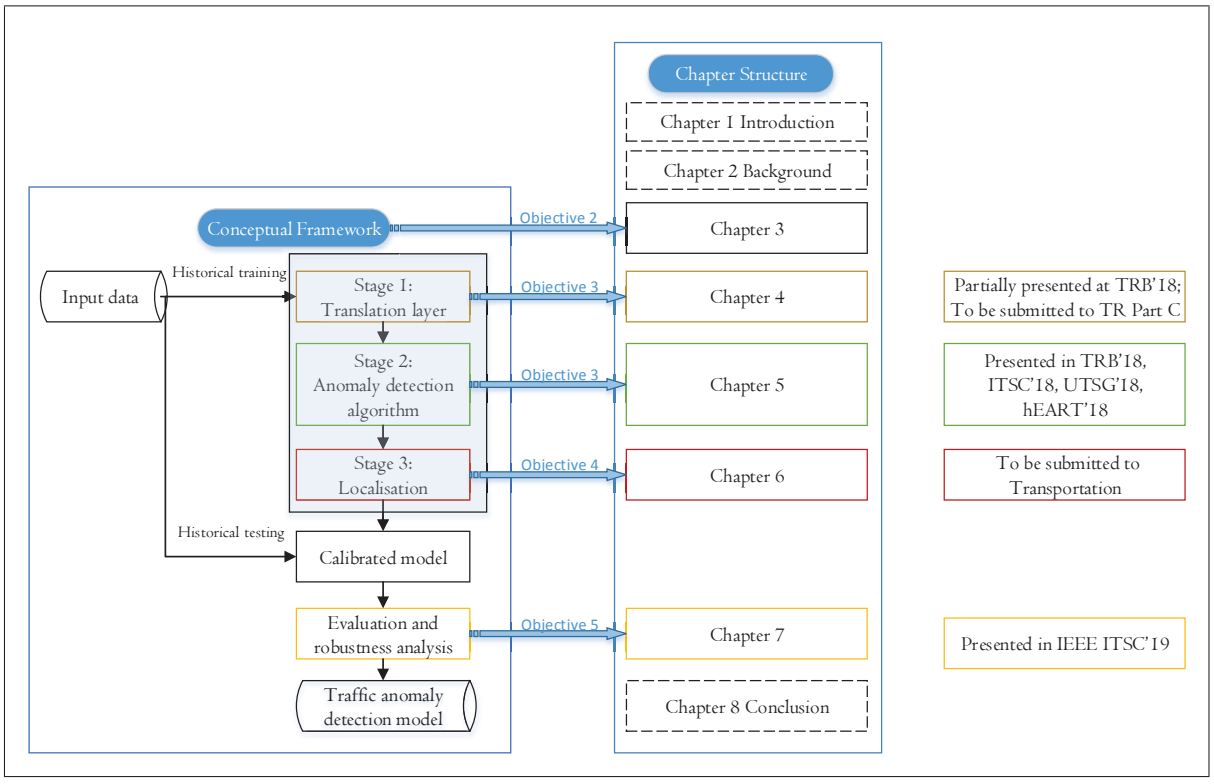


Figure 1.1: Summary of objectives in each chapter with corresponding publications

Chapter 2

Literature Review

The main objective of this chapter is to understand traffic anomalies, the types of anomalies and their corresponding distribution of impacts across an urban traffic network in order to identify the gaps in traffic anomaly detection and thus the current research challenges. Although there is no consensus in the literature regarding a definition of traffic anomaly, in practice anomalies in transport systems are generally related to traffic congestion. The comprehensive understanding of this traffic anomaly will enable traffic managers to identify the problem quickly, proactively react to the traffic disruptions and provide a promise of efficient and automatic traffic management and control. Hence, this section is focused on reviewing a wide range of literature on anomalies, and anomaly detection, both in the traffic research field and in other fields.

The structure of this chapter is as follows. First, section 2.1 will introduce the general understanding of anomaly detection and its application in various subjects. Then, in Section 2.2, the anomaly detection will be narrowed down to traffic related anomalies and the detection of these, before reviewing specific congestion detection techniques or algorithms with their corresponding advantages and limitations in Section 2.3. With the aim of developing traffic anomaly detection at the network level, some potential spatio-temporal models from other research fields are also reviewed in this part. Section 2.4 summarises the existing work, challenges and opportunities.

2.1 Research on Anomaly Detection

2.1.1 Definition of Anomaly

The concept of an anomaly has been defined in a number of different ways depending on the context. A commonly accepted definition of anomaly is from Hawkins (Hawkins, 1980).

"An anomaly is an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism."

An anomaly, therefore, is generally an abnormal behaviour or event that deviates from the standard, expected or normal behaviour. Anomalies can originate from a wide range of factors such as malicious activities or system failures. To some extent, anomalies threaten the reliability and robustness of a system and therefore need to be identified or detected promptly and accurately. Anomalies are important because they can indicate significant and rare events and can prompt critical actions to be taken in a wide range of application domains (Ahmed et al., 2016). The notion of anomaly can be demonstrated by points o_1, o_2 and region O_3 in Figure 2.1. Anomaly detection algorithms can, therefore, be defined as classifying the boundary of normal patterns and then identifying any behaviour that is outside the boundary.

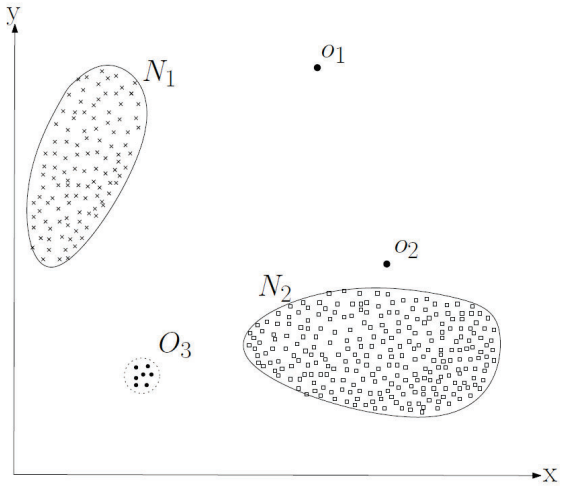


Figure 2.1: Example of anomalies (Source: Hojati et al. 2011)

2.1.2 Types of Anomaly

As discussed in Section 2.1.1, anomalies generally refer to something that deviates from the normal, expected or standard patterns. Before identifying the anomalies, one important step is to understand the nature of the anomaly. In this regard, Ahmed et al. (2016) and Chandola et al. (2009) suggested categorising anomalies according to their characteristics, as follows.

2.1.2.1 Point anomaly

As the name suggests, this type of anomaly refers to a particular data instance which deviates from the normal pattern of the dataset. One example of this kind of anomaly is o_1 or o_2 in Figure 2.1. Point anomalies usually result from random issues that may be caused by a wide-range of factors like different types of system failures.

2.1.2.2 Contextual anomaly

Behaviours or events that act anomalously in a specific context are usually referred to as contextual or conditional anomalies. Contextual anomalies can be characterised by two aspects: (1) the contextual attributes, such as geographical information for spatial data and sequence position for time series data; and (2) behavioural attributes, or direct attributes, such as the traffic flow for traffic incidents. Another property of this type of anomaly is that an identical data instance which has been identified as a contextual anomaly in one situation may be normal in a different situation.

Contextual anomalies have often been investigated using time-series or spatial data. For example, traffic flows on urban roads during accidents are usually lower compared to normal traffic conditions for the time of day and day of the week. On the other hand, low traffic flows may not be contextually anomalous at other times. Defining a contextual anomaly is not straightforward, however, since detecting the anomaly properly and precisely requires data for the relevant contextual and behavioural attributes (Chandola et al., 2009).

2.1.2.3 Collective anomaly

A collection of similar point anomalies based on a group of similarly unexpected behaviours or events is defined as a collective anomaly. Collective anomalies have been commonly investi-

gated with sequence data, graph data and spatial data. For example, the existence of low traffic flow values for an abnormally long period of time indicates an underlying phenomenon corresponding to potential abnormal congestion, while the low value by itself may be not necessarily an anomaly when it occurs in other locations in a sequence. The notion of this type is a set of similar points as shown in O_3 in Figure 2.1.

The difference between a collective anomaly and a point anomaly is, firstly, that collective anomalies are determined by the existence of a series of anomalous data instances, any one of which may not be an anomaly by itself. Point anomalies, on the other hand, can be defined directly without dependence on other similar points. Secondly, individual data instances within a set of collective anomalies are generally related, whereas point anomalies can occur regardless of their position vs other data points. On the other hand, contextual anomalies are basically subject to how the context contributes to the data, so, to some extent, when incorporating some context information, both point and collective anomalies can be classified as contextual anomalies. Figure 2.2 summaries the relationship among three types of anomalies discussed above.

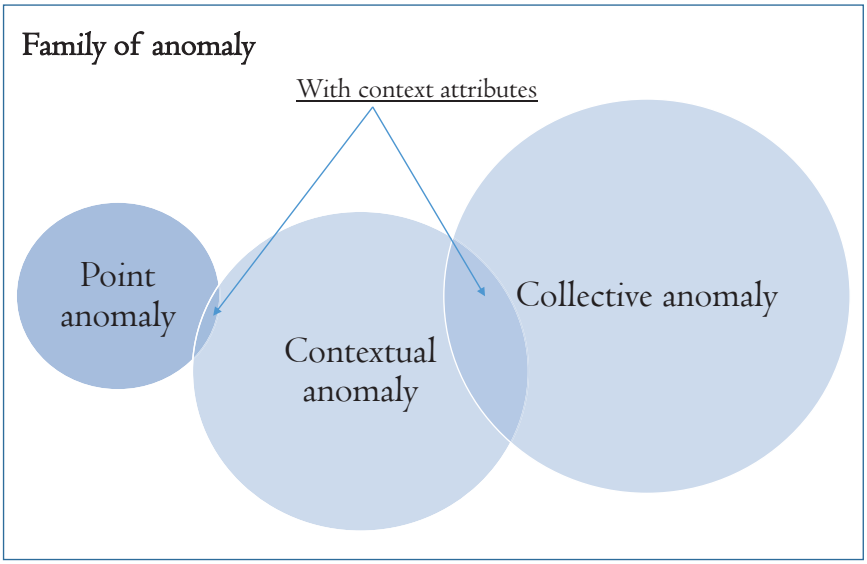


Figure 2.2: Family of anomalies and their interactions

2.1.3 Components of Anomaly Detection Frameworks

Anomaly detection is a methodological framework or algorithm to cluster or classify the original dataset and therefore identify minor or rare abnormal data instances. Basically, as shown in

Figure 2.3, there are three main components to anomaly detection, i.e., input, anomaly detection model and output, where anomaly detection model is the way to map the input to the output. There are a large number of anomaly detection methods used in statistics, machine learning and deep learning. These will be presented in detail in Section 2.3.

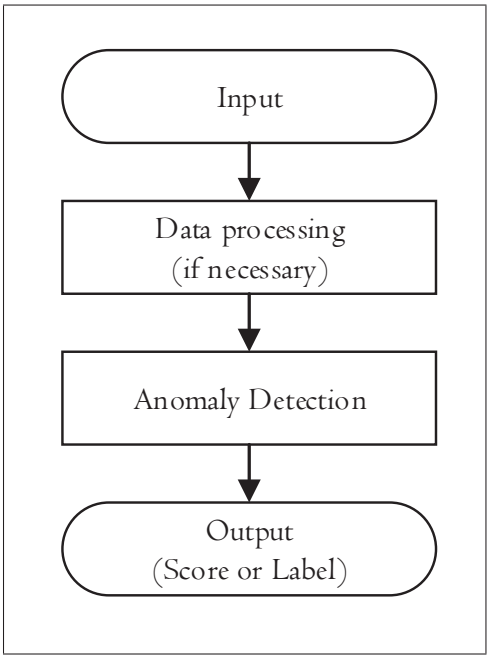


Figure 2.3: Flowchart of anomaly detection

2.1.3.1 Input of anomaly detection

A key factor of any anomaly detection is the nature of the input data, which can be characterised by a set of attributes or features. Inputs with respect to anomaly detection depend on the application or subjects of the problem. There might be different types of datasets, such as credit card transaction records for credit card fraud detection, audio data for speech fault detection and image data for theft detection in videos, etc.

2.1.3.2 Output of anomaly detection

One important aspect of anomaly detection is how anomalies are represented as output. Typically, the output of anomaly detection can be represented in the two following ways (Chandola et al., 2009; Ahmed et al., 2016).

Score Assigning an anomaly score or probability to each instance according to the degree or extent to which the instance is considered an anomaly. The anomalies are then selected by ranking the scores or probability, or by comparing them with a predefined threshold in the light of the specific domain knowledge.

Label As for this type of output, generally, a label is associated with a data instance to denote if that data instance belongs to the normal or anomalous class. In other words, here, the anomaly detection is a binary classifier. For example, points o_1, o_2 and region O_3 in Figure 2.1 would be labelled as anomalies while the remainder are normal. According to the availability of labels, anomaly detection algorithms can be classified into: 1) supervised learning with labels for both normal and anomalous classes; 2) semi-supervised learning with either normal labels or anomalous labels; and 3) unsupervised learning without any labels but subject to the implicit assumptions that normal instances are far more frequent than anomalies.

Considering the different levels of congestion or incidents, the concept of labelling has been used in this research where a probability will be assigned for each level, such as 'congested' and the 'uncongested' for recurrent congestion, or 'with incidents' and 'without incidents' for non-recurrent congestion detection under the assumption of binary classification. This is described in further detail in Chapter 3.

2.1.4 Anomaly Detection Applications and Algorithms

In general, anomaly detection has remained one of the most difficult tasks in data science due to the inherent difficulty of defining and quantifying the notion of anomaly precisely (Agovic et al., 2009). Nonetheless, anomaly detection has extensive applications in areas such as intrusion detection for computer science, health care, military monitoring and surveillance, and fraud detection for bank credit systems and insurance (Chandola et al., 2009). Specifically, there are well recognised implementations of anomaly detection in respect to identifying unexpected network intrusions (Ahmed et al., 2016), robot behaviour (Haddadin et al., 2017), web faults (Wang et al., 2016c) cable faults (Ali et al., 2015) and sensor-based faults (Wu et al., 2018a).

Anomaly detection algorithms have therefore been investigated in various disciplines, including artificial intelligence, deep learning, data mining and pattern recognition. For example,

Agrawal and Agrawal (2015) comprehensively summarised anomaly detection algorithms using data mining techniques to detect abnormal behaviours and provided an understanding of the existing techniques. Agrawal and Agrawal (2015) suggested that hybrid data mining approaches were being widely used because they could provide better results and overcome the drawback of individual approaches. Agrawal and Agrawal (2015) also suggested that the new approaches in the modification of decision tree or kernel-based approaches might yield more accurate results. These algorithms are described further in Section 2.3.

2.1.5 Challenges in Anomaly Detection

Challenges facing accurate anomaly detection include:

- The boundary between normal and abnormal data instances may sometimes be ambiguous, especially for observations at the edge of the normal or abnormal region.
- The normal or abnormal behaviours keep evolving temporally and hence a normal or an abnormal observation in the current state may be insufficient evidence in the future. On the other hand, some behaviours at the spatial level may keep propagating across the network and therefore affecting other activities close by in the network.
- Even though it is common to analyse anomalies using the concept of abnormal data instances, as in Figure 2.1, the definition of anomaly is not universal, especially when considering the variation in application domains. These different definitions make it difficult to transfer or translate anomaly detection techniques directly from one domain to another.
- In most cases, anomaly detection is supervised with labels. The availability of labels traditionally used for training or validation of anomaly detection is a major prerequisite, however.
- It can be very difficult to distinguish between data noise and actual anomalies, since both are data that tend to deviate from the normal region.

Given the challenges as summarised in Figure 2.4, although the anomaly detection problem is implicitly straightforward and easy to understand conceptually, it is difficult to solve in practice

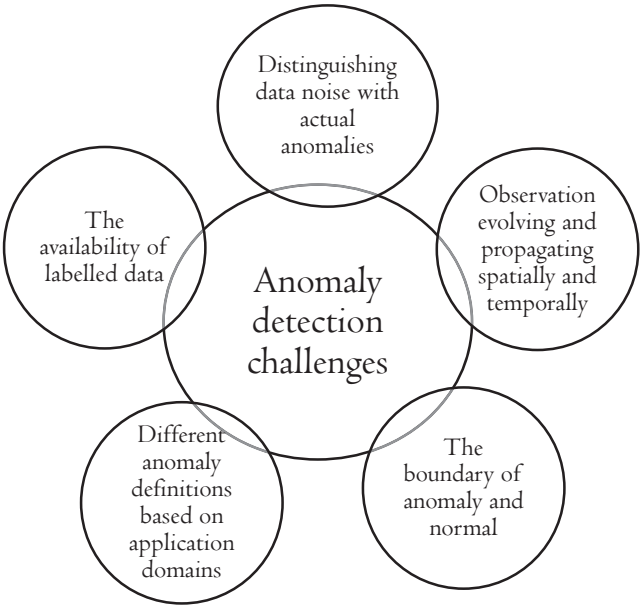


Figure 2.4: Challenges of anomaly detection

(Chandola et al., 2009). The important implications can be summarised as follows. Firstly, it is essential to adopt anomaly detection concepts from diverse disciplines including machine learning, data mining and information theory, according to the nature of the specific domain or problem. Secondly, the nature of a specific anomaly detection problem may be subject to factors such as characteristics of data, the availability of data, existence of data noise, type of anomalies, etc. Finally, since malicious activities or behaviours may evolve both spatially and temporally, the anomaly detection may need to take these spatio-temporal dynamics into consideration to reflect not only the current states but also the immediate future.

2.1.6 Summary of the Research on Anomaly Detection

This section has presented an overview of the literature on anomalies in general and anomaly detection techniques in particular. The systematic understanding of general anomalies must be applied to give insights into traffic anomaly detection. A detailed introduction to traffic anomaly detection including detection of recurrent congestion, early recurrent congestion and non-recurrent congestion will, therefore, be presented in the next section.

2.2 Research on Traffic Anomaly Detection

2.2.1 Background of Traffic Anomaly Detection

In practice, anomalies in the transportation field (i.e., traffic anomalies) are linked to traffic congestion. Generally, traffic congestion can be divided into two types: Recurrent Congestion (RC) and Non-Recurrent Congestion (NRC) (Anbaroglu et al., 2014a). RC refers to day-to-day congestion where traffic demand exceeds the road capacity. RC usually occurs during peak periods, and therefore commuters and network operators can anticipate RC in advance (Deniz and Celikoglu, 2011). With adequate advance notice, one can make plans to potentially mitigate RC and implement suitable intervention measures. On the other hand, NRC is usually caused by unpredictable and one-off events, (e.g., traffic incidents and accidents), planned events (e.g., concerts, football matches and road works) or inclement weather. These events may occur at any time of the day and any location. The congestion duration usually depends on the type of incident and the characteristics of the road network (Lomax and Margiotta, 2003).

To date, many studies have contributed to RC and NRC detection, but some challenges remain to be further investigated. Section 2.2.2 will first summarise the concept of traffic states, which are the essential input for congestion detection, as well as some recent studies into traffic states prediction. In order to identify the gaps and opportunities, this section will review the key studies of RC in Section 2.2.3, early prediction of RC in Section 2.2.4 and NRC in Section 2.2.5, respectively.

2.2.2 Traffic States

Traffic state is one of the fundamental concepts underpinning traffic stream analysis in the traffic engineering domain. Three main parameters characterise traffic states: speed, flow and density. If S is defined as a variable to represent a traffic state, S can be expressed as:

$$S = (v, q, k) \tag{2.1}$$

where:

v = speed, the rate of motion in distance per unit of time (e.g., km/h, m/s)

q = flow, the number of vehicles passing a point in a given period of time (e.g., veh/h, veh/s)

k = density, the number of vehicles occupying a given length of lane/roadway at a specified time instant (e.g., veh/km)

The fundamental relationship between v, q and k is as below.

$$q = kv \tag{2.2}$$

where speed refers to space mean speed, and density can be calculated as a function of detector occupancy ω (i.e., the proportion of time that a vehicle presents on the detector) and the effective vehicle length $\mathcal{L}$ by the traffic flow relationship $k = \omega \,/\, \mathcal{L}$.

Traffic states are essential for traffic engineering as indicators of traffic conditions. The US Department of Transportation has adopted the concept of Level of Service (LoS) for traffic systems and this is defined as "a qualitative measure describing operational conditions within a traffic stream, based on service measures such as speed and travel time, freedom to manoeuvre, traffic interruptions, comfort, and convenience." (Manual, 1965; Manual et al., 2000). For traffic flow, LoS is quantified by comparing the operating density k to jam density k_j in the six ranges shown in Table 2.1 and as illustrated for Greenshields' relationship in Figure 2.5.

Table 2.1: Level of Service ranges

Level of Service	Density k
A	$k < k_j \,/\, 10$
B, C, D	$k_j \,/\, 10 \leq k < k_j \,/\, 3$
E_1	$k_j \,/\, 3 \leq k < k_j \,/\, 2$
E_2	$k_j \,/\, 2 \leq k < 2k_j \,/\, 3$
F	$k_j k \geq 2k_j \,/\, 3$

Traffic state identification is of importance within areas of traffic control and operations and this has become an active research topic, especially for both real-time ITS traffic monitoring and decision-making (Xia et al., 2012). Apart from the practical application in ITS systems, traffic state identification as an indicator of congestion levels has also been used for traffic variable estimation problems (Han et al., 2010).

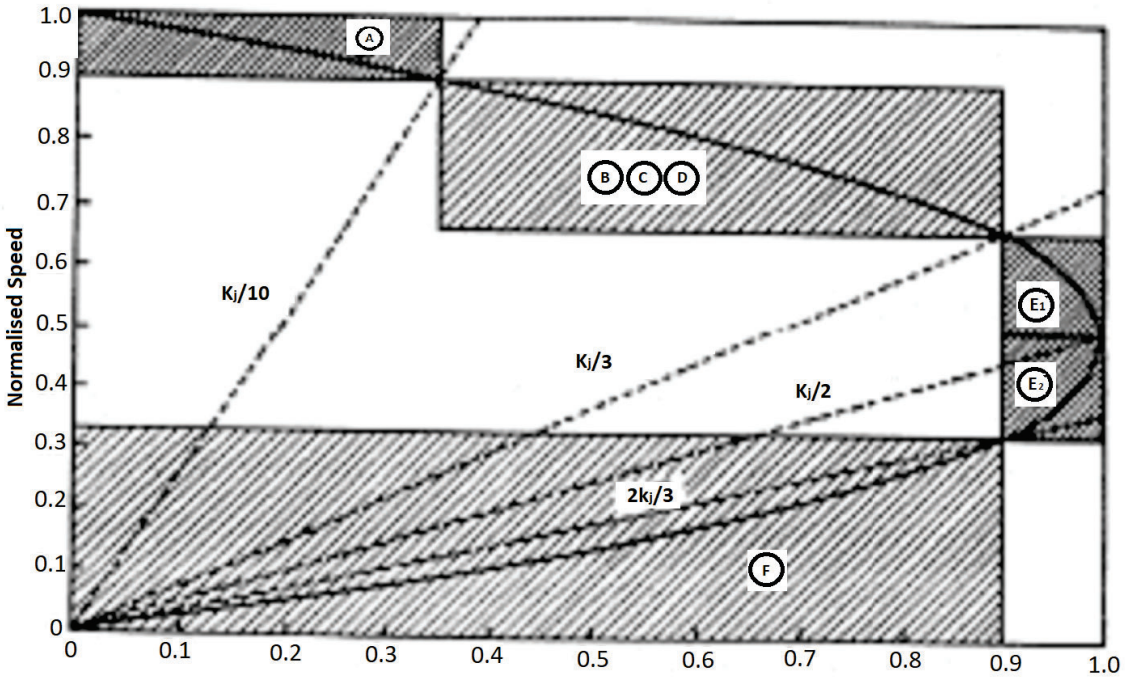


Figure 2.5: Level of service for traffic flow on a road shown on a normalised speed-flow Greenshields fundamental diagram (Source: Gerlough and Huber 1975)

Identifying traffic states refers to estimating all the relevant traffic variables, such as mean speed, density and traffic flow, of the network using real-time traffic data measured by data sensors deployed on the road network (Wang and Papageorgiou, 2005). More importantly, one of the main emphases within traffic state identification is to identify traffic variables that are not directly measured (Wang and Papageorgiou, 2005). For instance, approaches of mapping traffic data, including traffic flow and density, into traffic states have been attempted as inputs for LoS analysis in order to capture the dynamics of varying congestion levels with limited sensor data (Manual, 1965).

A wide range of different traffic state identification methods are available in the literature. A common approach is to use traffic flow, density and speed to identify traffic states by calibrating microscopic fundamental flow-speed-density relationships (Kerner and Rehborn, 1996; Geroliminis and Daganzo, 2008). While this approach is theoretically sound and intuitive, calibrating the fundamental relationship and determining the thresholds between different traffic states are difficult, especially in the absence of large volumes of traffic sensor data (Lu et al., 2015). Another common approach to estimate traffic states is to use clustering methods, such as fuzzy-logic (Zhang et al., 2017), k-nearest neighbours (Oh et al., 2016; Cai et al., 2016), etc.

These methods are effective in identifying traffic states based on historical data and can incorporate other explanatory variables such as time of day and weather information as additional variables.

A major issue of existing state estimation methods is that they lack generalisation and transferability; most of the approaches require auxiliary datasets that are either not readily available in real-time or are not easily transferable across sites (Han et al., 2010). Thus, a method that can estimate traffic states based on readily and transferable information is needed. Moreover, in terms of the context of traffic states estimation, while various methods have been used in the context of motorways, it seems that there are limited studies in practice for urban roads, due to their complex traffic environment.

Given the importance of traffic states identification, some ITS applications have adapted traffic states identification as an index for measuring motorway road conditions quantitatively. Compared to motorways, that have sophisticated management and control systems, however, traffic monitoring (i.e., cameras or loop detectors) in urban roads could be limited by their complex structure and dynamic traffic interactions. With this complexity, traffic congestion in urban roads may need more advanced ITS systems capable of analysing and mitigating traffic congestion effectively across the whole urban road network.

2.2.3 Recurrent Congestion Detection

RC has been extensively studied in recent years because of the necessity to manage urban road networks more effectively (Kong et al., 2016a; Liu et al., 2014). Early RC prediction is important, especially when road capacity is close to demand. It is important to predict RC precisely and as early as possible on these types of networks.

A substantial range of literature has contributed to varying levels of depth to identify and propose models to understand and deal with RC efficiently. At present, however, there is no unified evaluation measure for recurrent congestion (He et al., 2016a). Since recurrent congestion generally occurs due to a lack of supply for the excess demand for road travel, congestion usually leads to a decrease in travel speed and an increase in travel time and delays. These changes in travel variables make it possible to use traffic states as indicators of different severities of traffic

congestion. Congestion can, therefore, be measured in various ways based on the variability of traffic states.

Since these variables are important indexes of LoS and congestion, research efforts have resulted in a substantial number of prediction methods (Andrews Sobral et al., 2013; Yu et al., 2016). For example, An et al. (2016) proposed a three-step RC detection procedure with traffic speed input from Global Positioning System (GPS) equipped taxis; Yu et al. (2016) used Back Propagation Neural Networks (BPNN) to detect traffic congestion based on occupancy and traffic speed and proved that the BPNN was capable of detecting traffic congestion with stable performance; and Zhang et al. (2016) proposed traffic pattern identification as well as congestion detection using dictionary-based compression theory.

A large number of studies have explored the use of traffic states in the RC problem. Traditional RC prediction methods developed to estimate or predict traffic variables such as speed, density, flow, and travel time as well as delay, fall into two main types.

The first type of congestion detection is based on single indicators such as travel speed, directly derived from GPS for vehicular monitoring. For instance, the prevailing traffic speed can be indicative of the quality of traffic congestion and sometimes the severity of congestion (Rao and Rao, 2012). He et al. (2016a) classified traffic congestion in urban road networks into four categories based on its severity by using travel speed index, and used these to measure both the degree of congestion of road segments and networks, respectively. Kong et al. (2016b), meanwhile, incorporated the uncertainty and complexity of traffic states within a Fuzzy comprehensive evaluation to improve the traffic congestion estimation and prediction sufficiently by using traffic flow data.

On the other hand, considering the dynamic nature of traffic, especially in the context of urban road networks, several studies have used multiple attributes of traffic states to evaluate traffic congestion systematically. Nantes et al. (2016) proposed a Kalman Filter method to integrate the heterogeneous measurements from loop detectors, Bluetooth and GPS devices to estimate traffic states in arterial corridors. Yuan et al. (2012) presented a comprehensive traffic state estimator based on data derived from loop detectors and vehicle trajectories.

It is worth noting, however, that, over the last few years, while many studies have attempted

to predict or estimate traffic congestion on motorways or highways (e.g., Mazzenga and Demetsky, 2010 and Yildirimoglu and Geroliminis, 2013), the application of RC detection in the context of urban networks has been less well studied. This is due to their greater topological and control complexity and their vulnerability to a wider range of sources of interruption (e.g., intersections, adaptive traffic signal controls, pedestrian crossings and on-street parking) (Anbaroglu et al., 2014a).

In general, traffic congestion has previously been characterised as a clustering problem in terms of both theoretical and empirical aspects. Theoretical studies investigate the movement of vehicles at critical traffic states (Sugiyamal et al., 2008) while the empirical research focuses on making use of the spatio-temporal clusters to detect congestion patterns (Anbaroglu et al., 2014a). The existing literature has proposed important approaches to evaluate the traffic state and comprehensively understand traffic recurrent congestion. Gaps exist, however, firstly in estimating the traffic states at a network level, and, secondly, in achieving detection by incorporating spatio-temporal information.

2.2.4 Early Prediction of Recurrent Congestion

In road networks where demand closely matches the supply, RC may or may not occur during peak times on certain days. This is because traffic is a stochastic process and the temporal pattern of demand over the course of a given day shows day-to-day variability. The realisation of traffic demand on any given day may give rise to RC. Accurate early detection of the onset of RC in such networks enables traffic managers to formulate and implement proactive intervention measures to reduce the level of severity of congestion. Hence, the capability of early detection of RC is of immense practical value to road network managers.

Traditional RC prediction is commonly based on short-term traffic states prediction. Vlahogianni et al. (2014) has provided an important review on this. Most effort in the past decades has focused on predicting traffic volume or travel time using data collected from single point sources in motorways and freeways (Vlahogianni et al., 2014). To date, previous studies tend to focus on analysis and prediction of traffic states either once it has occurred or over relatively limited prediction horizons, typically a 5-minute or 15-minute time window. This short-term perspective provides little time for traffic operators to formulate and deploy management plans

proactively, and can even lead to the failure to accommodate the communication latency of various traffic sensor devices.

Little attention has been paid to the development of methods that can provide significant early warning of the formation of congestion and the characteristics of its spatio-temporal evolution (Ma et al., 2017). Such a method would substantially reduce the time-constraints affecting traffic operators and provide them with an RC index for proactive reactions. Hence, a more effective RC early detection method which can infer the onset of RC directly is desirable. At the same time, the recent developments in computational power and mathematical models enable researchers to expand prediction horizons (Vlahogianni et al., 2014). The impact of these will be further presented and reviewed in Section 2.3.1.

There are some obvious differences, however, between short-term traffic prediction and early RC prediction or detection. First, short-term traffic prediction is by nature a regression problem, with the output of continuous numbers such as traffic speed, travel time and traffic flow in a time series format (Ma et al., 2017). In comparison, the early RC prediction problem is a classification problem which can provide labels such as of severe level of traffic congestion. In practice, traffic managers may be interested in knowing whether that day is a good day (with smooth traffic) or a bad day (with congested traffic) with an early indicative index. Even though powerful deep learning or machine learning methods have been developed recently for short-term traffic prediction, no research has focused on the early RC detection problem.

2.2.5 Non-Recurrent Congestion Detection

NRC is usually caused by unpredictable and one-off events, (e.g., traffic incidents), planned events (e.g., concerts, football matches and road works) and inclement weather. In general, an entire process of traffic incident management is composed of detection, verification, response, information dissemination, on-site traffic management, investigation and clearance (Teng and Qi, 2003). Among all these components of incident management, a quick and effective response strategy relies on employing an accurate incident detection algorithm (Teng and Qi, 2003; Zhang et al., 2018b). To reduce substantial delays or traffic congestion, therefore, a reliable and efficient traffic incident detection algorithm is a desirable function for traffic management and control systems. It has been widely accepted that effective traffic detection methods, alongside the

implementation of proactive mitigation strategies, can contribute significantly to reducing the impact of incidents (Zhang et al., 2018b). Furthermore, as a key function of intelligent transport systems, reliable and prompt detection of traffic incidents is critical to enable traffic managers to alleviate congestion proactively under different levels of abnormal traffic conditions (He et al., 2013).

Over the past few years, a sizeable body of literature has emerged with the objective of using discrete vehicle trajectories or aggregated traffic patterns for traffic incident detection. This data is acquired by existing and emerging traffic sensors such as loop detectors, surveillance cameras and GPS (Yuan and Cheu, 2003). Generally, discrete vehicle trajectories are used for microscopic behaviour based detection, while aggregated traffic patterns usually represent features of vehicle groups during a specific time span and thus have been extensively used in the context of traffic control and management (Yuan and Cheu, 2003).

Traditional traffic incident detection methods have been proved valid in many applications (Asakura et al., 2017; Hawas, 2007). There is a vast body of literature devoted to modelling traffic incidents and automatic incident detection over the last few decades. Traffic incident detection was first developed in the early 1970s based on the occupancy measures at fixed road sections and diagnosed traffic incidents by comparing traffic variables on the upstream and downstream and then observed the effects of the incident on traffic (Payne and Tignor, 1978; Highways Agency, 2005). This type of statistical algorithm detects significant differences between observed data and traffic characteristics predicted by prior probability or by identifying the outliers based on the principle of standard normal deviation (Baiocchi et al., 2015).

Another type of statistical technique is based on recognising an abnormal pattern by using a fundamental flow-speed-occupancy diagram, such as the well-known McMaster algorithm (Persaud et al., 1990). The performance of all the aforementioned algorithms, however, lies in the accuracy of the thresholds chosen for identifying traffic incidents (Baiocchi et al., 2015).

Despite their effectiveness in detecting traffic incidents on any given arterial road, traditional detection methods still present certain challenges. Firstly, many methods have been developed and tested in the context of highways where the topology and traffic patterns are rather stable or simple compared to urban networks (He et al., 2013; Asakura et al., 2017). Accordingly, traffic detection within the whole urban networks has been rarely discussed in the

literature (Anbaroglu et al., 2014b). Secondly, most studies that are focused on individual links have the limitation of low transferability (Zhang et al., 2016). For example, a traffic incident detection algorithm that is effective in triggering an accurate alarm on typical roads may only work well in the same or similar traffic patterns based on which its parameters were calibrated. Thus, it seems impossible for one detection method to perform well under a wide range of different traffic patterns on different types of road networks. In other words, there is a need to develop a single method of traffic incident detection for application over the whole urban road network rather than applying different detection algorithms separately for different types of links.

Given that traffic pattern changes caused by traffic incidents cannot evolve arbitrarily in space and time, recently a couple of data-driven models making use of state-of-the-art artificial intelligent techniques, have been developed to address these challenges. These include the multiple model particle filter (Wang et al., 2016b), fuzzy systems (Hawas, 2007) and Support Vector Machine (SVM) (Yuan and Cheu, 2003), and each takes into account both spatial and temporal traffic data at the network level. For instance, Olutayo and Eludire (2014) introduced a hybrid machine learning classifier that combined neural networks and decision trees to analyse traffic incidents. Zhu et al. (2018a) applied convolutional neural networks to detect traffic incidents by incorporating spatial correlations captured by a connectivity matrix among neighbouring edges in a simulated road network. Gu et al. (2016) employed a dictionary-based method to map Twitter data into a high dimensional binary vector and then identify the traffic incidents based on high-dimensional feature spaces. The underlying principle of these data-driven methods is to analyse versatile measures or data sources so as to recognise the changes in patterns as evidence of the possible occurrence of an incident. Although a wide range of important contributions are found in the literature, most studies with the exception of Bao et al. (2019) have been devoted to motorway contexts while limited attention has been paid to urban networks because of their aforementioned complexity.

2.2.6 Summary of the Research on Traffic Anomaly Detection

Generally, efforts of traffic anomaly detection can be categorised into roadway-based algorithms, probe-based algorithms, arterial-application algorithms, sensor fusion-based algorithms and driver-based algorithms (Parkany and Xie, 2005). These anomaly detection algorithms sum-

marised in Table 2.2 have been used for tasks including RC detection and NRC detection.

Table 2.2: Summary of traffic incident detection algorithms

Algorithms	Description	Category	References
Roadway-based Algorithms	These algorithms applied in the context of highways based on loop detector data.	Statistical Algorithms; Logic Algorithms; Artificial Intelligence based Algorithms	Xu et al., 2016 Xu et al., 2013 Lu et al., 2012
Probe-based Algorithms	Using travel time and the traffic variables collected by probes to detect incidents in the highways or arterial streets.	Statistical Algorithms; Statistical Algorithm; Bayesian Model	Niver et al., 2000 Hellinga and Knapp, 2000 Park and Haghani, 2016
Sensor Fusion-based Algorithms	Using multiple data sources to detect traffic incidents.	Random Forest; Treiber-Helbing Filter; Support Vector Machine	Dogru and Subasi, 2018 Houbraken et al., 2015 Zhang and He, 2016
Arterial-application Algorithms	Detection of road incidents that results in non-recurrent jam on urabn roads.	Pattern Matching Algorithms; Random Forest Classifier; Probabilistic Topic Algorithms; Fuzzy Neural Network Algorithm; Logit Algorithm; Nueral Networks; Bayesian Model	Habtemichael et al., 2015 Dogru and Subasi, 2018 Kinoshita et al., 2015 Tang et al., 2017 Rossi et al., 2015 Ki et al., 2018 Sun and Sun, 2015
Heterogeneous Data-based Algorithms	Using new emerging data sources to detect traffic incidents.	Semi Naive Bayes (SNB) Classifier ; Regression Model	Gu et al., 2016 Steenbruggen et al., 2016

The differences between RC detection and NRC detection can be summarised as follows. First, the data used for RC and NRC are different. Although both methods need traffic variables as input, NRC needs the additional contextual information added explicitly by external systems/processes, such as the traffic incidents. Second, since there are 'congested' or 'uncongested' labels unavailable for RC, an extra label generation step is needed before the RC detection. Finally, RC is caused by overloaded demand and thus for RC early prediction in a period of time ahead may be possible whereas NRC cannot be predicted in advance.

There are two major challenges that remain unsolved. One is how, in the context of a complex and constantly changing traffic demand profile, to improve the accuracy of RC and NRC detection by considering the spatio-temporal propagation of RC and the magnitude of demand at a whole network level, as identified in Section 2.2.3 and in Section 2.2.5 respectively. The other one is the need to detect the onset of RC before its occurrence or to be able to give an early stage warning, as concluded in Section 2.2.4. More robust solutions in these areas would provide valuable information for traffic engineers to mitigate congestion more effectively in advance. In

the next section, a number of detection algorithms will be reviewed and summarised, together with the recent advances in computer science.

2.3 Detection Algorithms

To address the aforementioned challenges, it is essential to review existing anomaly detection algorithms. In order to extend the current research into spatio-temporal field, some statistical approaches and machine learning methods will be summarised in Section 2.3.1. Apart from the existing methods, a comprehensive review of potential methods will also be presented. After reviewing the existing statistical and machine learning methods, therefore, Section 2.3.2 will summarise the recent advances in deep learning before reviewing the evaluation methods for detection algorithms in Section 2.3.3.

2.3.1 Review of Anomaly Detection Algorithms

Generally, the family of anomaly detection algorithms can be classified into four categories as it is in the surveys by (Ahmed et al., 2016; Chandola et al., 2009): classification-based algorithms, clustering-based algorithms, statistical algorithms and information theory.

Classification and clustering algorithms are two main branches in machine learning, considered as supervised learning and unsupervised learning respectively according to whether or not they make use of pre-labelling. Classification-based algorithms rely on extensive expertise from relevant experts, such as in respect to the characteristics of anomalous patterns and their differences compared to the normal activity profile, as well as the availability of anomalous and normal datasets. On the other hand, clustering-based algorithms group similar data instances and do not need pre-labelled anomalies. To use clustering-based algorithms, it is necessary to accept the following key assumptions: 1) noise is considered as anomaly; 2) normal data instances lie close to the nearest cluster's centroid while anomalies are far away from the centroid; and 3) the density or the size of a cluster below a threshold is considered anomalous.

Statistical methods are usually based on some data statistics or underlying distribution, while information theory tries to distinguish the anomaly from the normal by understanding their

underlying characteristics and mechanisms. All algorithms have been summarised according to the above four categories, as shown in Figure 2.6.

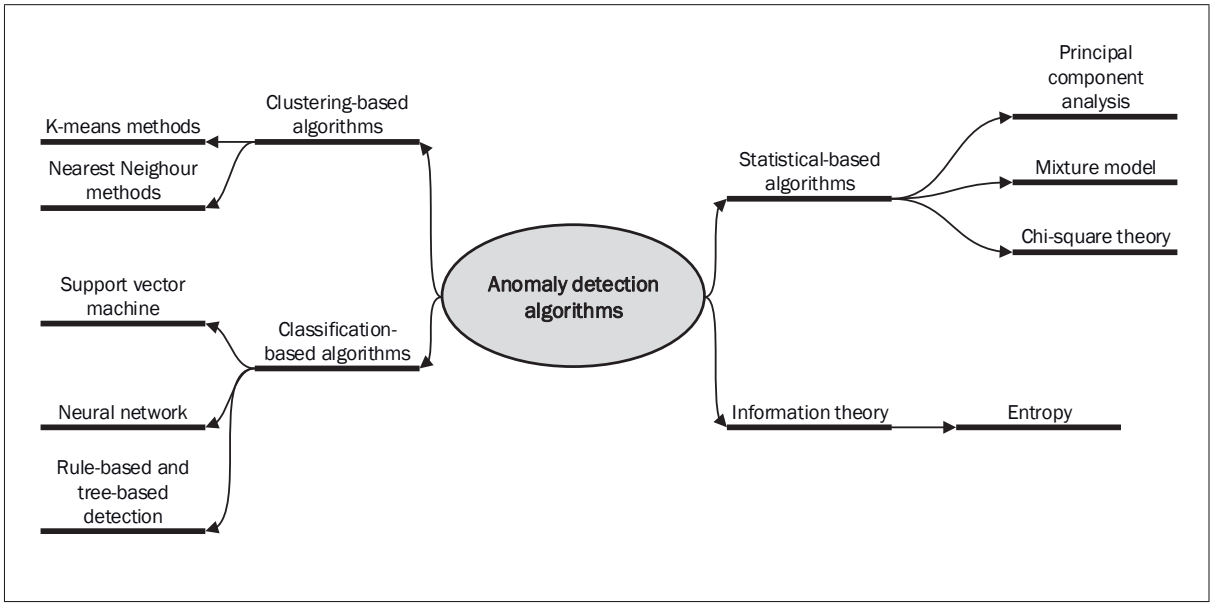


Figure 2.6: Categories of anomaly detection algorithms

The algorithm types listed in Figure 2.6 will be reviewed in this section. Different detection algorithms of varying complexity, principles and data requirements can perform differently when the data conditions are changed. The types of anomaly detection algorithms together with their pros and cons are summarised after the review.

2.3.1.1 Classification-based Algorithms

Support Vector Machine The basic principle of a SVM is to derive a hyper-plane that maximises the margin of separation in feature space between the negative and positive class (Cortes and Vapnik, 1995). Although the standard SVM algorithm involves supervised learning with labelled data, it is able to be adapted as a clustering method since it is ultimately derived from unsupervised learning (Ahmed et al., 2016).

SVM has been found to be one of the most powerful classification methods and has been used to detect anomalies by arriving at a binary classification based on a region or boundary learnt in the training dataset (Vapnik, 2013). The application of SVM in anomaly detection can be seen in Table 2.3. SVM has the advantages of providing a good out-of-sample generalisation,

robustness and convex optimisation but suffers the disadvantages of a lack of transparency of results and sensitivity to over-fitting the model selection criterion (Auria and Moro, 2007).

Since, in practice, however, training data often contain unexpected noise which invalidates the main assumptions of the SVM that all the sample data from training are independently and identically distributed (IID), the standard SVM may result in poor generalisation and regularisation.

Neural Network Artificial Neural Networks (ANNs) are inspired by the structure of biological networks and consist of neurons and multiple layers, i.e., an input layer, hidden layer and output layer, as shown in Figure 2.7. ANNs are designed to address complex linear or nonlinear problems where the interrelationships among inputs are not well defined and understood (Zhang et al., 1998). ANNs have many applications in respect to anomaly detection (Mirza and Cosan, 2018) and traffic incident detection (Ki et al., 2018) for either multiple or binary classifications.

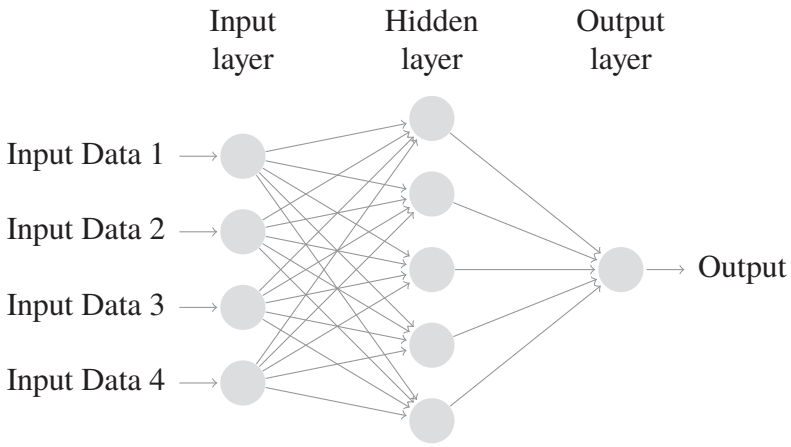


Figure 2.7: Topology of Neural Networks

The workflow of ANNs can be generalised as in Figure 2.8. The data are usually divided into training and testing datasets. The training datasets are used to train the initial network compared with the labelled data, and some stochastic gradient methods or loss functions are used to optimise the weights in order, for example, to conform to different normal classes. As the last step, the testing data instance is fed with optimised weights and bias into an activate function which is usually a logic function to decide if the testing instance is normal or not (Taylor and Addison, 2000; De Stefano et al., 2000).

One type of ANNs is Multiple Layer Perceptron (MLP). MLP is a class of feed-forward

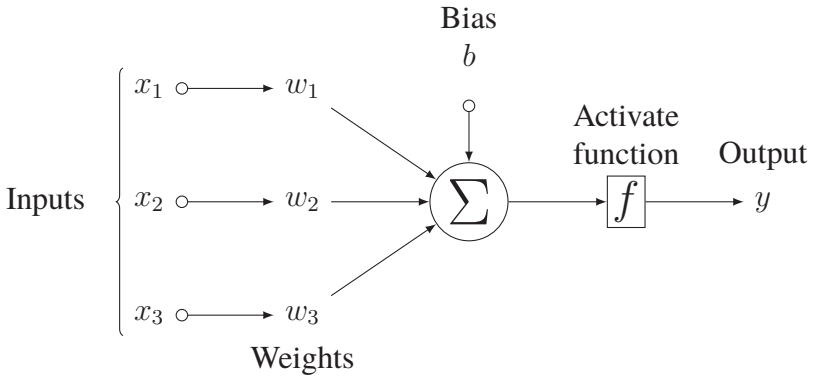


Figure 2.8: Workflow of Neural Network

artificial neural network which is inspired by the structure of biological networks and consists of neurons and multiple layers, i.e., input layer, hidden layer and output layer (Haykin, 1994). Apart from the input layer, each layer contains a neuron which is responsible for storing the acquired knowledge and operating a nonlinear transformation by using an activation function, such as a sigmoid function (Haykin, 1994). It is commonly used to solve difficult computational tasks like predictive modelling tasks or complex non-linear problems where the interrelationships among variables are not well understood. The predictive capability of MLP comes from the hierarchical structure where the features from various scales can be learned and remapped into high-dimensional features in order to relate with the target output (Cortez et al., 2012). MLP often treats anomaly detection in relation to a typical recognition problem (Hawas, 2007; Olutayo and Eludire, 2014; Lu et al., 2012) and thus has been selected as a representation of the application of a machine learning technique in traffic congestion detection.

Rule-based and Tree-based Detection The basic idea of rule-based detection is to learn the pattern of the normal behaviour in a system and consider everything that is not encompassed within the normal pattern as anomalous. The rule-based algorithms can be used in both binary classification and multiple classifications. In contrast to binary classification, where all data instances are classified into two disjointed classes, multiple classification maps a given independent and identically distributed training set: $(S = (x_i, y_i); 1 \leq i \leq n)$ consisting of n training instances $(x_i \in x, y_i \in y)$ by using a multi-label classifier $(n : x \rightarrow y)$ via a specific cost function.

Tree-based learning algorithms are considered to be one of the most-used supervised learn-

ing methods since they empower predictive models with high accuracy, stability and ease of interpretation (Pal, 2005). The high predictive power and accuracy arise from their inherent capability to learn non-linear relationships. The tree-based algorithms include decision trees (Quinlan, 1986), random forest (Breiman, 2001) and gradient boosting classifier (Friedman, 2001) and so forth.

For example, Random Forest (RF) was first introduced by Breiman (2001) as an ensemble learning method whose mainly underlying principle is to group a couple of methods together so as to improve the accuracy of classification or prediction. Specifically, RF starts with a standard machine learning technique, i.e., decision tree, and constructs a multitude of decision trees to output class labels of classification. RF has competitive advantages in respect to fast training time and its effectiveness in dealing with unbalanced and missing data, and hence it has been extensively used in the literature and has proved its superiority to detect traffic anomalies compared to normal decision tree classifiers (Liu et al., 2013).

On the other hand, Gradient Boosting Classifier (GBC) is one of the major and powerful classification machine learning methods first introduced in 2001 (Friedman, 2001) and designed to search a prediction model in the form of an ensemble of prediction trees. It builds the model by optimising a non-specific differentiable loss function in a stage-wise boosting fashion. GBC has been used in the transportation research domain because of its relatively good performance compared with other normal ensemble methods. For example, GBC has been applied to improve the travel time prediction (Zhang and Haghani, 2015) and crash prediction on urban roads (Ahmed and Abdel-Aty, 2013).

2.3.1.2 Clustering-based Algorithms

Another type of algorithm in machine learning are the clustering-based algorithms, where there are usually no labels available during the training process. Two methods, K-means and nearest neighbour, will be presented in this section.

K-means Clustering K-means clustering (MacQueen et al., 1967) is a method that partitions data instances into different clusters where each data instance belongs to the cluster with the

nearest mean, serving as a centroid of the cluster. This algorithm usually begins with a pre-specified number of clusters K and initialised K centroids, and assigns all data instances to the centroids before re-calibrating the centroids until finding the optimal ones.

Underpinning K-means clustering are the following assumptions.

- A data instance is treated as an anomaly either if the distance between an instance and centroid is larger than a predefined threshold; or
- if its distance to the normal cluster centroid is larger than the predefined thresholds.

In general, the essence of K-means algorithms lies in a distance function to compute the distance or similarity between two objects. Among all the existing distance metrics, the Euclidean distance has been the most commonly used, as Eq. 2.3.

$$d(x, y) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2} \tag{2.3}$$

where $x = (x_1, \ldots, x_m)$ and $y = (y_1, \ldots, y_m)$ are two input vectors with m features. It should be noted that in the Euclidean distance, all features are equally measured, even for the attributes in different scales, so it is necessary to normalise all features before applying the algorithms. The basic Euclidean distance, however, can be easily extendable to prioritise some features by the weighted Euclidean distance by adding the weighting factor s_i for the ith features, as follows.

$$d(x, y) = \sqrt{\sum_{i=1}^{m} (\frac{x_i - y_i}{s_i})^2} \tag{2.4}$$

Nearest Neighbour-based clustering The nearest neighbour algorithms (Dasarathy, 1991) build a predictive model and map the patterns into different classes by comparing the features of the patterns which have already been classified. For the past decade, the concept of nearest neighbour analysis has been used intensively in anomaly detection techniques. In contrast to K-means algorithms, the key assumptions for this algorithm are that normal data instances occur in dense neighbourhoods, while anomalies occur much further from their closet neighbours.

Typically, the nearest neighbour algorithm requires a similarity or distance measure defined between two data instances. Due to its sole dependence on distance or similarity, it is particularly effective in the situations when the distributions of the patterns and categories of patterns are unknown (Cai and Chou, 2003). Moreover, it is easy to use as well as having a low probability of errors.

The basic notion of this algorithm can be described as follows. Assume that there are N samples $(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N)$ which have been classified into k categories $(1, 2, \ldots, k)$. The generalised distance between $\mathbf{X}$ and $\mathbf{X}_i$ where $i \in [1, N]$ is defined as:

$$D(\mathbf{X}, \mathbf{X}_i) \;=\; 1 - \frac{\mathbf{X} \cdot \mathbf{X}_i}{\|\mathbf{X}\| \, \|\mathbf{X}_i\|} \tag{2.5}$$

where:

$\mathbf{X} \cdot \mathbf{X}_i$ = the dot production of vector $\mathbf{X}$ and $\mathbf{X}_i$
$\|\mathbf{X}\|$ = the modulus of $\mathbf{X}$
$\|\mathbf{X}_i\|$ = the modulus of $\mathbf{X}_i$
$D(\mathbf{X}, \mathbf{X}_i)$ = the generalised distance within the range of $[0, 1]$

According to the nearest neighbour principle, the data instance $\mathbf{X}$ is predicted as belonging to the same category as of $\mathbf{X}_k$, when the generalised distance between $\mathbf{X}$ and $\mathbf{X_j}$ ($j = 1, 2, \ldots, N$) is the smallest; i.e.,

$$D(\mathbf{X}, \mathbf{X}_k) \;=\; \min\{D(\mathbf{X}, \mathbf{X}_1), D(\mathbf{X}, \mathbf{X}_2), \ldots, D(\mathbf{X}, \mathbf{X}_N)\} \tag{2.6}$$

The distance between two data instances can be computed in different ways, such as Euclidean distance in Eq. 2.3 for continuous attributes and a simple matching coefficient for categorical attributes. Similar to the K-means algorithms, a key advantage of nearest neighbour is that they are unsupervised and purely data-driven and hence do not make any assumptions regarding the generative distribution of the data. Moreover, adapting the nearest neighbour algorithm to a different data type is very straightforward, primarily requiring just the definition of an appropriate distance measure for the given data. Nevertheless, it has limitations such as being computational expensive since it involves computing the distance of each data instance

to its nearest neighbours. The second limitation arises from its dependence on a distance measure, since defining distance measures between instances can be challenging when the data is complex, such as with graph data (Chandola et al., 2009).

2.3.1.3 Statistical Algorithms

Various other techniques such as Chi-square theory, probability mixture models and principal component analysis, have been developed based on the principles of statistical theory to detect anomalies.

Chi-square Theory Chi-square theory has been used for anomaly detection by Ye and Chen (2001) to detect a large departure of events from normal by using a distance measure based on the chi-square test statistic as

$$\chi^2 = \sum_{k=1}^{n} \frac{(O_k - E_k)^2}{E_k} \tag{2.7}$$

where:

O_k = the observed value of the kth variable
E_k = the expected value of the kth variable
n = the number of variables

χ^2 has a low value when an observation of the variable is near the expected, Following the $\mu \pm 3\sigma$ rule, when an observation, χ^2 is greater than $\overline{\chi^2} + 3S_{\chi}^2$ it is considered an anomaly.

Mixture Model Eskin (2000) proposed a mixture model for detecting anomalies and distinguished all data instances into two classes based on their probabilities: 1) abnormal events with a small probability λ; and 2) Normal events with majority probability of $1 - \lambda$.

In the context of the mixture model, these probability distributions are termed as *majority* (**M**) and *anomalous* (**A**) distributions. These two probability distributions form the generative distribution for the data **D** as

$$D = (1 - \lambda)M + \lambda A \tag{2.8}$$

Principal Component Analysis Principal Component Analysis (PCA) (Wold et al., 1987) extracts the dominant patterns in a data matrix in terms of a complementary set of scores. The application of PCA in the anomaly detection field has several benefits including the absence of assumptions about statistical distribution, significantly decreased dimensions without losing information and finally low computational complexity (Shyu et al., 2003).

It is generally a linear combination of p random variables $(A_1, A_2, \ldots, A_p)$ that are: 1) uncorrelated; 2) with variances sorted in order from high or low; and 3) a total variance equal to the variance of the original data.

The mathematical formulation of PCA can be expressed as Eq. 2.9.

$$y_i = e_i(x - \bar{x}) = e_{i1}(x_1 - \bar{x_1}) = e_{i2}(x_2 - \bar{x_2}) + \cdots + e_{ip}(x_p - \bar{x_p}) \tag{2.9}$$

where:

y_i = the ith principal component
i = $1, 2, \ldots, p$ and $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_p \geq 0$
$\mathbf{A}$ = an $n \times p$ data matrix of n observations on each of p variables $(A_1, A_2, \ldots, A_p)$
S = a $p \times p$ sample co-variance matrix of $A_1, A_2, \ldots, A_p$

Shyu et al. (2003) adapted PCA for anomaly detection by assuming that the number of normal instances is much higher than that of anomalies. A data instance (x) will be classified as an anomaly if

$$\sum_{i=1}^{q} \frac{y_i^2}{\lambda_i} > c_1 \quad \text{or} \quad \sum_{i=p-r+1}^{p} \frac{y_i^2}{\lambda_i} > c_2 \tag{2.10}$$

and is normal if

$$\sum_{i=1}^{q} \frac{y_i^2}{\lambda_i} \leq c_1 \quad \text{or} \quad \sum_{i=p-r+1}^{p} \frac{y_i^2}{\lambda_i} \leq c_2 \tag{2.11}$$

where c_1 and c_2 are outlier thresholds for creating a specific alarm α_1 and α_2.

$$\alpha = \alpha_1 + \alpha_2 - \alpha_1\alpha_2 \tag{2.12}$$

where

$$\alpha_1 = P(\sum_{i=1}^{q} \frac{y_i^2}{\lambda_i} > c_1 | x \text{ is a normal instance}) \tag{2.13}$$

and

$$\alpha_2 = P(\sum_{i=p-r+1}^{p} \frac{y_i^2}{\lambda_i} > c_2 | x \text{ is a normal instance}) \tag{2.14}$$

2.3.1.4 Information Theory

Another branch of anomaly detection belongs to information-theoretic measures, where entropy is one commonly used measure originating from the concept of *entropy*. Entropy was proposed by Clausius (1867) in the early 1850s to measure the disorder in thermodynamic systems, while later on Shannon (1948) adopted the concept of entropy to information theory. The definition of entropy in information theory is a measure of the uncertainty associated with a random variable (Bereziński et al., 2015). The underlying principle is that the more random the variable, the bigger the entropy and vice verse.

Assume that in a dataset D in which each data item belongs to a class ($x \in C_D$), the entropy of D relative to the $|C_D|$ - wise classification is defined as

$$H(D) = \sum_{x \in C_D} P(x) \log \frac{1}{P(x)} \tag{2.15}$$

where $P(x)$ is the probability of x in D.

One adaption of entropy, given that Y is the entropy of the probability distribution $(P(x|y))$, is conditional entropy, as follows.

$$H(D|Y) = \sum_{x,y \in C_D, C_Y} P(x,y) \log \frac{1}{P(x|y)} \tag{2.16}$$

where $P(xy)$ is the joint probability of x and y and $P(x|y)$ the conditional probability of x given y.

Other extensions of entropy include the relative entropy between two probability distribu-

tions $p(x)$ and $q(x)$ defined over the same $x \in C_D$ as

$$relEntropy(p|q) = \sum_{x \in C_D} P(x) \log \frac{p(x)}{q(x)} \tag{2.17}$$

and relative conditional entropy which is defined between two probability distribution ($p(x|y)$ and $q(x|y)$) over the same $x \in C_D$ and $y \in C_Y$ as

$$relCondEntropy(p|q) = \sum_{x,y \in C_D, C_Y} P(x,y) \log \frac{p(x|y)}{q(x|y)} \tag{2.18}$$

Entropy-based anomaly detection has been used intensively in the past. For example, Bereziński et al. (2015) proposed an entropy-based anomaly detection algorithm to detect modern malware based on anomalous patterns in the network.

2.3.1.5 Anomaly Detection Summary

The general detection algorithms, together with their applications, have been summarised in Table 2.3. The application areas include fraud detection, industrial damage detection, cyber-intrusion detection, textual anomaly detection, medical detection, image processing, sensor networks and other fields as mentioned previously in Section 2.1.4. Besides the areas listed in Table 2.3, there are some other emerging areas, such as robot behaviour detection (Haddadin et al., 2017), money laundering detection (Dreżewski et al., 2015), web fault detection (Wang et al., 2016c) and traffic surveillance (Bhuyan et al., 2016) and sensor network detection for healthcare (Haque et al., 2015). On the other hand, for the detection algorithms in Table 2.3, a wide range of algorithms including parametric methods (e.g., statistical profiling and Bayesian networks) and non-parametric methods (e.g., neural networks and nearest neighbour method) have been summarised.

Table 2.4 summarises the advantages and constraints for anomaly detection. These include, for classification-based algorithms, the need for supervised models with labelled instances of anomalies, while other algorithms can be used as unsupervised learning without labelled instances. Classification-based algorithms and clustering-based algorithms are quite fast in the testing phase because each test instance is compared with the pre-trained model or a small num-

ber of clusters. Other methods, however, are slow in both the training and test phases, especially when training the neural networks (Chandola et al., 2009).

Table 2.3: Summary of anomaly detection applications and corresponding techniques

Applications	Description	Detection Algorithms	References
Intrusion Detection	Detection of malicious activity in a computer related system; textbackslash	Statistical Profiling; Neural Networks; Mixture of Models; Rule-based Systems; Support Vector Machine; Statistical Model Bayesian Networks; Clustering Based Detection Nearest Neighbour; Spectral; Information Theory	Resende and Drummond, 2018 Mirza and Cosan, 2018 Moustafa et al., 2018 Herrera-Semenets et al., 2018 Shams and Rizaner, 2018 Moustafa and Slay, 2016 Kabir et al., 2017 Bostani and Sheikhan, 2017 Lin et al., 2015 Ma et al., 2016 Noble and Cook, 2003
Fraud Detection	Detection of commercially criminal activities, such as mobile phone call fraud, credit card fraud and insurance fraud	Neural Networks; Rule-based Systems; Clustering Based Detection; Statistical Profiling; Parametric Statistical Model; Information Theory	Fu et al., 2016 Chen et al., 2015 Nian et al., 2016 Black et al., 2017 Agarwal, 2005 Coppolino et al., 2015
Medical Anomaly Detection	Detection of patient condition anomalies or recording errors or instrumentation malfunction.	Neural Networks; Rule-based Systems Clustering; Statistical Model; Bayesian Networks; Nearest Neighbour	Todoroki et al., 2017 Mitchell and Chen, 2015 Chowdhary and Acharjya, 2016 Manogaran et al., 2018 Chandel et al., 2016
Industrial Damage Detection	Detection of machine malfunction	Non-parametric Statistical Model; Parametric Statistical Model; Spectral; Rule Based Systems Neural Networks; Mixture of Models;	Tibaduiza et al., 2016 Lorente et al., 2015 Leite et al., 2015 Zhou et al., 2015 De Fenza et al., 2015 Hollier and Austin, 2002
Image Processing Detection	Detection of motion abnormal appearances on the image.	Regression; Mixture of Models; Support Vector Machines; Bayesian Networks; Clustering; Neural Networks; Nearest Neighbour	Chen et al., 2005 Moustafa et al., 2018 Davy and Godsill, 2002 Chen and Jahanshahi, 2018 Vishnuvarthanan et al., 2016 Wu et al., 2018b Arora and Srivastava, 2015
Textual Anomaly Detection	Detection of interesting events or novel topics in the articles or news.	Statistical Profiling; Neural Networks; Mixture of Models; Clustering; Support Vector Machines;	Li et al., 2016 He et al., 2016b Kusetogullari et al., 2016 Fernández-Gavilanes et al., 2016 Patra et al., 2016
Sensor Networks Detection	Detection of malfunction or anomalous events based on sensor data	Rule-based Systems; Bayesian Networks; Neural Networks; Nearest Neighbour; Spectral; Parametric Statistical Model;	Branch et al., 2013 Zhao et al., 2017 Chine et al., 2016 Subramaniam et al., 2006 Chatzigiannakis et al., 2006 Karami and Wang, 2018

Table 2.4: Summary of detection algorithms with advantages and limitations

Category	Advantages	Limitations	Representatives
Classification based Algorithms	1. Classify the instance into multiple classes 2. Fast in the testing phase 3. Suitable for urban network level detection and prediction	1. Rely on accurate labels	Neural networks (e.g., Todoroki et al., 2017) Bayesian networks (e.g., Kabir et al., 2017) Support vector machines (e.g.,Shams and Rizaner, 2018) Rule-Based algorithms (e.g., Herrera-Semenets et al., 2018)
Clustering based Algorithms	1. Able to be used as unsupervised learning 2. Adapt to complex data type 3. Fast in the testing 4. Perform better with missing anomalies	1. Sensitive to the capabilities of methods in clustering the structure 2. Hard to optimised 3. Every instance is forced to be clustered into some cluster 4. Sensitive to the outliers 5. The computational complexity is high 6. Rely on the distance metric	Nearest Neighbour (e.g., Lin et al., 2015) K-means Clustering (e.g., Kumari et al., 2016) Expectation maximization (e.g., Smith et al., 2002)
Statistical Detection Algorithms	1. Provide a statistically justifiable solution 2. Prediction score is associated with confidence interval 3. Robust to anomalies 4. Able to be used as unsupervised learning	1. Rely on assuming that data is conformed to a distribution 2. Hypothesis tests for complex distribution is nontrivial 3. Difficult to capture the correlations among diverse contributes	Mixture model (e.g., Moustafa et al., 2018) Principle component analysis (e.g., Harrou et al., 2015)
Information Theoretic Detection Algorithms	1. Able to be used as unsupervised learning 2. No assumption about the distribution for the sample data	1. Sensitive to the choice of the information measures 2. Dependent on the substructure size when it is applied on sequences and spatial data 3. Hard to be associated with a score	Entropy (e.g., Coppolino et al., 2015)

2.3.2 Deep Learning Algorithms

The phrase *deep learning* refers to learning the relationships among data through back-propagation algorithms and multi-processing layers (LeCun et al., 2015). Deep learning is an active research area which has demonstrated remarkable success in a number of application areas such as image processing (Farabet et al., 2013), speech recognition (Hinton et al., 2012a) and adaptive control (Mnih et al., 2015). Unlike conventional neural networks with a single hidden layer, deep learning is a set of machine learning algorithms that tend to learn at multiple levels of abstraction (Zhang et al., 2018b). LeCun et al. (2015) comprehensively reviewed the development of deep learning algorithms and their applications. Two commonly-used methods in deep learning are Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM).

2.3.2.1 Convolutional Neural Networks

Convolutional Neural Networks (CNN) is a branch of deep learning techniques initially introduced in 1980 by Fukushima (1980) as a derivative of conventional multilayer neural networks. Deep convolutional nets provide good applications in dealing with video, images, audio and speech in their raw form (LeCun et al., 2015). In addition to the fully connected layers commonly found in conventional multilayers, a CNN includes convolutional layers and pooling layers, which will be presented later.

CNN is designed to use a form of 2D input such as image or speech signals, and this 2D input is able to incorporate the spatial information in the raw input. CNN is also easier to train as it benefits from the convolution layer and sub-sampling with fewer parameters compared with fully-connected networks when the number of hidden units is the same. Figure 2.9 shows an example of CNN called LeNet-5 (LeCun et al., 1998), which is well-known as a model for digital recognition of handwriting. There are two convolutional layers, two max-pooling layers and two fully-connected layers in LeNet-5.

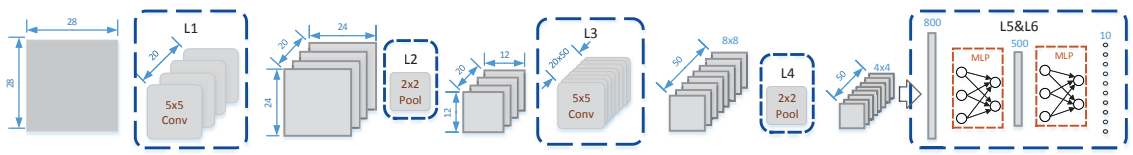


Figure 2.9: A LeNet-5 CNN (Source: LeCun et al. 1998)

The input to a convolutional layer is an $Ri \times Ci \times Ni$ image where Ri and Ci is the height and width of the input and Ni is the number of channels[1]. The convolutional layer has No filters (or kernels) of size $K \times K \times Ni$ where K is the **receptive field** size, which is smaller than that of the dimension of the input image. After applying the filters over the input image, it outputs No feature maps of size $Ro \times Co$, where $Ro = Ri - K + 1$ and $Co = Ci - K + 1$.

Then, each feature map is sub-sampled with max pooling or mean over $P \times P$ contiguous regions where *p* ranges from 2 for small images (e.g. 28×28 handwriting digit images) to 5 for larger inputs. Either before or after the sub-sampling layer, an additive bias and nonlinear activation function (e.g. $sigmoid$ and $tanh$) is applied to each feature map. Figure 2.10 illustrates the operation of a convolutional layer and a pooling layer. Units of the same colour have tied weights and units of different colours represent different filter maps.

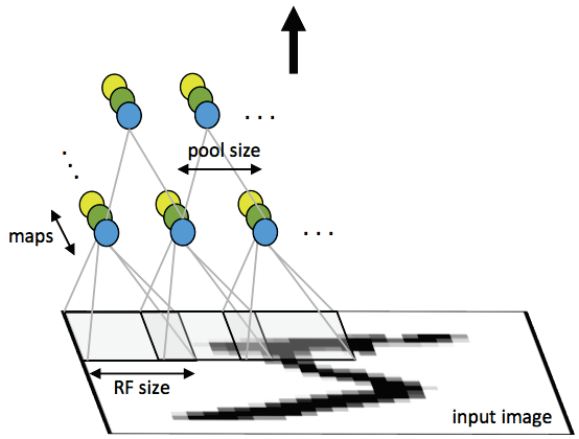


Figure 2.10: First layer of a CNN with pooling (Source: Ng et al. 2016)

The fully-connected layers are the same as in a standard multi-layer neural network. The input is a one-dimensional feature vector V_i with size N_i and the output is a one-dimensional feature vector V_o with size N_o. The weight matrix W has $N_i \times N_o$ values and $V_o = W \cdot V_i + bias$. Given the above notations, a real CNN can be described as shown in Table 2.5.

The locally-connected convolutional layers enable a CNN to capture complex spatial correlations (Krizhevsky et al., 2012), while reducing the number of parameters in the pooling layer, which makes a CNN potentially applicable to large-scale traffic networks (Karpathy et al., 2014).

Recent trends of big data analytics, supported by the existing and emerging sensor tech-

[1] An RGB image has $Ni = 3$ where RGB image refers to a *truecolour* image which stores a data array with the red, green and blue colour for the individual pixel.

Table 2.5: Configuration of LeNet-5

	Ni	No	Ri	Ci	Ro	Co	K	P	Activation Function
L1(Conv)	1	20	28	28	24	24	5		
L2(Pool)	20	20	24	24	12	12		2	$tanh$
L3(Conv)	20	50	12	12	8	8	5		
L4(Pool)	50	50	8	8	4	4		2	$tanh$
L5(FC)	800	500							$tanh$
L6(FC)	500	10							$softmax$

nologies in transportation, mean that data-driven deep learning has a decent opportunity to play a role in traffic estimation, prediction and traffic signal control (Wu et al., 2018c). Indeed, considerable efforts have been focused on the application of deep learning techniques to traffic-related prediction and estimation problems. For instance, Polson and Sokolov (2017) proposed a deep neural architecture combining a linear model and a sequence of $\tanh$ layers to predict traffic flows; and Wu et al. (2018c) exploited a hybrid deep learning algorithm to capture the spatial and temporal correlation for traffic flow forecasting. Ma et al. (2017) recently applied a CNN algorithm to learn the traffic as an image with the aim to predict large-scale and network-wide traffic speed, taking into account spatio-temporal traffic dynamics with relatively high accuracy. The study suggested that deep learning methods, especially CNN, can construct much deeper and sophisticated architectures than a conventional method and therefore can directly capture spatio-temporal traffic features and correlations as a whole in a large-scale network rather than on the traditional isolated links or corridors (Ma et al., 2017). Wu et al. (2018c) presented a Deep Neural Network (DNN) architecture which made full use of the spatio-temporal characteristics of traffic data by using a CNN and RNNs to learn, respectively, spatial and temporal features for traffic flow prediction, and demonstrated that the proposed DNN was capable of improving prediction accuracy. Wang et al. (2016a) used an error-feedback recurrent CNN structure for continuous traffic speed prediction to learn from prediction errors so as to adapt for abrupt traffic anomalies. Most applications of deep learning, however, are related to estimation or prediction problems, while limited studies have investigated its application in traffic anomaly detection.

2.3.2.2 Long Short Term Memory

Long Short Term Memory (LSTM) (Hochreiter and Urgen Schmidhuber, 1997) is another commonly used deep learning method originated from Recurrent Neural Networks (RNNs) (Rumelhart et al., 1986; Werbos, 1990). RNNs are a general class of artificial neural networks, and unlike feedforward neural networks (Rumelhart et al., 1988), RNNs can deal with sequences of time series inputs $(x_1, x_2, \ldots, x_t)$ by using their internal state memory to output a sequence $(y_1, y_2, \ldots, y_t)$ through an iteration over the Eq. 2.19 and Eq. 2.20.

$$h_t = \sigma(w^{h_x} x_t + w^{h_t} h_{t-1} + b) \tag{2.19}$$

$$y_t = f(w^{h_x} x_t + b) \tag{2.20}$$

where, h_t is the hidden layer at time t, x_t is the input at time t, y_t is the output, w and b are the weight and bias respectively, while f is the activation function, in which σ represents the Sigmoid activation function as defined in Eq. 2.21:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.21}$$

Traditional RNNs have two issues when dealing with the short-term prediction: 1) poor performance with a long time span and 2) difficulty in identifying the optimal time window size or lags (Ma et al., 2015a).

The main difference between RNN and LSTM comes from how they maintain information differently in memory over time during the feedback loops where LSTM has a set of gates which could be used for control the information and maintain information in memory for long periods. This architecture enables LSTM to learn longer-term dependencies effectively (Hochreiter and Urgen Schmidhuber, 1997).

Intuitively, more information is needed to decide how to integrate the previous information into the current decision, so the most recent information before the decision time step t may not be enough and information further back is therefore also necessary. LSTM is one of the

more practical ways to address these limitations of RNNs, as LSTM is known to solve long-term dependencies by using memory states. Specifically, this long-term information can be carried effectively with the help of input gate, output gate and forget gate (Hochreiter and Urgen Schmidhuber, 1997). LSTM was initially invented with the concept of gated recurrent units by Hochreiter and Urgen Schmidhuber (1997), and has been proved to exhibit superior capability for time series analysis because of its capability of connecting previous information with a rather long lag and effectiveness in handling long-term dependencies (Wu and Tan, 2016).

2.3.3 Evaluation Methods for Detection Algorithms

The performance measures, Detection Rate (DR), False Positive Rate (FPR), F-measurement, precision and Mean Time to Detection (MTTD) are commonly used in automatic anomaly detection research (Parkany and Xie, 2005) to evaluate the performance of classification problems.

DR is formulated as the rate of the total number of incidents detected by the total number of true incidents, as expressed by Eq. 2.22.

$$DR = \frac{TP}{FN + TP} \tag{2.22}$$

where TP is True Positive and FN is False Negative.

FPR is defined as the rate of the total number of false alarms to the total number of algorithm applications, in other words, the sum of the detection of True Negative (TN) and FP, given by Eq. 2.23.

$$FPR = \frac{FP}{TN + FP} \tag{2.23}$$

Another measure is F-measurement or F1 score as expressed by Eq. 2.24.

$$F - Measurement = \frac{2}{1/DR + 1/Precision} = \frac{2TP}{2TP + FP + FN} \tag{2.24}$$

Precision is defined as the percentage of true positives to all positives detected.

$$Precision = \frac{TP}{FP + TP} \tag{2.25}$$

MTTD is an indicator applied to evaluate the mean time required by a detection algorithm to make a decision for traffic incidents. It is measured as the mean delay in seconds between the apparent occurrence of an incident and its detection, averaged for all incidents detected over a period of time (Srinivasan et al., 2003). It can be defined as Eq. 2.26.

$$MTTD = \frac{1}{n}\sum_{i=1}^{n}(t_{id} - t_{io}) \tag{2.26}$$

where n is the number of correctly detected incident cases; t_{id} is the time where the incident was detected and t_{io} is the time when the incident occurred or was measured. In this study, for traffic incident detection, traffic data and incident data have been aggregated in five-minute intervals. After the aggregation, the occurrence of traffic incidents cannot be directly reflected by the data. Thus, t_{io} has been adapted into the time at the end of each five-minute interval in which the incident occurred. After this adaption, the MTTD used in this study is basically equivalent to the computation time which has been used in the previous study (Tang and Gao, 2005).

In addition to the evaluation indexes mentioned above, the confusion matrix (also known as the error matrix, see Figure 2.11), which is a table presenting the performance of the supervised learning algorithms, can also be used for the performance evaluation.

		Predicted condition	
	Total population	**Predicted non-incident**	**Predicted incident**
True condition	**Non-incident**	True Negative (TN)	False Positive (FP) *Type I error*
	Incident	False Negative (FN) *Type II error*	True Positive (TP)

Figure 2.11: Confusion matrix

Other traditional measures to evaluate and intuitively visualise the performance of binary classification are the Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC) (Fawcett and Provost, 1999). Fawcett (2006) presented the use of ROC in classification problems. Basically, ROC is generated by plotting the true positive rate versus the false positive rate for all thresholds ranging from 0 to 1. The ROC curve of an accurate classifier is presented in the upper left corner. Conversely, the ROC curve of a poor classifier should be close to the diagonal reference line. That diagonal line essentially represents a classifier that is not better than a random guess. A good classifier has an AUC approximating to 1, while a poor classifier has an AUC around 0.5. The ROC curve and AUC are insensitive to whether the predicted probability is properly calibrated to represent the probability of each class precisely.

2.3.4 Summary of Detection Algorithms

This section reviewed the existing anomaly detection algorithms and recent advances in computer science with two representative deep learning methods. CNN and LSTM are capable of capturing the spatial and temporal correlations and abstracting these during the modelling process (Yang et al., 2018; Wu et al., 2018c). This capability is a promising property for modelling spatio-temporal traffic features in this research, and thus a hybrid method combining CNN and LSTM would make it possible to address the gap of early detection of RC that was highlighted earlier.So as to evaluate detection algorithms, traditional anomaly detection metrics, such as detection rate, false positive rate and so forth, together with ROC curve and confusion matrix have been reviewed in Section 2.3.3.

2.4 Summary

This chapter has summarised the previous studies on anomalies and anomaly detection before narrowing down to the traffic anomaly related issues and detection and offering a comprehensive review of the existing and emerging techniques.

Anomaly detection is used in various research domains such as intrusion detection, medical anomaly detection and fraud detection. The definition of anomaly is heavily dependent on the field of application. After a brief presentation of anomaly detection in general, a detailed review was conducted in terms of the available detection algorithms and specific algorithms that have been used for traffic incident detection.

Although a lot of effort has been devoted to traffic-related anomaly detection, some challenges remain: 1) detection of traffic anomalies at the network level rather than at the corridor or link level, and localisation of the traffic anomaly; and 2) early detection of recurrent congestion, i.e., provision of an early alarm in respect to potential congestion in the network. Given the gaps identified, recent advances in computer science, particularly deep learning algorithms, may potentially pave the way to detect traffic anomalies in network and to predict the congestion as early as possible. Specifically, opportunities appear to arise from the availability of sophisticated spatio-temporal models for potential on-line incident detection, especially with the application of Convolutional Neural Networks and Long Short Term Memory. CNN, with its ability to deal with two-dimensional structures, together with LSTM, have the potential to be used in spatio-temporal detection in the traffic network level.

Chapter 3

Spatio-Temporal Anomaly Detection Frameworks

The main objective of this chapter is to develop a novel framework for traffic anomaly detection and early congestion prediction at a network level. Given the gaps and objectives summarised in Chapter 2, to achieve this objective, it is vital to deploy recent advances, especially deep learning techniques, for traffic anomaly detection with spatio-temporal features from a network.

This chapter is organised as follows. Section 3.1 defines the problem of traffic anomaly detection and the scope of this research. Section 3.2 sets out the general structure of the frameworks for both recurrent congestion detection and non-recurrent congestion detection. Section 3.3 summarises the problem definition and methodology framework.

3.1 Definition of the Problem

3.1.1 Definition of the Traffic Anomaly

As discussed in Section 2.1, unlike other domains in data science, such as classification and clustering, anomaly detection is typically customised according to the application subject since it is heavily reliant on subject knowledge. Even though a lot of research has been devoted to different types of traffic congestion, there is no consensus on the definition of traffic anomaly. It is essential to define what a traffic anomaly is at the beginning of the process of developing a conceptual methodological framework.

This research, therefore, defines traffic anomaly as follows:

*A **traffic anomaly** is a traffic observation which deviates significantly from normal free-flow traffic observations.*

In the domain of transport, there are two main types of traffic observations that vary anomalously from the normal free flow. They are recurrent congestion and non-recurrent congestion as shown in Figure 3.1. The definitions of recurrent congestion and non-recurrent congestion are generally linked to their causation. The first one is usually caused by scarce supply of road infrastructure and regularly excessive demand from road users, particularly throughout the morning and evening peak-hours. This kind of congestion is therefore referred to in the literature as recurrent congestion (Anbaroglu et al., 2014a; Emmerink et al., 1995).

The second one is specifically related to individual incidents such as random accidents, bad weather (e.g., fog, heavy rain and snow) and road works wherein the road capacity would be diminished dramatically during a certain period of time. Congestion of this type is referred to as non-recurrent congestion (Anbaroglu et al., 2014a; Emmerink et al., 1995). To clarify, the phrase *traffic anomaly* in this research only refers to these two categories of recurrent and non-recurrent congestion.

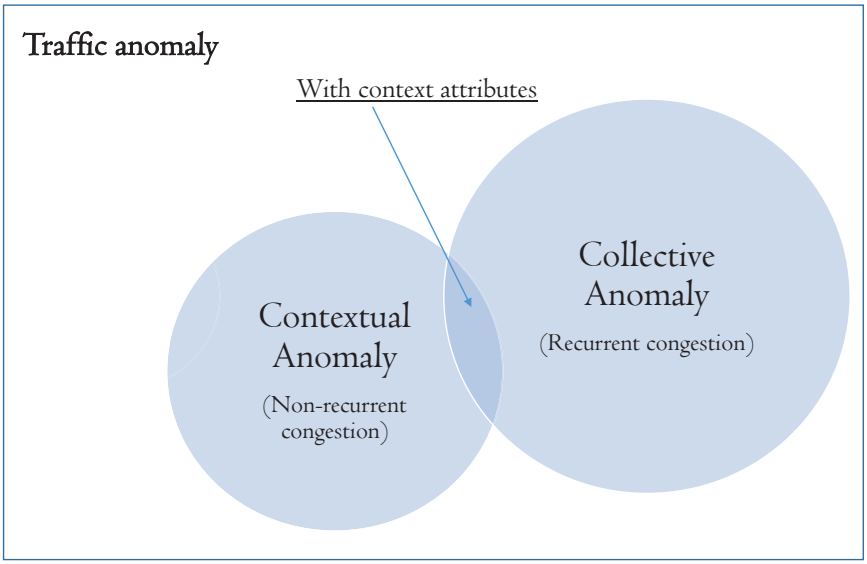


Figure 3.1: Family of traffic anomalies

3.1.2 Characteristics of the Traffic Anomaly

The characteristics of two types of traffic anomalies or congestion are varied. It is obvious that the first type of anomaly belongs to the class of collective anomalies, while the second is contextual anomalies due to the availability of contextual information. Since these traffic congestion anomalies commonly last for a period of time and the nature of the data input is time-series data, it should be noted that it is rare to find traffic anomalies belonging to the category of point anomaly. This research assumes that traffic anomalous variables associated with specific events or behaviours will be associated with contextual anomalies, and that this type of anomaly is related to the detection of non-recurrent congestion.

On the other hand, traffic anomalous variables caused by normally excessive traffic demand but not subject to any specific events are categorised as collective anomalies, and this type of anomaly corresponds to the recurrent congestion detection problem. This research covers both types of traffic anomaly.

3.1.3 Comparison of Two Traffic Anomaly Detection Problems

Even though these two types of traffic anomaly, i.e., recurrent congestion and non-recurrent congestion, can sometimes overlap, considering the nature of the requirement for labelled data,

the two research problems can be treated separately. Specifically, recurrent congestion is usually predicted without congestion labels such as severe congestion, moderate congestion and normal condition, whereas the non-recurrent congestion problem is usually in association with the real labelled contextual information as indicated in Section 2.1.2. Thus, for the purposes of machine learning, the former should be treated as an unsupervised learning problem while the latter will be trained as a supervised learning problem. Secondly, based on the comprehensive literature review on these congestion fields, the research challenges identified in Section 2.2 are different in terms of the availability of contextual information and the possibility of early detection.

Since recurrent congestion is usually caused by excessive demand during rush hours, similar to the traffic state estimation or prediction problem introduced in Section 2.2.2, it can be predicted based on traffic variables by using data-driven methods. The key objective, therefore, is to identify recurrent congestion as early as possible so that an early alarm can be given as an input for ITS in practice. The other objective is to develop a single method that can be used as a predictive model for a whole network by considering the spatio-temporal correlations rather than that of traditional corridor-based or link-based models. On the other hand, non-recurrent congestion is usually triggered by some random factors, such as car accidents, which can hardly be estimated in advance.

Detecting non-recurrent congestion on an urban road network is more challenging than that of recurrent congestion due to (1) the heterogeneous nature of an urban road network and (2) the random nature of non-recurrent events (Anbaroglu et al., 2014a). Traditionally, however, in the context of non-recurrent congestion, since detection methods are link-oriented and require the learning of a network with hundreds and thousands of links, a number of models are used. It is vital for us to identify the non-recurrent congestion at a network level using the clues from historical non-recurrent congestion patterns across the network within a single model rather than training numerous models for different links.

In summary, the methods developed in this chapter should solve two research questions.

- How to predict recurrent congestion as early as possible considering spatio-temporal information from a network-level
- How to detect traffic recurrent congestion and non-recurrent congestion respectively for

a network within a single model

As mentioned in Section 1.2.1, another disruptive factor affecting traffic management is sensor fault anomalies. Some types of sensor fault anomaly may belong to point anomaly. To ensure the reliability of traffic anomaly detection, the robustness of the proposed anomaly detection against factors such as sensor faults, data quality and others will be discussed in Chapter 7.

In order to solve both research questions, it is vital to develop a model which can, first, help to capture the spatio-temporal information from traffic networks, second be capable of processing a large network within only one model and, third, learn effectively from the historical information to build up the predictive power for early detection. The recent advance in deep learning techniques could be used to address both research questions. CNN can learn traffic spatial correlations of an entire urban traffic network by only one model for recurrent congestion and non-recurrent congestion. LSTM is capable of temporal correlation learning and it is effective in long-term dependencies prediction with historical traffic data. More explanation on this based on practical case studies will be included in following Chapters.

3.2 Methodological Framework

It is essential to construct a conceptual framework to formulate the process of recurrent congestion detection and non-recurrent congestion detection. This framework, in addition to robust analysis, needs to take practical issues such as data quality and sensor faults into consideration.

As implied by the flow of general anomaly detection in Section 2.1.3, the conceptual methodology for this research can be presented in Figure 3.2.

Figure 3.2 shows the three stage anomaly detection framework. The first stage is the translation layer to extract the spatio-temporal features. In the second stage, a deep learning technique is proposed that has the capability to process multi-dimensional large volume data to detect traffic anomalies, whereas the third stage serves to localise an anomaly within the network. The following sections from 3.2.1 to 3.2.3 will introduce these stages briefly.

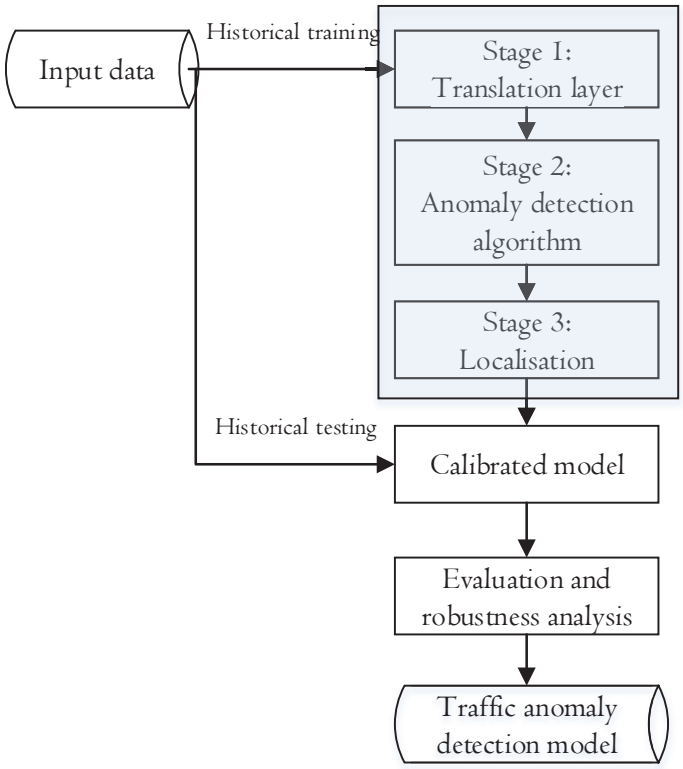


Figure 3.2: Conceptual methodology framework

- Translation layers to transfer data into the input format of the proposed network-level model
- Deep learning methods based anomaly detection for the identification of non-recurrent congestion and the early detection of recurrent congestion
- Localisation to find the congestion point

In this research, the possibility of using multiple traffic variables, i.e., traffic flow and occupancy, to detect the traffic anomalies by making use of the deep learning approach is explored. Given the different objectives of detecting both recurrent and non-recurrent congestion, the detailed framework for these two methods will be slightly different, as shown in Figure 3.3 and Figure 3.4 respectively. Specifically, in terms of non-recurrent congestion, a supervised CNN will be proposed to map the traffic non-recurrent congestion labels with traffic variables; whereas, for the recurrent congestion problem, the hybrid methods (i.e., the combination of CNN and LSTM) will be proposed to detect recurrent congestion anomalies.

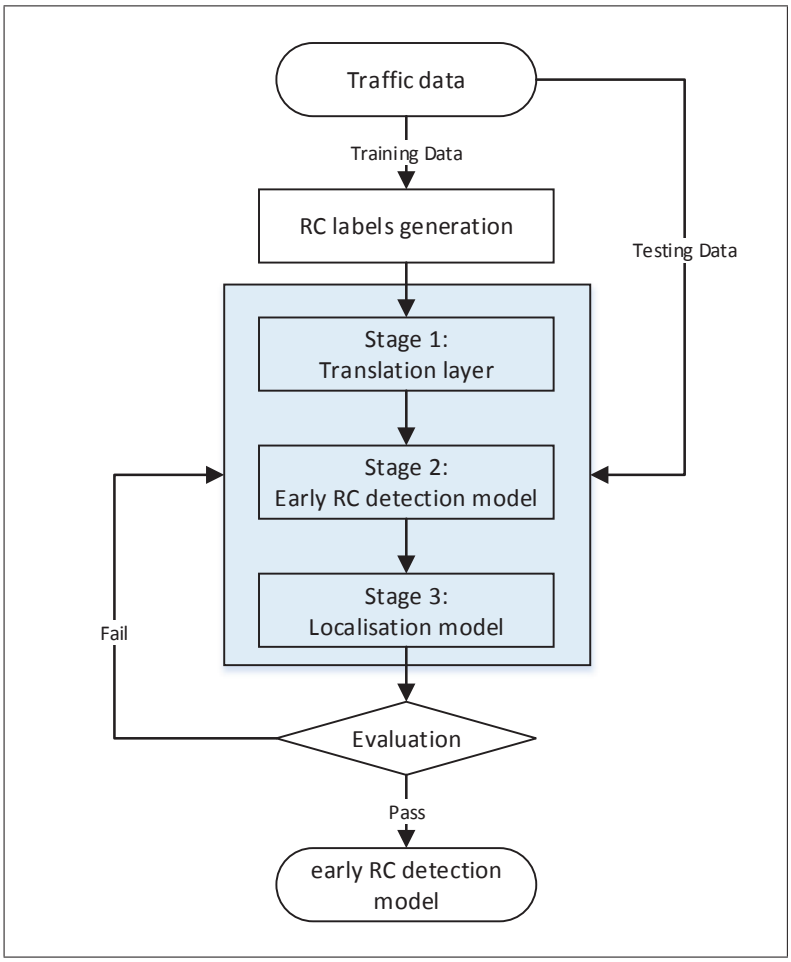


Figure 3.3: Methodological framework for early recurrent congestion detection

The difference between these two frameworks is summarised as follows.

- Input data: recurrent congestion detection is based only on traffic data while the process for non-recurrent congestion also includes the contextual causation information;
- Data preparation before stage one: recurrent congestion needs the stage of label generation or traffic state identification, whereas non-recurrent congestion has to match to traffic data and incident data;
- Detection model: both models should be capable of network-based detection incorporating both spatial and temporal information, but the detection of recurrent congestion is also aimed at early prediction and thus needs to adapt effectively at viable time scale.
- Output: the output of recurrent congestion detection is binary traffic congestion states while for non-recurrent congestion the output is the existence of traffic incidents.

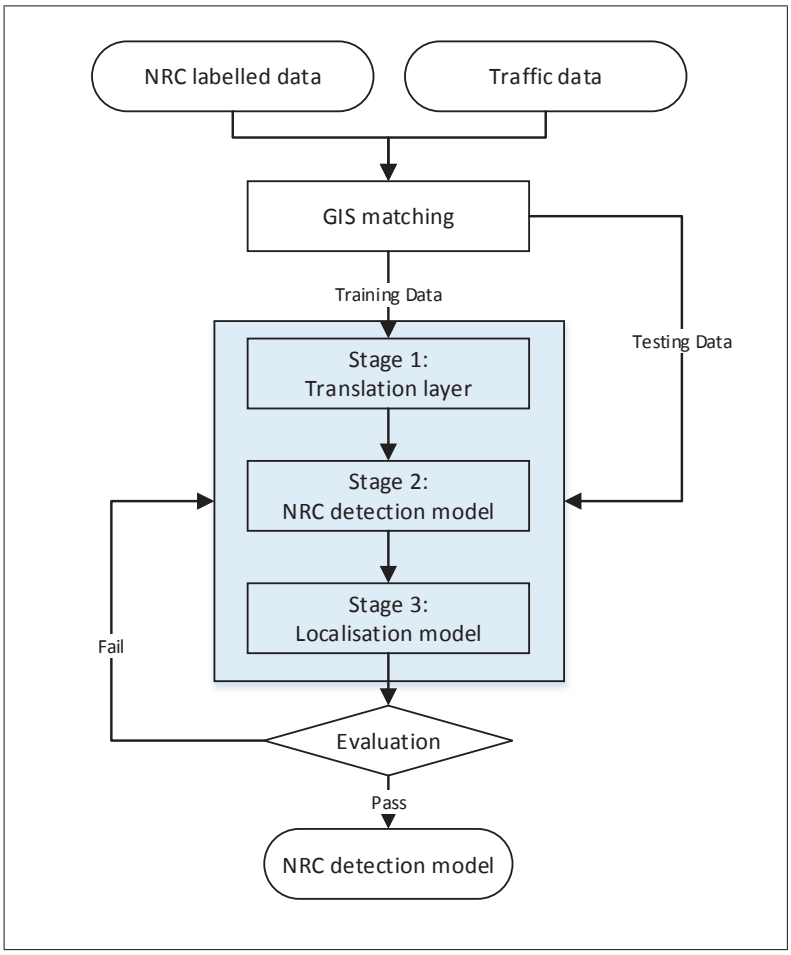


Figure 3.4: Methodological framework for non-recurrent congestion detection

3.2.1 Translation Layers

Considering the complexity and dynamic nature of urban traffic, a translation layer that transfers traffic data to an appropriate input format that enables the proposed detection model to learn the spatio-temporal features efficiently is important. The transformation between the natural formats of traffic data and multi-dimensional data is not straightforward (Ma et al., 2017).

The general logic behind this translation leads to three types of methods. The first type is based on the connection between traffic nodes or regions. In other words, network connectivity can be constructed by using node-to-node connection information. Thus, under the context of complex urban networks, a possible way to address this transformation is by using a connectivity matrix, which can establish a direct connection among all links for each time step.

The second type is to group all links in geographically similar locations by a grid and then assign a value, such as the maximum, minimum or average, to the centroid of the regional centre.

This type of method substantially reduces the size of the network and can ideally transform a complex network into a specified network with the required dimensions.

The final type of translation is using spatial and temporal data as, respectively, the x-axis and y-axis so as to build a two dimensional data matrix directly. Wang et al. (2016a) and Ma et al. (2017) converted network traffic into a time-space image where the x-axis and y-axis present the time and space of a matrix, respectively. Such a transformation that directly represents traffic variables with time and space may be straightforward and efficient for simple structural corridors, since it is easy to rank the sequence of space, but it is not directly applicable in the case of urban traffic networks where the spatial connections among corridors are complex, which could significantly increase the difficulty of ranking the sequence of space in a matrix.

Corresponding to the three types of methods identified above, in this research, three specific approaches have been proposed to translate the traffic network into inputs for the network-level anomaly detection model.

- Connectivity matrix
- Geographical grid
- Spatio-temporal two dimensional translation

Figures 3.5-3.7 show the concept of the three proposed translation layer methods. The values inside the figures do not reflect any traffic variables but do illustrate how each of the translation layers works.

The connectivity matrix constructs the direct connection matrix among all links without taking the ranking of space into account. This approach consists of two steps: (1) given the original traffic network, node location and flow direction, each cell representing a direct connection gets an index of 1 while each cell that does not represent a direct connection receives a index of 0; (2) assign the value of traffic flow for the cells with an index of 1 to the connectivity matrix. Each time step corresponds to a converted connectivity matrix with traffic variables, such as traffic flows.

The geographical grid translation layer groups neighboured traffic variables together to formulate a specified traffic network matrix as in Figure 3.6.

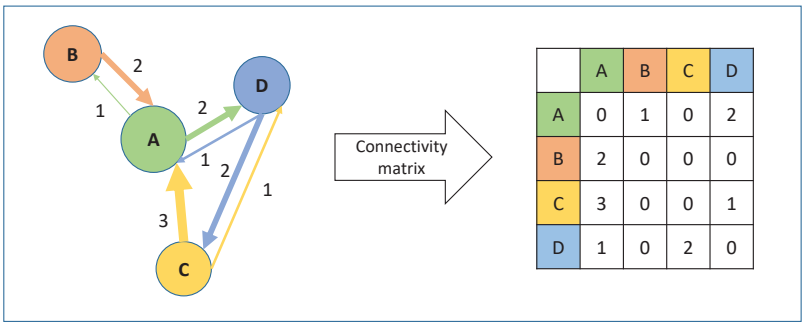


Figure 3.5: Connectivity matrix translation layers

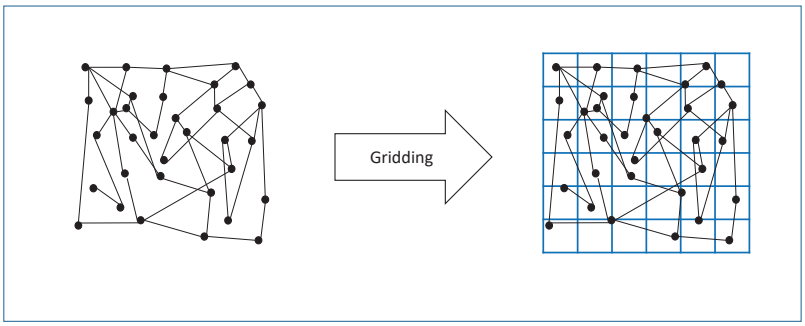


Figure 3.6: Geographical grid translation layers

For the spatio-temporal translation, time-series traffic flow data will be converted into a 3D time-space feature where the x-axis, y-axis and z-axis represent time, space and size of the time windows, respectively. Each cell inside the matrix represents the traffic states or traffic variables.

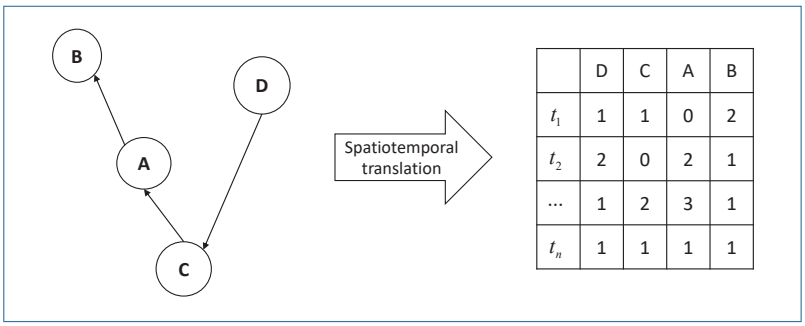


Figure 3.7: Spatio-temporal translation layers

3.2.2 Anomaly Detection Framework and Methods

As mentioned in Section 2.1.3, the anomaly detection frameworks generally contain three parts: input, detection algorithm and output. Section 3.2.1 presented how to translate the spatio-temporal network into the field of transport research and thereby three ways to generate the input for the detection model: 1) connectivity matrix; geographical grid; and 3) spatio-temporal translation. Based on the translation proposed in the last section, this section will focus on traffic anomaly detection methods.

Since recurrent congestion labels are essential for the framework governing the early detection of recurrent congestion, as shown in Figure 3.3, the initial stage for recurrent congestion detection will be the generation of recurrent congestion labels. Section 3.2.2.1 formulates the proposed label generation process by using an Expectation Maximisation algorithm. Based on the input translated in Section 3.2.1 and labels generated for recurrent congestion detection, CNN is proposed for capturing the spatial features while LSTM has been formulated to extract the feedback in time scales. A hybrid CNN-LSTM method is therefore proposed for the detection of recurrent congestion. On the other hand, since it is hard to early predict random events, LSTM which is capable of temporal information extraction has not been selected for NRC detection.

3.2.2.1 Traffic States Identification

In the first step of early RC detection, the Expectation Maximisation (EM) algorithm, is used to classify ("generate labels" in the parlance of the machine learning community) the traffic states according to different levels of severity of congestion. The primary objective of this step, as mentioned above, is to generate labels, i.e., indexes for congestion and non-congestion, as labelled outputs for the CNN-LSTM early RC detection model.

The EM algorithm has been proved to be an effective and transferable probabilistic traffic state classifier which can capture the latent features of the underlying distribution (Han et al., 2010). The application of the EM algorithm to classify different traffic states was originally introduced by Han et al. (2010). Generally, it can be viewed as a form of unsupervised learning method which, in this context, is used to generate labels that can, in turn, be used to drive a

more sophisticated CNN-LSTM based supervised deep learning method for detecting the early onset of congestion.

The traffic states as mentioned in Section 2.2.2 are related to the ratio α_i of traffic flow and occupancy.

$$\alpha_i = \frac{o_i}{q_i} \tag{3.1}$$

The traffic states are assumed to be clustered into two separable regimes by either congested or uncongested. Another assumption is that the probability distributions of the traffic states follows Gaussian distribution, defined as:

$$\begin{aligned} p(\boldsymbol{\alpha})|_{z=0} &\sim \mathcal{N}(\mu_0,\, \sigma_0^2) \\ p(\boldsymbol{\alpha})|_{z=1} &\sim \mathcal{N}(\mu_1,\, \sigma_1^2) \end{aligned} \tag{3.2}$$

where $\mathcal{N}(\mu,\, \sigma^2)$ is a Gaussian distribution with mean μ and variance σ^2 and $p(\alpha)|_z$ is the probability density function of a given traffic state. So,

$$p(\alpha_i) \;=\; \gamma_0 \mathcal{N}(\mu_0,\, \sigma_0^2) \;+\; \gamma_1 \mathcal{N}(\mu_1,\, \sigma_1^2) \tag{3.3}$$

where the parameters γ_0 and γ_1 are the mixture factors with the constraint of $\gamma_0 + \gamma_1 = 1$.

The Eq. 3.3 is a typical Gaussian Mixture Model (GMM) with unknown parameters $\boldsymbol{\Theta} = (\gamma_0,\, \gamma_1,\, \mu_0,\, \mu_1,\, \sigma_0^2,\, \sigma_1^2)$. So the probabilistic model is defined as:

$$p(\boldsymbol{\alpha}|\boldsymbol{\Theta}) \;=\; \sum_k \gamma_k p_k(\boldsymbol{\alpha}|\theta_k) \tag{3.4}$$

Each p_k is a Gaussian distribution function parameterised by θ_k, where $\theta_k = (\mu_k,\, \sigma_k^2)$ and $k = \{0,\, 1\}$.

With the Gaussian distribution of these two traffic states, the question is then modelled to solve the probability of $P(Z_i = 1|\boldsymbol{\alpha} = \alpha_i)$ and from Bayesian theory:

$$P(Z_i = 1|\boldsymbol{\alpha} = \alpha_i) \;=\; \frac{P(\alpha_i|Z_i = 1)P(Z_i = 1)}{P(\alpha_i)} \tag{3.5}$$

Eq. 3.5 is then formulated to:

$$P(Z = 1|\boldsymbol{\alpha} = \alpha_i) \;=\; \frac{\gamma_1 p_1(\boldsymbol{\alpha} = \alpha_i|\theta_1)}{p(\boldsymbol{\alpha} = \alpha_i|\boldsymbol{\Theta})} \tag{3.6}$$

In Eq. 3.6, only the parameters $\boldsymbol{\Theta}$ of the GMM are unknown to the model, and these parameters can be calculated by using maximum likelihood estimation with

$$\mathcal{L}(\boldsymbol{\Theta}|\boldsymbol{\alpha}) \;=\; p(\boldsymbol{\alpha}|\boldsymbol{\Theta}) \tag{3.7}$$

According to maximum likelihood estimation theory, the parameters are ones that maximise $\mathcal{L}$, i.e.,

$$\boldsymbol{\Theta}^* = \underset{x}{\operatorname{argmax}}\, \mathcal{L}(\boldsymbol{\Theta}|\boldsymbol{\alpha}) \tag{3.8}$$

So the problem is reduced to finding the parameters to cluster two different traffic states statistically. The EM algorithm is used to find maximum likelihood estimates of parameters in statistical models. The EM algorithm conducts an iteration of the expectation step, which creates an expectation of the log-likelihood for the parameters, and the maximisation step, which maximises the log-likelihood at the expectation step. The formulated expectation and maximisation steps will not be covered here due to the extensive derivatives, although readers may refer to Han et al. (2010) for more details.

3.2.2.2 Convolutional Neural Networks

This section will present methodological details of the proposed CNN algorithm which is vital for NRC detection and an important component for early RC detection.

Artificial neurons are basically processing units that usually find correlation between input variables $X = (x_1, x_2, \ldots, x_n)$ and target output $Y = (y_1, y_2, \ldots, y_n)$ through an activation function $f(\cdot)$ with weights $W = (w_1, w_2, \ldots, w_n)$ and bias b. Mathematically, the full process

of a neuron operation can be defined as shown in Eq. 3.9:

$$f(x) = \sum_{i=1}^{N} x_i \times w_i + b \tag{3.9}$$

where N is the size of the input vector, weights W and input variables X have the same dimension $\mathbb{R}^n$, while the activation function is the transformation $f(\cdot)\colon \mathbb{R} \rightarrow \mathbb{R}$. Multi-layer networks are formed by grouping these multiple processing units. CNN is a typical multi-layer network and typically consists of three types of processing units, i.e., convolution, pooling and fully connected layers.

In the convolution unit, a fixed-size window $k \times k$ runs over the input matrix x of size $M \times M$ to define a region of interest, and after that variable values are created inside the window as the input, with a size of $(M-k+1)\times(M-k+1)$ for neurons with an operation like the formula given in Eq. 3.9, where the feature is then extracted via an activation function $f(\cdot)$ in this process. The Rectified Linear Unit (ReLU), formulated as Eq. 3.10, is used here as an activation function due to its ability in detecting high-frequency features in a local neighbourhood (Nogueira et al., 2017).

$$f(x) = \max(0, x) \tag{3.10}$$

Typically, after the convolutional layer, there are pooling layers to reduce the variance of features by running a fixed-size window over the features to reduce their number and optimise the gain. The commonly used operation in this phase is to select the maximum value over the feature region generated by convolutional layers, as this maximum process ensures that the significant features can be obtained for detection, even with varying levels of translations or rotations (Nogueira et al., 2017).

After several rounds of convolutional and pooling operations, fully-connected layers make use of most parameters to learn all neurons in the previous layer and output to the current layer where the spatial notion of the matrix is reduced to that of a one-dimensional vector. To prevent overfitting caused by parameters-dominated fully connected layers, the dropout (Srivastava et al., 2014) approach is employed, which basically drops a couple of neuron outputs randomly.

The advantages of this technique lie in the ability to decrease the number of neurons, improving the speed of training and making the model practically effective.

After the convolutional layer, pooling layer and fully connection layer, a classifier layer, i.e., a particular type of fully-connected layer, is employed to calculate the class probability for each instance. The most common classifier is the softmax (Bengio et al., 2009) or so-called normalised exponential, which is a generalisation of the multinominal logistic function that generates a vector of real values ranging from 0 to 1 which represent the probability distribution. Eq. 3.11 shows the mathematical formula of softmax which gets the probability for each jth class given a sample vector X.

$$h_{W,b}(X) \;=\; P(y = j \mid X; W, b) = \frac{\exp X^T w_j}{\sum_{k=1}^{K} \exp X^T w_k} \tag{3.11}$$

The objective is to minimise the loss function based on updated weights and bias, as expressed in Eq. 3.12.

$$\underset{W,b}{\operatorname{argmin}}\, \mathcal{J}(W, b) \;=\; -\frac{1}{N} \sum_{i=1}^{N} (y^{(i)} \times \log h_{W,b}(x^{(i)}) + (1 - y^{(i)}) \times \log(1 - h_{W,b}(x^{(i)}))) \tag{3.12}$$

where y represents a possible class, x is the input data, i is a specific input, and N represents the total number of datasets. With the cost function defined, CNN can be trained in order to minimise the loss by using some optimisation algorithm. Stochastic Gradient Descent (SGD) is used to update the weights and bias gradually in search of the optimal solution, such that:

$$W_{ij}^{(l)} \;=\; W_{ij}^{(l)} - \alpha \frac{\partial \mathcal{J}(W, b)}{\partial W_{ij}^{(l)}} \tag{3.13}$$

$$b_{i}^{(l)} \;=\; b_{i}^{(l)} - \alpha \frac{\partial \mathcal{J}(W, b)}{\partial b_{i}^{(l)}} \tag{3.14}$$

where α denotes the learning rates which represent the learning intensity in each step. The training step of a CNN generally consists of a feed-forward stage to pass all information from

the first layer until the classifier, and a back-propagation stage to calculate errors δ and partial derivatives for weights and bias. These errors are then propagated through all the layers from the classifier back to the first layer. The label of the supervised model can be defined as:

$$f(x_t) = \begin{cases} 1, & \text{if a traffic anomaly occurs at time } t. \\ 0, & \text{otherwise.} \end{cases} \tag{3.15}$$

One example of a CNN model with a matrix input and an output of probability of traffic anomaly is shown in Figure 3.8.

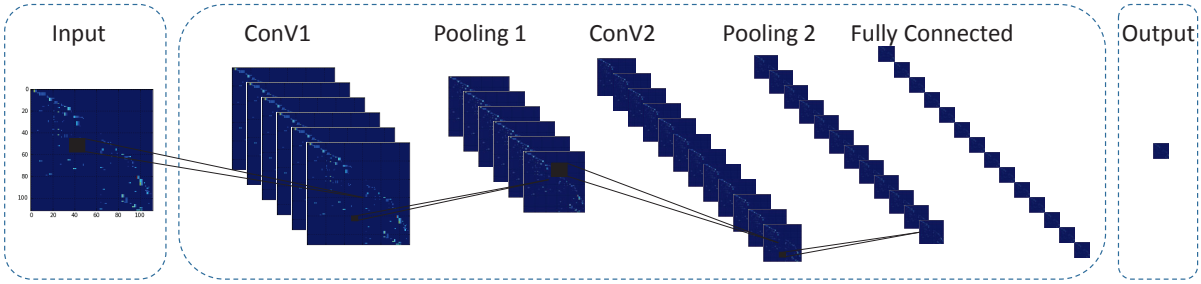


Figure 3.8: One example of a CNN model with a matrix input and an output of the probability of traffic incidents

3.2.2.3 Combined Methods

As mentioned earlier, CNN-LSTM is proposed for the early RC detection stage, where inputs have been transferred into the matrix by using spatio-temporal tensor extraction in Section 3.2.1, and outputs are labels generated from the first traffic state identification stage in Section 3.2.2.1.

The early detection model consists of a CNN and a multi-layer LSTM, as shown in Figure 3.9. The CNN is initially used for general classification but cannot predict the upcoming part of the matrix. LSTM, however, enables CNN to detect traffic congestion with partial data and predict future states because of its capability of sequence processing. A LSTM unit is with input (i.e. i), output (i.e. o), and forget (i.e. f) gates where each of these gates is a feed-forward neural network. i_t, o_t and f_t represent the activations of the input, output and forget gates, at time step t. The 3 exit arrows from the memory cell c to the 3 gates i, o and f represent the connections as shown in Figure 3.9. The small circles containing a $\times$ symbol represent

an element-wise multiplication between its inputs. The big circles containing a $\int$ represent the application of a differentiable function, such as the sigmoid function, to a weighted sum. The mathematical equations detailed this whole LSTM process $(i_t,\ C_t,\ f_t,\ o_t)$ with activation function are formulated in the Eq.3.22 - Eq.3.27.

In this stage, CNN reads the input as an image and obtains a fixed size vector representation of the initial input. After that, a multi-layer LSTM takes the representation, original input and the last output from previous timestamps to produce the desired output, i.e., traffic states in a certain timestamp. In order to localise traffic anomalies, the localisation will be used to output the standard vector output into a multidimensional output according to the different proposed localisation methods in Section 3.2.3. The main advantage of this combined model is that it delivers the early detection even with partial data because it can access the raw inputs and get any information missed by previous steps.

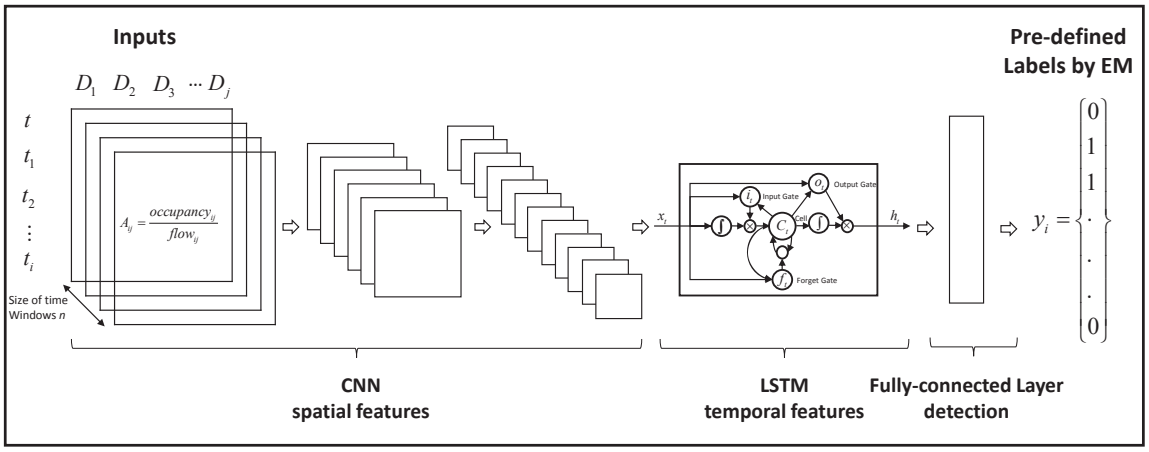


Figure 3.9: Overview of CNN-LSTM

The simplified version of the early RC detection system is defined as Eq. 3.16, where CNN takes the initial converted matrix, and Eq. 3.17, where LSTM outputs the detection results. The intuition behind this model, therefore, is that CNN reads the input matrix converted from historical traffic data and outputs a vector $\boldsymbol{v}$, and both $\boldsymbol{v}$ and the initial matrix will be fed into LSTM to detect RC early so as to avoid any missing information from the CNN step.

$$\boldsymbol{v} = f_{CNN}(matrix_{historical}) \tag{3.16}$$

$$Output \ = \ f_{LSTM}(\boldsymbol{v} + matrix_{historical}) \tag{3.17}$$

Among the CNN architecture, including convolutional layers, pooling layers and fully connected layer, the convolutional layer is possibly the most important (Krizhevsky et al., 2012). The convolutional layer connects the input matrix representation defined as $x \in \mathbb{R}^{p \times q \times r}$, where, $p,\ q$ and r are three dimensions of the input matrix, as defined in the tensor extraction step with a set of filters $k \in \mathbb{R}^{n \times p \times m \times m}$ with stride of size s_k, where n represents the number of convolutional filters and m is the size of kernel. The weights of the filter k are shared spatially and are different for every channel of the feature map. Then, the convolutional layer is formulated as Eq. 3.18:

$$y_i \ = \ f_a(\sum_i k_{ij} * x_i) \tag{3.18}$$

where, x_i is the i^{th} channel of input, k_{ij} is the convolution kernel, f_a is the activation function and y_j is the hidden layer extracted by the convolutional layer. For the convolutional layer of the CNN, the activation function ReLU (Nair and Hinton, 2010) is used. ReLU refers to Rectified Linear Unit, which is a non-saturating activation function that can capture the non-linearity of the neuron's output to converge and guarantee the non-negativity of the output (Nair and Hinton, 2010). Deep convolutional neural networks with ReLU train much faster than the other equivalent activation functions, such as $\tanh$ and Sigmoid function (Krizhevsky et al., 2012).

$$ReLU(x) \ = \ \max(0, x) \tag{3.19}$$

Max pooling layers formed by Eq. 3.20 serve to shrink the width and height of the feature map with filter size z and stride size s_z,

$$y_{ijk} \ = \ \max_{p,q} x_{i,j+p,k+q} \tag{3.20}$$

where, $x_{i,j,k}$ is the value of the i^{th} feature map at position j, k ; p is a vertical index in the local neighbourhood, q is a horizontal index in the local neighbourhood, $y_{i,j,k}$ is pooled and sub

sampled layer.

Fully connected layers are achieved through a dot product between the final layer y and weight matrix $\boldsymbol{W}^{\boldsymbol{v}}$ adding bias vector $\boldsymbol{b}$. Then the output is passed through an activation function f_a.

$$\boldsymbol{v} \;=\; f_a(\boldsymbol{W}^{\boldsymbol{v}} y + \boldsymbol{b}) \tag{3.21}$$

CNN is capable of capturing patterns in local regions and these captured patterns, vector $\boldsymbol{v}$, are then fed into a multilayer LSTM model which consists of input gate, output gate, forget gate and memory cell. While the input gate decides the passing of new memory, the forget gate decides whether to keep the previous memory or not and, finally, the output gate produces the outflow of the current hidden state. The final set of layers is composed of dropout (Srivastava et al., 2014) and fully-connected hidden layers which make a specific classification based on all features detected by the previous layers.

In addition to RNNs, the key idea of LSTM is the memory cell in hidden layers where errors can flow back forever and thus make the error flow tend to decay exponentially in the whole process from an input gate i_t, a self-recurrent connection neuron C_t, a forget gate f_t to an output gate o_t (Gers, 2001). The mathematical equations detailed this whole LSTM process $(i_t,\ C_t,\ f_t,\ o_t)$, as shown in Figure 3.9, with activation function are formulated in the equations below.

$$Input\ gate,\ i_t \;=\; \sigma(x_t W^{x_i} + h_{t_1} W^{h_i} + v W^{v_i} + y_{t-1} W^{y_i}) \tag{3.22}$$

$$Forget\ gate,\ f_t \;=\; \sigma(x_t W^{x_f} + h_{t_1} W^{h_f} + v W^{v_f} + y_{t-1} W^{y_f}) \tag{3.23}$$

$$Output\ gate,\ o_t \;=\; \sigma(x_t W^{x_o} + h_{t_1} W^{h_o} + v W^{v_o} + y_{t-1} W^{y_o}) \tag{3.24}$$

$$New\ state,\ \widetilde{C}_t \;=\; \tanh(x_t W^{x_c} + h_{t_1} W^{h_c} + v W^{v_c} + y_{t-1} W^{y_c}) \tag{3.25}$$

$$Final\ state,\ C_t\ =\ C_{t-1} \circ f_t + \widetilde{C}_t \circ i_t \tag{3.26}$$

$$Final\ hidden\ state,\ h_t\ =\ \tanh(C_t) \circ i_t \tag{3.27}$$

Here, $x_t \in \mathbb{R}^d$ is the current state of the original input and $\mathbb{R}^d$ refers to the number of input features, and $h_{t-1} \in \mathbb{R}^h$ is the previous hidden state and $\mathbb{R}^h$ is the number of hidden units. v is the output vector, y_{t-1} is the last output of the LSTM and σ denotes the standard logistics Sigmoid function. The output for each state is derived as:

$$Output,\ y_t\ =\ \sigma(h_t W^{y_i} + b) \tag{3.28}$$

The loss function L is defined as the root mean squared loss as formed by:

$$L\ =\ \sqrt{\frac{1}{n}\sum_i (\hat{\boldsymbol{y}} - \boldsymbol{y})^2} \tag{3.29}$$

where $\hat{\boldsymbol{y}}$ is the predicted output and $\boldsymbol{y}$ is the original output.

For the final fully connected layer of the LSTM, the $\tanh$ activation function is used as in Eq. 3.30,

$$\tanh(x)\ =\ \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.30}$$

while for the final output layer the Sigmoid activation function is used to make the output the probability of different traffic states (i.e., congestion or non-congestion).

For regularisation purposes, the L2 loss is used, as given by Eq. 3.31:

$$L2\ =\ \lambda \sum_{i=1}^{k} \omega_i^2 \tag{3.31}$$

where ω represents the weights and λ is a hyperparameter to determine the level of L2 loss to

be added into the final loss model. The weights and biases are updated and optimised with an Adam optimiser (Kingma and Ba, 2014).

3.2.3 Localisation

After the translation layer to transfer the traffic data into a multidimensional matrix format and a network-based anomaly detection model, another important step is to localise the traffic anomaly. In the previous section, to simplify the research questions, a network-level traffic anomaly detection incorporating the spatio-temporal information in the whole network was proposed. Thus the corresponding labels for traffic anomaly were set without considering the location of the traffic anomaly detected. In practice, after a network-level anomaly detection, an effective localisation is more important for rapid and proactive emergency reaction or traffic management. Since traditional RC or NRC methods are mainly link-based or segmentation-based, limited research has been devoted to traffic anomaly localisation in a network level. Considering the practical issue, another contribution in this research is to localise the traffic anomaly after it has been flagged at a network level. Thus, one method is via transposed matrix and the mathematical formulation for this is as follows.

Suppose that the matrix input after the translation layer can be simplified by a two-dimensional format as below.

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{d1} & x_{d2} & x_{d3} & \dots & x_{dn} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \tag{3.32}$$

where:

x = the traffic variables in the matrix cell, such as the ratio of flow and occupancy at segment n and at time d for spatio-temporal translation, or the ratio of flow and occupancy connected node n and d

d, n = the dimension of x-axis and y-axis respectively

y = the traffic anomaly index for position d

To localise a traffic anomaly $A(d, n)$, one can transpose the matrix to predict the labels for

the other dimension, i.e., n as Eq. 3.33.

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{d1} & x_{d2} & x_{d3} & \dots & x_{dn} \end{bmatrix}^{\top} = \begin{bmatrix} {y'}_1 \\ {y'}_2 \\ \vdots \\ {y'}_n \end{bmatrix} \tag{3.33}$$

Given the probability of occurrence of traffic anomaly from the horizontal and vertical, the traffic anomaly could be located with the combination of horizontal (row) and vertical (column) probability with parameters (b, w_1, w_2) as Eq 3.34.

$$w1 \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} + w2 \begin{bmatrix} {y'}_1 \\ {y'}_2 \\ \vdots \\ {y'}_n \end{bmatrix} + b = \begin{bmatrix} {y''}_1 \\ {y''}_2 \\ \vdots \\ {y''}_n \end{bmatrix} \tag{3.34}$$

Since limited research is available in this area, in this research, six different methods reviewed in Section 2.3.1 are proposed: weighted average probability, conditional probability, index, logistic regression, random forest and gradient boosting classifier. More descriptions their applications are included in Section 6.2.

3.2.4 Summary

Section 3.2 has presented the methodological framework for traffic anomaly detection and early prediction in order to meet the objective set out in Section 1.3. Three main components of the methodological framework, i.e., the translation layer, the anomaly detection methods and the process of localisation, have been formulated in detail. The translation layer is responsible for converting the traditional traffic data into a matrix feature space where the spatio-temporal information could be easily captured by a deep learning-based detection method. Apart from translation layer and detection methods, another important component is localisation which is a powerful function for a network-based model.

3.3 Summary

This chapter first defined the concept of traffic anomaly in Section 3.1 and then presented the scope and reemphasised the objectives of this study according to the current knowledge in the traffic domain.

To develop a network-based framework for traffic anomaly detection, this chapter presented a translation layer to transform the previous time-series data into a multi-dimensional matrix fashion, where three methods, i.e., connectivity matrix, geographical grid and spatio-temporal translation, have been introduced to incorporate the spatial and temporal features in the network.

The translation layers are then served as inputs for the detection model by making the use of the recent advances in deep learning with the aim of detecting RC and NRC earlier and more precisely. CNN and LSTM have been proposed to capture the spatial and temporal information, respectively, before the label generation phase for the early RC detection problem.

Since localising the traffic anomaly is of importance in practice, a fundamental method of localisation has been proposed in Section 3.2.3. The main idea behind the localisation is to combine different types of weights into the vector format output.

In summary, this chapter has highlighted and dealt with the second objective as set out in Section 1.3, which is to develop a novel spatio-temporal framework for traffic anomaly detection. To validate the framework, in the following chapters as shown in Figure 3.10, the application of the proposed framework based on case studies both for the early detection of recurrent congestion and the detection of non-recurrent congestion will be explored.

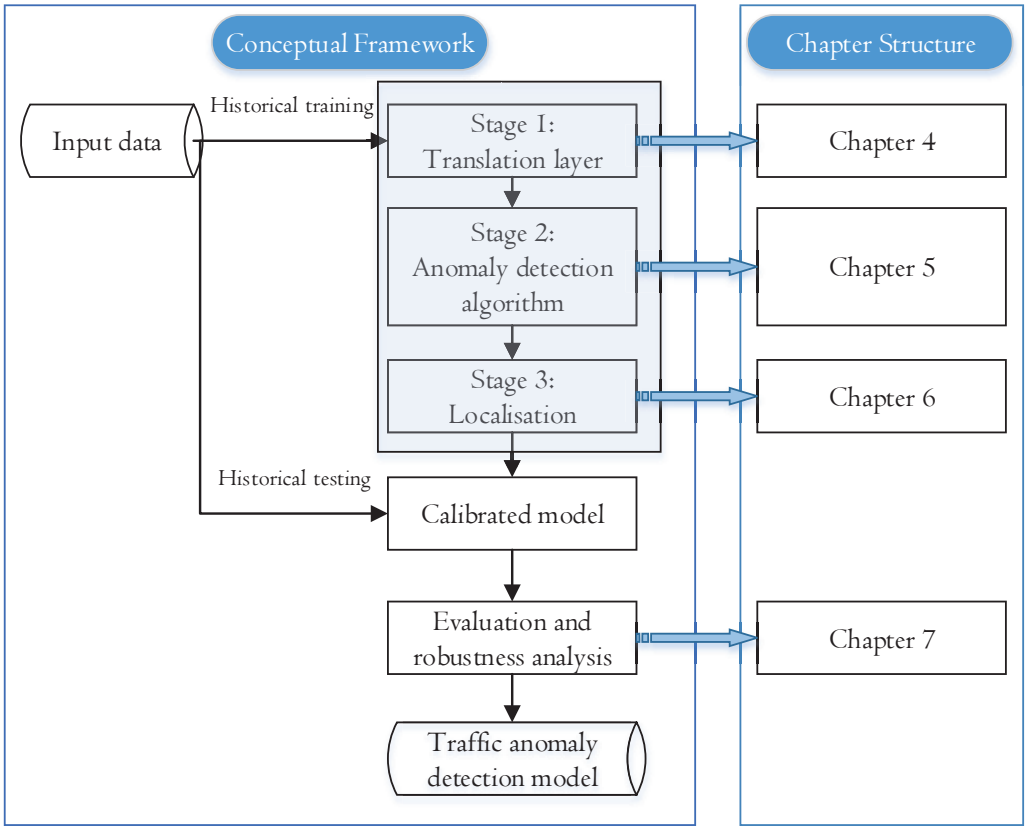


Figure 3.10: The conceptual methodology framework in relation to the chapter structure

Chapter 4

Translation Layers

Chapter 3 developed a general traffic anomaly detection framework that introduces the concept of a translation layer to deal with the inputs in the first stage of a detection model. One of the main challenges regarding network analysis incorporating spatio-temporal features is to transform the data from complex networks into a simple but representative format that can be processed by the subsequent components of the framework. This chapter will address this challenge by using two empirical case studies to evaluate the application of the proposed translation layer for the problem of recurrent congestion and non-recurrent congestion.

This chapter is organised as follows. Section 4.1 gives the research background and main aim of the chapter. Section 4.2 presents three translation layer methods, namely 1) connectivity matrix, 2) geographical grid and 3) spatio-temporal translation. Section 4.3 provides the evaluation results for these translation layer methods based on a case study in the city of Bath in respect to the detection and early prediction of recurrent congestion detection, while Section 4.4 does the same for the detection of non-recurrent congestion based on a case study in London. The workflow of the two empirical case studies is composed of a comprehensive analysis of the data, calibration, results and discussion. Section 4.5 summarises the main findings in this chapter and concludes according to the aim and objective set out in Chapter 1.

4.1 Introduction and Background

In the literature, most existing studies on RC prediction and NRC detection have focused mainly on upstream and downstream corridor-based analysis using either data-driven classification and clustering methods or fundamental diagrams (i.e., traffic speed, density and traffic flow diagrams). Nevertheless, the spatio-temporal dynamics of traffic in a network play an important role, especially in the propagation of congestion propagation arising from traffic anomalies. To date, limited research has systemically investigated traffic anomaly detection with attention to the spatio-temporal dynamics of traffic in a network.

Despite limited literature on traffic anomaly detection at a network level, some efforts which contribute to the task of estimating or predicting traffic variables, such as traffic speed (Ma et al., 2017) and traffic flow (Zhang et al., 2018a), can potentially be used or extended for this traffic anomaly detection research. In this regard, one important step in all the literature related to network analysis is to transfer or translate the complexity of the traffic network into a matrix that can reflect its spatio-temporal connections. For example, Ma et al. (2017) presented a time-space matrix representation to translate data from a large scale network into a two-dimensional (i.e., time and space) matrix which has been used as an image input for convolutional neural networks. In that study, the authors have evaluated the translation methods using two case studies: 1) a simple network with a set of circular roads and 2) a more complicated network with some roads connected to circular roads. Their research indicated that this spatio-temporal translation is very straightforward for networks with a simple topology, such as the first network, but for the second a compromise translation that segmented the roads into straight lines had to be proposed. The limitation of this compromise is that only the correlation among the segmented straight lines can be captured regardless of connections from other links crossing the straight lines. Nonetheless, this study represents a good starting point for the required time-space translation.

Zhang et al. (2018a) proposed a grid-based translation considering both inflow and outflow for traffic flow forecasting. Even though this method is a direct translation with geographical spatial correlations, the key issue affecting the performance was the size of each cell or the level of aggregation. It should be noted that this grid-based method can be ideally adapted to data analysis at a zone level. For example, the grid-based method could be used for origin-

destination matrix generation for traditional travel demand modelling. Since this method is good at the spatial correlations but less good at reflecting temporal relationships such as period, trend and closeness, a potential direction for future study could be developing a method that can incorporate the temporal features into such a model. On the other hand, Zhu et al. (2018a) investigated the use of a connectivity matrix as a translation layer for traffic incident detection based on the Sioux-Falls network. The connectivity matrix approach benefits from its simplicity in implementation and the ability to take account of direction and weights based on traffic variables, and is a well understood concept in transport planning and analysis. On the other hand, connectivity matrices can be sparse for large networks. The existing studies into network translation are summarised below in Table 4.1.

Table 4.1: Summary of existing studies in the network translation

Reference	Translation Methods	Advantages	Limitations
Ma et al., 2017 Wu et al., 2018c Kim et al., 2018 Ke et al., 2019	Time-space translation	This method which is widely accepted in transport research can directly generate a matrix with time and space features.	It is difficult to deal with complex spatial relations.
Zhang et al., 2018a Du et al., 2019 Bao et al., 2019 Yu et al., 2017	Geographically grid-based translation	It is capable of capturing spatial correlations.	The matrix generated by this method is spatially aggregated to some extent. In addition, it is computationally expensive.
Zhu et al., 2018a	Connectivity matrix	The translation is straightforward and based on the physical connection of different nodes.	The transformed matrix is sparse and sometimes large in the case of larger traffic networks.
Zheng et al., 2019	Feature matrix representation	The translation represents all features associated with the data in different cells in the matrix.	The selection of features and their combinations increases the complexity of the problem. It is dependent on the availability of multiple features.

In summary, some relevant studies have presented possible methods to translate a traffic network into a simple matrix. While these studies represent a substantial contribution to traffic network analysis, some research challenges and gaps remain. Firstly, limited research has been devoted to the traffic anomaly detection problem, including both recurrent congestion detection and non-recurrent congestion detection. Moreover, the methods reviewed above have limitations as well as advantages and thus a systematic evaluation is needed comprehensively to assess their performance in the traffic anomaly detection domain. Based on the objective of this research, the following sections set out in detail options to extend or adapt the previous studies into the traffic anomaly detection domain, together with a systematic evaluation of their performance in this context in the light of the detection metrics set out in Section 2.3.3.

4.2 Proposed Translation Layers

Out of all the translation layer methods reviewed in the last section, feature matrix representation depends significantly on the availability of multiple traffic features and has the problem of complexity in feature selection. Based on the limitations of the feature matrix representation, it was excluded from the comparative analysis in this chapter. Accordingly, only the connectivity matrix, grid-based translation and spatio-temporal translation are presented in this research. This section will introduce their adaptions to traffic anomaly detection problems, i.e., recurrent congestion detection and non-recurrent congestion detection. This section will also introduce the rationale behind these three translation layers.

4.2.1 Connectivity Matrix

As briefly introduced in Section 4.2.1, a connectivity matrix can build a matrix that represents direct connections between all links in a network for each time stamp. In this straightforward translation, the traffic variables will be naturally placed in a two dimensional matrix (see Figure 3.5) when the time window has not been taken into account. Time lags k can be added as another dimension into the connectivity matrix, as shown in Algorithm 1.

Algorithm 1: Connectivity Matrix Translation

Input: Historical traffic observations: $\{\mathbf{X}^0, \mathbf{X}^1, \ldots, \mathbf{X}^t\}$
Network corridor's connection with origin i and destination j: $S_{ij} \in \{0, 1\}$
Number of previous time steps or time lags: k
Output: A series of matrix $\{\mathbf{A}^0_{ijk}, \mathbf{A}^1_{ijk}, \ldots, \mathbf{A}^t_{ijk}\}$
Initialisation;
for *enumerate* i, j **do**
  if $S_{ij} = 1$ **then**
    Comments: if directly connected, assign the traffic variables; otherwise pass;
    Assign traffic observations $\mathbf{X}^t$ to $\mathbf{A}^t_{ij}$;
  else
    Pass;
  end
end
for *enumerate* t **do**
  Generate three-dimensional matrix with depth of k: $\mathbf{A}^t_{ijk}$
end

4.2.2 Geographical Grid

Broadly, the geographical grid based method is a graphical representation of raw road network data where each individual value contained in the matrix is an averaged traffic variable, such as average traffic flow or traffic speed. A typical example of geographical grid based translation is the heatmap which originated in 2D displays of the values in a data matrix.

Although the traditional link based flow prediction model is an efficient way to represent flows, it ignores its dependence on neighbouring road links. Thus, a geographical grid based method which could take into account the traffic conditions of neighbouring links would be a promising way to generate matrix input. For example, Zhang et al. (2018a) attempted to use the concept of traffic flow grid matrix as the input for traffic flow predictions. The input was decomposed into different channels such as inflow and outflow. A similar translation was also validated by Du et al. (2019) to predict the traffic passenger flows based on the flow heat maps from Beijing and New York. Bao et al. (2019) fused multiple data sources in a geographical grid and fed these into a deep learning approach in order to predict the crash risk in New York City. Based on its effectiveness in traffic prediction, geographical grid translation could have the potential for traffic anomaly detection.

Algorithm 2 shows the geographical grid procedure used in this thesis. The main properties

for the method are the dimension of the geographical grid, i.e., (i, j), historical traffic variables $\mathbf{X}$ and depth of the input or the time lags k. According to the physical location of the traffic sensors, the traffic flow in cell (i, j) in time slot t is the average of all the inflows or outflows. Similar to the connectivity matrix, the previous k time steps data can be added as an additional layer of depth of the matrix input to represent the time lags.

Algorithm 2: Geographical Grid Translation

Input: Historical traffic observations: $\{\mathbf{X}^0, \mathbf{X}^1, \ldots, \mathbf{X}^t\}$
Grid dimension: i, j
Number of previous time steps: k
Output: A series of matrix $\{\mathbf{A}^0_{ijk}, \mathbf{A}^1_{ijk}, \ldots, \mathbf{A}^t_{ijk}\}$
Initialisation;
for *enumerate* i, j **do**
  Calculate the average of inflow: $\mathcal{S}^t_{in,i,j}$;
  Calculate the average of outflow: $\mathcal{S}^t_{out,i,j}$;
  Inflow and outflow: $(\mathcal{S}^t_{in,i,j}, \mathcal{S}^t_{out,i,j})$
end
for *enumerate* t **do**
  Generate three-dimensional matrix with depth of k: $\mathbf{A}^t_{ijk}$
end

4.2.3 Spatio-Temporal Translation

Compared to the previous two translation methods whose initial two-dimensional matrix only includes the spatial information, spatio-temporal translation could incorporate temporal information directly during the construction of the initial matrix. Some studies have made some attempts at this space-time translation. For example, Ma et al. (2017) proposed a time-space matrix using time and space dimension information as the input for traffic prediction. The case studies were based on two types of topologies where the former is a simple circle while the latter is a set of segmented roads. Given the direct space and time translation, the research concluded that the proposed methods based on spatio-temporal input can help to extract the abstract features and therefore improve the accuracy of prediction. Another similar study has been done by Wu et al. (2018c) where the spatial and temporal features were extracted respectively and then fed into a matrix input for CNN.

Kim et al. (2018) assigned traffic speed according to the order of road segments and time steps, resulting in an image representation of traffic speed data. Even though the proposed

methods are commonly accepted and promising, however, their research does not discuss the logic of how to rank the sequence of road segments. Similarly, Ke et al. (2019) converted multi-lane traffic flow raw data to the multi-channel spatial-temporal matrix based on the same logic as the study of Kim et al. (2018) where the converted matrix represents one dimension as the space while the other dimension is taken as time steps or time spans. There remain gaps in the current studies, therefore, in respect to translating a realistic topology with complex interactions, as well as in determining the logic by which to order the sequence in the space dimension.

Considering the nature of traffic flows, this study proposed ranking the spatial sequence according to the direction in which traffic moves. With complex networks where multiple traffic directions are given, such as at a junction, the sequence will be ranked in the light of the order of historical traffic flow values. The point with the largest traffic flow will be picked first while the non-selected points neighbouring the same junction will be added into a pending list to be added only on the termination of the sequence starting with the largest traffic flow. Several conditions that would require changing the direction have been defined in the algorithm:

- 1) all points connected with the last point added into the sequence have already been included in the sequence;
- 2) all points in the pending list have already been included in the sequence list; and
- 3) if there are no points left in the pending list, a new start point should be generated randomly to be appended to the sequence list.

In summary, the algorithm starts from a random point in the traffic network and the sequence will be generated automatically based on the historical traffic variable characteristics, as formulated in Algorithm 3.

Section 4.3 and Section 4.4 will focus on testing these three translation layers in the context of both the recurrent congestion problem and the non-recurrent congestion problem. Case studies for the two research problems will be briefly introduced before evaluating the translation layers using the deep learning methods proposed in Chapter 3.

Algorithm 3: Spatio-temporal Translation

Input: Historical traffic observations: $\{\mathbf{X}^0, \mathbf{X}^1, \ldots, \mathbf{X}^t\}$
Number of spatial sequence: i
Number of time steps: j Number of day: d Number of selection methods: k
Output: A series of matrix $\{\mathbf{A}^0_{ijk}, \mathbf{A}^1_{ijk}, \ldots, \mathbf{A}^t_{ijk}\}$
Initialisation;
Random select a start point a;
Comments: selected by the maximum outflow
for *enumerate* n *points directly connected to point* a **do**
  if $X_{ba} = \max\{X_{1a}, X_{2a}, \ldots, X_{na}\}$ **then**
    Point b is appended into the sequence next to a;
  else
    Append into the pending list for selection after current sequence
  end
end
repeat
until *All* i *points have been placed into the sequence*;

Assign the traffic variables to matrix A^t_{ijk}

4.2.4 Summary

According to the methodological framework proposed in Section 3.2, the first main component is the translation layer. This section specifically formulated three translation layer methods, i.e., connectivity matrix, geographical grid and spatio-temporal translation. In general, the methods proposed here all have advantages and disadvantages. In order to verify their applicability, following sections will evaluate them with case studies of early RC detection and NRC detection.

4.3 Case Study of Early Recurrent Congestion Detection

As presented above, the main objective of this section is to evaluate the candidate translation layers based on the evaluation metrics proposed in Section 2.3.3 for recurrent congestion detection. The focus of recurrent congestion includes two parts: recurrent congestion detection and early prediction.

This section starts by briefly introducing the study area in the city of Bath. Then, before testing the translation layers, the selected traffic features are validated using scatter plots. Next, according to the process of generating labels based on the identification of traffic states proposed in Section 3.2.2.1, labels which will be used later for recurrent congestion detection are generated using the Expectation Maximisation method. The early prediction results are evaluated in terms of metrics (i.e., DR, FPR, Precision, F1 score, AUC and F1 score), confusion metrics and ROC curve based on the translated input, generated output and the proposed deep learning method. Furthermore, to evaluate how early it can detect or predict, a sensitivity analysis is presented with a variety of eight prediction horizons and eight time lags.

4.3.1 Data Description

Traffic flow and occupancy data are extracted from the study area of the City of Bath. To evaluate the proposed early recurrent congestion detection method, the study network consists of a small network with four sub links, i.e., two Eastbound roads (link 1 and link 4), one Westbound road (link 2) and a Southbound road (link 3), with 10, 6, 3 and 5 detectors respectively, as shown in Figure 4.1. The rationale behind this selection of the Lower Bristol Road and Upper Bristol Road in Bath is that the corridors are arterial roads which among the busiest roads connecting central Bath with neighbouring cities and rural areas.

The study area is geographically located within two main corridors with different traffic compositions. The traffic data covers every day from 7:00 to 18:55 in 15-minute time intervals over the course of two years from June 2015 to June 2017, with a total sample size of 35424. For an example of flow and occupancy data from Bath, please refer to Appendix B. The typical traffic flow profile time series for all detectors in the study area at different times of day are shown in Figure 4.2. The traffic flow time series clearly shows different traffic patterns since

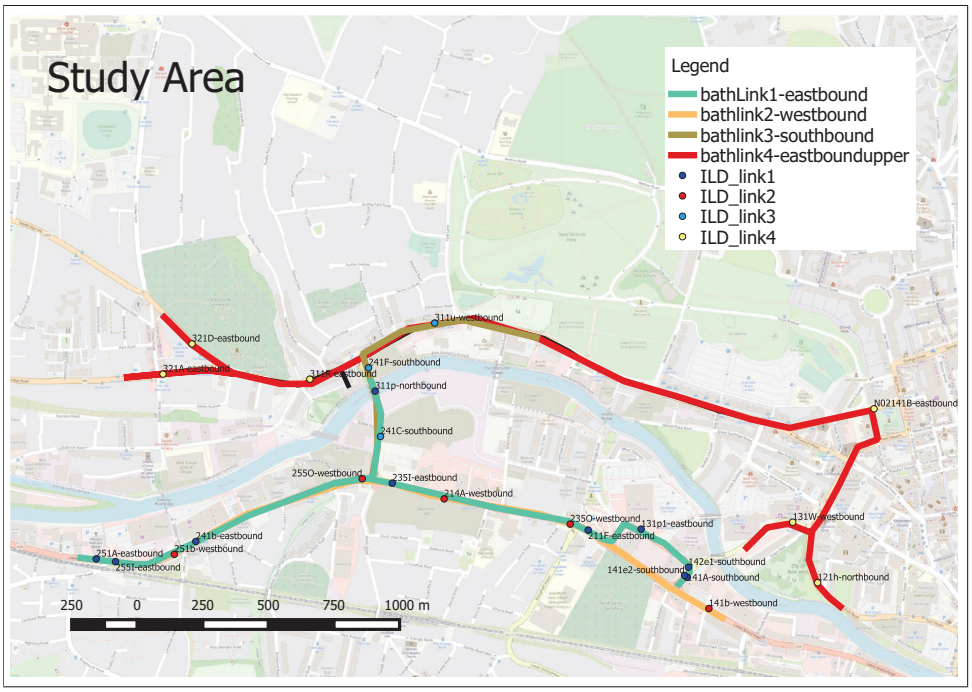


Figure 4.1: Study area in the city of Bath (Map Source: OpenStreetMap)

different streets have different profile shapes. The aggregated average traffic flow for time of day (i.e., 48-time points per day), day of week and the average traffic flow of sequenced detectors on study roads are shown in Figure 4.3. It is clear that the volume of traffic on weekdays is higher than on weekends for study roads.

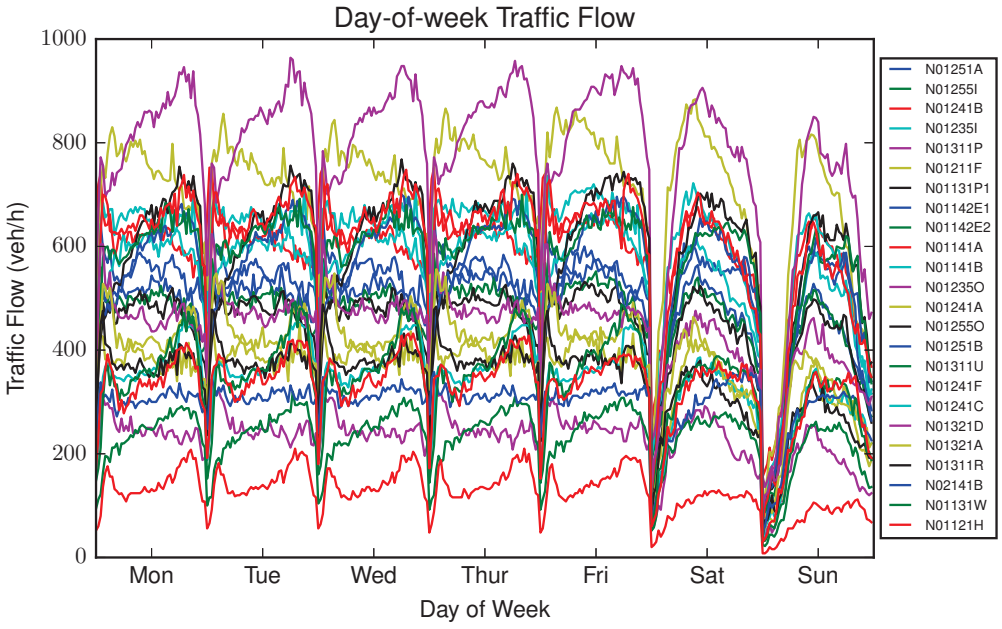


Figure 4.2: Traffic flow profile in the study area

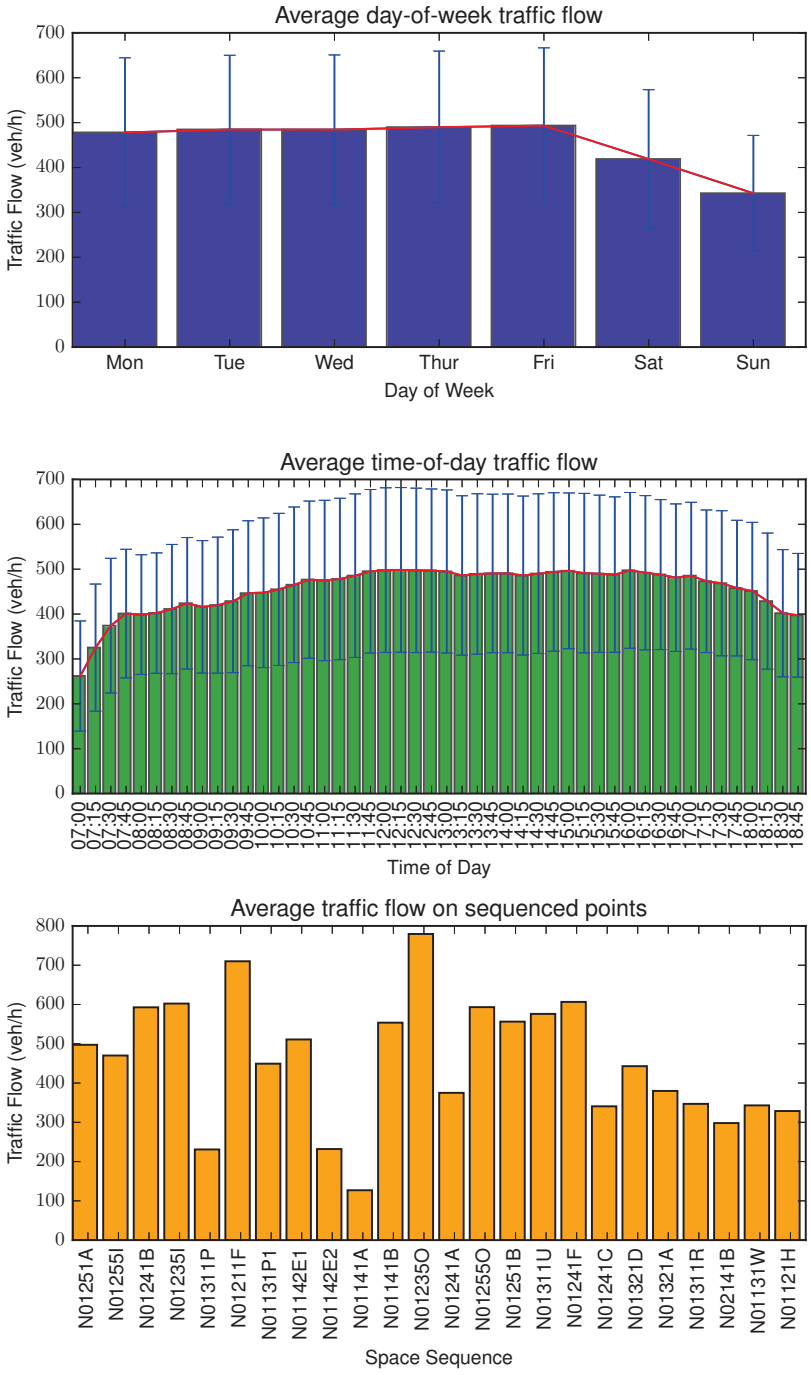


Figure 4.3: The average traffic flow in terms of day of week, time of day and 24 sequenced detectors in the RC study area

4.3.2 Congestion Label Generation

In respect to the detection of recurrent congestion, there are no available traffic states or congestion labels for the detection. As mentioned above, therefore, the Expectation Maximisation (EM) traffic state identification method has been used to generate the recurrent congestion labels (Han et al., 2010). Generally, the EM algorithm is used to find maximum likelihood estimates of parameters in statistical models, formulated in Eq. 3.8. The EM algorithm conducts an iteration of an expectation (E) step, which creates an expectation of the log-likelihood for the parameters, and a maximisation (M) step, which maximises the log-likelihood on the E step. Figure 4.4 shows the workflow for the E step and M step, where $Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(l-1)})$ is the expected value of the log-likelihood function of parameters set $\boldsymbol{\Theta}$, with respect to the current distribution of traffic states α and current estimates of the parameters $\boldsymbol{\Theta}$. The M step is to find the parameters that maximise $Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(l-1)})$ using differentiation with respect to different unknown parameters, i.e., Gaussian distribution parameters $\theta_k = (\mu_k, \gamma_k^2)$ and the mixture factors r_k.

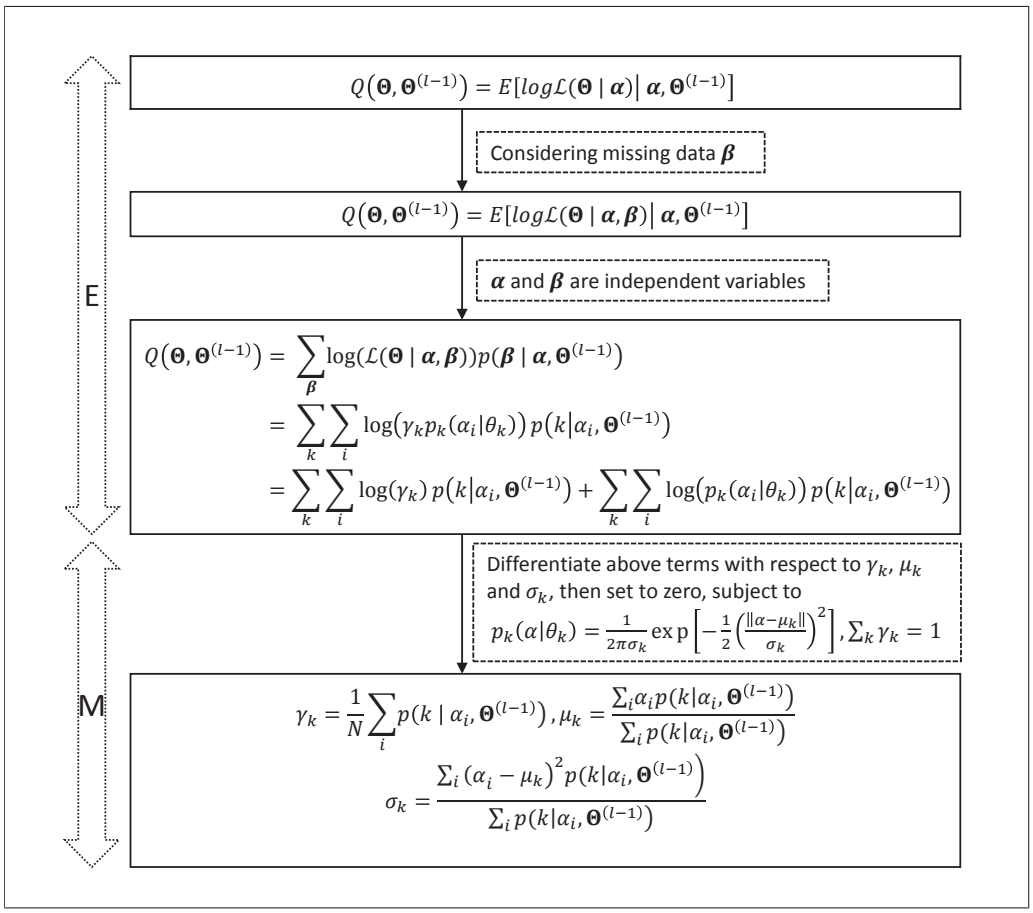


Figure 4.4: EM algorithm summary flowchart

Using the EM algorithm above, the two regimes of congestion and non-congestion can be clustered based on the busiest road segment. The scatter plots of the EM binary cluster results are shown in Figure 4.5 where the EM algorithm classifies the traffic states into two regimes based on traffic occupancy and traffic flow and assigns the reasonable probability of congestion for the observations. It can be observed from Figure 4.5 that the study area has a clear and credible separation for congestion and non-congestion regimes. This reflects the fact that the EM algorithm can optimise traffic states according to the iterative expectation and maximisation steps. The level of congestion is not simply decided by traffic flow and occupancy. As a result, the input for early detection consists of both traffic flow and occupancy features.

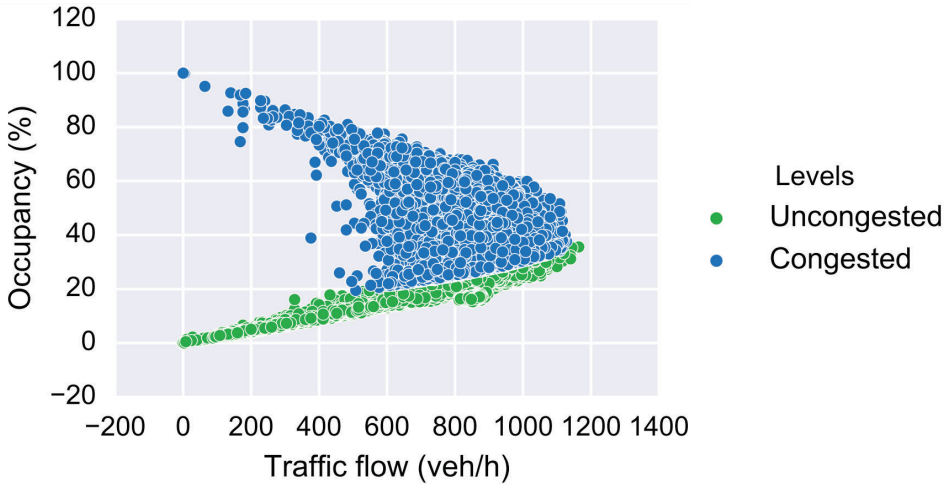


Figure 4.5: Traffic flow and occupancy patterns and traffic states clustered by the EM algorithm

The labelled traffic states can be simplified as Eq. 4.1 where 0 represents non-congestion and 1 stands for congestion. Varying levels of traffic congestion could possibly be defined, but as the main objective of this chapter is to evaluate the proposed translation layers, the commonly used binary labels are sufficient here.

$$f(x_t) = \begin{cases} 0, & \text{No congestion at time } t. \\ 1, & \text{Congestion at time } t. \end{cases} \tag{4.1}$$

4.3.3 Input Layer Representation

In this section, the data of the study area will be converted into multi-dimensional input according to the proposed translation layers. Figure 4.6, Figure 4.7 and Figure 4.8 show the translated input for the early detection model based on connectivity matrix, grid and spatio-temporal translation respectively.

The connectivity matrix translation is very straightforward. An example of translated traffic flow in a two-dimensional format is shown in Figure 4.6. The nodes in the connectivity matrix are main junctions inside the study area so that values inside the connectivity cell represent the traffic features of the link connecting two associated junctions. In this case study, twenty junctions have been identified in the study area. Thus, this translation eventually results in a $(2\times20\times20)$ matrix where 2 refers to two selected traffic features, i.e., traffic flow and occupancy, and (20×20) is the connectivity matrix with either traffic flow or occupancy values. Figure 4.6 shows an example with traffic flow values filled in each cell.

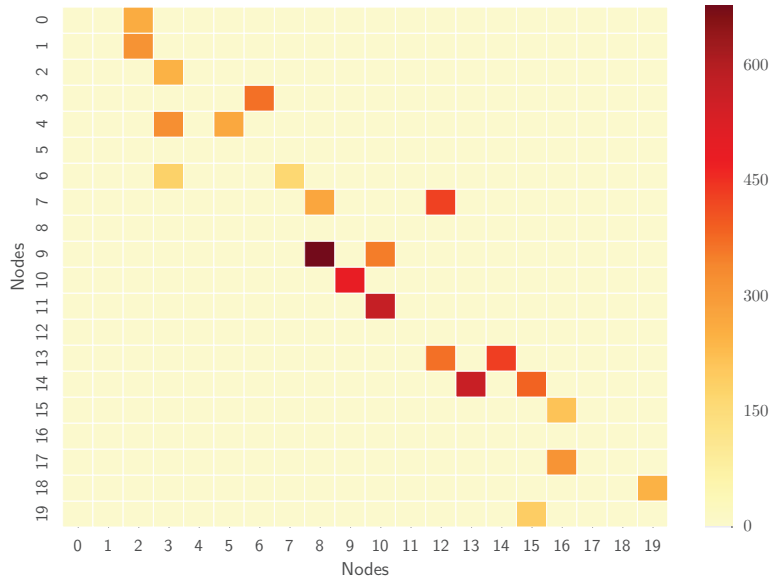


Figure 4.6: Input matrix with traffic flow values (veh/h) converted by the connectivity matrix for RC detection

Bao et al. (2019) tested different sizes of cells ($3 \times 8, 5 \times 15$ and 10×30) for crash risk prediction and suggested that the performance of crash risk prediction will decrease as the resolution increases. In this study, a relatively lower resolution with an x coordinate of three and a y coordinate of nine has been selected in the light of the size of the study area. Therefore,

the final cell size in Figure 4.7 is 380m which is a reasonable size for the city given that each cell includes maximum one ILD for one direction and the distribution of detectors is relatively sparse. This size was selected with the guidance that each cell has no more than one detection for each direction. In summary, the resulted input matrix using this translation is with the size of $(4 \times 9 \times 3)$, where 4 refers to traffic flows and occupancy for both inflow direction and outflow direction, while the rest stands for the size of the grid.

Figure 4.8 shows an example of typical spatial temporal translation, where the spatial sequence has been ordered according to the spatio-temporal algorithm proposed in the last section and the temporal sequence has been placed naturally in the light of the 15-minute time spans. The dimension after this type of translation is $(k \times 24 \times 2)$ where k is the number of time lags, 24 is the number of detectors and 2 stands for the two selected traffic features.

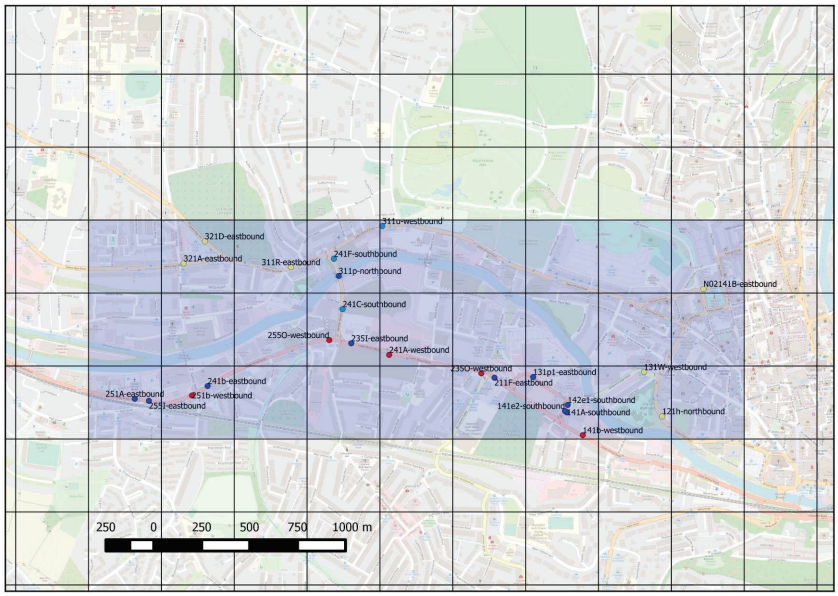
4.3.4 Model Calibration

In the previous section, two general traffic state classes, i.e., congestion and non-congestion, are clustered based on the EM algorithm. To evaluate the performance in respect to the early recurrent detection problem over a two hour period (i.e. with the change of time horizons and time windows from zero to seven), 8×8 experiments are assessed for each translation layer proposed in this section. Given this, a total of 192 early recurrent congestion detection tasks with different spatio-temporal combinations are developed and compared.

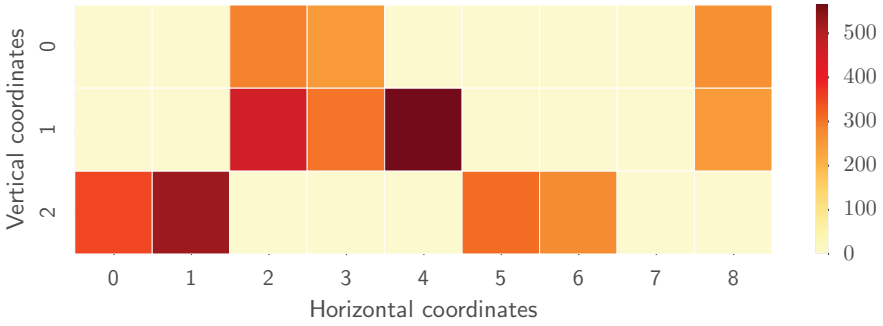
It is worth highlighting that the early detection model takes the information from a certain portion of time, defined by a past number of samples (i.e., time window), and predicts in a certain time step ahead (i.e., time horizons). The certain time ahead can help to indicate how far in advance the model can detect RC congestion.

The hybrid model based on CNN and LSTM is used for early RC detection. Instead of introducing further complexity to the evaluation of the translation layer, the early congestion prediction model used in this section is configured based on hyperparameters that have been commonly deployed in many previous studies (Krizhevsky et al., 2012; Ma et al., 2015b, 2017; Wu et al., 2018c; Zhu et al., 2018a; Dabiri and Heaslip, 2018; Bao et al., 2019).

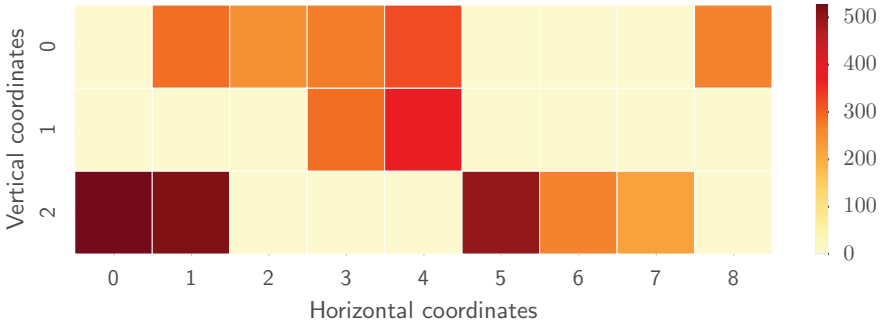
Specifically, there are two types of parameters considered when implementing the structure



(a) Grid map (source: OpenStreetMap)



(b) Translated inflow



(c) Translated outflow

Figure 4.7: Input matrix with traffic flow values (veh/h) converted by geographical grid translation for RC detection

of a CNN: the hyperparameters connected with each layer and the depth of the CNN, respectively (Ma et al., 2017). Table 4.2 provides a summary of these two types of CNN hyperparameters.

The selection of the appropriate hyperparameters connected with each layer, i.e., max pool-

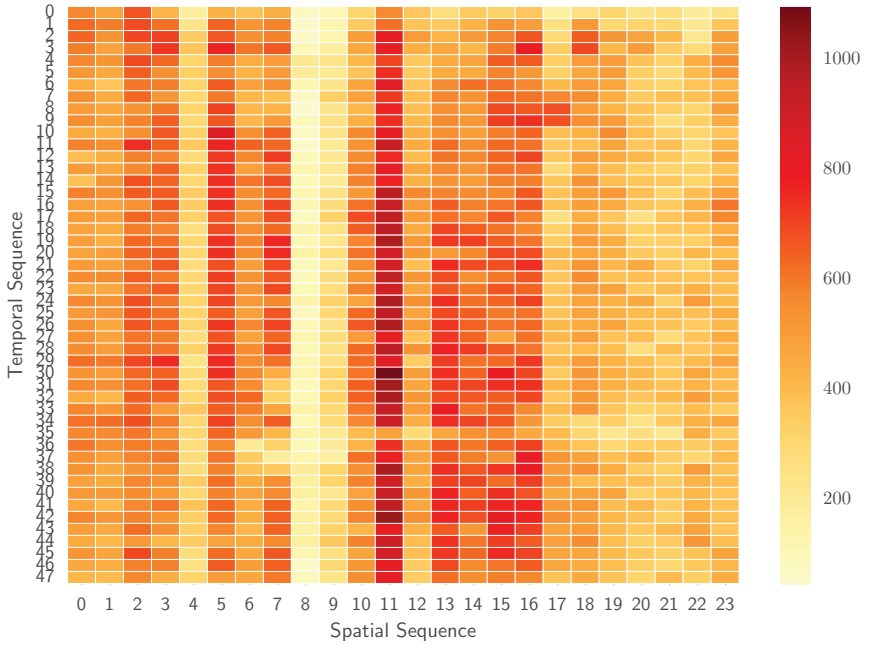


Figure 4.8: Input matrix with traffic flow values (veh/h) converted by spatio-temporal translation for RC detection

ing kernel size and convolutional filter size, relies heavily on expert judgement. Since no general rules can be directly used to find the optimal values of hyperparameters, several well-known CNN architectures, such as LeNet (LeCun et al., 1998), AlexNet (Krizhevsky et al., 2012) and VGG Net (Simonyan and Zisserman, 2014), are referred to. In respect to the max pooling size, 2×2, which has been adopted widely both in the AlexNet and the LeNet-5, is selected because it is a typical pooling kernel shape that has been shown to reduce the size of feature maps effectively. As for the convolutional layer kernel size, the size of 3×3 with a kernel number of 64 is used since it has the benefit of a decrease in the number of parameters and multiple applications of ReLU layers (Simonyan and Zisserman, 2014; Dabiri and Heaslip, 2018). For LSTM layers, different hidden cell sizes of 256 and 512, which are commonly used in previous studies (Ma et al., 2015b,a), are selected to determine the optimal sizes of hidden cells in different early recurrent congestion detection scenarios.

The depth of CNN, meanwhile, is often set to be neither too deep nor too shallow, so that the CNN can efficiently learn the complex spatial structure while still being guaranteed to converge at the end (Krizhevsky et al., 2012). Since the size will be reduced substantially during

Table 4.2: Summary of hyperparameters of a CNN

Parameters	**Function**
Parameter type I : Architecture (the depth of CNN)	
Convolutional layer (some research separates this step into a convolutional layer and an activation layer)	a) Computing the output of neurons that are connected to local regions in the input; and b) extracting the feature maps from the input matrix
Maxing pool layer	a) Neighbouring groups of feature maps; and b) reducing the size of feature maps
Fully connection layer	a) Computing the class scores resulting in a reduced output size; b) reducing the number of outputs by using matrix multiplication with a bias offset; and c) minimising the loss function
Parameter type II : Inner parameters in each layer (hyperparameters)	
Convolutional kernel number	The number of features
Convolutional kernel size	The size of convolutional regions
Convolutional kernel stride	The moving distance between each centre convolutional step
Activation function	ReLU/Sigmoid
Pooling layer kernel size	The size of local pooling regions
Pooling layer stride	The moving distance between each centre pooling step
Dropout proportion	The proportion of data dropped between layers to deal with the problem of overfitting
Fully connected neuron number	The number of classes
Loss function optimiser	Accuracy, root mean square error, etc.

convolution and pooling, however, the depth of CNN in this study is constrained by the input size of the matrix and thus the final net contains five learned layers including two convolutional layers, two LSTM and one fully-connected layer.

In summary, the net contains five layers with weights: the first two are convolutional layers, each followed by a max pooling, two LSTM layers, and the remaining one layer is a fully-connected layer, where the last fully connected layer is fed to a Sigmoid activation function to output the predicted probability of the binary classes. The convolutional kernels are connected to all kernels in the previous layer and the ReLU activation function is used for the output of every convolutional layer and the first fully-connected layer.

The CNN-LSTM is trained based on the RMSprop optimiser (Tieleman and Hinton, 2012) in the back propagation process. RMSprop serves as an optimiser to utilise the magnitude of recent gradients in order to minimise the loss function and has achieved remarkable performances in many previous studies (Zhu et al., 2018a; Ma et al., 2015b). It is used to keep a moving average of the squared gradient for each weight, and to update the weight and bias in each iteration during the optimisation. Given the decay rate γ and the learning rate η, the parameters are updated as follows:

$$r_t = \gamma f'(\theta_{t-1})^2 + (1-\gamma) f'(\theta_t)^2 \tag{4.2}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{r_t + \epsilon}} f'(\theta_t) \tag{4.3}$$

where $f'(\theta_t)$ is the gradient or the derivative of the loss function $\mathcal{L}$ with respect to the parameters θ at time step t and ϵ is an error term. The learning rate is set to be 0.001 with a 0.9 decay.

In order to avoid the problem of overfitting (Hawkins, 2004), a dropout technique (Srivastava et al., 2014) is used. This consists of setting the output of each hidden neuron to zero with a fixed probability (Hinton et al., 2012b). The dropped-out neurons, therefore, do not participate in the optimisation. Consequently, the neural network sub samples a different architecture, and all these architectures share weights (Krizhevsky et al., 2012). Since a neuron cannot depend on the presence of specific other neurons, dropout reduces complex co-adaptations of neurons and hence forces the neuron to learn more robust features. The ratio of a 0.5 dropout rate used in studies (Ma et al., 2015a; Zhu et al., 2018a; Dabiri and Heaslip, 2018) is also used in this study to reduce the potential overfitting. In addition to the dropout technique, performance trends based on training and validation were checked before detecting using the testing dataset.

4.3.5 Results and Analysis

The proposed early detection method is tested in this section, based on the input generated from Section 4.3.3 and supervised labels from the EM method. Three representative methods, i.e., MLP, RF and GBC, reviewed in Section 2.3.1 have been selected as baseline models for comparison. These methods have the ability to capture the complex and non-linear relationship between different features. The input data for the above statistical and machine learning methods will be normalised time-series traffic variable data. Furthermore, stochastic gradient methods are also used for these methods in order to minimise the error and optimise weights and bias step by step. On the other hand, the output for the conventional methods are the same labels or traffic states as for the proposed deep learning methods.

The performance is evaluated based on the following metrics:

- detection rate;
- false postive rate;
- F1 score;
- precision;
- area under the curve;
- confusion matrix; and
- ROC curve.

On the other hand, to evaluate how early the model can detect the RC, the sensitivity analysis based on different time windows and time horizons ranging from zero to eight is applied.

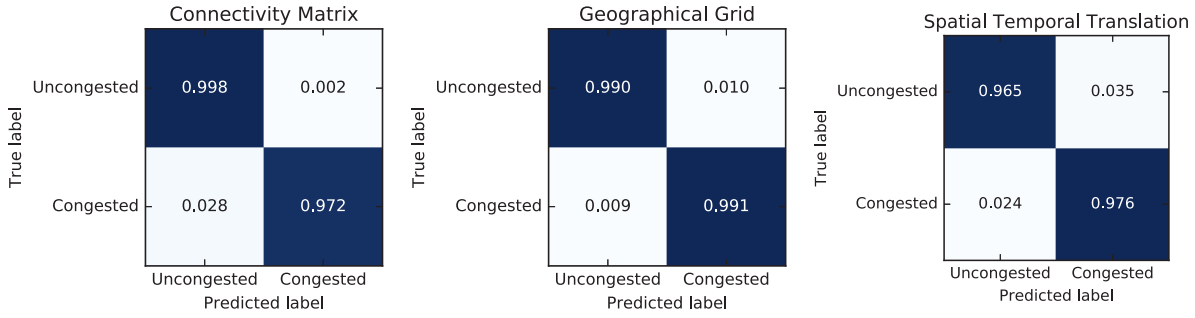
Table 4.3 shows the results of early detection with data samples in the current time step (i.e., time lag $= 0$ and prediction horizon $= 0$). In general, even though all translation layers show promising results with high accuracy and low false alarm rate, there is no translation layer that outperforms others in terms of all six evaluation metrics. Nonetheless, it is possible to identify the method which best fits the specific intended purpose of early detection of recurrent

congestion. For example, in terms of DR, which measures the percentage of true recurrent congestion detected, it is obvious that the geographical grid outperforms the others. On the other hand, if the aim is to achieve lower false alarm rates and high precision, the connectivity matrix could be selected. One may find the same recommendation based on Figure 4.9 which shows the confusion matrix for three translation layers. The values inside the confusion matrix represent the true negative rate, FPR, false negative rate and DR.

In this research, statistical tests of significance which are generally used to quantify evidence against a particular hypothesis being true by mainly using P-values are not used. One particular reason that P-value could not be used in this research is that using P values as the sole arbiter of what to accept as truth can also mean that some analyses are biased, some false positives are overhyped and some genuine effects are overlooked (Wasserstein et al., 2019).

Table 4.3: Recurrent congestion detection results based on different translation layers

Translation Layer	DR	FPR	F1 score	Precision	AUC
Connectivity Matrix	0.972	0.002	0.985	0.997	0.9993
Geographical Grid	0.991	0.010	0.988	0.985	0.9987
Spatial Temporal Translation	0.976	0.035	0.963	0.951	0.9958



(a) Connectivity Matrix (b) Geographical Grid (c) Spatial Temporal Translation

Figure 4.9: Recurrent congestion detection confusion matrix based on different translation layers

Figure 4.10 shows their corresponding ROC curves. Generally, a ROC curve is another common way to evaluate and intuitively visualise the performance of binary classifiers, and is generated by plotting the true positive rate versus the false positive rate for all thresholds ranging from 0 to 1. The ROC curve is plotted with true positive rates versus the FPR for all thresholds ranging from 0 to 1. Two main features are typically checked to interpret the ROC curve: 1) the position of the curve and 2) the steepness of the curve. Specifically, since the classifier with high

true positive rate and low FPR is always preferred, a good classifier is generally closed to the left upper corner and with a steep curve. Thus, for this case study, it is obvious that the rank in terms of ROC curve is connectivity matrix, geographical grid and spatial temporal translation.

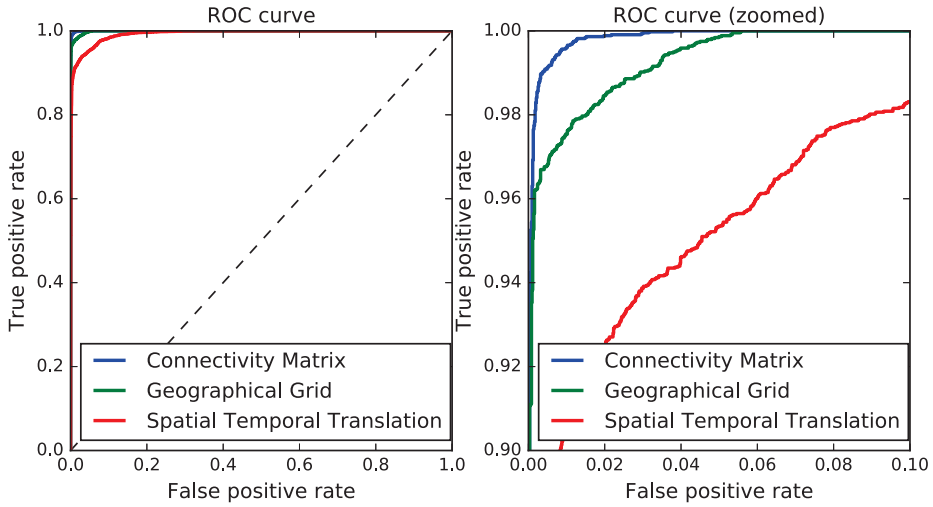
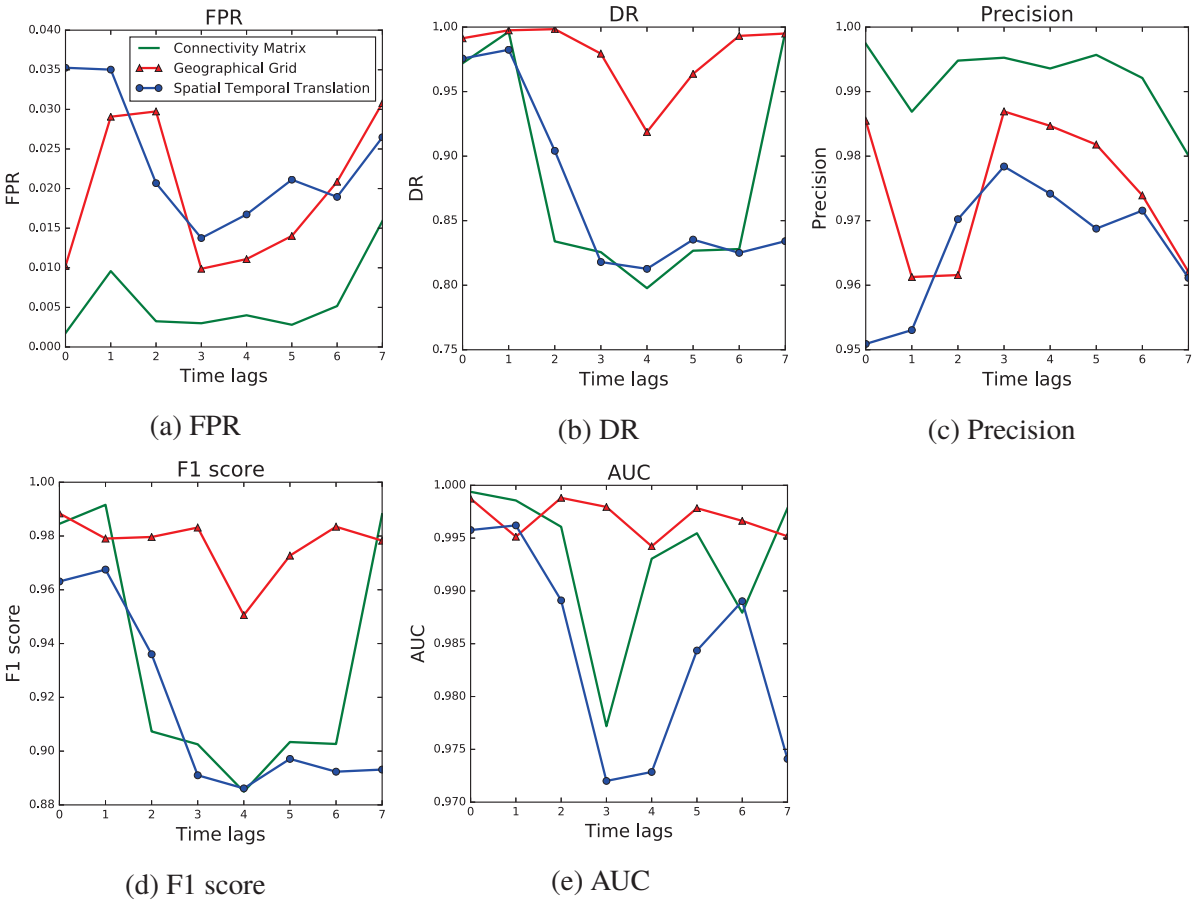


Figure 4.10: Recurrent congestion detection ROC curve

Figure 4.11 shows an example of a sensitivity analysis regarding time lags ranging from zero to seven which represents a time window from the current time step to a maximum time step of over two hours. Input data with different time lags have been used to predict multiple time steps (i.e., time horizons) ahead. There are several interesting points that can be summarised. First, the connectivity matrix consistently outperforms other translations in terms of lower FPR and high precision, regardless of the number of time lags, Second, the performance of the geographical grid is superior than the others based on DR, F1 score and AUC, no matter how many time lags are used. Thirdly, despite there being no unified performance conclusion in terms of the selection of time lags, it is obvious that for all translation layers there is a bowl shape in the metrics. For example, in the FPR plot, when the time lag $= 3$ all the various translation layers result in a rather small false alarm rate. Overall, though, the results are not consistent across the various metrics, translation layers and time lags, and the reason for this is that different metrics basically measure different types of performance of detection, as defined in Section 2.3.3. The three translation layers may have their individual advantages and limitations as they incorporate the traffic characteristics in varying formats. As for the time windows with a time interval of 15 minutes, including a long time windows (i.e., larger than one hour) may introduce noise into the detection and therefore a bowl shape should be expected. Note that the results above are based

on a prediction horizon equal to zero, a full result with all prediction horizons is presented in Appendix C.1.



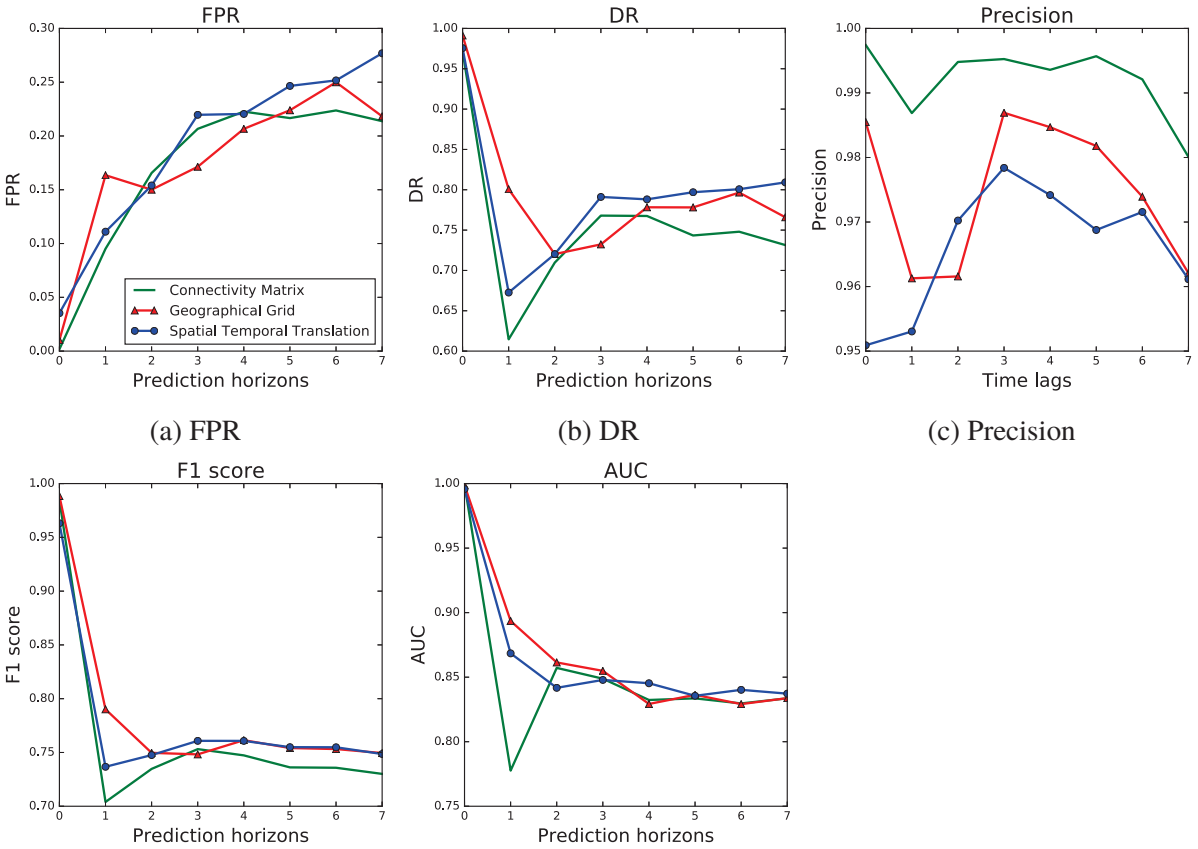
(a) FPR (b) DR (c) Precision

(d) F1 score (e) AUC

Figure 4.11: Recurrent congestion detection in terms of time lags with prediction horizon=0

Figure 4.12 presents the sensitivity in terms of prediction horizons. Compared to the sensitivity analysis for time lags where there is no generally consistent position for the relationship between all metrics and time lags, it is notable that the performance is gradually reducing as the prediction horizon increases. This decrease in the performance is reasonable as the traffic pattern may remain similar within a short time period. Another interesting point to make is that even though there is no consistent superior performance in terms of all metrics for the three translation layers, the connectivity matrix can obviously detect the traffic states more precisely according to the precision metrics. Regarding the prediction horizon, the performance for predicting around two hours, or seven steps, ahead is still acceptable with around 0.2 FPR, 0.82 DR, 0.96 precision which means the model is capable of early prediction of recurrent congestion.The horizons may be interpreted further through a comparison with baseline models, which will be followed up in Chapter 5. The results above are examples with the time lag equal to zero, a full

result with all time lags is illustrated in Appendix C.1.



(a) FPR (b) DR (c) Precision

(d) F1 score (e) AUC

Figure 4.12: Recurrent congestion detection in terms of prediction horizons with time lag=0

4.3.6 Summary of Early Recurrent Congestion Prediction

In summary, three translation layers have been evaluated in this section for RC detection and early prediction. The connectivity matrix may be chosen if a low FPR and high precision are required for the early prediction task while the geographical grid is attractive when a high detection rate and F1 score is sought. For the performance in terms of time lags and prediction horizons, the prediction result will decrease when the prediction horizon increases, but the accuracy for prediction two-hours ahead is still acceptable. On the other hand, the performance change in terms of time lags is on a case-by-case basis. Nevertheless, even though it is hard to find a conclusive result for performance based on the varying size of time lags, the bowl shape curves indicate that prediction with a time lag equal to three may be more accurate with lower FPR and higher precision.

4.4 Case Study of Non Recurrent Congestion Detection

The previous section mainly tested the translation layers for early recurrent congestion, evaluating three different translation methods based on varying sizes of time lags and prediction horizons. In this section, these three translation layers will be further evaluated for non-recurrent congestion detection, i.e., traffic incident detection. The NRC problem needs external knowledge on the occurrence and characteristics of traffic incidents, and thus a different case study area with both traffic incident data and traffic data has been used. Similar to the previous section, the evaluation will start with a description of the data for non-recurrent congestion detection, followed by the presentation of translation layers based on the data of an empirical study area. The non-recurrent congestion detection model used in this section is a CNN model with typical parameters listed in Section 4.3.4, supervised by traffic incident data obtained from Transport for London (TfL).

4.4.1 Data Description

Two types of data are used in this section, i.e., traffic data and incident data. These are both extracted from the study area near Russell Square in Central London which has a complex traffic network topology and varying traffic volumes. The study area contains three main train stations (i.e., Euston Station, King's Cross Station and St Pancras International Station), connecting London with other cities and countries, making it one of the busiest areas in Central London. The area also encompasses eight underground stations, hundreds of bus stations, part of Oxford Street (London's main shopping high street), well-known museums such as the British Museum and part of the University of London. In the following sections, a detailed description of traffic incident data and traffic data will be presented before an analysis of their relationship.

4.4.1.1 Traffic Incident Data

Traffic incident data from London was recorded in the Traffic Incident Management System (TIMS) and provided by TfL. The detailed description of traffic incident has been included in Appendix B.2. The case study area consists of 158 links and 39 of these links were affected by 97 unplanned incident events and 50 planned events during the study period from 1st Jan 2015

to 24th Mar 2015. The location of the study area, as well as traffic incident heatmaps of the London study area as a whole are shown in Figure 4.13 and Figure 4.14, respectively.

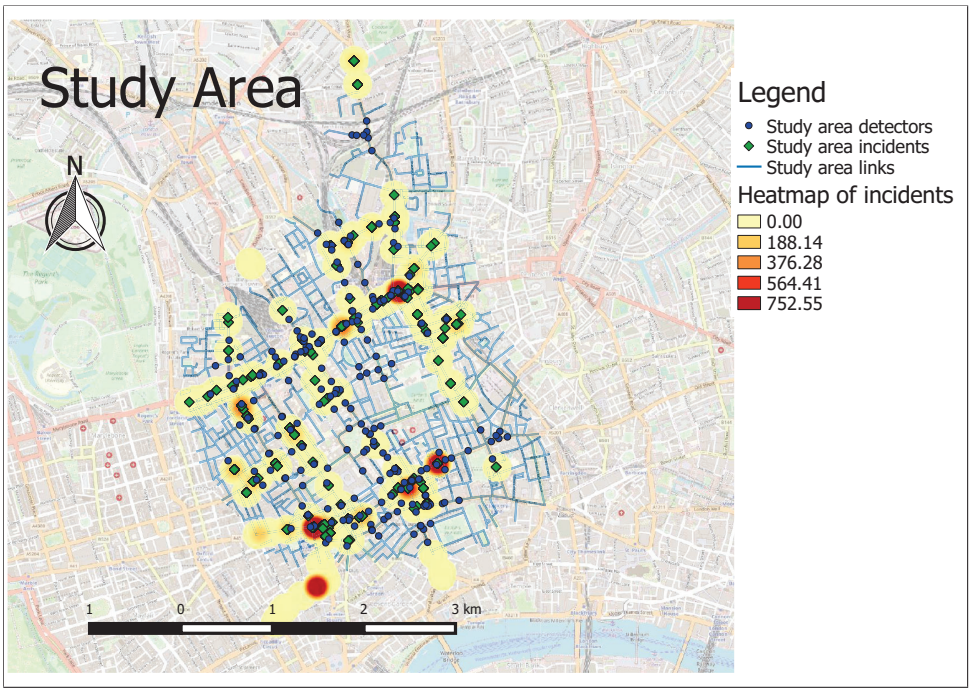


Figure 4.13: Study area and incident heatmap for NRC detection (Map Source: OpenStreetMap)

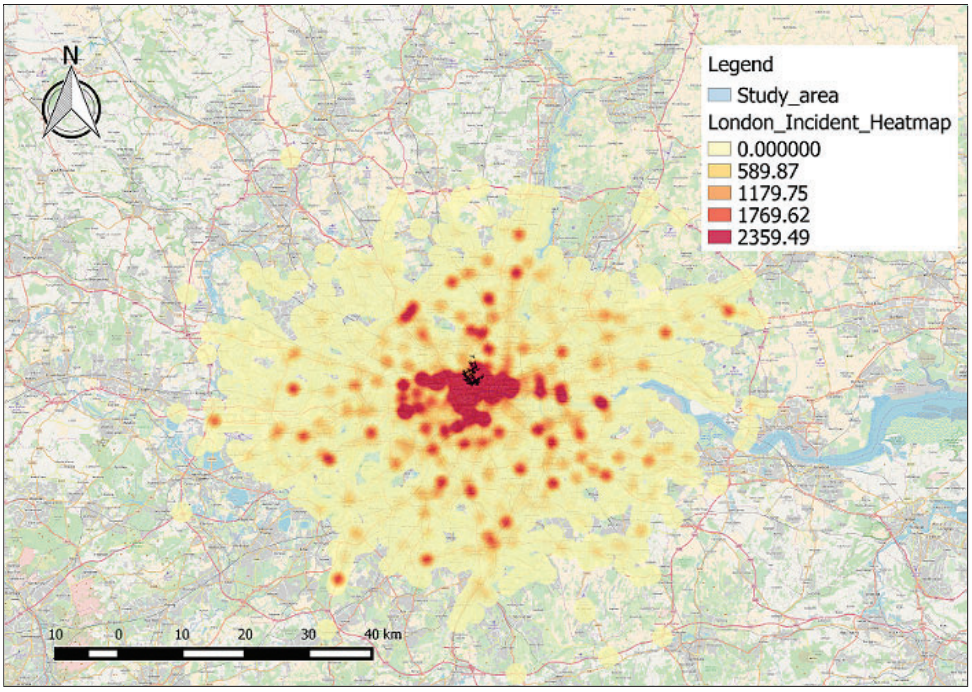


Figure 4.14: London incident heatmap (Map Source: OpenStreetMap)

As shown in Figure 4.13 and Figure 4.14, there was a large number of traffic incidents in the study area in central London. The numbers in the heatmaps represent the record of traffic

incidents in a one-hour resolution, which means that one traffic incident may be recorded several times if it lasted for several hours. The detailed traffic incident data has been included in the Appendix B.

Varying types and levels of severity of traffic incidents are recorded in the traffic incident dataset. Table 4.4 shows the detailed composition of traffic incidents in the study area. In general, there are two types of traffic incidents, i.e., planned and unplanned. According to the description given by TfL, the definitions of each category are as follows.

Table 4.4: Incident counts of the study area for NRC detection

Incident Category	Planned	Unplanned	Total
Traffic Incidents	0	61	61
Works	42	8	50
Hazard(s)	0	9	9
Special and Planned Events	8	0	8
Infrastructure Issue	0	17	17
Traffic Volume	0	2	2
Total	50	97	147

- Traffic incidents: collision, breakdown, emergency services incident.
- Works: utility replacement works, borough redevelopment works, TfL works,etc.
- Hazards: obstruction, spillage, surface damage, flooding, etc.
- Special and planned events: ceremonial event, exhibition, construction activity, etc.
- Infrastructure issue: traffic signals, barriers, etc.
- Traffic volume: tube strike, Christmas shopping, Operation Stack[1], etc.

Roadworks scheduled in advance account for a large proportion of the planned traffic incidents, while for the unplanned category, car accidents, generally occurring randomly, are the main types accounting for roughly 67% of all incidents.

Figure 4.15 illustrates the overall distribution of different types of traffic incidents within each of the two broad categories of planned and unplanned incidents, in terms of their duration.

[1]Operation Stack is a procedure to park (or 'stack') lorries whenever there is an urgent need to inhibit the flow of traffic because of some disruptions such as bad weather or industrial action.

The distribution plots reveal that: 1) the majority of planned anomalies lasted for a duration of less than five hours while, occasionally, some emergent infrastructure issues or hazards may lead to incidents with a long duration of around 10-15 hours; 2) the duration distribution of unplanned events has a long tail, i.e., a typical right-skewed distribution, which means the mean duration is longer than the duration of most frequent road works for unplanned events; and 3) the average planned roadworks tend to have a longer duration than that of unplanned works which may result from the generally heavier workload needed for the planned roadworks.

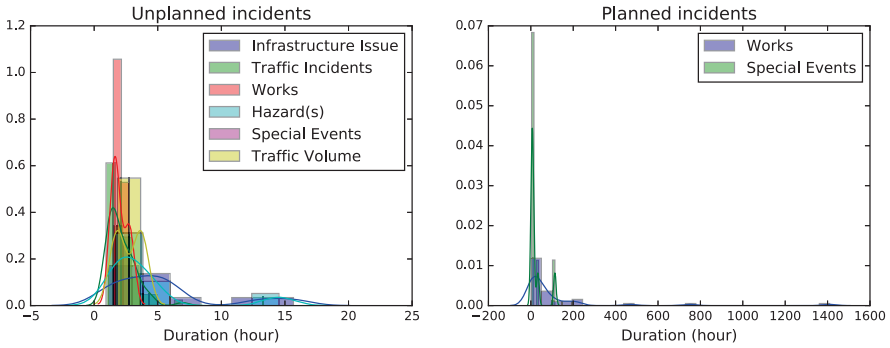


Figure 4.15: Traffic anomaly distribution in the study area with different types

Figure 4.16 shows the distribution of different types of traffic anomaly in terms of the time of day. The number in the bar represents the size of intervals of varied traffic anomaly types falling into each time of the day. Across all times of the day, by far the largest number of intervals is devoted to works, probably due to the longer duration of roadworks. Most car accidents occurred during the morning peak hours and afternoon peak hours.

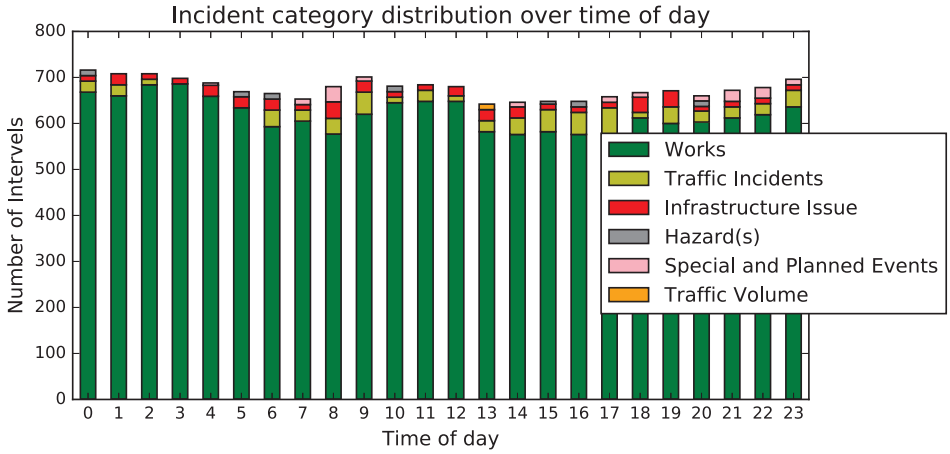


Figure 4.16: Traffic anomaly distribution in the study area with different types over time of day

4.4.1.2 Traffic Data

The traffic data, including traffic flow and occupancy, are recorded by Inductive Loop Detector (ILD). ILDs are widely used for providing inputs to the Split Cycle Offset Optimisation Technique (SCOOT) traffic control system (Hunt et al., 1981). They report vehicle presence or absence (0/1 values) sampled at 4Hz at the fixed location. Traffic variables such as flow and occupancy can be calculated from the reported data. The detailed calculation process has been explained in previous studies (Krishnan, 2008; Zhu, 2015), please refer to Appendix B for details. The traffic flow data have been aggregated into 5-min intervals between 00:00 and 23:55 every day during the course of around 80 days from 1st Jan 2015 to 24th Mar 2015.

The typical traffic flow time series for all detectors on different days of the week are shown in Figure 4.17, where a daily peak and off-peak traffic pattern of traffic volumes is evident. The traffic flow time series data show remarkably different patterns of traffic states due to the varying levels of ILD sensitivities and the characteristics of observing traffic. The aggregated average traffic flow for day of the week and for time of day, i.e., 288-time points per day, are shown in Figure 4.18. Similarly to the traffic data in the Bath case study, it is clear that the volume of traffic on weekdays is generally higher than that at weekends. The traffic flow between 9:00 and 18:00 is consistently high because of the large traffic volumes in the study area of Central London.

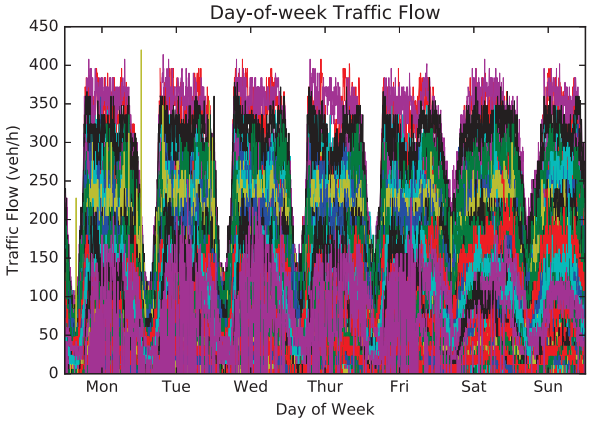


Figure 4.17: Traffic flow profile in the study area

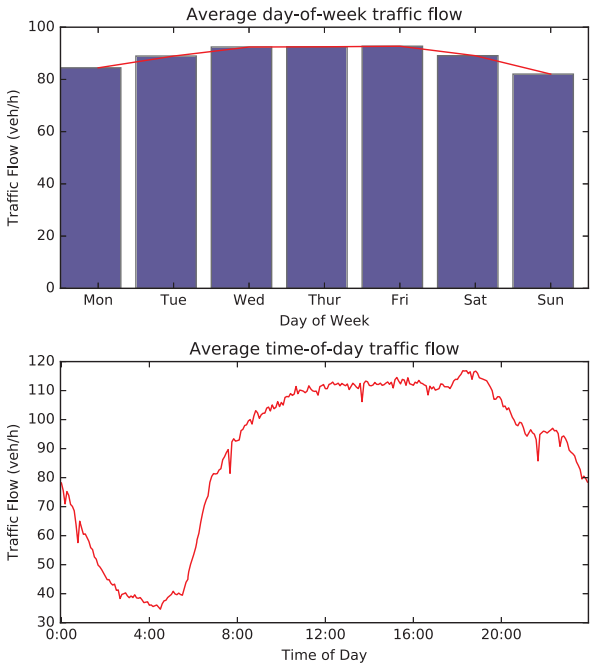
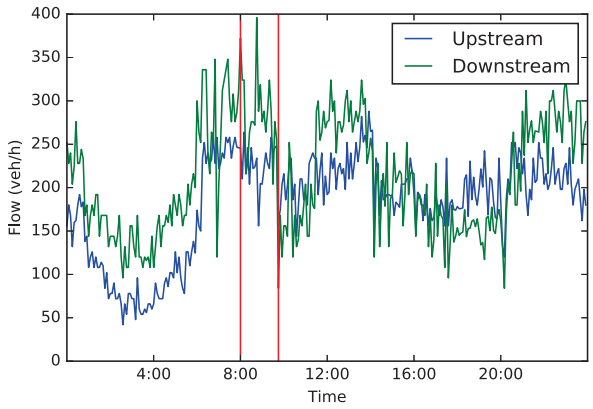


Figure 4.18: The average traffic flow in terms of day of week, time of day in the NRC study area

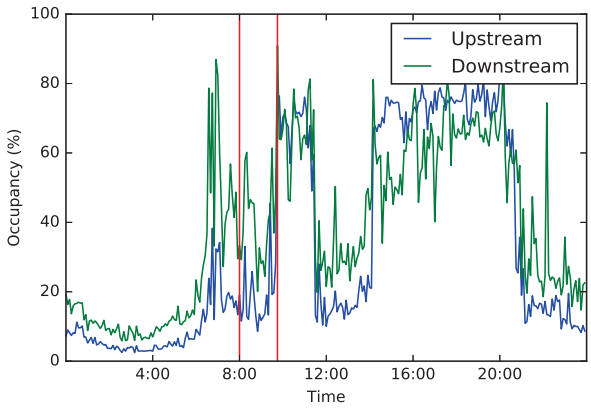
4.4.1.3 Relationship of Traffic Incident to Traffic Data

Figure 4.19 shows the relationship between traffic flow and occupancy, on the one hand, and traffic anomalies on the other hand (as recorded in the traffic incident dataset), both upstream and downstream of the anomaly point. During the traffic anomaly, the traffic flow in the upstream segment will drop significantly while the flow in the downstream will increase dramatically, followed by a sharp decrease.

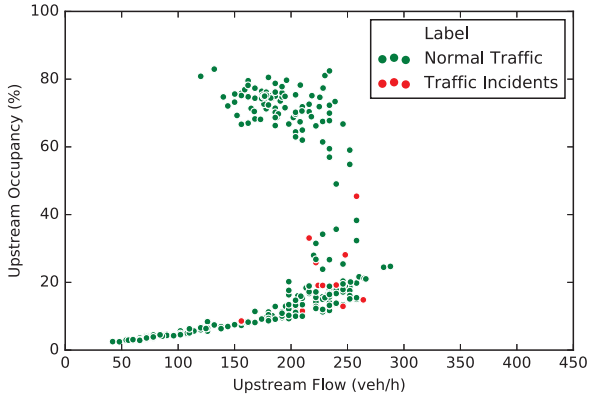
On the other hand, while the occupancy upstream will increase immediately after the traffic anomaly, since the traffic formulation during the traffic anomaly starts from the normal traffic for that time of day and day of the week, and thus already exhibits a wide potential variation in traffic jam severity, the scatter plots with the labels of traffic incidents may fall into different regimes of the LoS category. Thus, it is more difficult for the detection algorithm to capture this non-linear relationship between traffic data and incident data in the case of non-recurrent congestion than with the recurrent congestion problem, which in general has a clear separation of different traffic states.



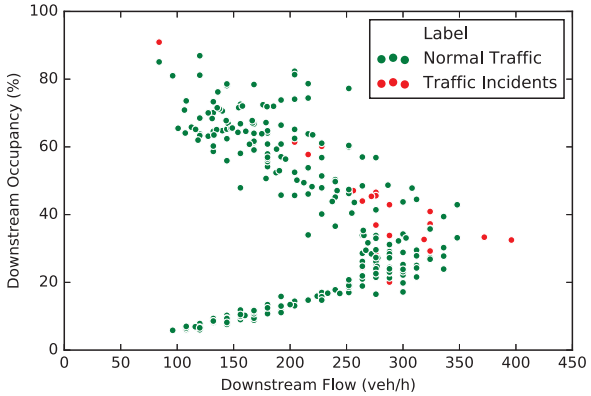
(a) Traffic flow of upstream and downstream



(b) Traffic occupancy of upstream and downstream



(c) Flow and occupancy of upstream



(d) Flow and occupancy of downstream

Figure 4.19: Relationship of traffic data and non-recurrent congestion data

4.4.2 Input Layer Translation

Following this detailed description of the traffic and incident data, this section translates that data into multiple dimensions according to the proposed translation layers. Figure 4.20 shows the input for the non-recurrent congestion detection model based on a connectivity matrix. Specifically, 112 main junctions were identified as the nodes in the study area, resulting in a final connectivity matrix input dimension of $(2 \times 112 \times 112)$ where 2 refers to two traffic features, i.e., traffic flow and occupancy, and 112 is the number of main junctions in the study area. It is worth noting that the sparsity of the matrix will increase exponentially with the increase in the size of the study area due to its assumption of direct connection. This kind of sparsity may be dealt with by taking into account two or multiple connections instead of one direct connection. Although this multiple connection analysis is beyond the scope of this research, it may be worth discussing in future research.

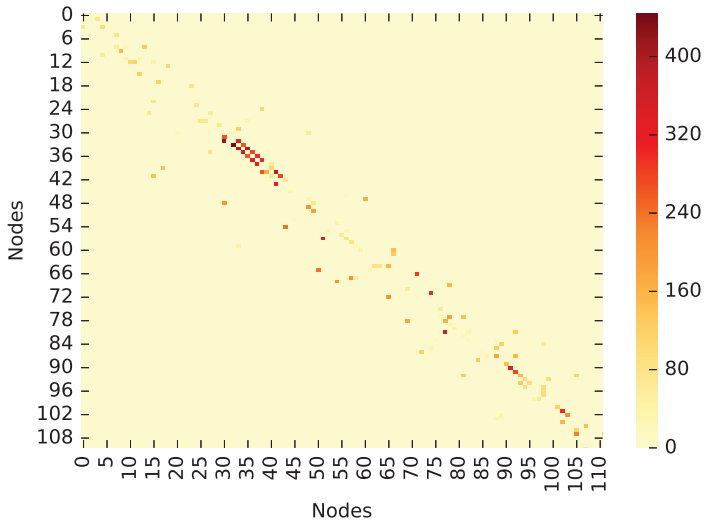
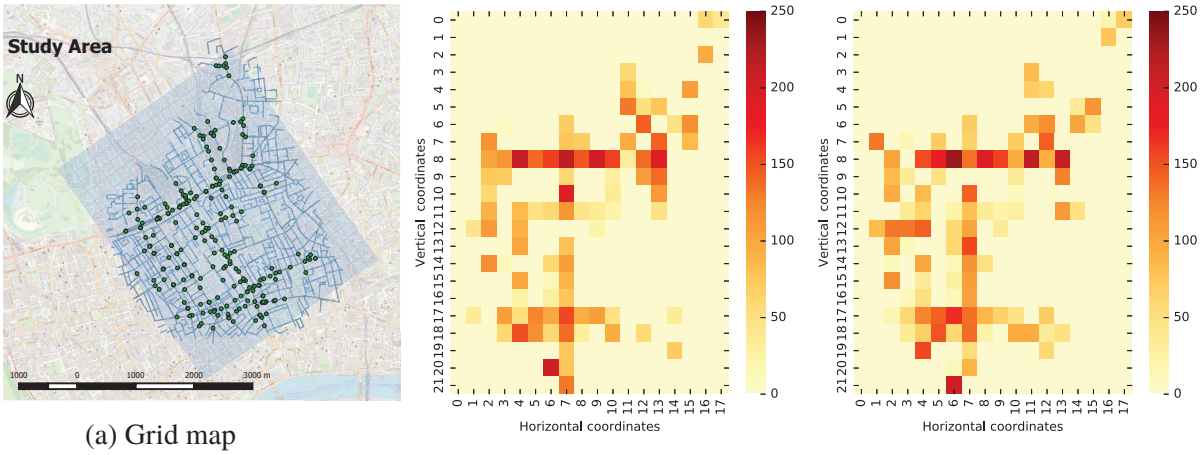


Figure 4.20: Input matrix with traffic flow values (veh/h) converted by the connectivity matrix for NRC detection

With regards to geographical grid translation, Figure 4.21 shows the geographical grid translation for the London case study. The position of the grid has been determined based on the directions of the main corridors in the study area, with the aim of guaranteeing that the neighbouring detectors in the map are placed directly next to each other in the grid. Considering the dense distribution of detectors in London, and based on a previous study (Bao et al., 2019), the resolution of the grid was increased to 200m to ensure each cell has a maximum of one detector for each direction and also to reduce the number of cells with no detectors. This grid leads to a final size of translated input of $4 \times 22 \times 18$ where 4 refers to the two traffic features for both the inflow and outflow direction and (22×18) are the axes of the translated input.



(a) Grid map
(Map Source: OpenStreetMap)

(b) Translated inflow

(c) Translated outflow

Figure 4.21: Input matrix converted by geographical grid translation for NRC detection

As mentioned in the Bath case study, the time lags have been included in the depth of the matrix instead of as a separate temporal dimension. Thus, for the spatio-temporal translation, the final dimension after this type of translation is $(k \times 237 \times 2)$ where k is number of time lags, 237 is the number of detectors and 2 stands for the two selected traffic features for each detector.

4.4.3 Model Calibration and Output

After the translation of input layers, two other important components for the non-recurrent congestion detection are the proposed algorithm and the labels. For the selection of the relevant hyperparameters readers can refer to Section 4.3.4.

Regarding the output labels, the labels of the busiest main corridor, namely Euston Road, have been used in order to simplify the evaluation. Figure 4.22 shows an overview of Euston road with the distribution of incidents. Euston road connects three main stations, three underground stations (Warren Street Station, Euston Station and King's Cross St.Pancras Station) and a couple of bus stations. During the study period, 33 traffic anomalies, i.e., 22.4% of the traffic anomalies in the study area, were identified on this road, as listed in Table 4.5 where the major type of non-recurrent congestion was traffic incidents, or car accidents.

Table 4.5: Incident counts on the Euston Road

Incident Category	Planned	Unplanned	Total
Infrastructure Issue	0	3	3
Traffic Incidents	0	20	20
Hazard(s)	0	4	4
Works	5	1	6
Total	5	28	33

Figure 4.23 shows the distribution of different types of traffic anomalies on Euston Road. As for the unplanned incidents, four types, namely infrastructure issue, traffic incidents, roadworks and hazards, occurred during the study period. Most traffic incidents lasted around 2-3 hours while the infrastructure issues and hazards had a relatively longer duration, i.e., 5-10 hours. In respect to planned incidents, roadworks were the only types that occurred during the study period, and it is obvious that the duration of planned roadworks is much longer than the

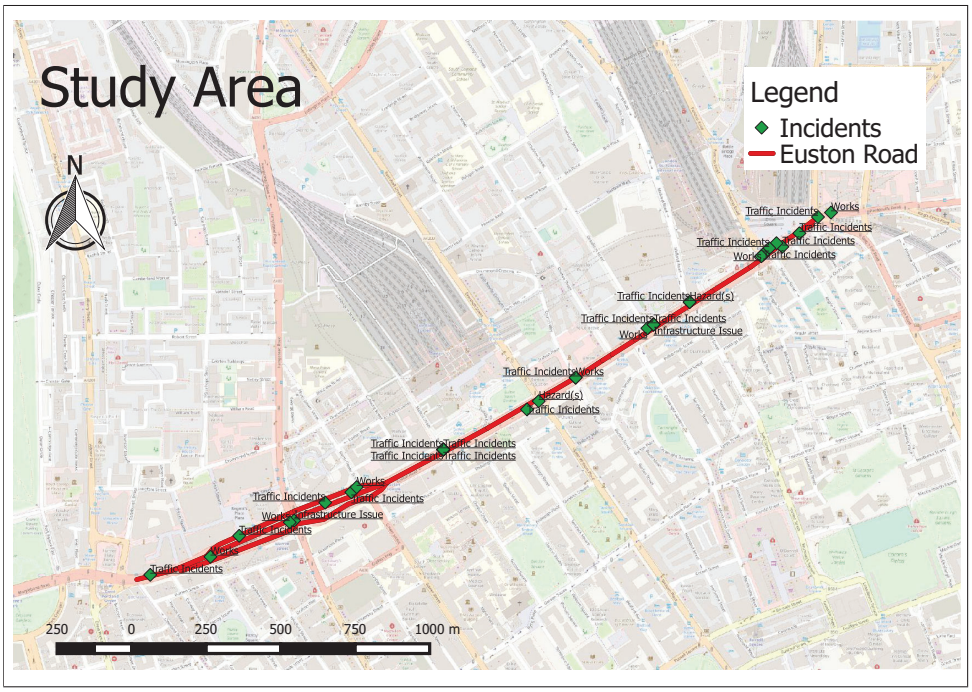


Figure 4.22: Traffic anomaly distribution on Euston Road (Map Source: OpenStreetMap)

unplanned works shown in the left figure. This longer period may because that the planned works tend to involve a larger maintenance project.

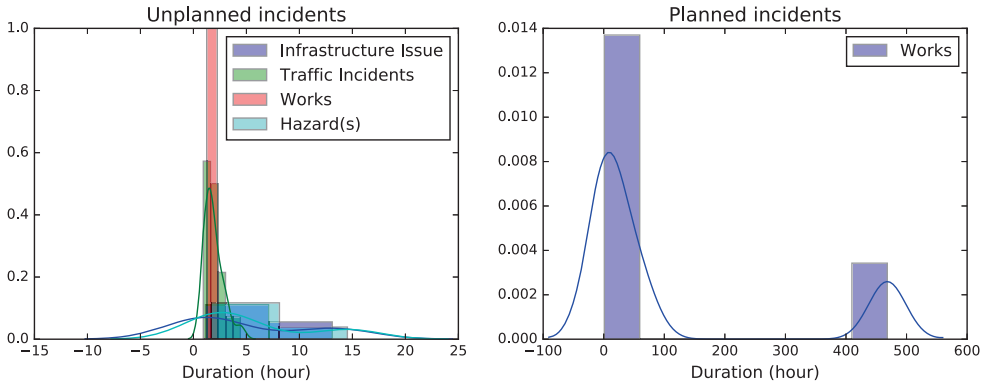


Figure 4.23: Distribution of different types of traffic anomaly on Euston Road

Figure 4.24 presents a histogram of the distribution of varied types of traffic anomaly according to the time of day. As expected, roadworks account for most intervals (5 minutes for each interval) across the whole day while the other three types occur over a smaller number of intervals. The car accidents may have a longer duration during the noon and midnight period which lead to a larger number of intervals involving in traffic incidents.

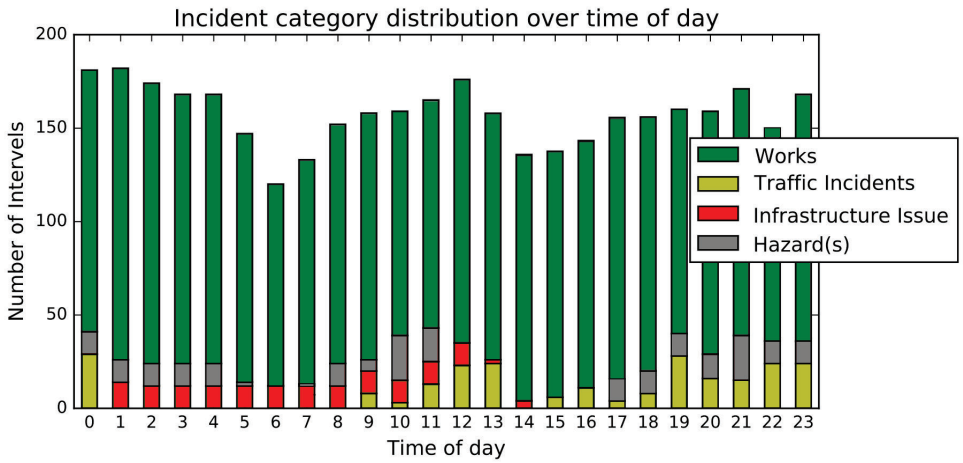


Figure 4.24: Distribution of different types of traffic anomaly on Euston Road according to the time of day

4.4.4 Results and Analysis

With the translated inputs, output and proposed detection model, this section will evaluate the model in terms of the following metrics:

- detection rate;
- false positive rate;
- F1 score;
- precision;
- area under the curve;
- confusion matrix;
- mean computational time to detect a NRC; and
- ROC curve.

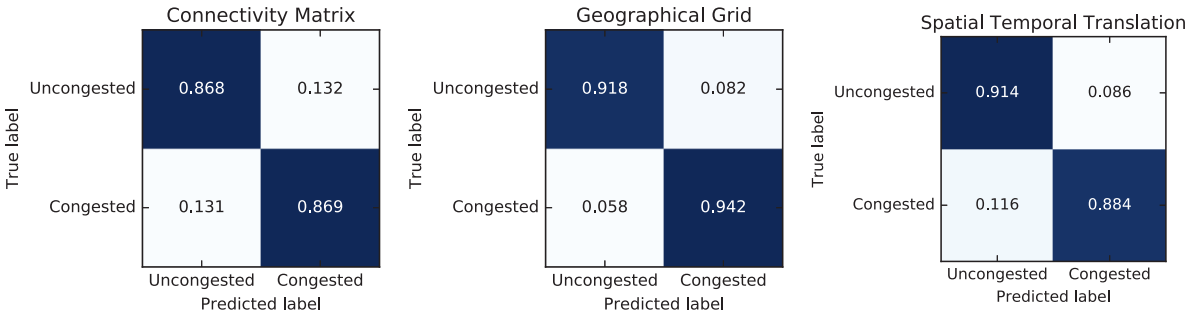
Compared to the early recurrent congestion detection discussed in Section 4.3, the sensitivity analysis with the change of prediction horizon is not applicable since traffic incidents cannot be predicted in advance due to their inherent randomness. The sensitivity analysis in respect to

time windows has been included in this section, however, in order to assess the impact of time lags on the model performance.

Table 4.6 shows the detection results based on different translation layers. Even though the connectivity matrix gave superior performance for recurrent congestion detection, its superiority is not transferred to a larger network like that of the London case study. This reduced performance may because of the sparsity of the translated matrix, as mentioned above. This sparsity may lead to insufficient feature extraction from the input layers. On the other hand, based on the results for all the metrics in the table, it is evident that the geographical translation method, which converts the traffic features in the format of inflow direction and outflow direction, is effective at supporting the proposed algorithm to detect the emerging and existing traffic anomalies. It outperforms the connectivity matrix and spatial temporal translation, with low false alarm rate and high precision, detection rate and F1 score. This result is confirmed by the confusion matrix in Figure 4.25. Figure 4.26 also demonstrates the superiority of the geographical grid method, as illustrated by its closeness to the left upper corner and the steepness of the curve in the ROC curve.

Table 4.6: Nonrecurrent congestion detection results based on different translation layers

Translation Layer	DR	FPR	F1 score	Precision	AUC	MTTD(s)
Connectivity Matrix	0.869	0.132	0.869	0.868	0.947	95.358
Geographical Grid	0.942	0.082	0.931	0.920	0.980	4.371
Spatial Temporal Translation	0.884	0.086	0.897	0.911	0.962	11.764



(a) Connectivity Matrix (b) Geographical Grid (c) Spatial Temporal Translation

Figure 4.25: Non-recurrent congestion detection confusion matrix based on translation layers

Regarding the sensitivity analysis in respect to time lags, as shown in Figure 4.27, it is remarkable that geographical grid translation consistently outperformed the other two methods

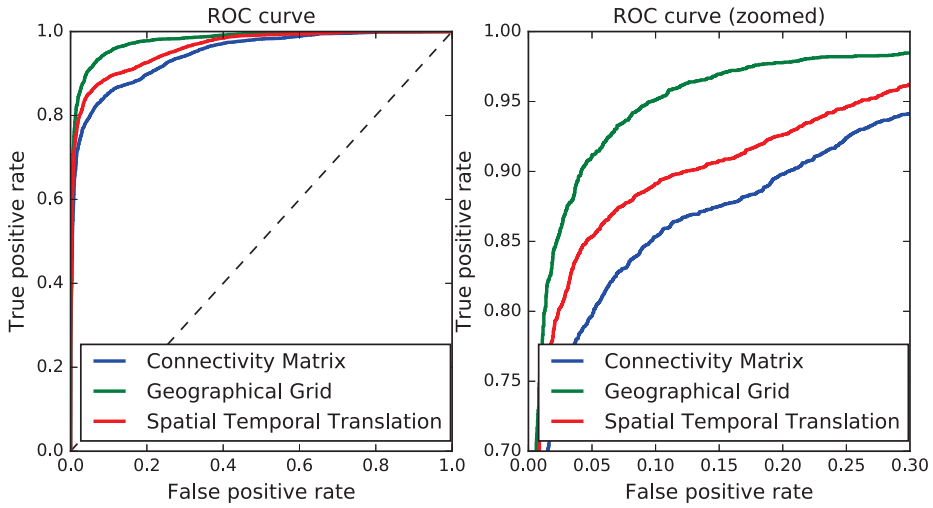
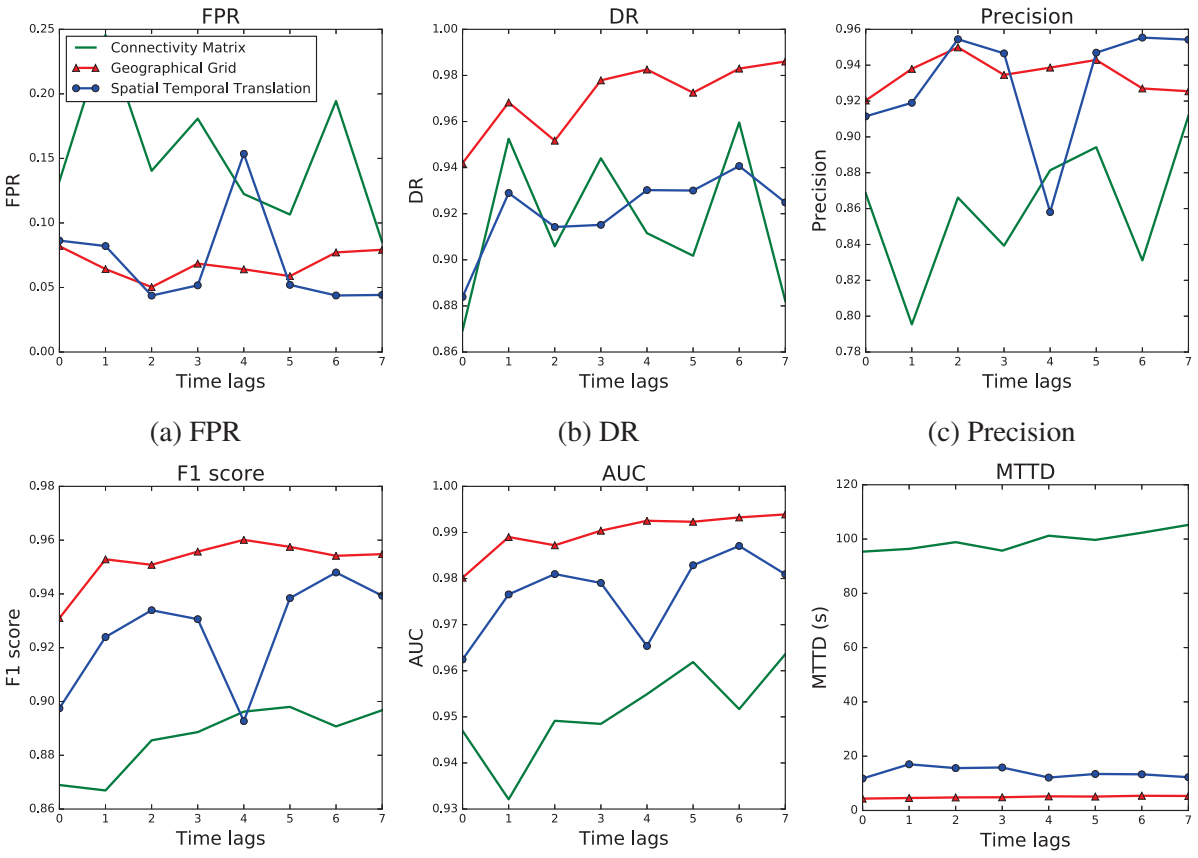


Figure 4.26: Non-recurrent congestion detection ROC curve

no matter the number of time lags, with FPR fluctuating below 0.1 and with DR over 0.94, etc. Overall, grid translation is less sensitive to the number of time lags. The performance of detection based on the connectivity matrix, however, is rather unstable due to its sparsity while that of spatio-temporal translation is acceptable but consistently less good than the performance of the geographical grid method. Considering the impact of time lags on accuracy, since many metrics that measure different aspects of the algorithm have been involved in the decision making, it is hard to find a conclusive recommendation. Nevertheless, it is notable that a time window of three could lead to a relatively lower FPR and higher precision together with other acceptable performance.

4.4.5 Summary of Non-Recurrent Congestion Detection

This section evaluated three translation layers for NRC detection based on a case study in London. The result suggested that geographical translation, which converts the traffic features in the format of inflow direction and outflow direction, is effective at supporting the proposed algorithm to detect NRC anomalies. It outperforms the connectivity matrix and spatial temporal translation across multiple evaluation metrics. The geographical grid might therefore be selected as the first component for an NRC detection methodological framework.



(a) FPR (b) DR (c) Precision

(d) F1 score (e) AUC (f) MTTD

Figure 4.27: Non-recurrent congestion detection in terms of time lags with prediction horizon=0

4.5 Summary

This chapter has been structured to address an important aspect of the third objective, i.e., to explore the translation of deep learning techniques to traffic anomaly detection by investigating three different translation layers. The chapter first introduced the background in respect to translation layers, comprehensively reviewing previous relevant studies in Section 4.1. After this literature review, three translation layers, connectivity matrix, geographical grid translation and spatio-temporal translation were described in Section 4.2 before evaluating these in respect to the problems of the early prediction of recurrent congestion and detecting non-recurrent congestion detection. Specifically, Section 4.3 presented the evaluation result based on a case study for the prediction of recurrent congestion in the City of Bath, while Section 4.4 illustrated the evaluation in respect to the detection of non-recurrent congestion based on a case study in London. For both case studies, the evaluation started from the data description where the reasons for the case study selection and data analysis were presented. This was followed by model calibra-

tion and output description, and finally, the evaluation results based on comprehensive metrics and sensitivity analysis.

Since several evaluation metrics assessing different aspects of the detection algorithms are investigated, it is hard to find a conclusive recommendation covering all translation layers in terms of all metrics. This very inconsistency, however, increases the flexibility of the potential practical applications, which may target different criteria. As for the recurrent congestion detection, the connectivity matrix approach would be recommended for the early prediction task if a low false alarm rate and high precision are required, while the geographical grid approach may be an attractive option where a high detection rate and F1 score are relevant. Another issue is that the prediction result will decrease when the prediction horizon increases but the accuracy for two-hour ahead prediction is still acceptable, with a precision of 0.96 and an FPR of 0.15. On the other hand, the existence of a bowl shape when evaluating the performance with increasing time lag indicated that adding unnecessary additional time lags may introduce noise rather than useful features into the detection algorithm. Recurrent congestion detection methods need to emphasise the impact of time lags. A wiser choice of time lags for recurrent congestion detection may lead to effective and accurate detection.

In terms of the detection of non-recurrent congestion, in contrast to recurrent congestion detection, where the output labels are not available, additional traffic incidents obtained from Transport for London have been used for incident detection. Another difference with the recurrent congestion detection is that the traffic network selected is much larger than that of the Bath case study, with around ten-fold more detectors more than that of the Bath case study. This larger network caused a relatively sparse connectivity matrix as the input for the detection model. As a result, the performance of the detection using the connectivity matrix input was worse than that for the geographical grid translation. With its advantages involving the consideration of direction and sufficient feature extraction, the geographical grid translation outperformed the others based on all metrics, regardless of the number of time lags in the sensitivity analysis. In addition, for the detection performance with geographical grid translation, a rather flattened bowl shape exists in the sensitivity analysis of time lags.

According to this overall performance comparison, connectivity matrix translation leads to superior performance in terms of low FPR, high precision and ROC for RC detection, while

geographical translation outperformed the others across all metrics for NRC detection. The different recommendations for these two tasks are reasonable since recurrent congestion entails the continuous correlation in terms of traffic stream direction which means recurrent congestion generally leads to traffic condition changes in the same direction. The connectivity matrix can reflect this direction correlation easily by connecting the neighbouring nodes by directions. On the other hand, NRC might impact on the traffic states of all directions in the area affected by traffic incidents. This impact on all traffic stream directions in a small area could be naturally represented by grid translation.

In summary, this chapter has dealt with the objective of translating the deep learning detection into traffic anomaly detection, as set out in Section 1.3. Based on the outcomes, the connectivity matrix translation and geographical translation will be selected as the translation layers for subsequent components of the RC and NRC traffic anomaly detection framework, respectively. To further validate the proposed detection algorithm itself, in the next chapter, the proposed deep learning will be compared with the conventional machine learning methods, both for early detection of recurrent congestion detection and detection of non-recurrent congestion.

Chapter 5

Deep Learning Based Anomaly Detection

In Chapter 4, three translation layers were developed to translate a traffic network with spatio-temporal features into a set of matrices. This chapter adapts the deep learning methods outlined in Section 3.2.2 and compares their performance with other conventional machine learning models, such as multilayer perceptron, random forest and gradient boosting classifier. Since few deep learning methods have previously been applied to the detection of traffic anomalies, a comprehensive review of deep learning for traffic prediction or forecasting is included here before undertaking the evaluation for both recurrent congestion and non-recurrent congestion detection.

This chapter is organised as follows. Section 5.1 presents the research background including a detailed literature review of the application of deep learning in traffic prediction. Section 5.2 presents the conceptual structure of the methodological framework and formulations of the anomaly detection problem for the two types of traffic anomalies. Section 5.3 gives the evaluation based on early recurrent congestion detection while Section 5.4 shows that of non-recurrent congestion detection in terms of DR, FPR, F1 score, precision, AUC, confusion matrix and ROC. Section 5.5 provides a summary of this chapter.

5.1 Introduction and Background

The previous chapter mainly focused on the translation input into deep learning methods. This section will focus on the application of the deep learning models for early RC and NRC detection.

To date, there are two main types of deep learning application in the domain of transportation could be summarised. Specifically, most studies have focused on the application of deep learning in the field of traffic prediction, such as traffic flow prediction (Huang et al., 2014; Polson and Sokolov, 2017; Du et al., 2019) and traffic speed prediction (Ma et al., 2015a; Kim et al., 2018; Ke et al., 2019). In addition, a few deep learning methods have been used for classification purposes, such as transportation mode classification (Dabiri and Heaslip, 2018) and crash detection or incident detection (Bao et al., 2019). Among these studies for the prediction, some used CNN (Polson and Sokolov, 2017) while the other applied hybrid methods based on multiple deep learning methods (Du et al., 2019).

For example, CNN and LSTM have been intensively used in the prediction and estimation problem to extract spatial and temporal features of traffic dynamics (Shi and Yeung, 2018). The hybrid model combining CNN and LSTM has been proved in many previous studies to have superior performance for traffic prediction (Ma et al., 2017; Yao et al., 2018; Wu et al., 2018c). The popularity of combining CNN and LSTM originates from its efficiency and effectiveness in extracting spatial and temporal features (Ma et al., 2017). Specifically, CNN is good at extracting spatial information because of its function of convolutional layers and pooling layers. Convolutional layers of a CNN connected locally instead of being fully connected like the traditional neural networks enable a CNN to deal efficiently with spatially-correlated problems (Krizhevsky et al., 2012). The pooling layers of CNN can significantly reduce the level of parameters in the model architecture which enable it to be generalised into large-scale problems (Karpathy et al., 2014). On the other hand, LSTM has the mechanism of gates which can control a long time-period memory which in turn can be particularly capable of dealing with temporal information and thus useful for the forecasting and prediction problem (Ma et al., 2017).

Another application of deep learning in transportation is to use reinforcement learning (Sutton and Barto, 2018) for traffic signal control. The research into traffic signal control involves a wide range of control theories, optimisation and decision making, however, which makes it a

very different topic compared to the traffic prediction and classification problem, and will not be reviewed in this research. Readers who are interested in this aspect may refer to the recent survey of traffic control methods by Wei et al. (2019). A summary of the application of deep learning in traffic prediction is shown in Table 5.1.

Table 5.1: Summary of deep learning applications in traffic prediction and detection

Reference	Deep Learning	Research Problem	Baseline Models
Huang et al., 2014	Deep Belief Networks	Traffic flow prediction	MLP, SVM, ARIMA, etc.
Lv et al., 2014	Stacked Autoencoders	Traffic flow prediction	MLP, SVM, Random Walk, etc.
Ma et al., 2015a	LSTM	Traffic speed prediction	MLP, SVM, ARIMA, etc.
Wang et al., 2016a	Recurrent CNN	Traffic speed prediction	CNN, SVM, ARIMA, etc.
Polson and Sokolov, 2017	CNN	Traffic flow prediction	linear model, vector auto-regression
Ma et al., 2017	CNN	Traffic speed prediction	RF, MLP, KNN, etc.
Fang et al., 2017	Deep Neural Networks (DNN)	Transportation modes prediction	SVM, KNN, etc.
Kim et al., 2018	Capsule Network	Traffic speed prediction	CNN
Wu et al., 2018c	DNN	Traffic flow prediction	LASSO, MLP, etc.
Dabiri and Heaslip, 2018	CNN	Traffic mode prediction	KNN, MLP, SVM, etc.
Zhang et al., 2018a	ST-ResNet	Traffic flow prediction	ARIMA, MLP, LSTM
Zhang et al., 2018b	LSTM and Deep Belief Network (DBN)	Traffic incidents detection	SVM and ANN
Ke et al., 2019	CNN	Traffic speed forecasting	MLP, ARIMA, LSTM, etc.
Du et al., 2019	Convolutional LSTM	Traffic flow prediction	ARIMA, ANN, etc.
Bao et al., 2019	Convolutional LSTM	Traffic crash risk prediction	ANN, ARIMA, GBC, etc.

Even though CNN and LSTM have been intensively used in traffic prediction, to date, limited research has investigated its effectiveness in traffic anomaly detection. Both types of traffic anomaly, i.e., the detection of recurrent and non-recurrent congestion, exhibit spatial dependencies on the traffic states in the local traffic network and temporal dependencies on the evolution or propagation of temporal traffic dynamics. Hence, it is vital to formulate the problem with the factors that can help extract these dependencies. To fill this gap, it is worth investigating

the application of CNN and LSTM in the traffic anomaly detection problem.

5.2 Methodology

Given the challenges and gaps mentioned above, this section will formulate the problem and transfer the deep learning techniques, i.e., CNN and LSTM, to the research field of traffic anomaly detection.

5.2.1 Recurrent Congestion Detection

As mentioned in Section 3.2.2, to overcome the aforementioned disadvantages of traditional traffic congestion detection, a two-stage detection method, i.e., (1) a traffic state identification stage and (2) an early RC detection stage, is proposed to detect suspicious RC early based on traffic flow and occupancy data observed from loop detectors. The primary objectives of this two-stage model are first to identify different traffic states (i.e., congestion and non-congestion) and secondly to model spatio-temporal dependencies so as to be able to detect potential recurrent traffic congestion early. This early detection is especially desirable for proactive traffic operation and control for real-time ITS systems. The labels of the recurrent congestion detection in this section are the same ones as those generated by the EM algorithm. Please refer to Section 4.3.1 for details.

A critical issue to be tackled with current applications of deep learning to the transportation field is determining an appropriate way to translate or represent traditional traffic data as a tensor, the essential input for deep neural nets. The transformation between the natural formats of traffic data and image data is not straightforward. Given the essential understanding of early RC detection problem and the evaluation results from the last chapter, it was determined that the connectivity matrix gives superior accuracy compared to other translation layers, based on the metrics such as DR and FPR. Thus, this chapter will carry out the evaluation based on the recommendation of the previous chapter and compare it with other baseline models or conventional machine learning methods. More specifically, for the connectivity matrix, the $x - axis$ represents the origin nodes of the network, the $y - axis$ represents the destination nodes, which are placed according to the direction of traffic flow, while the $z - axis$ stands for the features

of traffic information, i.e., traffic flow and occupancy. An additional dimension represents the number of frames which are time lags or windows (i.e., how many time steps, $t-1$, $t-2$, etc., have been taken as inputs). After the translation, each frame corresponds to a 4D matrix with origin nodes, destination nodes, features and time frame.

In terms of the detection method, the hybrid method combining CNN and LSTM is proposed for the early RC detection stage where inputs have been transferred into the matrix using the connectivity matrix, and outputs are binary labels generated from the traffic state identification stage in Section 4.3.1. The early detection model consists of a CNN and a multi-layer LSTM as shown in Figure 3.9. The CNN is initially used for general classification but it cannot handle the prediction of upcoming traffic states. LSTM, however, enables CNN to detect traffic congestion with partial data and predict future states because of its capability of sequence processing. In this stage, CNN reads the input as an image and obtains a fixed size vector representation of the initial input. After that, a multi-layer LSTM takes the representation, original input and the output from previous timestamps to produce the desired output, i.e., traffic states in a certain timestamp. An overview of the CNN-LSTM was previously provided in Figure 3.9.

Generally, every layer in a CNN model serves as a detection filter for features presented in the input data. The first layer recognises the relatively obvious features, the later layers gradually detect the more abstract features, while the last layer of a CNN makes a specific classification based on all features detected by previous layers. As aforementioned in Section 4.3.4, the selection of max pooling kernel size (2×2) and convolutional filter size (3×3) could be referred to several well-known CNN architectures, such as AlexNet (Krizhevsky et al., 2012) and LeNet (LeCun et al., 1998). On the other hand, the depth of the proposed model is constrained by the input size of the image and thus the final net contains five layers with weights.

The detailed CNN-LSTM architecture used in this research is shown in Table 5.2. The first two are Convolutional (Conv) layers, each followed by a max pooling, and the other two are LSTM layers, while the remaining fully connected layer is fed to a Sigmoid activation function to output the predicted probability of the binary classes. The Conv kernels are connected to all kernels in the previous layer and ReLU activation function is used for the output of every Conv layer and the first LSTM layer.

Table 5.2: LSTM CNN architecture for early RC detection

Layer (type)	Output Shape	Param #
Time Distributed Conv	(None, 1L, 32, 18L, 18L)	608
ReLU Activation	(None, 1L, 32, 18L, 18L)	0
Time Distributed Pooling	(None, 1L, 32, 9L, 9L)	0
Dropout	(None, 1L, 32, 9L, 9L)	0
Time Distributed Conv	(None, 1L, 64, 7L, 7L)	18496
ReLU Activation	(None, 1L, 64, 7L, 7L)	0
Time Distributed Pooling	(None, 1L, 64, 3L, 3L)	0
Dropout	(None, 1L, 64, 3L, 3L)	0
Time Distributed Flatten	(None, 1L, 576)	0
Bidirectional LSTM	(None, 1L, 64)	155904
Dropout	(None, 1L, 64)	0
Bidirectional LSTM	(None, 128)	66048
Dropout	(None, 128)	0
Dense	(None, 1)	129
Sigmoid Activation	(None, 1)	0

5.2.2 Non-recurrent Congestion Detection

A CNN-based incident detection algorithm was proposed to detect anomalous conditions by comparing traffic flow values with the historical traffic pattern. In order to implement the CNN, the time-series traffic flow data from ILDs must be converted into a matrix which shares the same pattern as that used to characterise general 2D images. From the discussion and evaluation of Chapter 4, it was evident that geographical grid translation outperformed the other methods in terms of almost all the metrics. Thus, in this section, a detection algorithm based on the geographical grid translation will be presented. Before presenting the results in comparison with other machine learning methods, however, it is necessary to elaborate on the typical functions or layers of a CNN.

5.2.2.1 Convolutional Layer

The convolutional layer is the most important layer in a CNN model (Krizhevsky et al., 2012) connecting the input connectivity matrix defined as $x \in \mathbb{R}^{p \times q \times q}$ with a set of filters $W \in \mathbb{R}^{n \times p \times m \times m}$ where p is the sample size, q is the size of the input matrix and n is the number of

convolutional filters. The single convolutional layer is formulated as follows:

$$\begin{aligned} f(x\colon W, b) \;&=\; h \;=\; h_k, k = 1, \ldots, n \\ h_k &= ReLU(x \otimes W_k + b_k) \end{aligned} \tag{5.1}$$

where $b \in \mathbb{R}^n$ is a bias for each filter output, k is the number of convolutional filters and $\otimes$ is the convolutional operator that applies on a single input and a single filter. The output $h \in \mathbb{R}^{n\times(q-m+1)\times(q-m+1)}$ is a set of feature maps extracted by the convolutional layer. A parameter sharing scheme is used in convolutional layers to control the number of parameters.

5.2.2.2 Max Pooling Layer

Pooling layers in the CNN model summarise the outputs of neighbouring groups of neurons in the same kernel map. A max pooling technique, the most common pooling strategy, is applied in every two-unit window for each sub-region to output the maximum. An important function of pooling layers is progressively to reduce the size of representation by half and filter out undesirable small values of traffic flow data, thus reducing overfitting. The commonly used operation is to select the maximum value over the feature region generated by convolutional layers, since this max process ensures that the significant features can be obtained for detection, even with varying levels of translations (Nogueira et al., 2017).

5.2.2.3 Fully Connected Layer and Optimisation

The fully-connected layer is generally employed in the last stage of hidden layers to control the dimension of the final output. The fully connected layer has full connections to all neurons in the previous layer. The activation in this layer firstly computes with a matrix multiplication followed by a bias offset. Then the output will be transformed according to the specified activation function.

A binary Sigmoid crossentropy loss function is used as an objective function to be minimised later. The loss function for a binary classification is given in Eq. 5.2.

$$\mathcal{L} = -\sum_i \hat{y}_i \log(y_i) - (1 - \hat{y}_i \log(1 - y_i)) \tag{5.2}$$

RMSprop (Tieleman and Hinton, 2012) serves as an optimiser to utilise the magnitude of recent gradients in order to minimise the loss function. It is used to keep a moving average of the squared gradient for each weight, and update the weight and bias in each iteration during the optimisation. Given the decay rate γ and the learning rate η, the parameters are updated as Eq. 4.3. After several rounds of convolutional and pooling operations, fully-connected layers make use of most parameters in order to learn all neurons in the previous layer and output to the current layer, where the spatial notion of the matrix is reduced to that of a one-dimensional vector. To prevent overfitting caused by parameter-dominated fully connected layers, the dropout (Srivastava et al., 2014) approach is employed. This basically drops a couple of neuron outputs randomly on the basis that decreasing the number of neurons improves the speed of training and makes the model practically effective. Finally, a classification layer is employed to calculate the class probability for each instance. The detailed architecture of the non-recurrent congestion detection model is appended in Table 5.3 and Figure 5.1.

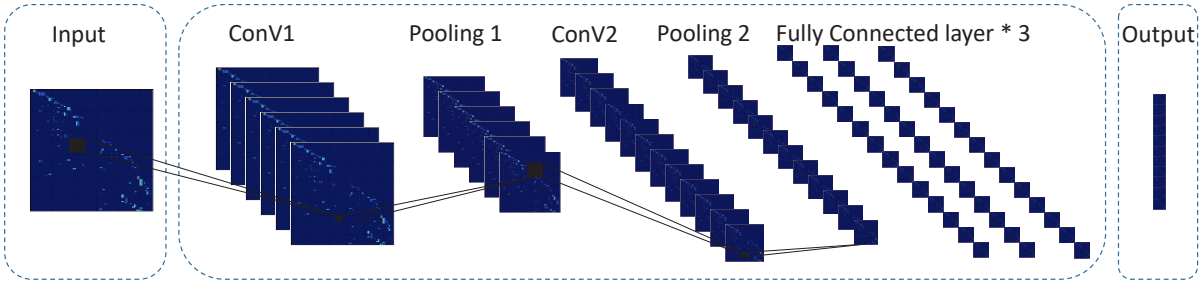


Figure 5.1: CNN architecture for NRC detection

In summary, the final net contains five layers with weights; the first two are Conv layers, each followed by a max pooling, and the other three are fully connected layers, while the final layer is fed to a Sigmoid activation function to output the predicted probability of the binary classes. The ReLU activation function is used for the output of every convolutional layer while the Sigmoid function has been used in the final layer.

5.2.3 Conventional Machine Learning Methods for Comparisons

In order to evaluate the performance of the proposed model, three commonly used link-based machine learning-based detection algorithms are used for comparison. They are Multi-layer Perceptron (MLP), Random Forest (RF) and Gradient Boosted Classifier (GBC). These methods

Table 5.3: CNN architecture for NRC detection

Layer (type)	Output Shape	Param #
Conv2D	(None, 32, 22, 18)	1184
ReLU Activation	(None, 32, 22, 18)	0
MaxPooling	(None, 32, 11, 9)	0
Dropout	(None, 32, 11, 9)	0
Conv2D	(None, 32, 9, 7)	9248
ReLU Activation	(None, 32, 9, 7)	0
MaxPooling	(None, 32, 4, 3)	0
Dropout	(None, 32, 4, 3)	0
Flatten	(None, 384)	0
Dense	(None, 64)	24640
ReLU Activation	(None, 64)	0
Dropout	(None, 64)	0
Dense	(None, 32)	2080
ReLU Activation	(None, 32)	0
Dropout	(None, 32)	0
Dense	(None, 1)	33
Sigmoid Activation	(None, 1)	0

are able to capture the complex and non-linear relationship between different features (Liu et al., 2013; Cortez et al., 2012).

Stochastic gradient methods are used to minimise the loss function and update the weights and bias step-by-step. After the optimisation, these methods with optimised weights are compared with the proposed CNN model. As a result, MLP is set up with hidden layers with a size of (32, 5); RF is configured to generate ten decision trees with a depth of 3; and GBC is tuned to form up to 200 estimators, at a learning rate of 0.05 and a depth of 10.

As for the input for these machine learning methods, the time series traffic data can be directly inputted into the models without the connectivity matrix transformation. The input data of these traditional methods are normalised time-series traffic data of the target main corridor or segmentation. Furthermore, stochastic gradient methods are also used in order to minimise the error and optimise weights and bias step-by-step. The RF, MLP and GBC methods have to be applied separately to detect incidents on each link.

5.2.4 Problem of Overfitting

Overfitting is a statistical phenomenon in which a model is exactly in line with or too close to a particular set of data and may therefore lose the ability to fit additional data or predict future instances reliably, especially when the model includes more terms than are necessary or uses more complicated approaches than are necessary (Hawkins, 2004). Ideally, the model should be able to describe the underlying patterns, capturing the true pattern but not the noise or outliers. With training according to a specific accuracy metric, however, the model may incorporate the noise as it may only fit a particular dataset but not generalise the other samples well enough. Overfitting is undesirable for a wide range of reasons, since 1) it may add no useful functions and hence waste resources; 2) it may possibly add irrelevant factors which could lead to worse decisions because the coefficients fitted to them add random variation to subsequent decisions (Hawkins, 2004).

Overfitting is a common problem in the machine learning and deep learning methods (Krizhevsky et al., 2012), and its presence can be suspected when the model accuracy in the training dataset is high but drops significantly with a new dataset. A simple way to detect overfitting in practice, therefore, is cross-validation (Kohavi, 1995) by examining the trained model with a validation dataset to check its predictive accuracy and to identify if significant variance exists. In this research, to avoid the problem of overfitting, the proposed model has been validated with the plots of accuracy metrics from a training dataset (70%) and a validation dataset (10%) before testing in the test dataset (20%).

In addition to the cross validation, a dropout technique (Srivastava et al., 2014) has been used for the deep learning techniques. This consists of setting the output of each hidden neuron to zero with a fixed probability (Hinton et al., 2012b). The dropped-out neurons therefore do not participate in the optimisation. Consequently, the neural network sub-samples a different architecture, and all these architectures share weights (Krizhevsky et al., 2012). Since a neuron cannot depend on the presence of specific other neurons, dropout reduces complex co-adaptations of neurons and hence forces the neuron to learn more robust features.

5.2.5 Summary

In this section, the detailed functions and parameters of the deep learning methods proposed in Section 3.2.2 have been profiled and transferred into traffic anomaly detection. The configuration of RC and NRC detection methods is based on many previous studies. Three representative machine learning methods have been selected as benchmarks against which the proposed detection methods can be compared. To avoid the overfitting problem, the dropout technique and cross-validation have been applied.

5.3 Results of Case Study of Early Recurrent Congestion Detection

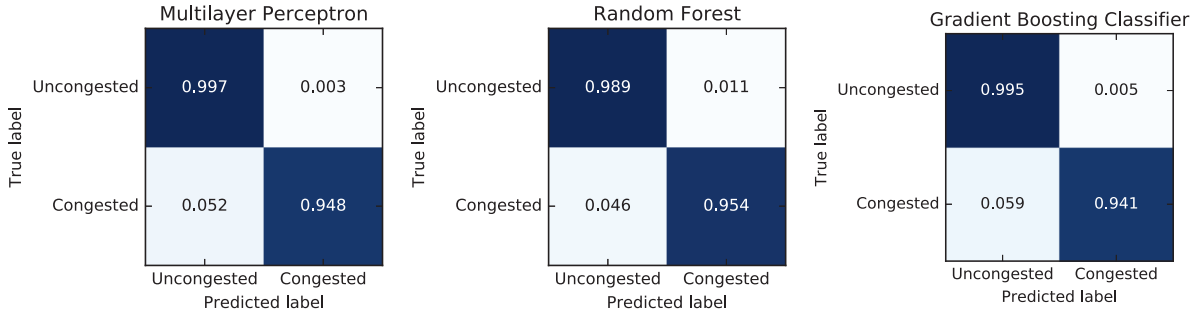
Table 5.4 shows the result of early recurrent congestion detection compared to conventional machine learning methods. Notwithstanding the generally good performance of both the proposed CNN-LSTM model and the alternatives in the classification of traffic congestion, the CNN-LSTM performed better in terms of low FPR. This means that CNN-LSTM has the potential to detect outliers more accurately than alternative methods. While a large number of parameters and tensor extraction processes are essential to configure CNN-LSTM, this method provides better detection accuracy than the benchmarks.

Table 5.4: Recurrent congestion detection results based on different methods

Method	DR	FPR	F1 score	Precision	AUC
LSTM-CNN	0.972	0.002	0.985	0.997	0.9993
Multilayer Perceptron	0.948	0.003	0.971	0.995	0.9994
Random Forest	0.954	0.011	0.969	0.984	0.9933
Gradient Boosting Classifier	0.941	0.005	0.966	0.992	0.9951

Of the four methods, CNN-LSTM performs best in terms of FPR and DR metrics. The CNN-LSTM method is computationally more intensive and takes longer to run, but, still, the time taken for CNN-LSTM method to run on a regular desktop PC is less than a second. Given that temporal granularity of traffic data provided by ITS systems is typically 1-minute or greater, the increased computational complexity is of no practical concern when implemented within a real-world ITS system. Figure 5.2 confirms the superior performance of the proposed methods

with the confusion matrix while Figure 5.3 gives an overview of the comparison based on the ROC curve where the proposed method has been placed closer to the left upper corner.



(a) Multilayer Perceptron (b) Random Forest (c) Gradient Boosting Classifier

Figure 5.2: Recurrent congestion detection confusion matrix based on different methods

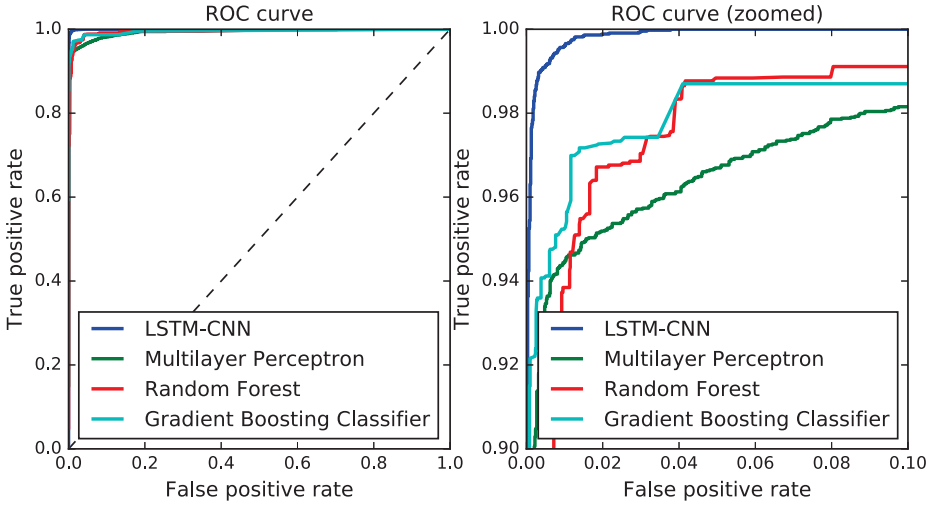
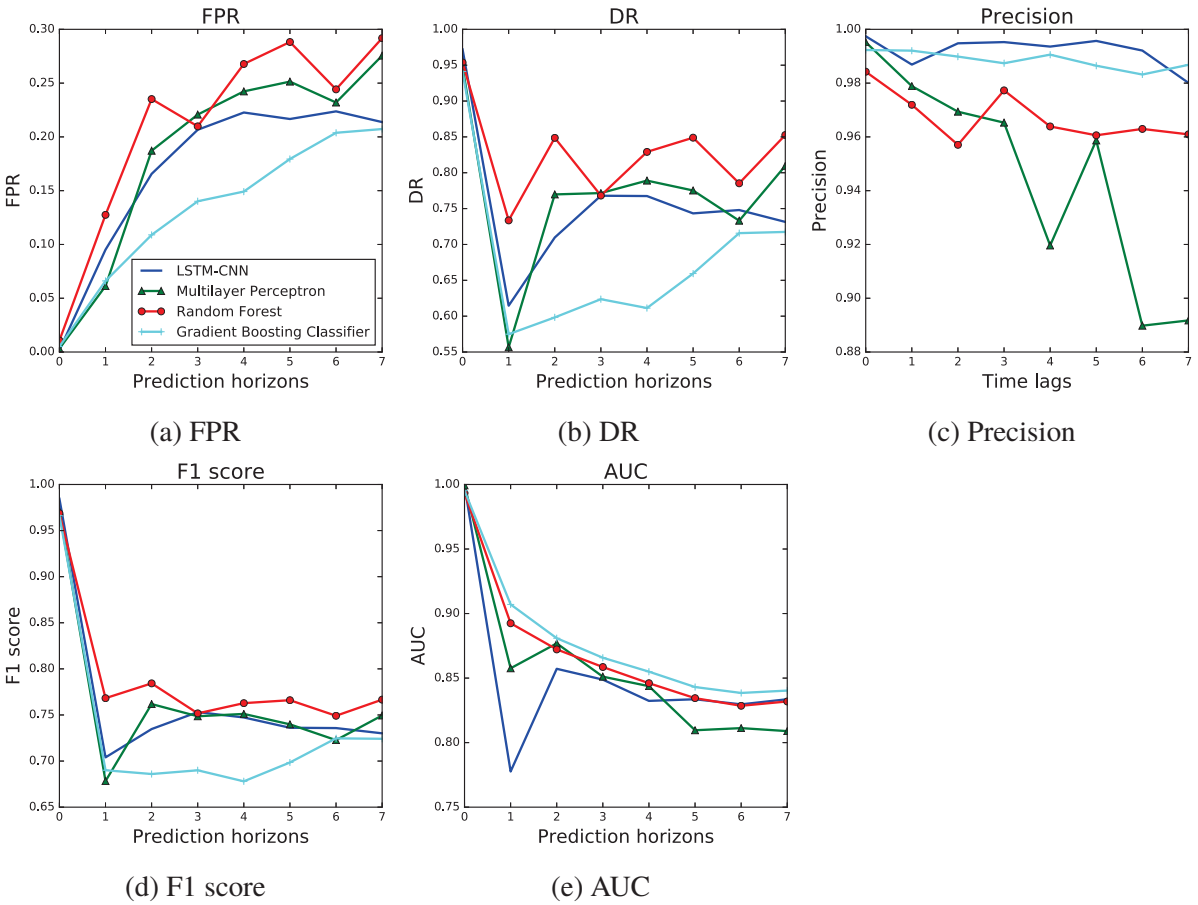


Figure 5.3: Recurrent congestion detection methods ROC curve

Figure 5.4 shows the relationship between RC detection accuracy and the number of time steps ahead when the prediction method is run. It is evident from the figure that the detection rate will decrease as the number of time steps increases. Even though the accuracy of the proposed early detection method varies with respect to the size of the time horizon, however, the method is able to predict RC accurately as early as seven-time steps (around two hours) in advance. This result is encouraging since it provides traffic managers around two hours to select a suitable intervention action and implement it.

Figure 5.5 presents the variation of accuracy in terms of the time windows to evaluate the performance of early detection in terms of the time lags used as input for the CNN-LSTM



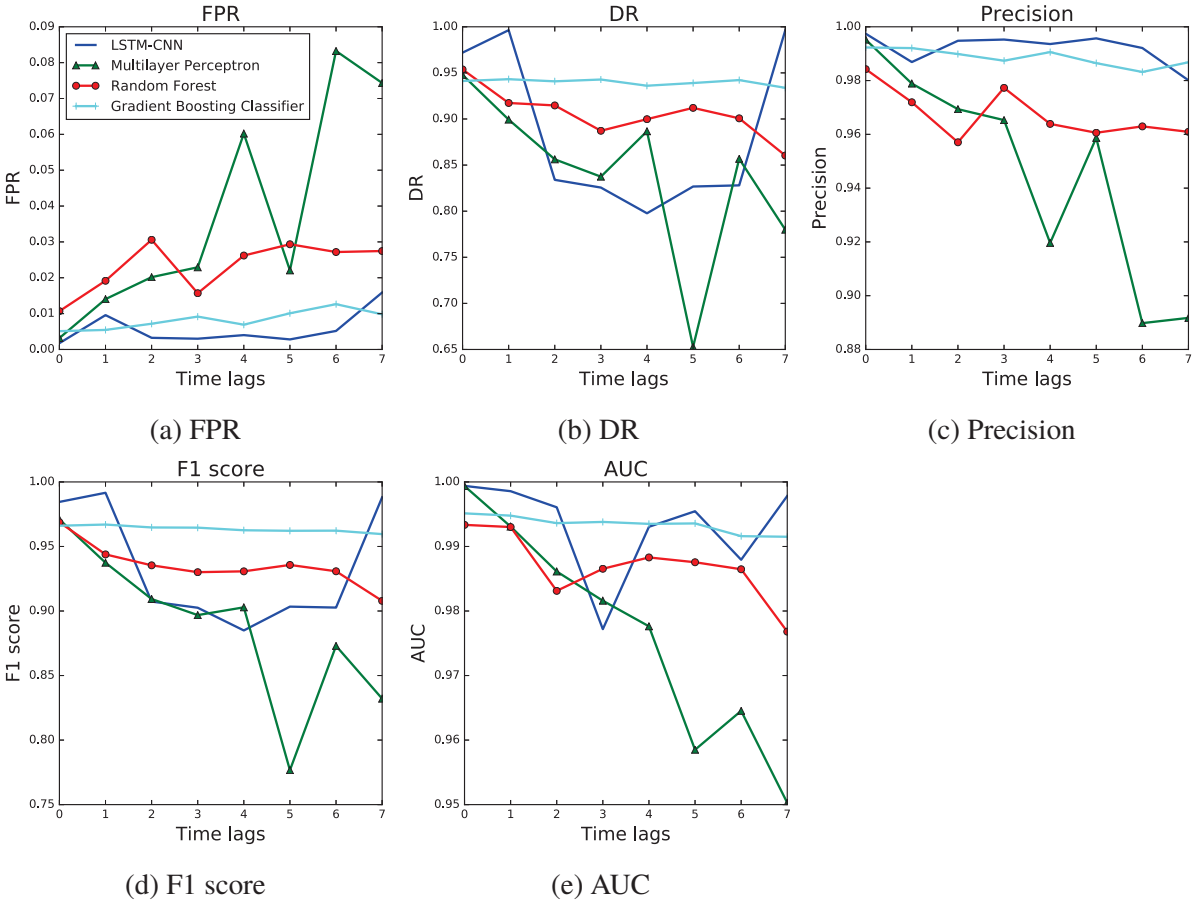
(a) FPR (b) DR (c) Precision

(d) F1 score (e) AUC

Figure 5.4: Recurrent congestion detection methods in terms of prediction horizons with time lag=0

model. Based on the six metrics tested, the sensitivity of the four deep learning methods varied significantly. Given the sensitivity analysis, it is obvious that the lowest FPR and highest precision could be obtained with a time window size of 2 or 3. On the other hand, the DR and F1 scores have a bowl shape which indicates that adding more historical information may not help with these two evaluation indexes. MLP and RF are insensitive to the number of time lags, while GBC is most immune to model structure with regards to time lags. The reasons of the poor performance these baselines compared to the new proposed method are mainly from two points: (1) lack of ability to model temporal correlations and learn from information from previous time steps; (2) limited benefit from massive amounts of data. For example, random forest can train a model with a relative small number of samples and get pretty good results but it will, however, quickly reach a point where more samples will not improve the accuracy. In contrast, the proposed deep learning method needs more data to deliver the same level of accuracy, but it will benefit from massive amounts of data, and continuously improve the accuracy. The full re-

sults based on eight time lags and eight prediction horizons have been included in the Appendix C.2.



(a) FPR (b) DR (c) Precision

(d) F1 score (e) AUC

Figure 5.5: Recurrent congestion detection methods in terms of time lags with prediction horizon=0

In this section, an early detection model based on the combination of unsupervised and supervised learning is presented for the early detection of recurrent traffic congestion. There are essentially two sub-problems in RC detection. The first is generating the reference labels as indicators for congested or non-congested states for RC detection and the second is early detection of RC congestion, i.e., providing an early alarm. In order to solve the first problem, a data-driven Expectation Maximisation algorithm is presented. It was assumed that traffic states could be categorised into congested and non-congested regimes. The proposed method clusters different traffic states iteratively to satisfy the maximum expectation effectively based on traffic flow and occupancy data from ILD sensors. In order to solve the second sub-problem, the spatio-temporal characteristics of urban traffic dynamics were learned and analysed in a CNN-LSTM based early detection model, in which the CNN element guarantees its capability of capturing

the spatial correlations among ILD data while the LSTM element ensures that long temporal lags are handled efficiently. The proposed model can also be generalised for prediction problems in road networks, where spatial correlations are significant factors for the prediction. The time-space relationship is captured during the feature extraction process in the deep learning neural layers of the proposed method.

Comprehensive evaluation metrics, including the Detection Rate, False Positive Rate, Precision and Mean Time to Detection, are used to evaluate the performance of the proposed method compared to more conventional machine learning methods. Based on the Bath case study, the proposed two-stage early detection approach has been proved to be promising, as it is able to detect recurrent congestion as early as seven time steps (i.e., around two hours ahead) in advance providing traffic managers with adequate time to implement suitable congestion mitigation actions. Comparison of the CNN-LSTM method with established methods from the literature indicates that the deep neural network methods such as the CNN-LSTM algorithm are capable of achieving accurate early congestion detection and outperforming normal multi-layer neural networks.

Additionally, this research was followed by an extension of the analysis to consider the implications learnt from this work in respect to the formation of congestion and signs of imminent congestion in order to generalise the conclusive results based on limited case studies. A sensitivity analysis that describes the robustness of performance in terms of the number of time lags and prediction horizons was conducted. This additional analysis shows that traffic data input decomposed by longer time lags does not necessarily improve the performance, even with an advanced sequence processing tool.

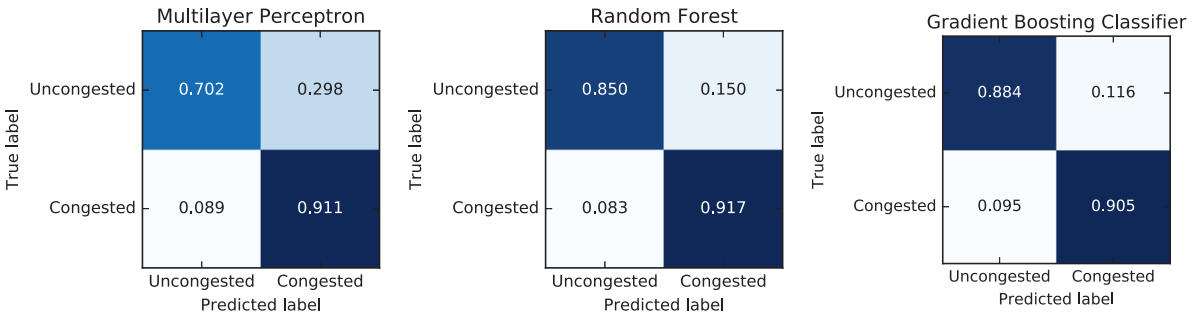
5.4 Results of Case Study of Non Recurrent Congestion Detection

This experiment is to examine the accuracy of the proposed CNN model for detection of non-recurrent or traffic incident related congestion. Table 5.5 shows the results for non-recurrent congestion detection compared with conventional machine learning methods. Notwithstanding the generally good performance of both the proposed CNN model and the alternatives in the

classification of traffic incidents, the CNN performed better in terms of low FPR and high F-measurements, which means that CNN has been potential for detecting traffic outliers with less mistaken flags (i.e., false positives and negatives). This superior performance is also verified by the confusion matrix in Figure 5.6 and the ROC curve in Figure 5.7. Despite the large number of parameters and the tensor extraction process that is essential to configure CNN, the method exhibits effective detection compared to the benchmark. It is reasonable that conventional machine learning methods have a shorter computational time compared to CNN due to the inherent constraints of the deep structure and a large number of parameters to be trained for the CNN. This computational time is still practically acceptable as it is far less than the aggregated time interval, i.e., 5 minutes.

Table 5.5: Recurrent congestion detection results based on different methods

Method	DR	FPR	F1 score	Precision	AUC	MTTD(s)
CNN	0.942	0.082	0.931	0.920	0.980	4.371
Multilayer Perceptron	0.911	0.298	0.825	0.754	0.902	0.227
Random Forest	0.917	0.150	0.888	0.860	0.963	0.078
Gradient Boosting Classifier	0.905	0.116	0.896	0.887	0.968	0.016



(a) Multilayer Perceptron (b) Random Forest (c) Gradient Boosting Classifier

Figure 5.6: Recurrent congestion detection confusion matrix based on different methods

Analysis of time lags is conducted in order to study their impact on the detection performance, and the result is presented in Figure 5.8. The conclusion is very similar to that of the recurrent congestion detection as the accuracy does not necessarily increase with the increase in time lags for all metrics. Even though it is hard to find a conclusive recommendation, however, one might identify that the proposed method has the lowest FPR and highest precision with a time lag equal to 2, and this indicates that the traffic information in the short past can help to introduce noise to the classification model. The inherent reason for this phenomenon is that the

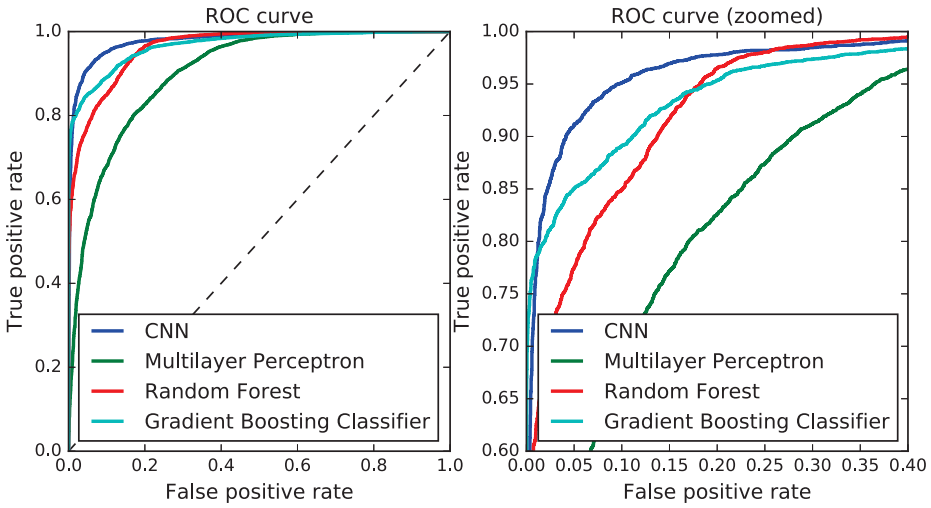
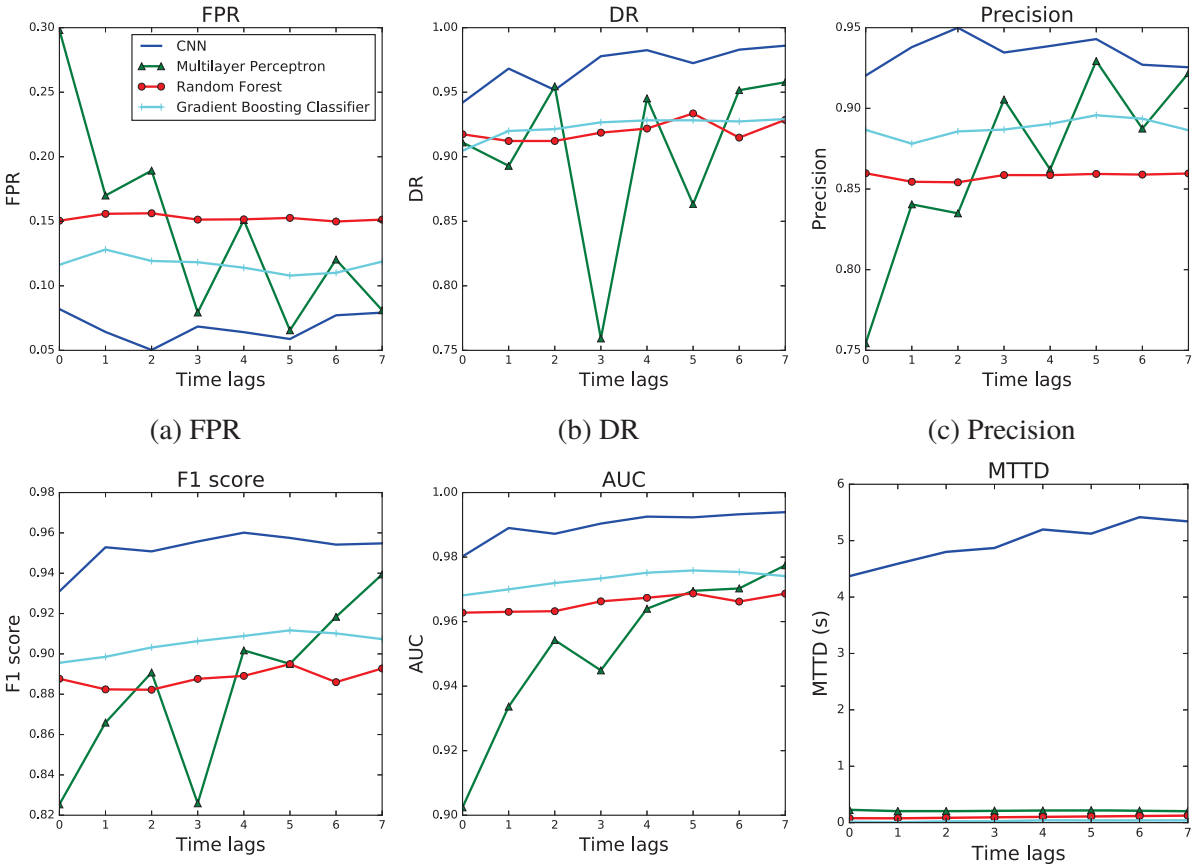


Figure 5.7: Recurrent congestion detection methods ROC curve

traffic states change rapidly in the centre of the city and generally traffic division is a common practice during the occurrence of traffic incidents in a short time.



(a) FPR (b) DR (c) Precision

(d) F1 score (e) AUC (f) MTTD

Figure 5.8: Non Recurrent congestion detection methods in terms of time lags with prediction horizon=0

In this section, a network-level incident detection model based on the CNN architecture is presented. The proposed model can be used for prediction problems in large urban networks where spatial correlation is a significant factor. Traffic data are extracted into tensors by using geographical grid translation to retain spatial correlations in the network before feeding them into the proposed model. In order to evaluate the performance of the new approach, it was applied to detect traffic incidents with traffic data and incident data collected from central London, and compared it with established alternative methods in terms of six metrics. The results indicate that the proposed deep-learning based detection model outperforms the baseline models. This comparison of the CNN method with the established methods from the literature indicates that deep neural network methods, such as the CNN algorithm, are capable of achieving accurate network-level traffic incident detection, especially for a large sized network.

5.5 Summary

This chapter has reviewed the application of deep learning techniques in transportation. The hybrid method LSTM-CNN has been used for the recurrent congestion detection in which the LSTM was applied to extract the temporal features while the CNN was used to capture the spatial features in the traffic network. The results of a case study from the City of Bath showed that the proposed model gave reasonable early prediction (with a DR value of 0.972, FPR value of 0.002, etc), outperforming the baseline models, i.e., MLP, RF and GBC.

On the other hand, the application of CNN in the context of non-recurrent congestion detection was also evaluated and compared with the baseline models. Results of a case study in London show that the CNN model is superior to the conventional machine learning methods to get information for existing and emerging traffic incidents.

Despite the superior detection results for both recurrent congestion detection and non-recurrent congestion detection, the models can only give prediction or detection results for target corridors or links. For a network-wide analysis, it is vital for any ITS applications to locate the bottlenecks precisely, i.e., recurrent congestion and incidents in the network. Building on the contribution of Chapter 4 and this chapter, therefore, the next chapter will illustrate a network-based detection model with the function of localisation by using deep learning methods.

Chapter 6

Localisation of Traffic Anomaly

The methodological framework presented in this thesis consists of three parts: (1) a translation layer to capture the spatial and temporal features; (2) detection methods in conjunction with the translation layer in order to detect traffic anomalies with reasonable accuracy compared to baseline models; and (3) localisation that can help to locate the traffic anomalies at a network level. Accurate detection of traffic anomalies based on the information at a network level is a very important step for a network-based model. Since traffic anomalies could occur on multiple road segments simultaneously, it is vital to decompose and locate traffic anomaly points precisely. This chapter will therefore focus on the question of localisation and evaluate the feasibility of locating traffic anomalies at a network level. The localisation function is of importance when the proposed method, or similar methods, are used at a network level.

Accurate measurement of traffic anomalies is a prerequisite to design effective interventions. The chapter will start with a review of studies relevant to the localisation of traffic anomalies, in Section 6.1, focusing on vision-based and sensor-based localisation and positioning. Section 6.2 formulates the methods for locating traffic anomalies at a network level. In this section, six different methods are presented. This is followed by an evaluation based on the detection of both recurrent and non-recurrent congestion, with multiple occurrences of RC and NRC at the network level in Section 6.3 and Section 6.4, respectively. Finally, Section 6.6 concludes this chapter with the main findings and limitations of the proposed localisation method.

6.1 Introduction and Background

Advances in ITS, exploiting various synergistic technologies such as collision warning, environmental monitoring and regulation enforcement, can help improve the safety and level of service of transportation networks. These applications of ITS, however, are largely dependent on the ability to accurately locate these threatens in respect to the transportation network (Amini et al., 2014).

A vast array of literature has attempted to solve the problem of localisation practically, not only in the field of transportation research (Amini et al., 2014), but also in a variety of other domains such as robotics (Rascon and Meza, 2017; Woodman and Harle, 2008), object detection in vision processing (Li et al., 2013) and, in medical research, cancer detection (Bratan et al., 2013). Localisation in these domains usually relies on advanced sensors to capture either the vision-based features or sensor-based features. Specifically, for visual object detection, the common features used are colour, shape, depth and edges (Luo et al., 2014). One typical solution to the robot localisation problem is particle filters because of their benefits in requiring significantly less computation and a smaller memory than the comparable methods (Fox et al., 1999). A particle filter is a sequential Monte Carlo method that employs a set of random weighted particles to represent the posterior distribution of the target state or feature (Candy, 2016). These features are relatively straightforward, with the input of photos, videos and signals, but for the matrix from transportation, it is hard to extract those features even with different transformed matrix due to the complexity and highly dynamic nature of the traffic network.

To date, limited studies have been conducted to localise the traffic anomalies from a network based on the traffic variables. With the recent advances, however, some studies have used deep learning methods to learn traffic networks as images (Ma et al., 2017; Wu et al., 2018c; Cui et al., 2018). For example, Ma et al. (2017) learned the traffic network as images to predict traffic speed. The study transformed the network into a spatio-temporal format with each node in the spatial sequence representing the location of sensors. By transferring the output from a value into a vector, the model is capable of short-term prediction for each node at a network level. This spatio-temporal translation has limitations, however, in respect to its poor transferability to larger networks with more complex topologies, as discussed in Section 4.1. Wu et al. (2018c) applied a spatio-temporal CNN LSTM to predict traffic flow where the inner outputs

of the deep learning models had been visualised. The authors concluded intuitively that partial cells were sufficient to capture the long-term memories within traffic flow data.

Cui et al. (2018) used a convolutional recurrent neural network to forecast traffic speed. They presented two examples to interpret why the proposed model is good at capturing spatial features with the visualisation of convolutional layer weights. A deep CNN model may have hundreds of thousands of hidden representations, however, such as the weight of convolutional neurons and the weight of fully connected neurons. Visualising this large number of hidden presentations needs substantial computational power and analytics. Although visualisation might be a useful tool to interpret the reasons for good performance by providing hidden output, it might not be sufficient for precisely locating the crucial cells that lead to final classification results.

In image processing, some studies (Zhou et al., 2017; Zhang et al., 2017) have attempted to modify traditional CNNs to enhance their interpretability by applying either labels or filters to the hidden representations in order to encode more semantically meaningful knowledge in convolutional layers. The interpretable labels, ranging from colours, materials, textures, objects and scenes, could be used to help identify the location of an object from an image (Zhou et al., 2017). In transportation, however, traffic anomalies are generally linked to a cell, or several cells, rather than a region in the transferred matrix. The localisation of traffic anomalies, therefore, is in nature different from the problem of identifying a region with a bounding box. Additionally, during the learning process, the size and dimension of hidden representations will be greatly reduced due to the pooling and dropout techniques (Krizhevsky et al., 2012). Referring to the location using the diminishing weights in an inverse order is a kind of heuristic problem that needs lots of computational power and research efforts and where, even so, optimal results might not be guaranteed. Moreover, the traffic states across similar types of roads have similar properties and characteristics such as the shape of the traffic profile and the congestion formulation process. These similarities, indistinct characteristics and the requirement for a high resolution of localisation all increase the difficulty of using weights to locate traffic anomalies precisely. Considering the discussion above and the scope of this research, it is clear that an efficient approach to localisation is needed.

This chapter therefore aims to extend the work of the preceding chapters in order to locate

traffic anomalies precisely within the network, building on the previous discussion of translation layers and detection methods. Different iterative localisation methods will be presented in the following sections. The proposed localisation methods will be evaluated with both recurrent congestion and non-recurrent congestion case studies.

6.2 Localisation Methods

As implied in Section 6.1, several iterative methods are proposed here to address the traffic anomaly detection problem. This section will present these methods. Since there are very limited relevant studies in this field, i.e. seeking to extend the application of deep learning to the traffic anomaly detection field, the localisation in this research will be calculated iteratively based on the result of the detection model proposed previously by changing the format of output from a single output to a vector. Specifically, suppose that the matrix input resulting from the translation layer can be simplified into a two-dimensional format, the proposed method will consequently detect the row index and column index with the input of the original matrix and transposed matrix, as formulated as below.

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{d1} & x_{d2} & x_{d3} & \dots & x_{dn} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_d \end{bmatrix} \tag{6.1}$$

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{d1} & x_{d2} & x_{d3} & \dots & x_{dn} \end{bmatrix}^{\top} = \begin{bmatrix} {y'}_1 \\ {y'}_2 \\ \vdots \\ {y'}_n \end{bmatrix} \tag{6.2}$$

where x is the traffic variables in the matrix cell at the location with the index (d, n), y is the traffic anomaly binary labels for the d row and y' is the traffic anomaly binary labels for the n columns. To localise a traffic anomaly $A(d, n)$, the deep learning model will be trained iteratively with the row index and column index labels and thus locate the traffic anomaly with horizontal and vertical index respectively. Since limited research is available in this research

field, in this study, six different methods are proposed: weighted average probability, conditional probability, index, logistic regression, random forest and gradient boosting classifier. These are summarised below, assuming that the vector of row probability is P_{row} and the vector of column probability is P_{col}.

- **Weighted average probability**: The weighted average method assumes that the row and column have equal weight in the final decision, and thus, for each cell, the probability of a traffic anomaly is $P_{ij} = 0.5P_{row,i} + 0.5P_{col,j}$.

- **Conditional probability**: This method is based on the conditional probability in which the accuracy of row α and the accuracy of col β in the detection model have been assigned as the weights, i.e., $P_{ij} = \alpha P_{row,i} + \beta P_{col,j}$.

- **Index**: Instead of detecting the location with the evidence of probability, this method uses the indexes or labels from rows and columns directly with the decision function in Eq. 6.3.

$$f(x_t) = \begin{cases} 0, & 0 \in [label_i \cup label_j] \text{ at time } t. \\ 1, & label_i \cap label_j = 1 \text{ at time } t. \end{cases} \tag{6.3}$$

- **Logistic regression**: In contrast to the straightforward methods mentioned above, logistic regression uses the probability with different weights and bias optimised by a stochastic optimisation function. Mathematically, logistic regression estimates a multiple linear regression.

$$\log_b \frac{p(y=1)}{1-p(y=1)} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \tag{6.4}$$

By simple algebraic manipulation, the probability that $y = 1$ is as in Eq. 6.5.

$$p = \frac{1}{1 + b^{\beta_0 + \beta_1 x_1 + \beta_2 x_2}} \tag{6.5}$$

where x_1 and x_2 refer to the probability of the occurrence of a traffic anomaly in rows and in columns respectively, and $\beta_0, \beta_1, \beta_2$ are the parameter sets to be optimised during the training process, while b is the base of the algorithms.

- **Random forest**: this method applies random forest methods to the probability of rows and columns to locate the probability for each cell. The details of the random forest method can be found in Section 5.2.3

- **Gradient Boost classifier**: this method uses a gradient boost classifier to calculate the probability for each cell. Section 5.2.3 presents the details of GBC.

The decision function for all methods except for the index method is as in Eq. 6.6.

$$f(x_t) = \begin{cases} 0, & p < 0.5 \text{ at time } t. \\ 1, & p >= 0.5 \text{ at time } t. \end{cases} \tag{6.6}$$

In summary, six methods with different levels of simplicity have been proposed in this section. To evaluate the proposed methods, the Bath and London case studies (described in Chapter 4) will be used to evaluate the performance of RC and NRC detection using the evaluation metrics set out in Section 2.3.3. For the recurrent congestion detection, according to the dimension of the connectivity matrix translation, the vector size of 20 for both rows and columns has been extracted as the traffic anomaly labels. On the other hand, for the non-recurrent congestion detection, the output vector size of 22 and 18 for rows and columns, respectively, will be used as the labels for NRC anomaly detection (following the dimensions of the geographical grid layer). The following two sections will present the results based on the case study of RC and NRC localisation.

6.3 Case Study of the Localised Early Prediction of Recurrent Congestion

6.3.1 Data Description

Data extracted from the City of Bath are used for the case study for the early detection of recurrent congestion. The input data for the recurrent congestion detection are the same as in Chapter 4 and Chapter 5, while the output has been constructed in a different way to detect traffic anomalies at a network level. Specifically, the format of the output has been extracted into a

matrix format in which two types of vector with binary labels have been generated for the row and column index respectively. The binary labels have been generated using EM algorithms for all detectors.

The final model was calibrated using 70% training dataset, 10% validation dataset and tested with 20% testing dataset. The selection of hyperparameters is dependent on many previous studies as set out in the previous chapters. One example of the row index and column index extracted from the matrix format is shown in Figure 6.1.

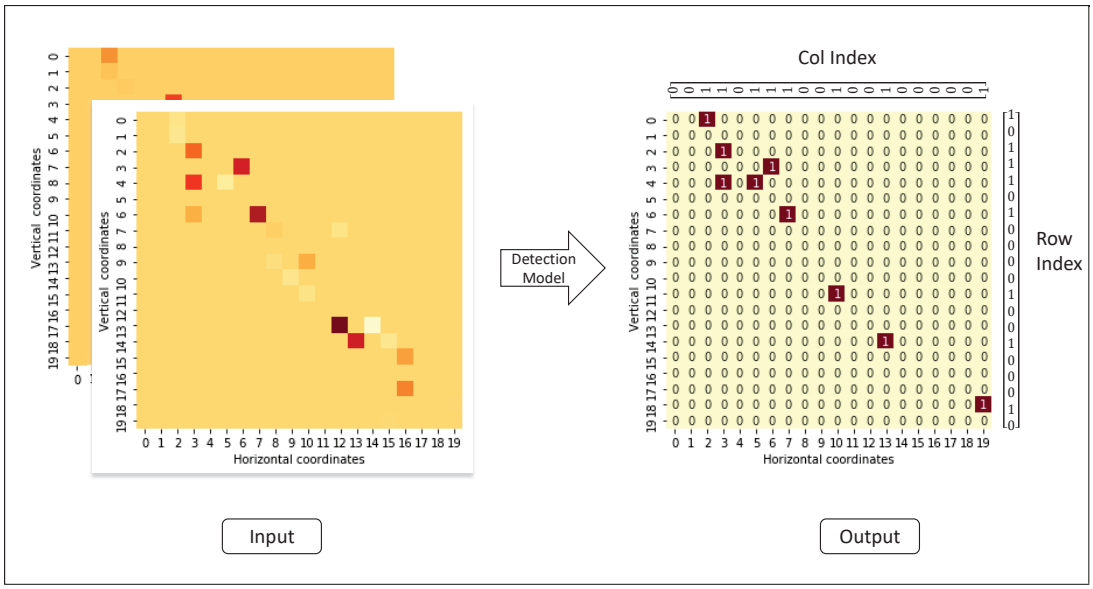


Figure 6.1: Localisation input and distribution of RC traffic anomalies in the matrix

Since the aim is to evaluate the accuracy of anomaly detection with the localisation function, the evaluation metrics listed in Section 2.3.3 are used but with some adaptions to reflect the changes in the output format. The main adaption is to aggregate the result in each cell across the matrix based on the testing dataset. More specifically, each cell inside the matrix except for the one without detector records will be aggregated for evaluation.

6.3.2 Results and Analysis

This section will present the results of early detection and localisation of recurrent congestion in the City of Bath case study. Table 6.1 shows the aggregated detection accuracy for the six investigated methods with current time lag and prediction horizon, i.e., time lag=0 and prediction horizon=0. In general, the random forest based localisation method outperformed others with

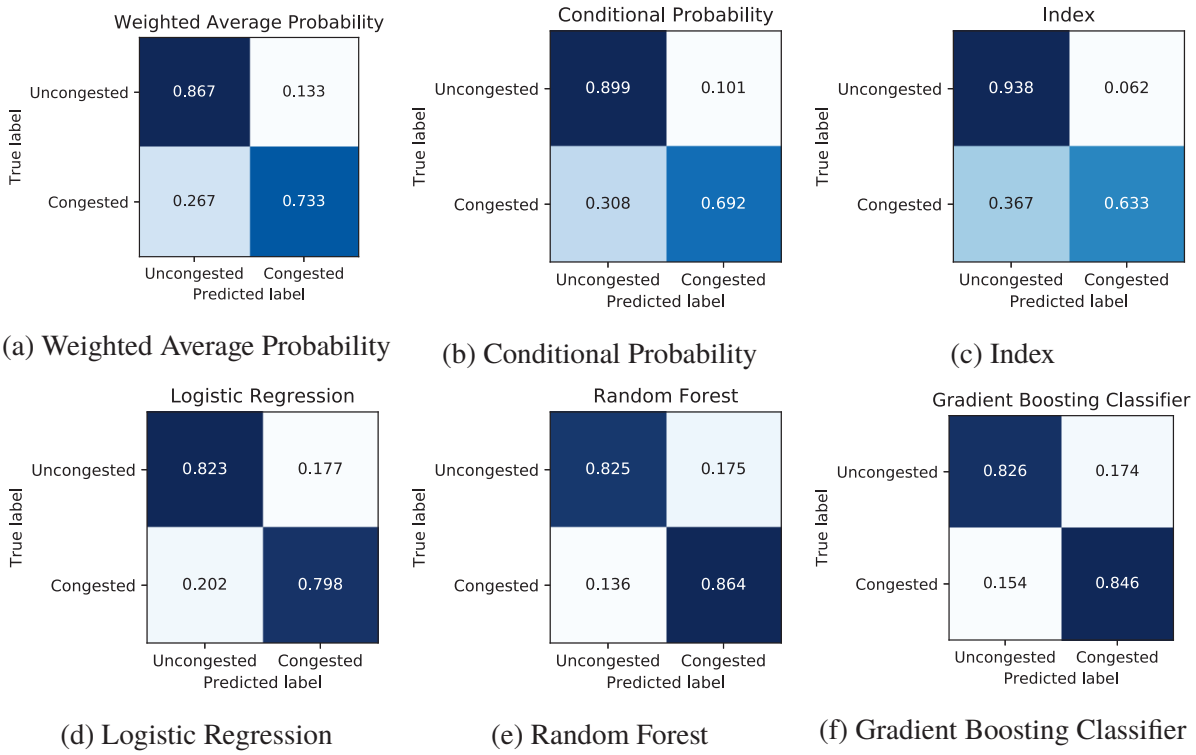
the highest DR, F1 score, precision and AUC. Even though the FPR rate is slightly higher than that of gradient boosting classifier, the difference is very marginal with a variation of around 0.01. AUC values are calculated with the probability distribution of binary labels. The index method does not use probability to locate the traffic anomaly, so the ROC curve and AUC value are not available for this method. Moreover, it is noted that the naive methods, including average probability, conditional probability and index method, are generally less accurate than the others. Similar results are also revealed in the confusion matrix in Figure 6.2. Figure 6.3 shows the ROC curves of the six methods. As discussed above, the ROC curve is unavailable for the index method, so the curve of the index method is missing from the plot. In light of the ROC curves, the order of performance is: random forest > gradient boosting classifier > logistic regression > conditional probability and weighted average probability. The superior performance of RF may originate from its ability to handle the individual predictions and combine these into a final prediction based on a majority vote on the individual predictions. This combination of individual predictions with voting systems significantly reduces the error rate (Breiman, 2001).

Table 6.1: Recurrent congestion detection results based on different localisation methods

Localisation Method	DR	FPR	F1 score	Precision	AUC
Weighted Average Probability	0.733	0.133	0.665	0.609	0.893
Conditional Probability	0.692	0.101	0.676	0.660	0.893
Index	0.633	0.062	0.683	0.742	N/A
Logistic Regression	0.798	0.177	0.808	0.818	0.895
Random Forest	0.864	0.175	0.847	0.831	0.928
Gradient Boosting Classifier	0.846	0.174	0.838	0.829	0.920

Figure 6.4 shows the changes of performance in terms of the increase of prediction lags. Firstly, except for FPR, the performance of random forest consistently outperformed than others no matter what time lags have been used. Even though the FPR of random forest is higher than that of naive methods, it is still an acceptable rate considering the dimension and complexity of traffic anomaly detection. For the random forest method, time lag of three is the changing point from the trend of decreasing to the trend of increasing. This result reemphasised the similar suggestion from previous chapters that the increased time lags do not necessarily contribute to the detection accuracy (Polson and Sokolov, 2017).

Figure 6.5 shows the localisation accuracy with the change of prediction horizons. As expected, DR, precision, F1 score and AUC curves have fluctuating but generally decreasing



(a) Weighted Average Probability (b) Conditional Probability (c) Index

(d) Logistic Regression (e) Random Forest (f) Gradient Boosting Classifier

Figure 6.2: Recurrent congestion detection confusion matrix based on different localisation methods

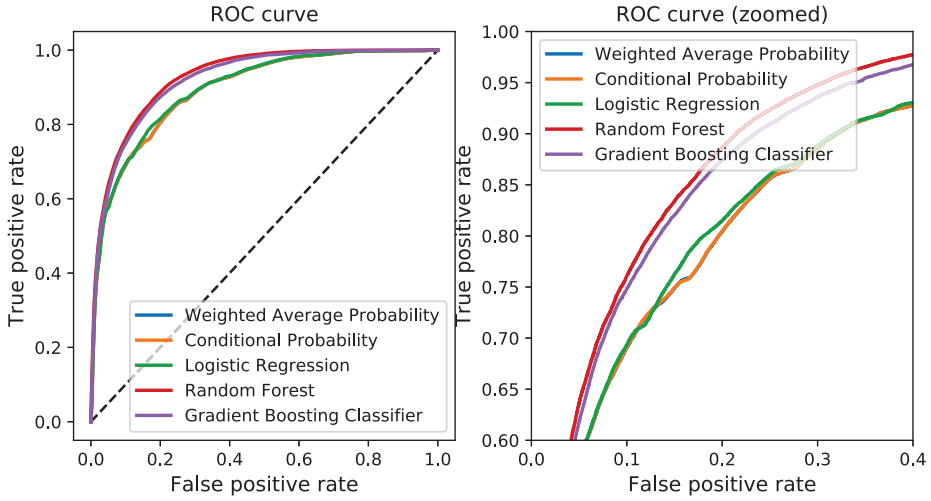
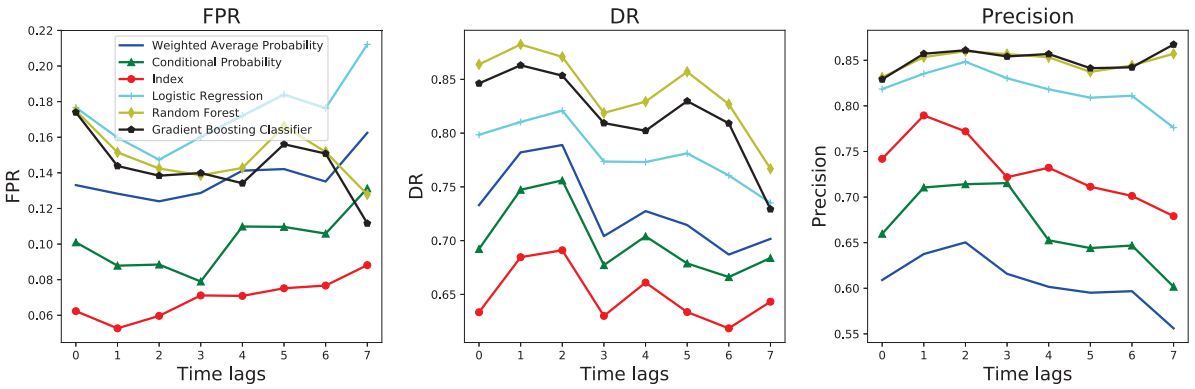
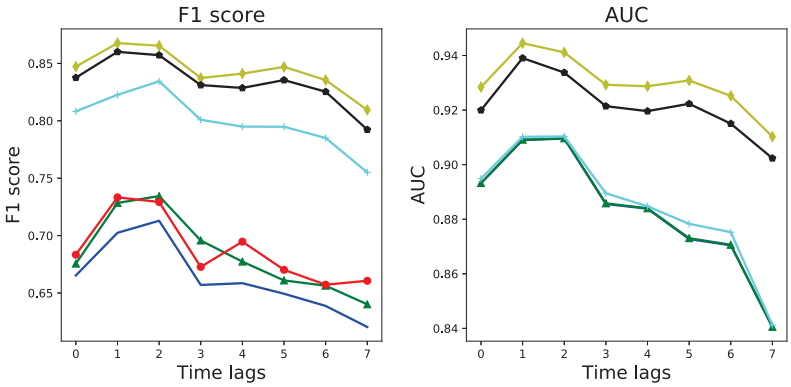


Figure 6.3: ROC curve for localised detection of recurrent congestion

trends for all six localisation methods. The reduced performance with larger traffic prediction horizons has been also suggested in Tan et al. (2016). Moreover, in terms of the different methods, random forest had superior performance based on the majority of the metrics across all prediction horizon. Even though the performance reduces with the prediction horizon, the detection performance is still acceptable, with a prediction horizon of seven. It is worth noting that



(a) FPR (b) DR (c) Precision



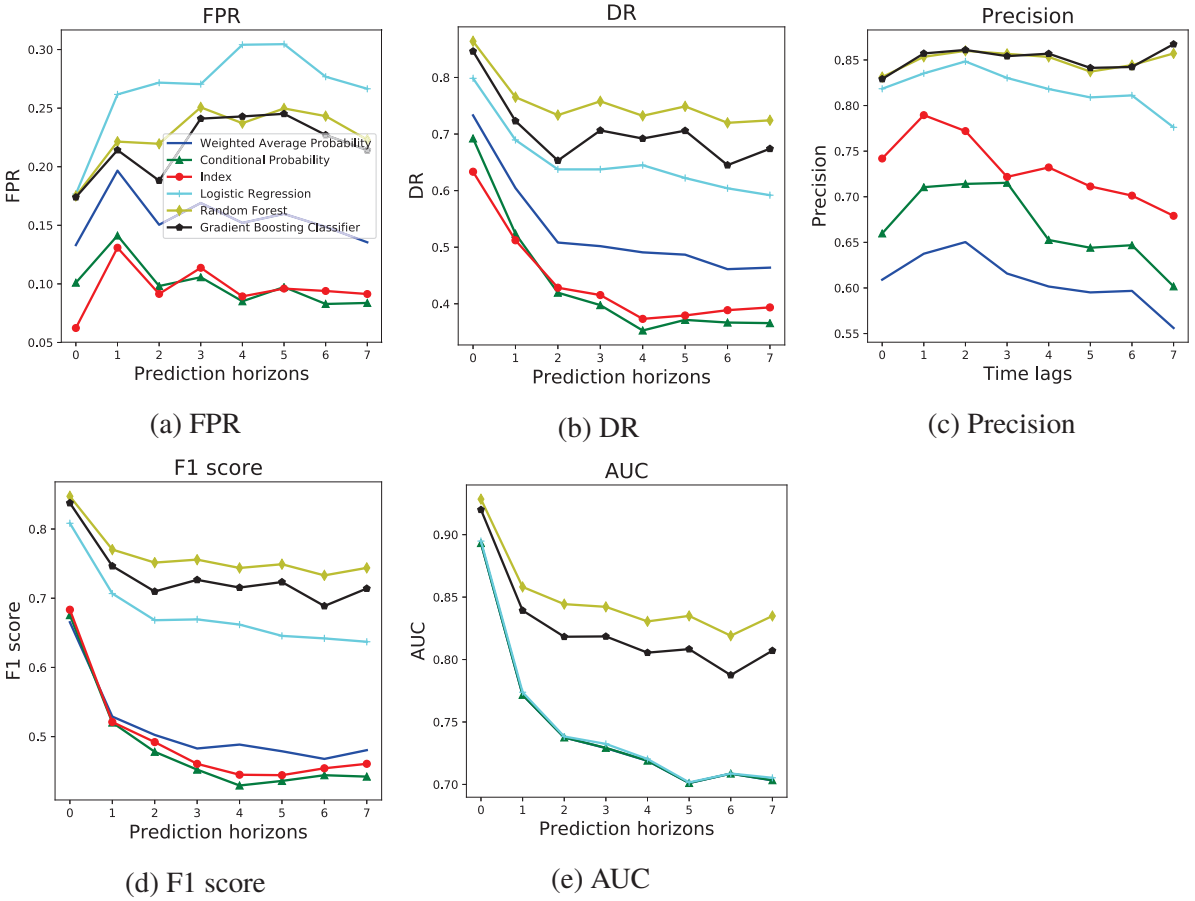
(d) F1 score (e) AUC

Figure 6.4: Localised detection of recurrent congestion in terms of time lags with prediction horizon=0

performance reduced compared to the model supervised by labels extracted from one link. This reduction is acceptable, however, since it is only a marginal reduction that is offset by a many hundred-fold increase in the resolution of detection. For example, DR dropped around 8.3% from 0.942 to 0.864, but the resolutions of the detection surged significantly from previously one label for a whole network to every single cell in a 20 by 20 matrix. The high resolution of detection makes it feasibility to locate traffic anomalies at the cell level by providing the row and column index. The full performance results with different combinations of time lags and time horizons have been included in Appendix C.3.

6.3.3 Summary

This section evaluated the performance of six traffic anomaly localisation methods in the context of the proposed connectivity matrix translation layer and CNN-LSTM early detection method. The new localisation function enables the model to detect multiple traffic anomalies occurring



(a) FPR (b) DR (c) Precision

(d) F1 score (e) AUC

Figure 6.5: Recurrent congestion detection and localisation in terms of prediction horizons with time lag=0

simultaneously across the network. The proposed method can predict recurrent congestion early by providing the row and column index. The evaluation is based on metrics aggregated across the network. The results suggest that random forest had superior performance in terms of most of the metrics, such as DR, AUC and precision. This superior performance is consistent even when the number of time lags and time windows has been increased. For the random forest method, it is notable that providing more information about previous time lags may not necessarily improve the detection accuracy. The time lag of three is the point at which the FPR curves change. The results also suggest that the model is capable of detecting seven time steps ahead with a reasonable accuracy.

6.4 Case Study of Localised Detection of Non-Recurrent Congestion

6.4.1 Data Description

The case study of the localised detection of non-recurrent congestion is based on the data extracted from London. The input data for the detection are the same as in Chapter 4 and Chapter 5, while the output has been constructed into a matrix to detect traffic anomalies at a network level according to Section 6.2. The grid translation layer was used as the input layer. And the output was transformed into corresponding dimensions before extracting the row index and column index, as shown in Figure 6.6.

The final model is calibrated using a 70% training dataset, 10% validation dataset and tested with 20% testing dataset. The hyperparameters are the ones used in the previous chapters, based on many previous studies (Krizhevsky et al., 2012; Ma et al., 2015a; Wu et al., 2018c).

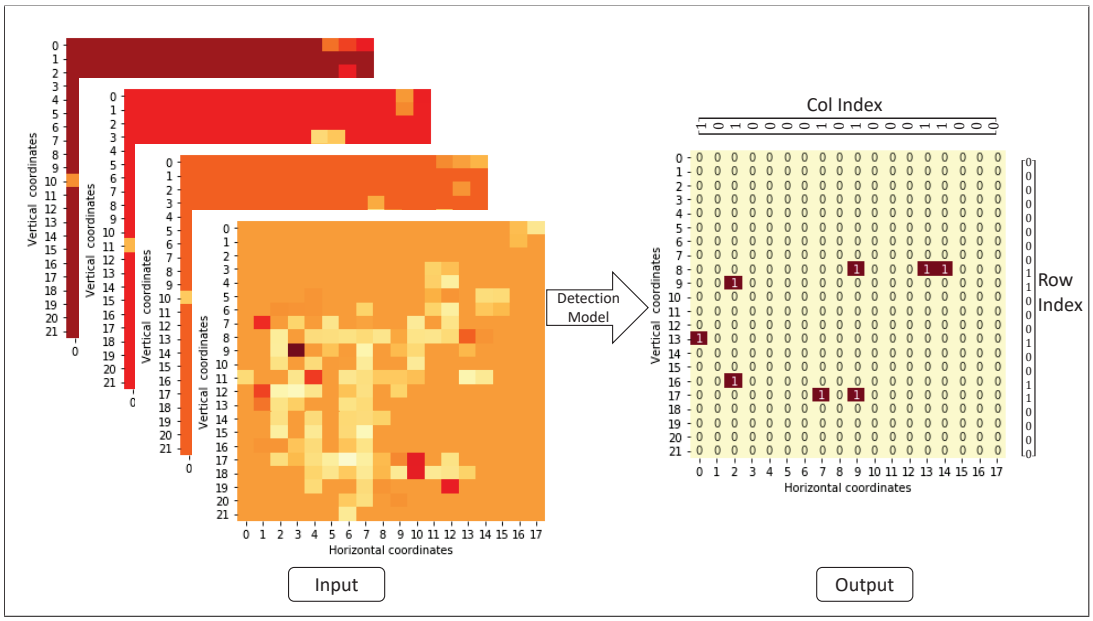


Figure 6.6: Localisation input and distribution of NRC traffic anomalies in the matrix

6.4.2 Results and Analysis

This section will present the evaluation results of NRC localisation. Table 6.2 shows the comparison between six methods with a time lag and prediction horizon equal to 0. Similar to the

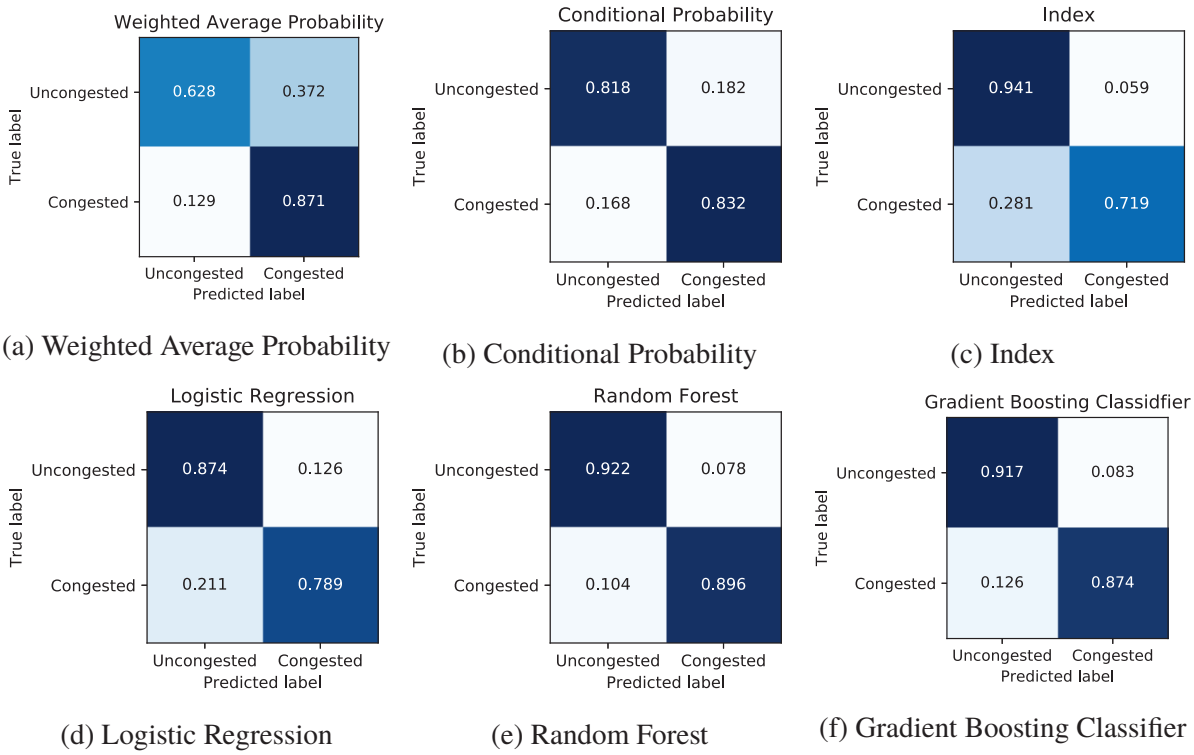
evaluation result with RC, the naive methods, i.e., average probability, conditional probability and index, are less comparative compared to the other three. In general, in terms of the localisation method, random forest outperformed the others with higher DR, F1-score, precision and AUC. Even though the FPR result for the random forest method is slightly worse than that for the index method, it is still a very low false detection rate.

Table 6.2: Nonrecurrent congestion detection results based on different localisation methods

Localisation Method	DR	FPR	F1 score	Precision	AUC	MTTD(s)
Weighted Average Probability	0.871	0.372	0.251	0.146	0.905	82.779
Conditional Probability	0.832	0.182	0.386	0.251	0.906	80.065
Index	0.719	0.059	0.571	0.473	N/A	82.922
Logistic Regression	0.789	0.126	0.824	0.862	0.908	157.484
Random Forest	0.896	0.078	0.908	0.920	0.971	158.705
Gradient Boosting Classifier	0.874	0.083	0.894	0.914	0.964	157.485

Figure 6.7 shows the confusion matrix for the six methods where the random forest has the highest detection rate and relatively low FPR compared to the other methods. Figure 6.8 explicitly presents the order of performance of different localisation methods, in which the curve for the random forest is closest to the left top corner which means that it has higher true positive rates with lower false positive rates.

Figure 6.9 shows the sensitivity analysis of localisation performance with the change of time lags. Since the differences in the performance of the six methods are relatively large, the shape of every performance curve is not obvious. Nonetheless, it is clear that a bowl shape exists for some localisation methods, such as the DR curve for logistic regression. Moreover, the performance of random forest is consistently superior to others based on DR, precision and AUC metrics. Although its FPR is consistently slightly higher than that of the index methods, it is still relatively low and acceptable for NRC detection. The index method has the lowest FPR rate because it only flags NRC when both row and column have an index with NRC, which greatly reduces the number of false positives. The random forest method, however, has the lowest FPR when the time lag equals to four. This optimal time lag has also been suggested by other studies (Ma et al., 2015a).



(a) Weighted Average Probability (b) Conditional Probability (c) Index

(d) Logistic Regression (e) Random Forest (f) Gradient Boosting Classifier

Figure 6.7: Confusion matrix for localised detection of non-recurrent congestion

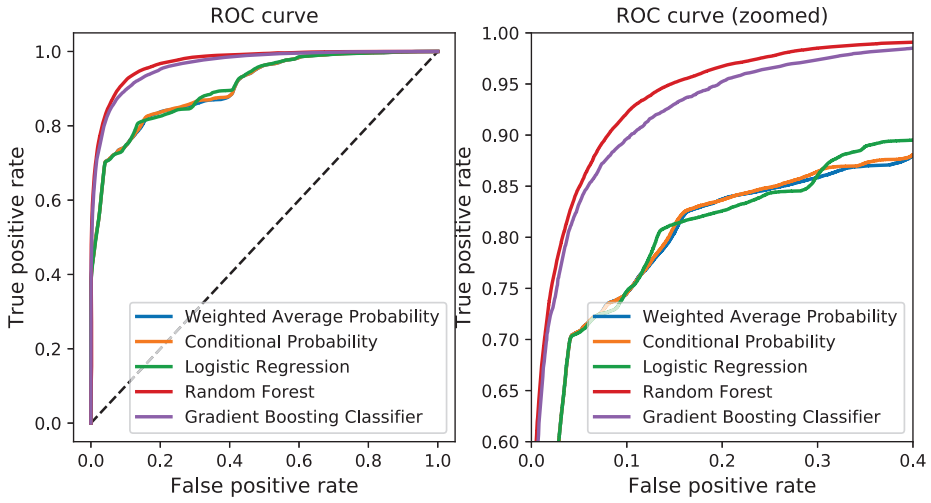
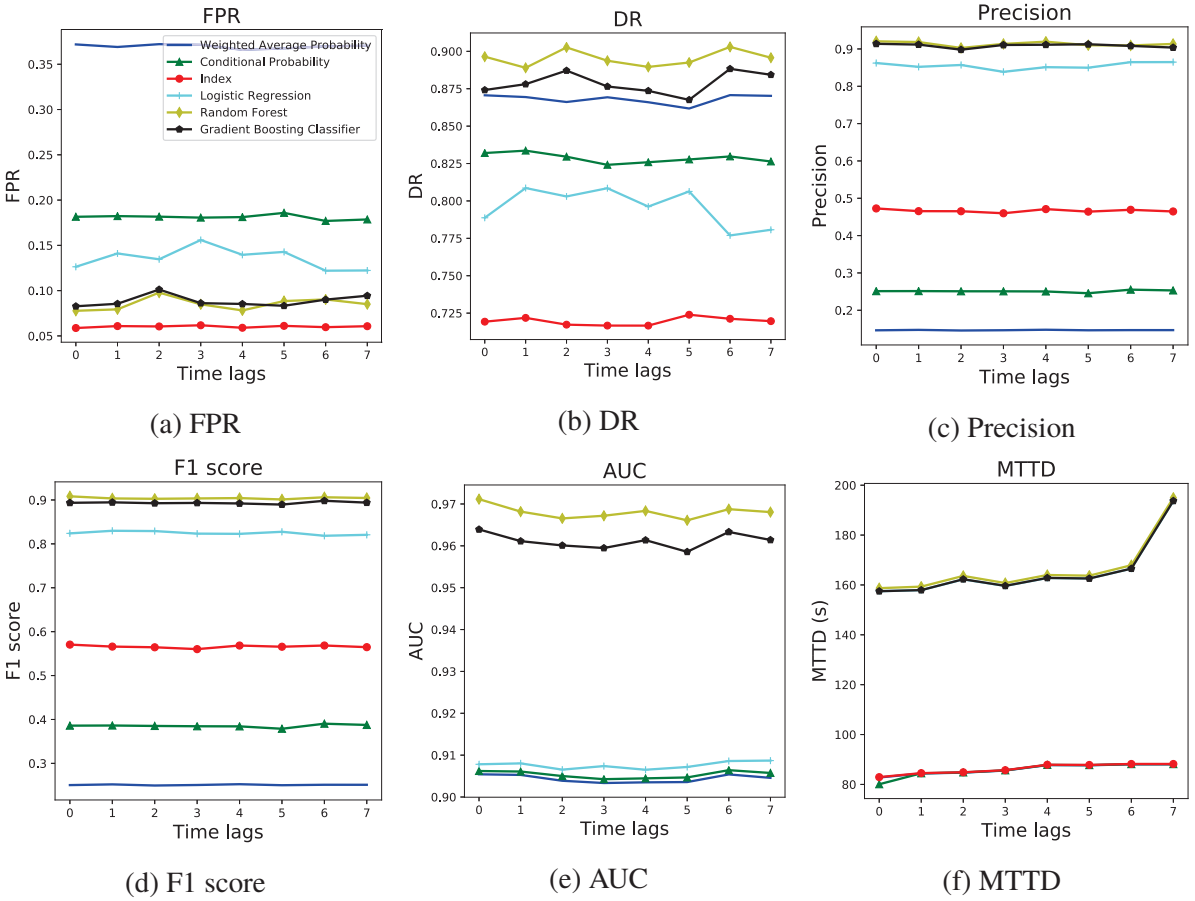


Figure 6.8: ROC curve for localised detection of non-recurrent congestion

6.4.3 Summary

This section presented the performance of the proposed methods for NRC detection and localisation. The results suggest that the proposed methods are capable of detecting and locating multiple NRCs at a network level. Similar to the results from RC detection, random forest consistently gave superior performance compared to others, with higher DR, precision and AUC.



(a) FPR (b) DR (c) Precision

(d) F1 score (e) AUC (f) MTTD

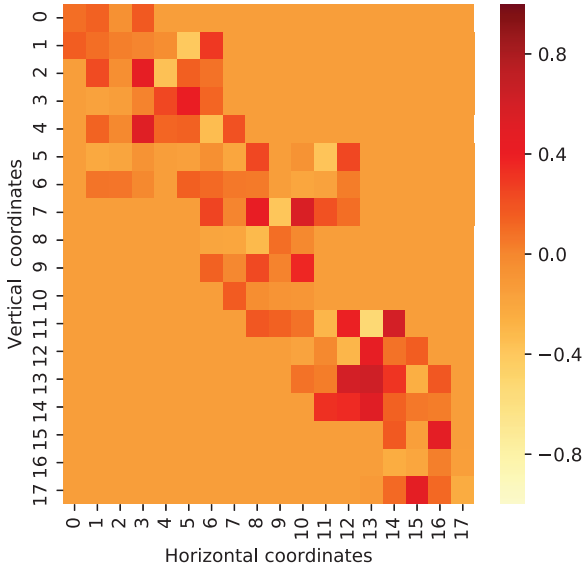
Figure 6.9: Localised detection of non-recurrent congestion in terms of time lags with prediction horizon=0

but relatively lower FPR no matter the size of time lags used in the model. The results also suggested that random forest has a FPR changing point when time lag equals to four, and adding additional traffic information beyond a time lag of four may, therefore, not contribute to the performance of NRC detection.

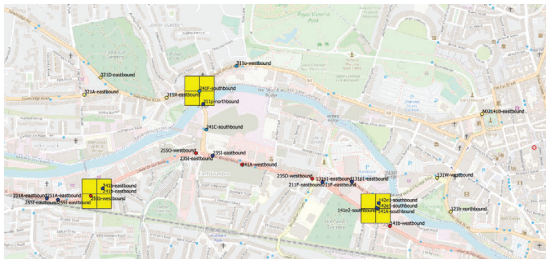
6.5 Visualisation of Localisation and its Limitations

To better understand how the proposed detection methods learn from the matrix for traffic anomaly detection, some examples of convolutional layer outputs have been visualised. Visualisation of the convolutional layer outputs has also been suggested by Wu et al. (2018c) and Cui et al. (2018). Figure 6.10 (a) visualises the first convolutional layer output, in which the dark red roughly reflects the location of the three busy segments highlighted in Figure 6.10 (b) with yellow boxes for early RC detection. Three segments are quite representative because they

are located near the main entrance or exit connecting either main corridors or signalled multiple lane roads. The area contains frequently-congested junctions connecting to the city centre. This example illustrates that the proposed method is capable of learning the spatial correlation from the matrix. This capability is not consistently indicated in the other examples of output layers, however, as shown in Figure 6.11, where two visualised outputs were extracted from the first convolutional layer but from different neurons. In contrast to Figure 6.10, the correlation between the output and physically congested sections of road cannot be interpreted straightforwardly. Meanwhile, the additional temporal dimension added into the model for early RC detection may introduce noise in terms of precisely locating RC, since RC may propagate from one cell to another in different time steps.



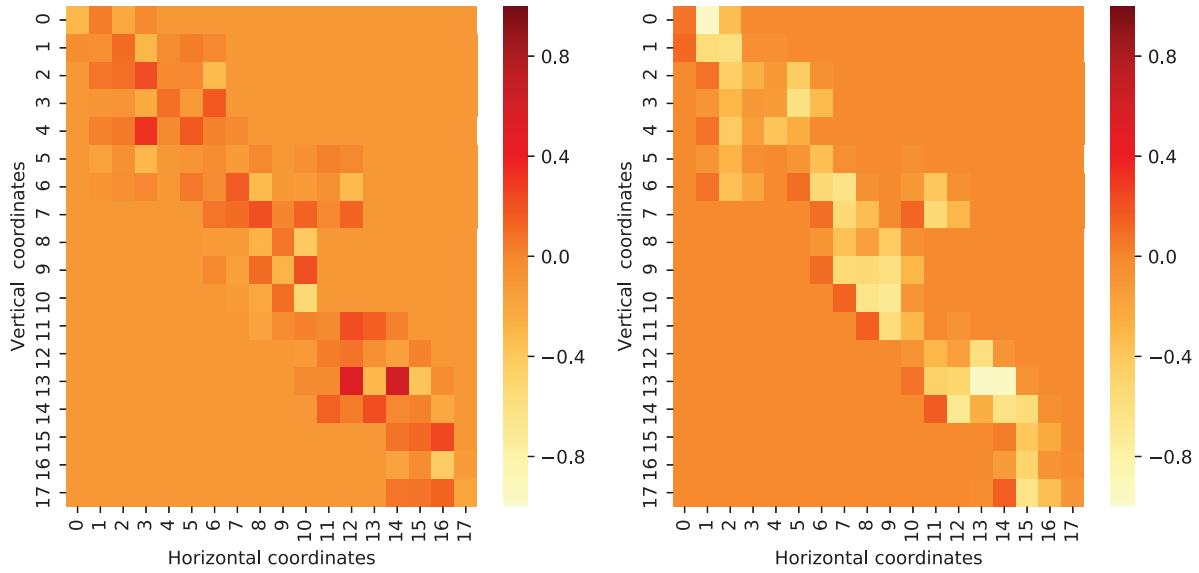
(a) Example of the first convolutional layer output



(b) The RC study area
(Map Source: OpenStreetMap)

Figure 6.10: Visualisation of hidden outputs for RC localisation and corresponding map

A similar understanding can be found for the NRC detection. Figure 6.12 (a) presents an example of the convolutional layer output where the areas with dark colours in the output matrix are all located in areas with most traffic incidents. Since the geographical grid translation has been used for this method, it is relatively straightforward to match the location on the map. Figure 6.12 (b) shows the maps of the study area, in which the green dots represent the location of loop detectors while the red dots refer to traffic incident points. Two yellow boxes highlighted the area with most traffic incidents. The upper box is located just outside Euston mainline rail station and underground station while the lower box is located near the busy junction between the

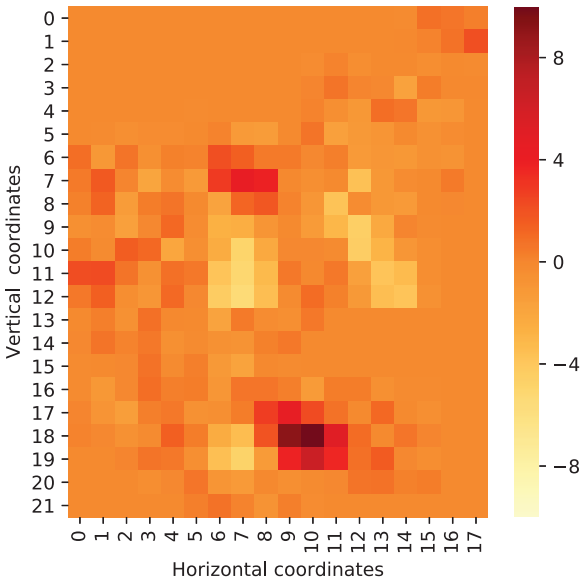


(a) Example of convolutional layer output (b) Example of convolutional layer output

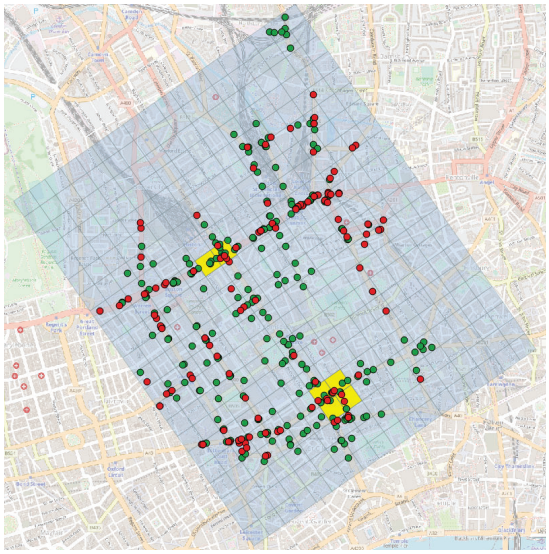
Figure 6.11: Examples of hidden output representation without significant features for RC localisation

A40, New Oxford Road and Theobolds Road. The mapping between the hidden representation of the convolutional layer and the incident maps indicate that the proposed model can extract spatial features into a high dimension. Nevertheless, this indication is not transferable to all hidden output matrices. Two abnormal examples have been given in Figure 6.13 where no significant cells or areas in the matrix could be located as indicative information for mapping traffic incidents. Additionally, there are many other representative areas in the graph convolution output matrix, but they all cannot be shown due to space limits.

Given the potential to find the location using the hidden outputs, at the beginning of this research, some efforts were attempted to locate the traffic anomaly using hidden weights, bias and outputs of the deep learning model. There were several significant challenges in this, however. First, as mentioned in Section 6.1, traffic anomaly detection in the field of transportation is by nature different from object detection in image processing due to having fundamentally different features. Secondly, traffic anomaly localisation requires a method to return the index to a cell level. This high resolution requirement is hard to achieve with the gradually shrinking size of hidden output during the pooling and flattening process of a CNN. Thirdly, a CNN could have millions of parameters, i.e., weights and biases. Referring to the location using the diminishing weights in an inverse order is a nonlinear heuristic problem which requires lots

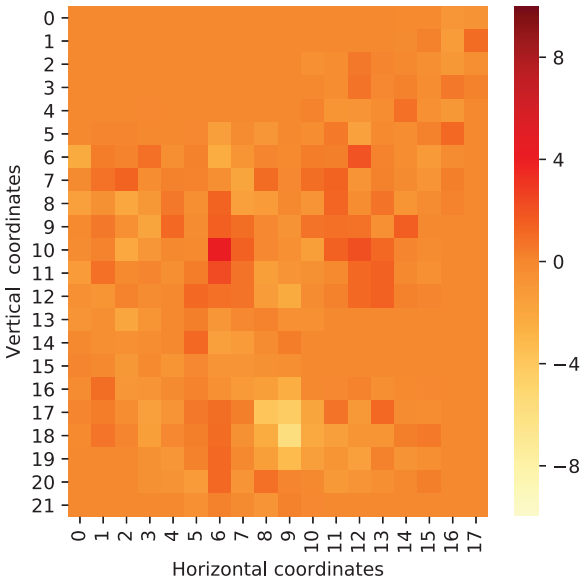


(a) Example of the first convolutional layer output

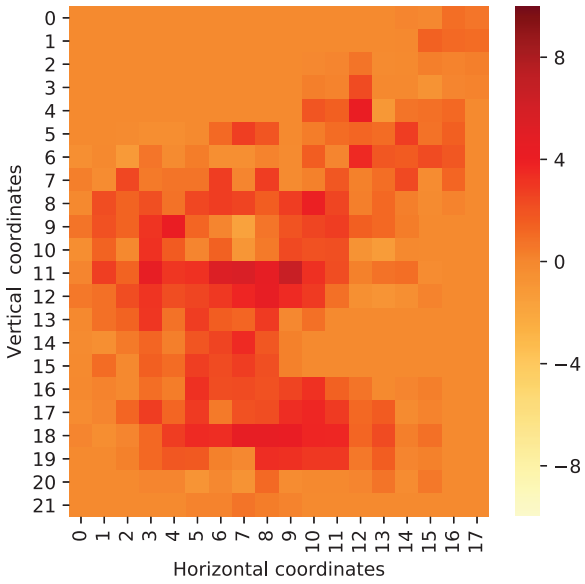


(b) The NRC study area
(Map Source: OpenStreetMap)

Figure 6.12: Visualisation of hidden outputs for NRC localisation and corresponding maps



(a) Example of convolutional layer output



(b) Example of convolutional layer output

Figure 6.13: Examples of hidden outputs without significant features for NRC localisation

of computational power and research efforts with optimal results not guaranteed. Considering these challenges and the scope of this research, the iterative methods presented in this chapter were finally adopted.

In summary, the visualisation could be an appropriate way to interpret the learning process of a deep learning model. By visualising the hidden output matrix and comparing with the

traffic states or traffic incidents of the traffic network, one can find that the proposed detection method effectively captures spatial dependencies sufficiently to identify critical areas in the traffic network.

6.6 Summary

This chapter has addressed the problem of localisation for a network-based model. The functionality of localisation is critical for a network-based model since it enables the model to predict multiple traffic anomalies occurring simultaneously across a complete traffic network. Different iterative methods ranging from simple average probability, conditional probability, index, logistic regression, random forest and gradient boosting classifier were used to predict and locate the traffic anomalies. Data from Bath and London were used to test RC and NRC localisation, respectively.

For RC and early RC localisation, random forest outperformed other models based on most of the metrics and it was shown that increasing the size of time lags might not be useful for the detection and localisation. The results suggested that the optimal time lag for the Bath case study is three as it provided the lowest false positive rate for early RC detection with the localisation function. In terms of prediction horizons, the model is also capable of predicting the traffic states two hours ahead with a reasonable accuracy. On the other hand, for NRC detection, the proposed models in which random forest had the superior performance are able to localise multiple traffic incidents at a network level.

This section also visualises the hidden representation of the deep learning methods. This visualisation can help to interpret its capability to capture the spatial correlations in the learning process. However, partial visualisation examples might not be enough to interpret the model performance properly. Considering the scope of this study, a solid and theoretical approach might be a worthwhile subject for future research.

Chapter 7

Robustness Analysis

The previous chapters in this thesis focused on theoretical traffic anomaly detection using an empirical framework. The experimental evaluation of this framework may be constrained, however, by the limitations of the case studies selected. In practice, to apply such traffic anomaly detection models in the real world, more experiments are needed to verify its robustness and reliability. This chapter will, therefore, focus on the practical issues in terms of the model's robustness with different network topologies, sensor faults and missing data caused by systematic errors.

Specifically, the structure of this chapter is as follows. Section 7.1 and Section 7.2 begin with a brief introduction of the robustness analysis in the transportation field and microscopic traffic simulation. Section 7.3 tests the proposed traffic anomaly detection framework (i.e., random forest localisation based CNN-LSTM model with a connectivity matrix translation layer) on two different simulation networks, i.e., the Sioux-Falls network and Luxembourg traffic network. Section 7.4 and 7.5 will further investigate the model with the involvement of sensor faults and missing data based on Sioux-Falls network. Section 7.6 summarises the main findings of this chapter.

7.1 Introduction and Background

Data-driven methods, such as machine learning and deep learning methods, map traffic conditions between current and past traffic data without explicitly formulating the physical traffic processes in the way that model-based models (Van Lint et al., 2005). Although data-driven methods require less expertise in traffic modelling and are fast and accurate, they are typical location-specific solutions with specific input or customised model configurations (Van Lint et al., 2005). It is therefore important to understand if the model that has been built in this thesis can maintain its original performance in different locations. Moreover, another factor that may 'damage' the performance is faulty traffic data or missing data (Van Lint et al., 2005; Luan et al., 2006). Thus, before the traffic anomaly detection model can be applied in practice, it is necessary to study its behaviour when faced with unreliable and missing input data.

Robustness is defined as the degree to which a system or component is capable of continuing to function effectively in the presence of internal and external invalid inputs or disturbances (Geraci et al., 1991). The formulation or evaluation of the theoretical model is generally subject to some constraints or prerequisites. Robustness analysis is a way to bridge the gap between the theoretical model and practical situations where unexpected structures, interruptions or input faults can occur (Bickel et al., 1976). The ability of the proposed model to resist disturbance and maintain the predesigned functionality is significant for the application of the model (Zhou et al., 2017). Robustness analysis is crucial for traffic engineering as it can ensure the generalisation of the theoretical model in the context of a real-world environment that may confront it with differences and challenges compared to the original environment in which the model was developed (Scott et al., 2006).

Generally, measures of robustness in network-based models can be categorised into two types. The first is based on the topology of networks without taking into account disruptions or unexpected environments while the second involves a wide range of complex factors that directly or indirectly cause disruptions, such as systematic sensor failures (Snelder et al., 2012). Regarding the first type, the topology of the urban road network has been analysed by many researchers (Jiang and Claramunt, 2004; Tian et al., 2016). The typical robustness analysis of road topology focuses on removal or blockage of one or more network links, particular those heavily travelled, and then checking the reliability of the network (Scott et al., 2006). The ob-

jective of this is to identify the bottlenecks or critical transit links in the traffic network (e.g., (Sullivan et al., 2010)). This objective is not coherent with the traffic anomaly detection problem, however, since this already has the functionality to identify interruptions or bottlenecks. In this research, therefore, the key robustness analysis in terms of network topology is equivalent to analysing its transferability, so as to answer the question of whether, if the network geometry or size changes, the model will maintain the same level of performance in terms of anomaly detection. To date, a wide range of research has presented relevant studies on the transferability of prediction models (Luan et al., 2018), but few studies have focused on the transferability of traffic anomaly models at a whole network level.

As for the second type of robustness analysis, which aims to assess the impact of interruptions on the model performance, of particular interest in this research is evaluating the model performance in the presence of sensor faults. Data extracted from ILD can be corrupted by various sensor faults. A sensor fault refers to ILDs consistently outputting erroneous data such as overestimating and underestimating the point flow (Robinson, 2005). The reasons for these erroneous data include broken cables, crosstalk[1], hanging[2], parked vehicles, communication failure between the ILD and the Traffic Management Centre (TMC) (Luan et al., 2006). Robinson (2005) comprehensively summarised different types of ILD errors and their corresponding treatment algorithms. Lee et al. (2010) proposed different algorithms to mimic the real-world data by introducing different types of error into a 'perfect' dataset obtained from microscopic simulation. To date, however, limited research has investigated these overestimated or underestimated sensor faults to a traffic anomaly detection problem.

On the other hand, apart from the overestimated or underestimated sensor faults, another special type of sensor fault is missing data which is generally caused by communication failure or malfunction. In such circumstances, no data will have been recorded in the TMC (Robinson, 2005). For example, on average 15% of the ILDs on the Dutch freeway monitoring system may be out of operation or producing unreliable measurements (Van Lint et al., 2005). A substantial amount of research has studied the prediction accuracy and robustness with respect to missing data (Chen et al., 2001; Van Lint et al., 2005; Tan et al., 2013; Laña et al., 2018; Tian et al., 2018), but this issue has not been adequately addressed in the literature on the application of

[1]Interaction between ILD and other electrical devices causes ILD to output faulty '1-bit'.
[2]the ILD getting stuck in either an 'on' or 'off' position.

traffic anomaly detection across an entire network.

In summary, accuracy and robustness with respect to different network geometries, and in respect to corrupt and missing input data are key aspects for traffic anomaly detection models before they can be applied in a real-time application environment. All these factors will be discussed in the following sections. The robustness analysis will focus on the problem of the early detection of recurrent congestion since it not only theoretically covers a supervised learning part as NRC detection but also involves an early alarm function which makes it more valuable to discuss within the limited space of this thesis.

7.2 Microscopic Traffic Simulation

The complexity of traffic dynamics and the difficulties in readily performing experiments with real-world traffic make simulation an increasingly popular and effective tool in traffic engineering, particularly in the presence of ITS systems (Toledo et al., 2005). Traffic simulation can be used for analysing and visualising a wide variety of dynamic problems, especially those associated with sophisticated processes that cannot readily be described analytically, or for those large-scale real-world situations that are highly detailed (Cascetta, 2013).

According to the level of detail, traffic flow models can generally be classified into microscopic, mesoscopic and macroscopic models, where microscopic ones look at the movements of each vehicle and their interactions within a traffic stream, whereas macroscopic models treat the traffic stream as a whole. Mesoscopic models, meanwhile, view the transitional situations between microscopic models and macroscopic models (Cascetta, 2013). Microscopic simulation details the motion of each vehicle and its interactions by synchronously modelling agent-based decisions such as acceleration, deceleration, lane change and route choice and updating these kinematic parameters of each movement at every simulation time step (Azevedo et al., 2017).

One example of a microscopic simulator is SUMO (Krajzewicz et al., 2002), which is an open-source software commonly used in a wide variety of applications, such as evaluating alternative treatments, testing new complicated scenarios and safety analysis (Cascetta, 2013). Table 7.1 presents the advantages and disadvantages of representative simulation software. Compared to other microscopic traffic simulation software, SUMO, created by the German Aerospace Cen-

tre (DLR), is open source and highly portable with various Application Programming Interfaces (APIs) to control the simulation remotely, and thus is ideally designed to handle large road networks (Krajzewicz et al., 2012). For a detailed comparison between SUMO and other microscopic traffic simulation software please refer to the review paper from Pell et al. (2013) and Saidallah et al. (2016).

SUMO has been used intensively in the transport research such as traffic signal control (Xie et al., 2012), route choice (Xie et al., 2014) and traffic mobility (Bedogni et al., 2015). Many studies have been carried out on traffic incident detection and monitoring using SUMO. For instance, Baiocchi et al. (2015) proposed a Vehicle Ad-Hoc Networks (VANET) framework for real-time incident detection validated through SUMO simulations. Vandenberghe et al. (2012) investigated the feasibility of expanding traffic monitoring systems with floating car data based on SUMO simulation.

7.3 Robustness with respect to Network Topology

This section investigates the accuracy and robustness of the proposed traffic anomaly framework with respect to different network geometries. The empirical study is in the context of the simulation environment. This section first includes the microscopic simulation description followed by the comparison and discussion of results. Two traffic networks with different levels of complexity and characteristics (e.g., traffic signal control, type of roads, number of lanes, etc.) have been used for analysing its robustness. One network is a relatively small network based on the Sioux-Falls network (LeBlanc, 1975) while the other one is a larger calibrated simulation of the City of Luxembourg (Codecá et al., 2017).

7.3.1 Sioux-Falls Simulation

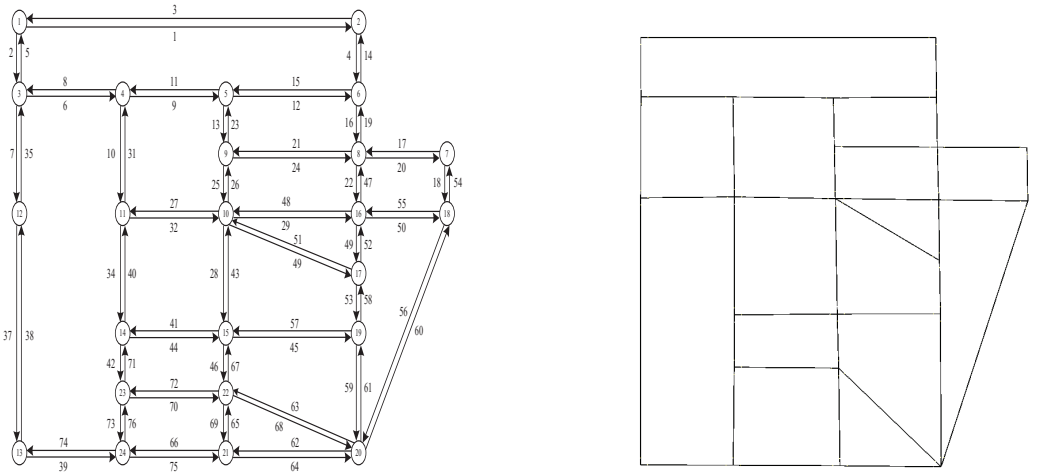
The network consists of 24 nodes and 76 links, with each link having a single lane with an ILD located downstream of each node (Figure 7.1 (a)). The simulation was implemented with a dynamic demand profile in SUMO (Krajzewicz et al., 2002).

The traffic profile has a typical pattern according to the typical traffic profile at different times of day and day of the week. The dynamic demand profile with a generic timeline for a

Table 7.1: Comparison between representative microscopic traffic simulation software

Software Name	Advantages	Disadvantages
VISSIM	(1) Allowing definition of the full range of vehicle types covering private and public transport; (2) Providing user-developed applications; (3) Powerful interfaces and integration; (4) The vehicle behaviour is taken into consideration such as changing lanes.	(1) Subscription fee; (2) Complexity issues involved with setting up.
TRANSIMS	(1) Time dependent OD demand-generation; (2) Private and public transport multi-model transport; (3) Providing significant changes in the travel forecasting process.	(1) No transport innovations available; (2) No dashboard with KPIs.
MATSIM	(1) An activity-based approach demand-generation; (2) Private and public multi-model transport; (3) Open source; (4) Able to simulate the traffic of a vast region; (5) Agent-based and generates individual activity.	(1) No transport innovations available; (2) No dashboard with KPIs; (3) Not detailed vehicle behaviour.
AIMSUN	(1) Allowing the modelling of different network models in the same simulation; (2) Able to reproduce real traffic conditions of any transport network; (3) Capable of communicating with external user-defined applications.	(1) Few ITS-function available; (2) Subscription fee.
Paramics	(1) Parallel simulator; (2) Private and public multi-model transport, parking and port; (3) Model of an entire city traffic system; (4) Providing a realistic representation of the landscape with 2D/3D visualisation.	(1) The reliance on origin-destination matrices to derive traffic volumes.
SUMO	(1) Assignment model; (2) The vehicle behaviour is taken into consideration such as changing lanes; (3) Private and public multi-modal transport; (4) Providing various APIs to remotely control the simulation; (5) Open source and free.	(1) No transport innovations available; (2) Software interface.

day of the week was generated by OD pairs in the network. Figure 7.2 shows an example of



(a) Topology of Sioux-Falls network (Source: LeBlanc, 1975)

(b) Topology in the SUMO simulation

Figure 7.1: Map of Sioux-Falls simulation

traffic flow and occupancy patterns across one day with data generated from the Sioux-Falls simulation. Both traffic flow and occupancy show a typical pattern at different times of the day. To verify the simulation further, the distribution of average traffic speed and travel distance have been plotted as Figure 7.3. The average traffic speed and travel distance are not symmetrically distributed. The travel distance is more concentrated to the median while travel speed is more skewed towards lower traffic speed due to traffic congestion.

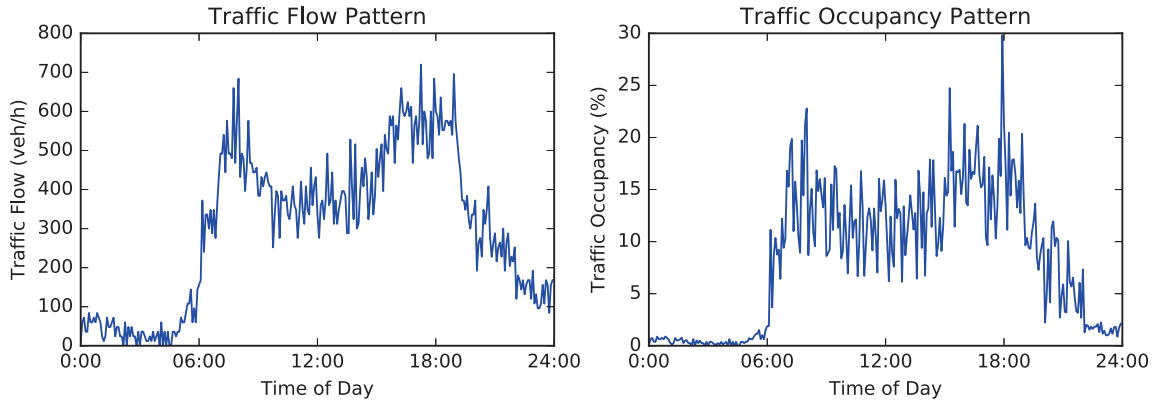
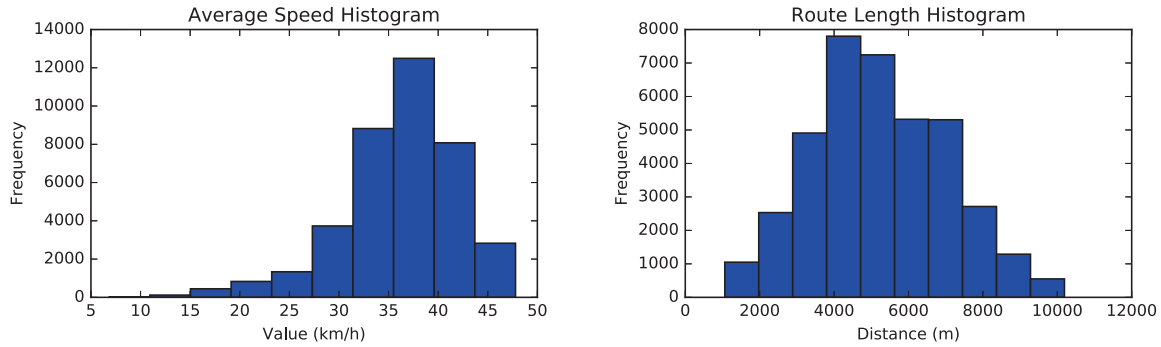


Figure 7.2: Example of traffic flow and occupancy from Sioux-Falls simulation

The traffic flow data were collected by loop detectors at five-minute intervals every day. The simulation ran for a total duration of ten weeks in simulation time. In order to take into account the warming up and cooling down period of the simulation, as well as the balance of datasets, data from 7:00 to 19:55 are used in this study as the input for the proposed traffic



(a) Distribution of average traffic speed (b) Distribution of route length or travel distance

Figure 7.3: Validation of Sioux-Falls simulation output

anomaly detection.

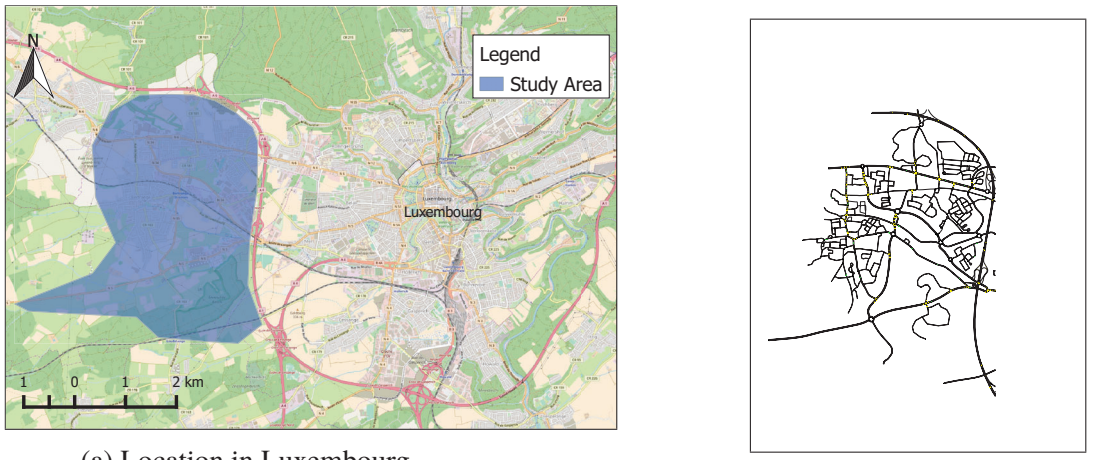
The input matrix was generated using a connective matrix where the x-axis and y-axis represent the nodes while values in each cell are the corresponding traffic data connected two nodes in time step t. The generated matrix has 24×24 dimension with the sample size of 11760, while the output is a vector of binary values where 1 stands for congestion and 0 for no congestion, respectively, as labelled by the EM algorithm. In this way, the input and output of the proposed model are defined.

7.3.2 Luxembourg Simulation

Another well-calibrated simulation used in this research is based on the road network in the City of Luxembourg (Codecá et al., 2017). Overall, the simulated vehicles made about 25 million trips during three months and traffic flows are collected by inductive loop detectors at five-minute intervals for ten weeks in simulation time. Figure 7.4 shows the topology of the simulation network. The main parameters of these simulations are listed in Table 7.2.

In order to validate the simulated data, traffic flow, occupancy, speed and travel distance have been plotted for verification. Specifically, Figure 7.5 shows the typical flow and occupancy patterns extracted from the Luxembourg simulation while Figure 7.6 indicates the existence of a long right tail of average speed and distance distribution. This long tail might be due to the high proportion of highway and arterial roads in the Luxembourg (Codeca et al., 2015).

Similar to the Sioux-Falls simulation, taking account of the warming up and cooling down



(a) Location in Luxembourg (Map Source: OpenStreetMap)

(b) Topology in the simulation

Figure 7.4: Map of the Luxembourg simulation case study

Table 7.2: Simulation parameter values

Parameter	Value
SUMO simulation duration (s)	86400
Simulation area (km^2)	8
Total number of edges	635
Total length of edges (km)	89
Vehicles inserted	287034
Number of traffic stream composition	8 (7 types of car and 1 bus)
Data resolution (s)	300

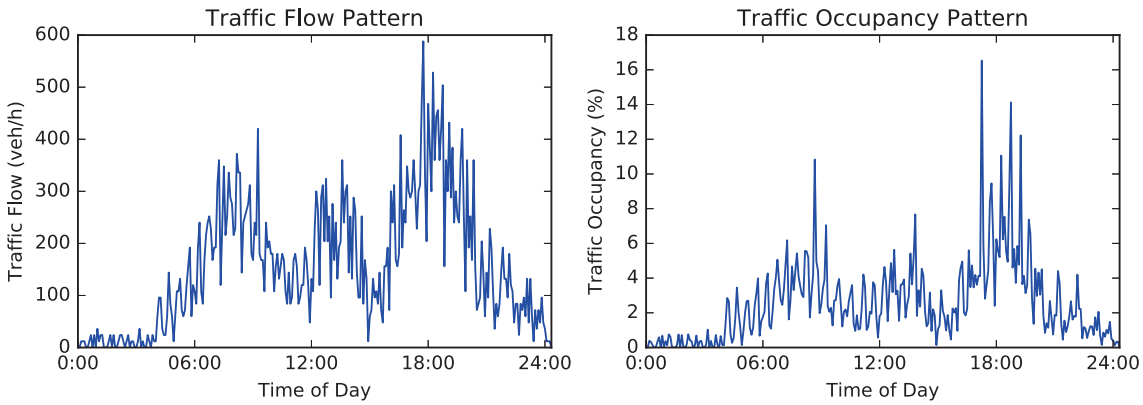
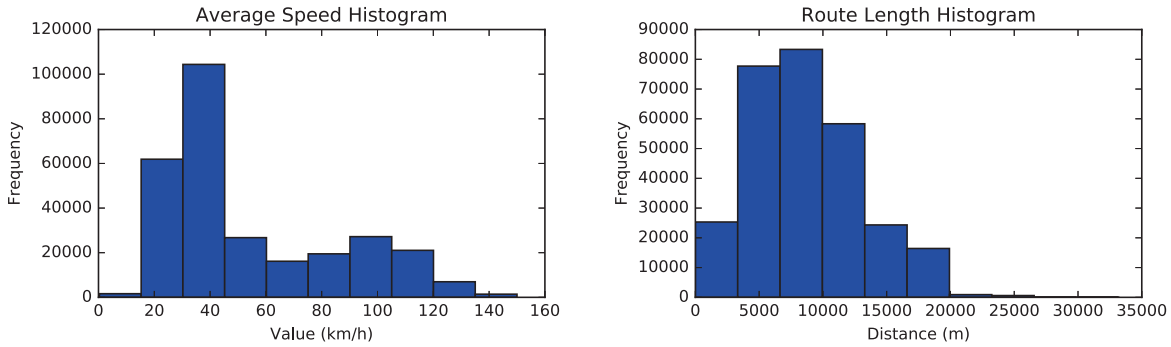


Figure 7.5: Example of traffic flow and occupancy from Luxembourg simulation

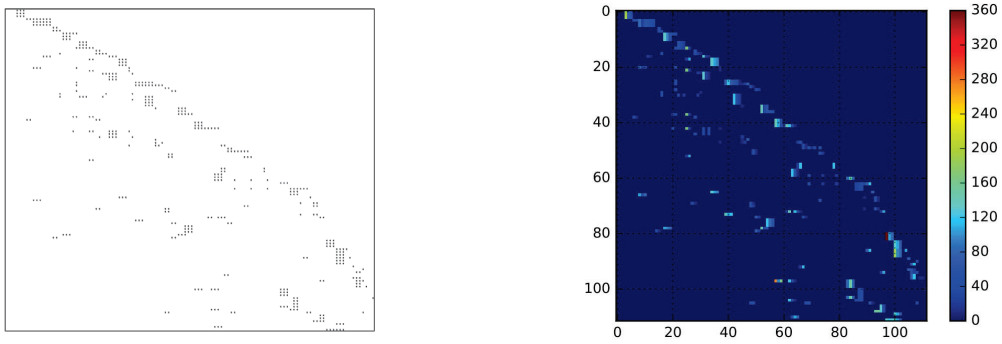
period and the balance of datasets, data from 7:00 to 19:55 are used in this study as the input for the proposed model. The input formatted with a connectivity matrix has been used for the proposed traffic anomaly detection model. Given the link flows on each edge and its connection with neighboured links in the network, a connectivity matrix could be generated. Figure 7.7



(a) Distribution of average traffic speed (b) Distribution of route length or travel distance

Figure 7.6: Validation of Luxembourg simulation output

shows an example of a converted geographical grid with traffic flow values.



(a) Connectivity Matrix (b) visualisation with traffic input

Figure 7.7: Model input for the proposed traffic anomaly detection method

7.3.3 Results and Analysis

The traffic data extracted from the Sioux-Falls and Luxembourg simulations have been used for early RC detection robustness analysis with respect to network topology for half an hour early prediction. In the conclusion to the previous Chapter 6, a time lag of three was recommended for RC detection, and thus this time lag has been applied in this section for both simulations. The output was labelled with EM algorithm. Both input and output have been formulated into a connectivity matrix for localisation, as configured in Chapter 6.

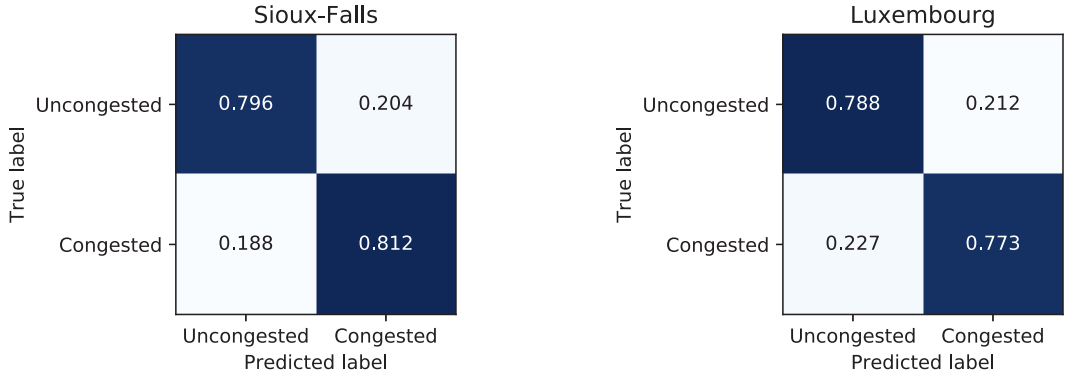
Table 7.3 shows the evaluation measurement for the proposed LSTM-CNN. In general, it is noted that results from both simulations are fairly accurate for 30 minutes ahead of detection in terms of all evaluation metrics. Typically, the results for the Sioux-Falls simulation case study

outperformed those for Luxembourg in terms of low FPR, high DR and a large percentage of F1 score.

Table 7.3: Early recurrent congestion detection results based on different simulation network

Network	DR	FPR	F1 score	Precision	AUC
Sioux-Falls	0.812	0.204	0.806	0.799	0.885
Luxembourg	0.773	0.212	0.779	0.785	0.865

Another visual way to interpret the results of the classification is via a confusion matrix for each method. Figure 7.8 shows the confusion matrix with normalisation by the size of classes (i.e., number of elements in each class) in order to have a more visual interpretation of which class is being misclassified in case of class imbalance. Specifically, the four values inside each matrix represent precision, false positive rate, false negative rate and recall from the top left to right bottom respectively. Both cases caused by either wrongly indicating a traffic anomaly or improperly identifying a normal instance, namely Type I Error and Type II Error in statistics, are presented in the confusion matrix clearly. It is worth mentioning that models with the lowest false positive rate or false negative rate do not necessarily have the best detection rate or precision out of all possible models, but they are important indicators of whether a model works effectively by making use of all available information in the data.



(a) Sioux Falls Simulation Network

(b) Luxembourg Simulation Network

Figure 7.8: Recurrent congestion detection based on different simulation networks

Figure 7.9 shows the ROC curve of the proposed early detection based on two simulation case studies. In this evaluation, the fact that both curves gather in the upper left corner suggests that the proposed model performs well in the classification of traffic anomalies, with low FPR and high true positive rate regardless of the network topology. The zoomed ROC curve shows

that notwithstanding the generally good performance of the proposed model in two case studies, the results based on the Sioux-Falls simulation study outperform than that of Luxembourg. The high performance of the proposed model can also be confirmed from the metrics (see Table 7.3).

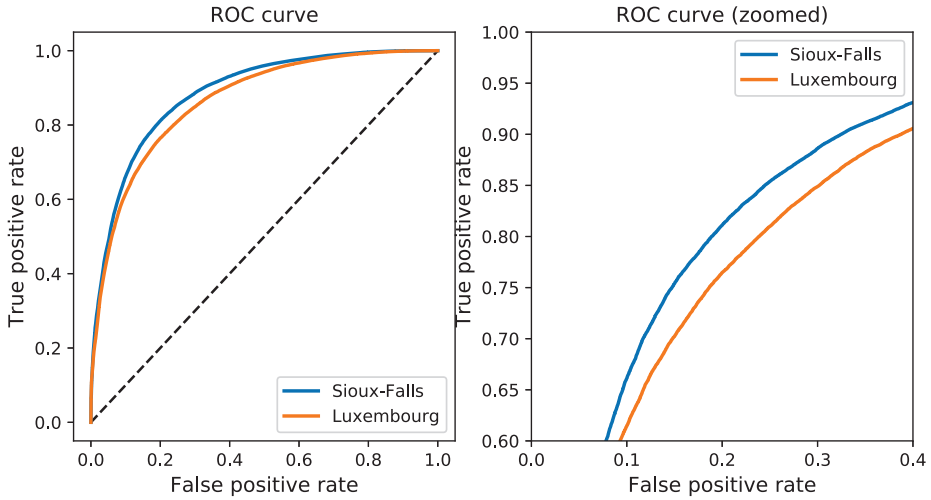


Figure 7.9: Recurrent congestion detection ROC curve for different simulation networks

7.3.4 Summary

The main contribution of this section is to test the proposed model in two network-level simulation case studies. Through a series of evaluations using data from well-calibrated simulation models of Sioux-Falls and the City of Luxembourg, the accuracy and scalability of the proposed method were demonstrated. Together with the real world result based on the Bath case study in Section 6.3.2, the LSTM-CNN algorithm is promising for large-scale traffic early recurrent congestion detection and thus deserves further investigation.

7.4 Robustness with Respect to Sensor Faults

All real traffic sensors are subject to a range of imperfections that degrade the quality of their outputs (Lee et al., 2010). Sensor Faults commonly exists in the ILD data (Luan et al., 2006; Lee et al., 2010). This section is aimed at investigating the impact of sensor faults on the performance of early RC detection.

7.4.1 Experimental Design

There are two types of sensor faults. The first type is caused by random errors, such as broken cables, communication failures and software errors (Luan et al., 2006). This type of sensor fault typically leads to missing data. In addition to these random errors, another type of errors is caused systematically. There are two forms of these systematic errors: (1) lower count bias and (2) higher count bias (Lee et al., 2010). For example, the lower count bias exists when two or more vehicles pass the ILD at the same time within the resolution frequency (i.e., 0.25 sec), in which case only one count will be measured by the detector. On the other hand, the higher count bias can occur for a cross-line ILD due to stationary or parking vehicles (Papageorgiou and Dinopoulou, 2003; Lee et al., 2010). The treatment of missing data and corrupted sensor faults is quite different, so these two types of faults will be introduced separately. This section will focus on the overestimated or underestimated sensor faults, while missing data will be discussed in the next section.

According to Lee et al. (2010), a single-lane ILD tends to underestimate the counts when the flow rate is high since vehicles are forced to shorten their headway compared to the ILD temporal resolution. This underestimation bias can be added to the simulation by randomly selecting points and reducing the values continuously. On the other hand, in the simulation, the temporal resolution is quite high, around 0.17 sec. This higher resolution and stationary vehicles might result in an overestimated count bias. This bias will be introduced in the simulation by adding more values to a period of time.

To set up a realistic plan, therefore, the overestimated bias sensor faults have been introduced randomly across roads in order to mimic stationary vehicles with an occupancy larger than 35% as taken in PeMS (Luan et al., 2006), i.e., within a duration of [0, 5 minutes] or [5, 10 min] or [10, 20 min] in the simulated data. On the other hand, the underestimated counts have been assigned with a random uniform sampling of network edges that are sources of traffic flow data detected to the simulation randomly across the edges randomly sampling from the top 5%, 10% and 15% flow rate in order to add bias to high flow rates (Roy and Saha, 2018) as these percentage ranges generally result in a short headway of less than 0.15 sec. since there is limited literature about the percentages of overestimated and underestimated sensor faults in ILD data, this research proposed three different levels of infections, i.e., 10%, 20% and 30%, while two

different biases are sampled randomly across the edges simultaneously.

7.4.2 Results and Analysis

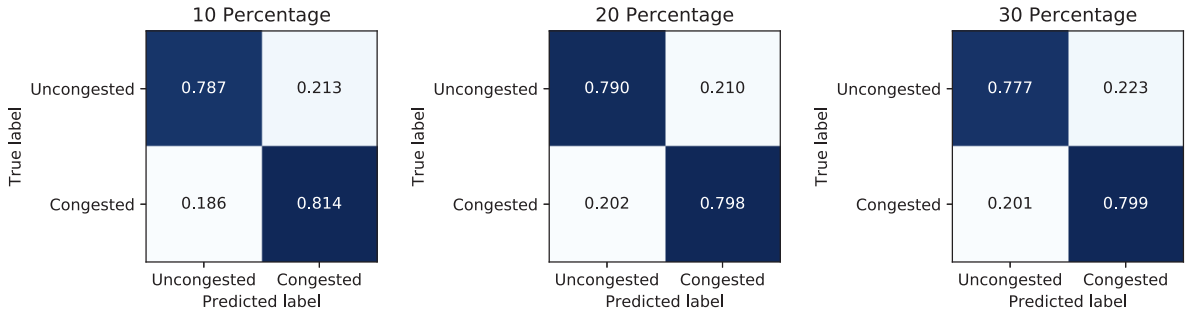
Table 7.4 presents the performance of the proposed model with different percentages of sensor faults. Generally, the performance is relatively stable in terms of different sensor fault scenarios for RC 30-min ahead prediction. More specifically, the performance starts to drop as the percentage of sensor faults increases from 0.814 DR to 0.799 DR. The FPR rates fluctuate around 0.2 and this flattened change indicates that the sensor faults have a marginal impact in terms of false alarm of recurrent congestion. Considering the magnitude of the performance change and percentage of sensor faults, it is obvious that the performance is not sensitive to the proportion of sensor faults. One possible reason is that after introducing the corrupted data, the labels of RC might be changed via the EM clustering.

Table 7.4: Early recurrent congestion detection results based on different percentage of sensor faults

Percentage of Missing	DR	FPR	F1 score	Precision	AUC
10 Percentage	0.814	0.213	0.803	0.791	0.881
20 Percentage	0.798	0.210	0.794	0.791	0.875
30 Percentage	0.799	0.223	0.790	0.782	0.870

Figure 7.10 shows the confusion matrix in order to have a more visual interpretation of which class is being misclassified in case of class imbalance. Specifically, one can see that the proposed classifier has a relatively small false positive rate and false negative rate based on different proportions of sensor faults, and this proves its ability in classifying different classes precisely, i.e, congestion and no congestion. It is worth mentioning that the performance with the lowest false positive rate or false negative rate does not necessarily have the best detection rate or precision out of all possible models, but they are important indicators of whether a model works effectively by making use of all available information in the data.

Figure 7.11 shows the ROC curve of the model performance with different levels of sensor faults. It is obvious from the zoomed ROC curve that the performance variation for three scenarios is similar with good performance no matter what threshold is set. It is notable that the model performs less well when there is a higher percentage of sensor faults.



(a) 10 Percentage (b) 20 Percentage (c) 30 Percentage

Figure 7.10: Recurrent congestion detection based on different percentages of sensor faults

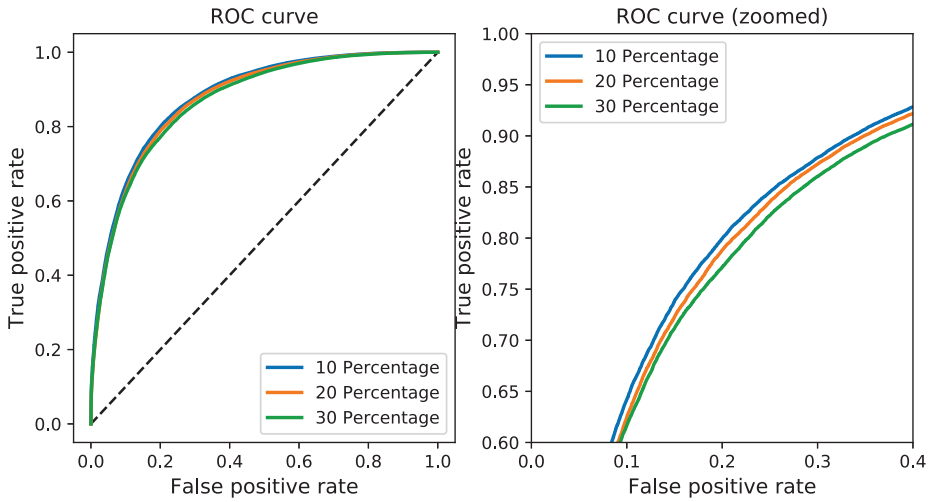


Figure 7.11: Recurrent congestion detection ROC curve for different percentages of sensor faults

7.4.3 Summary

This section demonstrates the robustness analysis of the proposed early RC detection model in the presence of overestimated and underestimated traffic values. The proposed model shows high reliability with different percentages of sensor faults where different scenarios have been set up in the experimental design.

7.5 Robustness with Respect to Missing Data

Missing data is a special type of sensor fault, different from the sensor faults that were introduced in the previous section in that this type of erroneous data have no input data rather than having overestimated or underestimated data. The phenomenon of missing data is very common for

ILD data for a variety of reasons including communication failures and maintenance issues (Robinson, 2005).

Many studies in the field of transport research have been devoted to robustness in respect to missing data and these tend to suggest that it is essential to check the model performance in the presence of missing data (Van Lint et al., 2005; Tian et al., 2018; Laña et al., 2018). This section presents the robustness analysis for the proposed early RC detection model in terms of different percentages of missing data.

7.5.1 Experimental Design

Generally, there are three different types of missing data, as shown in Figure 7.12. The first type of missing data is incidental failure. The main property of this is that it generally happens during a short duration without any continuous patterns either in time or in space. The second type is structural failure which might affect several neighbouring ILDs over a period of time, while the third type (intrinsic failure) basically has an impact on all ILDs during a period of time. All three types of missing data commonly exist in the real ILD data (Van Lint et al., 2005).

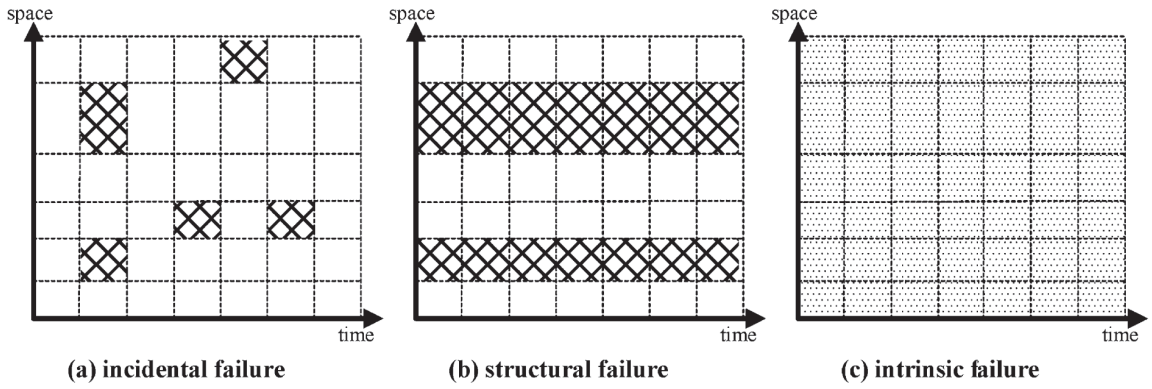


Figure 7.12: Classification of possible input failure (i.e. missing or unreliable data from traffic detectors) (Source: Van Lint et al., 2005)

To set up a realistic missing data scenario, missing data are assigned with a random uniform sampling of network edges that are sources of traffic flow data. Three different types of missing data, i.e., incidental failure, structural failure and intrinsic failure are randomly assigned with the total percentage of 10%, 20% and 30%. Specifically in SUMO simulation, Incidental failures were incidental points generated randomly across time and space. Structural failures were simulated in different random spaces with random period of time while intrinsic failure

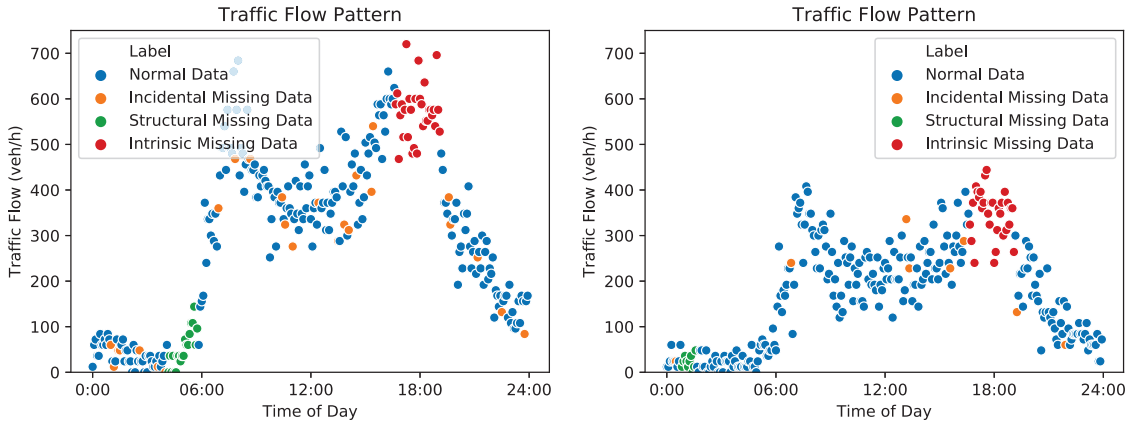


Figure 7.13: Examples of different missing data: example of two ILDs

Table 7.5: Early recurrent congestion detection results based on different percentage of missing data

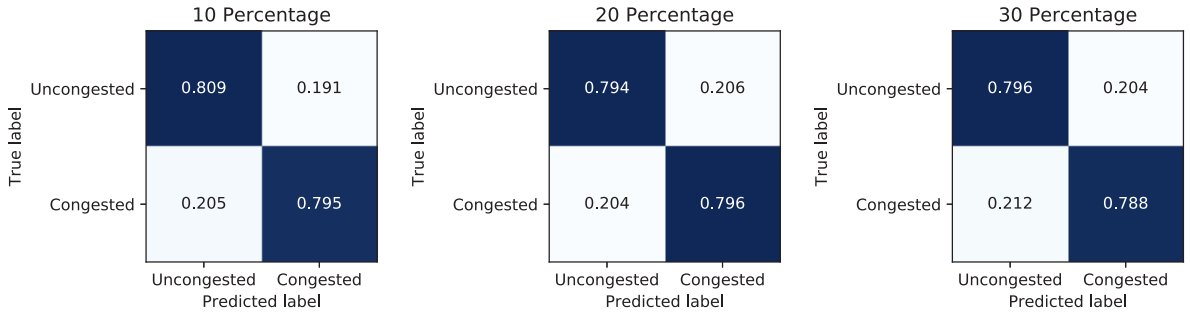
Percentage of Missing	DR	FPR	F1 score	Precision	AUC
10 Percentage	0.795	0.191	0.801	0.806	0.883
20 Percentage	0.796	0.206	0.795	0.795	0.875
30 Percentage	0.788	0.204	0.791	0.795	0.872

or systematic failure blocked a period of time for all detectors. An example of traffic data with different types of missing data has been shown in Figure 7.13.

7.5.2 Results and Analysis

Table 7.5 shows the results of model performance in terms of different percentages of missing data ranging from 10% to 30%. The detection rate generally decreases as the percentage of missing data increases. The magnitude of this drop, however, i.e., a 0.2% drop in DR on average for a 10% increase of missing data, is marginal compared to the extent to which missing data can increase without causing more false detection. This marginal drop can also be visualised from the confusion matrix in Figure 7.14.

Figure 7.15 shows the corresponding ROC curves with three different missing data scenarios. The steepness of the curve is relatively similar and this suggests that the model can predict with high true positive rates and low false positive rates with different percentages of missing data. According to the position of the curve, however, one can discriminate differences in the performance. The zoomed ROC curve shows the performance with 10% missing data is closer



(a) 10 Percentage (b) 20 Percentage (c) 30 Percentage

Figure 7.14: Recurrent congestion detection based on different percentage of missing data

to the left upper corner. The higher the proportion of missing data there is, the worse the model performance. It is worth noting that even though the performance drops as the proportion of missing data increases, the accuracy is with around a 0.78 detection rate and 0.2 FPR.

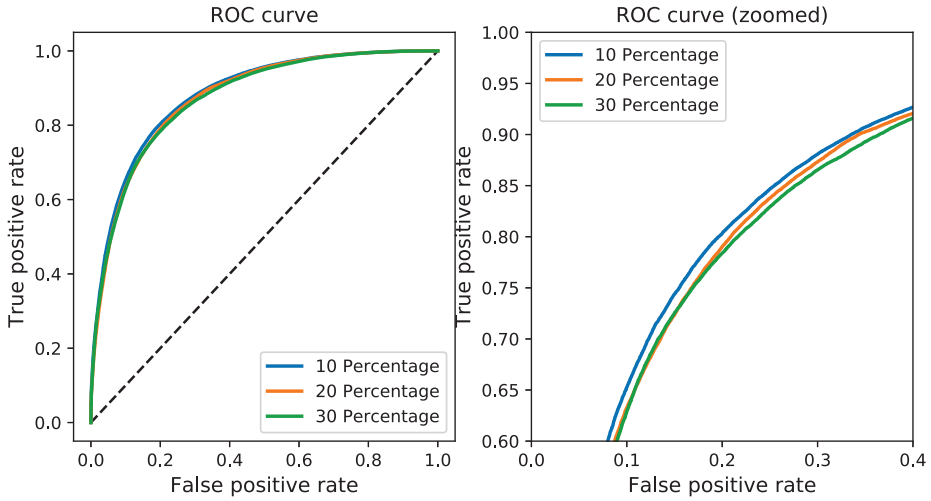


Figure 7.15: Recurrent congestion detection ROC curve for different percentages of missing data

Since the inputs are dependent in terms of both spatial and temporal dimension, single missing points or intervals of missing data represent redundancy of information and thus can be substituted with either the spatial or temporal neighbours during the neural learning process.

7.5.3 Summary

This section has investigated the robustness of the proposed traffic anomaly detection in terms of missing data. Three different proportions of mixed types of missing data have been designed

for the robustness analysis. Even though the accuracy of the model reduced as the proportions of missing data increases, it remained acceptable for 30 minutes ahead of detection.

7.6 Summary

This section has addressed the final objective of this research by investigating the robustness and transferability of the proposed model in terms of network topology, sensor faults and missing data. The proposed early RC detection model appears to be robust to corruption arising from these networks and data imputation schemes. Specifically, for different network topologies, with the total number of edges and ILD increased by around eight times, the performance only changed marginally with a 4% drop in DR, a 3% decrease in the F1 score and a 0.8% increase in the FPR for 30 minutes ahead RC detection. This marginal change with different network topologies indicates that the proposed model might be transferable into different traffic networks for early RC detection.

Regarding sensor faults and missing data, a similar marginal drop has also been suggested for both incidental (random) and structural or systemic input failure. Compared to the influence of missing data, however, the model is less sensitive to sensor faults, i.e., overestimated and underestimated counts.

The uniform good performance and marginal reduction when the model is challenged might originate from the simplicity of simulation in nature. When compared with previous real-world case studies, however, the detection framework yielded accurate and robust traffic anomaly detection no matter whether simulated data or real-world data was used.

Compared with traditional RC detection methods that use data from one or few adjacent detectors, the proposed method may not yield superior results. However, the proposed method has two advantages compared to traditional methods while maintaining reasonable detection accuracy. Firstly, since the proposed method uses network-wide data, the method can yield network-wide results even in the existence of missing data. Secondly, the proposed method is capable of predicting RC in a couple of time steps ahead with missing or faulty data which highlights its resilience of early detection function.

Chapter 8

Conclusion

Chapter 2 to Chapter 7 address each of the objectives set out at the beginning of this thesis. This chapter provides a summary by revisiting these objectives and emphasising the contributions of this research in Section 8.1. Apart from the conclusion and contributions, Section 8.2 discusses the limitations and suggests avenues for future research.

8.1 Revisiting the Objectives of this Research

The aim of this research was to develop novel traffic anomaly detection models that can incorporate the spatial and temporal traffic information to detect multiple traffic anomalies at the network level rather than one or several corridors. This spatio-temporal detection model is expected to be used for dynamic real-time analysis that will allow traffic managers to take prompt action to mitigate congestion and traffic incidents, reducing the late response caused by uncorrected alarms and further improving the accuracy and reliability of the urban road network. The research objectives to achieve this aim are presented in Section 1.3. The rest of this thesis addressed each of the listed objectives. This section revisits these objectives and summarises the relevant work and contributions.

(1) Understand traffic anomalies, the types of anomalies and their impact across an urban traffic network in order to identify the current gaps in literature about traffic anomaly detection and thus the relevant research challenges.

The first objective was to understand traffic anomalies, the types of anomalies and their corresponding distribution of impacts across an urban traffic network in order to identify the current research challenges and gaps in respect to traffic anomaly detection. This objective was addressed in Chapter 2.

Since there is no consensus on the definition of traffic anomalies and their detection. Section 2.1 presented an overview of general anomalies, types of traffic anomalies and anomaly detection. The definition of anomaly is heavily dependent on the field of application. A systematic understanding of general anomaly, however, helped to understand the concept of traffic anomalies and gave insight into traffic anomaly detection.

Section 2.2 classified traffic anomalies into two main types, Recurrent Congestion (RC) and Non-Recurrent Congestion (NRC). A comprehensive review was presented in terms of RC detection, early RC prediction and NRC detection. The difference between RC detection and NRC detection lies in three aspects: (1) whether or not contextual information is needed; (2) requirements of label generation; and (3) possibility of early prediction. Based on the review of existing research on traffic anomaly detection, two research gaps were identified: (1) improving the accuracy of RC and NRC detection by using spatio-temporal information across an entire traffic network; (2) investigating the possibility of detecting the onset of RC before its occurrence; and (3) investigating the robustness of techniques in terms of network topology and data quality.

Section 2.3 reviewed the four main types of detection algorithms, i.e., classification-based algorithm, clustering-based algorithms, statistical algorithms and information theory, as summarised in Section 2.3.1. Recent advances in deep learning, especially Convolutional Neural Networks (CNN) and Long Short Term Memory (LSTM), may provide opportunities to model the whole network taking into account the spatio-temporal information to predict RC as early as possible even with less than desired data quality.

(2) Develop a novel framework for multiple traffic incidents detection and early congestion prediction at an urban network level.

Chapter 3 focuses on the second objective, which is to develop a novel framework for traffic anomaly detection and early congestion prediction at a network level.

Section 3.1 defined the concept of traffic anomalies. Two detailed methodological frameworks for early prediction of RC and for NRC detection were presented in Section 3.2. The conceptual framework for traffic anomaly detection consists of three stages: (1) translation layers that transform the traffic data format into the desired input suitable for use by machine learning methods; (2) an anomaly detection model with the ability to make use of spatio-temporal information; and (3) localisation to locate the traffic anomaly precisely in the entire network. Specifically, Section 3.2.1 proposed three translation layer methods, i.e., connectivity matrix, geographical grid and spatio-temporal translation. The translation layers are then used as inputs for the detection model by making use of the recent advances in deep learning with the aim to detect RC and NRC early and precisely. CNN and LSTM were proposed as the anomaly detection models to capture the spatial and temporal information. Section 3.2.3 elaborated the methodology for localisation using iterative methods to locate traffic anomaly after the detection.

To evaluate the proposed detection framework, three representative methods in the literature, i.e., Multiple Layer Perceptron (MLP), Random Forest (RF) and Gradient Boosting Classifier (GBC), are selected as benchmarks. A set of comprehensive evaluation metrics including F1-score, detection rate, false positive rate, Receive Operating Characteristic (ROC) curve and computational time were presented to measure its accuracy in terms of different aspects.

(3) Develop a novel spatio-temporal model that can detect traffic anomalies and predict congestion at an early stage. Explore the application of deep learning techniques to anomaly detection and make use of advanced artificial intelligence and machine learning to develop a method to map the anomaly detection problem to deep learning.

The third objective was to develop a novel spatio-temporal model that can detect traffic anomaly and early congestion prediction and explore the application of deep learning techniques to anomaly detection. This leverages advanced artificial intelligence and machine learning and develops a method to map the anomaly detection problem to deep learning tools. This objective has been successfully addressed in Chapter 4 and Chapter 5.

Chapter 4 presented three translation layer methods, namely 1) connectivity matrix, 2) geographical grid and 3) spatio-temporal translation. These translation layers form the first processing layer in the framework that transforms the traffic data into the desired multi-dimensional format for further processing. Section 4.3 presented the evaluation results for the translation layers based on a case study using data from the City of Bath for the detection and early recurrent congestion prediction problems. The result suggested that the connectivity matrix would be recommended if a low false alarm rate and high precision are required for the early prediction task. RC generally affects the traffic upstream and downstream rather than whole areas or roads. The results also suggested that the accuracy for two-hour ahead prediction was still acceptable with a precision of 0.96 and with an FPR of 0.15. Meanwhile, the existence of a bowl shape when evaluating the performance with an increase of time lags indicates that adding unnecessary additional time lags as input to the model may introduce noise rather than useful features into the detection algorithm.

On the other hand, Section 4.4 showed the evaluation of the translation layer for NRC detection using a London case study. In contrast to the detection of recurrent congestion, where output labels are not available, additional traffic incident data obtained from Transport for London was used for incident detection. The results demonstrated that the performance of the geographical grid translation layer outperformed others in the sensitivity analysis based on all metrics regardless of the number of time lags used as explanatory variables.

Chapter 5 evaluated the proposed detection methods and compared these with the conventional machine learning methods (i.e., MLP, RF and GBC) with connectivity matrix translation and geographical translation selected as the translation layers for subsequent components of the RC and NRC traffic anomaly detection framework respectively. The hybrid method LSTM-CNN was used for the RC detection where the LSTM was applied to extract the temporal features while the CNN was formed to capture the spatial features in the traffic network. The results showed

that the proposed model gave reasonable early prediction in around two hours ahead (with a DR value of 0.972, FPR value of 0.002, etc), outperforming the baseline models in Section 5.3. The application of CNN to NRC detection given in Section 5.4 suggested that CNN was superior to the conventional machine learning methods to get the informative features for existing and emerging traffic incidents.

(4) Propose an approach for the localisation of traffic anomalies detected in the network in association with the proposed spatio-temporal anomaly detection.

The fourth objective was to propose an approach for localisation of traffic anomalies in a network in association with the proposed spatio-temporal anomaly detection at a network level. Chapter 6 focused on this objective by answering the question of localisation and evaluating the feasibility of locating traffic anomalies in the network.

Localisation is critical for a network-based model since it enables the model to predict multiple traffic anomalies that occur simultaneously across the traffic network. In order to have a better understanding of the localisation of traffic anomaly detection, Section 6.1 summarised a review of the relevant studies. Section 6.2 formulated six iterative methods ranging from simple average probability, conditional probability, index, logistic regression, random forest and gradient boosting classifier to predict and locate the traffic anomaly.

The key findings based on the evaluation and discussion in Section 6.3, Section 6.4 and Section 6.5 are summarised as follows: (1) random forest outperformed other models in terms of localising multiple traffic incidents at a network level based on the majority of metrics for RC and NRC detection; (2) the model is also capable of predicting the traffic states in two hours with a reasonable accuracy; and (3) visualisation can help to interpret the model's ability to capture the spatial correlations in the learning process.

(5) Investigate the effect of network topology, sensor faults and missing data on the accuracy of the network level traffic anomaly detection and early prediction.

The final objective in this research is to investigate the effect of network topology, sensor faults and missing data on the traffic anomaly detection and early prediction. Chapter 7 addressed this objective.

Section 7.1 reviewed the practical issues in terms of robustness with different network topologies, sensor faults and missing data caused by systematic errors. Section 7.3 tested the proposed traffic anomaly detection framework on two different simulation networks, i.e., the Sioux-Falls network and the Luxembourg traffic network. Section 7.4 and Section 7.5 further investigated the treatment with the involvement of sensor faults and missing data based on the Sioux-Falls network. The results suggested that the proposed model is capable of performing robustly with different network topologies, sensor faults and different levels of missing data.

8.2 Limitations and Future Research

A number of potential research topics related to the work in this research are recommended for future studies.

- For RC detection, the detection model is based on the congestion labels generated by the Expectation Maximisation (EM) algorithm since no traffic congestion states are available for direct use for the RC detection. This research assumed that the labels obtained from the EM algorithm represented the ground truth. Even though EM algorithms have been shown to be effective and reliable to classify different traffic states in many studies (Han et al., 2010; Zhu et al., 2018b), they still only approximate to true labels. Given the fact that it is hard to obtain true labels, one possible direction that might be of interest is to compare different clustering methods, such as nearest neighbours and k-means, systematically in order to group recurrent congestion states. Additionally, given the scope of this research, RC detection based on multiple traffic states has not been systematically tested in this thesis, with a simple binary classification being used instead. It is feasible to identify multiple traffic states, since the severity of levels of congestion is more informative for the end user and traffic managers.

- The traffic anomaly detection model proposed in this research is based on deep learning methods and thus belongs to the class of data-driven models. These have many advan-

tages (i.e., high accuracy and ease of implementation) but data-driven models have in common that they correlate the traffic conditions to current and past traffic data without explicitly addressing the formulation of these traffic conditions. This property leads to the limitation that they are location-specific, relying on specific input, model architecture and evaluation (Van Phuong et al., 2006). One possible solution to deal with this limitation for future research is to combine data-driven methods with the advantages of model-based approaches, such as fundamental traffic flow diagrams (Siebel and Mauser, 2006) and traffic flow models (Gerlough and Huber, 1976).

- This research mapped the relationship between traffic data (i.e., traffic flow and occupancy) and traffic anomalies. Given the vast volume of different data available online, data sourced from other feeds might be valuable for traffic anomaly detection. The possible sources include weather data, social media data (i.e., Twitter and Facebook), incident feeds and traffic data from other sensors such as probe vehicles and cameras. If more reliable data are available, data fusion for traffic anomaly detection could be promising to improve the accuracy of the detection and localisation. For example, the localisation method used in this research is an iterative method based on the detection results from row and column level. If higher resolution input data are given, such as 1-minute traffic data from probe vehicles, this iterative method might flexibly scale to provide a the precise location.

- Furthermore, it would be of immense practical value to extend the method to distinguish RC from other common anomalies such as sensor faults and NRC. With multiple reliable data resources, this accurate distinction would be a promising way to reduce the chances of false alarm significantly.

- This research applied deep learning techniques and customised them into traffic anomaly detection in the light of the proposed conceptual framework. The application of this conceptual framework could be extendable to short-term prediction problems that are inherently supervised learning problems with adjusted evaluation metrics such as mean absolute percentage error, mean squared error.

- The new ITS generation relies on vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), infrastructure-to-vehicle (I2V) and infrastructure-to-infrastructure (I2I) communication

(Alam et al., 2016). These new technologies added to the transportation infrastructure and vehicles are progressively revolutionising way that we travel. The detection framework proposed in this research might be adapted to take advantage of such technology developments for detecting traffic anomalies.

Bibliography

Agarwal, D. (2005). An empirical bayes approach to detect anomalies in dynamic multidimensional arrays. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 8–pp. IEEE.

Agovic, A., Banerjee, A., Ganguly, A., and Protopopescu, V. (2009). Anomaly detection using manifold embedding and its applications in transportation corridors. *Intelligent Data Analysis*, 13(3):435–455.

Agrawal, S. and Agrawal, J. (2015). Survey on anomaly detection using data mining techniques. *Procedia Computer Science*, 60:708–713.

Ahmed, M. and Abdel-Aty, M. (2013). A data fusion framework for real-time risk assessment on freeways. *Transportation Research Part C: Emerging Technologies*, 26:203–213.

Ahmed, M., Mahmood, A. N., and Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19–31.

Alam, M., Ferreira, J., and Fonseca, J. (2016). Introduction to intelligent transportation systems. In *Intelligent Transportation Systems*, pages 1–17. Springer.

Ali, M., Bhuiyan, R. H., Dougal, R., and Alam, M. N. (2015). Non-intrusive cable fault detection and methods. US Patent 9,103,864.

Amini, A., Vaghefi, R. M., Jesus, M., and Buehrer, R. M. (2014). Improving gps-based vehicle positioning for intelligent transportation systems. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 1023–1029. IEEE.

An, S., Yang, H., Wang, J., Cui, N., and Cui, J. (2016). Mining urban recurrent congestion evolution patterns from GPS-equipped vehicle mobility data. *Information Sciences*, 373:515–526.

Anbaroglu, B., Heydecker, B., and Cheng, T. (2014a). Spatio-temporal clustering for non-recurrent traffic congestion detection on urban road networks. *Transportation Research Part C: Emerging Technologies*, 48:47–65.

Anbaroglu, B., Heydecker, B., and Cheng, T. (2014b). Spatio-temporal clustering for non-recurrent traffic congestion detection on urban road networks. *Transportation Research Part C: Emerging Technologies*, 48:47–65.

Andrews Sobral, L. O., Schnitman, L., and De Souza, F. (2013). Highway traffic congestion classification using holistic properties. In *10th IASTED International Conference on Signal Processing, Pattern Recognition and Applications*.

Arora, P. and Srivastava, S. (2015). Gait recognition using gait gaussian image. In *Signal Processing and Integrated Networks (SPIN), 2015 2nd International Conference on*, pages 791–794. IEEE.

Asakura, Y., Kusakabe, T., Nguyen, L. X., and Ushiki, T. (2017). Incident detection methods using probe vehicles with on-board gps equipment. *Transportation Research Part C: Emerging Technologies*, 81:330–341.

Auria, L. and Moro, R. (2007). Advantages and disadvantages of support vector machines. *Credit Risk Assessment Revisited: Methodological Issues and Practical Implications*, pages 49–68.

Azevedo, C. L., Deshmukh, N. M., Marimuthu, B., Oh, S., Marczuk, K., Soh, H., Basak, K., Toledo, T., Peh, L.-S., and Ben-Akiva, M. E. (2017). Simmobility short-term: An integrated microscopic mobility simulator. *Transportation Research Record: Journal of the Transportation Research Board*, 2622:13–23.

Baiocchi, A., Cuomo, F., De Felice, M., and Fusco, G. (2015). Vehicular ad-hoc networks sampling protocols for traffic monitoring and incident detection in intelligent transportation systems. *Transportation Research Part C: Emerging Technologies*, 56:177–194.

Bao, J., Liu, P., and Ukkusuri, S. V. (2019). A spatiotemporal deep learning approach for city-wide short-term crash risk prediction with multi-source data. *Accident Analysis & Prevention*, 122:239–254.

Bedogni, L., Gramaglia, M., Vesco, A., Fiore, M., Härri, J., and Ferrero, F. (2015). The bologna ringway dataset: improving road network conversion in sumo and validating urban mobility via navigation services. *IEEE Transactions on Vehicular Technology*, 64(12):5464–5476.

Bengio, Y. et al. (2009). Learning deep architectures for ai. *Foundations and trends in Machine Learning*, 2(1):1–127.

Bereziński, P., Jasiul, B., and Szpyrka, M. (2015). An entropy-based network anomaly detection method. *Entropy*, 17(4):2367–2408.

Bhuyan, M. H., Bhattacharyya, D., and Kalita, J. K. (2016). A multi-step outlier-based anomaly detection approach to network-wide traffic. *Information Sciences*, 348:243–271.

Bickel, P. J., Holm, S., Rosén, B., Spjøtvoll, E., Lauritzen, S., Johansen, S., and Barndorff-Nielsen, O. (1976). Another look at robustness: a review of reviews and some new developments [with discussion and reply]. *Scandinavian Journal of Statistics*, pages 145–168.

Black, C., Chevallier, O. P., Haughey, S. A., Balog, J., Stead, S., Pringle, S. D., Riina, M. V., Martucci, F., Acutis, P. L., Morris, M., et al. (2017). A real time metabolomic profiling approach to detecting fish fraud using rapid evaporative ionisation mass spectrometry. *Metabolomics*, 13(12):153.

Bostani, H. and Sheikhan, M. (2017). Modification of supervised opf-based intrusion detection systems using unsupervised learning and social network concept. *Pattern Recognition*, 62:56–72.

Branch, J. W., Giannella, C., Szymanski, B., Wolff, R., and Kargupta, H. (2013). In-network outlier detection in wireless sensor networks. *Knowledge and Information Systems*, 34(1):23–54.

Bratan, F., Niaf, E., Melodelima, C., Chesnais, A. L., Souchon, R., Mège-Lechevallier, F., Colombel, M., and Rouvière, O. (2013). Influence of imaging and histological factors on prostate cancer detection and localisation on multiparametric MRI: a prospective study. *European Radiology*, 23(7):2019–2029.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Cai, P., Wang, Y., Lu, G., Chen, P., Ding, C., and Sun, J. (2016). A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. *Transportation Research Part C: Emerging Technologies*, 62:21–34.

Cai, Y.-D. and Chou, K.-C. (2003). Nearest neighbour algorithm for predicting protein subcellular location by combining functional domain composition and pseudo-amino acid composition. *Biochemical and Biophysical Research Communications*, 305(2):407–411.

Candy, J. V. (2016). *Bayesian signal processing: classical, modern, and particle filtering methods*, volume 54. John Wiley & Sons.

Cascetta, E. (2013). *Transportation systems engineering: theory and methods*, volume 49. Springer Science & Business Media.

CEBR (2014). The future economic and environmental costs of gridlock in 2030. Technical report, the Centre for Economics and Business Research.

Chandel, K., Kunwar, V., Sabitha, S., Choudhury, T., and Mukherjee, S. (2016). A comparative study on thyroid disease detection using k-nearest neighbor and naive bayes classification techniques. *CSI Transactions on ICT*, 4(2-4):313–319.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15.

Chatzigiannakis, V., Papavassiliou, S., Grammatikou, M., and Maglaris, B. (2006). Hierarchical anomaly detection in distributed large-scale sensor networks. In *11th IEEE Symposium on Computers and Communications (ISCC'06)*, pages 761–767. IEEE.

Chen, D., Shao, X., Hu, B., and Su, Q. (2005). Simultaneous wavelength selection and outlier detection in multivariate regression of near-infrared spectra. *Analytical Sciences*, 21(2):161–166.

Chen, F.-C. and Jahanshahi, M. R. (2018). Nb-cnn: deep learning-based crack detection using convolutional neural network and naive bayes data fusion. *IEEE Transactions on Industrial Electronics*, 65(5):4392–4400.

Chen, H., Grant-Muller, S., Mussone, L., and Montgomery, F. (2001). A study of hybrid neural network approaches and the effects of missing data on traffic forecasting. *Neural Computing & Applications*, 10(3):277–286.

Chen, S.-J., Zhan, T.-S., Huang, C.-H., Chen, J.-L., and Lin, C.-H. (2015). Nontechnical loss and outage detection using fractional-order self-synchronization error-based fuzzy petri nets in micro-distribution systems. *IEEE Transactions on Smart Grid*, 6(1):411–420.

Chine, W., Mellit, A., Lughi, V., Malek, A., Sulligoi, G., and Pavan, A. M. (2016). A novel fault diagnosis technique for photovoltaic systems based on artificial neural networks. *Renewable Energy*, 90:501–512.

Chowdhary, C. L. and Acharjya, D. (2016). A hybrid scheme for breast cancer detection using intuitionistic fuzzy rough set technique. *International Journal of Healthcare Information Systems and Informatics (IJHISI)*, 11(2):38–61.

Clausius, R. (1867). *The mechanical theory of heat: with its applications to the steam-engine and to the physical properties of bodies*. J. van Voorst.

Codeca, L., Frank, R., and Engel, T. (2015). Luxembourg sumo traffic (lust) scenario: 24 hours of mobility for vehicular networking research. In *2015 IEEE Vehicular Networking Conference (VNC)*, pages 1–8. IEEE.

Codecá, L., Frank, R., Faye, S., and Engel, T. (2017). Luxembourg SUMO Traffic (LuST) Scenario: Traffic Demand Evaluation. *IEEE Intelligent Transportation Systems Magazine*, 9(2):52–63.

Coppolino, L., D'Antonio, S., Formicola, V., Massei, C., and Romano, L. (2015). Use of the dempster-shafer theory for fraud detection: the mobile money transfer case study. In *Intelligent Distributed Computing VIII*, pages 465–474. Springer.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

Cortez, P., Rio, M., Rocha, M., and Sousa, P. (2012). Multi-scale internet traffic forecasting using neural networks and time series methods. *Expert Systems*, 29(2):143–155.

Cui, Z., Henrickson, K., Ke, R., and Wang, Y. (2018). Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *arXiv preprint arXiv:1802.07007*.

Dabiri, S. and Heaslip, K. (2018). Inferring transportation modes from GPS trajectories using a convolutional neural network. *Transportation Research Part C: Emerging Technologies*, 86:360–371.

Dasarathy, B. V. (1991). Nearest neighbor NN norms:NN pattern classification techniques.

Davy, M. and Godsill, S. (2002). Detection of abrupt spectral changes using support vector machines. an application to audio signal segmentation. In *ICASSP*, volume 2, pages 1313–1316.

De Fenza, A., Sorrentino, A., and Vitiello, P. (2015). Application of artificial neural networks and probability ellipse methods for damage detection using lamb waves. *Composite Structures*, 133:390–403.

De Stefano, C., Sansone, C., and Vento, M. (2000). To reject or not to reject: that is the question-an answer in case of neural classifiers. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 30(1):84–94.

Deniz, O. and Celikoglu, H. B. (2011). Overview to some existing incident detection algorithms: a comparative evaluation. *Procedia Social and Behavioral Sciences*, pages 1–13.

Dogru, N. and Subasi, A. (2018). Traffic accident detection using random forest classifier. In *Learning and Technology Conference (L&T), 2018 15th*, pages 40–45. IEEE.

Dreżewski, R., Sepielak, J., and Filipkowski, W. (2015). The application of social network analysis algorithms in a system supporting money laundering detection. *Information Sciences*, 295:18–32.

Du, B., Peng, H., Wang, S., Bhuiyan, M. Z. A., Wang, L., Gong, Q., Liu, L., and Li, J. (2019). Deep irregular convolutional residual lstm for urban traffic passenger flows prediction. *IEEE Transactions on Intelligent Transportation Systems*.

Edgeworth, F. (1887). Xli. on discordant observations. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 23(143):364–375.

Emmerink, R. H., Axhausen, K. W., Nijkamp, P., and Rietveld, P. (1995). The potential of information provision in a simulated road transport network with non-recurrent congestion. *Transportation Research Part C: Emerging Technologies*, 3(5):293–309.

Eskin, E. (2000). Anomaly detection over noisy data using learned probability distributions. In *In Proceedings of the International Conference on Machine Learning*. Citeseer.

Fang, S.-H., Fei, Y.-X., Xu, Z., and Tsao, Y. (2017). Learning transportation modes from smartphone sensors based on deep neural network. *IEEE Sensors Journal*, 17(18):6111–6118.

Farabet, C., Couprie, C., Najman, L., and LeCun, Y. (2013). Learning hierarchical features for scene labeling. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 35(8):1915–1929.

Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874.

Fawcett, T. and Provost, F. (1999). Activity monitoring: Noticing interesting changes in behavior. In *Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 53–62. ACM.

Fernández-Gavilanes, M., Álvarez-López, T., Juncal-Martínez, J., Costa-Montenegro, E., and González-Castaño, F. J. (2016). Unsupervised method for sentiment analysis in online texts. *Expert Systems with Applications*, 58:57–75.

Fox, D., Burgard, W., Dellaert, F., and Thrun, S. (1999). Monte carlo localization: Efficient position estimation for mobile robots. *AAAI/IAAI*, 1999(343-349):2–2.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232.

Fu, K., Cheng, D., Tu, Y., and Zhang, L. (2016). Credit card fraud detection using convolutional neural networks. In *International Conference on Neural Information Processing*, pages 483–490. Springer.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202.

Geraci, A., Katki, F., McMonegal, L., Meyer, B., Lane, J., Wilson, P., Radatz, J., Yee, M., Porteous, H., and Springsteel, F. (1991). *IEEE standard computer dictionary: Compilation of IEEE standard computer glossaries*. IEEE Press.

Gerlough, D. and Huber, M. (1975). Traffic flow theory, trb special report 165. *Transportation Research Board, Washington, DC*, 36.

Gerlough, D. L. and Huber, M. J. (1976). Traffic flow theory. *Transportation Research Board*, 165.

Geroliminis, N. and Daganzo, C. F. (2008). Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings. *Transportation Research Part B: Methodological*, 42(9):759–770.

Gers, F. (2001). Long short-term memory in recurrent neural networks. *PhD dissertation, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland*.

Gu, Y., Qian, Z. S., and Chen, F. (2016). From twitter to detector: Real-time traffic incident detection using social media data. *Transportation Research Part C: Emerging Technologies*, 67:321–342.

Habtemichael, F. G., Cetin, M., and Anuar, K. A. (2015). Incident-induced delays on freeways: quantification method by grouping similar traffic patterns. *Transportation Research Record: Journal of the Transportation Research Board*, 2484:60–69.

Haddadin, S., De Luca, A., and Albu-Schäffer, A. (2017). Robot collisions: A survey on detection, isolation, and identification. *IEEE Transactions on Robotics*, 33(6):1292–1312.

Han, J., Krishnan, R., Polak, J. W., and Barria, J. (2010). A New Method for Probabilistic Traffic State Identification Using Loop Detector Data: Theory and Empirical Results. In *the Universities' Transport Study Group (UTSG)*, Plymouth, UK.

Haque, S. A., Rahman, M., and Aziz, S. M. (2015). Sensor anomaly detection in wireless sensor networks for healthcare. *Sensors*, 15(4):8764–8786.

Harrou, F., Kadri, F., Chaabane, S., Tahon, C., and Sun, Y. (2015). Improved principal component analysis for anomaly detection: Application to an emergency department. *Computers & Industrial Engineering*, 88:63–77.

Hawas, Y. E. (2007). A fuzzy-based system for incident detection in urban street networks. *Transportation Research Part C: Emerging Technologies*, 15(2):69–95.

Hawkins, D. M. (1980). *Identification of outliers*, volume 11. Springer.

Hawkins, D. M. (2004). The problem of overfitting. *Journal of Chemical Information and Computer Sciences*, 44(1):1–12.

Haykin, S. (1994). *Neural networks: a comprehensive foundation*. Prentice Hall PTR.

He, F., Yan, X., Liu, Y., and Ma, L. (2016a). A traffic congestion assessment method for urban road networks based on speed performance index. *Procedia Engineering*, 137:425–433.

He, Q., Kamarianakis, Y., Jintanakul, K., and Wynter, L. (2013). Incident duration prediction with hybrid tree-based quantile regression. In *Advances in Dynamic Network Modeling in Complex Transportation Systems*, pages 287–305. Springer.

He, T., Huang, W., Qiao, Y., and Yao, J. (2016b). Text-attentional convolutional neural network for scene text detection. *IEEE Transactions on Image Processing*, 25(6):2529–2541.

Hellinga, B. and Knapp, G. (2000). Automatic vehicle identification technology-based freeway incident detection. *Transportation Research Record: Journal of the Transportation Research Board*, pages 142–153.

Herrera-Semenets, V., Pérez-García, O. A., Hernández-León, R., van den Berg, J., and Doerr, C. (2018). A data reduction strategy and its application on scan and backscatter detection using rule-based classifiers. *Expert Systems with Applications*, 95:272–279.

Highways Agency (2005). Variable message sign. Technical report, Highway Agency.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. (2012a). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012b). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Hochreiter, S. and Urgen Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Hojati, A. T., Charles, P., Ferreira, L., and bin Kabit, M. R. (2011). *An analysis of traffic incidents on an Australian urban road network*. Universiti Malaysia Sarawak.

Hollier, G. and Austin, J. (2002). Novelty detection for strain-gauge degradation using maximally correlated components. In *ESANN*, pages 257–262.

Houbraken, M., Audenaert, P., Colle, D., Pickavet, M., Scheerlinck, K., Yperman, I., and Logghe, S. (2015). Real-time traffic monitoring by fusing floating car data with stationary detector data. In *Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2015 International Conference on*, pages 127–131. IEEE.

Huang, W., Song, G., Hong, H., and Xie, K. (2014). Deep architecture for traffic flow prediction: deep belief networks with multitask learning. *IEEE Transactions on Intelligent Transportation Systems*, 15(5):2191–2201.

Hunt, P., Robertson, D., Bretherton, R., and Winton, R. (1981). SCOOT-a traffic responsive method of coordinating signals. *TRRL Report LR*, 1014:l93l.

Ikhrata, H. and Michell, P. (1997). Technical report of southern california association of governments' transportation performance indicators. *Transportation Research Record: Journal of the Transportation Research Board*, pages 103–114.

Jiang, B. and Claramunt, C. (2004). Topological analysis of urban street networks. *Environment and Planning B: Planning and Design*, 31(1):151–162.

Kabir, M. R., Onik, A. R., and Samad, T. (2017). A network intrusion detection framework based on bayesian network using wrapper approach. *International Journal of Computer Applications*, 166(4):13–17.

Karami, M. and Wang, L. (2018). Fault detection and diagnosis for nonlinear systems: A new adaptive gaussian mixture modeling approach. *Energy and Buildings*, 166:477–488.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Li, F. F. (2014). Large-scale video classification with convolutional neural networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1725–1732.

Ke, R., Li, W., Cui, Z., and Wang, Y. (2019). Two-stream multi-channel convolutional neural network (tm-cnn) for multi-lane traffic speed prediction considering traffic volume impact. *arXiv preprint arXiv:1903.01678*.

Kerner, B. S. and Rehborn, H. (1996). Experimental properties of complexity in traffic flow. *Physical Review E*, 53(5):R4275.

Ki, Y.-K., Heo, N.-W., Choi, J.-W., Ahn, G.-H., and Park, K.-S. (2018). An incident detection algorithm using artificial neural networks and traffic information. In *Cybernetics & Informatics (K&I), 2018*, pages 1–5. IEEE.

Kim, Y., Wang, P., Zhu, Y., and Mihaylova, L. (2018). A capsule network for traffic speed prediction in complex road networks. In *2018 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–6. IEEE.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization.

Kinoshita, A., Takasu, A., and Adachi, J. (2015). Real-time traffic incident detection using a probabilistic topic model. *Information Systems*, 54:169–188.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th international joint conference on Artificial intelligence-Volume 2*, pages 1137–1143. Morgan Kaufmann Publishers Inc.

Kong, X., Xu, Z., Shen, G., Wang, J., Yang, Q., and Zhang, B. (2016a). Urban traffic congestion estimation and prediction based on floating car trajectory data. *Future Generation Computer Systems*, 61:97–107.

Kong, X., Xu, Z., Shen, G., Wang, J., Yang, Q., and Zhang, B. (2016b). Urban traffic congestion estimation and prediction based on floating car trajectory data. *Future Generation Computer Systems*, 61:97–107.

Krajzewicz, D., Erdmann, J., Behrisch, M., and Bieker, L. (2012). Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal on Advances in Systems and Measurements*, 5(3&4):128–138.

Krajzewicz, D., Hertkorn, G., Rössel, C., and Wagner, P. (2002). Sumo (simulation of urban mobility)-an open-source traffic simulation. In *Proceedings of the 4th Middle East Symposium on Simulation and Modelling (MESM20002)*, pages 183–187.

Krishnan, R. (2008). *Travel time estimation and forecasting on urban roads*. PhD thesis, Imperial College London.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Kumari, R., Singh, M., Jha, R., Singh, N., et al. (2016). Anomaly detection in network traffic using k-mean clustering. In *Recent Advances in Information Technology (RAIT), 2016 3rd International Conference on*, pages 387–393. IEEE.

Kusetogullari, H., Grahn, H., and Lavesson, N. (2016). Handwriting image enhancement using local learning windowing, gaussian mixture model and k-means clustering. In *Signal Processing and Information Technology (ISSPIT), 2016 IEEE International Symposium on*, pages 305–310. IEEE.

Laña, I., Olabarrieta, I. I., Vélez, M., and Del Ser, J. (2018). On the imputation of missing data for road traffic forecasting: New insights and novel techniques. *Transportation Research Part C: Emerging Technologies*, 90:18–33.

LeBlanc, L. J. (1975). An algorithm for the discrete network design problem. *Transportation Science*, 9(3):183–199.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lee, T.-C., Krishnan, R., and Polak, J. (2010). Data blurring: modelling the imperfections of traffic sensor data in a microscopic traffic simulator. In *the 42nd Annual Conference of the Universities' Transport Study Group (UTSG)*.

Leite, V. C., da Silva, J. G. B., Veloso, G. F. C., da Silva, L. E. B., Lambert-Torres, G., Bonaldi, E. L., and de Oliveira, L. E. d. L. (2015). Detection of localized bearing faults in induction

machines by spectral kurtosis and envelope analysis of stator current. *IEEE Transactions on Industrial Electronics*, 62(3):1855–1865.

Li, C., Wang, H., Zhang, Z., Sun, A., and Ma, Z. (2016). Topic modeling for short texts with auxiliary word embeddings. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 165–174. ACM.

Li, W., Mahadevan, V., and Vasconcelos, N. (2013). Anomaly detection and localization in crowded scenes. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 36(1):18–32.

Lin, W.-C., Ke, S.-W., and Tsai, C.-F. (2015). Cann: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-Based Systems*, 78:13–21.

Liu, Q., Lu, J., and Chen, S. (2013). Traffic incident detection using random forest. In *Transportation Research Board 92nd Annual Meeting*, pages 13–1610, Washington, DC.

Liu, Y., Feng, X., Wang, Q., Zhang, H., and Wang, X. (2014). Prediction of urban road congestion using a bayesian network approach. *Procedia-Social and Behavioral Sciences*, 138:671–678.

Lomax, T. and Margiotta, R. (2003). *Selecting travel reliability measures*. the Institute.

Lorente, D., Zude, M., Idler, C., Gómez-Sanchis, J., and Blasco, J. (2015). Laser-light backscattering imaging for early decay detection in citrus fruit using both a statistical and a physical model. *Journal of Food Engineering*, 154:76–85.

Lu, H.-p., Sun, Z.-y., and Qu, W.-c. (2015). Big data-driven based real-time traffic flow state identification and prediction. *Discrete Dynamics in Nature and Society*, 2015.

Lu, J., Chen, S., Wang, W., and van Zuylen, H. (2012). A hybrid model of partial least squares and neural network for traffic incident detection. *Expert Systems with Applications*, 39(5):4775–4784.

Luan, J., Guo, F., Polak, J., Hoose, N., and Krishnan, R. (2006). Inductive loop detector data cleaning treatments and their effect on performance of urban link travel time models. In *Transportation Research Board 85th Annual Meeting*, pages 06–0989, Washington, DC.

Luan, J., Guo, F., Polak, J., Hoose, N., and Krishnan, R. (2018). Investigating the Transferability of Machine Learning Methods in Short-Term Travel Time Prediction. In *Transportation Research Board 97th Annual Meeting*, pages 18–02742, Washington, DC.

Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., Zhao, X., and Kim, T.-K. (2014). Multiple Object Tracking: A Literature Review. *arXiv e-prints*, page arXiv:1409.7618.

Lv, Y., Duan, Y., Kang, W., Li, Z., and Wang, F.-Y. (2014). Traffic flow prediction with big data: a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):865–873.

Ma, T., Wang, F., Cheng, J., Yu, Y., and Chen, X. (2016). A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks. *Sensors*, 16(10):1701.

Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., and Wang, Y. (2017). Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, 17(4):818.

Ma, X., Tao, Z., Wang, Y., Yu, H., and Wang, Y. (2015a). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, 54:187–197.

Ma, X., Yu, H., Wang, Y., and Wang, Y. (2015b). Large-scale transportation network congestion evolution prediction using deep learning theory. *PloS one*, 10(3):e0119044.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1(14):281–297.

Manogaran, G., Vijayakumar, V., Varatharajan, R., Kumar, P. M., Sundarasekar, R., and Hsu, C.-H. (2018). Machine learning based big data processing framework for cancer diagnosis using hidden markov model and gm clustering. *Wireless Personal Communications*, 102(3):2099–2116.

Manual, H. C. (1965). Special report 87. *TRB, National Research Council, Washington, DC*, 397.

Manual, H. C. et al. (2000). Transportation research board. *National Research Council, Washington, DC*, 113:10.

Mazzenga, N. and Demetsky, M. (2010). Mitigation of Recurring Congestion on Freeways. In *Transportation Research Board 89th Annual Meeting*, volume 434, page 16.

Mirza, A. H. and Cosan, S. (2018). Computer network intrusion detection using sequential lstm neural networks autoencoders. In *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. IEEE.

Mitchell, R. and Chen, R. (2015). Behavior rule specification-based intrusion detection for safety critical medical cyber physical systems. *IEEE Transactions on Dependable and Secure Computing*, 12(1):16–30.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529.

Mohan, D. (2009). Intelligent transportation systems (its) and the transportation system. *Information Technology and Communications Resources for Sustainable Development*, pages 1–15.

Moustafa, N., Creech, G., and Slay, J. (2018). Anomaly detection system using beta mixture models and outlier detection. In *Progress in Computing, Analytics and Networking*, pages 125–135. Springer.

Moustafa, N. and Slay, J. (2016). The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set. *Information Security Journal: A Global Perspective*, 25(1-3):18–31.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*, pages 807–814.

Nantes, A., Ngoduy, D., Bhaskar, A., Miska, M., and Chung, E. (2016). Real-time traffic state estimation in urban corridors from heterogeneous data. *Transportation Research Part C: Emerging Technologies*, 66:99–118.

Ng, A., Ngiam, J., Yu F, C., Mai, Y., Suen, C., Coates, A., Maas, A., Hannun, A., Huval, B., Wang, T., and Tandon, S. (2016). Convolutional neural network. http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/. Accessed: 2016-05-30.

Nian, K., Zhang, H., Tayal, A., Coleman, T., and Li, Y. (2016). Auto insurance fraud detection using unsupervised spectral ranking for anomaly. *The Journal of Finance and Data Science*, 2(1):58–75.

Niver, E., Mouskos, K. C., Batz, T., and Dwyer, P. (2000). Evaluation of the transcom's system for managing incidents and traffic (transmit). *IEEE Transactions on Intelligent Transportation Systems*, 1(1):15–31.

Noble, C. C. and Cook, D. J. (2003). Graph-based anomaly detection. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 631–636. ACM.

Nogueira, K., Penatti, O. A., and dos Santos, J. A. (2017). Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognition*, 61:539–556.

Oh, S., Byon, Y.-J., and Yeo, H. (2016). Improvement of search strategy with k-nearest neighbors approach for traffic state prediction. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1146–1156.

Olutayo, V. and Eludire, A. (2014). Traffic accident analysis using decision trees and neural networks. *International Journal of Information Technology and Computer Science*, 2:22–28.

Pal, M. (2005). Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 26(1):217–222.

Papageorgiou, M. and Dinopoulou, V. (2003). Disaggregation of cross-lane loop measurements of occupancy. *Traffic Engineering & Control*, 44(5).

Park, H. and Haghani, A. (2016). Real-time prediction of secondary incident occurrences using vehicle probe data. *Transportation Research Part C: Emerging Technologies*, 70:69–85.

Parkany, E. and Xie, C. (2005). A complete review of incident detection algorithms & their deployment: what works and what doesn't. Technical report, University of Massachusetts Transportation Center.

Patra, B. G., Das, D., and Bandyopadhyay, S. (2016). Ju_nlp at semeval-2016 task 6: detecting stance in tweets using support vector machines. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 440–444.

Payne, H. J. and Tignor, S. C. (1978). Freeway incident-detection algorithms based on decision trees with states. *Transportation Research Record: Journal of the Transportation Research Board*, 682.

Pell, A., Meingast, A., and Schauer, O. (2013). Comparison study of software tools for online traffic simulation supporting real-time traffic management of road networns. In *Proceedings of 20th ITS World Congress, ToNyo*.

Persaud, B. N., Hall, F. L., and Hall, L. M. (1990). Congestion identification aspects of the mcmaster incident detection algorithm. *Transportation Research Record: Journal of the Transportation Research Board*, 1287.

Polson, N. G. and Sokolov, V. O. (2017). Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies*, 79:1–17.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106.

Rao, A. M. and Rao, K. R. (2012). Measuring urban traffic congestion-a review. *International Journal for Traffic & Transport Engineering*, 2(4).

Rascon, C. and Meza, I. (2017). Localization of sound sources in robotics: A review. *Robotics and Autonomous Systems*, 96:184–210.

Resende, P. A. A. and Drummond, A. C. (2018). Adaptive anomaly-based intrusion detection system using genetic algorithm and profiling. *Security and Privacy*, 1(4):e36.

Robinson, S. (2005). *The development and application of an urban link travel time model using data derived from inductive loop detectors*. PhD thesis, Imperial College London.

Rossi, R., Gastaldi, M., Gecchele, G., and Barbaro, V. (2015). Fuzzy logic-based incident detection system using loop detectors data. *Transportation Research Procedia*, 10:266–275.

Roy, R. and Saha, P. (2018). Headway distribution models of two-lane roads under mixed traffic conditions: a case study from india. *European Transport Research Review*, 10(1):3.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533.

Rumelhart, D. E., Hinton, G. E., Williams, R. J., et al. (1988). Learning representations by back-propagating errors. *Cognitive Modeling*, 5(3):1.

Saidallah, M., El Fergougui, A., and Elalaoui, A. E. (2016). A comparative study of urban road traffic simulators. In *MATEC Web of Conferences*, volume 81, page 05002. EDP Sciences.

Schrank, D. L. and Lomax, T. J. (2015). *2015 urban mobility scorecard*. Texas Transportation Institute, Texas A & M University.

Scott, D. M., Novak, D. C., Aultman-Hall, L., and Guo, F. (2006). Network robustness index: A new method for identifying critical links and evaluating the performance of transportation networks. *Journal of Transport Geography*, 14(3):215–227.

Shaheen, S. and Finson, R. (2013). Intelligent transportation systems. Technical report, UC Berkeley: Transportation Sustainability Research Center.

Shams, E. A. and Rizaner, A. (2018). A novel support vector machine based intrusion detection system for mobile ad hoc networks. *Wireless Networks*, 24(5):1821–1829.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423.

Shi, X. and Yeung, D.-Y. (2018). Machine Learning for Spatiotemporal Sequence Forecasting: A Survey. *arXiv e-prints*, page arXiv:1808.06865.

Shyu, M.-L., Chen, S.-C., Sarinnapakorn, K., and Chang, L. (2003). A novel anomaly detection scheme based on principal component classifier. Technical report, Miami University CORAL GABLES FL Department of Electrical and Computer Engineering.

Siebel, F. and Mauser, W. (2006). On the fundamental diagram of traffic flow. *SIAM Journal on Applied Mathematics*, 66(4):1150–1162.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Singh, B. and Gupta, A. (2015). Recent trends in intelligent transportation systems: a review. *Journal of Transport Literature*, 9(2):30–34.

Smith, R., Bivens, A., Embrechts, M., Palagiri, C., and Szymanski, B. (2002). Clustering approaches for anomaly based intrusion detection. *Proceedings of Intelligent Engineering Systems through Artificial Neural Networks*, pages 579–584.

Snelder, M., Van Zuylen, H., and Immers, L. (2012). A framework for robustness analysis of road networks for short term variations in supply. *Transportation Research Part A: Policy and Practice*, 46(5):828–842.

Srinivasan, D., Loo, W. H., and Cheu, R. L. (2003). Traffic incident detection using particle swarm optimization. In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No. 03EX706)*, pages 144–151. IEEE.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Steenbruggen, J., Tranos, E., and Rietveld, P. (2016). Traffic incidents in motorways: An empirical proposal for incident detection using data from mobile phone operators. *Journal of Transport Geography*, 54:81–90.

Subramaniam, S., Palpanas, T., Papadopoulos, D., Kalogeraki, V., and Gunopulos, D. (2006). Online outlier detection in sensor data using non-parametric models. In *Proceedings of the 32nd International Conference on Very Large Data Bases*, pages 187–198. VLDB Endowment.

Sugiyamal, Y., Fukui, M., Kikuchi, M., Hasebe, K., Nakayama, A., Nishinari, K., Tadaki, S. I., and Yukawa, S. (2008). Traffic jams without bottlenecks-experimental evidence for the physical mechanism of the formation of a jam. *New Journal of Physics*, 10.

Sullivan, J., Novak, D., Aultman-Hall, L., and Scott, D. M. (2010). Identifying critical road segments and measuring system-wide robustness in transportation networks with isolating links: A link-based capacity-reduction approach. *Transportation Research Part A: Policy and Practice*, 44(5):323–336.

Sun, J. and 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.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Systematics, C. (2005). Traffic congestion and reliability: Trends and advanced strategies for congestion mitigation. *Final Report, Texas Transportation Institute*.

Tan, H., Feng, G., Feng, J., Wang, W., Zhang, Y.-J., and Li, F. (2013). A tensor-based method for missing traffic data completion. *Transportation Research Part C: Emerging Technologies*, 28:15–27.

Tan, H., Wu, Y., Shen, B., Jin, P. J., and Ran, B. (2016). Short-term traffic prediction based on dynamic tensor completion. *IEEE Transactions on Intelligent Transportation Systems*, 17(8):2123–2133.

Tang, J., Liu, F., Zou, Y., Zhang, W., and Wang, Y. (2017). An improved fuzzy neural network for traffic speed prediction considering periodic characteristic. *IEEE Transactions on Intelligent Transportation Systems*, 18(9):2340–2350.

Tang, S. and Gao, H. (2005). Traffic-incident detection-algorithm based on nonparametric regression. *IEEE Transactions on Intelligent Transportation Systems*, 6(1):38–42.

Taylor, M. A. (2008). Critical transport infrastructure in urban areas: Impacts of traffic incidents assessed using accessibility-based network vulnerability analysis. *Growth and Change*, 39(4):593–616.

Taylor, O. and Addison, D. (2000). Novelty detection using neural network technology. In *COMADEM 2000: 13 th International Congress on Condition Monitoring and Diagnostic Engineering Management*, pages 731–743.

Teng, H. and Qi, Y. (2003). Application of wavelet technique to freeway incident detection. *Transportation Research Part C: Emerging Technologies*, 11(3-4):289–308.

Tian, Y., Zhang, K., Li, J., Lin, X., and Yang, B. (2018). Lstm-based traffic flow prediction with missing data. *Neurocomputing*, 318:297–305.

Tian, Z., Jia, L., Dong, H., Su, F., and Zhang, Z. (2016). Analysis of urban road traffic network based on complex network. *Procedia Engineering*, 137:537–546.

Tibaduiza, D. A., Mujica, L. E., Rodellar, J., and Güemes, A. (2016). Structural damage detection using principal component analysis and damage indices. *Journal of Intelligent Material Systems and Structures*, 27(2):233–248.

Tieleman, T. and Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *Coursera: Neural Networks for Machine Learning*, 4(2):26–31.

Todoroki, Y., Han, X.-H., Iwamoto, Y., Lin, L., Hu, H., and Chen, Y.-W. (2017). Detection of liver tumor candidates from ct images using deep convolutional neural networks. In *International Conference on Innovation in Medicine and Healthcare*, pages 140–145. Springer.

Toledo, T., Koutsopoulos, H., Ben-Akiva, M., and Jha, M. (2005). Microscopic traffic simulation: Models and application. In *Simulation Approaches in Transportation Analysis*, pages 99–130. Springer.

Van Lint, J., Hoogendoorn, S., and van Zuylen, H. J. (2005). Accurate freeway travel time prediction with state-space neural networks under missing data. *Transportation Research Part C: Emerging Technologies*, 13(5-6):347–369.

Van Phuong, T., Hung, L. X., Cho, S. J., Lee, Y.-K., and Lee, S. (2006). An anomaly detection algorithm for detecting attacks in wireless sensor networks. In *International Conference on Intelligence and Security Informatics*, pages 735–736. Springer.

Vandenberghe, W., Vanhauwaert, E., Verbrugge, S., Moerman, I., and Demeester, P. (2012). Feasibility of expanding traffic monitoring systems with floating car data technology. *IET Intelligent Transport Systems*, 6(4):347–354.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer Science & Business Media.

Vishnuvarthanan, G., Rajasekaran, M. P., Subbaraj, P., and Vishnuvarthanan, A. (2016). An unsupervised learning method with a clustering approach for tumor identification and tissue segmentation in magnetic resonance brain images. *Applied Soft Computing*, 38:190–212.

Vlahogianni, E. I., Karlaftis, M. G., and Golias, J. C. (2014). Short-term traffic forecasting: Where we are and where we're going. *Transportation Research Part C: Emerging Technologies*, 43:3–19.

Wang, J., Gu, Q., Wu, J., Liu, G., and Xiong, Z. (2016a). Traffic speed prediction and congestion source exploration: A deep learning method. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 499–508. IEEE.

Wang, R., Fan, S., and Work, D. B. (2016b). Efficient multiple model particle filtering for joint traffic state estimation and incident detection. *Transportation Research Part C: Emerging Technologies*, 71:521–537.

Wang, T., Zhang, W., Ye, C., Wei, J., Zhong, H., and Huang, T. (2016c). Fd4c: Automatic fault diagnosis framework for web applications in cloud computing. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(1):61–75.

Wang, Y. and Papageorgiou, M. (2005). Real-time freeway traffic state estimation based on extended kalman filter: a general approach. *Transportation Research Part B: Methodological*, 39(2):141–167.

Wasserstein, R. L., Schirm, A. L., and Lazar, N. A. (2019). Moving to a world beyond "p< 0.05".

Wei, H., Zheng, G., Gayah, V., and Li, Z. (2019). A Survey on Traffic Signal Control Methods. *arXiv e-prints*, page arXiv:1904.08117.

Weisbrod, G., Vary, D., and Treyz, G. (2003). Measuring economic costs of urban traffic congestion to business. *Transportation Research Record: Journal of the Transportation Research Board*, 1839:98–106.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1-3):37–52.

Woodman, O. and Harle, R. (2008). Pedestrian localisation for indoor environments. In *Proceedings of the 10th International Conference on Ubiquitous Computing*, pages 114–123. ACM.

Wu, C., Guo, C., Xie, Z., Ni, F., and Liu, H. (2018a). A signal-based fault detection and tolerance control method of current sensor for pmsm drive. *IEEE Transactions on Industrial Electronics*.

Wu, Y., Hassner, T., Kim, K., Medioni, G., and Natarajan, P. (2018b). Facial landmark detection with tweaked convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):3067–3074.

Wu, Y. and Tan, H. (2016). Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework.

Wu, Y., Tan, H., Qin, L., Ran, B., and Jiang, Z. (2018c). A hybrid deep learning based traffic flow prediction method and its understanding. *Transportation Research Part C: Emerging Technologies*, 90:166–180.

Xia, J., Huang, W., and Guo, J. (2012). A clustering approach to online freeway traffic state identification using its data. *KSCE Journal of Civil Engineering*, 16(3):426–432.

Xie, X.-F., Feng, Y., Smith, S. F., and Head, K. L. (2014). Unified route choice framework: Specification and application to urban traffic control. *Transportation Research Record*, 2466(1):105–113.

Xie, X.-F., Smith, S. F., Lu, L., and Barlow, G. J. (2012). Schedule-driven intersection control. *Transportation Research Part C: Emerging Technologies*, 24:168–189.

Xu, C., Liu, P., Yang, B., and Wang, W. (2016). Real-time estimation of secondary crash likelihood on freeways using high-resolution loop detector data. *Transportation Research Part C: Emerging Technologies*, 71:406–418.

Xu, C., Tarko, A. P., Wang, W., and Liu, P. (2013). Predicting crash likelihood and severity on freeways with real-time loop detector data. *Accident Analysis & Prevention*, 57:30–39.

Yang, G., Wang, Y., Yu, H., Ren, Y., and Xie, J. (2018). Short-term traffic state prediction based on the spatiotemporal features of critical road sections. *Sensors*, 18(7):2287.

Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., Gong, P., Ye, J., and Li, Z. (2018). Deep multi-view spatial-temporal network for taxi demand prediction. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Ye, N. and Chen, Q. (2001). An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems. *Quality and Reliability Engineering International*, 17(2):105–112.

Yildirimoglu, M. and Geroliminis, N. (2013). Experienced travel time prediction for congested freeways. *Transportation Research Part B: Methodological*, 53:45–63.

Younes, M. B. and Boukerche, A. (2015). A performance evaluation of an efficient traffic congestion detection protocol (ecode) for intelligent transportation systems. *Ad Hoc Networks*, 24:317–336.

Yu, H., Wu, Z., Wang, S., Wang, Y., and Ma, X. (2017). Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. *Sensors*, 17(7):1501.

Yu, X., Xiong, S., He, Y., Wong, W. E., and Zhao, Y. (2016). Research on campus traffic congestion detection using BP neural network and Markov model. *Journal of Information Security and Applications*, 31:54–60.

Yuan, F. and Cheu, R. L. (2003). Incident detection using support vector machines. *Transportation Research Part C: Emerging Technologies*, 11(3-4):309–328.

Yuan, Y., Van Lint, J., Wilson, R. E., van Wageningen-Kessels, F., and Hoogendoorn, S. P. (2012). Real-time lagrangian traffic state estimator for freeways. *IEEE Transactions on Intelligent Transportation Systems*, 13(1):59–70.

Zhang, F., Li, J., Chen, J., Sun, J., and Attey, A. (2017). Hesitant distance set on hesitant fuzzy sets and its application in urban road traffic state identification. *Engineering Applications of Artificial Intelligence*, 61:57–64.

Zhang, G., Patuwo, B. E., and Hu, M. Y. (1998). Forecasting with artificial neural networks:: The state of the art. *International Journal of Forecasting*, 14(1):35–62.

Zhang, J., Zheng, Y., Qi, D., Li, R., Yi, X., and Li, T. (2018a). Predicting citywide crowd flows using deep spatio-temporal residual networks. *Artificial Intelligence*, 259:147–166.

Zhang, Q., Nian Wu, Y., and Zhu, S.-C. (2017). Interpretable Convolutional Neural Networks. *arXiv e-prints*, page arXiv:1710.00935.

Zhang, Y. and Haghani, A. (2015). A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies*, 58:308–324.

Zhang, Z. and He, Q. (2016). On-site traffic accident detection with both social media and traffic data. In *Proc. 9th Triennial Symp. Transp. Anal.(TRISTAN)*.

Zhang, Z., He, Q., Gao, J., and Ni, M. (2018b). A deep learning approach for detecting traffic accidents from social media data. *Transportation Research Part C: Emerging Technologies*, 86:580–596.

Zhang, Z., He, Q., Tong, H., Gou, J., and Li, X. (2016). Spatial-temporal traffic flow pattern identification and anomaly detection with dictionary-based compression theory in a large-scale urban network. *Transportation Research Part C: Emerging Technologies*, 71:284–302.

Zhao, Y., Wen, J., Xiao, F., Yang, X., and Wang, S. (2017). Diagnostic bayesian networks for diagnosing air handling units faults–part i: Faults in dampers, fans, filters and sensors. *Applied Thermal Engineering*, 111:1272–1286.

Zheng, M., Li, T., Zhu, R., Chen, J., Ma, Z., Tang, M., Cui, Z., and Wang, Z. (2019). Traffic accident's severity prediction: a deeplearning approach based cnn network. *IEEE Access*.

Zhou, B., Bau, D., Oliva, A., and Torralba, A. (2017). Interpreting Deep Visual Representations via Network Dissection. *arXiv e-prints*, page arXiv:1711.05611.

Zhou, C., Huang, S., Xiong, N., Yang, S.-H., Li, H., Qin, Y., and Li, X. (2015). Design and analysis of multimodel-based anomaly intrusion detection systems in industrial process automation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(10):1345–1360.

Zhou, Y., Sheu, J.-B., and Wang, J. (2017). Robustness assessment of urban road network with consideration of multiple hazard events. *Risk Analysis*, 37(8):1477–1494.

Zhu, L. (2015). Data fusion for travel time estimation. Master's thesis, Imperial College London.

Zhu, L., Guo, F., Krishnan, R., and Polak, J. W. (2018a). The Use of Convolutional Neural Networks for Traffic Incident Detection at a Network Level. In *Transportation Research Board 97th Annual Meeting*, pages 18–00321, Washington, DC.

Zhu, L., Guo, F., Polak, J. W., and Krishnan, R. (2018b). Urban link travel time estimation using traffic states-based data fusion. *IET Intelligent Transport Systems*, 12(7):651–663.

Appendix A

List of Acronyms and Abbreviations

API Application Programming Interface

ATMS Advanced Traffic Management Systems

ADAS Advanced Driver Assistance Systems

ATIS Advanced Traveller Information Systems

AVCSS Advanced Vehicle Control Safety Systems

AI Artificial Intelligence

ANNs Artificial Neural Networks

AUC Area Under the Curve

ARIMA Auto Regressive Integrated Moving Average

BPNN Back Propagation Neural Networks

CNN Convolutional Neural Networks

Conv Convolutional Layer

DLR German Aerospace Centre

DNN Deep Neural Networks

DR Detection Rate

EM Expectation Maximisation

F1 score F measurement

FC Fully Connected Layer

FN False Negative

FP False Positive

FPR False Positive Rate or False Alarm Rate

GBC Gradient Boosting Classifier

GIS Geographical Information Systems

GMM Gaussian Mixture Model

GPS Global Positioning System

I2I Infrastructure-to-Infrastructure

I2V Infrastructure-to-Vehicle

ICT Information and Communication Technology

ILD Inductive Loop Detector

ITMS Traffic Incident Management Systems

ITS Intelligent Transport Systems

KNN K-Nearest Neighbours

KPI Key Performance Indicator

LASSO Least Absolute Shrinkage and Selection Operator

LoS Level of Service

LSTM Long Short Term Machine

MIDAS Motorway Incident Detection and Automated Signalling

MLP Multiple Layer Perceptron

MTTD Mean Time to Detection

NRC Non Recurrent Congestion

OD Origin Destination

PCA Principal Component Analysis

RC Recurrent Congestion

ReLU Rectified Linear Unit

RF Random Forest

RGB Red, Green and Blue

RNNs Recurrent Neural Networks

ROC Receiver Operating Characteristic

SCOOT Split Cycle Offset Optimisation Technique

SGD Stochastic Gradient Descent

SNB Semi Naive Bayes classifier

SUMO Simulation of Urban MObility

SVM Support Vector Machine

TfL Transport for London

TMC Traffic Management Centre

TN True Negative

TP True Positive

UTC Urban Traffic Control

V2I Vehicle-to-Infrastructure

V2V Vehicle-to-Vehicle

Appendix B

Data Preprocessing

B.1 Loop Detector Data Preprocessing

ILDs are widely used to collect traffic variables. Detectors, usually 2m long and 1.5 wide, operate with the electromagnetic induction principle to produce a binary '1' bit under the presence of the vehicle, otherwise '0' bit is generated. The ILD used in this research are polled with a frequency of 4 Hz, which means they detect the existence of a vehicle every 250 milliseconds. Based on the binary 'state' and number of 250 milliseconds, the basic traffic variables, i.e. flow and occupancy, can be calculated as below. Flow is the number of vehicles that pass a fixed point during unit time, which can be calculated by the equation below.

$$q = \frac{D_{0,1}}{T} \tag{B.1}$$

where, q is the flow, $D_{0,1}$ is the number of '0' bit to '1' bit translations detected by the ILD, in vehicles and T is the duration of time that count took place (per hour).

Occupancy measures the proportion of time that a vehicle presents on the ILD. We can obtain the occupancy from the ILD raw data by the equation below.

$$o = \frac{D_1}{D_0 + D_1} \tag{B.2}$$

where o is the occupancy, D_0 is the number of '0' bit detected by the ILD and D_1 is the number of '1' bit detected by the ILD. The examples of flow and occupancy calculated from raw ILD data have been presented in the following sub sections.

B.1.1 Bath ILD Data

Bath ILD data were provided by the council of Bath. Table B.1 and Table B.2 show the examples of raw traffic flow and occupancy data respectively. These traffic data cover every day from 7:00 to 18:55 in 15-minute intervals during the course of two years from June 2015 to June 2017.

Table B.1: Samples of Bath traffic flow data

```
;================================================
; EXTRACT @C:\PROGRAM FILES (X86)\SIEMENS TRAFFIC
;
; Th 22-Jun-2017 15:06:39
;                                          Det    Det
;                          Start   End   Flow*  Flow*
;  Site      Day  Date     Time   Time    Mean   Count
;               yyyymmdd hh:mm hh:mm   veh/h
;----------- -- -------- ----- ----- ------ ------
N0125501     FR 20150501 07:00 07:15    272     96
N0125501     FR 20150501 07:15 07:30    360    107
N0125501     FR 20150501 07:30 07:45    424     96
N0125501     FR 20150501 07:45 08:00    468     96
N0125501     FR 20150501 08:00 08:15    452    107
N0125501     FR 20150501 08:15 08:30    456     96
N0125501     FR 20150501 08:30 08:45    556     96
N0125501     FR 20150501 08:45 09:00    576    107
N0125501     FR 20150501 09:00 09:15    544     96
N0125501     FR 20150501 09:15 09:30    560    107
N0125501     FR 20150501 09:30 09:45    548     96
N0125501     FR 20150501 09:45 10:00    624     96
N0125501     FR 20150501 10:00 10:15    612    102
N0125501     FR 20150501 10:15 10:30    572    104
N0125501     FR 20150501 10:30 10:45    532     96
N0125501     FR 20150501 10:45 11:00    580    104
N0125501     FR 20150501 11:00 11:15    656     96
N0125501     FR 20150501 11:15 11:30    620    104
N0125501     FR 20150501 11:30 11:45    704     96
N0125501     FR 20150501 11:45 12:00    728    104
N0125501     FR 20150501 12:00 12:15    516     96
N0125501     FR 20150501 12:15 12:30    756    104
N0125501     FR 20150501 12:30 12:45    692     96
N0125501     FR 20150501 12:45 13:00    668    104
N0125501     FR 20150501 13:00 13:15    744     96
N0125501     FR 20150501 13:15 13:30    696    104
N0125501     FR 20150501 13:30 13:45    736     96
N0125501     FR 20150501 13:45 14:00    732    104
N0125501     FR 20150501 14:00 14:15    776     96
N0125501     FR 20150501 14:15 14:30    720    104
;================================================
```

Table B.2: Samples of Bath occupancy data

```
;================================================
; EXTRACT @C:\PROGRAM FILES (X86)\SIEMENS TRAFFIC
;
; Tu 13-Jun-2017 09:41:28
;                                          Det    Det
;                        Start   End      Occ*   Occ*
;  Site      Day  Date    Time  Time      Mean  Count
;               yyyymmdd hh:mm hh:mm        %
;----------- -- -------- ----- ----- ------ ------
N0125501     FR 20150501 07:00 07:15    5.2     96
N0125501     FR 20150501 07:15 07:30    7.0    107
N0125501     FR 20150501 07:30 07:45    7.9     96
N0125501     FR 20150501 07:45 08:00    8.7     96
N0125501     FR 20150501 08:00 08:15    8.3    107
N0125501     FR 20150501 08:15 08:30    9.2     96
N0125501     FR 20150501 08:30 08:45   11.4     96
N0125501     FR 20150501 08:45 09:00   11.0    107
N0125501     FR 20150501 09:00 09:15   10.2     96
N0125501     FR 20150501 09:15 09:30   11.0    107
N0125501     FR 20150501 09:30 09:45   13.1     96
N0125501     FR 20150501 09:45 10:00   16.0     96
N0125501     FR 20150501 10:00 10:15   12.5    102
N0125501     FR 20150501 10:15 10:30   11.4    104
N0125501     FR 20150501 10:30 10:45   10.9     96
N0125501     FR 20150501 10:45 11:00   10.6    104
N0125501     FR 20150501 11:00 11:15   13.2     96
N0125501     FR 20150501 11:15 11:30   12.0    104
N0125501     FR 20150501 11:30 11:45   13.4     96
N0125501     FR 20150501 11:45 12:00   14.2    104
N0125501     FR 20150501 12:00 12:15    9.7     96
N0125501     FR 20150501 12:15 12:30   14.0    104
N0125501     FR 20150501 12:30 12:45   13.2     96
N0125501     FR 20150501 12:45 13:00   12.6    104
N0125501     FR 20150501 13:00 13:15   16.4     96
N0125501     FR 20150501 13:15 13:30   14.3    104
N0125501     FR 20150501 13:30 13:45   13.5     96
N0125501     FR 20150501 13:45 14:00   14.0    104
N0125501     FR 20150501 14:00 14:15   13.8     96
N0125501     FR 20150501 14:15 14:30   13.1    104
;================================================
```

B.1.2 London ILD Data

London traffic flow and occupancy data were extracted and calculated based on the raw ILD output recorded at 4 Hz available through a system called the London SCOOT Archive Database (LSAD). This IT system is hosted at TfL and jointly owned by Imperial College London and TfL (Robinson, 2005; Krishnan, 2008). Table B.3 shows an example of extracted ILD data in 5-minute time intervals.

Table B.3: Samples of London ILD data

Detector Name	Time	Traffic Flow (veh/5mins)	Occupancy (%)
N01/070a1	10:20:00	17	54.33
N02/028c1	10:20:00	12	7.67
N01/143c1	10:20:00	11	17.58
N01/166c1	10:20:00	12	12.92
N01/311a1	10:20:00	7	6.83
N12/184c1	10:20:00	19	31.08
N01/233e1	10:20:00	19	15.75
N10/072a1	10:20:00	11	7.67
N01/156m1	10:20:00	2	1.67
N03/032c1	10:20:00	8	76
N10/064e1	10:20:00	25	41
N12/063c1	10:20:00	16	23.42
N03/187c1	10:20:00	1	1.08
N01/478c1	10:20:00	6	5.25
N02/089f1	10:20:00	11	38.67
N02/032b1	10:20:00	18	20.67
N01/107f1	10:20:00	19	15.17
N01/383b1	10:20:00	9	16.25
N01/154a1	10:20:00	22	36.75
N01/349b1	10:20:00	8	10.67
N10/255x1	10:20:00	21	17.92
N01/198b1	10:20:00	4	93.5
N12/004c1	10:20:00	10	12.25
N02/142h1	10:20:00	0	100
N10/147d1	10:20:00	15	11.42
N01/241a2	10:20:00	30	31.92
N12/091q1	10:20:00	4	3.5
N12/166d1	10:20:00	8	8
N01/229c1	10:20:00	14	10.42
N12/171l1	10:20:00	24	21.75
N02/065h1	10:20:00	23	18.25

B.2 Traffic Incident Data

The data summarises operator entries recorded in the Traffic Incident Management System (TIMS). Each Row is identified by the combination of day, hour, sev, and evt_id. The detailed description of London traffic incident data is presented in Table B.4, while Table B.6 shows an example of traffic incident provided by TfL.

Table B.4: Description of London traffic incident data

Column Name	Description
Day	date the row refers to as a number, YYYMMDD
Hour	hour of the day that the row refers to, HH
Sev	severity of the incident in the hour referenced in each row as Table B.5
PLAN	planned or unplanned incident with values where P for planned incidents and U for unplanned incidents
TLRN	Y/N indicator for if the incident took place on the TLRN (Transport for London Road Network)
SRN	Y/N indicator for if the incident took place on the SRN (Strategic Road Network)
TUNNEL	Y/N indicator for if the incident took place in a Tunnel
Loca	text description of the location
Cat1	first level category description for the incident
Cat2	second level category description for the incident
Cat3	third level, detailed, category description for the incident
Easting	x coord, British National Grid (+init=epsg:27700)
Northing	y coord, British National Grid (+init=epsg:27700)
Dur	duration in minutes, valued $(0, 60]$
Evt_id	ID of the TIMS event. This does not uniquely identify each row, as the evolution of event status over time is detailed in this dataset.

Table B.5: Level of severity of London traffic incident

Abbr	Description
UNK	Unknown
NAE	Not Active at End
NDE	Not Active During
NES	Not Active at Start
MIN	Minimal
MOD	Moderate
NAT	No Action
SER	Serious
SEV	Severe

Table B.6: Samples of traffic incident data from TfL

Day	Hour	Sev	PLAN	TLRN	SRN	TUNNEL	Loca	Cat1	Cat2	Cat3	Easting	Northing	Dur	Evt_id
20150206	20	MOD	P	Y	N	N	GRAY'S INN ROAD	Works	Utility	BT	530500.0	182840.0	60.000000	76027
20150206	22	MOD	P	Y	N	N	GRAY'S INN ROAD	Works	Utility	BT	530500.0	182840.0	60.000000	76027
20150206	15	NES	P	Y	N	N	GRAY'S INN ROAD	Works	Utility	BT	530500.0	182840.0	2.316667	76027
20150206	19	MOD	P	Y	N	N	GRAY'S INN ROAD	Works	Utility	BT	530500.0	182840.0	60.000000	76027
20150206	18	MOD	P	Y	N	N	GRAY'S INN ROAD	Works	Utility	BT	530500.0	182840.0	60.000000	76027
20150206	17	MOD	P	Y	N	N	GRAY'S INN ROAD	Works	Utility	BT	530500.0	182840.0	60.000000	76027
20150206	15	MOD	P	Y	N	N	GRAY'S INN ROAD	Works	Utility	BT	530500.0	182840.0	30.683333	76027
20150206	21	MOD	P	Y	N	N	GRAY'S INN ROAD	Works	Utility	BT	530500.0	182840.0	60.000000	76027
20150206	23	MOD	P	Y	N	N	GRAY'S INN ROAD	Works	Utility	BT	530500.0	182840.0	60.000000	76027
20150207	12	MOD	P	Y	N	N	GRAY'S INN ROAD	Works	Utility	BT	530500.0	182840.0	60.000000	76027

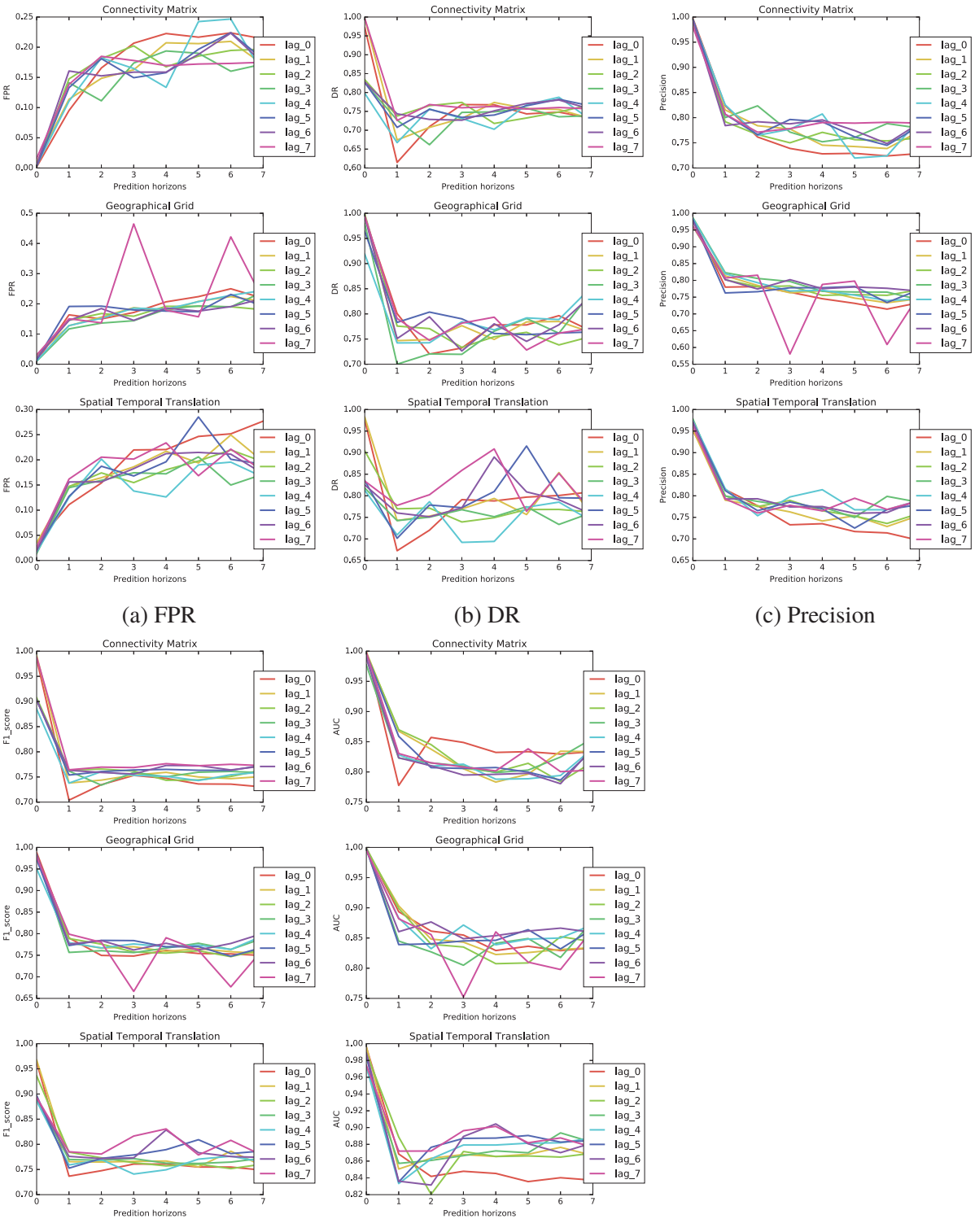
Appendix C

Full Results of Translation Layer, Method and Localisation

This chapter presents full results for the evaluation of translation layers, detection methods and localisation for early recurrent congestion detection. The evaluation is based on the indicators including Detection Rate (DR), False Positive Rate (FPR), precision, F1 score and Area Under the Curve (AUC) for time lags and early prediction horizons ranging from zero to seven (i.e., around two hours ahead).

C.1 Full Sensitivity Analysis of Translation Layers for Early RC Detection

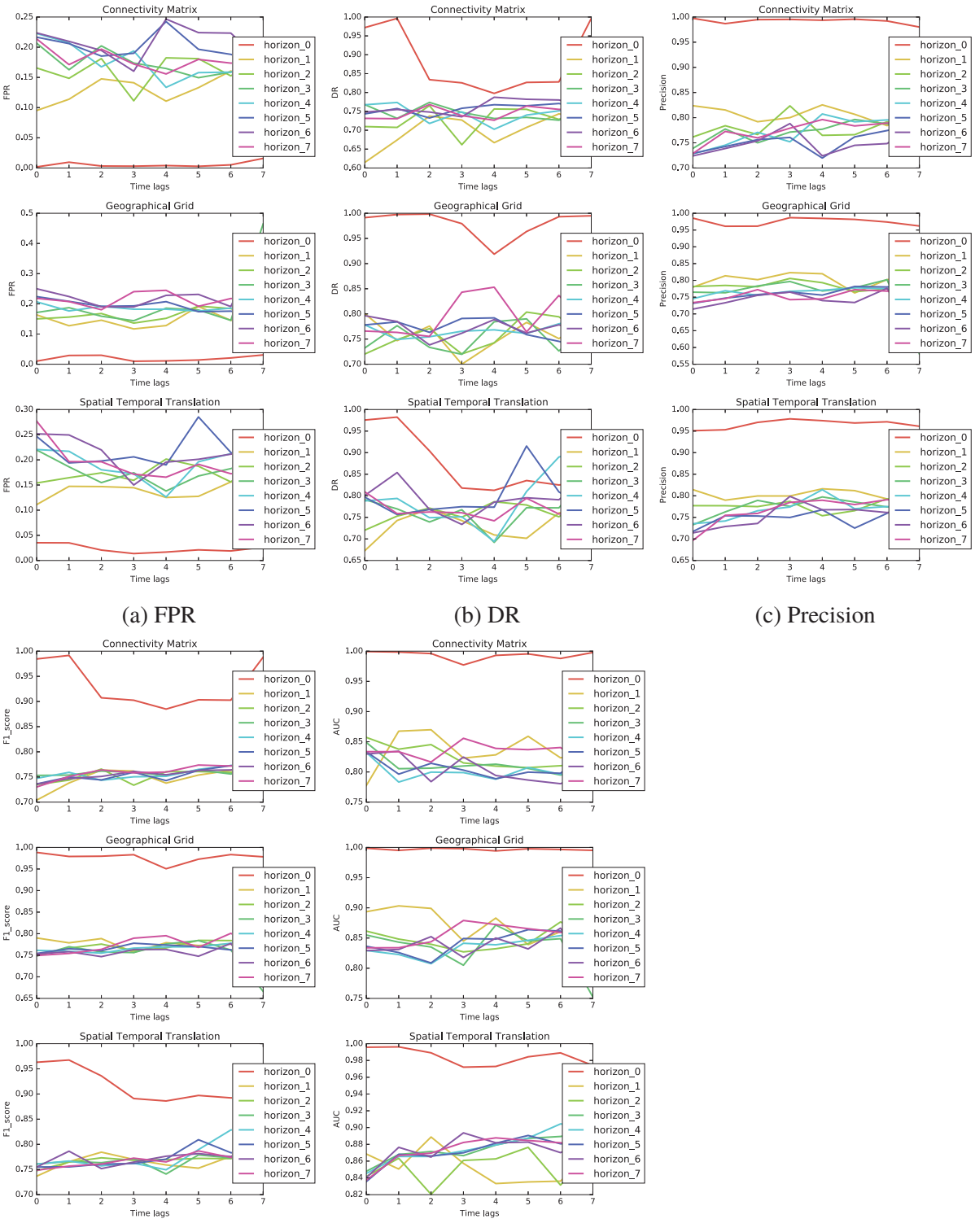
This section presents full results of translation layer evaluation for early recurrent congestion detection in Chapter 4 by including results in terms of all time lags and prediction horizons. Specifically, Figure C.1 shows the results in terms of prediction horizons, while Figure C.2 presents that of time lags for early recurrent congestion detection.



(a) FPR (b) DR (c) Precision

(d) F1 score (e) AUC

Figure C.1: Full results of translation layer evaluation for early recurrent congestion detection in terms of prediction horizons



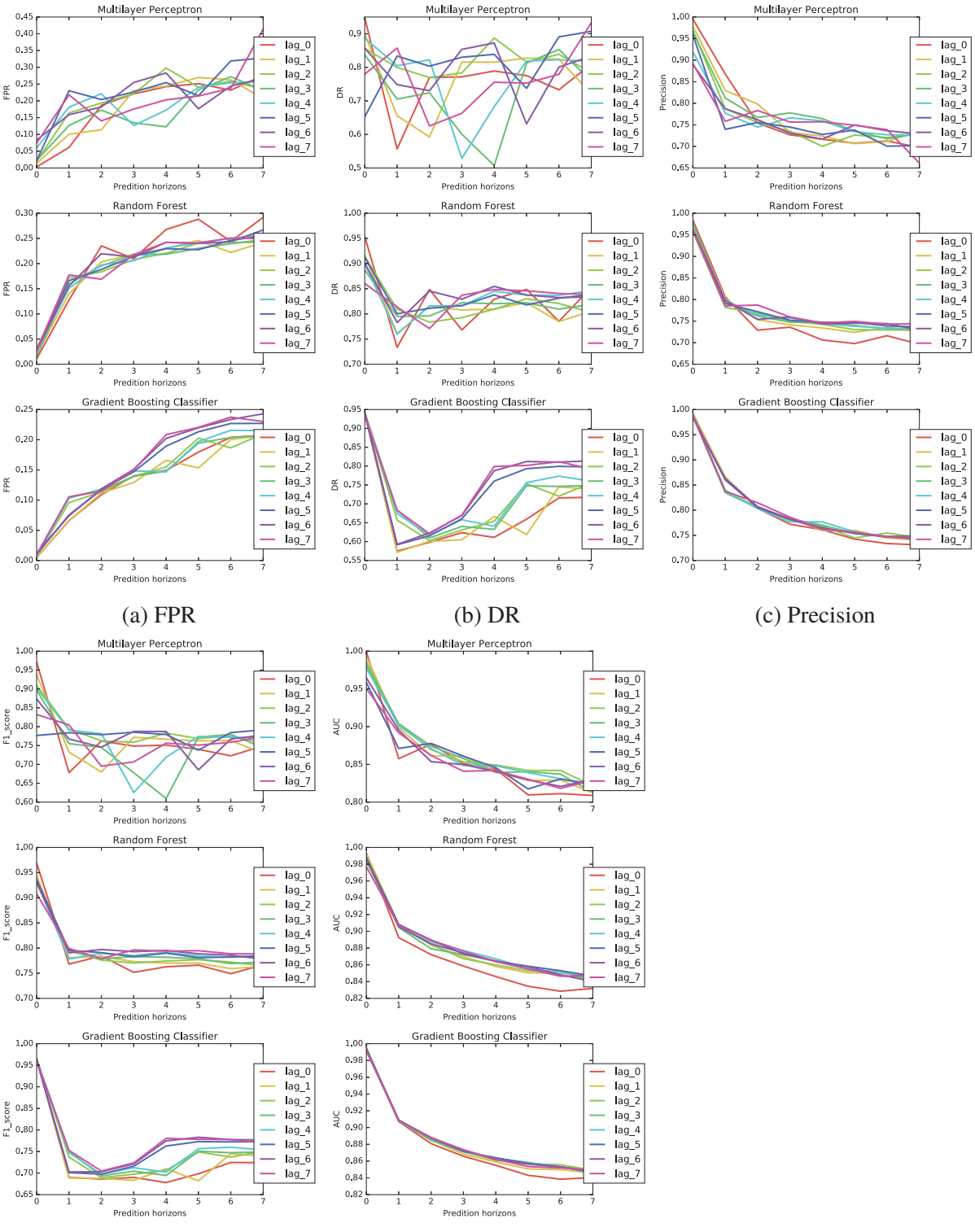
(a) FPR (b) DR (c) Precision

(d) F1 score (e) AUC

Figure C.2: Full results of translation layer evaluation for early recurrent congestion detection in terms of time lags

C.2 Full Sensitivity Analysis of Detection Methods for Early RC Detection

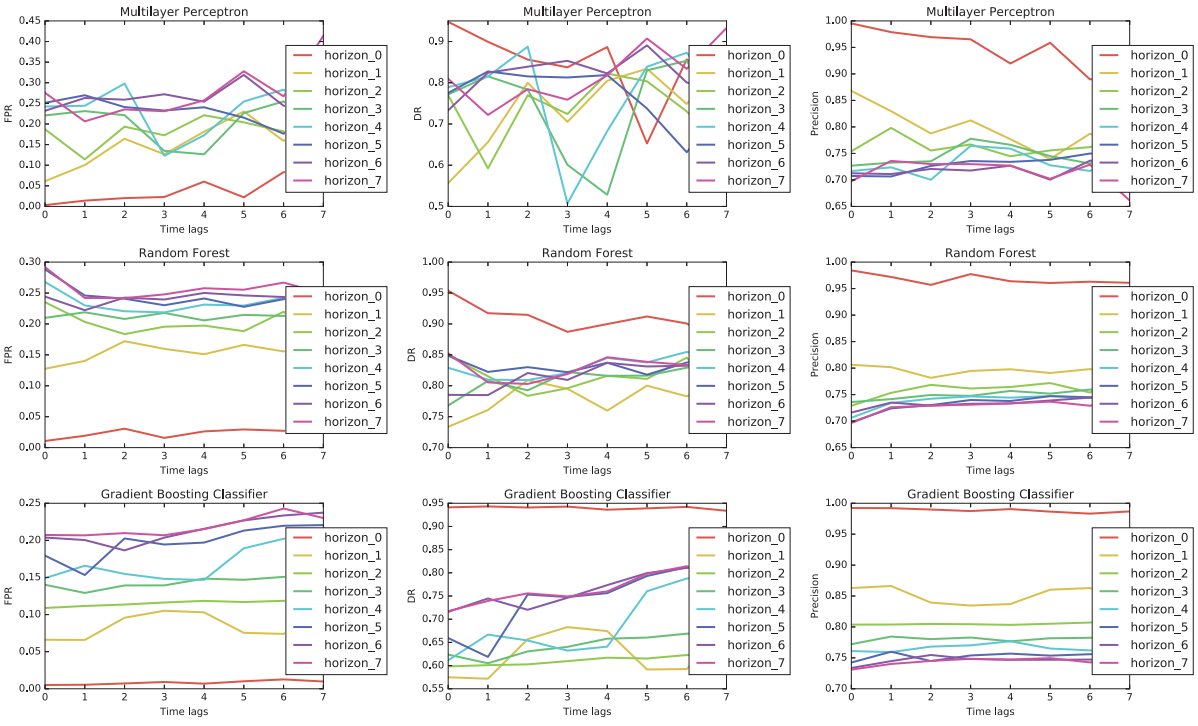
This section presents full results of deep learning based detection methods evaluation for early recurrent congestion detection in Chapter 5 by including results in terms of all time lags and prediction horizons. Specifically, Figure C.3 shows the results in terms of prediction horizons, while Figure C.4 presents that of time lags for early recurrent congestion detection.



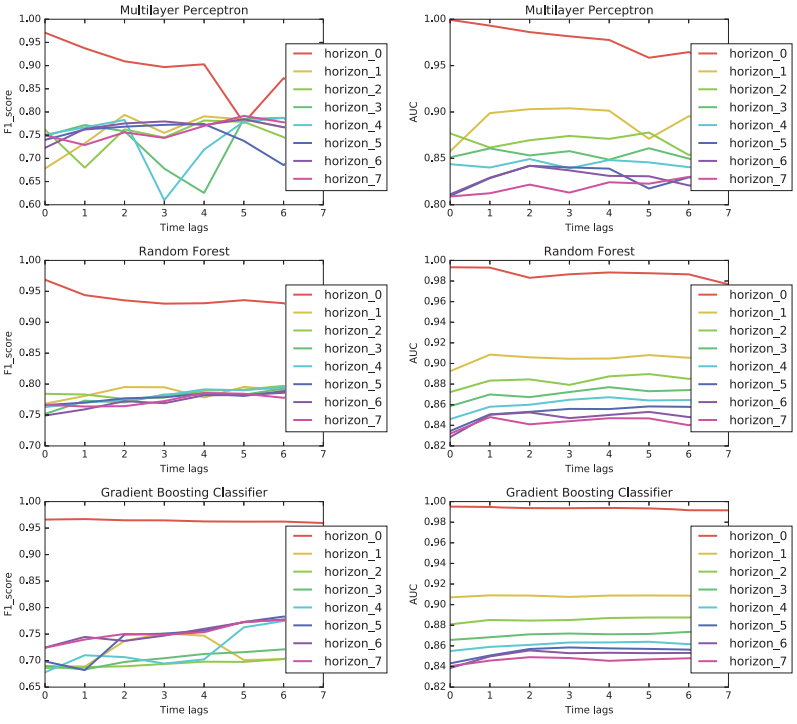
(a) FPR (b) DR (c) Precision

(d) F1 score (e) AUC

Figure C.3: Full results of detection method evaluation for early recurrent congestion detection in terms of prediction horizons



(a) FPR (b) DR (c) Precision

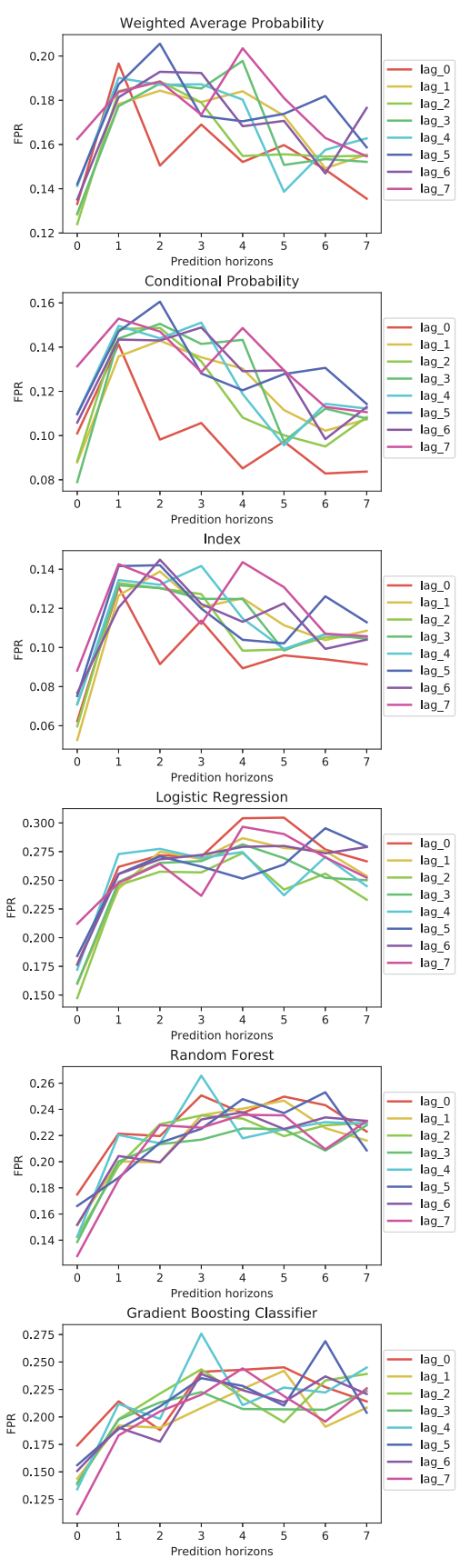


(d) F1 score (e) AUC

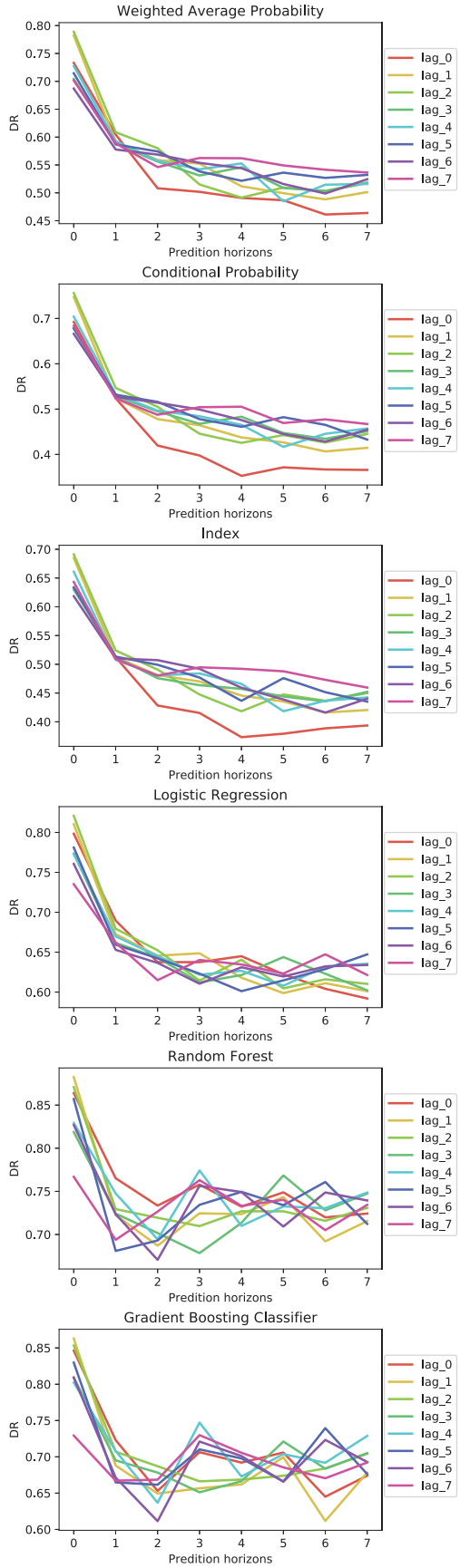
Figure C.4: Full results of detection method evaluation for early recurrent congestion detection in terms of time lags

C.3 Full Sensitivity Analysis of Localisation for Early RC Detection

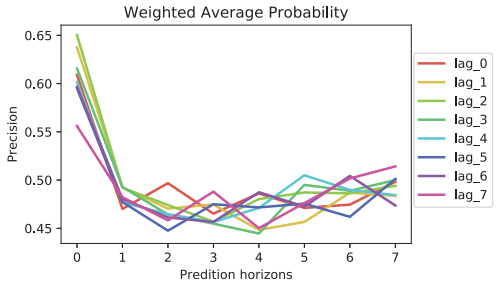
This section presents full results of localisation evaluation for early recurrent congestion detection in Chapter 6 by including results in terms of all time lags and prediction horizons. Specifically, Figure C.5 shows the results in terms of prediction horizons, while Figure C.6 presents that of time lags for early recurrent congestion detection.

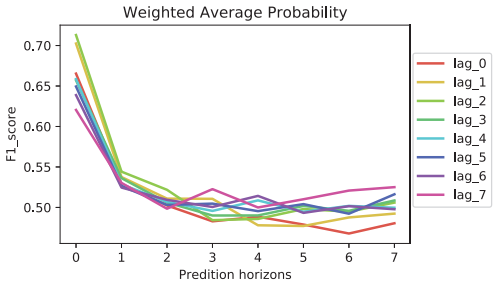

Weighted Average Probability
Conditional Probability
Index
Logistic Regression
Random Forest
Gradient Boosting Classifier
FPR
Predition horizons
lag_0
lag_1
lag_2
lag_3
lag_4
lag_5
lag_6
lag_7

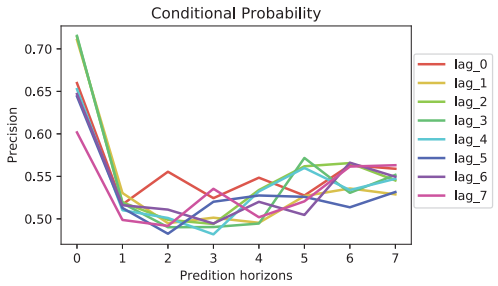
(a) FPR

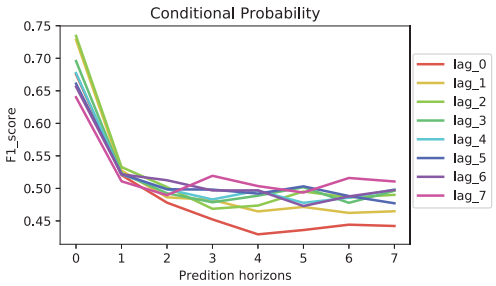

Weighted Average Probability
Conditional Probability
Index
Logistic Regression
Random Forest
Gradient Boosting Classifier
DR
Predition horizons
lag_0
lag_1
lag_2
lag_3
lag_4
lag_5
lag_6
lag_7

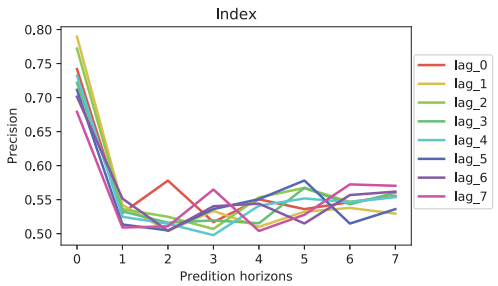
(b) DR

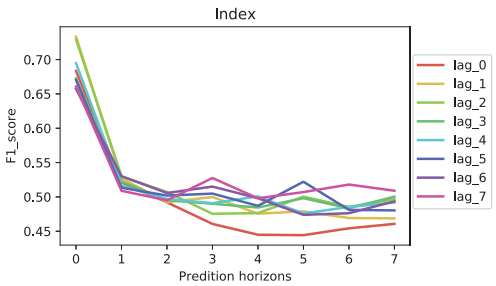


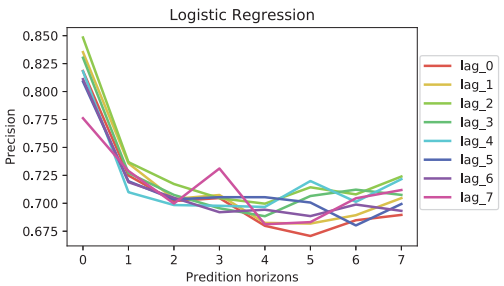


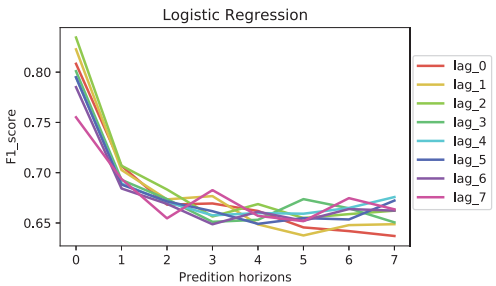


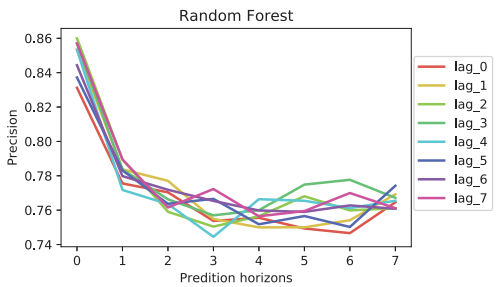


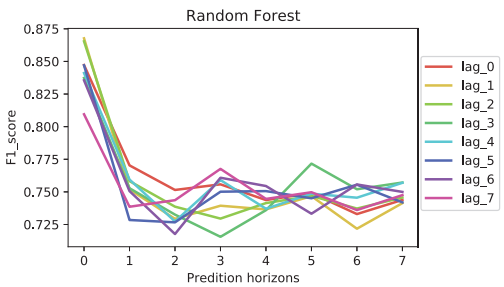


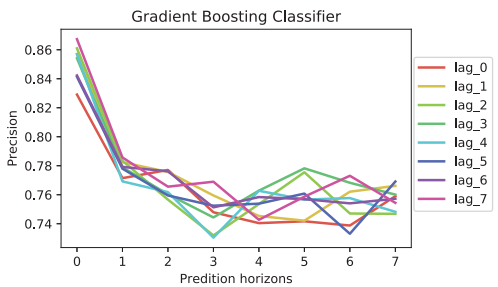


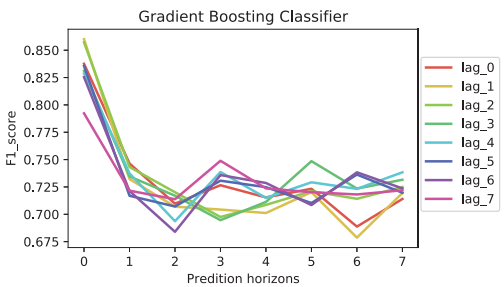






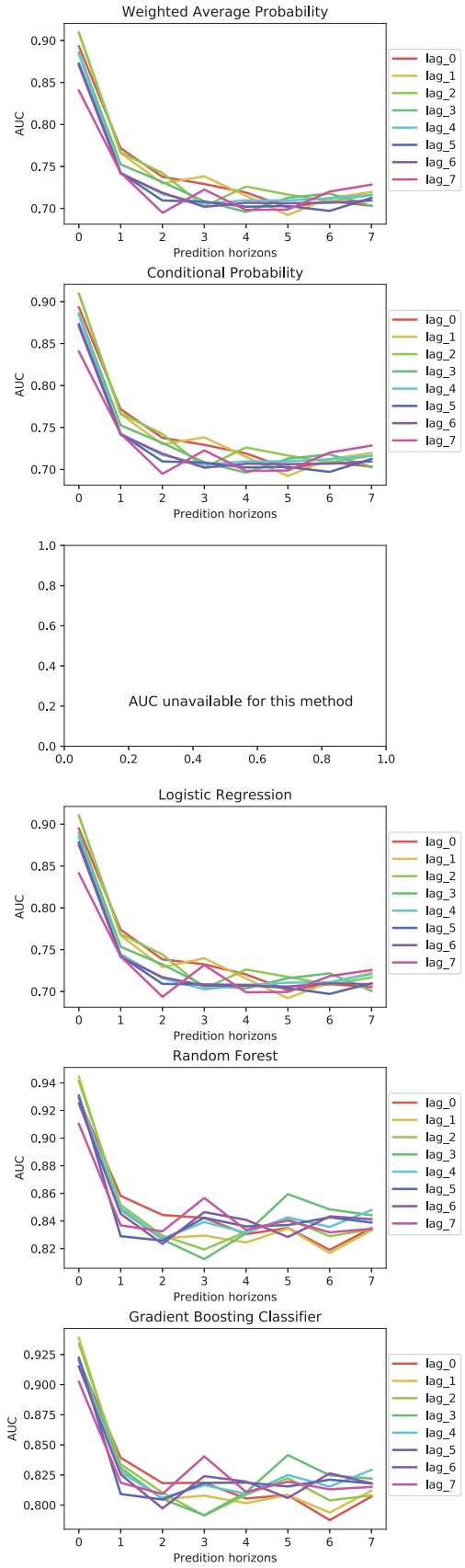






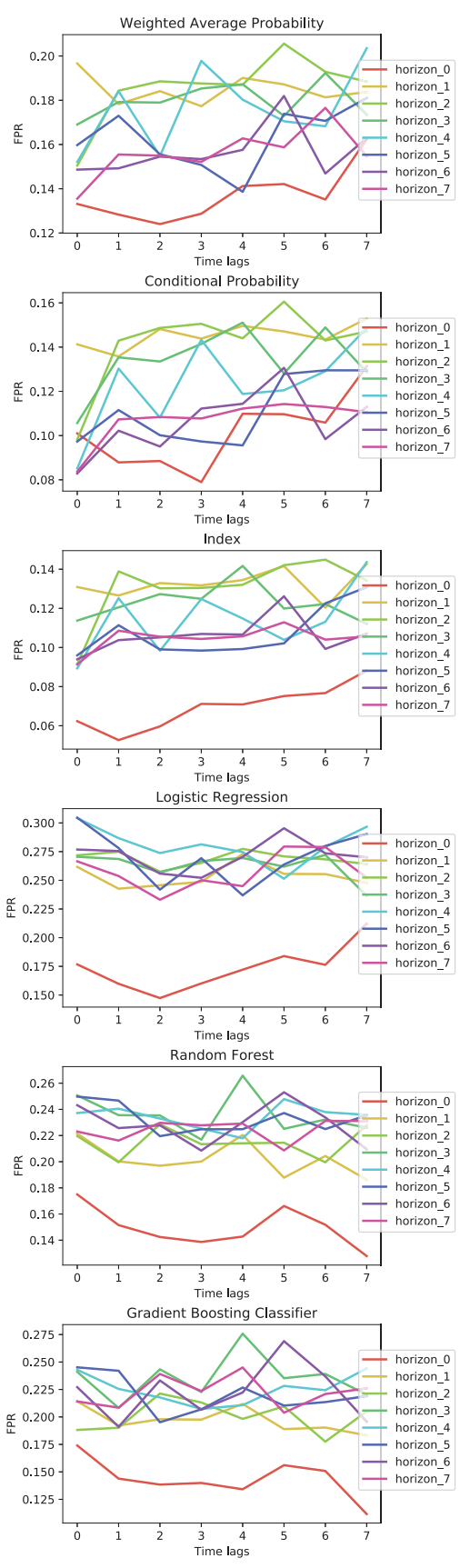
(c) Precision

(d) F1 score

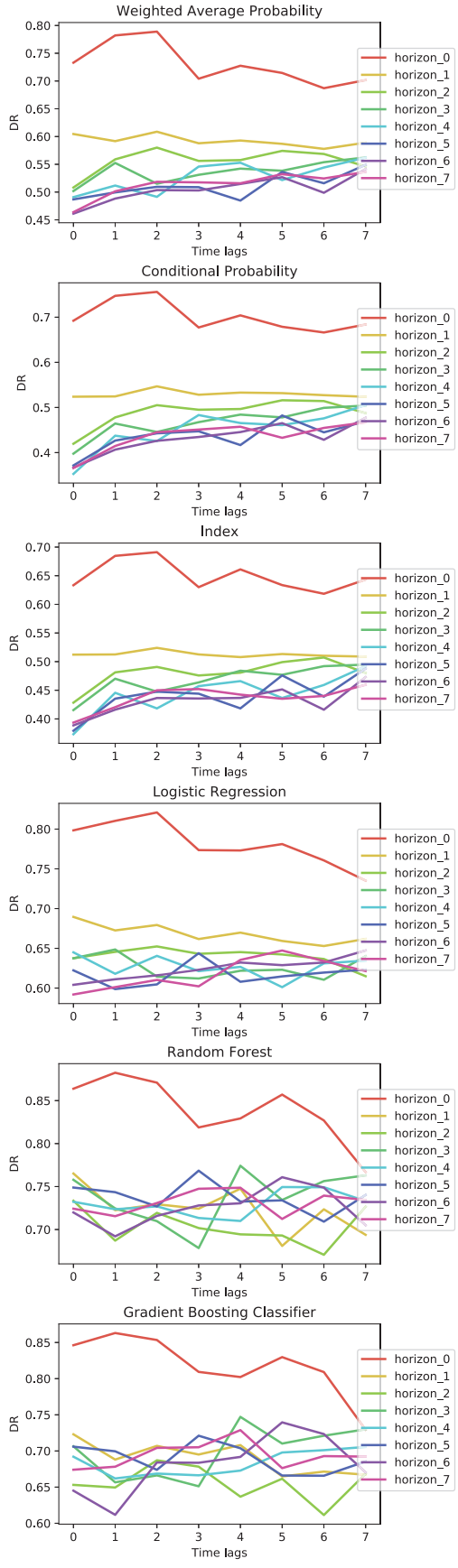


(e) AUC

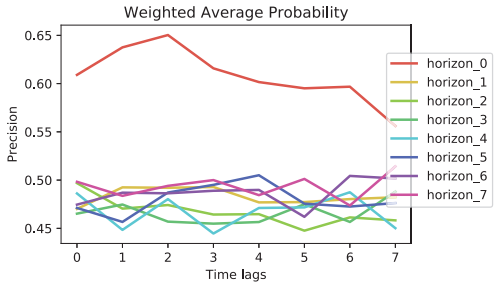
Figure C.5: Full results of localisation evaluation for early recurrent congestion detection in terms of prediction horizons

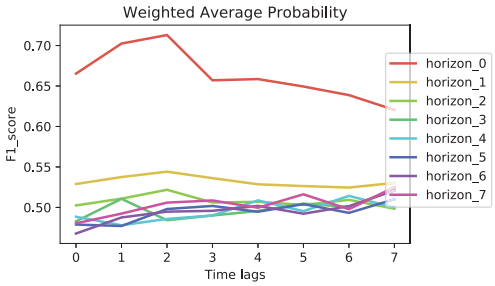

Weighted Average Probability
Conditional Probability
Index
Logistic Regression
Random Forest
Gradient Boosting Classifier
FPR
Time lags
horizon_0
horizon_1
horizon_2
horizon_3
horizon_4
horizon_5
horizon_6
horizon_7

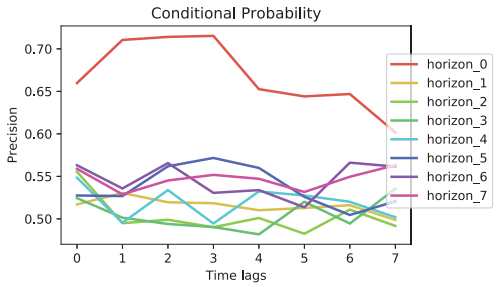
(a) FPR

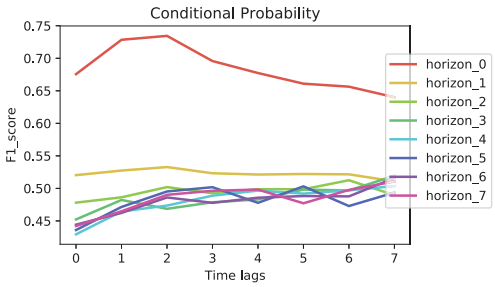

Weighted Average Probability
Conditional Probability
Index
Logistic Regression
Random Forest
Gradient Boosting Classifier
DR
Time lags
horizon_0
horizon_1
horizon_2
horizon_3
horizon_4
horizon_5
horizon_6
horizon_7

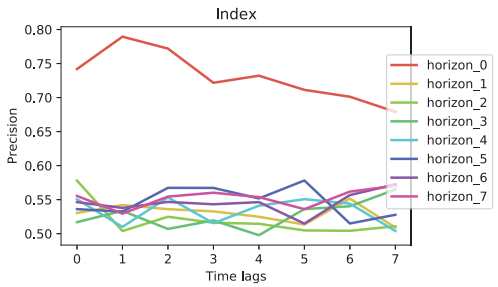
(b) DR

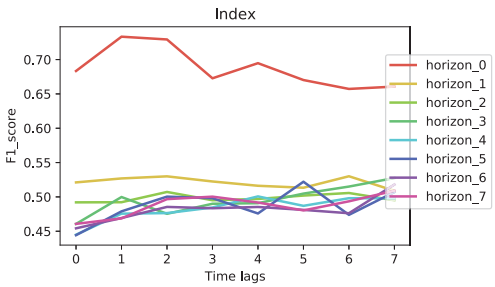


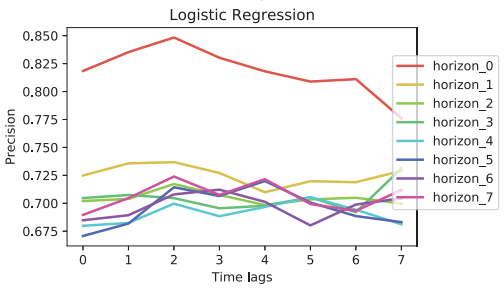


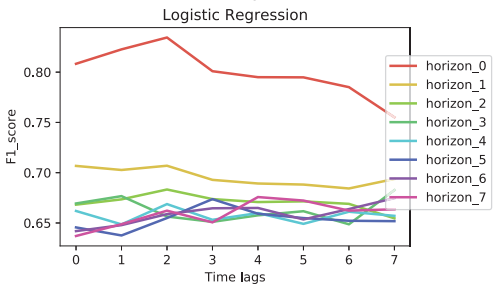


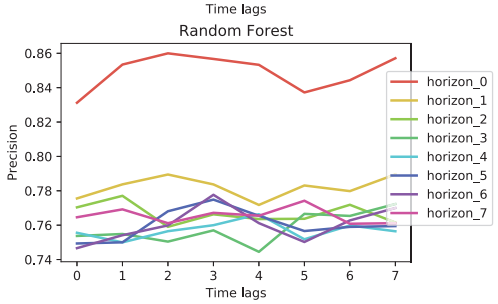


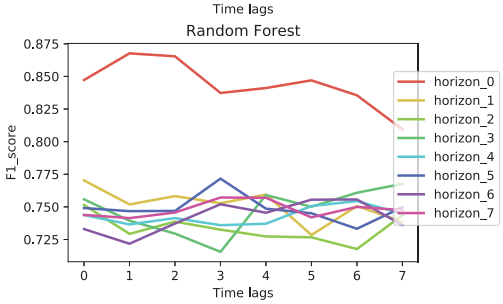


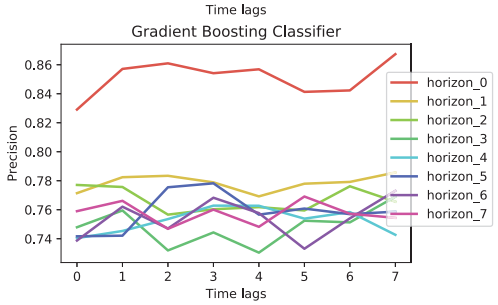


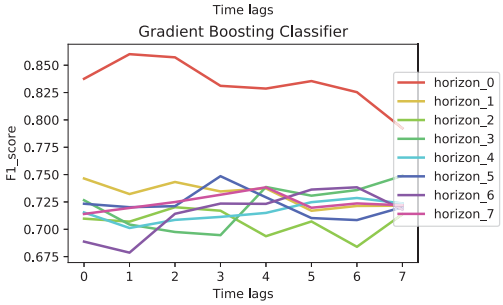






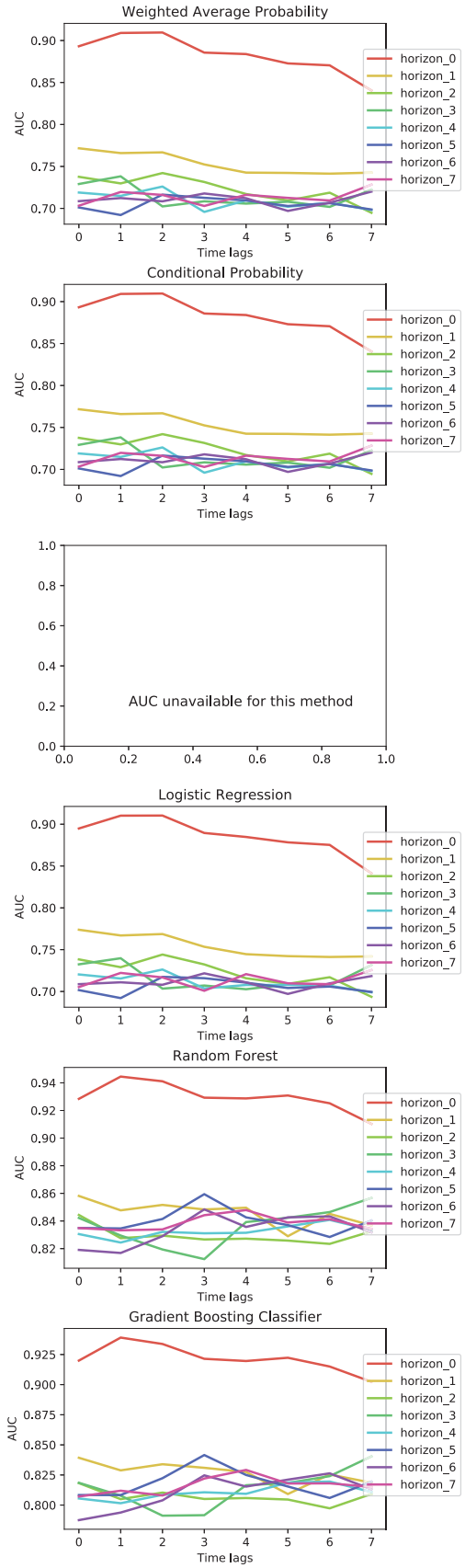






(c) Precision

(d) F1 score



(e) AUC

Figure C.6: Full results of localisation evaluation for early recurrent congestion detection in terms of time lags