SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN

Analyzing Deep Learning Models for Traffic Prediction

Submitted by

Gary GOH Shing Wee

Thesis Advisor

Dr. Alexander BINDER and Dr. LIM Kwan Hui

Information Systems Technology and Design

A thesis submitted to the Singapore University of Technology and Design in fulfillment of the requirement for the degree of Masters of Engineering (Research)

2021

Thesis Examination Committee

TEC Chair:	Prof. Zhou Jianying
Main Advisor:	Prof. Lim Kwan Hui
Co-advisor:	Prof. Alexander Binder
TEC Member 1:	Prof. Berrak Sisman
TEC Member 2:	Prof. Liu Jun

Abstract

Information Systems Technology and Design

Masters of Engineering (Research)

Analyzing Deep Learning Models for Traffic Prediction

by Gary GOH Shing Wee

Traffic prediction problems concern the mobility of human crowds in urban city landscapes. The study of these problems is crucial to improve the efficiency of traffic flows in urban networks and people's quality of life. Deep learning, in recent years, has achieved impeccable results and outperformed traditional statistical methods for solving traffic-related problems in the literature. However, deep learning suffers from poor interpretability. The aim of this dissertation is to investigate deep learning models applied in the traffic domain, and interpret the behaviors of the models by analyzing salient global feature importance and specific input-to-output relationship. First, a literature review of traffic prediction problems and deep learning models is included, followed by a review of analyzer methods for interpreting deep learning models. Second, we explore the usage of Twitter's tweets as a new source of additional information in the new digital age to improve the performance of a crowd flow prediction model *ST-ResNet* on the city-wide crowd flow prediction task, as well as to provide additional context to the prediction in the form of human natural language. Third, we take inspirations from deep learning attribution methods such as *Integrated Gradients* and *SmoothGrad*, and proposed a novel improved method, *SmoothTaylor*, which is derived from the Taylor's theorem. Finally, we discuss future work as we share some preliminary applied results of the above attribution methods on the traffic status prediction for graph-based traffic status prediction model.

Publications

1. Goh, G. S. W., J. Y. Koh, and Y. Zhang (2018). “Twitter-Informed Crowd Flow Prediction”. In: *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 624–631. DOI:10.1109/ICDMW.2018.00097.

2. Goh, G. S. W., S. Lapuschkin, L. Weber, W. Samek, and A. Binder (2021). “Understanding Integrated Gradients with SmoothTaylor for Deep Neural Network Attribution”. In: *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 4949–4956. DOI:10.1109/ICPR48806.2021.9413242.

Acknowledgements

In this page, I would like to give the rightful acknowledgements to the many important people who have made this thesis possible.

Firstly, I attribute this thesis to my lead advisor, Prof. Alexander Binder, who provided me with much guidance and being my main pillar of support throughout the most difficult period of my postgraduate study journey. Without your invaluable support and guidance into the right direction, this thesis would not have existed.

Secondly, I would like to express my gratitude to Prof. Lim Kwan Hui, who is kind to be willing to accept me as your student in very difficult circumstances and support me in achieving my endeavors.

Thirdly, I also like to give thanks to my first advisor, Prof. Yue Zhang, who have guided me greatly in during the start of my postgraduate study journey, and served as a tremendous mentor to me.

I would also like to acknowledgement my external advisor, Prof. Patrick Jaillet, who have provided me with much support even at great distance apart.

I also want to give thanks to Singapore-MIT Alliance for Research and Technology (SMART) for providing me with the funding necessary to support my research.

I am also extremely grateful towards the faculty and staff of Singapore University of Technology and Design's (SUTD) Information Systems Technology and Design (ISTD) pillar how have helped me along my journey. Especially to Sin Chee who have shown great care and understanding towards me, and took the extra time to advise me making the right decisions.

Finally, I would like to express my special appreciation to my dearest wife who have given me unwavering support since the beginning, and being there for me during all my difficult times.

Words cannot express how grateful I am towards everyone who have supported me. Thank you for all your efforts to help me complete this degree. I will forever remember them in my life.

Contents

List of Figures

List of Tables

List of Abbreviations

AI	**A**rtificial **I**ntelligence
ANN	**A**rtificial **N**eural **N**etwork
ARIMA	**A**uto-**r**egressive **I**ntegrated **M**oving **A**verage
ATV	**A**verage **T**otal **V**ariation
AUC	**A**rea **U**nder **C**urve
AUPC	**A**rea **U**nder **P**erturbation **C**urve
AUTVC	**A**rea **U**nder Multi-scaled Average **T**otal **V**ariation **C**urve
BN	**B**ayesian **N**etwork
CAM	**C**lass **A**ctivation **M**apping
CNN	**C**onvolutional **N**eural **N**etwork
CRF	**C**onditional **R**andom **F**ield
DBN	**D**eep **B**elief **N**etwork
DeepLIFT	**D**eep **L**earning **I**mportance **F**ea**T**ures
DCRNN	**D**iffusion **C**onvolutional **R**ecurrent **N**eural **N**etwork
DDoS	**D**istributed **D**enial **o**f **S**ervice
DT	**D**ecision **T**ree
ES	**E**xponential **S**moothing
GCN	**G**raph **C**onvolutional **N**etwork
GPS	**G**lobal **P**ositioning **S**ystem
Grad-CAM	**Grad**ient-weighted **C**lass **A**ctivation **M**apping
GRU	**G**ated **R**ecurrent **U**nit
GSM	**G**lobal **S**ystem for **M**obile Communications
HMM	**H**idden **M**arkov **M**odel
IG	**I**ntegrated **G**radient
ITS	**I**ntelligent **T**ransportation **S**ystem
KF	**K**alman **F**ilter
kNN	**k**-**N**earest **N**eighbor
LIME	**L**ocal **I**nterpretable **M**odel-agnostic **E**xplanations
LRP	**L**ayer-wise **R**elevance **P**ropagation
LSTM	**L**ong **S**hort-**t**erm **M**emory
LVQ	**L**earning **V**ector **Q**uantization
MLP	**M**ulti-**l**ayer **P**erceptron
MSE	**M**ean **S**quared **E**rror
OBNR	**O**nline **B**oosting **N**on-parametric **R**egression
PCA	**P**rincipal **C**omponent **A**nalysis
PoS	**P**arts-**o**f-**s**peech
RMB	**R**estricted **B**oltzmann **M**achine
ReLU	**R**ectified **L**inear **U**nit
ResNet	**Res**idual **Net**work

RF	**R**andom **F**orest
RFN	**R**elational **F**usion **N**etwork
RNN	**R**ecurrent **N**eural **N**etwork
RMSE	**R**oot **M**ean **S**quared **E**rror
SAE	**S**tacked **A**uto-**e**ncoder
SHAP	**SH**apley **A**dditive ex**P**lanations
ST-GCNN	**S**patio-**t**emporal **G**raph **C**onvolutional **N**eural **N**etwork
ST-ResNet	**S**patio-**t**emporal **Res**idual **Net**work
SVR	**S**upport **V**ector **R**egression
TGC-LSTM	**T**raffic **G**raph **C**onvolutional **L**ong **S**hort-**t**erm **M**emory
VLD	**V**ehicular **L**oop **D**etector
xAI	Explainable **A**rtificial **I**ntelligence

Dedicated to my newborn daughter Caris…

Chapter 1

Introduction

This introductory chapter presents a background on transportation systems and the increasing relevance of deep learning applied in this domain and its shortcomings, which form the motivations of this dissertation. A background on the topic is given in Section 1.1. Next, the objectives of the thesis are outlined in Section 1.2, and a description of how the chapters of this thesis is organized is given in Section 1.3.

1.1 Background

Rapid urbanization results in huge population and vehicle growth in cities and increases the toll on transportation infrastructure. Therefore, intelligent transportation system (ITS) is essential to enable efficient mobility of large number of people within urban landscapes. The goal of ITS is to reduce traffic congestion, which translates to lesser time lost, improved productivity, better air quality, and more sustainable energy usage (Wang, Zhang, Liu, et al., 2019). Aided with the advancement of information systems and technology, and the collection of large streams of data from traffic sensors installed all across the transportation network, a modern ITS is expected to be able to analyze and process them at real-time to allow for quick decision making. The central components of an ITS are machine learning models that learn statistical patterns from large datasets which generalizes to learned systems that can accurately and reliably predict the future traffic conditions.

Recent advancements in deep learning push the performance boundaries of Artificial Intelligence (AI) in several applied research areas such as in computer vision and natural language processing. They often outperform traditional statistical machine learning models (e.g. Hidden Markov Model, Conditional Random Field, Decision Trees, Support Vector Machine, Bayesian Network) by a large margin. Thus, it is expected to observe an increased number of studies that propose deep learning models to replace traditional machine learning models in ITS with the aim to boost the accuracy. However, the increase in performance comes at the cost of poor interpretability. Due to the highly non-linearity in deep neural networks, they are often treated as a black box (Figure 1.1). This results in the difficulty to explain for the decisions of deep neural networks, poor input-to-output inference, and eventually lead to the lack of trust between humans and AI systems.

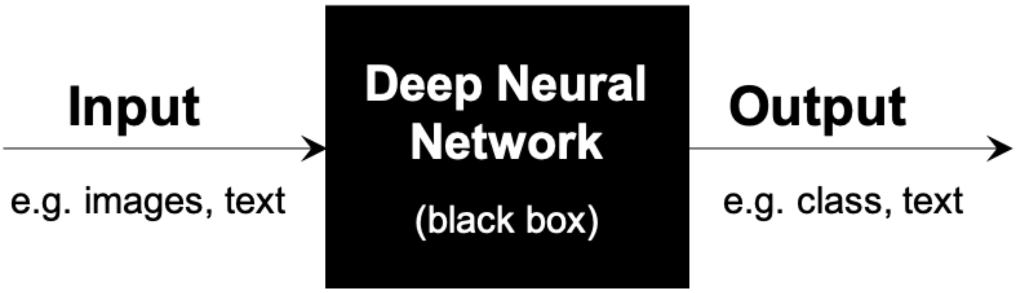


FIGURE 1.1: Deep neural network seen as a black-box, which ignores the intricacy and explanations behind the results of the model and focuses largely on just the input and output.

1.2 Objectives

The overall objective addressed by this dissertation is to analyze deep learning models applied to traffic prediction with two main approaches. First, we explore the utilization of exogenous data sources (e.g. social media, weather, holidays) apart from the traditional data streams from sensors in the network, to not only be used as an additional source of input to the model with the aim to improve model accuracy, but also be used as a way to gain insights and explain for the model's predictions. Second, we examine various Explainable Artificial Intelligence (xAI) methods that are designed to interpret deep neural networks and apply them to produce insights from global input feature importance maps and specific input-to-output attribution heatmaps. In addition, we explore the application of xAI methods on specific deep neural networks that solve traffic-related problems and explain from spatial and temporal points of view. The goal of xAI is to bridge the gap between humans and AI and facilitate the building of trust between them so as to encourage more widespread usage of AI from research into practice.

1.3 Thesis Organization

The rest of the thesis is organized as follows. In Chapter 2, a literature review is presented on machine learning methods that are applicable to the traffic domain, various traffic prediction problem types and the respective models proposed to solve them. xAI methods that are generally applied to deep learning are also reviewed. Next in Chapter 3, we present a study on incorporating tweets from a popular microblogging site, Twitter, and fusing them into the inputs to a deep neural network that solves for crowd flow prediction. In Chapter 4, we refine an existing xAI deep neural network attribution method and propose a novel theoretical algorithm, which we show its effectiveness in the image classification context. Lastly, in Chapter 5, we present preliminary results on the application of the selected xAI methods on a specific deep neural network on the traffic status prediction, and conclude by discussing directions of future work.

Chapter 2

Review of Literature

This chapter presents a literature review on the three main areas of interest identified in this thesis and their related work.

The first Section 2.1 is a review of relevant machine learning methods that are commonly applied to traffic prediction problems. Secondly in Section 2.2, we describe several different types of traffic prediction tasks and review their respective deep neural network models proposed in the literature to solve them. Finally in Section 2.3, a review of Explainable Artificial Intelligence (xAI) methods that are relevant to deep neural networks in general is presented.

2.1 Machine Learning Methods

In this section, we conduct a survey on the general techniques in machine learning models for traffic prediction. We briefly review traditional learning methods, followed by some general architectures found in most deep learning methods, and then focus more on recent developments in deep learning methods in Section 2.2, which have been shown to greatly improve the performance when compared to traditional learning methods.

2.1.1 Traditional Learning

For the prediction of continuous values such as travel times, traffic speeds and traffic flows, the most straightforward approach is to treat it as a linear regression problem (Ide and Sugiyama, 2011; Zheng and Ni, 2013). Apart from the spatial properties of the input, Zheng and Ni (2013) also considers learning different weights to represent the contribution of temporal properties. However, ultimately linear model cannot represent non-linearity in its function and thus limits the capacity. To overcome this limitation, other methods such as Decision tree (DT) (Gal et al., 2017) and Hidden Markov Model (HMM) (Yang, Guo, et al., 2013) are proposed. These methods partition the input space and fit each segment independently, resulting in non-linear overall functions. Additional boosting techniques such as boosting and Random Forests (RF) (Leshem and Ritov, 2007) are used to improve the accuracy.

For time series forecasting, classical methods include Auto-regressive Integrated Moving Average (ARIMA) and Exponential Smoothing (ES) (Ding et al., 2011; Hamed et al., 1995; Li, He, et al., 2017; Tran et al., 2015; Williams and Hoel, 2003; Van Der Voort et al., 1996). Kalman Filter (KF) (Guo, Huang, et al., 2014; Em et al., 2019; Lint, 2008) is shown to produce better performance when the data contains high uncertainty.

Support Vector Regression (SVR) (Jin et al., 2007; Tang et al., 2019; Wu, Ho, et al., 2004; Hong et al., 2010; Castro-Neto et al., 2009; Asif et al., 2014) is also often proposed and shown to produce good accuracy. However, when the traffic data is highly irregular and abnormal, other methods such as k-Nearest Neighbor (k-NN) (Wang, Tang, et al., 2019; Rahmani et al., 2013; Chang et al., 2012) and Online Boosting Non-parametric Regression (OBNR) (Wu, Xie, et al., 2012) can be more effective. These non-parametric methods are more sensitive to short-term variations and thus are able to handle spikes more easily. However, they are unable to effectively leverage on useful spatio-temporal features in the data to predict overall traffic at a large-scale network level.

For traffic classification of discrete values such as determining the mobility status or transportation mode of a subject (e.g. stationary, walking or driving), methods proposed include HMM (Krumm and Horvitz, 2004; Sohn et al., 2006; Zhu, Zheng, et al., 2012; Zheng, Xie, et al., 2008), Conditional Random Field (CRF) (Patterson et al., 2003; Liao, Patterson, et al., 2007), and Bayesian Network (BN) (Yin et al., 2004; Stenneth et al., 2011; Zhu, Peng, et al., 2016). This thesis focus more on prediction of continuous values, and thus we leave the discussion here as related work.

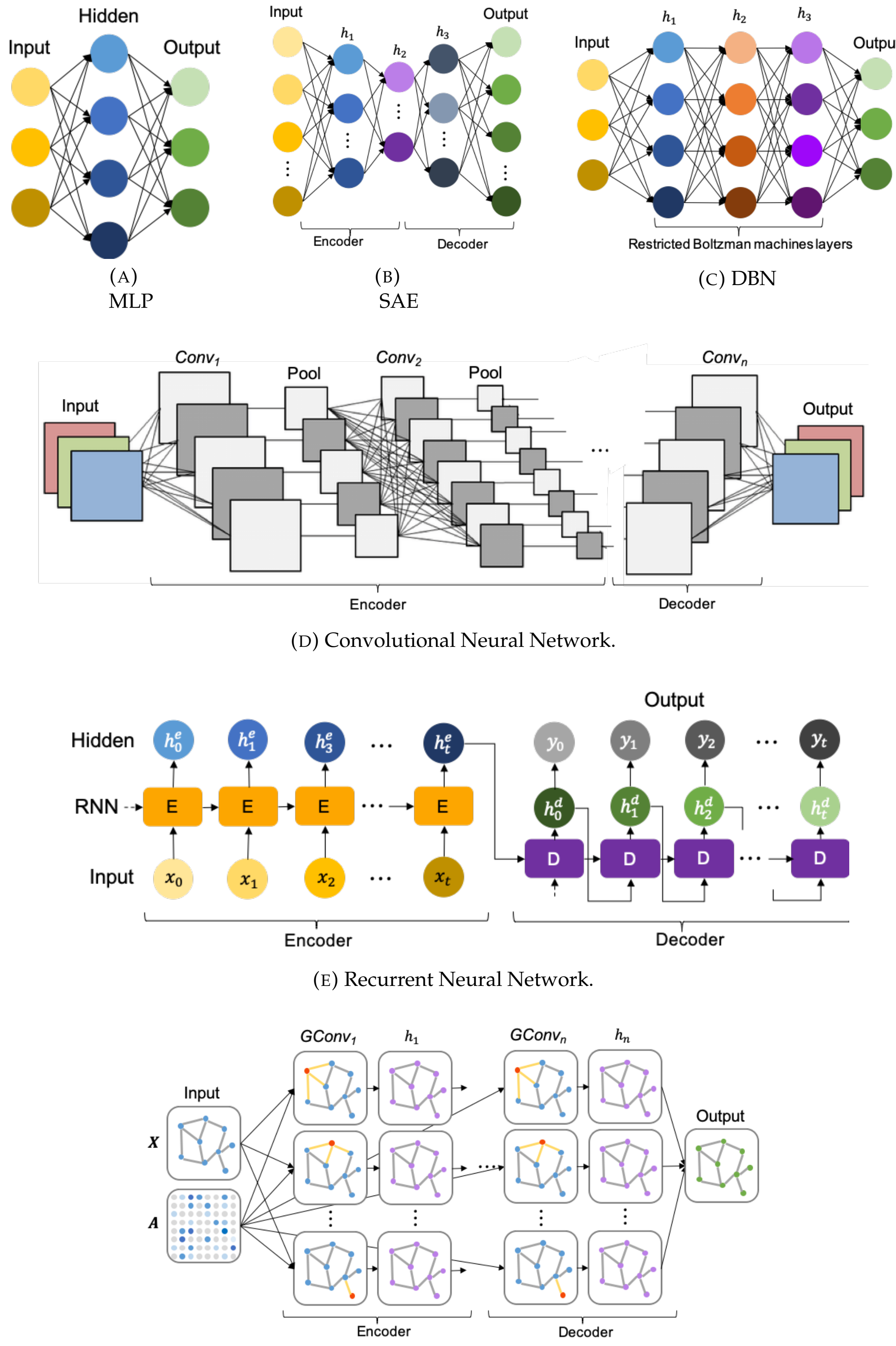
2.1.2 Deep Learning

The recent advancement of deep neural network algorithms, greater access and availability of large data corpus, and improved computing power, have created a conducive environment for deep learning to grow and mature.

Towards the direction of deep learning, early work in the domain of traffic prediction include shallow Artificial Neural Network (ANN) mostly with Multi-layer Perceptron (MLP) (Jindal et al., 2017; Xiaojian and Quan, 2009; Huang and Ran, 1995; Huang, Tang, et al., 2013; Habtie et al., 2015; Akiyama and Inokuchi, 2014; Wang, Cao, et al., 2017; Jiang and Fei, 2015; Zheng and Lee, 2006), a fully-connected feed-forward neural network (see Figure 2.1a) with a minimum of three layers (input, hidden and output layer). However, MLP is unable to represent spatial correlations in traffic networks well. Furthermore, its shallow architecture also makes it difficult to apply on larger-scale prediction problems.

Deep learning methods are able to learn highly dimensional functions with more flexible architecture designs. These designs are able to represent more complex non-linear spatio-temporal dependencies. They also scale more efficiently to predict traffic status for the entire transportation network.

One of such design is Convolutional Neural Network (CNN) (Fukushima and Miyake, 1982; Krizhevsky et al., 2012), is widely popular in the traffic prediction literature (Ma et al., 2017; Wang, Zhang, Cao, et al., 2018) due to its ability to encode spatial dependencies effectively. CNN is extensively popularized in the research area of computer vision. It utilizes multiple kernel filters which slide across the input space, together with convolutional operations followed by a non-linear activation function to detect non-linear pattern across the input feature map and to ultimately detect and ultimately non-linear decision boundaries. Max or average pooling is sometimes used to aggregate information and reduce representation dimensionality. This process repeats multiple times to form depth in the neural network (see Figure 2.1d), allowing further dependencies to be learnt. However, as road networks only occupy a small proportion of the spatial



(A) MLP

(B) SAE

(C) DBN

(D) Convolutional Neural Network.

(E) Recurrent Neural Network.

(F) Graph Convolutional Network.

FIGURE 2.1: Diagrams of various deep learning design architectures for traffic status prediction.

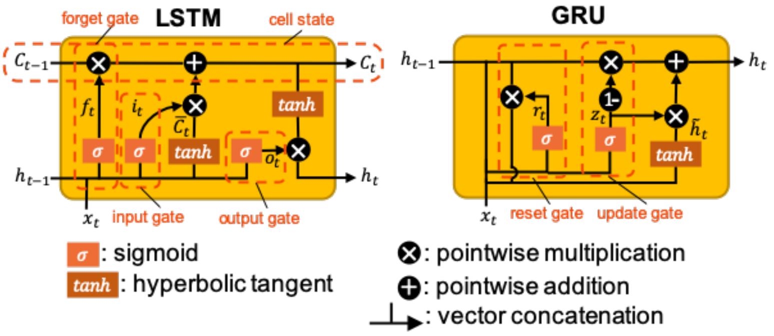


FIGURE 2.2: Illustrations of the LSTM and GRU design.

space, the input matrix is quite sparse. Consequently, training CNN for traffic prediction while covering the whole two-dimensional space in which the roads are sparsely embedded is largely inefficient.

To address the above limitation, a generalization of CNN known as Graph Convolutional Network (GCN) (Zhao, Song, et al., 2020; Geng et al., 2019; Xu and Li, 2019; Bai et al., 2019; Fang et al., 2019; Chen, Chen, et al., 2020; Guo, Lin, et al., 2019; Li, Yu, et al., 2018) is widely proposed for the context of traffic prediction in recent years. Modeling the inherent physical structure of a connected road network as a mathematical graph, together with a pre-defined adjacency matrix, GCN learns a function that maps feature values to output feature for every node in the graph (see Figure 2.1f). This approach is able to train models at a faster rate and achieve better performance over time.

Another popular design is Recurrent Neural Network (RNN) (Elman, 1990), which is intended to model sequential data, and thus very applicable in traffic prediction (Ramakrishnan and Soni, 2018) to encode temporal dependencies. RNN is widely applied to natural language processing tasks such as machine translation, text generation, speech-to-text/ text-to-speech and captioning. RNN links units together by taking the output of a previous unit and feeding it as an additional input to the next, passing on information from past input values (see Figure 2.1e). This mechanism enables RNN to learn long-term dependencies from other past time instances. However, vanilla RNN suffers from the vanishing and/or exploding gradient problem, which hinders the learning process as gradients either diminish to zero or explode to large values, causing the performance to be saturated or to deteriorate quickly during training. Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), a variant of RNN, is proposed to address this limitation. This is done by adding the update gate which regulates the flow of signals out of the unit, and the forget gate which allows the unit to ignore or retain the previous state value from the previous time step. The Gated Recurrent Unit (GRU) (Cho et al., 2014), a simplified variant of LSTM, is also commonly used as it requires less time to train. The illustrations for LSTM and GRU designs are shown in Figure 2.2. In addition, bi-directional variants are often used, where the weights are trained separately in both forward and backward directions (see Figure 2.3), increasing the capacity of the model.

Other related deep neural network models such as Stacked Auto-encoder (SAE) (see Figure 2.1b) (Lv, Duan, et al., 2015) and Deep Belief Network (DBN) (see Figure 2.1c) (Jia et al., 2016; Yang, Dillon, et al., 2017; Huang, Song, et al., 2014; Koesdwiady

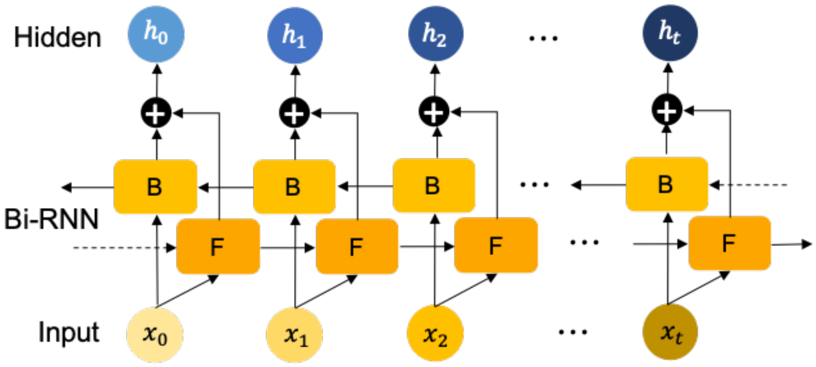


FIGURE 2.3: Illustration bi-directional RNN mechanism.

et al., 2016; Lee et al., 2009) are early implementations of deep learning structures into traffic prediction. However, due to their rigid architecture types and high number of weights required, it does not have enough design flexibility as compared to the more prevalent CNN and LSTM units.

2.2 Traffic Prediction Problems

Traffic prediction covers a wide range of problem types and thus requires various specific deep learning model architectures to handle them. We first focus on the tasks which require forecasts of future traffic conditions, followed by a brief overview of studies done on other traffic-related tasks which are beyond the scope of our study.

2.2.1 Traffic Status Prediction

A frequent traffic management task is the prediction of future traffic status across the network. Usually, the traffic status of a node in a network is either measured by the average traffic speeds of vehicles passing through that point (the slower the speeds, the longer the travel times, resulting in the worsening of the traffic status), or by the number of traffic flows going through that point in a short time interval (the more the flow counts, the higher chance of a congestion). Traffic speeds can be regarded as continuous values to predict (regression), or they can also be categorized into discrete groups of different continuous intervals (classification), while traffic flow counts are discrete integer values.

Traffic flows are traditionally measured by Vehicular Loop Detector (VLD) sensors installed under roads that provide the number of vehicles passing through it during a small time interval. With the increased usage of mobile technology recently, Global System for Mobile Communications (GSM) signals data that provide sequences of Global Positioning System (GPS) points are also commonly used to measure traffic flows and speeds. Some VLD sensors also collect traffic speed data.

The ability to predict which roads will be congested or clear in the future is very useful for applications such as travel time estimation, dynamic traffic control and navigation route planning. This is especially so when there are sudden non-routine huge traffic spikes in small areas that cause major congestion that ripple throughout the network. These spikes are usually caused by events that attract or deter people to or from a localized area, causing excessive traffic in the surroundings.

Across the literature, there are generally two ways to predict network-wide traffic status. The first is network-based, where the aim is to directly predict future traffic status values at each node. The second is region-based, which is dependent on how a city is spatially divided into multiple discrete but adjacent regions, and the traffic flows are counted as the aggregated number of vehicles/people leaving (outflows) and entering (inflows) from one region to another. One advantage of region-based prediction is that it is able to represent traffic status from all spatial regions, while the network-based prediction is constrained to only predict traffic status at locations of the installed sensor nodes. If the sensors are sparsely installed (i.e. far apart from each other) that might be due to cost constraints, then network-based traffic prediction lack sufficient information to effectively model the traffic status in between them. Another advantage of region-based prediction is the increased protection of users' data privacy. With the aggregation of individuals data performed at regional level, specific personal data is lost making backwards inference much more difficult and thus improving data privacy security.

We discuss some of the models presented in the literature from each category in the following.

Network-based

For network-based traffic flow prediction with deep learning models, one of the earlier study is proposed by Lv, Duan, et al. (2015) who adopt SAE in their deep learning model, as well as some others (Yang, Dillon, et al., 2017; Leelavathi and Devi, 2016; Jia et al., 2016). Others propose to use DBN (Jia et al., 2016; Yang, Dillon, et al., 2017; Huang, Song, et al., 2014; Koesdwiady et al., 2016; Lee et al., 2009), which is built by stacking multiple Restricted Boltzmann Machine (RMB). Both SAE and RMB are trained greedily layer by layer with the purpose to capture spatial and temporal dependencies directly. Most of them append a fully-connected MLP at the end of the model as the predictor layer for network-based traffic. However, SAE and DBN are often overly parameterized for the context of traffic status prediction. Thus, newer work in the literature propose to use more sophisticated and efficient deep learning design architectures to further improve the prediction accuracy.

Yu, Li, et al. (2017) utilize multiple stacks of LSTM to predict traffic flows during peak hours. Zhao, Chen, et al. (2017) and Fu et al. (2016) design a 2-dimensional LSTM network, with units from each dimension to encode both temporal and spatial observations. Cui, Ke, et al. (2018) propose a deep stacked bi-directional LSTM to predict traffic speeds, while Wang, Gu, et al. (2016) use CNN with a recurrent layer for error-feedback from previous time steps. Several publications (Lv, Xu, et al., 2018; Wang and Li, 2018; Yu, Wu, et al., 2017; Wu and Tan, 2016; Wu, Tan, et al., 2018) integrate both CNN and RNN into a single hybrid deep neural network model. Specifically, CNN is used to capture topology awareness features (spatial), while LSTM is used to capture periodicity features (temporal). However, training RNN generally takes a long amount of time. In addition, since road networks are quite sparse (i.e. a road usually only connect with other road a few times), using CNN to encode spatial inputs may not be the most efficient.

To address the limitations of RNN and CNN, GCN models are proposed to model the road network directly as graphs, so that there are fewer parameters to learn. Li, Yu, et al. (2018) find a suitable integration of CNN, RNN (GRU) and GCN in their approach known as Diffusion Convolutional Recurrent Neural Network (DCRNN) to predict network traffic speeds. Yu, Wu, et al. (2017) propose a model based on GCN known as Spatio-temporal Graph Convolutional Neural Network (ST-GCNN) and show a 14 times improvement in training speed as compared to DCRNN. Cui, Henrickson, et al. (2019) propose Traffic Graph Convolutional Long Short-Term Memory Neural Network (TGC-LSTM), which introduce k-hops adjacency matrix and a matrix to denote reachable nodes to model traffic impact transmission according to traffic flow theory. Jepsen et al. (2020) propose Relational Fusion Network (RFN), a GCN-based model that takes into account intricacies of road networks such as edge curvature, sharp exit turns, that may affect traffic speeds.

Region-based

As for region-based traffic prediction problems, Zhang, Zheng, et al. (2018) propose a deep CNN model based on Residual Network (ResNet), which is a technique used to mitigate the effects of vanishing gradient problem as CNN models become deeper, to predict city-wide crowd flows. Specifically, the entire city is split via a grid with equal-sized squares, and each snapshot of the city traffic flows is treated as an image where inflows and outflows are viewed like different color channels in images. Similarly, Sun, Wu, et al. (2020) propose a deep learning model based on CNN with residual units to predict future traffic flows but uses taxi's GPS trajectory data instead of traditional sensors to estimate and construct the traffic flow matrices. Ma et al. (2017) also took a similar approach by treating the traffic data on a grid as an image, but for the traffic speed prediction problem. To leverage on temporal dependencies in previous snapshots, other studies (He, Chow, et al., 2019; Yao, Tang, et al., 2019) first apply CNN to encode spatial information in each snapshot, and then apply LSTM to encode temporal information in a sequence of snapshots.

2.2.2 Other Traffic-related Problems

In this section, we briefly summarize some other traffic-related prediction problems that are out of the scope of this thesis now but may be relevant in the future.. One of them is travel time estimation (Jindal et al., 2017; Li, Fu, et al., 2018; Yuan et al., 2020; Wang, Zhang, Cao, et al., 2018; Zhang, Wu, Sun, et al., 2018), which takes an origin-destination pair or a trajectory path on the road network and the departure time as inputs, and predicts the overall traveling time. Another task is the travel demand estimation (Wang, Cao, et al., 2017; Kuang et al., 2019; Geng et al., 2019; Xu and Li, 2019; Bai et al., 2019; Chu et al., 2018; Ye et al., 2021; Yao, Wu, et al., 2018), which predicts the future transportation requests from various regions of a city. This is commonly framed in the context of predicting taxi demand or cab-sharing, with the end goal of optimizing supply to demand allocation. Others include traffic data generation (Song et al., 2019; Wu, Chen, et al., 2017), traffic signal control (Li, Lv, et al., 2016; Van der Pol and Oliehoek, 2016; Gao et al., 2017; Genders and Razavi, 2016), traffic accident prediction (Chen, Song, et al., 2016; Sun, Dubey, et al., 2017; Fouladgar et al., 2017; Zhang, He, et al., 2018; El Hatri and Boumhidi, 2018) and classification of road conditions (Nolte et al., 2018; Ramanna et al., 2021), road signs (Zhang, Huang, et al., 2017; Zeng et al., 2017; Li, Møgelmose, et al., 2016; Li and Yang, 2016) and mode of transportation (Liu and Lee, 2017; Qin et al., 2019; Wang, Luo, et al., 2020; Liu, Wu, et al., 2019). Some of the above mentioned problems are crucial for the application of autonomous vehicles which is a highly foreseeable technology to be applied in the future in a large-scale manner.

2.3 Explainable Artificial Intelligence

Explainable Artificial Intelligence (xAI) for deep learning is of paramount importance towards its widespread adoption, as it aims to provide answers to the biggest shortcoming of deep learning - the lack of transparency and poor interpretability. Even though deep neural networks are able to consistently perform at high accuracy levels, it is still crucial to understand how they work correctly or incorrectly, and why they arrived at a particular prediction. The non-linear mappings inside deep neural networks obfuscate the relationship between the input and output. As a result, the lack of a human user understanding of the internal functions of these models hinder efforts to debug errors, seek for design improvement ideas, and also the validation of predictions.

There are several ways in which undesirable biases and errors can be introduced into the model without our knowledge at various stages of model learning pipeline. At data collection, the data may be contaminated as not all the data in the train datasets are guaranteed to be of the best quality since the data labeling process is highly susceptible to human errors. During training, there might be hidden bugs in the model implementation that results in defective model parameters. At test time, out-of-distribution or out-of-domain test inputs result in potentially erroneous predictions, as those outlier inputs are not well supported by training samples from which the network can generalize towards the outliers.

Explanation methods provide deep machine learning models the ability to explain their complex behaviors in understandable terms to humans, which helps to establish human users trust in deep learning systems. They also serve as good guides to researchers and engineers to better understand the models, the problem that the models aim to solve, and the datasets used for training and testing.

There are generally two explanation approaches for deep learning xAI: intrinsically explainable and post-hoc explainable (Du et al., 2019). Intrinsic explainability is about building a new model with improved explainability, possibly deviating from design choices that would be made if only prediction accuracy would be of interest, while post-hoc takes the model as is and tries to derive an explanation after training. There are also two types of explanations: global and local. Global explanations describe what input features are important to the model in general, while local explanations describe how the model arrived at a specific output for a specific input.

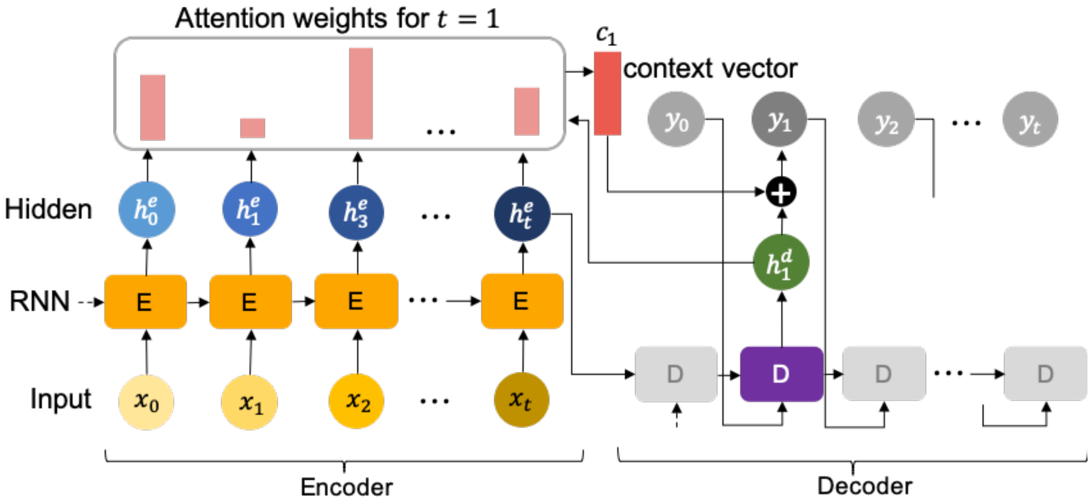


FIGURE 2.4: Illustration the attention mechanism.

We review the different explanation methods based on the above categorization in the following sections.

2.3.1 Intrinsic Explainable Methods

Intrinsic explainable methods are self-explanatory components that are built into the deep neural network inner structure which are designed to be more easily interpretable. These added components are trained together with the models' initial parameters. Thus, it may impact the model prediction performance for better or worse depending on several conditions such the datasets and/or model design, therefore limiting its general applicability.

Intrinsic methods that generate global explanations are mostly implemented by adding interpretable constraints into the model. Sabour et al. (2017) introduce a group of neurons known as capsules to augment deep neural networks. The activation vectors of active capsules are able to semantically represent certain humanly understandable concepts which allow users to verify useful patterns that the models also capitalize on. Zhang, Wu, and Zhu (2018) add a regularization loss in filters at high-level convolution layers in CNN-based models known as interpretable CNN, so that users can understand how the CNN memorize certain patterns but at a small cost of performance.

A widely utilized approach for generating local explanations intrinsically from deep learning models is the use of attention weights generated from the attention mechanism (Mnih et al., 2014; Bahdanau et al., 2016; Rayhan and Hashem, 2020; Xie et al., 2021). The attention component maps the importance parts of the input features by learning to assign higher weights to parts that help to improve accuracy (see Figure 2.4). Visualizing the weights for specific input-output pair allows users to interpret which part of the input is being focused most by the model. However, they also extend the number of learnable model parameters, which increase training time especially if the input sequences are long.

Lastly in this category, there are also prototype-based methods (Sato and Yamada, 1995; Schölkopf and Smola, 2003; Chen, Song, et al., 2016), or also known as Learning Vector Quantization (LVQ). Prototypes are learnt by selecting training data samples that can best represent the class it belongs. Prototype-based models then predict by calculating similarity scores of a given input to the prototypes and assign the the label of the most similar prototype as the output. Given an unseen test input, these methods identify the top most similar (based on some distance measure) prototypes which a human user can then use as references by example to better understand how this input relates to the predicted output.

2.3.2 Post-hoc Explainable Methods

Post-hoc explainable methods analyze deep neural networks after they are fully pretrained, and do not interfere with the learning process. The aim is to extract useful patterns that have been learnt by the models and can be found in the model's parameters or learned representations.

There are visualization techniques proposed to translate learned global representations into an intuitive format for human users to understand. One of the earlier work is done by Simonyan et al. (2014) who introduce a technique to construct visualizations in the input space based on the gradients of the output neurons that is calculated using back-propagation with respect to the input. Several other work follow suit in the same direction with little suggested improvements (Yosinski et al., 2015; Wei et al., 2015). Mahendran and Vedaldi (2016) and Mordvintsev et al. (2015) also create visualizations that can help users to understand how a deep learning model learns by reconstructing input images from layer activations rather than to explain a prediction. However, their visualizations often contain unnatural colors or repeated recognizable image fragments that do not fit as a coherent whole (see Figure 8 in Nguyen, Yosinski, et al. (2016)), resulting in anomalous images as explanations. Nguyen, Yosinski, et al. (2016) improve the quality of the visualizations to a great extent by considering the multi-faceted nature of high layer neurons. There are also studies done to interpret encoded intermediate layers within a pre-trained deep CNN (Zhang and Zhu, 2018; Erhan et al., 2009). Similarly, Aubry and Russell (2015) visualize the importance of preselected input feature maps by measuring the responses of intermediate layers, while Olah et al. (2017) generates visualizations for convolution layers which activate the output neuron. Lu (2015) apply Principal Component Analysis (PCA) on the outputs of deep CNN models and visualize the class embeddings which highlight the presence of semantic concepts learnt by the models. An alternate approach to visualize feature representation is the use of up-convolution (Dosovitskiy and Brox, 2016), which relies on training a new neural network that takes in a feature map as input and reconstruct a corresponding image as an explanation of the feature map.

Apart from visualizing CNN-based models, there are also several work that visualize useful representations learnt by RNN (Kádár et al., 2017; Karpathy et al., 2016). The last encoded hidden state of RNN is studied and shown to contain meaningful complex semantic concepts.

An alternative generic method to analyze global behaviors of a complex model is known as model extraction (Bastani et al., 2019; Kazhdan et al., 2020; Zhang, Yang, et al., 2019). The aim of model extraction is to approximate a complex model using an easily interpretable model (for e.g. linear regression, decision tree). With a good enough approximation with comparable prediction accuracy, the behavior of the complex model can be understood through the parameters of an easily interpretable model.

To analyze local input-to-output behaviors, one idea is to view it as a feature attribution problem. These methods assign a relevance score for each input feature that contributes towards a model's output. A class of model-agnostic gradient-based approaches include using gradient map (Baehrens et al., 2010; Simonyan et al., 2014), Integrated Gradients (IG) (Sundararajan et al., 2017), SmoothGrad (Smilkov et al., 2017) and its variants (e.g. VarGrad (Adebayo et al., 2018), SmoothGrad Squared (Hooker et al., 2019)), Input-Grad (Shrikumar et al., 2017), Expected Gradients (Erion et al., 2020), and SmoothTaylor (Goh et al., 2021). Some model-specifc methods include Class Activation Mapping (CAM) (Zhou et al., 2016) and Gradient-weighted CAM (Grad-CAM) (Selvaraju et al., 2017).

Another class of method involves calculating the relevance scores using a modified version of back-propagation. These methods include DeConvNet (Zeiler and Fergus, 2014; Springenberg et al., 2015), Guided Backpropagation (Springenberg et al., 2015), Layer-wise Relevance Propagation (LRP) (Bach et al., 2015; Binder et al., 2016), Deep Learning Important FeaTures (DeepLIFT) (Shrikumar et al., 2017), DeepTaylor (Montavon et al., 2017) and PatternNet (Kindermans et al., 2018).

Ribeiro et al. (2016) propose a model-agnostic method known as Local Interpretable Model-agnostic Explanation (LIME), which approximates an interpretable linear model locally around a selected input, and uses the weights of the linear model based on its input features to generate local explanations. Similarly, Lundberg and Lee (2017) present SHapley Additive exPlanations (SHAP) which also locally approximates a complex model function around the specific input and measures the impact of dropping a feature onto the prediction.

Another type of model-agnostic method to generate local explanations that is often used is perturbation-based explanation. This line of work (Zeiler and Fergus, 2014; Ancona et al., 2017) measures the importance of an input feature by measuring how much the final prediction scores change when the feature is perturbed (or omitted/occluded to represent unknown information). However, perturbation-based methods are often very computationally expensive especially when the dimension of the input is huge, since the features are perturbed one at at time. An alternative model-specific approach through mask perturbation (Fong and Vedaldi, 2017) address the above limitation. A new deep neural network can also be trained (Dabkowski and Gal, 2017) to predict the attribution mask so as to improve the efficiency of the perturbation-based explanation method.

The computation of specific adversarial samples for deep learning models is a great tool to understand how incorrect predictions are caused by vulnerable points in the input space (Szegedy et al., 2014; Su et al., 2019; Koh and Liang, 2017). Similarly, Zhang, Wang, et al. (2018) propose a method to identify potential biased representations in CNN. With deeper understanding of adversarial or biased samples, it can help researchers to detect and fix errors in the training data as well the model implementation.

Chapter 3

Twitter-informed Crowd Flow Prediction

3.1 Introduction

Fine-grained crowd flow prediction within a city is valuable for traffic control management and could improve travelling experience and public safety. Crowd flows refers to traffic flows that are aggregated spatially over a region in a city, and temporally over a time interval. Accurate knowledge of future crowd flows could lead to better travel time estimations and more optimized route selection during navigation for general users (Niu et al., 2015). It could also facilitate governments and/or urban city planners to strategize and enforce targeted traffic control measures in advance to curb the level of congestion in the city, and could potentially be very useful in averting overcrowding situations in specific regions. For example, in September 2017, a huge crowd of people gathered together at a train station in Mumbai on a rainy morning rush hour when four trains arrived simultaneously, resulting in a tragic stampeded that killed 23 people. 36 people also died in a stampede during Shanghai's 2015 New Year's Eve celebration. These tragedies could have been avoided or at least mitigated if authorities are advised with future crowd flow predictions and take early preventive measures, such as setting up blockades, broadcasting warnings, or conducting evacuations.

Many studies have been conducted on the traffic flows prediction problem, and recent work achieves relatively reasonable accuracy (Abadi et al., 2015; Ni et al., 2014; Xu, Kong, et al., 2014; Zhang, Zheng, et al., 2018). These approaches focus on capturing patterns from historical observations of traffic flows to predict future observations. Due to the nature that traffic flows are largely periodical, such as the predictable peaks in the morning and evening rush hours, relying on past observations is generally effective. However, the poor predictive performance arises when there are non-recurring events that can influence large-scale crowd movement, which cannot be inferred from historical data. Examples of such events include, traffic incidents, road closures, road works, sports events, musical concerts, celebratory events, or any other events that cause sudden interests in particular regions such as the sudden congregation of "Pokemon Go" players in specific random spawn locations to catch rare in-game creatures. These events can be rare and only affect small regions in short time intervals, yet it is especially during these critical periods, that accentuates the need for more accurate crowd flow predictions so that the relevant authorities can respond timelier to the situation.

We consider utilizing real-time texts from the Internet, which can contain information on such critical non-recurring events, by feeding them as additional inputs to an existing crowd flow prediction baseline model. More specifically, we focus on tweets to represent non-recurring crowd flows influencing information, as it has been demonstrated that Twitter is able to react to news events more quickly when compared with traditional media (Petrović et al., 2010). It presents a huge well of untapped freely available information, which explains the extensive research attention on tweets information extraction in recent years. In this chapter, we aim to address the following research questions:

1. Can tweets be useful for crowd flow prediction?
2. How are tweets related to traffic / crowd flows?

We analytically answer the above questions through conducting empirical experiments in the context of Singapore, experimenting with two traffic flow datasets – Vehicular Loop Detector (VLD) signals across the city which measure the number of vehicles passing on roads, and Global System for Mobile Communications (GSM) signals which measure the number of people moving from point to point.

We employ the Spatio-temporal Residual Network (*ST-ResNet*) crowd flow prediction model, as proposed by Zhang, Zheng, et al. (2018), as our baseline. *ST-ResNet* is an end-to-end deep neural network predictive model built to forecast the citywide crowd flows, which has been shown useful for predicting crowd flows in Beijing and New York, but not yet for Singapore. Its design is also highly flexible and allows easy integration of additional inputs.

Incorporating tweets as inputs to an end-to-end structured prediction model is challenging due to the large presence of noise found in unstructured text. Some tweets may be irrelevant to the search keywords used to extract them. In addition, efficiency is a crucial factor for real-time metropolitan-level crowd flow prediction. Thus, we explore several efficient linguistic features such as tweet counts, tweet tenses and tweet sentiments, which are relatively insensitive to noise, and extend upon our baseline model to receive tweet information as additional inputs. Results over four years of data suggest that tweet information is indeed relevant to traffic, significantly reducing prediction errors for both road traffic and mobile phone signals. We additionally find that people tend to tweet more nearing relevant traffic-influencing events, which suggests that tweets are good indicators to crowd flows. To our knowledge, we are the first to investigate the usage of tweets to the crowd flow prediction task. Regrettably, we are unable to release the traffic flow datasets due to confidentiality issues. However, our code for the extended model and the tweets dataset are publicly available[1], and we strongly encourage readers to reproduce our results in the context of other cities.

The rest of the chapter is organized as follows. Section 3.2 introduces our problem statement formally along with the datasets used in our experiments. Section 3.3 summarizes the internal structures of *ST-ResNet* and how it is configured to take tweets inputs. The experiment settings are detailed in Section 3.4, along with the results and our discussions. Section 3.5 presents related work in this field. Finally, Section 3.6 concludes this chapter.

[1] https://github.com/garygsw/twitter-crowd-flow-prediction

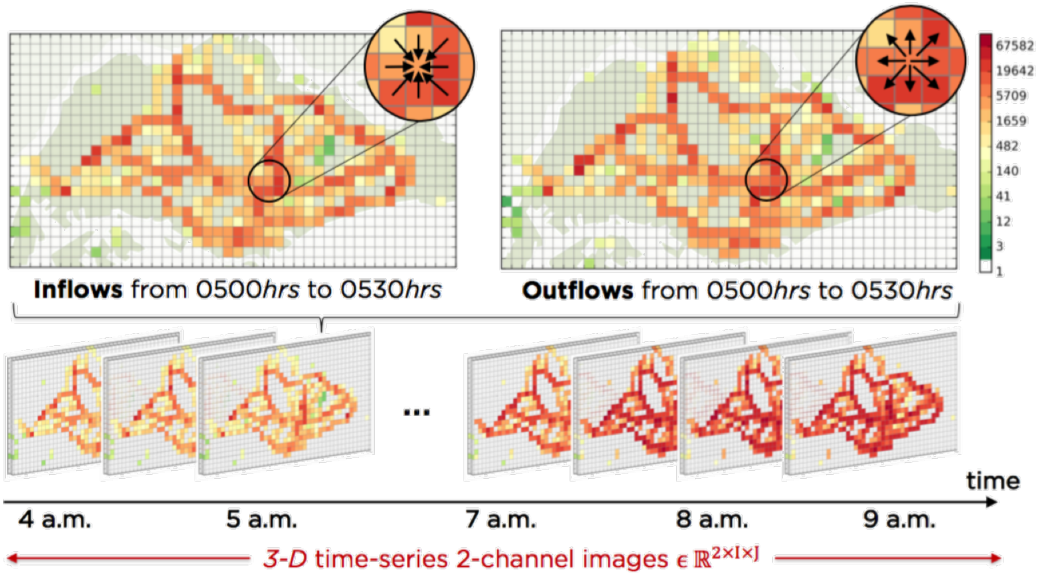

67582
19642
5709
1659
482
140
41
12
3
1
Inflows from 0500*hrs* to 0530*hrs*
Outflows from 0500*hrs* to 0530*hrs*
...
time
4 a.m.
5 a.m.
7 a.m.
8 a.m.
9 a.m.
3-D time-series 2-channel images ϵ ℝ^2×I×J

FIGURE 3.1: Visual representation of spatio-temporal crowd flows with a time series of 3D 2-channel images.

3.2 Problem Statement

There are numerous ways to define spatial regions, but for the context of our study, we use an evenly-spaced grid map of dimensions $I \times J$ to partition a city, where each grid cell denotes a spatial region. The size of the map is based on the limits of the latitudes and longitudes. Each of the grid cell contains two types of crowd flows: inflows and outflows, as illustrated in Figure 3.1. Together, crowd flows for every region within a time interval can be represented by a 3-dimensional image-like matrix with 2 channels; one for inflow and the other for outflow.

At the t^{th} time interval, the crowd flows in all $I \times J$ regions is denoted as a tensor $\mathbf{X}_t \in \mathbb{R}^{2\times I\times J}$ where $(\mathbf{X}_t)_{0,i,j} = x_t^{in,i,j}$ denotes the inflows and $(\mathbf{X}_t)_{1,i,j} = x_t^{out,i,j}$ denotes the outflows. The crowd flow prediction problem becomes a rolling horizon time series prediction task, where the aim is to predict the next time interval's image. Formally, the crowd flow prediction problem is defined as: given historical observations $\{\mathbf{X}_t | t = 0, ..., n-1\}$ and any additional inputs from external factors, predict $\mathbf{X}_n$.

For our experiments, we used two different sets of citywide traffic flow data from Singapore to construct the crowd flow tensors. Their metadata is tabulated in Table 3.1. We use multiple datasets from different time spans so as to validate the performance of the predictive model, since cross validation is not feasible for the rolling horizon prediction problem

In addition, we collected weather and public holidays datasets, as well as the extracted tweets information from a set of tweets. All of these datasets corresponds to the time span of the traffic flow datasets. The dataset preparation process is as follows:

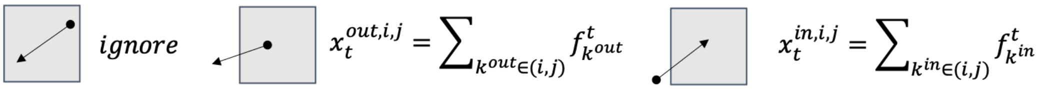
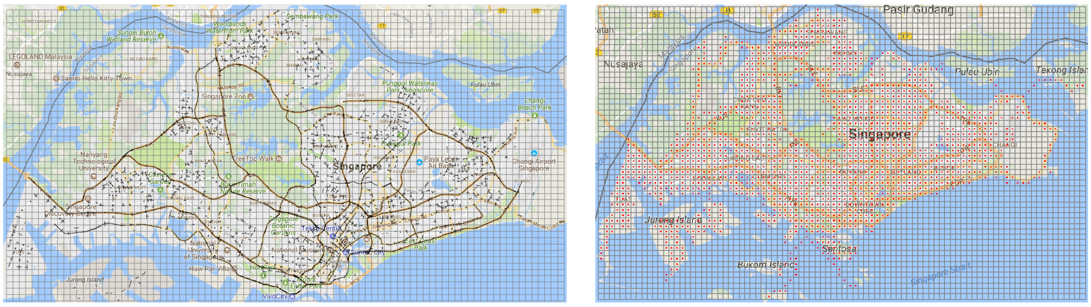

ignore
$x_t^{out,i,j} = \sum_{k^{out} \in (i,j)} f_{k^{out}}^t$
$x_t^{in,i,j} = \sum_{k^{in} \in (i,j)} f_{k^{in}}^t$

FIGURE 3.2: Aggregation of the type of flows depending on direction and position of VLD sensor. Here, f_k^t represents the flow in road link k at time interval t.



(A) Road links with VLD sensors.

(B) Traversable grids.

FIGURE 3.3: Singapore map overlaid with the grid map.

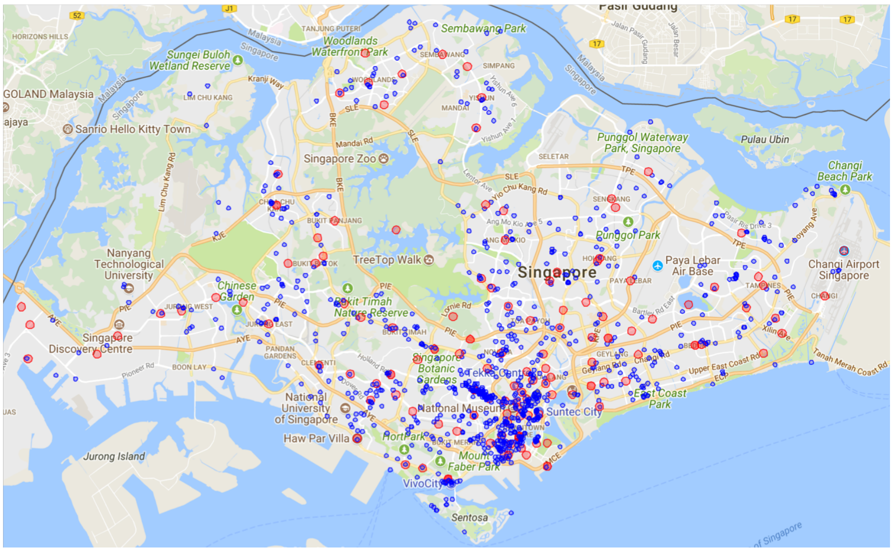


FIGURE 3.4: Singapore map overlaid with selected locations of with tweet mentions extracted. Red points denote train stations; others are in blue.

3.2.1 Vehicular Loop Detector Sensors

We aggregate signals from over 57,000 VLD sensors that are installed across major road intersections and expressways in Singapore, as shown in Figure 3.3a. These sensors count the number of vehicles that passes through each point.

Let $\mathbb{Q}$ be a collection of VLD signals at the t^{th} time interval. For grid cell (i, j)t that refers to the region in the i^{th} row and the j^{th} column, the aggregated inflows and outflows at the time interval t are defined respectively as:

$$\mathbf{x}_t^{in,i,j} = \sum_{q \in \mathbb{Q}} |q|, \quad q_s \notin (i,j) \wedge q_e \in (i,j) \tag{3.1}$$

$$\mathbf{x}_t^{out,i,j} = \sum_{q \in \mathbb{Q}} |q|, \quad q_s \in (i,j) \wedge q_e \notin (i,j) \tag{3.2}$$

where $|q|$ is the value of the VLD signal in $\mathbb{Q}$; $q_s \to q_e$ represents a VLD signal starting from point q_s and ending with point q_e; $q_i \in (i, j$ means that point q_i lies within grid cell $(i, j$, and vice versa (see Figure 3.2).

3.2.2 Global System for Mobile Communications Signals

We aggregate GSM communication signals, where the location shifts of mobile phone users are estimated based on the closest cellular tower connected to their phones. As the origin and destination points could be situated quite far away, there is a need to infer their most likely trajectories in their path through the grid in order to construct the crowd flow tensors. A map of traversable regions is marked out so as to ensure all points are reachable by any point, as shown in Figure 3.3b. Subsequently, a breadth-first search shortest path algorithm is used to infer the trajectories for every origin-destination pairs.

Let $\mathbb{P}$ be a collection of trajectories at the t^{th} time interval. For grid cell (i, j), the aggregated inflows and outflows at the time interval t are defined respectively as:

$$\mathbf{x}_t^{in,i,j} = \sum_{Tr \in \mathbb{P}} |p|, \quad \forall k > 1, \quad g_{k-1} \notin (i,j) \wedge g_k \in (i,j) \tag{3.3}$$

$$\mathbf{x}_t^{out,i,j} = \sum_{Tr \in \mathbb{P}} |p|, \quad \forall k > 1, \quad g_{k-1} \in (i,j) \wedge g_k \notin (i,j) \tag{3.4}$$

where $Tr : g_1 \to g_2 \to ... \to g_{|Tr|}$ is a trajectory in $\mathbb{P}$, and g_k are points along the trajectory; $|p|$ is the number of people travelling on trajectory Tr.

3.2.3 External Data

Weather information is scrapped from a website[2] which provides historical hourly weather information. The sub-factors include temperature, wind speeds, and one-hot-vectors to represent one of the 8 different weather conditions – sunny, cloudy, overcast, rain, light rain, heavy rain, fog and haze. The public holidays and weekends can be

[2] www.timeanddate.com

TABLE 3.1: Crowd flow datasets description.

Dataset	VLD	GSM
Datatype	Vehicular flow counts	Origin-destination pairs
Timespan	Set 1: 1 Mar 2013 - 30 Jun 2013 Set 2: 1 Sep 2014 - 31 Dec 2014 Set 3: 1 Dec 2015 - 31 Mar 2016	1 Aug 2017 - 30 Nov 2017
# of time intervals (T)	5,856	1,464
Grid map size ($I \times J$)	(89, 49)	(90, 54)

TABLE 3.2: Examples of Twitter search keywords used.

Train stations	Points of interests	Estate names
Braddell	Esplanade Theatre	Serangoon
Dhoby Ghaut	Marina Bay	Tampines
Outram Park	Singapore Indoor Stadium	Clementi

inferred from the calendar and is encode by a binary vector that correspond to every time interval.

3.2.4 Tweets Data

Our tweets are collected based on a set of 369 search keywords that are appropriately selected to cover key regions within Singapore, extracting a total of 1.28 million tweets. Several keywords are shown in Table 3.2. The full list of search keywords is attached together with our code.

These keywords are based on the names of locations with high capacity to hold large crowds, as well as names of regions which are representative of localized regions, such as train stations names, estate names, towns, campuses etc. The spatial distribution of these locations are shown in Figure 3.4. The tweets are aggregated spatially based on the latitudes and longitudes of the location of its search keyword, and temporally based on its creation time.

Here we have a $T \times I \times J$ matrix containing a set of tweets relevant to each grid cell (i, j), where T is the total number of time intervals in our datasets.

3.3 Model

An overview of the *ST-ResNet* model's architecture is shown in Figure 3.5. The aim is to predict the next crowd flow at time t. The original model comprises of four components. The first three components in the middle possess the exact same structure, but each of them models the spatio-temporal correlations from historical observations at different time granularity – weekly, daily and hourly. The fourth component, shown on the left of Figure 3.5, considers external factors that affect the crowd flows across the entire city, such as the weather, day of the week and public holidays. We extend the model by adding a fifth component, which extracts a set of features from a tweet stream

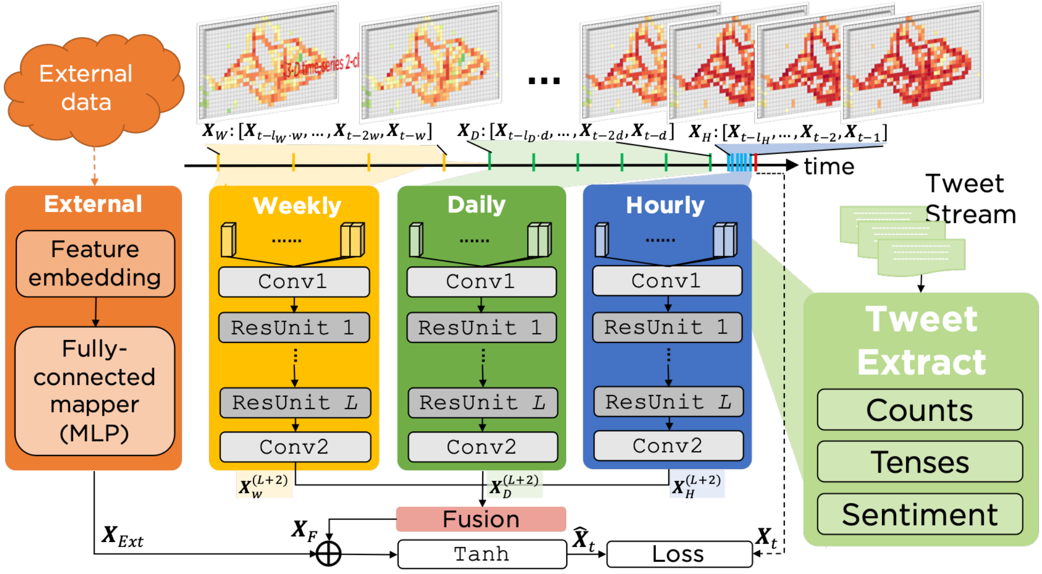


FIGURE 3.5: *ST-ResNet* overall architecture with extended tweet extract component.

and appends them to the inputs of the hourly component before feeding it through the neural networks. The following sub-sections describes each component more attentively, but we refer readers to the paper of Zhang, Zheng, et al. (2018) for more details.

3.3.1 Historical Observations Component

The weekly, daily and hourly components takes in an ordered concatenated series of crowd flow matrices from past observations but at three different lengths of time intervals apart denoted by W, D and H respectively. The size of each of the sequence is parameterized by l_W, l_D and l_H. The idea behind the choice of these time granularity originates from how traffic patterns are usually found in these time intervals. These two main elements form the internal structure of this component:

Convolutions

The idea is to stack multiple Convolutional Neural Network (CNN) to explore spatio-temporal correlations between nearby and distant inflows and outflows across different historical time intervals. Each convolutional mapping is denoted by *Conv*, and is defined as:

$$\mathbf{X}^{i+1} = f(\mathbf{W}^{(i)} * \mathbf{X}^{(i} + \mathbf{b}^{(i)}) \tag{3.5}$$

where $*$ denotes the convolution function; f is the Rectified Linear Unit (ReLU) activation function; $\mathbf{W}^{(i)}$ and $\mathbf{b}^{(i)}$ are learnable parameters; and i is the index of each stack.

Residual Units

To be able to stack multiple CNN without incurring training degradation, L number of residual units are used, where each unit denoted by ResUnit. Each residual unit defines the mapping as follows:

$$\mathbf{X}^{i+1} = \mathbf{X}^{l} + \mathcal{F}(\mathbf{X}^{l}; \theta^{(l)}), \quad l = 1, ..., L \tag{3.6}$$

where $\mathcal{F}$ is the residual function, where it contains two stacks of convolution with ReLU, and $\theta^{(l)}$ includes all learnable parameters in the l^{th} residual unit.

Following the original implementation of the model, batch normalization is also added before applying ReLU. The final part of each component is joined by a last convolution layer *Conv2* where the dimensions of the outputs will match the original dimensions of the crowd flow tensor. The outputs for each component are denoted by $\mathbf{X}_{H}^{(L+2)}$, $\mathbf{X}_{D}^{(L+2)}$ and $\mathbf{X}_{W}^{(L+2)}$.

3.3.2 External Component

The external component considers exogenous knowledge which have been shown to be crucial in influencing future crowd flows. Crowd flows during public holidays or weekends can be considerably different compared to flows during normal work days or weekdays. Weather also plays an important factor in determining the behavior of crowd flows. Lagged weather conditions (i.e. weather at $t-1$) is used to forecast the crowd flows at time interval t.

3.3.3 Integrating Tweet Features

In this component, we extract features from real-time tweets that might be relevant in helping explain for non-recurring crowd flows by introducing two new parameters, denoted by $(lag^{-}, lead^{+}$ which represent a time interval window of tweets to be included in the inputs. In particular, lag^{-} refers to the number of time intervals before t during which the tweet features are extracted, while $lead^{+}$ refers to the number of time intervals from t onwards. The size of the source tweet information window is thus $lag^{-} + lead^{+}$. For real-time prediction, it would be infeasible to consider any $lead^{+} > 0$ since it will be unknown at time interval $t-1$. However, we included $lead^{+}$ in our experiments as we intend to analyze potential relationships that future tweets might have with future crowd flows. The features that are extracted from tweets include:

Tweet Counts

The simplest way to represent the level of interests in a particular location is to simply track the counts of tweets that refers to the region. Based on our hypothesis, if the tweet counts from a particular grid cell is high at some specific time interval, crowd flows from nearby cells should be higher than normal. The intuition is that if a particular region gains interests as measured by tweet counts, crowds are more likely to flow there. For example, as shown in Figure 3.6 and Table 3.3, there is a spike in the counts of tweets that has specific mentions of "Jalan Besar Stadium", which turns out to be referencing a large-scale soccer match with higher levels of induced crowd flows into the region throughout the day. This feature is denoted as $\mathbf{T}_c \in \mathbb{R}^{I \times J}$.

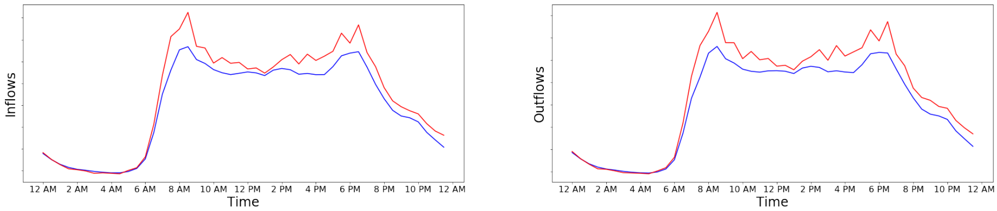


FIGURE 3.6: Jalan Besar case study: Crowd flows in nearby regions during a present large-scale event on 1 March 2013 around 7.30pm. Crowd flows are higher than expected. Note: Blue lines represent daily mean flows, while the red lines represent specific flows on that day.

Time	**Tweet text**
17:38	Now at Jalan Besar stadium alr!!
17:34	Now going to jalan besar for ball picker
19:12	Finally the jam cleared. Now en route to Jalan Besar !!
19:34	Kickoff at the Jalan Besar Stadium - Albirex Niigata (S) 0-0 Home United! #SLeague
19:50	A bit late but thrilled to be watching #ALB v #HUFC @Jalan Besar Stadium

TABLE 3.3: Jalan Besar case study: sample of the relevant tweets that reveal that a major event is ongoing at a particular location.

Tweet Tenses

Although the tweets are collected on a real-time basis, the tweets might not refer to an event that is happening at present time. People might tweet about some event that has already happened in the past, or will only happen in the future. For example, as shown in Figure 3.7 and Table 3.4, there was a huge spike of tweets with mentions of "Fort Canning" referring to a far future concert event that has no relation to the present, and did not contribute to any anomalies in the crowd flows in the corresponding nearby regions. Hence, solely basing on tweet counts to measure current interests level in a region without considering the time dimension might not be the most accurate. Tenses information is derived from Parts-of-speech (PoS) tags of the root verb. We take the tags <VBD> and <VBN> for past tense, <VBG>, <VBZ>, and <VBP> for present tense, and <MD> for future tense, and take their counts. We use the Stanford PoS tagger (Toutanova and Manning, 2000) to obtain the tags. This feature is denoted as $\mathbf{T}_T \in \mathbb{R}^{3 \times I \times J}$.

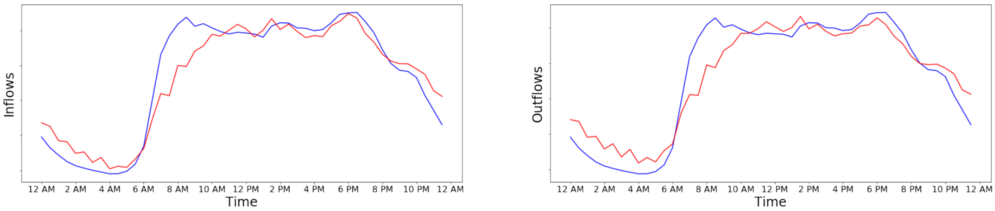


FIGURE 3.7: Fort Canning case study: Crowd flows in nearby regions on 29 June 2013. Crowd flows in that region on that day is comparable to the daily average. Note: Blue lines represent daily mean flows, while the red lines represent specific flows on that day.

Time	**Tweet text**
16:30	@Ai_Arakawa FALL OUT BOY IS COMING TO SG. AUG 6 @FORT CANNING . TICKETS AT SISTIC. HOSTED BY @LiveEmpire #FOBinSG
16:31	@qatarairways FALL OUT BOY IS COMING TO SG. AUG 6 @FORT CANNING . TICKETS AT SISTIC. HOSTED BY @LiveEmpire #FOBinSG
...	... x 121

TABLE 3.4: Fort Canning case study: sample of the irrelevant tweets mentions of a far future event on 6 August 2013

Tweet Sentiment

So far, we assumed that a high spike of interest at a particular location induces large crowd flows around it. However, this assumption may not hold especially when the interests are negative, and instead may suggest the opposite by reducing crowd flows in the region. Examples of such events include a last-minute cancellation of an event or outbreak of a disastrous event. For example, an undesirable major flooding incident happened in Paya Lebar, and was discussed heavily on Twitter. However, this increased spike of tweet counts did not induce additional crowd flows into the region but instead did the opposite (see Figure 3.8 and Table 3.5). This is intuitive as people are less inclined to travel in regions that are negatively portrayed. Inversely, we also expect crowd flows in positively interpreted regions to surge. Thus, we also explore adding sentiment as an extra source of information to help measure the degree of such scenarios. Tweet sentiment information is extracted using a simple counting of positive and negative words based on a manually annotated sentiment lexicon by Hu and Liu (2004). This feature is denoted as $\mathbb{T}_s \in \mathbb{R}^{2\times I\times J}$.

For each of the sub-features, a sequence of matrices that corresponds to the tweets information time window is prepared, and aggregated via simple summation to obtain a single matrix. Finally, they are concatenated with the input sequence in the hourly component to allow the model to also explore dependencies between tweets and the crowd flow matrices. e choose to merge the tweet features with the hourly component's inputs because our underlying intention is to use tweets to model short-term effects to crowd flows. These three features are specially designed to be simple, insensitive to noise, and highly efficient to extract from tweets, as opposed to using more complex methods such as neural networks to represent tweet information, which is important for real-time prediction to be effective.

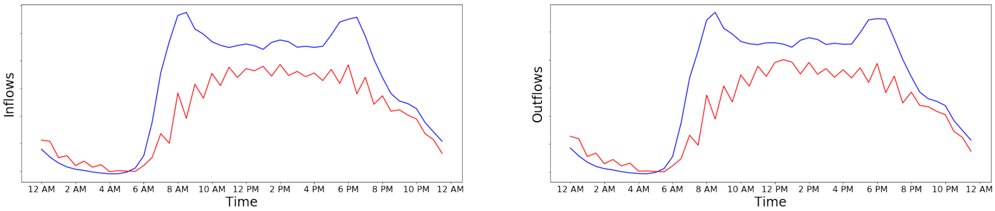


FIGURE 3.8: Paya Lebar case study: Crowd flows in nearby regions during a negatively-potrayed event on 28 April 2013 around 11am. Crowd flows are lower than expected. Note: Blue lines represent daily mean flows, while the red lines represent specific flows on that day.

Time	Tweet text
10:58	Wah f*ck the road near paya lebar mrt flood like shit its fucking knee deep in water !!!
10:59	wow rain till gt flood at paya
10:59	Oh my gosh... Paya Lebar is flooded. Literally.. Like shin-level.
11:09	Water level falls below 90%. High Flood Risk.11:09:13 #SgFlood,,,#SgFlood

TABLE 3.5: Paya Lebar case study: sample of the relevant tweets that reveal that a disaster is ongoing at a particular location.

3.3.4 Data Fusion

Finally, the outputs from the components, $\mathbf{X}_H^{(L+2)}$, $\mathbf{X}_D^{(L+2)}$, $\mathbf{X}_W^{(L+2)}$, and X_{Ext} are fused together to produce a prediction tensor $\hat{\mathbf{X}}_t$. The fusion is performed in two steps as follows:

Parametric-matrix-based Fusion

The first step fused the first three historical observations components via a parametric-matrix-based method to form $\mathbf{X}_F$ with the following mapping:

$$\mathbf{X}_F = \mathbf{W}_W \circ \mathbf{X}_W^{(L+2)} + \mathbf{W}_D \circ \mathbf{X}_D^{(L+2)} + \mathbf{W}_H \circ \mathbf{X}_H^{(L+2)} \tag{3.7}$$

where $\circ$ denotes the element-wise multiplication operator; $\mathbf{W}_W$, $\mathbf{W}_D$ and $\mathbf{W}_H$ are learnable parameters that fine-tune the level of effect from the weekly, daily and hourly component on each grid cell, allowing the model to specify the level of effect from each past observation component on every region locally.

Fusion with External Component

The second step simply adds up the output from the first step with the external component output, and applies a hyperbolic tangent function to the sum to transform the output values to be in the range $[-1, 1]$. The function for this step is as follows:

$$\hat{\mathbf{X}}_t = \tanh(\mathbf{X}_F + \mathbf{X}_{Ext}) \tag{3.8}$$

The model is then trained to minimize the Mean Squared Error (MSE) between the predicted flow matrix and the true flow matrix. The MSE loss function is defined as:

$$\mathcal{L}(\theta) = \sum_{t}^{T} \|\mathbf{X}_t - \hat{\mathbf{X}}_t\|_2^2 \tag{3.9}$$

where θ are all the learnable parameters in the model. Note that since not all grid cells contain crowd flows due to non-traversable regions, we modify the loss function by adding a mask to only calculate loss for specific regions where crowd flows exist.

3.4 Experiments

3.4.1 Settings

Baselines

Apart from *ST-ResNet* as our main baseline, we also compare the results with two other simple baselines – historical average and persistence model. The historical average model predicts the next inflow and outflow values by using the average of the past observations in the same grid cell, and corresponding time interval in the week. The persistence model simply takes the most recent observation of the crowd flows as the next time interval prediction.

3.4.2 Preprocessing

We use the Min-Max normalization to scale the crowd flows values and the extracted tweets features values into the range of $[-1, 1]$, and $[0, 1]$ for the wind speeds and temperature.

Hyperparameters

We use Keras with Tensorflow as the backend to implement our models. The training is done via back-propagation (Adam), with a fixed learning rate of 0.0002. The other hyperparameters in the model are set as follows: $L = 2$, $l_H = 4$, $l_D = 1$, and $l_W = 1$. Batch size used is 32. The convolutions use 64 filters with kernel size of 3×3, while *Conv2* uses 2 filters with the same kernel size. The last four weeks (i.e. 28 days) in each dataset is selected to be the test set, while the rest is the train set. From the train set, 10% validation is used as the development test set, and the remainder 90% is used to train the model in the development phase. Until an early-stop is reached or up to 500 epochs, the development phase ends, and the training continues on with the full train set evaluated with the original test set for 100 epochs.

3.4.3 Results and Discussion

The results of our extended models (i.e. with tweets information) with the tweet information time window fixed at $(2^-, 0^+)$, and the baseline models (i.e. without tweets information) are shown in Table 3.6. The error values reported in Table 3.6 are the normalized Root Mean Square Error (RMSE) in percentages.

TABLE 3.6: Comparison of the results amongst the baselines and extended models. Note: underlined: did not beat baseline; bold: best score for dataset; †: statistical insignificant.

Model	**Dataset**				**Average**
	VLD1	**VLD2**	**VLD3**	**GSM**	
ST-ResNet *(Main baseline)*	3.1278	3.4302	3.4586	2.2520	3.0672
Historical average	5.2428	5.6838	5.0124	2.3585	4.5744
Persistence model	4.3876	4.2329	4.9451	4.9921	4.6391
ST-ResNet					
+ Counts	**3.1073**	3.2965	3.2345	2.2369	**2.9688**
+ Tenses	3.1113	3.4238	3.2665	2.2581†	3.0149
+ Counts + Tenses	3.1459	3.3231	**3.2294**	2.2271	2.9814
+ Sentiment	3.1300	3.4255†	3.4609†	2.2441	3.0651
+ Counts + Sentiment	3.1984	3.2578	3.2498	2.3321†	3.0095
+ Counts + Tenses + Sentiment	3.1578	**3.2455**	3.3409	**2.2072**	2.9879

For every experiment in Table 3.6, except for the main baseline, we also conduct a paired t-test using the error reduction values from the main baseline and each contender model, denoted by b_i and c_i respectively, to test for statistical difference between each matched prediction point i. Assuming that the difference in the error reduction values are normally distributed, the null hypothesis is $H_0 : b_i = c_i$ and alternative hypothesis is $H_1 : b_i > c_i$. Those results marked with † are found to be statistically insignificant which support that claim that it is no better than the main baseline.

Comparisons between Baselines

We observe that all of the models outperform the simple baselines, historical average and persistence model, by a notable margin, signifying the effectiveness of *ST-ResNet*. With error values just a little above 3%, it indicates that our main baseline is highly competitive. It also implies that crowd flows in Singapore are easy to predict.

Effect of Tweet Counts

Extended models with tweet counts are able to reduce the errors by 3.28% on average. The error reduction is small yet statistically significant margin, and this observation is consistent for all datasets. This shows that tweets are indeed highly relevant to crowd flow, and crucial to improve prediction performance.

Through our experiments, we also investigate whether information from tweets are indeed unique from historical traffic patterns, so as to assess the value of tweet counts to the crowd flow prediction problem. We addressed the following questions: do people tweet only during peak hours when communing to and from work, and does the tweet counts duplicate historical traffic pattern? If so, this might qualify any derived information from tweets as any patterns found in tweets would only be a replication of those found in traffic flows. Our findings from Figure 3.10 show a strong pattern that suggests that the total tweet counts diminish around the sleeping hours (2am-5am), slowly increase along the day (from 5am onwards), and peaks during the night (11pm), and it contains minimal correlation with the peak hour traffic patterns. Thus, we conclude that the tweet counts contain some useful information that does not overlap with the historical crowd flows components.

Effect of Tweet Tenses and Sentiment

We observe that models with tweet tenses and/or sentiment have shown conditional effectiveness across different datasets. Our experiments reveal a combination of datasets which fetched positive outcomes and others that performed below the main baseline (underlined in Table 3.6). Upon investigation, we attribute the cause to some degree of feature misrepresentation from their original intentions, resulting in the treatment of these features as noise. For instance, as shown in Figure 3.9 and Table 3.7, a high negative sentiment count is detected within the tweets when a public train broke down during an evening rush hour. We originally expected lower crowd flows near the affected region, as explained in Section 3.3.3, however, the crowd flows increased instead. This can be probably explained by a sudden surge of demand in private cars, taxi and busses in the region which may have contributed to the increased crowd flows. We also provide a few examples of tweets that are labelled as either past or future tense in Table 3.8. Similarly, as per earlier discussed, we deemed these tweets as irrelevant since they are not in present tense. However, these tweets should be considered relevant as they refer to some near-recent or near-future activities which can be useful for prediction.

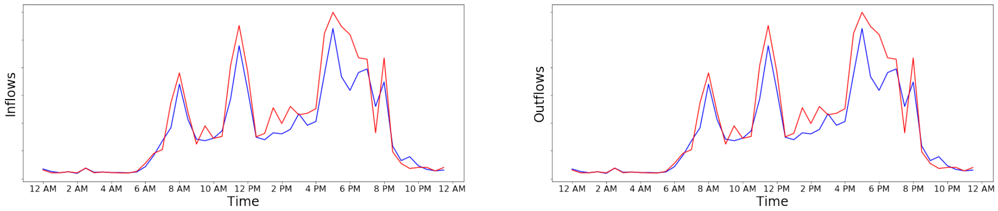


FIGURE 3.9: Serangoon case study: Crowd flows during a time with strong negative sentiment event on 19 June 2013 at 6pm. Crowd flows are higher than expected. Note: Blue lines represent daily mean flows, while the red lines represent specific flows on the day.

Time	Tweet text
18:23	walao mrt breakdown at serangoon omg zzzz
18:29	F*cking train stalling at serangoon
18:41	Train broke down at ne line. Now waiting for bus at serangoon . But bus stop overcrowded.
18:50	Human traffic at serangoon is insane
18:57	You gotta be f*cking kidding me. The entire serangoon mrt breakdown

TABLE 3.7: Serangoon case study: sample of the relevant tweets with strong negative sentiment that reveal that a train breakdown at a particular station.

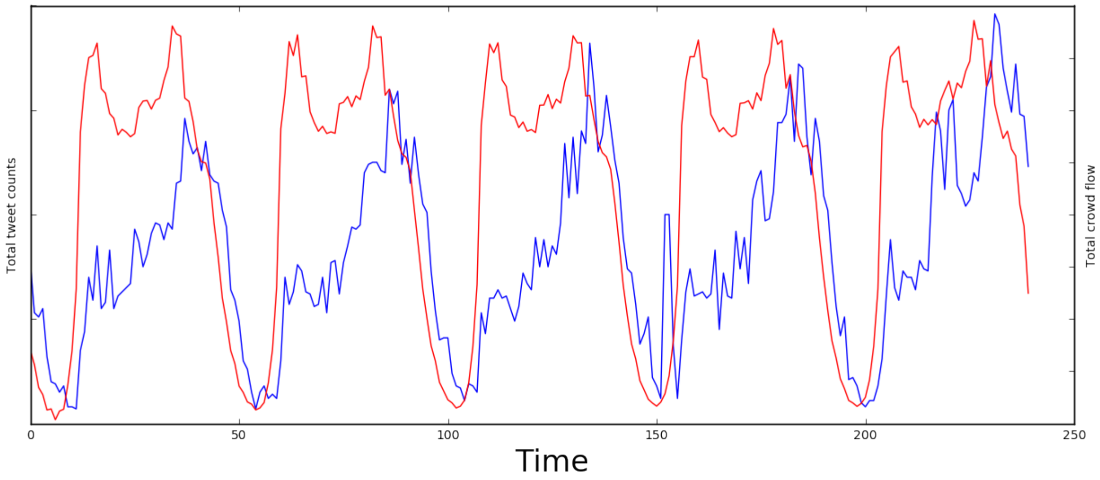


FIGURE 3.10: Total Tweet counts vs. Total crowd flows from 1 March 2013 to 7 March 2013. Blue line denotes total Tweet counts. Red line denotes total crowd flows.

Tense	**Tweet text**
Future	Weekend drive in the morning is the bestttt. I can go from hougang to jurong in 10 mins hahaha
	So there's this dude in paya lebar square dancing while holding a cigarette.
	Dinner with Dad @Simpang Bedok . I can say this is the one of the best Mee Goreng I've tasted in my ...
Past	Just landed safely back at Changi. I told the cab driver I want to go to Bedok . He told me ""sorry I don't take children"
	Just posted a photo @Fort Canning Hill Park
	Finally watched it and it was a great movie. #guardiansofthegalaxy @Golden Village @Yishun

TABLE 3.8: Tenses feature misrepresentation; traffic-relevant tweets yet it is in future or past tense.

Lag$^-$	**Lead**$^+$			
	0	1	2	3
0	3.3389	3.2240	3.2765	3.2038
1	3.2474	3.2168	3.2045	3.2660
2	3.2128	3.2594	3.2547	3.2712
3	3.2092	**3.1993**	3.2267	3.2608

TABLE 3.9: Sensitivity analysis of various Tweet information time windows using the *ST-ResNet* + Tweet Counts Model. Note: values are in RMSE.

Effect of Tweet Time Interval Window

We vary the tweet information time window $(lag^-, lead^+)$ and report the results in Table 3.9, where it lists the average RMSE amongst all VLD datasets for the corresponding time interval window. The optimal time interval window is empirically determined to be $(3, 1)$. Intuitively, the larger the window size, the better the results of the prediction should be, since more tweets are used. However, this pattern is only observed with higher lag^- which tends to achieve better result, as compared to when higher $lead^+$ is used. This suggests that one limitation in using tweets in our setting, which is that tweets do not contain information about future events if these events are unforeseeable in the present, as such types of events can only induce people to tweet about it when it happens then.

Reliability of Twitter as Data Source

It is well known that tweets contain large amount of noise as evidenced in examples shown in above case studies. This accentuates the challenge in mining useful information from tweets, and also risk noise being introduced to the predictive model. Furthermore, the tweets that we collected only cover certain grids in the map as we acknowledge the sparsity of their spatial coverage, as shown in Figure 3.4. Even though the tweets are collected based on locations that are deemed to be representative of well-known clustered regions, this process also relies on local experts to handpick these locations which could be non-trivial for bigger cities. One premise of this approach also rely on the general usage of twitter adopted by the general population. The accessibility of mobile technologies and to social media might not be as widespread as a highly connected city like Singapore.

3.5 Related Work

Similar to our motivations, He, Shen, et al. (2013) also used twitter to improve traffic prediction linear regression model. In their approach, an optimization process is designed to transform the semantics of the tweets into a salient traffic-indicator matrix. We believe that similar approaches to encode the semantics of tweets into the prediction model can be perform in our context and we leave it for future work. Liao, Zhang, et al. (2018) also employ trending online crowd queries as additional input to improve traffic speed prediction accuracy. In the literature, there have been several works that utilize social media as a source of data to study traffic mobility patterns, which are well summarized by Lv, Chen, et al. (2017), Xu, Li, and Wen (2018), and Ni et al. (2014).

Some works propose to use tweets, instead of traditional physical sensors such as VLD and road cameras, to monitor real-time traffic. Carvalho et al. (2012) argue that the latter methods are expensive, require high maintenance, provide poor spatial coverage and usually inaccessible by the public, all of the disadvantages which the former is able to overcome. Wang, Al-Rubaie, et al. (2014) further capitalize on geo-tagged tweets to determine congestion along a specific highway in London. However, it is also well-known that only a small proportion of tweets contain geo-tags and thus they might not fully represent the level of social activity at their respective regions. Semwal et al. (2015) and Shekhar et al. (2016) conducted correlations of traffic-related tweets with traffic congestion, instead of traffic flows. Besides, the key challenge as identified in most studies above, is to reliably classify tweets as being traffic relevant, especially due to the large presence of noise in unstructured natural language text. With the added complexity of spatio-temporal granularity, it only hinders efforts to improve the accuracy of the tweet relevance filtering process. Furthermore, existing studies are often conducted in smaller scales, within specific space and time constraints. Our work is closely related, as we also adopt tweets to represent additional traffic information, but do away with the complex irrelevant tweet filtering step and instead use simple features that are more insensitive to noise.

Liao, Zhang, et al. (2018) to encode additional information such as geographical structure of the road network during large-scale social events, road intersections information and trending online crowd queries

There has been a line of previous work that use tweets for predictions but applied in other domains such as movies' box office prediction (Asur and Huberman, 2010), Distributed Denial-of-Service (DDoS) attacks prediction (Wang and Zhang, 2017), and presidential election results prediction (Tumasjan et al., 2010). While these domains do not face spatio-temporal challenges as pointed out by Zhang, Zheng, et al. (2018), the prediction task in their domain is inherently more difficult as patterns are not as predictable due to the presence of more complex interacting human behaviors, and tye lack of reliable data indicators. Nevertheless, they demonstrate that tweets are good source of information to predict crowd behaviors, and suggest that tweets are indeed informative on telling human behaviors in different aspects of social life ranging from entertainment and politics due to its nature to contain rich real-time information. Thus, our findings are consistent with such existing investigations.

3.6 Summary

In this chapter, we explored the effectiveness of using tweets to crowd flow prediction by extending upon an existing state-of-the-art crowd flow prediction model known as *ST-ResNet*, adding various linguistic features from real-time tweets. These features include tweet counts, tweet tenses' counts, and tweet sentiments' counts. Through our empirical experiments with two different datasets used to represent traffic flows in Singapore, we found that tweets are indeed useful to improving the prediction accuracy up to 3.28% on average, and tested to be statistically significant. We also shared several ways how tweets are related to crowd flows, with respect to the tweet features extracted and the choice of time interval window chosen. The development of our framework to use tweets as additional source of information for crowd flow prediction is still very much work in progress. For future work, it could be a good direction to look deeper into the contextual meaning that can be found in the tweets using more advanced natural language processing methods, while also considering efficiency for real-time prediction. Multilingual text processing can also be useful, especially in cities like Singapore, where non-English languages such as Malay, Chinese and Singlish are widely used, resulting in several misclassifications during feature extraction.

Chapter 4

Understanding Integrated Gradients with Smooth Taylor

4.1 Introduction

Deep neural networks have displayed remarkable success in various large-scale, real-world and complex artificial intelligence tasks in computer vision (Huang, Liu, et al., 2017; Ren et al., 2017; Tan and Le, 2019) and natural language processing (Edunov et al., 2018; Johnson et al., 2017). However, these high performing non-linear neural models, unlike traditional machine learning models, act like a *black box* which suffers from poor input-to-output inference and interpretability. Due to the nature of how deep neural network algorithms are designed, it is difficult to explain *what* or why an *individual* input result in the model arriving at a particular output (Fan et al., 2020). This major disadvantage hinders human experts to fully understand the basis and the reasoning of every prediction a deep neural model makes for each input, limiting the extent of its application in practice.

With the aim to better understand the complex input-to-output behavior of a deep neural network, a number of previous work (Baehrens et al., 2010; Simonyan et al., 2014; Zeiler and Fergus, 2014; Springenberg et al., 2015; Zhou et al., 2016; Zintgraf et al., 2016; Binder et al., 2016; Shrikumar et al., 2017; Sundararajan et al., 2017; Montavon et al., 2017; Selvaraju et al., 2017; Samek, Montavon, et al., 2019) focus on the problem of attribution. Attributions measure the contribution of the model's output explained in terms of its input variables. For instance, for image classification systems, an attribution method assigns a relevance score to every pixel of the input image that explains for the model's predicted class. There are many applications where such an ability to "explain" for a complex model's decision is crucial. Attributions act as supporting evidence to explain the rationale of a model's decision. This helps to facilitate the building of trust between humans and automated systems (Gilpin et al., 2019), and encourage higher adoption of deep neural networks in practice, especially in high-risk application areas. The importance of attribution is more apparent in view of the recent vulnerability discoveries in deep neural networks against malicious and yet unnoticeable to-the-human-eye adversarial attacks (Nguyen and Date, 2015; Moosavi-Dezfooli et al., 2017).

Sundararajan et al. (2017) proposed *Integrated Gradients* (*IG*) as an attribution method for deep neural networks, which unlike other methods (Zeiler and Fergus, 2014; Springenberg et al., 2015; Binder et al., 2016; Zhou et al., 2016; Shrikumar et al., 2017; Montavon et al., 2017; Selvaraju et al., 2017), is fully independent of the composition of the model's structure, and can be easily implemented with access to just the input's gradients after back-propagation. As such, it is computationally efficient to compute, and can be widely applied to various deep neural networks architectures and tasks.

However, *IG* require a selected baseline as a benchmark, which raises the question on how such a baseline is to be chosen. In addition, just as with other gradient-based methods (Baehrens et al., 2010; Simonyan et al., 2014), *IG* often create attribution maps that are noisy which affects the ease of its interpretability. For example, compare the saliency maps (attribution maps visualized by a 2D image) of *IG* (center two) with other methods (Simonyan et al., 2014; Binder et al., 2016; Smilkov et al., 2017) in Figure 4.1, which is based on a DenseNet (Huang, Liu, et al., 2017) with 121 layers pretrained for the ImageNet image classification task. The noisiness of its explanations is visually striking.

Those noise pixels seemingly scattered at random across the maps as shown in Figure 4.1 may indeed reflect the true behavior of the gradients of the deep neural model: as the networks get deeper, the gradients across the input space fluctuate more sharply, resembling white noise, which is described as the shattering gradient problem (Balduzzi et al., 2017). To tackle the noisiness issue, Smilkov et al. (2017) proposed the *SmoothGrad* technique, which uses a random sampling strategy around the input with averaging of the obtained attributions to produce visually sharper attribution maps.

In this chapter, our contributions are as follows:

- We present *SmoothTaylor* as a theoretical concept bridge between *IG* and *SmoothGrad*. Unlike *IG*, it does not require a selected fixed baseline. Under additional assumptions, *SmoothTaylor* is an instance of *SmoothGrad*. Regarding novelty, *SmoothTaylor* is derived from the Taylor's theorem. Experimental results show that *SmoothTaylor* is able to produce higher quality attribution maps that are more sensitive and less noisy as compared to *IG*.

- From the perspective of gradient shattering, we explain why *SmoothGrad* and *SmoothTaylor* deteriorate with too small amount of added noise.

- We emphasize smoothness as a second quality measure for attribution and introduce multi-scaled average total variation as a new evaluation measure for smoothness of the attribution maps.

- We further propose adaptive noising for individual input samples to optimize for either predictor sensitivity of the generated attribution map or the noisiness of it. We show that it results in large improvements in performance compared to constant noise levels.

- This chapter aims at a better understanding of existing gradient-based attribution methods.

The rest of the chapter is organized as follows. Section 4.2 briefly describes *IG* and *SmoothGrad*. In Section 4.3, we derive *SmoothTaylor* as a theoretical bridging concept. Next, in Section 4.4, we conduct experiments by applying the attribution methods on a large-scale image classification problem to generate attribution maps. These attribution maps are quantitatively evaluated and compared. Adaptive noising is discussed in Section 4.5. Lastly, we conclude this chapter in Section 4.6.

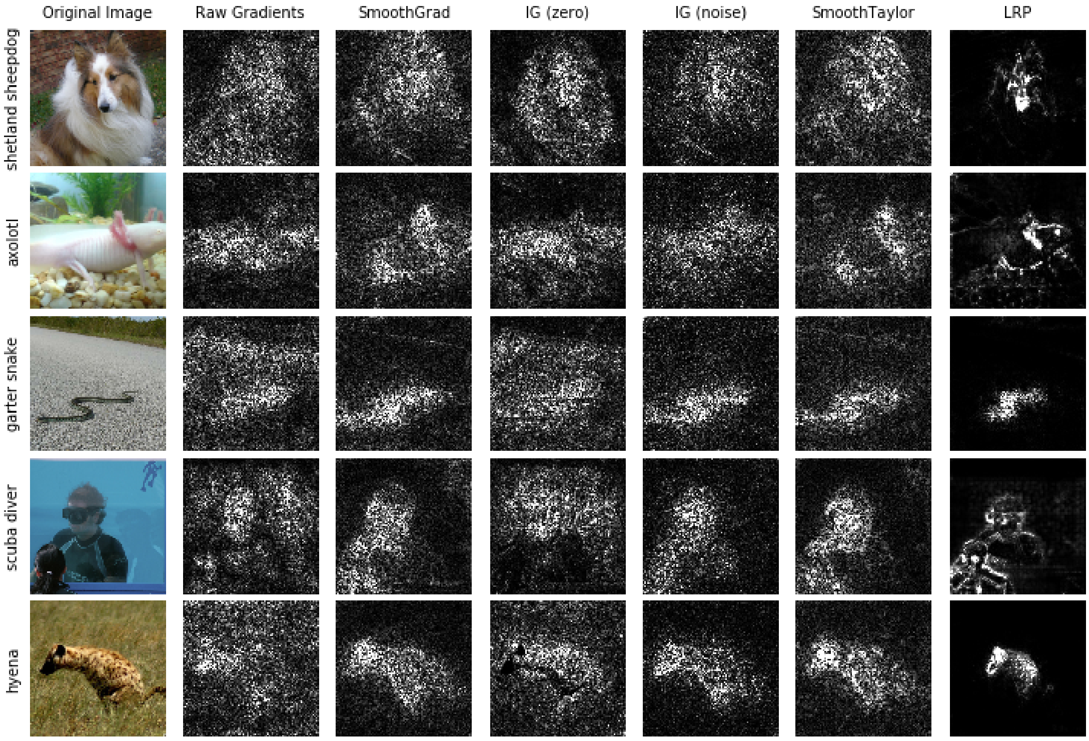


FIGURE 4.1: Comparison of saliency maps computed by different attribution methods. These saliency maps show the relative contributions of each input pixel that explains for the model's prediction. Columns from the left: original input image; raw gradients; *SmoothGrad*; *IG* with zero as the baseline ($M = 50$); *IG* with noise as the baseline ($N = 1$); *SmoothTaylor* (σ =5e-1, $R = 150$); *Layer-wise Relevance Propagation*. Setup: DenseNet121 image classifier pretrained for ImageNet. Normalized absolute values are used to visualize the attribution maps and values above 99th percentile are clipped.

4.2 Preliminaries

4.2.1 Integrated Gradients

Suppose one aims to explain the prediction of a deep neural network represented by a function f for input x. The integrated gradient (Sundararajan et al., 2017) for the i^{th} dimension of the input is defined as follows:

$$IG_i(x, z) := (x_i - z_i) \times \int_{\alpha=0}^{1} \frac{\partial f(z + \alpha \times (x - z))}{\partial x_i} d\alpha \tag{4.1}$$

The gradient of f in the i^{th} dimension is denoted by $\frac{\partial f(x)}{\partial x_i}$, and z is a selected input baseline. In practice, the path integral is usually approximated by a summation across discrete small intervals m with M steps along the straightline path from input x to baseline z, as follows:

$$IG_i(x,z) \approx (x_i - z_i) \times \frac{1}{M} \sum_{m=1}^{M} \frac{\partial f(z + \frac{m}{M} \times (x - z))}{\partial x_i} \tag{4.2}$$

Note that the attributions of the *IG* method satisfy some desirable properties. First, it satisfies *implementation invariance* since the computations are only based on the gradients of f, and are fully independent on any aspects of the models. It also fulfils the *completeness* axiom, which ensures that the attributions add up to the output difference between input x and baseline z (i.e. $\sum_i IG_i(x,z) = f(x) - f(z)$).

Thus, it is recommended to choose baseline z to be zero (with a near-zero score, i.e. $f(z) \approx 0$) to represent the absence of input features. This acts as a basis for comparison and thus allows for the interpretation of the attributions to be a function of solely the individual input features. For images, this is a fully black image, which is argued to be a natural and intuitive choice. However, a black image is usually a statistical outlier to most pretrained models, which makes explanations relative to implausible outlier points seem irrelevant. Another disadvantage of using zero as the baseline is that input features that are zero or near-zero will never appear on the attribution maps since multiplier $x_i - z_i$ will be almost close to zero. For example in Figure 4.1, saliency maps of *IG* with zero as the baseline mostly fail to highlight objects of interests represented by dark-colored pixels.

An alternative baseline with the same near-zero score property is also proposed – uniform random noise. To address the issue of which random noise baseline to be chosen, a valid approach is to draw different noise baselines $z^{(n)}$ to compute N *IG* mappings, and average over them[1]:

$$\overline{IG}_{noise}(x) = \frac{1}{N} \sum_{n=1}^{N} IG(x, z^{(n)}) \tag{4.3}$$

This slight extension does seem to improve *IG* and result in more sensitive attribution maps with less noise, though there is still much room for improvement. Moreover, it should be noted that uniform random noise is also an unseen outlier, thus it guides to generate explanations that are no more meaningful than the zero baseline. Perhaps, the need for this method to fix a baseline that is consistent enough for all inputs, and at the same time does not deviate too far from the points in the dataset, is a fundamental flaw in its design, as such a baseline may not exist.

4.2.2 SmoothGrad

While the original *SmoothGrad* technique (Smilkov et al., 2017) smooths the raw gradients over the input space, it can be viewed as a general procedure which computes an

[1] https://github.com/ankurtaly/Integrated-Gradients/

attribution map by averaging over multiple attribution maps of an arbitrary gradient-based attribution method (denoted as $\mathcal{M}$) with multiple N' noised inputs:

$$SmoothGrad(x) = \frac{1}{N'} \sum_{n=1}^{N'} \mathcal{M}(x + \epsilon),\ \epsilon \sim \mathcal{N}(0, \sigma'^2) \tag{4.4}$$

Gaussian noise with parameter σ' is used to smoothen the input space of the attribution method and construct visually sharper attribution maps. It is briefly discussed in their paper that σ' needs to be carefully selected to get the best result. If too small, the attribution maps are still noisy; if too large, the maps become irrelevant.

4.3 SmoothTaylor

In this section, we explain the derivation of *SmoothTaylor*. Firstly, we discuss the motivation of our proposed improvement from the Taylor's theorem approximation perspective. Any arbitrary differentiable function f can be approximated by Taylor's theorem with the first order term while ignoring all other higher order terms:

$$f(x) \approx f(z) + \sum_i (x_i - z_i) \frac{\partial f(z)}{\partial x_i} \tag{4.5}$$

This yields an explanation, which describes how the output of the model $f(\cdot)$ in point x is different from the output of the same model in point z. Notably, it is an explanation for x relative to z. This raises the valid issue on how the point z should be chosen.

Secondly, in statistics, a valid method to deal with uncertainty is to compute an average over an uncertain quantity. In the case of uncertainty about which point z should be chosen, the proper approach is to draw several roots $z^{(r)}$ (according to some method which we defer the discussion till later) and average over them, so as to improve the power of the approximation:

$$f(x) \approx \frac{1}{R} \sum_{r=1}^{R} \left[f(z^{(r)}) + \sum_i (x_i - z_i^{(r)}) \frac{\partial f(z^{(r)}))}{\partial x_i} \right] \tag{4.6}$$

Equation (4.6), in turn, is a discrete approximation for the integral (with S which has to be a measurable set):

$$f(x) \approx \int_{z \in S} f(z) + \sum_i (x_i - z_i) \frac{\partial f(z)}{\partial x_i} dz \tag{4.7}$$

We are now ready to outline our method. Based on the concepts described above, the smooth integrated gradient in the i^{th} dimension of an input x within a set of roots $z \in S$ is defined as follows:

$$SmoothTaylor_i(x) := \int_{z \in S} (x_i - z_i) \frac{\partial f(z)}{\partial x_i} dz \tag{4.8}$$

Equation (4.8) has two salient differences to *IG* from Equation (4.1). First, the explanation point z_i in the inner product $(x_i - z_i)$ is part of the integral, whereas in *IG*, it is outside of it. Second, the integration set S is not a path from x to some point z as it was in *IG*.

Similarly, for the reason of efficient computation, the integral can also be approximated using a discrete summation over R multiple roots $z^{(r)}$:

$$SmoothTaylor_i(x) \approx \frac{1}{R}\sum_{r=1}^{R}(x_i - z_i^{(r)})\frac{\partial f(z^{(r)})}{\partial x_i}, z^{(r)} \sim S \tag{4.9}$$

Equation (4.9) is derived from the averaged Taylor's theorem approximation in Equation (4.6) by choosing a set of roots such that the model output score difference between each root $z^{(r)} \in S$ and input x is almost close to zero (i.e. $\forall r : f(x) - f(z^{(r)}) \approx 0$). As a result, the inner summation term $f(z^{(r)})$ is canceled out with $f(x)$, and the remaining terms can be explained as the sum of the smooth integrated gradients across all dimensions. Note that this loosely satisfies the *completeness* axiom just like the *IG* method. It also fulfils the *implementation invariance* property.

The next issue is to decide on a suitable method to generate the roots $z^{(r)}$. If one is interested in classification or segmentation as pixel-wise classification, then one would want to choose the set S to be a set of points where the prediction output class switches. However searching these points on the training dataset might result in roots which are too far away from the input x to be explained, which will impact the quality of the Taylor approximation. One alternative is to seek for a random set of points sufficiently close to x, so that the quality of the Taylor approximation is acceptable, and also sufficiently far away, so that the noise from the gradient shattering effect in deep networks (Balduzzi et al., 2017) can be canceled out by averaging over many z from many different linearity regions. A simple approach, inspired by *SmoothGrad*, is to add a random variable ϵ to input x, where ϵ can be drawn from a Gaussian distribution with standard deviation σ being the noise scaling factor:

$$z^{(r)} = x + \epsilon, \text{where } \epsilon \sim \mathcal{N}(0, \sigma^2) \tag{4.10}$$

The choice of the σ value should be carefully selected, and it is further discussed in Section 4.5. This follows the principle of choosing $z^{(r)}$ to be close to x and also sufficiently far away, so that the need for a good Taylor approximation and averaging effect of the noise in the gradients can be balanced.

Theorem: If the roots in *SmoothTaylor* are chosen as per Equation (4.10), then the discrete version of *SmoothTaylor* as given in Equation (4.9) is a special case of *SmoothGrad* with $\mathcal{M} = \nabla f(x + \epsilon) \cdot \epsilon$.

This theorem does not hold for other choices of the set S in Equation (4.9), thus *SmoothTaylor* defines an algorithm class of its own.

SmoothTaylor offers an alternative formulation to *IG*, where the selection of a fixed baseline is not required. The above theorem establishes *SmoothTaylor* with a choice of roots as in Equation (4.10) as a theoretical bridging concept between *IG* and *SmoothGrad*.

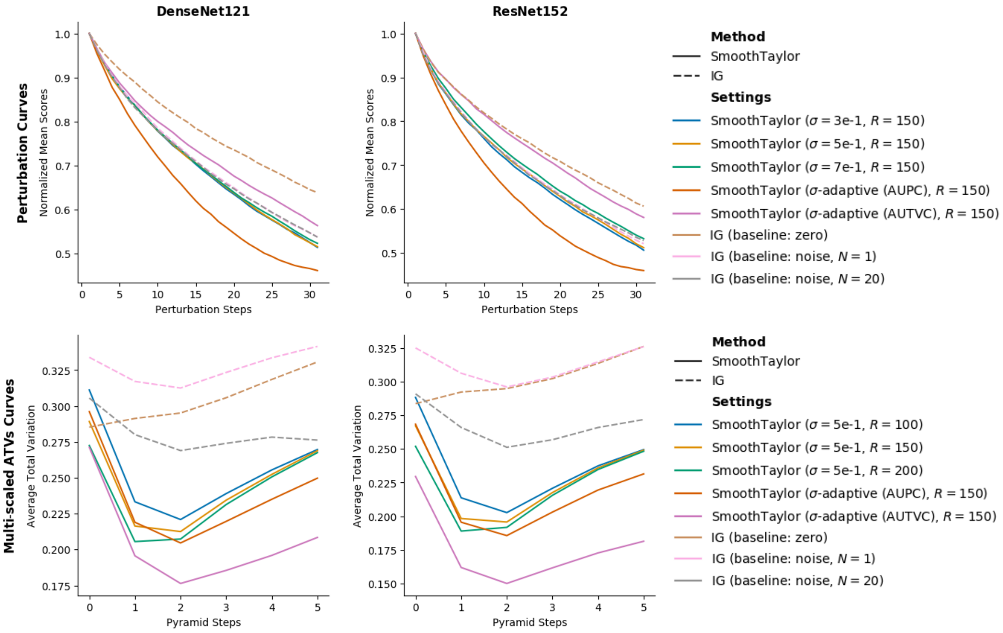


FIGURE 4.2: Evaluation metrics curves; the lower the curve the better. Right: Legends. Top row: Perturbations curves. Bottom row: Multi-scaled TV curves. Left column: Based on DenseNet121. Right column: Based on ResNet152.

4.4 Experiments

We apply *SmoothTaylor* and *IG* (Sundararajan et al., 2017) attribution techniques, and compare their results. We choose to analyze them on the image classification task. The goal is to compare the quality of the attribution maps computed by these two methods. To encourage reproducibility, we publicly release our source code[2]. Here, we describe our experiment setup and evaluation metrics.

4.4.1 Setup

We use the first 1000 images from the ILSVRC2012 ImageNet object recognition dataset (Russakovsky et al., 2015) validation subset as the scope of our experiment. It is a 1000 multi-class image classification task, with each image preprocessed to be the size of 224 × 224 pixels. We choose two deep neural image classifier models, DenseNet121 (Huang, Liu, et al., 2017) and ResNet152 (He, Zhang, et al., 2015), that are both pre-trained on the ImageNet dataset to apply the attribution methods. We compute the attributions with respect to the function of the predicted class for each input image regardless of the ground truth label. Therefore, the attribution process is entirely unsupervised.

[2] https://github.com/garygsw/smooth-taylor

4.4.2 Hyperparameters

For the *SmoothTaylor* method, we vary the parameter values for the number of roots R to be 100, 150, and 200, and the noise scaling factor σ to be $3\mathrm{e}{-1}$, $5\mathrm{e}{-1}$, and $7\mathrm{e}{-1}$. The magnitudes of the noise scaling factor are decided to be roughly in the range of the average values of the inputs after normalization. For *IG*, we choose total steps M to be 50, and vary the type of baselines used. We use the zero (black image) baseline, and random uniform noise baselines with different samples sizes N to be 1, 5, 10, and 20.

4.4.3 Evaluation Metrics

Sundararajan et al. (2017) argued against empirical methods for evaluating attribution methods, and thus decided to rely on an axiomatic approach to determine the quality of an attribution method. However, axiom sets might be incomplete, and for a data-driven science, a quantitative evaluation is often aligned with the goals. Furthermore, there are limitations to qualitative evaluation of attribution maps due to biases in human intuition towards simplicity whereas deep neural models which might be over-parametrized and thus of high complexity. Therefore, in our experiments, we use the following two quantitative metrics:

Perturbation Approach

One such metric suggested by Samek, Binder, et al. (2017) relies on selecting the top salient regions of pixels in the input image by attribution and successively replacing them with random noise (also known as pixel perturbation), and then measuring the drop in model output scores. A higher score drop signifies a more sensitive attribution method, since the attributions are able to better identify the salient parts of the input that explain the model's output.

We describe our pixel perturbation evaluation procedure formally as follows. First, we use a sliding local window of kernel size $k \times k$ in the input image space to find an ordered sequence $\mathcal{O} = (r_1, r_2, ..., r_L)$ that contains the top-L most salient non-overlapping regions. The sorting of the regions is based on the average absolute attribution values of the pixels' location within each kernel window, from the highest to lowest (most relevant first). A high average absolute attribution value in a region r_l denotes a high presence of evidence that supports the model's prediction.

Second, we follow the sequence of ordered regions in $\mathcal{O}$ to apply the perturbations on. Let $g(x, r)$ be a function which performs the perturbation on some input image x at region r, where information in that region is removed by the replacement of the value of its pixels with random values drawn from a uniform distribution across the valid input value range. The function g is then successively applied starting with the original input image $x^{(0)} = x$. The input image for the next step $x^{(l)}$ is iteratively updated after perturbation at step l for L times:

$$\forall\, 1 \leq l \leq L:\; x^{(l)} = g(x^{(l-1)}, r_l) \tag{4.11}$$

At each step l, we consider P number of different random perturbation samples and compute the mean score $\bar{y}^{(l)}$:

$$\bar{y}^{(l)} = \frac{1}{P}\sum_{p=1}^{P} f(x^{(l-1)^{(p)}}) \tag{4.12}$$

The perturbation with the median output score is selected as the actual perturbation to update. To quantitatively measure the strength of an attribution method, we look at how much these mean output scores drop with steps l. That can be quantified by taking the Area Under the Perturbation Curve (AUPC) (see Figure 4.2 (top)) after normalizing each mean score $\bar{y}^{(l)}$ at each step l with the original score $f(x)$, and averaged over all images in the dataset. Throughout our experiments, we use kernel size $k = 15$, number of perturbations $L = 30$, and perturbation sample size $P = 50$.

Average Total Variation

We use Average Total Variation (ATV) as the second evaluation metric to measure the smoothness or the total amount of noise of each pixel with its local neighbors. We consider a saliency map $\mathcal{S}$ as vector of size $h \times w$ to represent every pixel. Taking only absolute values, a min-max normalization (with values above 99th percentile clipped off) is applied on an attribution map to construct a saliency map. The ATV of $\mathcal{S}$ is computed as follows:

$$ATV(\mathcal{S}) = \frac{1}{h \times w}\sum_{i,j \in \mathcal{N}} \|\mathcal{S}_i - \mathcal{S}_j\|_p \tag{4.13}$$

Here, $\mathcal{N}$ defines the set of pixel neighbourhoods (adjacent horizontal and vertical pixels) and $\|\cdot\|$ is the ℓ_p norm. We use the established ℓ_1-norm in our experiments.

In addition, we construct Gaussian pyramids (Burt and Adelson, 1983) on the saliency maps by repeatedly scaling their dimensions down by 1.5 and applying a Gaussian smoothing filter to remove information. This process is repeated for each saliency map until the size of the map is smaller than 30×30 pixels. We then compute the ATV of the scaled and blurred saliency maps at each step – we call them multi-scaled ATVs. Subsequently, after averaged over all images, we take the Area Under the multi-scaled ATVs curve (AUTVC) (see Figure 4.2 (bottom)) as the measure quantity to evaluate the quality of an attribution method.

4.4.4 Results

We compute the attribution maps using a few different attribution methods based on two pretrained image classifiers on the ImageNet dataset. Examples of these attribution maps are visualized as saliency maps in Figure 4.1.

Qualitatively, we can observe that *SmoothTaylor* produces visually sharper saliency maps as compared to *IG*. In addition, they are better at highlighting distinctive regions that explain the model's prediction. While it is not the best method that produces the

TABLE 4.1: Area under the curves results. Note: Lower AUPC and AUTVC is better.

Attribution Method		**Image Classifier Model**			
		DenseNet121		ResNet152	
IG					
baseline	N	AUPC	AUTVC	AUPC	AUTVC
zero	-	23.63	1.52	22.87	1.51
noise	1	21.51	1.62	21.05	1.54
	5	21.54	1.52	20.99	1.43
	10	21.46	1.45	**21.02**	1.37
	20	**21.43**	**1.39**	**21.02**	**1.32**
SmoothTaylor		DenseNet121		ResNet152	
σ	R	AUPC	AUTVC	AUPC	AUTVC
3e−1	100	21.24	1.28	20.83	1.20
	150	21.19	1.24	20.79	1.16
	200	**21.13**	1.22	**20.78**	1.14
5e−1	100	21.25	1.23	21.00	1.14
	150	21.20	1.19	20.95	1.10
	200	**21.13**	1.16	20.86	1.07
7e−1	100	21.39	1.20	21.37	1.08
	150	21.30	1.15	21.32	1.04
	200	21.30	**1.12**	21.14	**1.01**

TABLE 4.2: Area under the curves results for *SmoothTaylor* with extreme hyperparameter values. Note: Lower AUPC and AUTVC is better.

SmoothTaylor		**Image Classifier Model**			
Hyperparameters		DenseNet121		ResNet152	
σ	R	AUPC	AUTVC	AUPC	AUTVC
5e−1	10	21.74	1.55	21.43	1.43
1e−4	100	23.45	1.79	23.00	1.55
1e−3	100	23.60	1.53	23.14	1.48
1e−2	100	23.90	1.57	23.46	1.23
1e−1	100	22.03	1.43	**21.44**	1.22
1	100	**21.88**	**1.17**	22.16	**1.04**
2	100	23.54	1.19	24.48	1.27

least noise or the most sensitivity (see saliency maps produced by Layer-wise Relevance Propagation (LRP) (Binder et al., 2016)), *SmoothTaylor* offers ease of implementation and fulfils the two current fundamental axioms of an attribution method.

TABLE 4.3: Area under the curves results with Adaptive Noising. Note: Lower AUPC and AUTVC is better.

SmoothTaylor		**Image Classifier Model**			
Hyperparameters		DenseNet121		ResNet152	
σ	R	AUPC	AUTVC	AUPC	AUTVC
Adaptive-AUPC	150	**19.55**	1.14	**19.30**	1.05
Adaptive-AUTVC	150	22.14	**0.99**	22.52	**0.85**

Next, we discuss the results using quantitative evaluation measures. A summary of the experimental results is shown in Table 4.1 with the AUPC and AUTVC values for each experiment run. The Simpson's rule is used to compute the area under the curves. We analyze the results based on two objectives – sensitivity and noise level, and also compare the results based on two different classifier models.

Sensitivity

As observed in Figure 4.2 (top), when compared to *IG*, the attribution maps of *SmoothTaylor* are able to cause a larger classification score drop as perturbation step increases. Expectedly, the AUPC values for *SmoothTaylor* are also lower, showing that *SmoothTaylor* is more sensitive to relevant explanations points in the input space than *IG*. The averaged *IG* with noise baselines are shown to have large improvements; almost close to the performance of *SmoothTaylor* at our chosen hyperparameters, though still a little worse. Their improvements also produce diminishing marginal returns as N increases beyond more than 5. On closer inspection with Table 4.1, it shows that our choice for σ values did not produce any significant effect on the AUPC values, which is worth investigating further in Section 4.4.5. However, the AUPC values clearly decrease as R increases. This is expected as the "smoothing" effect is greater when we draw more roots, resulting in a statistically better representation of z which improves the power of the Taylor approximation.

Noise level

The *SmoothTaylor* method clearly generates attribution maps that are much less noisy than *IG*. As seen in multi-scaled ATV curves in Figure 4.2 (bottom), all the curves for *SmoothTaylor* are lower that the curves for *IG*. We also compare the effect of σ and R on the noisiness of the attribution maps of *SmoothTaylor*. First, the AUTVC values decrease as R increases. This is also expected due to the increase "smoothing" effect. Second, the AUTVC values seem to increase as σ increases. However, we believe that this relationship is not monotonically true, as the selection of our σ values may be too low across all images in the dataset. We discuss this further in Section 4.4.5.

DenseNet121 vs. ResNet152

The sensitivity improvements in the perturbation curves by *SmoothTaylor* over *IG* is noticeably lesser for ResNet152 as compared to DenseNet121. One hypothesis is that

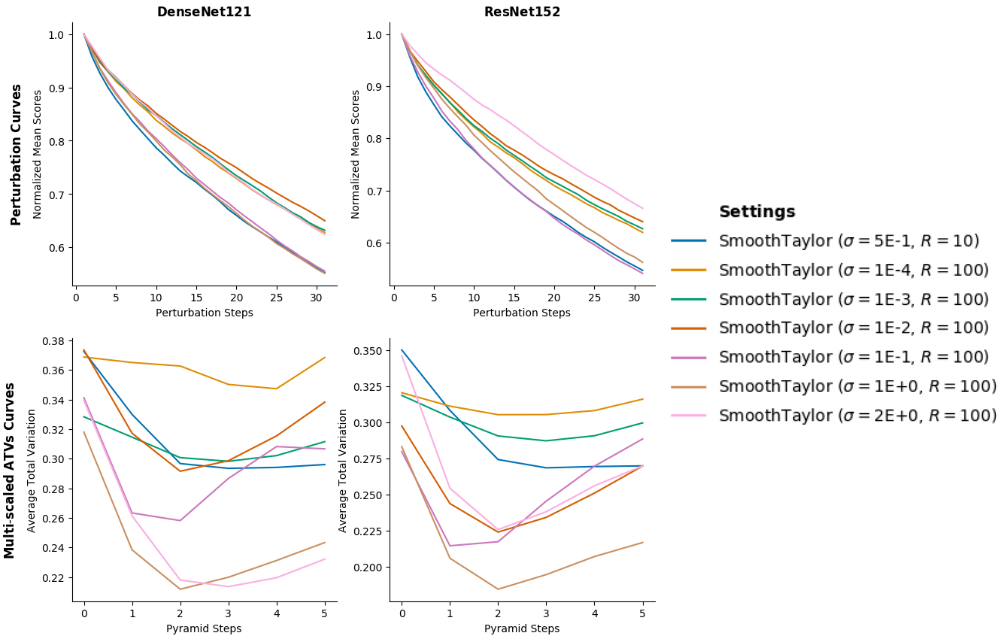


FIGURE 4.3: Evaluation metrics curves for the study of the impact of varying the noise hyperparameter; the lower the curve the better. Top row: Perturbation curves. Bottom row: Multi-scaled TV curves. Left column: Based on DenseNet121. Right column: Based on ResNet152.

the gradients from ResNet152 are less noisy to begin with, since residual networks are shown to have reduced shattering gradients effect. Thus, with more reliable gradients to explain for the model's prediction, the effectiveness of smoothing is also reduced.

4.4.5 Noise Hyperparameter Sensitivity Analysis

We choose a range of σ values as high as 2 and as low as $1\text{e}{-4}$, while fixing R to be 100. The effects of different values of the noise scale parameter for *SmoothTaylor* are displayed in Figure 4.3, and its results are summarized in Table 4.2.

We can observe that for too small noise choices such as $1\text{e}{-4}$ or $1\text{e}{-3}$, the AUPC sensitivity is lower than for choices in the order of $1\text{e}{-1}$. This can be explained from the effect of gradient shattering in deep networks: when the gradient has a large component resembling white noise, as observed by Balduzzi et al. (2017), then using averages is a statistically reasonable attempt to remove the white noise component. Rectified Linear Unit (ReLu) networks consist of zones with locally linear predictions – see Figure 3 in the paper of Novak et al. (2018) for a clear illustration of this effect.

The gradient is constant within each such zone. Above averaging requires to sample the gradient at many different local linearity zones around the sample of interest x. In particular averaging requires z_i to be outside of the linearity zone in which x is in. This explains why a very small amount of noise will not result in an effective averaging of white noise, as most of the samples z_i would just stay in the local linearity zone of x and fail to sample different gradient values.

Algorithm 1: Adaptive Noising

```
Parameters: Max. iterations $i_{max}$, learning rate $\alpha$, learning decay $\gamma$, max. stop
            count $s_{max}$
Input     : Input $x$, root size $R$, model $f$
Output    : Optimal $\sigma^*$ value
begin
    $\sigma \leftarrow \frac{1}{N}\sum|x|$;
    AUC $\leftarrow$ ComputeAUC($x, R, f, \sigma$);
    $i \leftarrow 1$; $s \leftarrow 0$; $\sigma^* \leftarrow \sigma$; AUC$^* \leftarrow$ AUC;
    while $i \leq i_{max}$ do
        AUC$_s \leftarrow$ ComputeAUC($x, R, f, |\sigma + \alpha|$);
        if AUC$_s >$ AUC then
            $\sigma \leftarrow |\sigma - \alpha|$;
            AUC$_s \leftarrow$ ComputeAUC($x, R, f, \sigma$);
        else
            $\sigma \leftarrow |\sigma + \alpha|$;
        end
        if AUC$_s >$ AUC then
            if $s \leq s_{max}$ then
                $\alpha \leftarrow \alpha * \gamma$; $s \leftarrow s + 1$;
            else
                break
            end
        else
            $s \leftarrow 0$;
            if AUC$_s <$ AUC$^*$ then
                AUC$^* \leftarrow$ AUC$_s$; $\sigma^* \leftarrow \sigma$;
            end
        end
        AUC $\leftarrow$ AUC$_s$; $i \leftarrow i + 1$;
    end
end
```

The size of the local linearity zone is sample-dependent (Novak et al., 2018). This observation supports the claim that the noise scale σ needs to be carefully calibrated within a certain range (i.e. it cannot be too small or too big) for every individual sample x in order for the attribution maps of *SmoothTaylor* to be of high quality. Therefore, based on this observation, we go further and propose an adaptive improvement to *SmoothTaylor* in the next section.

4.5 Adaptive Noising

Ideally, the value of noise scale σ should depend on each individual input, and not generally fixed to all inputs. Thus, we propose an adaptive noising technique to search for an optimal noise scale value for each input, so as to optimize the *SmoothTaylor* method.

We adopt an iterative heuristic line search approach to design our algorithm. The goal is to find an optimal value for σ such that the attribution maps can be the most sensitive or least noise (quantified by AUPC or AUTVC respectively). As such, while fixing R, we search for σ^* for each input such that the AUPC or AUTVC of its attribution map is minimized. We describe our algorithm in Algorithm 1.

In our proposed iterative optimization procedure, we search for σ^* within maximum iterations of i_{max}. We include an early stopping mechanism with maximum stop count s_{max}. At each iteration, σ is updated with learning rate α which direction depends on a line search. The learning rate is reduced by a factor learning decay $\gamma < 1$ whenever the current iteration's Area Under Curve (AUC) is greater than the previous one. In our experiment, we use $R = 150$ and set maximum iterations $i_{max} = 20$, maximum stop count $s_{max} = 3$, learning rate $\alpha = 0.1$, learning decay $\gamma = 0.9$, and use the same setup from the AUC computation in our earlier experiments.

We report the results from using adaptive noising in Table 4.3 and compare with the results from previous experiment runs. With adaptive noising, we are able to obtain the best AUPC or AUTVC values among all runs. However, it is to be noted that computing AUPC is computationally expensive and slow while computing AUTVC is much faster. The results conclusively show that *SmoothTaylor* with adaptive noising is preferable over constant noise injection.

4.6 Summary

Explaining for all deep neural model decisions is a huge challenge given the vast taxonomy of model types and scope of problems. Thus it is crucial to find a simple attribution method that is easily applied to various model architectures so as to encourage widespread usage. In this chapter, we bridge *IG* and *SmoothGrad* and proposed *SmoothTaylor* from the Taylor's theorem perspective. In our experiments, we also introduce multi-scaled average total variation as a new measure for noisiness of saliency maps. We further proposed adaptive noising as a hyperparameter tuning technique to optimize our proposed method's performance. From the experimental results, *SmoothTaylor* is able to produce attribution maps that are more relevance-sensitive and with much less noise as compared to *IG* .

Chapter 5

Conclusion

This concluding chapter presents some preliminary work on the application of Explainable Artificial Intelligence (xAI) methods on traffic status prediction in Section 5.1, discuss directions of future work in Section 5.2, and finally provides an overall summary on the thesis in Section 5.3.

5.1 Preliminary Applications

We apply some faster to implement xAI methods on a network-based traffic status prediction task, specifically a model based on Graph Convolutional Network (GCN) known as Diffusion Convolutional Recurrent Neural Network (DCRNN) (Li, Yu, et al., 2018). We choose this traffic prediction tasks and deep learning model as little research work have been done in this area. The dataset chosen for our study is the METRA-LA dataset (Jagadish et al., 2014), which contains time series data of traffic speeds across 207 speeds sensors installed in Los Angeles city. The goal of DCRNN is to predict a sequence of future traffic speeds across all 207 nodes for T periods, given the previous traffic speeds for the last T periods. We use parameter value $T = 12$ in our experiment, with a period interval of 5 minutes.

We apply our proposed *SmoothTaylor* and *Integrated Gradients* (*IG*) on DCRNN and visualize the global feature importance both spatially (via heatmaps) and temporally (via bar graphs) as shown in Table 5.1. From the visualizations, we are able to observe sensor locations that are better at explaining the prediction output. Nodes with darker colors indicate locations with sensors' value that is more sensitive to the prediction output. Visually, *SmoothTaylor* is clearly able to create more contrast in the heatmap as compared to *IG*, reinforcing its superiority.

From the spatial heatmap generated by *SmoothTaylor* in Table 5.1 (top), we deduce a few useful interpretations. First, nodes along the major highway US 101 (northwest to southeast of map) are mostly colored red, highlighting them as critical input nodes for model to make its prediction. This makes sense since the highway leads further south to the East Los Angeles Interchange, the world's busiest freeway interchange. This highway contains huge amount of traffic flows in and out of the California state. Thus, traffic speeds from sensors along this highway are expected to be acutely telling of future traffic speeds due to the sheer volume of vehicles traveling on this road link. Second, on the other hand, nodes along roads surrounding Glendale (eastern side of map), a suburb city of Los Angeles, are mostly colored blue. This suggests that they are of less critical importance in the model's prediction function relative to other nodes. This may be explained by the smaller population of the city, and the more regular

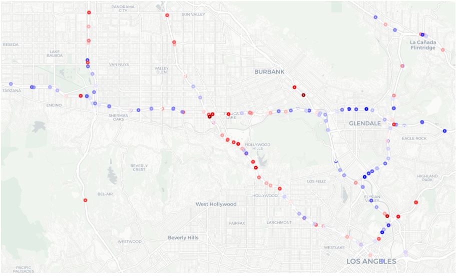
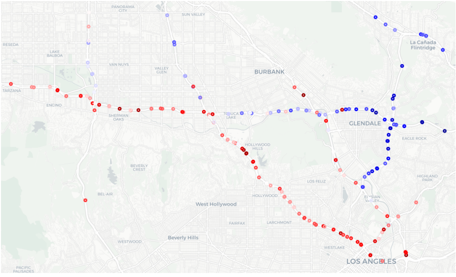
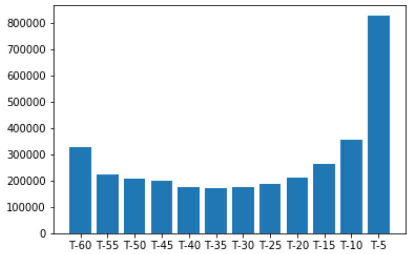
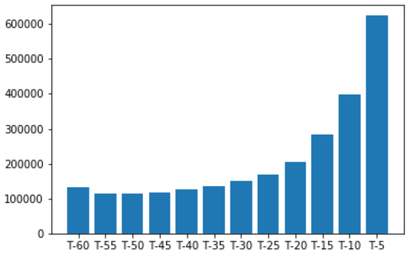
	Methods	
	IG	**SmoothTaylor**
Spatial		
Temporal		

TABLE 5.1: Visualizations of global feature importance generated using *SmoothTaylor* and *IG* for both spatial and temporal features.

traffic patterns by its inhabitants that does not provide much useful discriminatory information to the prediction function. Third, sensor input values of nodes nearer to road intersections are more useful than nodes along straight roads. This also makes sense as congestion often occur around intersections as they are often bottlenecks in road networks.

From the temporal graphs generated in Table 5.1 (bottom), we observe that periods closer to the present (prediction time) are given higher attribution values as compared to periods further away. This effect is more prominent for *SmoothTaylor* as compared to *IG*. This observation suggests that the propagation of traffic speeds into the future is mostly short-term dependent, with historical values from 5 to 15 minutes ago being most important relative to values from older periods.

5.2 Future Works

In this section, we discuss potential future projects that can be derived from this thesis.

From the Twitter-informed crowd flow prediction study in Chapter 3, there is good potential for further improvement if more advanced natural language processing techniques are employed to extract useful information from text to improve short-term traffic prediction. One major challenge is the handling of huge amount of noise present in Tweets which require additional preprocessing steps such as spams filtering, parsing multiple languages, localization filtering etc.

Perhaps, we can also consider extracting additional information from other web platforms in search of relevant information that is useful for traffic prediction. For example, some useful information that can be extracted include events detection, traffic jams reporting, car park availability, road maintenance schedules etc. There is so much more information that can be utilized as additional external inputs to improve model performance or better explain the models' behaviors. However, we must also be wary of fake/false information that are commonly found in the web. Sufficient effort must be done to choose the data sources wisely, or perform some fact checking to ensure the validity/integrity of the extracted information.

From the *SmoothTaylor* study in Chapter 4 and some preliminary work done in Section 5.1, we can suggest numerous ways to explore further on how xAI can be better applied to traffic prediction. Further research on new ways to visualize the outputs of the interpretation tools is useful, as it is currently difficult to visualize the complex spatio-temporal dependencies using ordinary visualization tools. Of course, there are also many opportunities to implement the xAI methods on other types of deep neural networks for various other traffic-related tasks, so as to evaluate its general applicability, and strengths and weakness in various context. Furthermore, a data-driven approach to evaluate the attributions heatmaps, especially when there is a heterogeneous data source is also an unexplored research topic and thus a part of future work.

5.3 Summary

This thesis explores the topic on explaining deep learning models for the traffic prediction, which is not yet commonly explored in the literature. While it is reassuring to see the state-of-the-art performance of traffic prediction continually being bested by newer deep learning models work, we cannot be too overconfident that these models can perform well in real-life applications. Deep learning models are renowned to be highly difficult to interpret. The link between the input and output cannot be observed directly without xAI methods to interpret them.

We provide a brief literature review in Chapter 2 on machine learning methods specific to traffic prediction, various traffic prediction problems, and xAI methods for deep learning in general. Then in Chapter 3, we explore the value of augmenting Twitter as an additional data source in a crowd flow prediction deep neural network model, *ST-ResNet*. From the experiments, we show that Twitter do contribute to the improvement of the prediction accuracy but with some limitations, though much more can be explored to better exploit the rich amount of information from social media in its natural language form. Through the experiments, we also conduct several case studies to show how tweets can be used to also explain for certain irregular crowd flows, highlighting the usefulness of tweets in the context of traffic prediction. Next in Chapter 4, we take a more generic view on attribution methods that are highly applicable and agnostic to the architecture of deep learning models. We focus deeply on a particular attribution method known as *IG*. Drawing inspiration from another method known as *SmoothGrad*, we propose a novel attribution method known as *SmoothTaylor*. We test the methods with experiments on the ubiquitous image classification task, and show that it is able to better explain input-to-output behavior of deep neural networks as compared to the other model-agnostic gradient-based attribution methods. Finally, we conclude in this final chapter by discussing some preliminary work and directions on future work.

Bibliography

Abadi, A., T. Rajabioun, and P. A. Ioannou (2015). “Traffic Flow Prediction for Road Transportation Networks With Limited Traffic Data”. In: *IEEE Transactions on Intelligent Transportation Systems* 16, pp. 663–662.

Adebayo, J., J. Gilmer, I. Goodfellow, and B. Kim (2018). *Local Explanation Methods for Deep Neural Networks Lack Sensitivity to Parameter Values*. arXiv: 1810.03307 [cs.CV].

Akiyama, T. and H. Inokuchi (2014). “Long term Estimation of Traffic Demand on Urban Expressway by Neural Networks”. In: *2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS)*, pp. 185–189. DOI: 10.1109/SCIS-ISIS.2014.7044899.

Ancona, M., E. Ceolini, C. Öztireli, and M. Gross (2017). “Towards better understanding of gradient-based attribution methods for deep neural networks”. In: *arXiv preprint arXiv:1711.06104*.

Asif, M. T. et al. (2014). “Spatiotemporal Patterns in Large-Scale Traffic Speed Prediction”. In: *IEEE Transactions on Intelligent Transportation Systems* 15.2, pp. 794–804. DOI: 10.1109/TITS.2013.2290285.

Asur, S. and B. A. Huberman (Aug. 2010). “Predicting the Future with Social Media”. In: *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology* 1, pp. 492–499. DOI: 10.1109/wi-iat.2010.63.

Aubry, M. and B. C. Russell (2015). “Understanding Deep Features with Computer-generated Imagery”. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2875–2883.

Bach, S., A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek (2015). “On Pixel-wise Explanations for Non-linear Classifier Decisions by Layer-wise Relevance Propagation”. In: *PloS one* 10.7, e0130140.

Baehrens, D., T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K. R. Müller (2010). “How to Explain Individual Classification Decisions”. In: *Journal of Machine Learning Research* 11, pp. 1803–1831. ISSN: 15324435. eprint: 0912.1128.

Bahdanau, D., K. Cho, and Y. Bengio (2016). *Neural Machine Translation by Jointly Learning to Align and Translate*. arXiv: 1409.0473 [cs.CL].

Bai, L., L. Yao, S. Kanhere, X. Wang, and Q. Sheng (2019). “STG2seq: Spatial-temporal Graph to Sequence Model for Multi-step Passenger Demand Forecasting”. In: *Proceedings of the 28th International Joint Conference on Artificial Intelligence, (IJCAI)*, pp. 1981–1987. DOI: 10.24963/ijcai.2019/274.

Balduzzi, D., M. Frean, L. Leary, J. Lewis, K. Wan-Duo Ma, and B. Mcwilliams (2017). “The Shattered Gradients Problem: If Resnets are the answer, then what is the question?” In: *Proceedings of the 34th International Conference on Machine Learning (ICML)*.

Bastani, O., C. Kim, and H. Bastani (2019). *Interpreting Blackbox Models via Model Extraction*. arXiv: 1705.08504 [cs.LG].

Binder, A., S. Bach, G. Montavon, K.-R. Müller, and W. Samek (2016). "Layer-Wise Relevance Propagation for Deep Neural Network Architectures". In: *Information Science and Applications (ICISA) 2016*. Ed. by K. J. Kim and N. Joukov. Springer Singapore, pp. 913–922. ISBN: 978-981-10-0557-2.

Burt, P. J. and E. H. Adelson (1983). "The Laplacian Pyramid as a Compact Image Code". In: *IEEE Transactions on Communications* COM-3.4, pp. 532–540.

Carvalho, S., L. Sarmento, and R. Rossetti (2012). *Real-Time Sensing of Traffic Information in Twitter Messages*.

Castro-Neto, M., Y.-S. Jeong, M.-K. Jeong, and L. Han (Apr. 2009). "Online-SVR for Short-term Traffic Flow Prediction under Typical and typical Traffic Conditions". In: *Expert Systems with Applications* 36, pp. 6164–6173. DOI: 10.1016/j.eswa.2008.07.069.

Chang, H., Y. Lee, B. Yoon, and S. Baek (Sept. 2012). "Dynamic Near-term Traffic Flow Prediction: System-oriented Approach based on Past Experiences". In: *Intelligent Transport Systems, IET* 6, pp. 292–305. DOI: 10.1049/iet-its.2011.0123.

Chen, K., F. Chen, et al. (Jan. 2020). "Dynamic Spatio-Temporal Graph-Based CNNs for Traffic Flow Prediction". In: *IEEE Access* 8, pp. 185136–185145. DOI: 10.1109/ACCESS.2020.3027375.

Chen, Q., X. Song, H. Yamada, and R. Shibasaki (2016). "Learning Deep Representation from Big and Heterogeneous Data for Traffic Accident Inference". In: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 338–344.

Cho, K., B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio (2014). *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. arXiv: 1406.1078 [cs.CL].

Chu, J. et al. (2018). "Passenger Demand Prediction with Cellular Footprints". In: *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pp. 1–9. DOI: 10.1109/SAHCN.2018.8397114.

Cui, Z., K. Henrickson, R. Ke, and Y. Wang (2019). "Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-scale Traffic Learning and Forecasting". In: *IEEE Transactions on Intelligent Transportation Systems* 21.11, pp. 4883–4894.

Cui, Z., R. Ke, and Y. Wang (2018). "Deep Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction". In: *ArXiv* abs/1801.02143.

Dabkowski, P. and Y. Gal (2017). "Real Time Image Saliency for Black Box Classifiers". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 6970–6979.

Ding, Q. Y., X. F. Wang, X. Y. Zhang, and Z. Q. Sun (Jan. 2011). "Forecasting Traffic Volume with Space-Time ARIMA Model". In: *Advanced Materials Research* 156, pp. 979–983. DOI: 10.4028/www.scientific.net/AMR.156-157.979.

Dosovitskiy, A. and T. Brox (2016). "Inverting visual representations with convolutional networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4829–4837.

Du, M., N. Liu, and X. Hu (2019). "Techniques for Interpretable Machine Learning". In: *Communications of the ACM* 63.1, pp. 68–77.

Edunov, S., M. Ott, M. Auli, and D. Grangier (2018). "Understanding Back-Translation at Scale". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 489–500. DOI: 10.18653/v1/d18-1045. arXiv: 1808.09381.

El Hatri, C. and J. Boumhidi (2018). "Fuzzy Deep Learning Based Urban Traffic Incident Detection". In: *Cognitive Systems Research* 50, pp. 206–213.

Elman, J. L. (1990). "Finding Structure in Time". In: *Cognitive Science* 14.2, pp. 179–211. DOI: 10.1207/s15516709cog1402_1.

Em, A., M. Sarvi, and S. Bagloee (July 2019). "Using Kalman filter Algorithm for Short-term Traffic Flow Prediction in a Connected Vehicle Environment". In: *Journal of Modern Transportation* 27, pp. 222–232. DOI: 10.1007/s40534-019-0193-2.

Erhan, D., Y. Bengio, A. Courville, and P. Vincent (2009). *Visualizing Higher-layer Features of a Deep Network*. Tech. rep. 3. University of Montreal.

Erion, G., J. D. Janizek, P. Sturmfels, S. Lundberg, and S.-I. Lee (2020). *Improving performance of deep learning models with axiomatic attribution priors and expected gradients*. arXiv: 1906.10670 [cs.LG].

Fan, F., J. Xiong, and G. Wang (2020). "On Interpretability of Artificial Neural Networks". In: arXiv: 2001.02522. URL: http://arxiv.org/abs/2001.02522.

Fang, S., Q. Zhang, G. Meng, S. Xiang, and C. Pan (July 2019). "GSTNet: Global Spatial-Temporal Network for Traffic Flow Prediction". In: *Proceedings of the 28th International Joint Conference on Artificial Intelligence, (IJCAI)*, pp. 2286–2293. DOI: 10.24963/ijcai.2019/317.

Fong, R. C. and A. Vedaldi (2017). "Interpretable Explanations of Black Boxes by Meaningful Perturbation". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3429–3437.

Fouladgar, M., M. Parchami, R. Elmasri, and A. Ghaderi (2017). "Scalable Deep Traffic Flow Neural Networks for Urban Traffic Congestion Prediction". In: *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2251–2258.

Fu, R., Z. Zhang, and L. Li (2016). "Using LSTM and GRU Neural Network Methods for Traffic Flow Prediction". In: pp. 324–328.

Fukushima, K. and S. Miyake (1982). "Neocognitron: A New Algorithm for Pattern Recognition Tolerant of Deformations and Shifts In Position". In: *Pattern Recognition* 15.6, pp. 455–469. ISSN: 0031-3203. DOI: 10.1016/0031-3203(82)90024-3.

Gal, A., A. Mandelbaum, F. Schnitzler, A. Senderovich, and M. Weidlich (Mar. 2017). "Traveling Time Prediction in Scheduled Transportation with Journey Segments". In: *Information Systems* 64, pp. 266–280. DOI: 10.1016/j.is.2015.12.001.

Gao, J., Y. Shen, J. Liu, M. Ito, and N. Shiratori (2017). "Adaptive Traffic Signal Control: Deep Reinforcement Learning Algorithm with Experience Replay and Target Network". In: *arXiv preprint arXiv:1705.02755*.

Genders, W. and S. Razavi (2016). "Using a Deep Reinforcement Learning Agent for Traffic Signal Control". In: *arXiv preprint arXiv:1611.01142*.

Geng, X., Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu (July 2019). "Spatiotemporal Multi-Graph Convolution Network for Ride-Hailing Demand Forecasting". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01, pp. 3656–3663. DOI: 10.1609/aaai.v33i01.33013656.

Gilpin, L. H., D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal (2019). "Explaining Explanations: An overview of Interpretability of Machine Learning". In: *Proceedings of the 2018 IEEE 5th International Conference on Data Science and Advanced Analytics, DSAA 2018*, pp. 80–89. ISBN: 9781538650905. DOI: 10.1109/DSAA.2018.00018. eprint: 1806.00069.

Goh, G. S. W., S. Lapuschkin, L. Weber, W. Samek, and A. Binder (2021). "Understanding Integrated Gradients with SmoothTaylor for Deep Neural Network Attribution". In: *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 4949–4956. DOI: 10.1109/ICPR48806.2021.9413242.

Guo, J., W. Huang, and B. M. Williams (June 2014). "Adaptive Kalman Filter Approach for Stochastic Short-term Traffic Flow Rate Prediction and Uncertainty Quantification". In: *Transportation Research Part C: Emerging Technologies*, pp. 50–64. DOI: 10.1016/j.trc.2014.02.006.

Guo, S., Y. Lin, N. Feng, C. Song, and H. Wan (July 2019). "Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01, pp. 922–929. DOI: 10.1609/aaai.v33i01.3301922.

Habtie, A. B., A. Abraham, and D. Midekso (2015). "Cellular Network Based Real-Time Urban Road Traffic State Estimation Framework Using Neural Network Model Estimation". In: *2015 IEEE Symposium Series on Computational Intelligence*, pp. 38–44. DOI: 10.1109/SSCI.2015.16.

Hamed, M. M., H. R. Al-Masaeid, and Z. M. Bani Said (1995). "Short-term Prediction of Traffic Volume in Urban Arterials". In: *Journal of Transportation Engineering* 121, pp. 249–254.

He, J., W. Shen, P. Divakaruni, L. Wynter, and R. Lawrence (Aug. 2013). "Improving Traffic Prediction with Tweet Semantics". In: *IJCAI*, pp. 1387–1393.

He, K., X. Zhang, S. Ren, and J. Sun (2015). "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

He, Z., C.-Y. Chow, and J.-D. Zhang (2019). "STCNN: A Spatio-Temporal Convolutional Neural Network for Long-Term Traffic Prediction". In: *Proceedings of the 20th IEEE International Conference on Mobile Data Management (MDM)*, pp. 226–233. DOI: 10.1109/MDM.2019.00-53.

Hochreiter, S. and J. Schmidhuber (Dec. 1997). "Long Short-term Memory". In: *Neural computation* 9, pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

Hong, W.-C., Y. Dong, W.-M. Hung, and S.-Y. Wei (2010). "Seasonal Adjustment in a SVR with Chaotic Simulated Annealing Algorithm Traffic Flow Forecasting Model". In: *2010 Second World Congress on Nature and Biologically Inspired Computing (NaBIC)*, pp. 560–565. DOI: 10.1109/NABIC.2010.5716266.

Hooker, S., D. Erhan, P.-J. Kindermans, and B. Kim (2019). "A Benchmark for Interpretability Methods in Deep Neural Networks". In: *Advances in Neural Information Processing Systems*, pp. 9737–9748.

Hu, M. and B. Liu (2004). "Mining and Summarizing Customer Reviews". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 168–177. DOI: 10.1145/1014052.1014073.

Huang, G., Z. Liu, L. Van Der Maaten, and K. Q. Weinberger (2017). "Densely Connected Convolutional Networks". In: *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269. ISBN: 9781538604571. DOI: 10.1109/CVPR.2017.243. arXiv: 1608.06993.

Huang, H., Q. Tang, and Z. Liu (Jan. 2013). "Adaptive Correction Forecasting Approach for Urban Traffic Flow Based on Fuzzy-Mean Clustering and Advanced Neural Network". In: *Journal of Applied Mathematics*.

Huang, S.-H. and B. Ran (June 1995). "An Application of Neural Network on Traffic Speed Prediction Under Adverse Weather Condition". In: *Proceedings of the Transportation Research Board 82nd Annual Meeting*.

Huang, W., G. Song, H. Hong, and K. Xie (2014). "Deep Architecture for Traffic Flow Prediction: Deep Belief Networks With Multitask Learning". In: *IEEE Transactions on Intelligent Transportation Systems* 15.5, pp. 2191–2201. DOI: 10.1109/TITS.2014.2311123.

Ide, T. and M. Sugiyama (Aug. 2011). "Trajectory Regression on Road Networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 25, pp. 203–208.

Jagadish, H. V., J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, and C. Shahabi (July 2014). "Big Data and Its Technical Challenges". In: *Communications of the ACM* 57.7, pp. 86–94. ISSN: 0001-0782. DOI: 10.1145/2611567.

Jepsen, T. S., C. S. Jensen, and T. D. Nielsen (2020). "Relational Fusion Networks: Graph Convolutional Networks for Road Networks". In: *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12. DOI: 10.1109/TITS.2020.3011799.

Jia, Y., J. Wu, and Y. Du (2016). "Traffic Speed Prediction using Deep Learning Method". In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1217–1222. DOI: 10.1109/ITSC.2016.7795712.

Jiang, B. and Y. Fei (2015). "Traffic and Vehicle Speed Prediction with Neural Network and Hidden Markov model in Vehicular Networks". In: *2015 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1082–1087. DOI: 10.1109/IVS.2015.7225828.

Jin, X., Y. Zhang, and D. Yao (2007). "Simultaneously Prediction of Network Traffic Flow Based on PCA-SVR". In: *International Symposium on Neural Networks*, pp. 1022–1031.

Jindal, I., Tony, Qin, X. Chen, M. Nokleby, and J. Ye (2017). "A Unified Neural Network Approach for Estimating Travel Time and Distance for a Taxi Trip". In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3655–3661.

Johnson, M. et al. (2017). "Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation". In: *Transactions of the Association for Computational Linguistics* 5, pp. 339–351. ISSN: 2307-387X. DOI: 10.1162/tacl_a_00065. arXiv: 1611.04558.

Kádár, A., G. Chrupała, and A. Alishahi (2017). "Representation of Linguistic Form and Function in Recurrent Neural Networks". In: *Computational Linguistics* 43.4, pp. 761–780.

Karpathy, A., J. Johnson, and L. Fei-Fei (2016). "Visualizing and Understanding Recurrent Networks". In: *International Conference on Learning Representations, ICLR Workshop Track Proceedings*.

Kazhdan, D., B. Dimanov, M. Jamnik, and P. Liò (2020). *MEME: Generating RNN Model Explanations via Model Extraction*. arXiv: 2012.06954 [cs.LG].

Kindermans, P.-J., K. T. Schütt, M. Alber, K.-R. Müller, D. Erhan, B. Kim, and S. Dähne (2018). "Learning How to Explain Neural Networks: PatternNet and PatternAttribution". In: *International Conference on Learning Representations (ICLR)*.

Koesdwiady, A., R. Soua, and F. Karray (2016). "Improving Traffic Flow Prediction With Weather Information in Connected Cars: A Deep Learning Approach". In: *IEEE Transactions on Vehicular Technology* 65.12, pp. 9508–9517. DOI: 10.1109/TVT.2016.2585575.

Koh, P. W. and P. Liang (2017). "Understanding Black-box Predictions via Influence Functions". In: *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 1885–1894.

Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'12, pp. 1097–1105.

Krumm, J. and E. Horvitz (2004). "LOCADIO: Inferring Motion and Location from Wi-Fi Signal Strengths". In: *The 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, MOBIQUITOUS 2004*, pp. 4–13. DOI: 10.1109/MOBIQ.2004.1331705.

Kuang, Y. Yunjun, Tan, Y.-S. Li, and J. Yang (May 2019). "Predicting Taxi Demand Based on 3D Convolutional Neural Network and Multi-task Learning". In: *Remote Sensing* 11, p. 1265. DOI: 10.3390/rs11111265.

Lee, H., P. Pham, Y. Largman, and A. Ng (2009). "Unsupervised Feature Learning for Audio Classification using Convolutional Deep Belief Networks". In: *Advances in Neural Information Processing Systems*. Ed. by Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta. Vol. 22, pp. 1096–1104.

Leelavathi and S. Devi (2016). "An Architecture of Deep Learning Method to Predict Traffic Flow in Big Data". In: *International Journal of Research in Engineering and Technology* 05, pp. 461–468.

Leshem, G. and Y. Ritov (2007). "Traffic Flow Prediction using Adaboost Algorithm with Random Forests as a Weak Learner". In: *Proceedings of World Academy of Science, Engineering and Technology*. Vol. 19, pp. 193–198.

Li, C. and C. Yang (2016). "The Research on Traffic Sign Recognition based on Deep Learning". In: *2016 16th International Symposium on Communications and Information Technologies (ISCIT)*, pp. 156–161. DOI: 10.1109/ISCIT.2016.7751612.

Li, L., Y. Lv, and F.-Y. Wang (2016). "Traffic Signal Timing via Deep Reinforcement Learning". In: *IEEE/CAA Journal of Automatica Sinica* 3.3, pp. 247–254. DOI: 10.1109/JAS.2016.7508798.

Li, L., S. He, J. Zhang, and B. Ran (Feb. 2017). "Short-term Highway Traffic Flow Prediction based on a Hybrid Strategy Considering Temporal–spatial Information". In: *Journal of Advanced Transportation* 50. DOI: 10.1002/atr.1443.

Li, Y., K. Fu, Z. Wang, C. Shahabi, J. Ye, and Y. Liu (July 2018). "Multi-task Representation Learning for Travel Time Estimation". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1695–1704. DOI: 10.1145/3219819.3220033.

Li, Y., R. Yu, C. Shahabi, and Y. Liu (2018). "Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting". In: *International Conference on Learning Representations (ICLR)*.

Li, Y., A. Møgelmose, and M. M. Trivedi (2016). "Pushing the "Speed Limit": High-Accuracy US Traffic Sign Recognition With Convolutional Neural Networks". In: *IEEE Transactions on Intelligent Vehicles* 1.2, pp. 167–176. DOI: 10.1109/TIV.2016.2615523.

Liao, B., J. Zhang, et al. (2018). "Deep Sequence Learning with Auxiliary Information for Traffic Prediction". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.

Liao, L., D. J. Patterson, D. Fox, and H. Kautz (2007). "Learning and Inferring Transportation Routines". In: *Artificial Intelligence* 171.5, pp. 311–331. ISSN: 0004-3702. DOI: 10.1016/j.artint.2007.01.006.

Lint, J. W. C. van (2008). "Online Learning Solutions for Freeway Travel Time Prediction". In: *IEEE Transactions on Intelligent Transportation Systems* 9.1, pp. 38–47. DOI: 10.1109/TITS.2008.915649.

Liu, H. and I. Lee (2017). "End-to-end Trajectory Transportation Mode Classification using Bi-LSTM Recurrent Neural Network". In: *Proceedings of the 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, pp. 1–5. DOI: 10.1109/ISKE.2017.8258799.

Liu, H., H. Wu, W. Sun, and I. Lee (2019). "Spatio-Temporal GRU for Trajectory Classification". In: *IEEE International Conference on Data Mining (ICDM)*, pp. 1228–1233. DOI: 10.1109/ICDM.2019.00152.

Lu, Y. (2015). "Unsupervised Learning on Neural Network Outputs: with Application in Zero-shot Learning". In: *arXiv preprint arXiv:1506.00990*.

Lundberg, S. M. and S.-I. Lee (2017). "A Unified Approach to Interpreting Model Predictions". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 4768–4777.

Lv, Y., Y. Chen, X. Zhang, Y. Duan, and N. L. Li (Jan. 2017). "Social media based transportation research: the state of the work and the networking". In: *IEEE/CAA Journal of Automatica Sinica* 4, pp. 19–26. DOI: 10.1109/JAS.2017.7510316.

Lv, Y., Y. Duan, W. Kang, Z. Li, and F.-Y. Wang (Apr. 2015). "Traffic Flow Prediction With Big Data: A Deep Learning Approach". In: *IEEE Transactions on Intelligent Transportation Systems* 16, pp. 865–873. DOI: 10.1109/tits.2014.2345663.

Lv, Z., J. Xu, K. Zheng, H. Yin, P. Zhao, and X. Zhou (July 2018). "LC-RNN: A Deep Learning Model for Traffic Speed Prediction". In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI*, pp. 3470–3476. DOI: 10.24963/ijcai.2018/482.

Ma, X., Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang (Apr. 2017). "Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction". In: *Sensors* 17, p. 818. DOI: 10.3390/s17040818.

Mahendran, A. and A. Vedaldi (May 2016). "Visualizing Deep Convolutional Neural Networks Using Natural Pre-images". In: *International Journal of Computer Vision* 120.3, pp. 233–255. ISSN: 1573-1405. DOI: 10.1007/s11263-016-0911-8.

Mnih, V., N. Heess, A. Graves, et al. (2014). "Recurrent Models of Visual Attention". In: *Advances in neural information processing systems*, pp. 2204–2212.

Montavon, G., S. Lapuschkin, A. Binder, W. Samek, and K. R. Müller (2017). "Explaining Nonlinear Classification Decisions with Deep Taylor Decomposition". In: *Pattern Recognition* 65, pp. 211–222. ISSN: 00313203. DOI: 10.1016/j.patcog.2016.11.008. arXiv: 1512.02479.

Moosavi-Dezfooli, S. M., A. Fawzi, O. Fawzi, and P. Frossard (2017). "Universal Adversarial Perturbations". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 86–94. ISBN: 9781538604571. DOI: 10.1109/CVPR.2017.17. eprint: 1705.09554.

Mordvintsev, A., C. Olah, and M. Tyka (2015). *Inceptionism: Going Deeper into Neural Networks*. URL: https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html.

Nguyen, A. and C. L. Date (2015). "Deep Neural Networks are Easily Fooled : High Confidence Predictions for Unrecognizable Images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. ISBN: 9781467369640. DOI: 10.1109/CVPR.2015.7298640. eprint: arXiv:1412.1897v1.

Nguyen, A., J. Yosinski, and J. Clune (2016). "Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks". In: *Visualization for Deep Learning Workshop, International Conference in Machine Learning (ICML)*.

Ni, M., Q. He, and J. Gao (2014). *Using Social Media to Predict Traffic Flow under Special Event Conditions*.

Niu, X., Y. Zhu, Q. Cao, X. Zhang, W. Xie, and K. Zheng (Aug. 2015). "An Online-Traffic-Prediction Based Route Finding Mechanism for Smart City". In: *International Journal of Distributed Sensor Networks* 11, p. 970256. DOI: 10.1155/2015/970256.

Nolte, M., N. Kister, and M. Maurer (2018). "Assessment of Deep Convolutional Neural Networks for Road Surface Classification". In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 381–386.

Novak, R., Y. Bahri, D. A. Abolafia, J. Pennington, and J. Sohl-Dickstein (2018). "Sensitivity and Generalization in Neural Networks: an Empirical Study". In: *International Conference on Learning Representations (ICLR)*. eprint: 1802.08760.

Olah, C., A. Mordvintsev, and L. Schubert (2017). "Feature visualization". In: *Distill* 2.11.

Patterson, D., L. Liao, D. Fox, and H. Kautz (Aug. 2003). "Inferring High-Level Behavior from Low-Level Sensors". In: vol. 2864, pp. 73–89. ISBN: 978-3-540-20301-8. DOI: 10.1007/978-3-540-39653-6_6.

Petrović, S., M. Osborne, and V. Lavrenko (June 2010). "Streaming First Story Detection with Application to Twitter". In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT '10)*, pp. 181–189.

Qin, Y., H. Luo, F. Zhao, C. Wang, J. Wang, and Y. Zhang (2019). "Toward Transportation Mode Recognition Using Deep Convolutional and Long Short-Term Memory Recurrent Neural Networks". In: *IEEE Access* 7, pp. 142353–142367. DOI: 10.1109/ACCESS.2019.2944686.

Rahmani, M., E. Jenelius, and H. N. Koutsopoulos (Oct. 2013). "Route Travel Time Estimation Using Low-Frequency Floating Car Data". In: pp. 2292–2297. DOI: 10.1109/ITSC.2013.6728569.

Ramakrishnan, N. and T. Soni (2018). "Network Traffic Prediction Using Recurrent Neural Networks". In: *Proceeding of the 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 187–193. DOI: 10.1109/ICMLA.2018.00035.

Ramanna, S., C. Sengoz, S. Kehler, and D. Pham (2021). "Near Real-time Map Building with Multi-class Image Set Labeling and Classification of Road Conditions Using Convolutional Neural Networks". In: *Applied Artificial Intelligence*, pp. 1–31. DOI: 10.1080/08839514.2021.1935590.

Rayhan, Y. and T. Hashem (2020). *AIST: An Interpretable Attention-based Deep Learning Model for Crime Prediction*. arXiv: 2012.08713.

Ren, S., K. He, R. Girshick, and J. Sun (2017). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6, pp. 1137–1149. ISSN: 01628828. DOI: 10.1109/TPAMI.2016.2577031. arXiv: 1506.01497.

Ribeiro, M. T., S. Singh, and C. Guestrin (2016). ""Why Should I Trust You?" Explaining the Predictions of Any Classifier". In: *22nd ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1135–1144. ISBN: 9781450342322. arXiv: 1710.10720.

Russakovsky, O. et al. (2015). "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3, pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

Sabour, S., N. Frosst, and G. E. Hinton (2017). "Dynamic Routing Between Capsules". In: *Advances in Neural Information Processing Systems (NIPS)*.

Samek, W., A. Binder, G. Montavon, S. Bach, and Klaus-Robert Muller (2017). "Evaluating the Visualization of What a Deep Neural Network Has Learned". In: *IEEE Transactions on Neural Networks and Learning Systems* 8.11, pp. 2660–2673.

Samek, W., G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller (Eds.) (2019). *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Vol. 11700. LNCS. Springer, pp. 1–435.

Sato, A. and K. Yamada (1995). "Generalized Learning Vector Quantization". In: *NIPS*. Vol. 95, pp. 423–429.

Schölkopf, B. and A. Smola (2003). "Kernel Methods and Support Vector Machines". In: *Encyclopedia of Biostatistics*.

Selvaraju, R. R., M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra (2017). "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626. DOI: 10.1109/ICCV.2017.74.

Semwal, D., S. Patil, S. Galhotra, A. Arora, and N. Unny (Mar. 2015). "STAR: Real-time Spatio-Temporal Analysis and Prediction of Traffic Insights using Social Media". In: *Proceedings of the 2nd IKDD Conference on Data Sciences*. Association for Computing Machinery, pp. 1–4. DOI: 10.1145/2778865.2778872.

Shekhar, H., S. Setty, and U. Mudenagudi (Sept. 2016). "Vehicular Traffic Analysis from Social Media Data". In: *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, pp. 1628–1634. DOI: 10.1109/ICACCI.2016.7732281.

Shrikumar, A., P. Greenside, and A. Kundaje (2017). “Learning Important Features through Propagating Activation Differences”. In: *34th International Conference on Machine Learning, ICML 2017*. Vol. 7, pp. 4844–4866. ISBN: 9781510855144. arXiv: 1704.02685.

Simonyan, K., A. Vedaldi, and A. Zisserman (2014). “Deep inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”. In: *2nd International Conference on Learning Representations, ICLR 2014 - Workshop Track Proceedings*, pp. 1–8. arXiv: 1312.6034.

Smilkov, D., N. Thorat, B. Kim, F. Viégas, and M. Wattenberg (2017). “SmoothGrad: Removing Noise by Adding Noise”. In: *Workshop on Visualization for Deep Learning, International Conference on Machine Learning (ICML)*. arXiv: 1706.03825. URL: http://arxiv.org/abs/1706.03825.

Sohn, T. et al. (2006). “Mobility Detection Using Everyday GSM Traces”. In: *UbiComp 2006: Ubiquitous Computing*. Ed. by P. Dourish and A. Friday, pp. 212–224. ISBN: 978-3-540-39635-2.

Song, H. Y., M. S. Baek, and M. Sung (2019). “Generating Human Mobility Route Based on Generative Adversarial Network”. In: *2019 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pp. 91–99. DOI: 10.15439/2019F320.

Springenberg, J. T., A. Dosovitskiy, T. Brox, and M. Riedmiller (2015). “Striving for Simplicity: The All Convolutional Net”. In: *3rd International Conference on Learning Representations, ICLR 2015 - Workshop Track Proceedings*, pp. 1–14. arXiv: 1412.6806.

Stenneth, L., O. Wolfson, P. Yu, and B. Xu (Nov. 2011). “Transportation Mode Detection using Mobile Phones and GIS Information”. In: *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, pp. 54–63. DOI: 10.1145/2093973.2093982.

Su, J., D. V. Vargas, and K. Sakurai (Oct. 2019). “One Pixel Attack for Fooling Deep Neural Networks”. In: *IEEE Transactions on Evolutionary Computation* 23.5, pp. 828–841. ISSN: 1941-0026. DOI: 10.1109/tevc.2019.2890858.

Sun, F., A. Dubey, and J. White (2017). “DxNAT — Deep Neural Networks for Explaining Non-recurring Traffic Congestion”. In: *2017 IEEE International Conference on Big Data (Big Data)*, pp. 2141–2150. DOI: 10.1109/BigData.2017.8258162.

Sun, S., H. Wu, and L. Xiang (2020). “City-wide Traffic Flow Forecasting Using a Deep Convolutional Neural Network”. In: *Sensors* 20.2, p. 421.

Sundararajan, M., A. Taly, and Q. Yan (2017). “Axiomatic Attribution for Deep Networks”. In: *34th International Conference on Machine Learning (ICML)*. Vol. 7, pp. 5109–5118. ISBN: 9781510855144. arXiv: 1703.01365.

Szegedy, C., W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus (2014). *Intriguing Properties of Neural Networks*. arXiv: 1312.6199 [cs.CV].

Tan, M. and Q. V. Le (2019). “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”. In: *36th International Conference on Machine Learning (ICML)*, pp. 10691–10700. ISBN: 9781510886988. arXiv: 1905.11946.

Tang, J., X. Chen, Z. Hu, F. Zong, C. Han, and L. Li (Sept. 2019). “Traffic Flow Prediction based on Combination of Support Vector Machine and Data Denoising Schemes”. In: *Physica A: Statistical Mechanics and its Applications* 534, p. 120642. DOI: 10.1016/j.physa.2019.03.007.

Toutanova, K. and C. D. Manning (2000). "Enriching the Knowledge Sources used in a Maximum Entropy Part-of-speech Tagger". In: *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics*. Vol. 13, pp. 63–70. DOI: 10.3115/1117794.1117802.

Tran, T., Z. Ma, H. Li, L. Hao, and T. Quang Khai (Jan. 2015). "A Multiplicative Seasonal ARIMA/GARCH Model in EVN Traffic Prediction". In: *International Journal of Communications, Network and System Sciences* 08, pp. 43–49. DOI: 10.4236/ijcns.2015.84005.

Tumasjan, A., T. Sprenger, P. Sandner, and I. Welpe (2010). "Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment". In: *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*. Vol. 10, pp. 178–185.

Van der Pol, E. and F. A. Oliehoek (2016). "Coordinated Deep Reinforcement Learners for Traffic Light Control". In: *Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016)*.

Van Der Voort, M., M. Dougherty, and S. Watson (1996). "Combining Kohonen Maps with ARIMA Time Series Models to Forecast Traffic Flow". In: *Transportation Research Part C: Emerging Technologies* 4.5, pp. 307–318. ISSN: 0968-090X. DOI: 10.1016/S0968-090X(97)82903-8.

Wang, C., H. Luo, F. Zhao, and Y. Qin (Apr. 2020). "Combining Residual and LSTM Recurrent Networks for Transportation Mode Detection Using Multimodal Sensors Integrated in Smartphones". In: *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13. DOI: 10.1109/TITS.2020.2987598.

Wang, D., A. Al-Rubaie, J. Davies, and S. S. Clarke (Dec. 2014). "Real Time Road Traffic Monitoring Alert based on Incremental Learning from Tweets". In: *Evolving and Autonomous Learning Systems (EALS) 2014 IEEE Symposium*, pp. 50–57. DOI: 10.1109/EALS.2014.7009503.

Wang, D., W. Cao, J. Li, and J. Ye (2017). "DeepSD: Supply-Demand Prediction for Online Car-Hailing Services Using Deep Neural Networks". In: *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pp. 243–254. DOI: 10.1109/ICDE.2017.83.

Wang, D., J. Zhang, W. Cao, J. Li, and Y. Zheng (2018). "When Will You Arrive? Estimating Travel Time Based on Deep Neural Networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 2500–2507.

Wang, H., X. Tang, Y.-H. Kuo, D. Kifer, and Z. Li (Feb. 2019). "A Simple Baseline for Travel Time Estimation using Large-scale Trip Data". In: *ACM Transactions on Intelligent Systems and Technology* 10, pp. 1–22. DOI: 10.1145/3293317.

Wang, J., Q. Gu, J. Wu, G. Liu, and Z. Xiong (2016). "Traffic Speed Prediction and Congestion Source Exploration: A Deep Learning Method". In: *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 499–508. DOI: 10.1109/ICDM.2016.0061.

Wang, W. and X. Li (2018). *Travel Speed Prediction with a Hierarchical Convolutional Neural Network and Long Short-Term Memory Model Framework*. arXiv: 1809.01887 [cs.LG].

Wang, Y., D. Zhang, Y. Liu, B. Dai, and L. H. Lee (2019). "Enhancing Transportation Systems via Deep Learning: A Survey". In: *Transportation Research Part C: Emerging Technologies* 99, pp. 144–163. ISSN: 0968-090X. DOI: 10.1016/j.trc.2018.12.004.

Wang, Z. and Y. Zhang (2017). "DDoS Event Forecasting using Twitter Data *". In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4151–4157.

Wei, D., B. Zhou, A. Torrabla, and W. Freeman (2015). *Understanding Intra-Class Knowledge Inside CNN*. arXiv: 1507.02379 [cs.CV].

Williams, B. and L. Hoel (Nov. 2003). "Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical Results". In: *Journal of Transportation Engineering* 129, pp. 664–672. DOI: 10.1061/(ASCE)0733-947X(2003)129:6(664).

Wu, C.-H., J.-M. Ho, and D. Lee (2004). "Travel-time prediction with support vector regression". In: *IEEE Transactions on Intelligent Transportation Systems* 5.4, pp. 276–281. DOI: 10.1109/TITS.2004.837813.

Wu, H., Z. Chen, W. Sun, B. Zheng, and W. Wang (2017). "Modeling Trajectories with Recurrent Neural Networks". In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence, (IJCAI)*, pp. 3083–3090. DOI: 10.24963/ijcai.2017/430.

Wu, T., K. Xie, D. Xinpin, and G. Song (2012). "A Online Boosting Approach for Traffic Flow Forecasting under Abnormal Conditions". In: *9th International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 2555–2559. DOI: 10.1109/FSKD.2012.6234335.

Wu, Y. and H. Tan (2016). "Short-term Traffic Flow Forecasting with Spatial-temporal Correlation in a Hybrid Deep Learning Framework". In: *ArXiv* abs/1612.01022.

Wu, Y., H. Tan, L. Qin, B. Ran, and Z. Jiang (2018). "A Hybrid Deep Learning based Traffic Flow Prediction Method and its Understanding". In: *Transportation Research Part C: Emerging Technologies* 90, pp. 166–180. ISSN: 0968-090X. DOI: 10.1016/j.trc.2018.03.001.

Xiaojian, G. and Z. Quan (2009). "A Traffic Flow Forecasting Model based on BP Neural Network". In: *2009 2nd International Conference on Power Electronics and Intelligent Transportation System (PEITS)*. Vol. 3, pp. 311–314. DOI: 10.1109/PEITS.2009.5406865.

Xie, G., Q. Li, and Y. Jiang (2021). "Self-attentive Deep Learning Method for Online Traffic Classification and its Interpretability". In: *Computer Networks* 196, p. 108267. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2021.108267.

Xu, S., S. Li, and R. Wen (June 2018). "Sensing and Detecting Traffic Events using Geosocial Media Data: A Review". In: *Computers Environment and Urban Systems* 72. DOI: 10.1016/j.compenvurbsys.2018.06.006.

Xu, Y., Q.-J. Kong, R. Klette, and Y. Liu (Dec. 2014). "Accurate and Interpretable Bayesian MARS for Traffic Flow Prediction". In: *IEEE Transactions on Intelligent Transportation Systems* 15, pp. 2457–2469. DOI: 10.1109/TITS.2014.2315794.

Xu, Y. and D. Li (Sept. 2019). "Incorporating Graph Attention and Recurrent Architectures for City-Wide Taxi Demand Prediction". In: *ISPRS International Journal of Geo-Information* 8, p. 414. DOI: 10.3390/ijgi8090414.

Yang, B., C. Guo, and C. S. Jensen (July 2013). “Travel Cost Inference from Sparse, Spatio-temporally Correlated Time Series using Markov models”. In: *Proceedings of the Very Large Data Base (VLDB) Endowment* 6, pp. 769–780. DOI: 10.14778/2536360.2536375.

Yang, H.-F., T. S. Dillon, and Y.-P. P. Chen (2017). “Optimized Structure of the Traffic Flow Forecasting Model With a Deep Learning Approach”. In: *IEEE Transactions on Neural Networks and Learning Systems* 28.10, pp. 2371–2381. DOI: 10.1109/TNNLS.2016.2574840.

Yao, H., X. Tang, H. Wei, G. Zheng, and Z. Li (2019). “Revisiting Spatial-Temporal Similarity: A Deep Learning Framework for Traffic Prediction”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 5688–5675.

Yao, H., F. Wu, et al. (Apr. 2018). “Deep Multi-View Spatial-Temporal Network for Taxi Demand Prediction”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1.

Ye, J., L. Sun, B. Du, Y. Fu, and H. Xiong (2021). “Coupled Layer-wise Graph Convolution for Transportation Demand Prediction”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*.

Yin, J., X. Chai, and Q. Yang (2004). “High-Level Goal Recognition in a Wireless LAN”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 578–584.

Yosinski, J., J. Clune, A. Nguyen, T. Fuchs, and H. Lipson (2015). “Understanding Neural Networks through Deep Visualization”. In: *arXiv preprint arXiv:1506.06579*.

Yu, H., Z. Wu, S. Wang, Y. Wang, and X. Ma (June 2017). “Spatiotemporal Recurrent Convolutional Networks for Traffic Prediction in Transportation Networks”. In: *Sensors* 27, p. 1501. DOI: 10.3390/s17071501.

Yu, R., Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu (2017). “Deep Learning: A Generic Approach for Extreme Condition Traffic Forecasting”. In: *Proceedings of the 2017 SIAM International Conference on Data Mining (SDM)*, pp. 777–785. DOI: 10.1137/1.9781611974973.87.

Yuan, H., G. Li, Z. Bao, and L. Feng (2020). “Effective Travel Time Estimation: When Historical Trajectories over Road Networks Matter”. In: *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. SIGMOD ’20, pp. 2135–2149. ISBN: 9781450367356. DOI: 10.1145/3318464.3389771.

Zeiler, M. D. and R. Fergus (2014). “Visualizing and Understanding Convolutional Networks”. In: *European Conference on Computer Vision (ECCV)*, pp. 818–833. ISSN: 16113349. DOI: 10.1007/978-3-319-10590-1_53. arXiv: 1311.2901.

Zeng, Y., X. Xu, D. Shen, Y. Fang, and Z. Xiao (2017). “Traffic Sign Recognition Using Kernel Extreme Learning Machines With Deep Perceptual Features”. In: *IEEE Transactions on Intelligent Transportation Systems* 18.6, pp. 1647–1653. DOI: 10.1109/TITS.2016.2614916.

Zhang, H., H. Wu, W. Sun, and B. Zheng (July 2018). “DeepTravel: a Neural Network Based Travel Time Estimation Model with Auxiliary Supervision”. In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 3655–3661. DOI: 10.24963/ijcai.2018/508.

Zhang, J., Q. Huang, H. Wu, and Y. Liu (2017). “A Shallow Network with Combined Pooling for Fast Traffic Sign Recognition”. In: *Information* 8.2. ISSN: 2078-2489. DOI: 10.3390/info8020045.

Zhang, J., Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li (2018). "Predicting Citywide Crowd Flows using Deep Spatio-temporal Residual Networks". In: vol. 259, pp. 147–166. DOI: 10.1016/j.artint.2018.03.002.

Zhang, Q., W. Wang, and S.-C. Zhu (Apr. 2018). "Examining CNN Representations With Respect to Dataset Bias". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1.

Zhang, Q., Y. N. Wu, and S.-C. Zhu (2018). "Interpretable Convolutional Neural Networks". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8827–8836. DOI: 10.1109/CVPR.2018.00920.

Zhang, Q., Y. Yang, H. Ma, and Y. N. Wu (2019). "Interpreting CNNs via Decision Trees". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6261–6270.

Zhang, Q. and S.-C. Zhu (2018). "Visual Interpretability for Deep learning: A Survey". In: *Frontiers of Information Technology and Electronic Engineering* 19, pp. 27–39.

Zhang, Z., Q. He, J. Gao, and M. Ni (2018). "A Deep Learning Approach for Detecting Traffic Accidents from Social Media Data". In: *Transportation Research Part C: Emerging Technologies* 86, pp. 580–596. ISSN: 0968-090X. DOI: 10.1016/j.trc.2017.11.027.

Zhao, L., Y. Song, et al. (2020). "T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction". In: *IEEE Transactions on Intelligent Transportation Systems* 21.9, pp. 3848–3858. DOI: 10.1109/TITS.2019.2935152.

Zhao, Z., W. Chen, X. Wu, P. C. Y. Chen, and J. Liu (2017). "LSTM network: A Deep Learning Approach for Short-term Traffic Forecast". In: *IET Intelligent Transport Systems* 11.2, pp. 68–75. DOI: 10.1049/iet-its.2016.0208.

Zheng, J. and L. M. Ni (June 2013). "Time-Dependent Trajectory Regression on Road Networks via Multi-Task Learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 27, pp. 1048–1055.

Zheng, W. and D.-H. Lee (Feb. 2006). "Short-Term Freeway Traffic Flow Prediction: Bayesian Combined Neural Network Approach". In: *Journal of Transportation Engineering* 132.2, pp. 114–121. DOI: 10.1061/(ASCE)0733-947X(2006)132:2(114).

Zheng, Y., X. Xie, and W.-Y. Ma (Sept. 2008). "Understanding Mobility Based on GPS Data". In: *Proceedings of the 10th ACM conference on Ubiquitous Computing (Ubicomp 2008)*.

Zhou, B., A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba (2016). "Learning Deep Features for Discriminative Localization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2929. DOI: 10.5465/ambpp.2004.13862426.

Zhu, Y., Y. Zheng, L. Zhang, D. Santani, X. Xie, and Q. Yang (2012). *Inferring Taxi Status Using GPS Trajectories*. arXiv: 1205.4378.

Zhu, Z., B. Peng, C. Xiong, and L. Zhang (June 2016). "Short-Term Traffic Flow Prediction with Linear Conditional Gaussian Bayesian Network". In: *Journal of advanced transportation* 50. DOI: 10.1002/atr.1392.

Zintgraf, L. M., T. S. Cohen, and M. Welling (2016). "A New Method to Visualize Deep Neural Networks". In: *Workshop on Visualization for Deep Learning, International Conference on Machine Learning (ICML)*. eprint: 1603.02518.