A Bus Delay Prediction Model Based on Machine Learning Algorithms

Ma, Carl Ka To Fang, Hugo Wen, Cato Yuan

Shanghai High School International Division July 31, 2020

The Chinese version of this essay (page 18) is translated from the English version and is for reference only. In case of any discrepancy in the meaning of wording between the Chinese version and the English version, the English version shall prevail.

此论文的中文版本（第 18 页）按照英文版本翻译，并仅供参考。中文版本中的用语的含义如果有与英文版本有出入的，以英文版本为准。

ABSTRACT

Bus delay prediction is a practical feature made available in many smart cities, allowing people to estimate their travel time when taking public transportation. Existing algorithms are built on the prerequisite that every vehicle in the bus network is equipped with GPS capable of sharing real-time location. This is not the case in developing cities. Therefore, the aim of this paper is to introduce a bus delay prediction model that only requires common, accessible data like date and time, weather, bus route, and traffic incident reports. While these broad categories of data have little meaning, they are preprocessed into specific factors that show a much stronger correlation with bus delay, such as peak or off-peak hours, visibility, precipitation, etc. A total of 24 input features are passed into a feed-forward neural network, trained with data collected in Toronto, Canada, over the past six years. While Toronto is used as an example because it provides convenient access to open data, the model is designed to work in all cities around the world. The performance of this algorithm is satisfactory: its 81% accuracy, despite being far from the near-100% accuracy archived by algorithms utilizing real-time GPS location, is fairly acceptable in a developing city before advanced systems are installed.

INDEX TERMS: Bus delay, delay prediction, machine learning, neural network, open data, smart cities

基于机器学习算法的公交延误预测模型

马嘉涛　　方书戈　　文元

上海中学国际部　　2020 年 7 月 31 日

The Chinese version of this essay (page 18) is translated from the English version and is for reference only. In case of any discrepancy in the meaning of wording between the Chinese version and the English version, the English version shall prevail.

此论文的中文版本（第 18 页）按照英文版本翻译，并仅供参考。中文版本中的用语的含义如果有与英文版本有出入的，以英文版本为准。

摘要

公交延误预测已经在许多智慧城市中实现，可是目前依靠公交实时定位的电脑算法却难以应用在暂无此类数据的发展中城市。本文将介绍一个仅以时间、天气、公交线路、交通事故报告等现有数据为输入的基于机器学习的公交延误预测模型。虽然这四类输入与公交延误的关系不明显，但是经过处理后，将被转换成共 24 个更加详细、更加相关的参数，如高低峰时段、能见度、降水量等，纳入此模型的核心算法——前馈神经网络。只要使用当地数据训练，模型即可应用在全球各地。本文以多伦多市近六年来的数据为例，训练此模型，并得到高达 81%的准确率。尽管此模型无法与将近 100%准确的实时定位算法比较，却仍是发展中城市在引进新一代技术前最佳的针对公交延误预测的解决方案。

关键词：公交延误，延误预测，机器学习，神经网络，开放数据，智慧城市

Table of Contents / 目录

1 Introduction 5
1.1 Background 5
1.1.1 Literature Review 5
1.1.2 Objectives 6
1.2 Assumptions 6
2 The Model 7
2.1 Input Parameters 7
2.2 Data Preprocessing 7
2.2.1 Date and Time 7
2.2.2 Weather 8
2.2.3 Bus Route and Traffic Incidents 8
2.3 Neural Network 9
2.3.1 Purpose of Neural Network 9
2.3.2 Feed Forward Neural Network 10
2.3.3 Network Architecture 11
2.3.4 Performance of Neural Network 11
3 Model and Analyses 11
3.1 Sensitivity Analysis 11
3.2 Application to Other Cities 12
3.2.1 Difficulties 12
3.2.2 Opportunities 13
4 Model and Evaluations 14
4.1 Strengths 14
4.2 Weaknesses 15
4.3 Potential Modifications 15
5 Conclusion 16
第一章 引言 18
一、背景 18
（一）文献综述 18
（二）研究目标 19
二、研究假设 19
第二章 机器学习模型 20
一、输入参数 20

二、数据预处理.....20
（一）日期及时段.....20
（二）天气.....20
（三）公交线路与交通事故.....21
三、神经网络.....22
（一）神经网络的功能.....22
（二）选择前馈神经网络的理由.....23
（三）神经网络架构.....23
（四）神经网络性能.....24
第三章 模型与分析.....24
一、敏感度分析.....24
二、应用于其他城市.....24
（一）困难.....24
（二）机遇.....25
第四章 模型与评价.....26
一、优势.....26
二、局限.....27
三、改进空间.....28
第五章 结论及展望.....29
References / 参考文献.....30

1 Introduction

The emergence of smart cities fueled by advances in Artificial Intelligence (AI) changed the way people live over the past decade. From optical character recognition to automated driving systems, AI has brought countless benefits, simplifying human tasks or even achieving ones otherwise impossible by humans. One prerequisite of AI, however, is large sets of training data. This problem is hopefully solved in smart cities, where data is collected from the infinitely many sensors belonging to the Internet of Things and utilized to improve the quality of city services. Furthermore, according to the European Commission, a smart city is one with an efficient transport network, an upgraded water supply, a more interactive and responsive city administration, and so on. The focus of this essay will be on the first characteristic—transportation.

1.1 Background

Transportation has been revolutionized in smart cities with the introduction of applications like Google Maps, which provide a variety of services including route planning, real-time traffic information, and more. These services are made possible by the joint efforts of the mapping service provider (e.g. Google Maps) and the local transport department, responsible for the technology and the access to data, respectively. In some developed cities, the two offices are able to collaborate successfully, enhancing their services and bringing added convenience to the public.

One example of a successful implementation is in the city of Hong Kong (HKSAR). Not only does Google Maps return a feasible route when the user enters a starting point and destination, it also considers factors like the bus schedule and potential delays if one chooses to take public transportation. According to Fabrikant (2019) from Google Research, bus delay predictions adopted by Google Maps rely on a sophisticated machine learning model of which details are not disclosed. As a result, an additional literature review is conducted to examine existing pieces of research and methodologies on the implementation of bus delay predictions.

1.1.1 Literature Review

The significance of this topic and the potential benefits it brings urge researchers worldwide to conduct similar explorations.

In terms of algorithms, early researchers like Angelo (1999) and Patnaik (2004) used statistical models—namely, nonlinear time series and multivariable regression. Each of these early algorithms has their own limitations when applied to the prediction of bus delays, as Yin, Zhong, et al. (2006) have summarized: low accuracy, poor performance, or invalid assumptions. Yin and Zhong then suggested that the rapid development of machine learning enables the modeling of bus delays with support vector machines (SVM) and artificial neural networks (ANN). Indeed, these machine learning algorithms are becoming the spotlight of discussion. Sjafjell, Dahl, and Skogen (2014) proposed means to optimize the SVM and ANN models, eventually reaching a conclusion that their implementation of the SVM model would outperform the ANN model with less training, while the opposite holds true given more training.

In terms of input parameters, Patnaik (2004) collected data specifically for his multivariable regression model, including the exact number of passengers boarding or exiting the bus. Recent models featuring machine learning are capable of processing large datasets, so researchers are inspired to use the real-time location of buses as an input parameter, persuading cities to install GPS systems on every vehicle. Sjafjell's SVM and ANN models, for example, analyze the data collected by the GPS trackers during the three previous trips.

1.1.2 Objectives

Utilizing real-time GPS location as an input parameter in a bus delay prediction model, as Google Maps must have done in developed cities like Hong Kong, would no doubt increase the model accuracy. However, an issue surfaces in developing cities, where buses are not equipped with GPS systems. Is it still possible to predict bus delays with machine learning in these cities with data other than the real-time location of buses? In this paper, a model that only takes into account the data at hand[1], data collected by existing nodes in the Internet of Things, will be discussed and evaluated. The objective of this project is therefore to create a generalized bus delay prediction model based on machine learning that can provide fairly reliable bus delay predictions, not just for developed cities, but, more importantly, for developing cities.

1.2 Assumptions

Several basic assumptions made in this model are listed along with the corresponding justifications below.

Assumption 1: The data downloaded from the city's open data portal is accurately and comprehensively collected and posted. The method of data collection has not changed significantly over the past few years.

> **Justification:** The machine learning model is trained over past data. If the integrity of such data is not promised, the learning outcomes cannot be ensured as well. Furthermore, if the method of data collection changes, the same data might reflect different scenarios in reality, hence leading to confusion and inaccuracies.

Assumption 2: The traffic and road conditions in the city has not changed significantly over the past few years.

> **Justification:** Significant changes in traffic and road conditions prompt people to alter their habits, such as switching between public and private means of transportation, detouring from a usual route, or avoiding travel at all—all of which would adversely impact the accuracy of the machine learning model.

Assumption 3: Four main categories of data (see Section 2.1)—namely, date and time, weather, traffic, and bus route—are fed into a neural network (see Section 2.2). These four categories are assumed to play a major role in causing bus delays. In addition, there are assumptions specific to each category, which will be discussed in Section 2.1.

---

[1] The said "data at hand" refers to common data collected by city or government authorities that is not specifically designed to predict bus delays. Examples include weather, demographics, or traffic incident reports. All datasets used in this paper are downloaded from online open data portals hosted by city or government authorities.

Justification: From observations and logical reasoning, these four categories do have correlations, if not causations, with bus delays. Take date and time for instance: roads are known to be congested during the morning and evening peak hours of a weekday. Further justifications will be discussed in Section 2.1.

2 The Model

2.1 Input Parameters

The four categories of input data are listed below in detail. In this paper, real-life data from Toronto, Canada, is used as samples during the training and testing of this model. Raw data is therefore collected from open data portals of the City of Toronto, the Government of Canada, the Toronto Transit Commission (TTC), and the Toronto Police Service. Datasets are downloaded in a variety of formats, so some must be preprocessed before they are of any use as inputs to the model. The "useful inputs" after preprocessing work is also listed in the table below, along with examples of their potential values.

Table 2.1 Input Parameters of the Bus Delay Prediction Model

Raw Input	Processed Input	Sample Values
Bus route number	*No preprocessing required*	1, 6, 927, …
Weather condition	*No preprocessing required*	"Moderate Snow"
Visibility	*No preprocessing required*	10.5 (km)
Precipitation	*No preprocessing required*	11.7 (mm)
Date	Date	"2018/02/13"
	Holiday?	True, False
	Weekend?	True, False
	Season	"Spring", "Summer", …
Time (Hour of the day)	Hour of the day	0, 13, 23, …
	Rush hours?	True, False
Traffic incidents	Traffic incidents	*See Section 2.2.3*
	Distance to incident site*	*See Section 2.2.3*

2.2 Data Preprocessing

To input the data from the Toronto Transit Commission (TTC), the first step is pre-processing. Specifically, all of the data is converted from columns in the text files from TTC's official website to a number in the range from 0 to 1 to reduce the neural network's processing time.

2.2.1 Date and Time

The date can be classified into holiday, weekend, or work day. When inputted into the neural network, each date is represented by a string of three numbers for whether it belongs to each of the above categories. The time of day is inputted as the hour, since specifying the minute or even second will unnecessarily complicate the pre-processing, and does not significantly impact the accuracy of the prediction model. The time is classified into rush hour

and non-rush hour. According to a renowned traffic surveying organization, TomTom, the period from 7 to 9 AM and from 3 to 6 PM will be defined as rush hour. This parameter is represented as a zero for non-rush hour, and a one for rush hour.

2.2.2 Weather

For simplification reasons, all weather data is obtained from Toronto's city center weather station. This is done to reduce the complexity of the preprocessing operations, and since the weather conditions are mostly uniform across the city, using only one weather station for reference will not heavily impact the prediction model.

Weather related factors include temperature, visibility, and specific types of weather. The temperature is measured in Celsius, visibility in meters, and the types of weather are divided into rain, snow, and none of the above. Temperature and visibility are converted into a number between zero and one based on the total range of values in the data, and the weather is expressed in a string of zeros and ones, similar to the date.

2.2.3 Bus Route and Traffic Incidents

The first step to estimating the impact of traffic incidents is to determine the locations of bus stops in each route. The coordinates of all bus stations are provided in longitudes and latitudes in TTC's open data, but the bus numbers and their respective station coordinates are spread over several files, as shown in the image below. Data that is not inputted into the neural network, such as the shape of routes or names of bus routes, are filtered out, and the data used in the training dataset is compiled into one file. This includes bus ID numbers, start and end times of each trip, and the coordinates of each station in longitude and latitude.

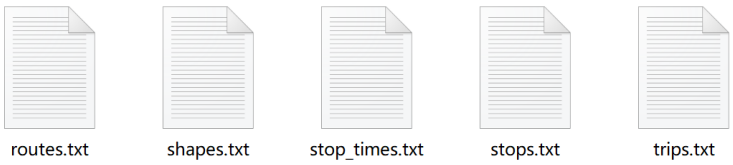


Figure 2.2a TTC Open Data Format

For example, routes.txt contains bus numbers and their respective route IDs. The route IDs are then used to locate the coordinates of each stop in stops.txt. In the final dataset, route IDs are not used, so bus numbers and the coordinates of each stop in the bus route are combined into one file.

The traffic component measures the impact of nearby traffic incidents. To determine whether a traffic incident is "nearby," a virtual box is drawn with the northernmost and southernmost latitudes, and westernmost and easternmost longitudes of all stations in the route. If the incident is within the box, then it is considered near enough to impact bus delays. Conversely, the incident will be considered to have no impact on the bus route if it lies outside the virtual box. As shown in the figure below, the box is drawn with straight lines since when zoomed in to the scope of a city, longitudes and latitudes can be regarded as linear. In addition, traffic incidents usually affect nearby traffic for around two hours, so only incidents that occurred this time range are considered for each bus trip.

Figure 2.2b Calculation of the Distance to a Point of Accident

The distance between a traffic incident and bus route is defined to be the average of distances between the incident and all of the bus stations. To factor in the possibility of multiple incidents qualifying for the above criteria, and the difference between their distances to the bus route, the reciprocals of the distances are added to assign a higher weight to incidents that occurred closer to the bus route, and also to ensure that the sum is a number between zero and one.

2.3 Neural Network

2.3.1 Purpose of Neural Network

The reason a neural network is used to predict bus traffic is due to the neural network's ability to find complex relationships that are not easily expressed through other methods. A simple example of such relationships will be given to illustrate this point.

A	B	A and B
0	0	0
1	0	0
0	1	0
1	1	1

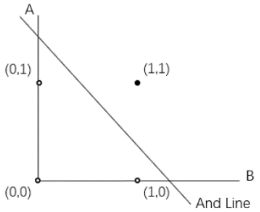


Figure 2.3a "AND" Relationship

A	B	A or B
0	0	0
1	0	1
0	1	1
1	1	1

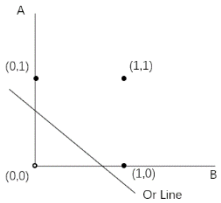


Figure 2.3b "OR" Relationship

The "And" statement is a basic operation in computer science. In the example, "A" and "B" are two Boolean variables. The result of "A and B" is true if and only if both "A" and "B" are true. If all possible values of "A" and "B" are graphed onto a graph, an "And Line" can be drawn that divides the result of "A and B" to the two sides of the lines. The same logic can be applied to the "Or" statement. The result of "A or B" is true if a least one of "A" or "B" is true. This operation can also be expressed using an "Or Line" that divides the results of the operation into two sides. The ease of representing these relationships with lines is significant, in complex situations, these lines can be calculated through regression. However, for more complex operations, this method of division no longer works.

A	B	A XOR B
0	0	0
1	0	1
0	1	1
1	1	0

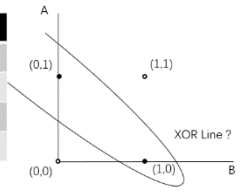


Figure 2.3c "XOR" Relationship

The "XOR" statement is representative of the "Exclusive Or" operation. The result of "An XOR B" is true if and only if A and B are not equal to each other. This type of relationship can easily appear in real life, where two variables can both lead to an outcome when independent but negate each-other out. However, this operation is not easily represented with a line, and attempting to represent the statement using a line often becomes difficult and convoluted. A neural network can identify these relationships through the use of nodes in the hidden layer of a neural network. While this is a simplified representation of the advantage of neural networks, the concept still applies to relationships more complex than the "Exclusive Or" operation, where the number of variables is increased and the relationships between variables become nonlinear.

For the specific topic of bus delay, the determinants of bus delay often have a non-linear relationship with the bus delay. For example, a determinant of bus delay would be whether the day is a weekday or weekend. Another determinant of bus delay could be weather conditions. Bus delay is affected by both weekday and weather conditions not independently but instead in relationship with one another.

2.3.2 Feed Forward Neural Network

The three most popular neural network types are Recursive Neural Networks, Convolutional Neural Networks, and Feed-Forward Neural Networks. Using the correct neural network type is integral to the success of the neural network in predicting bus traffic. Convolutional Neural Networks are specialized for image recognition as they can utilize convolution and pooling to identify special features of the input, thus allowing them to perform better than other neural networks. The spatial aspects of the bus delay are already simplified in the pre-processing stage of the neural network, and many inputs of the neural network do not have spatial features, so the data on bus delay does not have spatial relationships and is therefore not suited for a Convolutional Neural Network.

A Recursive Neural Network's advantage is that it can process the data in a specific time in relation to the data from previous times. It does this by passing the output of previous inputs of the neural network as the inputs for the next batch of inputs. For buses, the time relationship between bus delay is not strong. Having bus delay on one day does not mean that there will be a delay on a subsequent day, and the temporal relationship between consecutive days is often a result of the similar input values on these days and not the result of a delay on the previous day. Another disadvantage of using the Recursive Neural Network on bus delay is that it requires a much higher processing time than other neural network types. This is because there is an intensive calculation that is required to determine how outputs will be passed on to the subsequent input, which greatly increases the time that is required to train the

neural network.

In the end, a Feed-Forward Neural Network is chosen to predict bus delay as it is a general-purpose neural network that can be applied to most scenarios.

2.3.3 Network Architecture

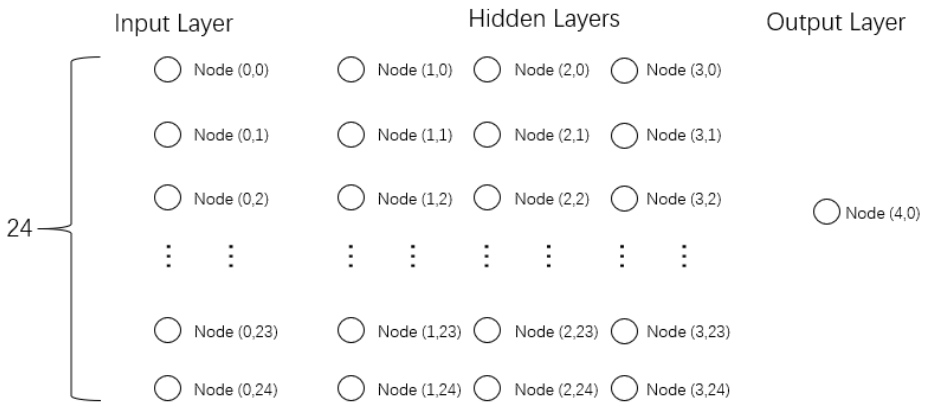


Figure 2.3d Network Architecture

The input layer is of dimensions one by twenty-four. This is because the number of input features is twenty-four. Through experimentation between different hidden layers dimensions, the final dimensions of the hidden layers is three by twenty-four. Other considerations for the hidden layer included increasing the number of hidden layers, decreasing the size of each individual hidden layer, and changing the shape of the hidden layers. However, a three by twenty-four rectangular hidden layer seem to work the best on the data and produced the best results. For all of the hidden layers, a Rectified Linear activation function was used. Finally, the output layer included a singular node utilizing a Sigmoid activation function. Since the output of the neural network is a Boolean value representing whether there will be traffic delay, a Sigmoid function is used to determine the confidence the neural network has in the output, with the Boolean value with the higher confidence being chosen in the end.

2.3.4 Performance of Neural Network

The final result of the neural network is very successful. Through 50 epochs, which translates to less than two hours of training, this model has reached a stable accuracy of 81%, correctly predicting the delay four out of five times. To test the accuracy, several samples are chosen randomly from the bus routes in the year 2020, since only data from before this year is used in the training dataset. The traffic and weather conditions are inputted into the neural network, and across some 200 tests throughout several training cycles, the average accuracy is 81%.

3 Model and Analyses

3.1 Sensitivity Analysis

One of the most time-consuming tasks during preprocessing is the calculation of the distance (and distance reciprocal) from the bus route to a point of traffic accident. The computer

has to deal with a lot of comparisons: date, time, and whether the accident happened within the longitude and latitude constraints, etc. If the factor "traffic incidents" is not considered, then the training and running of the model will become much faster.

As a result, a sensitivity analysis is conducted by removing the factor "traffic incidents" from the model. One input feature is therefore deducted from the 24 input features, now totaling 23 input features into the neural network. The result of several test runs after retraining the network is surprising. There is no significant drop in accuracies. The accuracy rate remains at 81%, like before.

After rounds of additional examination, there is one possible explanation as to why this happens. The most likely reason is that the accidents recorded in the Killed or Seriously Injured (KSI) dataset are very limited. Although the excel file has tens of thousands of records, it is only full because records start from 2006. In reality, when considering the period between 2014-2019, only few accidents are recorded per month. When considering the fact that buses run twenty-four-seven, the event that an accident causes severe bus delay is very rare. As a result, the neural network virtually ignores this dimension. This explains why there is no significant difference when this dimension is not fed into the network altogether.

3.2 Application to Other Cities

Now that the model's performance is verified by data in Toronto, it is reasonable to consider the difficulties and opportunities when applying it to other cities.

3.2.1 Difficulties

Availability of Open Data

Certain cities in developing countries do not store (or provide to the public) data that is considered "generally available" in this paper. For example, traffic incident reports may not be released to the public in cities where privacy is treated with extra caution. Some cities do not have open data portals at all, and access to "open data" requires registration or licensing. In this case, with the lack of historical data, it is harder to train the machine learning model.

Differences in File Formats

Even when there is access to open data, the datasets are often presented very differently in one city and another. These differences range from smaller ones like units—American and British cities use the imperial unit system, while others adopt the metric unit system—to larger differences like the criteria of a "traffic accident." In the Toronto datasets used in this paper, only the "Killed or Seriously Injured" cases are documented, while cities like Chicago record almost all cases, minor or major, with or without injuries. The way how traffic incidents are considered in this model is described in Section 2.2.2. Note that every incident, given that it fits the time and space criteria (i.e. within three hours and within the virtual longitude and latitude constraints), is considered with equal weights in the preprocessing stage. Using the same model for Toronto and Chicago is therefore unreasonable because minor traffic incidents that can be quickly resolved may also be mistakenly considered as a cause of bus delays. As a result, more preprocessing work is needed before this model can be copied and applied to another city as differences in file format may have huge impacts on the model.

Variations across Countries or Regions

Another aspect to consider when applying the model to another city is the variations of environment and people across different countries or regions. Environment variations include climate patterns and the conditions of infrastructure. In a city located at a tropical zone, its residents are most likely used to high levels of precipitation, and high-quality infrastructure like water draining systems is built to ensure safe road conditions during times of heavy rain. On the other hand, in a city that rains only occasionally, precipitation of the same level may be devastating: drivers are less familiar with wet, slippery ground, and roads may be quickly flooded due to poor infrastructure. Speaking of drivers, people's driving habits vary. In a city where traffic rules are strictly followed, there is a smaller chance of traffic congestions and collisions. These variations, though obvious, are hard to measure. A machine learning model trained by data in one city may place high weights on one category of input (i.e. considering weather as the most influential factor of bus delay), whereas people in another city are used to extreme weathers and are able to drive carefully in harsh conditions to avoid accidents and delay.

3.2.2 Opportunities

Despite the many difficulties listed above, this machine learning model, designed to be universally compatible, has its own shines and promising opportunities.

In terms of file formats, although datasets take different forms, the information conveyed is ultimately similar. Take the following example regarding weather data in the city of Toronto and New York, respectively.

Station Name: TORONTO CITY CENTRE			
Date/Time	Max Temp (°C)	Min Temp (°C)	Total Precip (mm)
2014/1/1	-8.2	-13.9	0
2014/1/2	-13.8	-18.8	0
2014/1/3	-6.6	-21.8	0
2014/1/4	0.4	-6.8	0
2014/1/5	1.3	-2.2	10.6
2014/1/6	2.9	-15.6	7
2014/1/7	-14.8	-21.7	0
2014/1/8	-7.7	-15.4	0
2014/1/9	-4	-11.5	0
2014/1/10	2.8	-5	2.8
2014/1/11	5.2	1.9	10.1
2014/1/12	2.7	1.1	0
2014/1/13	4.7	1.1	0
2014/1/14	4.3	0.8	0
2014/1/15	2.2	-3.8	0

Figure 3.2a Weather Data reported by Toronto, Canada

Station: **NY CITY CENTRAL PARK, NY US USW00094728** Generated on 07/25/2020

Year	Month	Day	Temperature (F)			Precipitation				
			24 Hrs. Ending at Observation Time		At Observation	24 Hour Amounts Ending at Observation Time				At Obs. Time
			Max.	Min.		Rain, Melted Snow, Etc. (in)	Flag	Snow, Ice Pellets, Hail (in)	Flag	Snow, Ice Pellets, Hail, Ice on Ground (in)
2014	01	01	33	24		0.00		0.0		0.0
2014	01	02	33	18		0.33		3.1		0.0
2014	01	03	18	9		0.29		3.3		5.9
2014	01	04	29	8		0.00		0.0		5.9
2014	01	05	40	27		0.14		T		3.9
2014	01	06	55	19		0.36		0.0		1.2
2014	01	07	19	4		0.00		0.0		0.0
2014	01	08	22	9		0.00		0.0		0.0
2014	01	09	32	22		0.00		0.0		0.0
2014	01	10	37	30		0.11		T		0.0
2014	01	11	58	37		0.50		0.0		0.0
2014	01	12	54	38		0.05		0.0		0.0
2014	01	13	51	37		0.00		0.0		0.0
2014	01	14	52	44		0.38		0.0		0.0
2014	01	15	47	39		0.00		0.0		0.0

Figure 3.2b Weather Data reported by New York City, USA

At a glance, the two datasets are presented in dramatically different fashions—there are no common fields at all. Date and time, temperature, and precipitation are reported in different formats, different categories, and different units. However, a few lines of code in the preprocessing stage are enough to convert from one format to another. After conversion, data in New York can be fed directly to the machine learning model without triggering any errors.

Responding to the difficulty in Chicago, where minor traffic incidents may also be counted as a major factor of traffic congestions, a simple solution is to filter out these insignificant datapoints before the entire dataset is fed into the model. This step can be done if each datapoint is categorized as "minor" and "major"—which is indeed the case in Chicago. The Chicago Police gives detailed information pertaining to every incident, including the

number of vehicles, pedestrians, and bystanders involved. These pieces of information can be helpful in determining whether one incident has a higher chance than another to cause traffic congestions and hence bus delays.

In terms of variations across countries and regions, this obstacle can be solved by retraining the machine learning model with historical data of the new city under investigation. That way, the model is able to identify the city's unique characteristics (i.e. which factors are highly correlated to traffic congestions) and make appropriate decisions (i.e. which factors should be attributed more weights in the prediction of bus delays). Generalization, flexibility, and universality are perhaps the best terms to describe the qualities of machine learning models, which are not programmed explicitly, but designed to allow computers to learn, to identify patterns, and to make accurate judgements. As such, the promising opportunities and ambitious expectations of this bus delay prediction model based on machine learning and artificial intelligence are well summarized.

4 Model and Evaluations

4.1 Strengths

Comprehensive & Flexible Training Data

As listed in Section 2.1, the model learns from a variety of factors to predict bus delays. This enables more accurate predictions that can be helpful to planning trips.

Although the experiment is based on the traffic in Toronto, the model is flexible enough to be applied to any location with enough traffic and weather data. While Toronto is an example of how the model can be implemented, it can be easily adapted to other cities as well.

High Accuracy for Traffic Prediction

Across several tests, the average prediction accuracy is 81%. Considering how there are many factors that are to some degree random or cannot be quantized, the model can be deemed successful even at this level of accuracy. The aforementioned factors include human-related reasons of delay such as the driver over-sleeping, which may cause unpredictable delays, or some drivers tending to drive faster than others, which may reduce delays but cannot be inputted as a factor into the neural network.

Short Training Time

On average, a run through 50 epochs takes less than two hours, even on a personal laptop. Furthermore, the model can be trained even on a personal laptop, and does not interfere with other processes on the computer. Rather, pre-processing the data in this case takes longer than one training cycle. Overall, since the pre-processing is a one-time computation, the model is portable and can easily be trained on a different device if needed.

Not Dependent on Real Time Data

While many modern algorithms predict bus delays through real time data, such as GPS locations of buses, this model learns only from past data. Real time data may not be easily accessible to the public, so applications of prediction models based on real time data may be

limited to cities willing to provide the data. On the other hand, this model can predict bus delays through learning from readily available open data.

4.2 Weaknesses

More General View of Accuracy

While the model has a high accuracy considering the randomness in predicting traffic delays, in general the 81% accuracy may not be very high either. Therefore, the model might not be very useful under serious circumstances that do not allow room for mistakes. Instead, it serves more as a reference than a prediction to be relied on.

Inconvenience in Pre-processing

As detailed in Sections 2.1 and 2.2, filtering and compiling the data from multiple sources may be time-consuming. This is the downside of a prediction model that learns from past data, as compared to models that predict bus delays using real time data. While the model can be adjusted to accept different parameters, there may be a need to systematically pre-process all of the data to efficiently apply the model to different cities.

Requires Large Volumes of Training Data

In most situations, more training data can increase the accuracy of the model. However, this data is not always available. One of the reasons Toronto was chosen to implement the model was because the open data starts from 2014. In many other cities, traffic data was systematically measured only in the recent few years, which may be insufficient to train and test the model, and may result in low accuracy in the predictions.

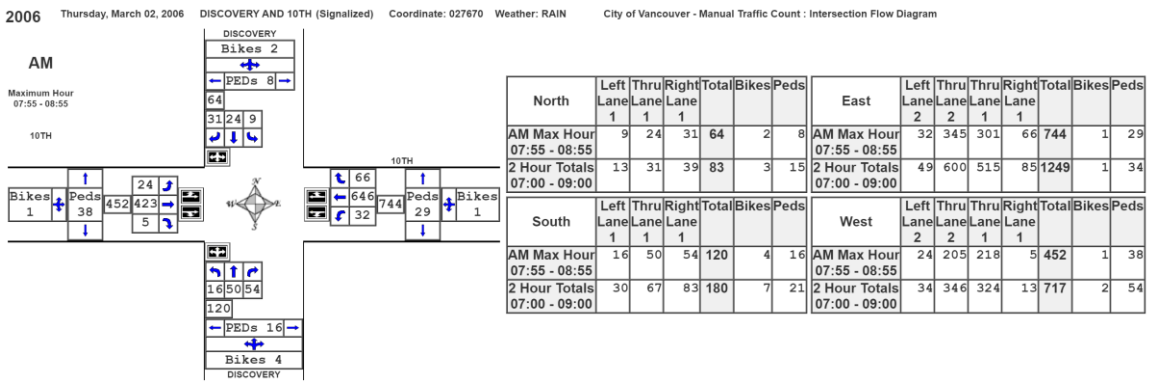
Reliance on Past Trends

One of the assumptions of this model is that traffic and road conditions have not significantly changed over time. This assumption is made so that the model can be trained with past data to predict future delays. However, in fast developing cities that are constructing new roads, traffic conditions may be improving, which can reduce the likelihood of bus delays. When the assumption no longer holds true, the accuracy of the model might decrease since past data cannot reflect future trends. In this case, predictions based on real time data may be more accurate.

4.3 Potential Modifications

Additional Inputs

The four current input categories, namely date and time, weather, bus route, and traffic incident reports, are selected to maximize worldwide compatibility. However, there are certainly additional inputs that can be considered if one wishes to sacrifice a bit of this compatibility. For example, if the scope of application is limited to all Canadian cities, one additional input parameter may be considered: intersection traffic count. Many Canadian cities set up devices like digital cameras and sensors at major road intersections to automatically count the exact number of vehicle or pedestrians passing through the intersection. A snapshot from the City of Vancouver is shown below:



North	Left Lane 1	Thru Lane 1	Right Lane 1	Total	Bikes	Peds
AM Max Hour 07:55 - 08:55	9	24	31	64	2	8
2 Hour Totals 07:00 - 09:00	13	31	39	83	3	15

East	Left Lane 2	Thru Lane 2	Thru Lane 1	Right Lane 1	Total	Bikes	Peds
AM Max Hour 07:55 - 08:55	32	345	301	66	744	1	29
2 Hour Totals 07:00 - 09:00	49	600	515	85	1249	1	34

South	Left Lane 1	Thru Lane 1	Right Lane 1	Total	Bikes	Peds
AM Max Hour 07:55 - 08:55	16	50	54	120	4	16
2 Hour Totals 07:00 - 09:00	30	67	83	180	7	21

West	Left Lane 2	Thru Lane 2	Thru Lane 1	Right Lane 1	Total	Bikes	Peds
AM Max Hour 07:55 - 08:55	24	205	218	5	452	1	38
2 Hour Totals 07:00 - 09:00	34	346	324	13	717	2	54

Figure 4.3 City of Vancouver Intersection Flow Diagram

The level of detail is surprising in this diagram: the number of vehicle and pedestrians passing through the intersection is reported along with a time interval and the direction of travel. These details, if considered as additional inputs to the model, may reveal the exact time and direction of traffic congestions, thus improving the model accuracy.

Neural Network Modifications

Changing the neural network specifics might potentially improve the accuracy of this machine learning model. There are many small tweaks to make, for example, the number of hidden layers, the dimensions of each hidden layer, the activation function, etc. Optimization of a neural network takes time and skill, so it is left for future researchers.

More Efficient Preprocessing Programs

Since the datasets used by this model are downloaded from online open data portals, files come in various formats. As a result, data preprocessing takes a very long time, perhaps even longer than the time needed to run the neural network. One area of improvement is to modify the preprocessing programs into more efficient ones. For example, when determining whether a given date is a holiday or not, instead of establishing an input/output gateway to a file on the hard drive containing all past and future holidays, reading and comparing each line, a smarter program would load the file into a variable (i.e. into memory) first, before making the appropriate search and comparisons. As such, there are other areas to improve in terms of program efficiency.

5 Conclusion

Despite the trend toward developing models that predict delays based on real-time data such as GPS locations, this model has achieved considerable success through learning from only past data. By incorporating both preprocessing algorithms and machine learning, the neural network has reached a stable accuracy of 81%.

To further enhance the model, there are still improvements that can be made to increase both the training speed and accuracy. As described in Section 4.3, modifying the model to accept more inputs can likely improve the accuracy of the model. Along with this, modifications can also be made to the neural network to allow more efficient and accurate learning and predictions, and writing more efficient preprocessing programs can further contribute to reducing the overall training time of the model.

With a more developed neural network, this model may be applied to cities that measure traffic and weather data but do not support the technology to provide real-time data. In these cities, local governments and concerned individuals can use this model to predict bus delays using only past and current open data, aiding them in planning more efficient trips.

第一章 引言

一、背景

近年来人工智能的研究与应用推动了一座座智慧城市的发展，彻底改变了人们的生活方式。从光学字符识别（OCR）到自动驾驶系统，人工智能带给人们的便利不计其数。人工智能不仅可以替人们完成一些繁琐的计算工作，还可以实现一些人们之前可望却不可及的任务。但是，使用人工智能的一个关键条件是要有足够多的数据，令电脑可以从中训练、学习、找规律。这个问题有幸在智慧城市中得以解决，因为大量来自物联网中的数据被采集、分析及利用于改善城市的服务和运作。此外，根据欧盟的资料，一座智慧城市拥有便利的交通运输网络、健全稳定的水供应、贴近民生的行政机关等等。此论文的焦点将会放在智慧城市的第一个特点——便利的交通。

智慧城市中的交通相比传统城市中的交通有着翻天覆地的变化。一些包括谷歌地图、百度地图在内的软件向用户们提供了前所未有的服务，包括路线规划、实时路况等等。这些功能是地图服务提供商（谷歌地图或百度地图）和当地交通部门合作所得到的成果——前者负责技术支持，而后者负责提供数据。在一些发达城市中，这两个部门可以紧密地合作，改善服务，为人们的交通出行带来更多的便利。

一个成功的合作案例在香港特别行政区。当用户在谷歌地图等应用程序中输入了一个始发地和目的地，并选择公共交通时，程序不仅会计算出一个合适的交通路径，还会考虑到公交的首末班车、运行时刻及可能发生的延误。根据谷歌研究部门的学者 Fabrikant (2019)，谷歌地图所采用的公交延误预测模型基于一个复杂的机器学习模型，其细节并不向外公布。因此，为了更好了解与此话题相关的现有研究，以下是围绕公交延误预测模型的文献综述。

（一）文献综述

公交延误这个话题十分具有意义，一旦研究出一个准确可靠的模型，将有助推动改革城市交通，为智慧城市的发展做出贡献。如此大的收获促使全球各地的学者都对此话题做出相关的研究及探索。

论算法，早期的研究者 Angelo (1999) 和 Patnaik (2004) 分别采用了非线性时间序列分析和多元线性回归等统计模型。应用在公交延误预测中，这两个统计模型都有各自的局限性，如准确率低，性能差，研究假设不成立等等。Yin, Zhong, et al. (2006) 提出，在机器学习算法高速发展和普及的时代，支持向量机(SVM)和神经网络(ANN)已逐渐成为此话题中大势所趋的算法。的确，这两种机器学习算法用作实现公交延误预测模型确实成为了近年来学者们研究及讨论的焦点。Sjafjell, Dahl, and Skogen (2014) 优化了 SVM 和 ANN 模型，最终得出一个结论：在他们的研究中，若训练数据较少，则 SVM 胜过 ANN 模型；若训练数据较多，则 ANN 胜过 SVM 模型。

论输入参数，Patnaik (2004) 为了其多元线性回归模型，专门采集了多项数据，包括特定时间内公交车上车和下车的具体人数。近年来基于机器学习的模型可以处理庞大的数据组，因此学者们开始倾向使用公交的实时数据，并呼吁各城市在其公交车上加装全球定位系统 (GPS)。公交实时定位成为了新一代模型的核心输入参数: Sjafjell 的 SVM 和 ANN 模型就以一条公交线路前三次运行时所记录的 GPS 定位为输入，推算出本次运行时延误的概率。

（二）研究目标

谷歌地图在香港等发达城市之所以可以做出准确的公交延误预测，或许就是利用公交车上的实时 GPS 定位作为输入参数。但是，问题出现在发展中的城市——当地的公交车并不会搭载 GPS 系统。在这种情况下，是否还可以利用除了实时定位以外的其他数据，推断出公交的延误概率呢？这篇论文将会探讨一个只用现有数据[2]为输入的机器学习模型。本项目的目标便是构建一个通用的机器学习模型用于预测公交延误，并为所有城市——不论是发达城市或发展中城市——提供一个相对准确、可靠的公交延误预测。

二、研究假设

此模型基于三个主要假设。

假设一：开放数据平台上的数据都是准确且完整的；采集这些数据的方法在过去几年中没有明显的变化。

> **理由：**此机器学习模型的训练数据组是过去几年的真实数据。若无法保证历史数据的准确度和完整度，则模型的训练、学习过程将受到影响，进而减低模型最终的准确度。此外，若采集数据的方法有所改变，看似相同的数据在过去的几年中可能具有不用的含义，因此造成混淆和误差。

假设二：城市的交通和道路状况在过去几年中没有明显的变化。

> **理由：**明显的交通和道路状况变化将促使人们改变自己的生活习惯，例如选择其他交通工具出行，使用其他道路出行，或错峰出行。人们习惯的变化将影响此机器学习模型的准确度。

假设三：此模型的输入参数（具体见第二章）分为四大类别：日期和时间、天气、交通和公交线路。这四个类别的数据对公交延误有一定程度的影响。

> **理由：**根据观察和逻辑思考，这四个类别中的参数可以被认定为是造成普通公交车延误的主要原因。即使这四个类别中的参数和公交延误不是因果关系，它们之间所存在的关联也是不容否定的。因此，这四大类适合作为机器学习的输入参数。

---

[2] 现有数据指的是由当地行政部门所采集的普通开放数据，而并不是为了公交延误预测模型专门采集的数据。现有数据包括天气、人口信息、交通事故调查等等。此研究所用的所有数据组均来自相关城市行政部门的开放数据平台。

第二章 机器学习模型

一、输入参数

此模型的输入参数列举在表 1 中。这篇论文将使用一组真实的数据来运行、训练，及测试模型，以方便之后的分析与评价。基于对数据完整性和查阅权限的考虑，这组数据将会取自加拿大多伦多市。本文内所涉及的数据组均从多伦多市政府、加拿大政府、多伦多运输局（TTC）、多伦多警察局等单位的开放数据平台下载；下载格式不一，部分参数因此需要经过预处理才可输入模型。预处理后的可用输入参数也列举在表 1 中，并随附一个例值可供参考。

表 2.1 公交延误预测模型的输入参数

输入参数（处理前）	输入参数（处理后）	例值
公交线路	*无需预处理*	1, 6, 927, …
天气状况描述	*无需预处理*	“中雪”
能见度	*无需预处理*	10.5 (km)
降水量及降雪量	*无需预处理*	11.7 (mm)
日期	日期	“2018/02/13”
	假期?	True，False
	周末?	True，False
	季节	“春”，“夏”，…
时段（24 小时制）	时段	0，13，23
	早晚高峰时段？	True，False
交通事故	交通事故	*见第二章*
	距交通事故的距离*	*见第二章*

二、数据预处理

建立 AI 模型的第一步是预处理多伦多运输局（TTC）提供的官方开放交通数据。所有输入的数据将会被标准化成零到一之间的数字，从而减少神经网络运算的时间。

（一）日期及时段

日期会分为假期、周末和工作日。每个日期会被转化为三个数字，通过一与零代表次日期是否符合上述的三种分类。具体时间则会分类为高峰和非高峰时段。根据 TomTom 的统计，早高峰和晚高峰分别被定义为上午七到九时和下午三到六时。高峰时段将会被一代表，而非高峰时段会被零代表。

（二）天气

为了简化预处理天气数据的过程，所有数据都会以多伦多城区气象局为准。在减少汇编数据的复杂度的同时，多伦多不同地区的天气总体差异较小，所以位于市中心的气象局数据能够代表多伦多总体的天气状况。

天气相关因素包括温度、可视度和天气情况。根据多伦多城区气象局提供的天气数据，温度与可视度会分别以摄氏度与米为单位。输入到神经网络时，这两项参数会根据总体数据范围被转化成零到一之间的数字。天气情况包括降雨、降雪。类似于日期，这项参数会被转化为一串零和一的数字来代表是否属于各类天气情况。

（三）公交线路与交通事故

预处理交通事故数据的第一步是得出每个公交线的站点位置。多伦多运输局提供的开放数据包含了所有公交站点的经度和纬度，但是每个公交线和对应的站点被分布在很多个文件里（如下图）。预处理交通相关的数据时，不被用在训练神经网络的数据，比如路线形状和公交线的名字，会被删除。训练用到的数据则被汇编到同一个文件里。这包含了公交线路号码、公交车抵达各个站点的时间和每个站点的经纬度坐标。

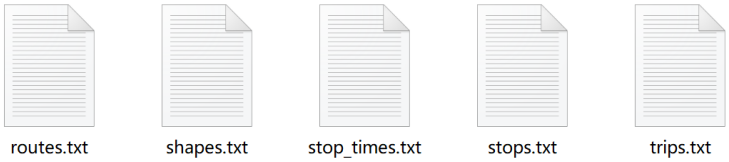


图 2.2a 多伦多运输局开放数据格式

比如，在查找公交线路号码和对应的站点坐标会首先用到 routes.txt 里的号码和 ID 码。ID 码会用来在 stops.txt 里查找站点坐标，从而得出每个公交线路里站点的位置。由于 ID 码不包含在训练数据里，处理后线路号码和对应的坐标会被合并到同一个文件里。

交通事故部分会计算公交路线附近发生的事故对延误的影响。一个事故是否在路线的“附近”会通过所有站点的坐标确定。如下图所示，黄色的长方形是用所有站点最南和北面的纬度和最东和西面的经度组成的。如果该事故位于长方形内，会被视为“附近”的事故。相反，不在此框内的事故被视为不会对此公交路线的延误造成影响。虽然经纬线是弧形的，但是放大到城市的范围后可以被简化为直线。因此，长方形是由直线组成的。此外，交通事故造成影响的持续时间多数小于两小时，所以每个路线的预处理只会考虑两小时内发生的交通事故。

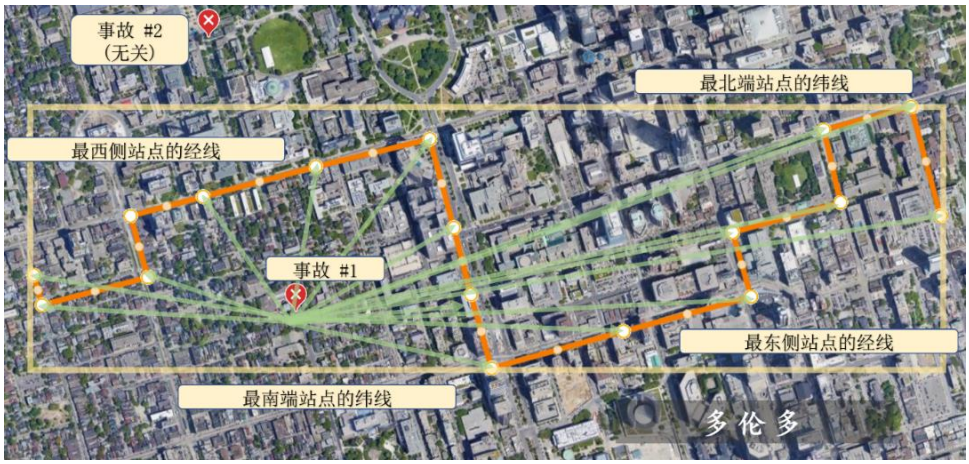


图 2.2b 计算交通事故离公交车路线的距离

交通事故离公交车路线的距离被定义为事故与所有站点之间距离的平均值。在有超过一个交通事故的情况下，所有事故造成的影响总和被定义为所有距离倒数的和。此计算方式会考虑到距离近的事故会造成更高的影响，同时也能够确保最终的数值会在零和一之间。

三、神经网络

（一）神经网络的功能

使用神经网络来预测公交车流量的原因是由于神经网络能够找到复杂的关系，而这些关系很难通过其他方法来表达。

A	B	A and B
0	0	0
1	0	0
0	1	0
1	1	1

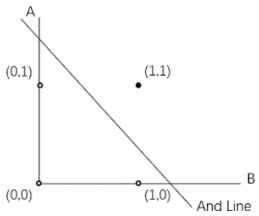


图2.3a “AND”逻辑

A	B	A or B
0	0	0
1	0	1
0	1	1
1	1	1

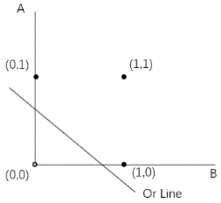


图2.3b “OR”逻辑

“与”语句是计算机科学中的逻辑。在图中，“A”和“B”是两个布尔值变量。当A”和“B”都等于1，“A与B”的结果为1。如果将“A”和“B”的所有可能值都绘制到图上，则可以绘制一条线，将“A与B”的结果划分到线的两侧。可以将相同的逻辑应用于“或”逻辑。如果“A”或“B”中的至少一个为1，“A或B”的结果为1。也可以使用一条线表示该逻辑，将结果分为两边。用线表示这些关系很容易，在复杂情况下，可以通过求导来计算这些线。但是，对于更复杂的操作，这种划分方法不再起作用。

A	B	A XOR B
0	0	0
1	0	1
0	1	1
1	1	0

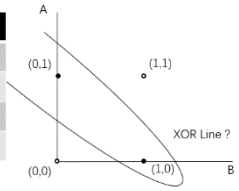


图2.3c “XOR”逻辑

“XOR”代表“异或”逻辑。当A和B彼此不相等时，“A异或B”的结果为1。这种关系很容易出现在现实生活中，其中两个变量在相互独立时都可以导致结果实现，但是同时出现时会彼此抵消。此逻辑很难用一行来表示，并且尝试使用一条线来表示常常会很困难和费解。神经网络可以通过使用神经网络的隐藏层来识别这些关系。尽管这是神经网络优势的简化表示，但该概念仍适用于比“异或”运算更复杂的关系，比如变量数量增加或变量之间的关系变为非线性。

对于公交车延迟的特定主题，公家车延迟的决定因素通常与公家车延迟具有非线性关系。例如，公交车延迟的一个决定因素是一天是工作日还是周末。公交车延误的

另一个决定因素可能是天气状况。公交车的延误不仅受工作日和天气状况的个别影响，而是两个因数的结合体。

（二）选择前馈神经网络的理由

三种最受欢迎的神经网络类型是递归神经网络，卷积神经网络和前馈神经网络。使用正确的神经网络类型对神经网络成功预测公交流量至关重要。卷积神经网络专用于图像识别，因为它可以利用卷积和池化来识别输入的特殊功能，从而使它比其他神经网络性能更好。公交车延迟的空间方面已经在神经网络的预处理阶段得到了简化，并且神经网络的许多输入不具有空间特征，因此总线延迟上的数据不具有空间关系，因此不适合用于卷积神经网络。

递归神经网络的优势在于，它可以相对于以前的数据在特定时间内处理数据。它通过传递神经网络的输出作为下一批输入的输入来实现。公交车延迟和前一天的延迟的时间关系不强。一天有公交车延误并不意味着第二天会有延误，连续几天之间的延误通常是由于这些天的输入值相似而不是前一天的延误所致。 在公交车延迟上使用递归神经网络的另一个缺点是，与其他神经网络类型相比，它需要更长的处理时间。这是因为需要进行大量计算才能确定输出将如何传递到后续输入，这大大增加了训练神经网络所需的时间。

最后，选择了前馈神经网络来预测总线延迟，因为它是可以应用于大多数情况的通用神经网络。

（三）神经网络架构

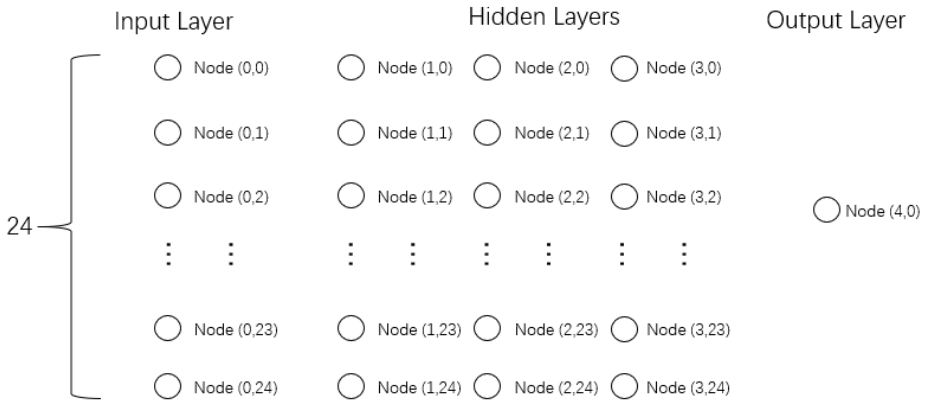


图2.3d 神经网络架构

输入层的尺寸为一乘二十四。这是因为输入要素的数量是二十四个。通过在不同的隐藏层尺寸之间进行实验后，决定让隐藏层的最终尺寸为三乘二十四。隐藏层的其他考虑因素包括增加隐藏层的数量，减小每个单独隐藏层的大小，以及更改隐藏层的形状。但是，三乘二十四的矩形隐藏层似乎在数据上效果最好，并且产生了最佳结果。对于所有隐藏层，都使用了整流线性激活函数。最后，输出层利用了“Sigmoid“的激活函数。由于神经网络的输出是一个布尔值，表示是否将成为流量延迟，因此使用

Sigmoid 函数确定神经网络对输出的置信度，最后选择具有较高置信度的布尔值。

（四）神经网络性能

最后，神经网络训练的结果非常成功。通过 50 期的训练，此神经网络在两个小时内就可以稳定的达到 81%的准确率。为了测量出该准确率，2020 年的巴士路线中被随机选出了很多测试样例，因为训练数据里只包含 2020 年前的数据。这些样例被输入到神经网络中，而在数次训练以及大约 200 次的测试后，最终得出的平均准确率是 81%。

第三章 模型与分析

一、敏感度分析

由于此模型涉及了庞大的数据组，数据预处理的耗时很长。其中，最复杂的一项便是计算公交线路至交通事故发生点的距离。要计算出这个值，电脑需要考虑日期、时间、及事故发生点是否位于指定经纬区域内等等。若省略“交通事故”这个因素，则可大大缩减模型的训练和运算时间。

本模型共计 24 个输入参数。在接下来的灵敏度分析中，将会把“交通事故”这一项从输入参数中移除，剩 23 个输入参数。当模型被重新训练后，几次的运行结果都令人惊讶：模型的准确率并没有显著下降，依然维持在 81%左右。

经过几轮额外分析后，得出以下结论。之所以模型的准确率没有改变，是因为 KSI 数据组中记录的导致死亡及重伤的事故非常有限。尽管文件里看似有成千上万的记录，但由于本文件是从 2006 年开始记录，每年更新，日积月累，记录便显得多了。实际上，若只考虑 2014-2019 这六年来，每月发生的事故屈指可数。考虑到公交车全天运作，一个月里偶尔发生的车祸、事故对市内公交运行时刻基本没有影响。即使偶尔有影响，神经网络也会因发生的频率太少而忽视这项输入。这项输入本身对公交延误预测的帮助就不大，因此，当这项输入被移除后，对模型也并没有太大的影响。

二、应用于其他城市

该模型的性能已通过多伦多的数据进行了验证。若要将其应用于其他城市时，需考虑一下的困难与机遇。

（一）困难

数据的采集与公开

某些发展中国家并没有开放数据的概念，也因此不会采集或公开那些“开放”、“现有”的数据供此模型作训练和学习。比方说，交通事故报告在某些城市里可能被视为当事人的隐私，所以并不公开。在这种情况下，想要获取当局内部的文件，就必须要经过申请、审批等繁琐的程序，为此模型的运行带来一定的阻碍与困难。

不同的文件格式

即使访问开放数据不受权限问题的影响，从不用城市、不同机关、不同部门下载

的数据组往往都以不同的文件格式呈现。格式的差异可大可小，有些可以被一次性地转换成理想的格式，有些则会严重地影响机器学习模型的判断。一些小问题包括计数单位的不同——英美两国使用英制单位，而其他地方则使用公制单位。一些大问题则是对关键词定义的不用。比方说，在多伦多的数据组中，只有“致死或严重身体伤害(KSI)”的交通事故才被列作“事故”而记录下来。然而，在芝加哥等城市，所有交通事故——不论大或小、有无人员受伤——都会被列作“事故”记录存档。两座城市对“交通事故”的定义不同，若盲目地将这两组看似相同的数据输入进机器学习模型，模型的错误率将显著提高。这是因为，只要一起交通事故符合时间和空间上的标准（即发生在三小时内及地理位置于经纬度限制内，详见第二章），它就会被此机器学习算法以同样大的权重纳入模型。这个算法应用在多伦多的数据组里没有问题，但在芝加哥，一起没有伤亡、可以在几分钟内解决的小事故，可能会被机器学习模型认定为导致公交延误的主要原因。所以，若要将此模型应用于其他城市，必须先确保其输入文件的格式、表述方法和关键词定义一致。

国家与国家间的差异

应用此模型到其他城市的另一个难点在于国家（地区）与国家（地区）之间的种种差异。其一是环境的差异，包括一座城市的气候和基础建设。在一座位于热带地区的城市中，人们或许已经习惯了频繁的强降水。该城市有关排水的基建或许已经非常成熟、完善了；所以即使在强降水的天气中，道路状况依然可以得到保障，车辆拥堵的情况也不会发生。相反，在一座偶尔下雨的城市中，相同程度的降水可能造成不可想象的后果：欠佳的排水系统导致路面积水，而车辆驾驶员也不习惯湿滑的路面。驾驶员素质就是国家与国家之间其二的差异。若当地的驾驶员都可以严格遵守交通规则，则出现道路拥堵的几率就更小。环境与人的差异虽然非常明显，但却难以测量、难以数字化、难以换做输入参数供机器学习模型训练和学习。因此，以一座城市的数据组做训练的机器学习模型可能将某一个特定的输入加以巨大的权重，可这一个输入参数未必在另外一座城市里对公交延误有着如此大的影响。

（二）机遇

尽管将此模型应用于其他城市的过程中难免会面临以上三个问题，此模型仍具有良好的通用性，且其独一无二的亮点更是保证了其受到广泛应用的机遇。

论文件格式，虽然数据组的格式往往不会一致，但它们的内容终究是大同小异。用下图中分别取自多伦多和纽约市的气象数据为例：

Station Name: TORONTO CITY CENTRE

Date/Time	Max Temp (°C)	Min Temp (°C)	Total Precip (mm)
2014/1/1	-8.2	-13.9	0
2014/1/2	-13.8	-18.8	0
2014/1/3	-6.6	-21.8	0
2014/1/4	0.4	-6.8	0
2014/1/5	1.3	-2.2	10.6
2014/1/6	2.9	-15.6	7
2014/1/7	-14.8	-21.7	0
2014/1/8	-7.7	-15.4	0
2014/1/9	-4	-11.5	0
2014/1/10	2.8	-5	2.8
2014/1/11	5.2	1.9	10.1
2014/1/12	2.7	1.1	0
2014/1/13	4.7	1.1	0
2014/1/14	4.3	0.8	0
2014/1/15	2.2	-3.8	0

图3.2a 加拿大多伦多市的气象数据

Station: **NY CITY CENTRAL PARK, NY US USW00094728** Generated on 07/25/2020

Year	Month	Day	Temperature (F)			Precipitation				
			24 Hrs. Ending at Observation Time		At Observation	24 Hour Amounts Ending at Observation Time				At Obs. Time
			Max.	Min.		Rain, Melted Snow, Etc. (in)	Flag	Snow, Ice Pellets, Hail (in)	Flag	Snow, Ice Pellets, Hail, Ice on Ground (in)
2014	01	01	33	24		0.00		0.0		0.0
2014	01	02	33	18		0.33		3.1		0.0
2014	01	03	18	9		0.29		3.3		5.9
2014	01	04	29	8		0.00		0.0		5.9
2014	01	05	40	27		0.14		T		3.9
2014	01	06	55	19		0.36		0.0		1.2
2014	01	07	19	4		0.00		0.0		0.0
2014	01	08	22	9		0.00		0.0		0.0
2014	01	09	32	22		0.00		0.0		0.0
2014	01	10	37	30		0.11		T		0.0
2014	01	11	58	37		0.50		0.0		0.0
2014	01	12	54	38		0.05		0.0		0.0
2014	01	13	51	37		0.00		0.0		0.0
2014	01	14	52	44		0.38		0.0		0.0
2014	01	15	47	39		0.00		0.0		0.0

图3.2b 美国纽约市的气象数据

两组气象数据看似截然不同。日期和时间、温度、降水量等数据都以不同的格式、不同的分类、不同的单位记录。可是，其实只要短短几行的预处理程序就可以完成格式转换的工作。预处理完成后，纽约市的气象数据便可以直接输入到机器学习模型中，且不会出现任何错误。

上文提到的芝加哥问题其实也不难解决。在芝加哥，所有交通事故，不论严重与否，都会被记录下来。要防止那些微不足道的事故被机器学习模型误认为会导致道路拥挤的最佳办法，便是将其在数据预处理阶段删除。要做到这一点，必须分清楚哪些交通事故属于“轻微”，哪些属于“严重”。这个分类工作确实可行，因为芝加哥警方的交通事故报告中都会记录详细的事故描述，包括涉事的车辆、行人、第三者等等。这些信息都有助于衡量某个交通事故的严重程度，并由此可以推算出此事故导致道路拥挤、公交延误的概率。

论国家与国家之间的差异，这个问题无法直接解决。可是，只要利用一套新的数据组就可以重新训练机器学习模型，令此模型可以发现新数据组中的特点。通用性，灵活性，和兼容性或许就是此模型最突出的特点——它并不是由一行行代码所合成的软件，永远只会依程序处理数据；相反，此模型通过训练数据组，令电脑学习、找规律、继而做出相当准确的判断。综上，此基于机器学习和人工智能的公交延误预测模型在国际应用这个舞台上具有极大的机遇、希望和未来。

第四章 模型与评价

一、优势

全面和可变的输入数据

从第二章可以看出，此模型会通过数个方面的因素学习和预测未来的延误。多方面的数据能够增加预测的准确率。

虽然实验里用到了多伦多的交通数据，但此模型也可以用其他城市提供的数据训练和预测延误。

相对高的准确率

多次测试后，此模型预测的准确率平均有 81%。对公交车延误造成影响的因素有很多无法被量化或有一定的随机性。这包括了许多与人相关的因素，比如司机没有准时发车造成无法预测的延误，或是有些司机开车速度较快，可能会减少延误的可能性但无法被输入进神经网络。综上所述，此模型能够达到 81%的准确率是很成功的。

训练速度快

以 50 期训练为标准，笔记本电脑也可以稳定的在两个小时内完成神经网络的训练。训练过程要求的计算量也不高，能够在普通的电脑上运行且不会影响电脑的其他进程。相对来说，预处理会有更大的运算量。因为预处理只需要运行一遍，所以总体的训练过程较短，重新在其他设备上训练此模型也很方便。

不依赖于实时数据

现代多数算法是根据实时数据做出预测的，比如巴士上的 GPS 定位。这些实时数据很多城市不会开放的提供或不做测量，所以利用实时数据的模型使用仅限于能够获取实时数据的城市。反过来说，此模型的训练和预测只会用到相对更容易获取的过去交通数据，使用范围则更加广泛。

二、局限

精准度有限

虽然在交通情况有一定随机性的情况下此模型的准确率不低，但 81%总体来看也不是非常高。如果情况不允许任何差错，此模型可能逊于用实时数据的算法。因此，此模型可以提供参考价值，但不能完全被依赖于预测未来的延迟。

繁琐的预处理过程

就如第二章所述，筛选和汇编多方面的数据需要的时间和计算量都很多。对数据的需求能够提高准确率，但同时也需要更多的预处理准备。并且，不同城市发布的数据类型和格式之间会有区别，所以每个城市都有可能需要重新系统性的完成一次预处理才可以训练模型。

对数据的需求量大

大多时候，更大的数据量可以增加预测的准确率。不过，不是所有城市都提供或者测量交通数据。选择多伦多为实验对象的原因之一就是可以获取从 2014 年开始的所有交通以及天气数据，有足够的数据充分的训练和测试模型。在许多近期发展的城市，只有近几年的交通数据开始被系统性的测量。这可能导致预测的准确率下降、限制了可以应用此模型的城市。

根据过去的趋势预测未来的延误

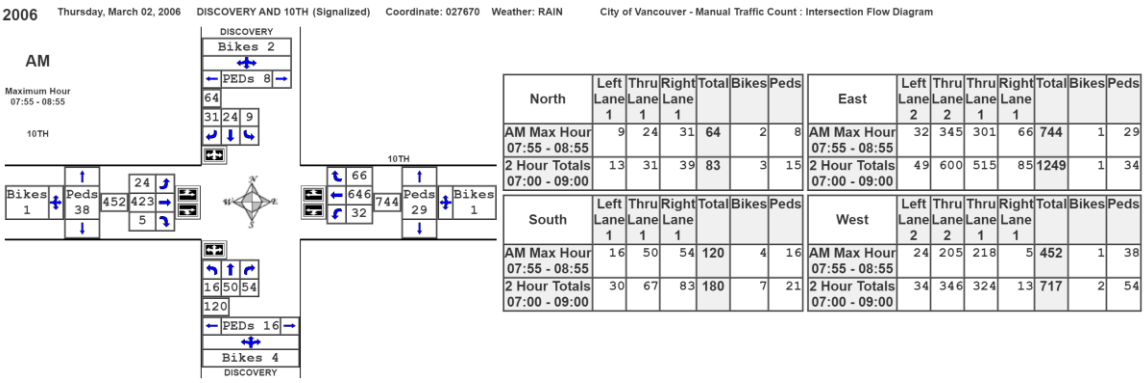
在第一章里描述的第二个假设是城市的交通和道路状况在过去几年中没有明显的变化。如此假设是因为此模型需要通过以前数据的趋势预测未来的延误。在一个快速发展的城市，道路的改良同时也会改善交通路况、减少堵车以及降低延误的可能性。如果上述的假设不成立，过去的趋势将无法反映出未来的路况。此模型的准确率会因

此降低。这类快速发展或近期改造的城市里，利用实时数据预测延误的算法可能更准确。

三、改进空间

（一）其他输入参数

当前的四大输入类别，即日期和时间，天气，公交路线和交通事故报告，是为了以最大程度提高此模型的全球兼容性。但是，若牺牲一点兼容性，便可考虑一系列其他的输入参数。例如，如果应用范围限于所有加拿大城市，则可以考虑一个加拿大城市常常记录的参数：路口流量。加拿大许多城市在主要道路交叉路口设置了诸如数码相机和传感器之类的设备，以自动计算通过交叉路口的车辆或行人的数量。下图便是一张温哥华市某个路口的流量统计表。



North	Left Lane 1	Thru Lane 1	Right Lane 1	Total	Bikes	Peds
AM Max Hour 07:55 - 08:55	9	24	31	64	2	8
2 Hour Totals 07:00 - 09:00	13	31	39	83	3	15

East	Left Lane 2	Thru Lane 2	Thru Lane 1	Right Lane 1	Total	Bikes	Peds
AM Max Hour 07:55 - 08:55	32	345	301	66	744	1	29
2 Hour Totals 07:00 - 09:00	49	600	515	85	1249	1	34

South	Left Lane 1	Thru Lane 1	Right Lane 1	Total	Bikes	Peds
AM Max Hour 07:55 - 08:55	16	50	54	120	4	16
2 Hour Totals 07:00 - 09:00	30	67	83	180	7	21

West	Left Lane 2	Thru Lane 2	Thru Lane 1	Right Lane 1	Total	Bikes	Peds
AM Max Hour 07:55 - 08:55	24	205	218	5	452	1	38
2 Hour Totals 07:00 - 09:00	34	346	324	13	717	2	54

图 4.3 温哥华路口流量图

此图的详细程度令人惊讶：通过交叉路口的车辆、行人，都根据时间段和行进方向分类。如果将这些的详细信息作为此机器学习模型的输入，则可协助模型更准确地判断交通拥堵的确切时间和方向，提高模型的准确性。

（二）神经网络微调

对神经网络的微调可能可以提高此机器学习模型的准确率。具体需要进行许多小的调整，例如，隐藏层的数量，每个隐藏层的大小，激活函数等等。神经网络的优化需要时间和技巧，因此留给其他学者研究优化。

（三）改进预处理程序

由于此模型使用的数据集是从线上开放数据平台下载的，因此文件的格式五花八门。数据预处理需要很长时间，甚至可能比运行神经网络所需的时间还要长。要缩短数据预处理的时间，就必须找到当前程序的漏洞，并对其进行优化。例如，若要判断给定日期是否为本地文件中指定的假期时，不应逐行读取储存在硬盘上的文件，然后再进行比较判断。正确的做法应现读取整个文件，并将其载入内存。这样将大大减少重复访问硬盘所需的时间。像这样的问题还有很多，需逐项优化，以提升数据预处理程序的时效性。

第五章 结论及展望

尽管现代延误预测模型有着向实时数据的发展趋势，此模型只依赖于过去的数据但同样也很成功。通过数个预处理程序与机器学习算法，此模型的神经网络已经能够稳定的达到 81%的准确率。

此模型还有一定的发展空间。通过改变神经网络的输入参数与其他变量可以持续提高此模型的准确率。同时，效率更高的预处理程序也能有效的减少总体训练模型所需的时间。

在完善神经网络后，此模型可以应用到测量交通与天气数据的城市，尤其是暂时未能测量实时数据的城市。有需求的人员可使用此模型来通过以前收集的数据预测未来的巴士延误，从而更好的计划出行安排。

References / 参考文献

Alpaydın, E. (2010). *Introduction to Machine Learning.* Cambridge: The MIT Press.

Chollet, F. (2018). *Deep Learning with Python.* Shelter Island: Manning.

Fabrikant, A. (2019, June 27). *Predicting Bus Delays with Machine Learning*. Retrieved from Google AI Blog: https://ai.googleblog.com/2019/06/predicting-bus-delays-with-machine.html

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 436-444.

Smart cities. (n.d.). Retrieved from European Commission: https://ec.europa.eu/info/eu-regional-and-urban-development/topics/cities-and-urban-development/city-initiatives/smart-cities_en

Toronto Traffic. (2020). Retrieved from TomTom: https://www.tomtom.com/en_gb/traffic-index/toronto-traffic/

Zanella, A., Bui, N., Castellani, A., Vangelista, L., & Zorzi, M. (2014, February). Internet of Things for Smart Cities. *IEEE INTERNET OF THINGS JOURNAL*, 22-32.