Web-enabled Smart City Applications for Urban Transport and Parking Operations

PRAMATH JOSHI, MINDULA LIYANAGE DON, SIDNEY GUILDFORD CHIKODI, AUSTIN BITANGA, MOURITA CHOWDHURY, YIXIN MU, VIVIENNE FARRELL

Swinburne University of Technology, Hawthorn VIC 3122, Australia

RICHARD B. WATSON

Ryan Watson Consulting Pty Ltd, Vermont VIC 3133, Australia

PETER J. RYAN

Defence Science & Technology Group, Fishermans Bend VIC 3207, Australia

This paper describes the development of a prototype website for traffic, parking and transport in a smart city. Machine Learning (ML) tools are applied to open datasets from the City of Melbourne, Australia to develop a set of Application Programming Interfaces (APIs) that provide useful information for the city's managers and citizens. The APIs accessed from this website enable users to query the ML models and obtain answers to questions such as: which parts of the city have the greatest pedestrian traffic, or the availability and cost of parking spots. The freeware tool RStudio was used for Big Data analytics while Machine Learning with Plumber was used to wrap the R code into APIs and Swagger to specify and document them. Postman and Swagger were used for testing while Docker was employed to package the APIs into standard containers for cloud deployment. The prototype website was developed using Wix and deployed on the Nectar cloud. The resulting website provides predictive models for COM traffic, parking, and transport and demonstrates the application of online smart city services for city planners and managers.

CCS CONCEPTS •Computing methodologies • Machine learning • Machine learning algorithms

Additional Keywords: Smart Cities, Big Data Analytics, Machine Learning, APIs, RStudio, Plumber, Swagger, Docker, Wix, Nectar

1 INTRODUCTION

Cities worldwide are attempting to transform themselves into so-called 'Smart Cities', which are composed of and monitored by pervasive information and communication technology including systems connected by the Internet of Things (IoT). Such cities generate vast amounts of data, known as Big Data, from their IoT-connected sensors. Big Data analytics and machine learning (ML) are required to refine, analyse, interpret, and extrapolate this data to assist city residents, as well as government and industry in benefitting from these new capabilities.

Many cities make their data accessible as open data for developers who can apply ML to develop Application Programming Interfaces (APIs) that can be accessed over the Internet by city managers and citizens. This paper describes the development of a prototype website for predictive modelling of transport and parking in the City of Melbourne (COM), Australia using data from the COM's open data portal. The investigation was conducted as a research project for final year software engineering students at Swinburne University of Technology.

2 REVIEW OF PREVIOUS WORK

Personal mobility and transport of people, goods, and services are critical for efficient operation of a potential Smart City and thus transport is one of the key issues for a Smart City [1]. This includes pedestrian and vehicle movement as well as parking requirements both on street and off street. These are respectively known as smart pedestrian, smart traffic or transport, and smart parking.

Future transport options for Smart Cities are discussed by Mualla et al. [2] and Nikitas et al. [1]. These involve Connected and Autonomous Vehicles (CAVs), autonomous Personal and Unmanned Aerial Vehicles (PAVs and UAVs) and Mobility-as-a-Service and also the IoT.

Smart parking systems are needed to reduce congestion and pollution in Smart Cities. Al-Turjman and Malekloo surveyed smart parking systems in Smart Cities [3] and discussed a cloud-based hybrid concept for parking. IoT based systems are described by Mudaliar et al. [4] and Gopal et al. [5] that use sensors, Arduino™ microcontrollers, and cloud servers. Many smart parking APIs have been deployed worldwide. Australian examples include the Car Park API developed by the New South Wales (NSW) Department of Transport [6] and a similar one developed by the Australian Capital Territory (ACT) Government [7]. However these do not use ML models.

Zheng et al. studied pedestrian traffic in major cities [8]. They modified the Highway Capacity Model to include the pedestrian mode including pedestrian delay, vehicle interactions, and jaywalking. Their model can be applied to evaluate and select optimal pedestrian routes. Galanis et al. studied the walkability of urban streets and developed an index based on traffic flow and walking behavior [9]. Feliciani et al. developed a simulation model for street crossings to assess pedestrian fatalities and compared their model with real data from Winnipeg (Canada) and Melbourne (Australia). [10]. Recently Akhter et al. [11] reported on an IoT intelligent sensor node system for pedestrian counting while Vitello et al. [12] described a pedestrian mobility model that could be used for smart city simulations.

3 CITY OF MELBOURNE DATASETS

Melbourne is the capital of the southeastern state of Victoria in Australia with its greater urban area having a population approaching 5 million. The COM, that represents the Central Business District (CBD), has created an Open Data Portal [13] that contains datasets classed as either related to (1) parking, (2) safe mobility, (3) city sensors, (4) land use and employment, (5) 3D data, (6) major developments, (7) greening laneways, (8) environment, (9) city statistics and forecasts, (11) rooftop project, and (12) transport.

Of particular interest for the current project are the datasets related to parking, safe mobility and city sensors. These were used as sources of data for analysis and development of predictive ML models.

3.1 Vehicle Classification Dataset

The COM Traffic Count Vehicle Classification database covers numbers and types of vehicles on Melbourne roads over the period 2014 – 2017. The number of vehicles and the number of each category was recorded each hour for this period. The resulting dataset has 60, 000 rows; 28 columns and is about 7.5 MB in size and is available at: https://data.melbourne.vic.gov.au/Transport/Traffic-Count-Vehicle-Classification-2014-2017/qksr-hqee

3.2 On-street Parking Dataset

The COM uses in-ground parking bay sensors in most CBD on-street parking bays that record when a vehicle arrives and departs. Each record also includes the parking restriction for the bay and whether the vehicle has overstayed that restriction. These data were collected in 2019. An event will include if a vehicle was present or not, which can determine stay time and vacancy time. The resulting dataset has 42.7 million rows and 20 columns and is over 7 GB in size and is available at: https://data.melbourne.vic.gov.au/Transport/On-street-Car-Parking-Sensor-Data-2019/7pgd-bdf2

3.3 Pedestrian Counting Dataset

The COM collects data on pedestrian movement within the CBD using a set of IR laser sensors. These detect movement rather than images so no personal information is collected. The dataset contains all sensor readings since the system was initiated in 2009. It has 3.3 million rows each of 10 columns and is roughly 270 MB in size and is available at: https://data.melbourne.vic.gov.au/Transport/Pedestrian-Counting-System-2009-to-Present-counts-/b2ak-trbp

These three datasets are summarized in Table 1.

Table 1: COM datasets used for machine learning

Name	Rows x Columns	Size
Traffic Count Vehicle Classification (2014 - 2017)	60170 x 28	7.5 MB
On Street Car Parking Sensor Data 2019	42.7M x 20	>7 GB
Pedestrian Counting System	3.3M x 20	270 MB

4 CITY OF MELBOURNE DATA ANALYSIS AND MACHINE LEARNING MODELS

Initially Machine Learning as a Service (MLaaS) options such as IBM Watson™ Studio and Microsoft Azure were trialled and some preliminary analysis was performed. Traffic flow is shown in Figure 1 applying MS Azure to the Traffic count classification dataset. However, while these MLaaS systems are highly capable systems that enable data manipulation, pre-processing and visualization, they did not meet the requirement of availability as open source freeware to match the funding of the student project. The free versions soon used up their resource quotas and payment was requested to continue [14]. Further the visualizations are 'canned' and cannot readily be customized.

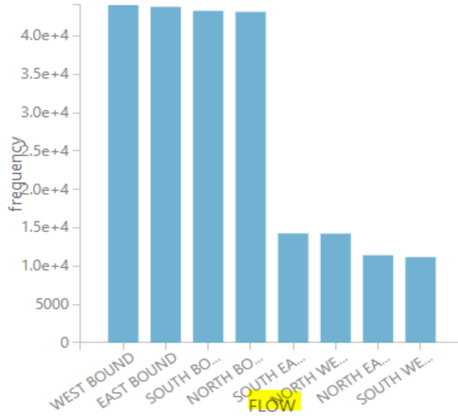

Figure 1: MS Azure Studio analysis showing traffic flow direction from the traffic analysis dataset

Instead the open-source system RStudio [15] that uses the R programming language was investigated and found to be quite suitable for the project. RStudio Desktop was thus deployed to analyse the datasets, split them into testing and training subsets, then create ML models and develop APIs for the various applications. A web API is an interface for software applications analogous to a graphical user interface used by humans [16]. The datasets were generally split 80:20 into training and testing datasets as is common practice in ML [17, 18]. This process is shown in Figure 2.

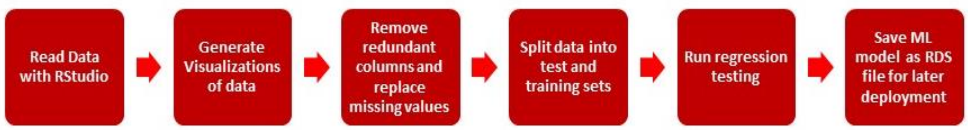

Figure 2: Flowchart for data analysis and machine learning

The Plumber R package [19] was used to wrap the ML R code into runnable APIs and the Swagger open source tool [20] then used to specify and document them. The Hypertext Transfer Protocol (HTTP) client Postman [21] together with Swagger-generated client libraries were used for API testing. The software package Docker was used to package the APIs into standardized containers for development, shipment and cloud deployment [22]. Docker avoids issues with programming languages and Web Server Gateway Interfaces. This process is shown schematically in Figure 3.

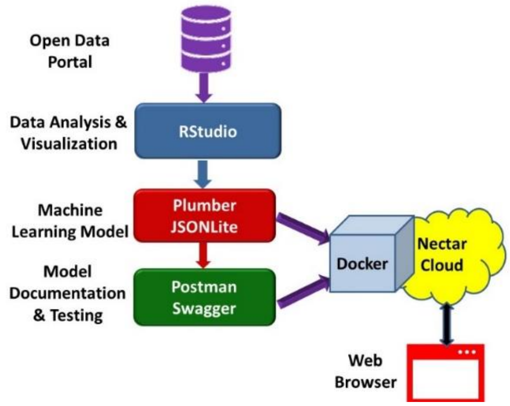

Figure 3: Data analysis, ML development, and API deployment process

5 PROTOTYPE WEBSITE

The aim of the project was to deploy the developed APIs on a prototype website. The website front end was developed using the free and code-free website builder Wix together with customized javascript, Cascading Style Sheets (CSS) and HTML code [23]. Wix provides ready-made templates and also has its own editor (Wix Editor) to create a site from scratch.

The website back end was developed using Wix functionality with some additional javascript coding. Issues with web interface design such as text input for dates rather than numeric were investigated and software solutions obtained where possible. The home page for the prototype web system focused is shown in Figure 4. This enables users to access predictive models for traffic, transport, and parking, based on ML models developed from open COM datasets over the Internet.

Figure 4: Home page of prototype Smart City website developed by Swinburne University student project team

The APIs accessed from this website enable users to query the ML models and obtain answers to questions such as:

- What parts of the city have the greatest pedestrian traffic?
- How many parking spots are likely to be available for specified periods and at what cost?
- What sorts of vehicle and in what numbers are using particular roads?

The traffic interface is shown in Figure 5. This shows that for the given conditions, there were predicted to be 13 bicycles travelling in that suburb.

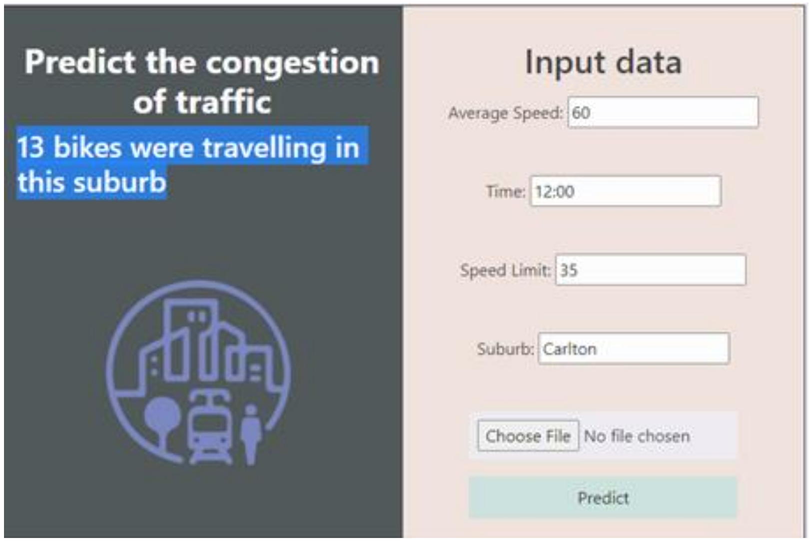

Figure 5: The Traffic Interface for the Smart City website

6 RESULTS

The following sections describe preliminary analyses of the three datasets.

6.1 Analysis and ML for Traffic Count Classification Dataset

The Traffic Count Vehicle Classification dataset was analyzed by RStudio. For ease of analysis, this dataset was refined and some columns were removed that have maximum null values or unnecessary data. The transformed columns that were used for the analysis include suburb, speed limit, time, average speed, maximum speed, and bike and are shown in Table 2.

Table 2: Columns used for analysis of Vehicle Classification Count dataset, description, and their data types

Column Name	Description	Data Type
Suburb	Suburb where the road is located	String
Speed limit	Speed limit of the surveyed road	Number
Time	60 minute period when data was captured	String
Bike	Number of bicycles recorded	Number
Average speed	Average speed of vehicles crossing the sensor	Number
Maximum speed	Maximum speed travelled over the sensor	String

Here 'Bike' has been chosen as the target variable. Hence, the final output resulting when the details of suburb, speed limit, time, average speed, and maximum speed are entered is the number of bikes moving in that particular suburb at a particular time of the day with a specific speed. The data was split into a training set (80%) and a testing set (20%). The ML algorithm used was Multiple Linear Regression with the Bike column being the dependent variable and the other 5 columns the independent variables. Linear regression was applied since the target is a numerical value, namely the number of bicycles.

To test the API, the libraries – “plumber” and “jsonlite” are used and then it is run on the local host = "127.0.0.1" with port = “8080”. Sample output in the Swagger user Interface (UI) is shown below in Figure 6 for set conditions of maximum speed, average speed, time speed limit and suburb indicating that there were roughly 11 bicycles that matched these conditions.

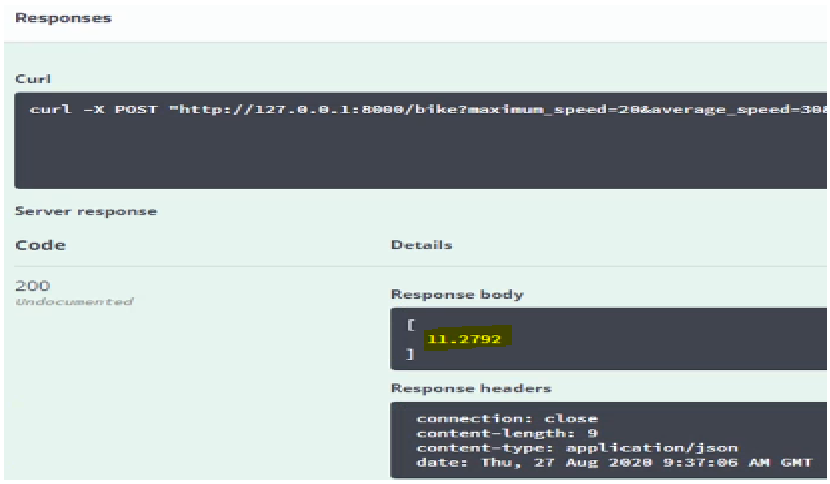

Figure 6: Response to query about numbers of bicycles that met certain conditions in Swagger User Interface

6.2 Analysis and ML for On-street Parking Dataset

The dataset ‘On-street Car Parking Sensor Data - 2019’ was retrieved from the COM Open Data portal and reduced from 42.7 million rows to 20,000 rows to enable initial analysis. Some preliminary analysis was performed using IBM Watson Studio to determine feature importance of the 20 columns. The key feature for this dataset is the 'inviolation count' that states how many parking events were violated, that is broke the restrictions. From the COM site it can be shown that about 1.5 million of the 42.7 million parking events were in violation of the restrictions.

Together with data splitting using Python this enabled the dataset to be reduced to the seven transformed columns of interest shown below in Table 3.

Table 3: Columns used for analysis of On Street Parking dataset and their data types

Column Name	Data Type
*InViolation	String
*AreaName	Integer
DurationMinutes	Integer
*Month (e.g. May)	Integer
Day (0-31)	Integer
*Period (AM or PM)	Integer
*Time (e.g. 9:01)	String

Further data visualization, manipulation and pre-processing were performed using R, Python and Microsoft Excel™. Within RStudio, a ML model was created employing the Logistic Regression method on the target variable 'InViolation'. The dataset was split into a training set (80%) and a testing set (20%). Logistic Regression was employed here since the target variable has only two logical values True or False. The ML model was deployed onto Docker and the API's were tested with Swagger and Postman.

Testing the model for two random events where no violation and violation occurred predicted probabilities of 1.99e-9 and 0.4391. Thus for the first event, non-violation was predicted correctly to be effectively zero while there was a 43.9% chance for the second event to be a parking violation. Further testing showed that values of greater than 0.01 indicated a parking violation. Choosing eight random events where there was a parking violation yielded an average of 0.257 whereas for eight random non-violation events an average of less than 0.0001 was determined. This indicated that any value greater than 0.01 will generally designate a violation.

6.3 Analysis and ML for Pedestrian Counting Dataset

The dataset Pedestrian Counting System has 20 columns and over 3 million rows. This was transformed to a 5-column dataset removing variables that were deemed to be redundant for the present purpose as shown in Table 4.

Table 4: Columns used for analysis of Pedestrian Counting dataset and their data types

Column Name	Data Type
Month	String
Mdate	Integer
Year	Integer
Time	Integer
Hourly_Count	Integer

As for the previous two datasets, this dataset was split into 80% training and 20% testing. With the dependent variable, an integer value Hourly-Count, a multiple linear regression ML model was created and then a Plumber API produced that has the four independent variables (year, month, day of month and time of day) as endpoints. The model was tested using the Swagger UI generated automatically by RStudio. Comparison of the model with observations showed excellent agreement.

7 CONCLUSIONS

A prototype Smart City application has been developed that enables users to make predictions about traffic, parking, and pedestrian transport characteristics. Big Data analytics and Machine Learning were applied to City of Melbourne open datasets for traffic count vehicle classification, parking (both on-street and off-street in parking facilities), and pedestrian travel, leading to predictive models that were deployed over the Internet through APIs. Analyses for all three datasets were discussed describing the processes involved and some sample output was demonstrated.

Little work of this nature has been undertaken in Australia, hence it has the potential to be of great benefit to Australian local government councils and members of the public who must deal with them. This project is, however, a work in progress and many steps have still to be undertaken before the APIs reach the production stage, including access controls and logging capability. It is planned that the lessons learned will be documented in a follow-on paper. Future work will also include development of ML models for other datasets such as the

COM microclimate environmental system (air quality) and the proposed COM smart waste system that could be included in the prototype.

ACKNOWLEDGEMENTS

The authors wish to acknowledge Mr Will McIntosh from the Smart City Office, City of Melbourne and Dr Suneel Jethani, DataVic, Department of Premier and Cabinet (Victoria), for useful discussions about the City of Melbourne and Victorian Open Data Portals. They also wish to thank the Department of Computer Science and Software Engineering at Swinburne University of Technology for supporting the project.

REFERENCES

[1] Nikitas, A., Michalakopoulou, K., Njoya, E. T. and Karampatzakis, D. Artificial intelligence, transport and the smart city: Definitions and dimensions of a new mobility era. *Sustainability*, 12, 7 (2020), 2789.

[2] Mualla, Y., Najjar, A., Galland, S., Nicolle, C., Haman Tchappi, I., Yasar, A.-U.-H. and Främling, K. *Between the megalopolis and the deep blue sky: Challenges of transport with UAVs in future smart cities.* City, 2019.

[3] Al-Turjman, F. and Malekloo, A. Smart parking in IoT-enabled cities: A survey. *Sustainable Cities and Society*, 49 (2019/08/01/ 2019), 101608.

[4] Mudaliar, S., Agali, S., Mudhol, S. and Jambotkar, C. IoT based smart car parking system. *Int J Sci Adv Res Technol*, 5, 1 (2019), 270-272.

[5] Gopal, D. G., Jerlin, M. A. and Abirami, M. A smart parking system using IoT. *World Review of Entrepreneurship, Management and Sustainable Development*, 15, 3 (2019), 335-345.

[6] NSW Department of Transport *Car Park API.* City.

[7] ACT Government Open Data Portal dataACT *SmartParking API.* City.

[8] Zheng, Y., Elefteriadou, L., Chase, T., Schroeder, B. and Sisiopiku, V. Pedestrian Traffic Operations in Urban Networks. *Transportation Research Procedia*, 15 (2016/01/01/ 2016), 137-149.

[9] Galanis, A., Botzoris, G. and Eliou, N. Pedestrian road safety in relation to urban road type and traffic flow. *Transportation Research Procedia*, 24 (2017/01/01/ 2017), 220-227.

[10] Feliciani, C., Gorrini, A., Crociani, L., Vizzari, G., Nishinari, K. and Bandini, S. Calibration and validation of a simulation model for predicting pedestrian fatalities at unsignalized crosswalks by means of statistical traffic data. *Journal of Traffic and Transportation Engineering (English Edition)*, 7, 1 (2019/07/27/ 2019), 1-18.

[11] Akhter, F., Khadivizand, S., Siddiquei, H. R., Alahi, M. E. E. and Mukhopadhyay, S. IoT enabled intelligent sensor node for smart city: pedestrian counting and ambient monitoring. *Sensors*, 19, 15 (2019), 3374.

[12] Vitello, P., Capponi, A., Fiandrino, C., Giaccone, P., Kliazovich, D. and Bouvry, P. High-precision design of pedestrian mobility for smart city simulators. In *Proceedings of the 2018 IEEE International Conference on Communications (ICC)* (2018). IEEE, [insert City of Publication],[insert 2018 of Publication].

[13] City of Melbourne *City of Melbourne Open Data.* City, 2020.

[14] Cross, B. *Comparing Machine Learning as a Service: Amazon, Microsoft Azure, Google Cloud AI, IBM Watson.* City, 2018.

[15] Rstudio *Open Source & Professional Software for Data Science Teams 2020.* City, 2020.

[16] Rutqvist, D., Kleyko, D. and Blomstedt, F. An Automated Machine Learning Approach for Smart Waste Management Systems. *IEEE Transactions on Industrial Informatics*, 16, 1 (2019), 1-9.

[17] Detective, T. D. *The 80/20 Split Intuition and an Alternative Split Method.* City, 2020.

[18] Gour, R. *Train and Test Set in Python Machine Learning — How to Split.* City, 2020.

[19] Plumber. *Plumber: An API Generator for R.* City.

[20] Swagger *Swagger: API Development for Everyone.* City.

[21] Postman *Postman, The Collaboration Platform for API Development.* City.

[22] Docker *What is a Container? A standardized unit of software.* City, 2020.

[23] Wix *Wix - Create Your Website.* City.