Pin Pointing Pain Points: Vehicular traffic flow intensity detection and prediction through mobile data usage.

Maurice Saliba

Supervisor: Dr. Charlie Abela

Co-supervisor: Dr. Colin Layfield

Faculty of ICT

University of Malta

28-06-2018

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Artificial Intelligence

Dataset Anonymization

Declaration

I, the undersigned, declare that the dataset of mobile CDRs provided by GO p.l.c., which cover the period between August 2016 and September 2017, have been anonymized at source.

Maurice Saliba

28-06-2018

Acknowledgements

I would like to thank my supervisor Dr. Charlie Abela and Co-supervisor Dr. Colin Layfield who taught me amongst many other things to let the data speak for itself. Discussing with them the various gradual achievements and caveats made it possible to foster insight.

I would like to thank GO p.l.c for supporting this dissertation by providing me with the anonymized mobile usage dataset and allowing me to use the high-end computing resources needed for the highly intensive processing required by this dissertation. I would also like to thank Dr. Ing. Joseph C. Attard for encouraging me to pursue this Masters course. I also acknowledge the assistance of my colleagues at GO who were of great support and inspiration, especially when I discussed with them certain technical aspects of the project.

On a personal note, I would like to thank my wife Andrea for her great support, love and constant encouragement. I would like to thank also my parents, Benny and Carmen, and the rest of my family especially my nephew Kyle and niece Shirley who proved to be a useful forced distraction when needed. I would like to express my gratitude towards my close friends who did their best to drive me forward towards completion by constantly sending me pictures of them having beers on a great night out without me.

Last but not least I would like to thank all the academic and non-academic staff of the department of artificial intelligence. It was a great academic journey that enriched my experience and perspective towards the world of science.

Abstract

A novel approach, consisting of an ensemble of data-mining and machine learning techniques, is proposed to prove that it is possible to extract and predict vehicular traffic patterns from mobile usage data. An anonymized mobile phone usage dataset from a telecommunications provider in Malta was used to generate an origin-destination (OD) matrix that defines the top two activity hubs through clustering. The OD matrix was used to infer user trips over fastest routes between these top two locations across time. We then applied spatial binning techniques to deduce the aggregate distribution of traffic load on the traffic network. A predictive model based on an artificial neural network was trained with the whole network traffic flow load in a time series to predict traffic level for specific nodes.

Daily trip distribution showed to have a very strong correlation ($r = 0.94$, $p < 1.1e - 11$) with those reported in the National Household Travel Survey (NHTS). Similarly a significantly strong linear relationship ($r = 0.69$, $p < 0.001$) was found when comparing mean hourly route trip delays with mean trip delay estimation recorded with Google APIs. To evaluate traffic flow count method, we compared our results with manual counts retrieved from a 2016 study by Nigel Pace. Strong instances of correlation ($r = 0.75$,$p < 0.05$) for low congested traffic points were observed. These contrasted with weak negative correlation ($r = 0.45$,$p < 0.05$) for traffic flow in locations were traffic congestion frequently occurs. Traffic flow prediction through an ANN proved to be efficient with F1-Scores ranging from 0.58 to 0.9 for different road segments in experimentation.

The proposed solution needs improvement by adding a dynamic traffic assignment to the whole algorithm. This would give more accurate results, especially for traffic flow points that tend to be congested, by capturing user route selection changes and get more precise localization of delay causes.

Contents

List of Figures

List of Tables

1. Introduction

1.1 Economic development and urbanization impact on transport

Land transport is a societal reality that is essential for people to get to work, leisure and other places for other purposes. Transport is essential for delivering goods and services. Land transportation has undoubtedly evolved at a fast pace and late technology advancements are making vehicular transportation more efficient, less polluting, faster, safer and more comfortable. There are many land transport modes which include bus, rail and private car as the most generally used. In Malta 60% of commuters opt for the car as the preferred mode of transport. This is stated in the modal split report in the National Transport Household Survey 2010 report document [1].

Economic development and urbanization come at a cost. It surely has a direct impact on the increase of traffic congestion and all undesirable consequences it brings with it. Traffic congestion is especially synonymous with urban places where the private car is the preferred mode of travelling. The EU Transport Directorate (2018) mentions how traffic congestion in urban areas in the EU is costing 100 billion Euro every year which amounts to 1% of the EU GDP [2]. Colak et al. elaborate on the crippling effect on the economy because of traffic congestion [3].

Traffic congestion amounted to 43% (€117.9 million) of external costs in Malta in 2012, which is the origin of the mobile traces datasource used in this study [4]. Other causes of external costs related to traffic include accidents, climate change, air pollution and noise which are all directly incremented by traffic congestion. No policy change scenario envisages an external cost of €151.1 million and €154.1 million for the years 2020 and 2030 respectively incurred on the economy of Malta.

In the US, traffic congestion is similarly a cause of concern. Interesting but worrying facts are listed in a US mobility research done in 2015 [5]. It states that the extra miles travelled by Americans in 2014 were 6.9 million at the cost of $160 billion. Congestion costs in the USA is on the increase. In the year 2000 it was reported to be at the level of $114 billion.

Traffic delays have a heavy impact on the shipment industry as well. Travel costs increase when travel time increases. Pick-up and delivery time estimation become less accurate because of traffic congestion. Transport companies need to take costly measures in order to make up for this and the increase in cost is more often than not passed to the consumer [5, 6].

1.2 Addressing Traffic congestion

Both car users and public transport users tend to get frustrated from unnecessary delays on the road. This time is deducted from a healthy lifestyle or from productivity hours.

Drivers can adapt to smartly mitigate delay times. Individual drivers can hear radio adverts or check CCTV to inform themselves about the traffic situation before departing or while driving for better planning. Use of software such as Google Maps [1], Apple Maps [2] or Waze [3] help to have an informed decision how to schedule trips and decide what route to take. These applications might even suggest other

[1] https://www.google.com/maps (accessed April 3, 2018)
[2] https://apple.com/ios/maps (accessed April 3, 2018)
[3] https://waze.com/en-GB (accessed April 3, 2018)

transport modes that offer faster or more convenient alternatives to get to the same destination.

Efficient traffic management should be at the top of national transportation agencies' agenda. Possible measures that can be taken by transport authorities include making different modes of transport more widely available and encourage the public to use it. Smart technology is another means to alleviate travelling frustration by giving information, instructions and control traffic flow in an automated manner. For more uptake of public transport the public for instance can be informed and educated through mobile applications. Mobile applications can be used to make the public transport experience more efficient, practical and the preferred choice. There are other deterrents such as increases in vehicle license tax and adding of parking fees to force drivers off the road and make them use public transport or cleaner ways of transportation such as cycling.

Law enforcement is another way to facilitate traffic flow. This would diminish road accidents or casual road blockages that can cause flow disruptions. Automatic number plate recognition (ANPR), through camera feeds processing, can be used to measure traffic flow and even to apply a toll to users in certain traffic zones as a deterrent for private car use. Park and ride systems shift away concentration of traffic from urban centres [4]. Nuaimi et al. for instance show how concerted efforts can lead to smarter cities by analysing static data and make infrastructural changes by opening or modifying roads. In this study dynamic data was used to manage traffic lights to alleviate congestion, inform the public through their smart phones about the traffic status and control logistics related to movement of goods [7].

Investment in the transport infrastructure to expand capacity is difficult to directly justify with a simple cost benefit analysis model [8]. Increase in road capacity might seem a simple straightforward solution to alleviate traffic. However infrastructure alterations might not necessarily equate to easing of traffic. Such costly changes might just spatially shift the problem elsewhere or fail to lead to

the expected result. Forecasting of the gains made by road capacity increase or any other transport system changes may be distorted if induced traffic is not taken in consideration. Induced traffic may result from changes in route choice, peak hour traffic, modal split, overall transport volume, land use and quality of public transport services [9]. When formulating return on investment functions induced traffic should not be ignored.

1.3 Traffic information and management systems

Traffic management systems primarily monitor traffic status in the road network and take traffic control decisions, such as, increasing or decreasing lanes in a tidal lane system based on traffic data. The traffic data, on which traffic management decision logic is based on, must be updated frequently and it should be reliable. Intelligent traffic management systems are more efficient when the traffic control decisions are based on real-time streams of traffic data. Processing real-time feeds is challenging both in terms of computational resources and design but is more reactive to abnormal situations such as accidents or unusual weather conditions since it is modelled on a running sample [10]. Traffic related data stream processing might entail heavy real time processing of high variety data coming from multiple sources. Modern approaches, such as big data based information systems, become essential in order to create automated control systems that alleviate the load on the transport network [11].

Traffic Information Systems (TIS) can tap into mobile usage records as a main source of information. Such TIS leverage mobile data collection that has wide coverage, is reliable, accurate and is frequently updated [12]. Less coverage is to be expected in rural areas where base stations are highly dispersed when compared to urban areas. Mobile vehicle geolocation has limited spatial resolution. For example it cannot be used for traffic flow counts on lanes, whereas it could be easily done with inductive loops.

1.4 Traveller centric traffic flow probing

Obviously, the dynamics of traffic flow is determined by the travel needs of the masses. The daily commutes of every individual impacts those of others. The interaction on a large scale of all the vehicles in a time series is difficult to model and to predict in a robust and responsive manner [13]. Traffic sensors, cameras and induction loops are all sources of information that can be used to both detect high traffic intensity or even forecast it beforehand. However, the coverage these techniques offer is limited. Camera feeds and inductive-loop detectors cannot be installed in every road of the transport infrastructure. Devices carried by travellers, or embedded in vehicle, can be possibly used to build smart solutions for traffic management [13].

1.4.1 Passive vs Active data collection

Long before the information era started, spatio-temporal data on human mobility was collected in various forms and modalities. One of the methods used to gather such information is to do straightforward surveys[14, 3]. However these are expensive methods because a lot of manual work needs to be carried out. Besides they could only give a snapshot of reality at a given point in time. Generally, these types of surveys are done every five to ten years [10]. The data made available would be too static and increasing the frequency of survey taking would directly require more human resources assigned to the process. As mobile telecommunications and gradual adoption of its services came into the picture, at the turn of the millennium, more data points could be collected in an automated fashion. A limitation, which comes with mobile user related information, is the lack of demographic knowledge on the mobile owner. Surveys gather such information and make stratified sampling possible in order to have a more representative sample [3].

1.4.2 Application of mobile traces analytics

Mobile traces can be processed and used to offer location based services that have a wide application spectrum that go beyond solving mobility issues [15, 14, 16, 17]. An individual's location and its relation with that of others within the context of the continuum of time is invaluable in many ways. This formidable datasource, however, poses a challenge. Location data, which usually comes in large amounts, has to be harvested, ingested efficiently and ideally processed in real time for the required final purpose which is value added location based services.

The range of application and branches of research abound on remote collection of mobile users' geolocation information. To name a few applications include: traffic patterns and prediction modelling, crowd management, hotspot detection, lost device recovery, emergency rescue, use for investigative authorities, location-based recommendation and advertising systems, contextualized information, social interaction based application, epidemiology etc.

Calabrese et al. emphasized that studies on human mobility patterns would be vital for better sustainable urban planning and a boost for the environment's well being given that transportation in 2004 accounted for 22% of primary energy use[14].

Steenbruggen et al. mention how mobile geolocation data can be used to differentiate weekday traffic patterns from those in the weekend [18]. Another specific type of prediction based on mobile usage discussed in [15] is jam detection. Macroscopic monitoring and analysis of vehicle mobility through mobile traces is a wide area of study on its own which can branch in many fields of study [18].

1.5 Problem definition

From this research it is required to demonstrate that it is possible to attain an accurate measure of traffic flow and predict traffic flow for specific locations from predefined time intervals ahead. It is required to prove that this can be possibly

done by constructing a predictive model and make use of inference techniques that base themselves on data usage records collected from the mobile cellular network. As we will expand in Chapter 2, the trajectory path plotted by the mobile antenna through which users are given service is far from being a true picture of the actual path of the user. Another aspect of the problem is to detect when and where user trips start and finish and how this can be translated into traffic load on the road infrastructure.

An algorithm must be devised to deduce the most probable path taken by the user for his most common trips. The predictive model has to predict the traffic in a responsive manner since predictions that take a long time to compute will not be useful.

1.5.1 Research Questions

Research will be done in a direction outlined by the questions below:

1. Is the resolution of mobile data usage cell tower location fit for purpose to measure vehicular traffic flow on the road network?

2. How is it possible to extract the geolocation of main areas of activity from user's mobile data usage records?

3. Is it possible to extract trip information that is based on the users' main areas of activity?

4. What is the best approach to analyse traffic flow on the road infrastructure over time in space, given that trip information is available?

5. Is it possible to model traffic flow over time with machine learning techniques that use mobile data usage or processed data derived from it? How much time ahead can the model predict traffic flow with an acceptable margin of error in such a way that the prediction is useful and practical for trip planning and traffic management systems?

1.6 Aims and objectives

In this dissertation we will focus on measuring traffic flow and predict how traffic flow changes over time for a selection of locations by using mobile data usage. There are different metrics that we found to be treated as a topic in literature to measure traffic as it is discussed in Section 2.5. By traffic flow we opted for simplicity and chose volume in a specific point in time as a definition to work with. Within this definition, a direct relationship between traffic flow and traffic congestion is not necessarily implied. To determine the traffic slowdown due to congestion the capacity of the road segment needs to be known in order to check the volume-over-capacity ratio. This metric does impact the traffic delay [10].

A combination of data mining and machine learning techniques will be used to devise a data processing pipeline. This data processing pipeline will:

1. consume raw event data records containing cell tower locations and date time and carry out preliminary descriptive statistical analysis.

2. zoom into the main areas of activity of users by using unsupervised machine learning techniques that cluster the most dense groups of geolocation data points.

3. determine routes between these main activity areas by using third party tools and collect spatial grid aggregate data from daily trips done along these routes from thousands of users.

4. use the transformed data which is representative of traffic flow in various locations to train and validate a predictive model using artificial neural networks [14, 10, 15, 19].

5. feed visualization tools that enable visual inspection of traffic patterns projected on maps.

A selection of methods that are encountered in literature will be applied and evaluated. The real challenges arise in the quest for a high spatio-temporal resolution when modelling traffic, given that mobile usage records' geolocation dataset is sparse and tracks the position of users with a considerable margin of displacement error [15, 16].

1.7 Dissertation outline

This dissertation started by introducing the reader to the problematic nature of vehicular traffic. It continues by expanding the socio-economic impact of traffic and how it can be addressed with modern technology. At the outset, it is mentioned how mobile data usage has great potential to monitor traffic conditions and to predict it over time.

A background on traffic flow detection and prediction, and an overview of related literature, will be given in Chapter 2 "Background and literature review". The proposed method to demonstrate the soundness of certain selected implementations of certain concepts inspired by literature will be elaborated in Chapter 3 "Methodology". Validity and versatility of the created model will be evaluated and discussed in the "Evaluation and Results" chapter. The "Conclusion" chapter will summarize what has been achieved and to what extent in this dissertation, while highlighting limitations in the process. Finally, Chapter 6 "Future Works" will discuss possible improvements and potential future projects that can build on the work done in this dissertation.

2. Background and Literature Review

This chapter will go over mainstream techniques and approaches that make use of mobile data for traffic flow detection and prediction.

2.1 Mobile location data sources

Network derived user location is an important attribute of a mobile cellular network. It is used to trigger call and data session handover and to enable a network to locate a user. Network paging is used to find the initial location of a mobile user. Other records are generated when there is a location update and hand over information [20]. Network signalling records contains rich metadata to troubleshoot network issues. These records include also geolocation data.

From a telecommunications background perspective there are two types of generated records. These types are call data records (CDR) and event data records (EDR). An EDR, differently from a CDR, comprises other forms of activity other then calling. Both CDR and EDR data are generated by network elements to capture and report user activity within the network. Reporting frequency and record triggering events can be configurable, allowing operators to trade off between keeping at their lowest the quantity of generated records that are hungry of storage

resources and providing enough data for billing/troubleshooting purposes.

Mobile internet is the service that generates most records. As soon as the user connects to the network, a first record is generated, providing all of the available information, including which cell tower is providing the service. Since data sessions span over a long period of time, periodic updates are required, allowing billing related entities to control whether the user may continue to make use of the service or not. These updates may be triggered by either of the following:

1. Volume - a new record is generated as the user consumes more than a pre-configured volume.

2. Time - if the user is idle, a new update record is still generated after a specific amount of time from the previous record.

3. Network Trigger - operators may decide to generate a record each time there is a specific change (for instance, a change in radio access technology)

Together with call records, SMS records (messaging) and data traffic (2G/GSM, 3G/UMTS, 4G/LTE) records can also be stored. SMS records structure are similar to those of CDRs [14]. A CDR structure would include the A-party (who is calling), the B-party (the person who is receiving the call), call duration, date and time of calls amongst other things which might not prove to be useful for location deducing purposes. The location is implicitly the sector of the base station antenna which was managing the call/sms and where ultimately the CDR has its origin. The trigger for a cell handover or for a 4G to 3g or 2G handover is dependent on the received signal strength as well as cell congestion [11]. This has an implication on location detection as we will see later on. A data event record would include volume of transmitted data in the session.

Mobile device location traces have their limitations when used for vehicular traffic analyses. In contrast to surveys, they lack demographics [14, 3] and market share of the mobile service provider that made the dataset available for scientific research might not be really representative of the commuting patterns [20]. Many studies

highlighted the importance of removing bias when preprocessing such datasets before any further processing is done [21, 10]. Passive data, gathered in the form of CDRs, are not suited to extract different modes of travel, route assignment and classify detailed activity types [3].

Mobile device location data is not only limited to data that originates from cellular networks. Global positioning system (GPS) is the most reliable source of geolocation because of its higher resolution with lower margin of error. This data is generated by the device and needs to be stored and communicated from the user's mobile with his own specific authorization. Using GPS data for a mobility study is more challenging because it needs the consent of users to get such data and drains the battery quickly especially because of long signal acquisition time[22]. Thus users would be reluctant to have such service running in the background on their mobile phones all the time [23].

CDR data was the mobile location data source mostly used in recent research [17]. The intention of our research is to use data usage EDRs since these can have a higher temporal resolution. CDR data can be more commonly generated when a user is not moving unless he is using hands free in his car. CDRs therefore would be more suited for home and work location detection whilst data usage records would be more generated frequently both when user is moving and stationary. From the literature review it results that most research projects use voice CDRs to trace mobility. Research projects that were found to rely on mobile data usage to detect vehicular traffic or predict it include [15, 20].

Other sources of geolocation include social mobile application recorded events such as check-ins in facebook [15]. Such data can be accessed by available APIs [1].

[1]https://developers.facebook.com/docs/graph-api/reference/v3.0/checkin (accessed April 7, 2018)

2.1.1 Mobile usage data format and dataset sample structure

It is important to analyse in depth the structure of mobile records dataset sample and the method of collection thereof in order to understand possible limitations and strengths in related research. Another topic of special interest is the use of secondary datasets used to validate results achieved when modelling travel on mobile generated data. Hoteit et al. (2014) utilized mobile data coming from 1 million users between July and October 2009 [15]. The data consisted of calling and messaging parties' anonymous id together with data of when users made a data session. Calabrese et al. (2013) used data originating from CDRs (a sample of 1 million mobile users in Massachusetts) which included calling id, time of when call/sms was sent or received and when a data session is initiated [14]. Interestingly Calabrese et al. (2013) used vehicle safety inspection data as well. This data was used especially for evaluation. Vehicle safety inspection data was later used to approximately verify the kilometres covered by inferring the trajectory. The time window used by [14] was 3 months long and the area under study was metropolitan Boston.

Calabrese et al. and Colak et al. stressed several reasons why research data samples collected with surveys present a lot of disadvantages when compared to mobile device generated data including sample size which is smaller, update frequency and certain types of time windows that are seldom considered or not captured by surveys such as seasonality, public holidays and weekends [14, 3].

Gonzalez et al. mention two datasets in their research. The first sample was of 100,000 individuals sub-sampled from a wider dataset population of 6 million anonymized phone users. Similar to other research aforementioned, the data which was used included id of device from which calls or sms originated or terminated and location of tower projected over time. The reported average area covered by a cell tower was 3 km^2 with 30% of cell towers having a coverage of 1 km^2 or less.

The second dataset consisted of 206 mobile users whose actual location was traced every two hours for a week. By comparing analysis of this dataset to the first one Gonzalez et al. found irregularity in calling patterns observed when using CDR data only. Displacements were recorded for consecutive calls in order to construct a travelled distance distribution model.

Hoteit et al. use two datasets which have GPS location of 86 mobile users in various places in the world [17]. One dataset is generated by sub-sampling the original one in order to emulate a sparse CDR dataset. Authors were forced to do this since no real CDR dataset was available for their research.

In literature two different types of tools have been found to be employed to aggregate location data. Airsage datasets were found often to be used in literature [15, 24, 14, 12, 22, 3]. Basically Airsage does not simply just record the tower cell sector but depending on a refined triangulation algorithm it gives a more precise location. Hoteit et al. (2016) [17] use MACACOApp which is an app that records mobile data usage but most importantly also GPS data. As already aforementioned GPS technology gives a more accurate geolocation. However the data sample size is smaller in comparison to data samples collected in the form of raw CDR data in other studies.

2.1.2 Mobile position inference from Floating cellular data

Collection of localization data that makes use of mobile phone data connectivity with base stations is commonly referred to as floating car data or more specifically floating cellular data (FCD) for sensor data originating mainly from cellular networks. A specific technique to actually determine a user's location is based on triangulation as done by the Airsage solution [15]. An intuition about triangulation is shown in Figure 2.1 which is reproduced from an article by Phil Locke [25]. The red ellipse is the location boundaries for the phone. Proprietary algorithms process data received from mobile service providers and outputs refined location information to customers. It was reported that in testing carried out by Geostat

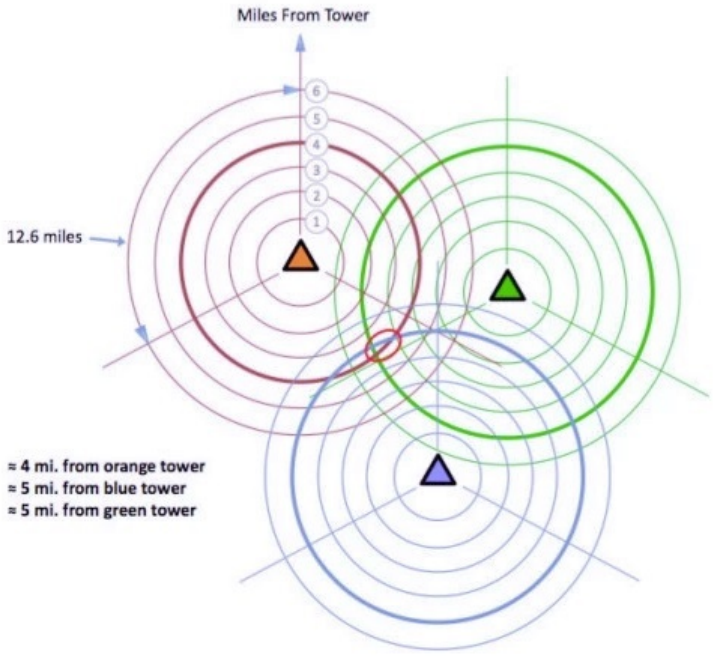


Figure 2.1: Illustration of the triangulation concept which is based on signalling and negotiation with nearby towers. Triangulation computes the location information inferred from signal strength experienced by the mobile phone user. If no second or third tower was present the location would be considered anywhere within the sector on the circumference that is demarcated by the signal strength. The red ellipse is the location boundaries for the phone. As reproduced from article by Phil Locke [25].

Inc. Airsage got accurate classification of congested traffic 91% of the time [22]. No technical background was made available on how these algorithms get a more precise location of mobile users and this is most likely attributed to the fact that the algorithms behind the solution are patented [2].

It is stated in [3] that Airsage location computation accuracy is within 200 to 300 metres. Calabrese et al. state that the degree of location precision reported by AirSage is an average of 320m and median of 220m [14]. As already mentioned, AirSage has been used in [15] as well. In comparison [16] simplistically mentions that 30% (average is $3km^2$) of the towers are placed in a density of 1 tower per square km. This roughly would mean that, at most, an unprocessed location

[2]https://patents.justia.com/assignee/airsageinc

retrieved from cellular location data would have a maximum error of around 500m.

Location given by cell tower position, to which mobile phone user is connected to, is not useful for precise localization since recorded geolocation might be hundreds of metres away from the actual position. In our research we cannot make use of Airsage datasets. Such solutions must be already in accordance with mobile network operators from where mobile usage data is acquired.

Therefore, in this research some effort had to be dedicated to devise a simple triangulation or clustering method that can achieve more accurate mobile user location than the actual cell tower location. Since the grouping of multiple mobile users within a grid of location cells is more practical in giving a more clear traffic flow measure an essential topic to treat in traffic congestion research is spatial binning. Spatial binning determine geographical aggregate statistics.

MapReduce frameworks such as Spatial Hadoop exist so the expensive temporal geospatial analytics are done within an acceptable time window [26, 27]. Such tools provide the possibility of doing spatial joins that can correlate spatial features extracted from sources such as OpenStreetMaps with mobility data from a mobile usage dataset [28].

2.2 Privacy and data anonymization

Mobile subscribers location and CDRs are highly sensitive data, therefore anonymization is required to protect the privacy of individuals when using such data for research purposes. Some standard practices that make anonymization procedures more robust are listed in Laurila et al. [29]. For instance Laurila et al. exposes facts regarding potential privacy breach risks within datasets that have unique identifiers hashed. One of these is contractual binding where data users are legally bound to not attempt to reverse-engineer identity. Another method to further protect sensitive data through anonymization is truncation of data. For example only a part of the data is kept if it is decided that it is enough and meets the purpose of the data

processing exercise. Also Laurila et al. also give a detailed account of techniques used to hash unique identifiers. Shin et al. elaborates on how to use k-anonymity algorithm so that location data of a user makes his identity undistinguishable from other k-1 other users in the same region [30].

Another way to guarantee privacy is limiting data retention. To safeguard privacy travel paths of a specific mobile user is kept for not more than 1 day in [15] and for not more than 2 days in [14]. It is the norm to assign a hashed anonymous identifier to each mobile user in such a method as well.

2.3 Big Data, the cloud and large scale real time stream processing

Liu Jun et al. expand on the recent phenomenon of computing systems not keeping up the pace with the vast increase in storage requirements [11]. In this study it is explained how intensive computing speeds are compromised when there is a lot of data reading and writing and when it is required to move both input and output data around in a distributed system for further processing and consumption.

Big data frameworks are suited for such scenarios. It shards the volume of data on a cluster of nodes and makes the addition of a new node in the infrastructure seamless. Failure of a node will not disrupt an ongoing global process since data will be redundant. Data redundancy is implemented by replication of data in blocks residing in nodes across the cluster. Hot replacement of failing nodes is also a smooth operation in big data infrastructure. The main paradigm shift to attain a high performance gain was to limit distribution of data across the network because this would mean less efficiency because of network transmission latency. The main strength of new Big Data architecture resides in offloading processing to the nodes where the data is located and only the resulting required information is transmitted back to a centralized node where the driver program is [31].

When is the data infrastructure of a system in need of a shift to the Big Data

paradigm and traditional RDBMS systems cease to be the preferred choice? When you have the 3 V's which are volume, velocity and variety in the data is a recipe for big data introduction as a part of the solution. This is quite applicable to the processing of the multitude of mobility data which comes in huge amounts and need to squeeze out information in the least amount of time. Our dataset that was collected between August 2016 and September 2017 is 150 GB in size. Building a predictive model on this dataset of such size and retrain in real-time would require a big data solution.

Currently some of the leading frameworks in this area are Hadoop and Spark. Hadoop is treated in detail in [11] and revolves around the MapReduce programming model. This work shows how enormous amounts of data is stored in a distributed fashion on HDFS (hadoop file system) which is highly scalable and fault tolerant. Spark is used extensively in lambda architectures that include both nightly batch and real time batch processing. Liu Jun et al. research is not directly related to the analysis of mobility behaviour but describes well how to process mobile device generated data traffic [11]. It also gives a good account on how to monitor the infrastructure through various metrics and tooling. There are many papers related to mobile user travel pattern prediction that make use of big data innovation [11, 29, 32]. Toole et al. [10] state that dataset size can be an issue for computation when determining the origins and destinations (OD) matrix (refer to Sections 3.5 and 2.4). In this same study parallelization is used to assign routes to trips.

2.4 Origin and destination matrices computation

A consistent recurrence in traffic flow analyses literature is the study of how to deduce origin and destination (OD) locations for travelling vehicles [21]. Many research articles confronted the problem posed by traffic congestion detection by first deducing the OD matrix [10, 21, 19, 20, 14, 3].

2.4.1 Use of OD matrices to store information on main user locations and trips inbetween

ODs are used to extract main activity hubs. Gonzalez et al. state that 40% of the time users are at their two preferred locations [16]. Therefore most trips can be mostly explained as being between several locations since users tend to be highly inclined to be regular in spatial and temporal terms. All this leads to safely assume that the majority of trips are between home and work. In literature it is commonly found that locations that were likely to be recorded in OD matrices were home and work [20, 3]. In [20] home location is detected for user by checking which 500 metres square cell has the most activity during the night for every specific user.

Colak et al. label zones such as home and work and tries to find purpose behind other types of trips [3]. In this work it is mentioned as well how ODs are analysed in terms of stays and trips. Frequency of calls and time of day determine the labelling of these stays. It is stated that it was not possible to categorize other types of stays other than home and work. So these types of stays were labelled under the 'other' class.

Calabrese et al. (2011) put forward the concept of virtual location which is derived from fused visited locations by the user [20]. This research devises an algorithm that localizes the centroid of important locations in a user trip that are to be labelled as the origin or destinations of particular users. The method analyses which points are in the proximity of others within a 1 km radius.

2.4.2 Cleaning of data, removal of noise and minimization of displacement error

A common occurrence in literature is to remove users that do not make enough mobile usage. The behaviour of these is less predictable and its more difficult to generate trips from OD data for this type of users. In [10] users that do not make enough calls are filtered out from the dataset and [3] filters out users with low

activity when labelling activity zones.

Displacement errors due to sudden change of cell tower for various reasons are reported to make datasets inconsistent. False displacements are reduced in [21] by using a time window of 10 minutes. The most common location in the 10 minute window was considered the actual location. A time window of 1 hour is then used to detect trips. In Calabrese et al. (2011) a low pass filter is used to minimize localization errors [20]. To reduce sudden movements due to cell tower handover clustering is used. The same concern is raised in [3] in a similar fashion and it is described how CDR data contains jumps or oscillations which introduce noise in the dataset.

It is mentioned in [3] how Airsage dataset inherently provides triangulation that gives medoids as processed data. Filtering out of noise in a Rio de Janeiro CDR dataset is done by labelling stays only if records are registered for a user for more than 10 minutes. When observing stays for users for a longer period of time it is possible to get more clear patterns where the stays are actually visited by users or not. Toole et al. [10] remove noise from mobile phone calls deduced trajectories by using the stay algorithm proposed in Zheng et al. [33]. A location is labelled as a 'stay' whenever user makes a set of calls within a time window greater than a given threshold. The centroid is then calculated for a set of locations that are close to each other in order to compute a better approximate location of the user. In [21] estimation of OD matrices can be found to be unreliable because of sampling bias. Equally [10] stresses that bias needs to be removed when constructing OD matrices.

An important attribute to consider in OD matrices is its resolution since it might be important to aggregate data for statistical purposes. Not a lot of information was found in literature on this. In [3] census tracts and town boundaries are chosen for OD resolution level. No justification for this choice is given though.

2.4.3 Use of scaling methods to get more accurate OD matrices

Scaling methods to scale traffic flow counts are often used to obtain OD matrices that reflect reality better. In [21] a scaling factor β was used to get an OD matrix for the scaled up traffic flow between nodes. Traffic flowing from node i to node j was scaled up by this scaling factor β. The scaling factor is obtained by inputting optimization formulas, route choice probabilistic models, network data and the original OD matrix in a simulation engine. The scaling is then distributed as shown in Equation 2.1

$$OD_{ij} = \sum_{ij}(t - OD_{ij}) * \beta_{ij} \tag{2.1}$$

In Equation 2.1, OD_{ij} is the final resulting actual OD matrix. $(t - OD_{ij})$ is the transient OD matrix that contains trip data from origin nodes to destination nodes. Transient here means that the node to node trips may be missing the actual nodes' information because CDR data does not capture all locations in the trips made. Thus $(t - OD_{ij})$ represents only a segment of the actual trips. i, j represent different links between nodes. A simulation platform, MITSIMLab[3], was used to find a scaling factor β_{ij} for every transient OD link.

Colak et al. uses the iterative proportional fitting (IPF) upscaling method [3]. Here Colak determined the expansion factor for each tract and in the IPF took in consideration trips to destinations as well. In his conclusions Colak stated that the IPF Procedure to distribute user CDRs according to population might have been too simplistic of an approach.

2.4.4 Route selection for OD matrices

Route selection is necessary to link origins and destinations from OD matrices to generate OD trips. In [21] route is determined by a function of least travel time

[3]https://its.mit.edu/software/mitsimlab (accessed November 14, 2017)

path. In [10] Open Street Maps[4] (OSM), which is an open source map editing framework, is used to infer routing. Some studies assign trips to a user when there are consecutive calls in the same day and the calls are done from different locations. An example approach is that two consecutive 'stays' that are not more than 1 day apart would constitute a trip [3, 10]. OD matrices determined trips would not be sufficient to model traffic on a network. Microscopic traffic assignment dependent on these trip generation exercises needs to be modelled. Toole et al. for instance implements incremental traffic assignment (ITA) in which trips are added to network incrementally [10]. Then on each iteration routes are assigned according to capacity saturation of roads. It is admitted however that Wardrop's equilibrium adapts better since routes are changed dynamically depending on congestion. However ITA algorithm is chosen because it is simpler to implement. Colak et al. relies on a probabilistic model for traffic assignment. Departure times for trips are set according to pre-set distribution of departures [3].

2.4.5 OD matrices evaluation

Evaluation related to OD matrix generation is generally done by correlating the generated locations and trips to survey data. Toole et al. [10] compare survey data traffic load on road network with that generated through OD matrices formed from mobile CDRs. Simulation generated routes for the latter have been produced with the ITA approach. Toole et al. state however that other methods should be further explored to remove uncertainty from the proposed techniques.

Iqbal et al. collected traffic count data on a spread of 3 days in 13 locations and this data was used for calibration of the system proposed [21]. For validation another day was used with 4 different locations. Prediction root mean square error (RMSE) and root mean square (RMS) percent errors were 335.09 and 13.59% respectively. In [20] evaluation was done against a tract by tract census. Euclidean distance was calculated and the distribution of the trip distance confirms Gonzalez

[4]https://www.openstreetmap.org (accessed December 7, 2017)

affirmation that trips follow a random walk [16] which is discussed in section 2.6 (See equation 2.2).

$$P(x) = (x + 14.6) - 0.78^{-x/60} \; with \; R2 = 0.98 \tag{2.2}$$

Here euclidean distance added error and to have it visualized with lines emanating from and linking nodes, although it might prove to be simpler, it would not give more insight on the road infrastructure use. In the OD trip analysis done by [20] it is estimated amongst other things that a user makes 5 trips on weekdays and 4.5 during the weekend. This matches approximately the US census data which is 4.18 during weekdays and 3.86 on weekends. Study concludes that the OD matrices that are produced with the proposed methods can be of great value to those who are responsible for traffic planning.

Colak et al. carried out evaluation against traffic surveys and already available OD matrices from department of transportation [3]. The validation however was done against a morning sample. Colak et al. boasted of trip generation and attraction correlation near to 1 for both cities in study namely Boston and Rio de Janeiro. The correlation with already validated datasets is highest when OD matrices are generated from aggregations done on larger polygons.

Colak et al. reported OD matrix limitations. Suitability of CDRs to determine ODs is only good at a certain resolution. It is stated that better results are attained when using higher resolution for home or work location detection and aggregation within larger zones (towns or districts) for OD trips representation. OD matrices are less fitted to get information on the whole travel model which for example includes modal split information.

2.5 Traffic flow measurement and pattern extraction

Traffic flow measurement can be explained in terms of vehicle count per t amount of time or even in a more descriptive way with a metric that measures travel performance as volume over road capacity V/C [10]. The latter metric has more information since a road with low capacity may be more congested than another that has the same rate of traffic flow but a higher capacity. In a more elaborate metric proposed by [10] a road can be possibly classified as a function of betweenness and usage. Classes are defined as connector (high betweenness and high usage), attractor (low betweenness and high usage), peripheral (high betweenness and low usage) and local (low betweenness and low usage).

2.6 Model fitting to human mobility

Mathematical modelling of human mobility is important to predict with a stated certainty the location of a mobile user in time since data collected from mobile devices is sparse. Interpolation methods were used to describe human mobility patterns in [15]. These are namely linear-interpolation, nearest-neighbour interpolation and cubic interpolation. Linear-interpolation would simply project the mobile user position at time (t) by plotting a straight line from the last previously recorded location and the one right immediately after. This method's error margin is widened if data collection time interval is longer for data points pertaining to the same individual that is moving. As for the nearest-neighbour method, location is attributed to the previous recorded value or to the subsequent depending which is the closest on the time axis. The cubic interpolation is best explained when contrasted with the linear one. This method as explained in [15] is described as "shape preserving". The slopes shaping the curves are deduced from derivatives and give a less sharp demarcation and better guess depending on a series of data

samples.

In [16] both the variation of displacements for consecutive 'steps' (i.e. call location and respective time at which call is made) and the radius of gyration distribution was modelled as truncated power-law which is referred to in all the work as a levy-flight (See Figure. 2.2 and Equation 2.3 for illustration of displacement distribution modelling). A levy-flight is a random walk where the probability distribution of the steps taken is heavy-tailed. This model explains what is the probability distribution $P(\Delta r)$ of the distance travelled from radius of gyration by individuals who travel as far as 400km (D_1) and those who travel as far as 80km (D_2). D_1 and D_2 are cutoff values.

$$P(\Delta r) = (\Delta r + \Delta r_0)^{-\beta} exp(-\Delta r/\kappa) \tag{2.3}$$

with exponent $-\beta$=1.75 ± 0.15 (mean ± standard deviation), Δr_0=1.5 km and cutoff values $\kappa|_{D_1} = 400km$ and $\kappa|_{D_2} = 80km$

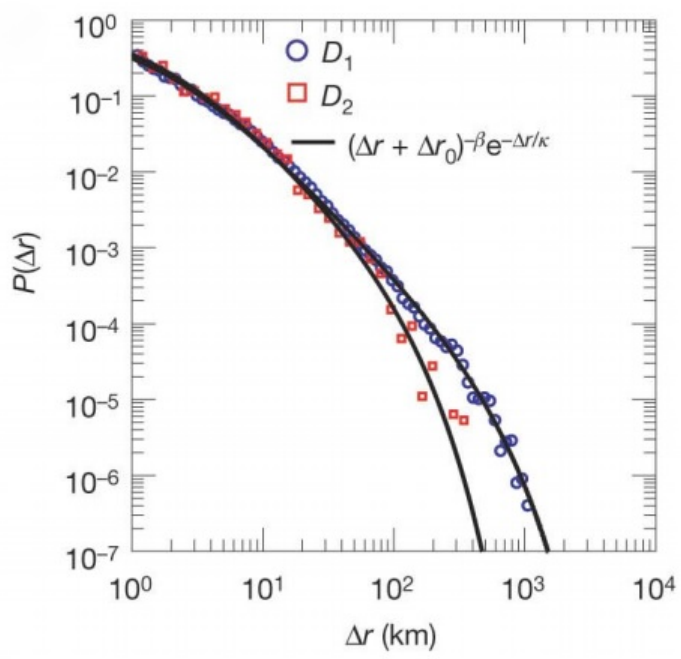


Figure 2.2: Truncated levy flight human motion modelling. Reproduced from [16]

This mathematical model is cited and verified in [14]. The methodology adopted

in [17] suggested approaches how to determine home and work locations, span of movement and complete trajectory. Two datasets were compiled. One dataset is a sub-sample of another dataset which has a higher resolution because it has been composed of GPS geolocation data. The sparsity of the second dataset have been mimicked by a cumulative distribution function in order to create a virtual CDR dataset. Only users with high activity were considered in order to have less irregularity. Home and work locations were determined with a mode function with catch-all time boundaries for day and night where supposedly users are either at work or home respectively. For span of movement a similar mathematical approach was adopted as the ones in [15, 16]. As for the actual movement trajectory error was calculated by calculating the euclidean distance of each CDR data-point from the actual GPS recording which is nearest in time. Some techniques were used to lessen the margin of error. Since most of the time the typical mobile phone user is static, data completion is attained by applying a list of inference rules for which different results are achieved when estimating users location, hence the name of the paper "filling the gaps".

A problem was outlined in [14] about detecting a lot of trips in very short distance which do not reflect statistical data given by surveys. This is explained as being caused by fluctuating random connections with towers which spatially misplace the user when in reality he is not actually physically moving. This issue was tackled by mathematically creating so called by [14] 'virtual locations' (a mass/group of traced positions in a given radius of Airsage resolution) and actually recording a movement when user moves from a virtual location to the other. Calabrese et al. limit static location detection to the home location and the proposed process how to manage to get each user's location is similar to that expounded in [17]. In a novel style this work studies the relationship between total trip length calculated from mobile phone location data and vehicle kilometres travelled (VKT) and urban features such as entropic type, population density, intersection density, average distance to non-work destinations, distance to subway stations and high-

way exits. These urban features were derived from US Census of 2000 and activity travel surveys.

Estimation of load error is proportional to concentration of users in a given block [15]. For recorded error with value less than 1 km a probability of 80.78% of being within commonly travelled territory contrasts with a probability of 19.22% when user travels outside of it. From the opposite perspective of having error greater than 1 km, the probability of being inside radius of gyration was found to be 40.25% and that of being outside is 59.75%.

In [14] results state that 49.40% of mobility variation can be explained for individual mobile users and 56.48% for vehicle associated mobility in terms of trip length.

In [16] results point to the phenomenon that the greater is the radius of gyration the less symmetric in shape is the probability density function which gives the probability of a user being in a given location (x,y). Also the margin of error increases similarly as stated in [15]. It is also shown how individual mobility is well described by a levy-flight. Also a probability density function has been implemented to give the likelihood a user is at a certain given place in time.

The techniques used to further refine the location based on the assumed location home interval gives results in the range of 92%-95% of cases within 100m [17]. Techniques will produce large errors (in the range of 50km) when user travels long distances and may not return to home location during the usual time interval.

Error distance from trajectory depends on radius of gyration [15]. Interpolation methods are found to be most suited depending on distance from the geometrical centre of all the movement. Nearest neighbour is most suited for r_g less than 3 km. Between 3 km and 10 km both linear and cubic interpolations perform well. For commuting travelling patterns trajectory is best estimated with a cubic interpolation. Interesting insights were contributed in [14] where it is stated that job accessibility and distance to non-work destinations are inversely proportional to total trip length. Distance from subway does increase trip length for individual

mobile users but it does not impact vehicle use. This means that subway commodity does not necessarily decrease vehicle use in the surrounding area. Vehicular trip length decreases when correlated with increase in intersection density but not so for individual mobile users. Urban entropy and population does significantly impact trip length. Thus this study can help a lot in urban planning and large scale policy making. In [17] it is affirmed that the solution of data completion augmented by the placing of users in their home location at inferred intervals of time produces better results then what was achieved in literature.

There are many approaches in literature how to classify group mobility patterns under specific categories. Hoteit et al. (2014) segmented mobile users depending on the width of the radius of gyration (r_g). The different distinguished categories of users are listed as sedentary, urban, peri-urban users and commuters. The classification boundary was decided upon steep changes in the cumulative distributed function of the radius of gyration. Respectively these classification labels fall in the ranges $r_g <= 3km, 3km < r_g <= 10km, 10km < r_g <= 32km, 32km < r_g$ [15]. The radius of gyration (see eq. 2.4) is the notion outlined by the sum of all displacements from the centre of mass divided by the number of trips. This parameter describes how distributed are the trips far away from the zone where the user mostly frequently returns. Repeated utilization of this mathematical notion is found in [15, 16, 17].

$$r_g = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\vec{p}_i - \vec{p}_{centroid})^2} \tag{2.4}$$

where

$$\vec{p}_{centroid} = \frac{1}{n}\sum_{i=1}^{n}\vec{p}_i \tag{2.5}$$

where r_g is the radius of gyration and $\vec{p}$ is the magnitude of the displacement vector.

In [17], the hypothesis that an individual tends to be found with high probability

at his home or place of work leads to classification of individual calling activity locations. The classification method labels these locations as *'stop-by'* categories. The 'stop-by' category 'stop-by home' is demarcated by the night time interval where a user is expected to be at home. 'stop-by-flexhome' is a refinement over and above 'stop-by-home' were night time interval varies per user. 'Stop-by-spothome' fills data gaps or corrects errors when there are exceptional errors where user is expected to be in home location, as indicated by previous category.

3. Methodology

One of the main objectives (O1) was to extract meaningful features from mobile data usage that would serve as the basis to build a traffic flow model. Before choosing an approach and constructing an algorithm that maps raw data and translates it into traffic flow metrics a thorough familiarization exercise with the data was due. A feasibility check had to be carried out on whether it was possible that by devising an algorithm a direct relationship is established between mobile usage data and traffic flow. If data was found to be too sparse, both temporally and spatially, augmentation would have been required by tapping into other datasets. Such datasets could include ANPR data collected from video streams, call data records available openly on the internet, accident reports and anything related to road transport which would convey additional information on human mobility patterns.

In this chapter we will elaborate how we investigated a number of approaches and how we decided to dig deeper with a set of techniques preferred to others depending on how practical the solution was and how it would give a better result. The methodology presented in this chapter has a step by step scientific method in which procedures were devised to test hypotheses. The data that results from a devised procedure was analysed and if it was not working properly further optimizations were done and alterations were discussed. Finally conclusions were drawn and communicated in Sections 4 and 5 and where results were not found

aligned or partially aligned with the hypotheses, suggestions how further research can be done were outlined in Section 6.1.

The main hypotheses' experimental procedures that have been tested and refined in this methodology are

1. Clustering of main user activity locations.
2. Trip generation from main activity clusters.
3. Trip count and trip delay measurement.
4. Traffic flow load distribution on road infrastructure.
5. Traffic flow prediciton with different prediction time interval ahead.

3.1 Mobile data collection and structure

The anonymized dataset was provided by GO Plc Malta[1] which is one of the main Maltese telecommunication services providers. The anonymization process was done by the company itself and authors had no access to the original dataset. The dataset recorded ranges from October 2016 to September 2017 which is a full year of data. However it was decided to concentrate only on the month of October 2016 so that analysis and model learning is faster by working on prototypes tested on a sample of the data. The data volume for the month of October was 11 GB. The whole dataset volume amounted to 150 GB. Notwithstanding the fact that all experimentation was only done with data from October 2016, results were considered to be satisfactory even though it is known that for certain machine learning algorithms, training with more data would probably give better results.

The number of distinct cell towers that mobile phone users connected to amounted to several thousands but distinct cell locations amount to only a few hundreds since

[1] https://go.com.mt (accessed May 4, 2018)

a cell tower shares antennas for different technologies. Precise figures cannot be disclosed due to commercial sensitivity.

The procedure how data is collected from the network of cell towers was discussed with the company engineers. The engineers stated that EDR/CDR records are generally buffered to file on the network element. Files are closed periodically. Files are then collected and processed by the mediation platform, which parses, enhances, and extracts all of the necessary information from these records. New files are then sent towards billing and other entities as per required. The delay to make records available in data-warehouse for further processing will have a direct impact on real time traffic flow prediction.

Unique data usage users amounted approximately to 108,000 for the month of October 2016. This month was chosen for its heavy traffic characteristics because schools start and university students start to travel with their cars adding to the load of traffic. This sample represents roughly half of the provider's subscriber base which is just over 200,000. This was found to be coherent with figures stated in the Malta Communications Authority (MCA) Data report sheet[2]. This figure was derived from all the distinct users that make calls or use data. It is important to note however, that the data users considered in this study might not necessarily be directly proportional to the number of moving vehicles at a certain point in time. The dataset includes static users, users who are just passengers in the car, users that have more than one device and other users that make use of other means of transport. Such factors must be taken in consideration when setting up the proposed solution and evaluating results. The records' data structure is shown in Table 3.1.

[2]https://mca.org.mt/articles/data-report-sheet-drs-latest-figures-published (accessed November 8, 2017)

Data item	Description	Example value
A_NUM	user hashed identifier.	5a8bd7889fb3051b10f249a5554c803a
TIMESTAMP	date and time of usage.	2017-01-01 00:00:00.000
SOURCE	Type of Record. Data or Voice.	DATA
CELL_ID	Cell identifier	3073
TOWN	Cell town	Paola
DURATION	Duration of call or data session in seconds	60
VOLUME	Volume of data used in session in kilobytes. Applicable only for records of data usage.	324.34
LONGITUDE	longitutidnal coordinate	14.50664
LATITUDE	latitude coordinate	35.87
RAT_TYPE	Network technology	LTE

Table 3.1: Description of data fields in the mobile usage raw dataset

3.2 Dataset preliminary analysis

The total number of records of data session or voice call type for the month of october 2016 was 125 million with 78% of these records representing data usage records. This means that data usage records are four times as much as the call data records. This fact evidently gives an edge on other research that used calling data records as their data source since the frequency of users location recording is much higher. Higher temporal resolution reflects higher spatial resolution. A user might not make data sessions for a long period of time and therefore his travelling information would be missing for this period. Higher spatio-temporal resolution conduce to better results both when extracting main user activity hubs and when measuring traffic flow counts. Lower sampling rates lead to interpolation error. In Section 2.4 it is described how in literature data with a resolution under a given threshold is filtered out.

Table 3.2 shows some summary statistics about the main unprocessed data set. Minimum and maximum timestamps show that data stretches for the whole month under analysis. The total count of data usage records is 97 million. The data ses-

sion's mean duration was approximately 16 minutes which was quite discouraging. This would entail that on average a wait of 16 minutes would be required to write to data storage a mobile cell EDR. This is not desirable for near real-time future traffic count forecasts because the time for the detected departure is retrieved much later than it would actually have happened in such a way that predictions become useless. This would boil down to having a data session duration length which contributes to a considerable displacement error. Until the user connects to the next cell there is a distance covered within the average of 16 minutes and a standard deviation of 22 minutes which is also very high. For a vehicle driving at an average of 40km per hour this would translate to an average displacement error of 10km.

Summary	**timestamp**	**data session duration (s)**	**volume (Kb)**
count	97718761	97718761	97718761
mean	null	944.702465855047	1008228.7463008869
stddev	null	1367.394246	3791128.444
min	2016-10-01 00:00:00.000	0	0
max	2016-10-31 23:59:59.000	3600	3.5590011E7

Table 3.2: Basic summary statistics of main EDR dataset.

Some interesting facts were noted when a frequency diagram was plotted, see Figure 3.1). 15% of the EDRs have a data session duration of 1 hour. This duration is the limit set by the telecommunications provider for a mobile usage EDR. These records are generated for users who are not moving. Records with such duration were filtered out for a better summary statistics since the main focus is on records that are related to movement. As a consequence more precise statistical information was acquired which describes better the possible level of displacement error and how long does it take to register the first record after a user moves from one location to another.

After we removed the 1 hour duration EDRs newly calculated summary statistics show that the mean and standard deviation decreased to 8 minutes and 14 minutes respectively. This is a 50% gain with respect to previous statistical data. Further looking at the data session duration frequency plot, by overlaying a cumulative distribution it is shown that 80% of the records are below the 5 minute

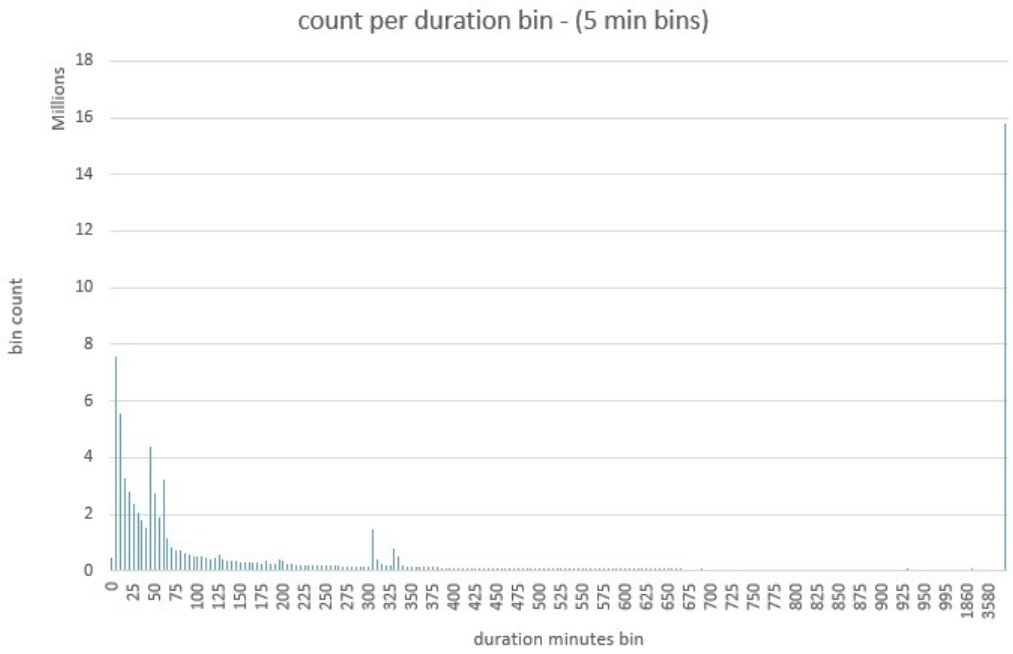


Figure 3.1: The single line column bar spike on the right represents sessions of 1 hr duration.

mark. These data facts have to be all taken in consideration when assessing the usefulness of the prediction results in evaluation and results in Chapter 4.

Summary	**timestamp**	**data session duration (s)**	**volume (Kb)**
count	82589696	82589696	82589696
mean	null	458	1094048
stddev	null	827	4031503
min	2016-10-01 00:00:00.000	0	0
max	2016-10-31 23:59:59.000	3599	35590011

Table 3.3: Basic summary statistics of main EDR dataset after removing 1 hour duration EDRs.

3.3 Algorithm Selection

One of the main challenges involved in this study was to assign vehicular traffic to the road network depending on surrounding cell tower traffic in a time series.

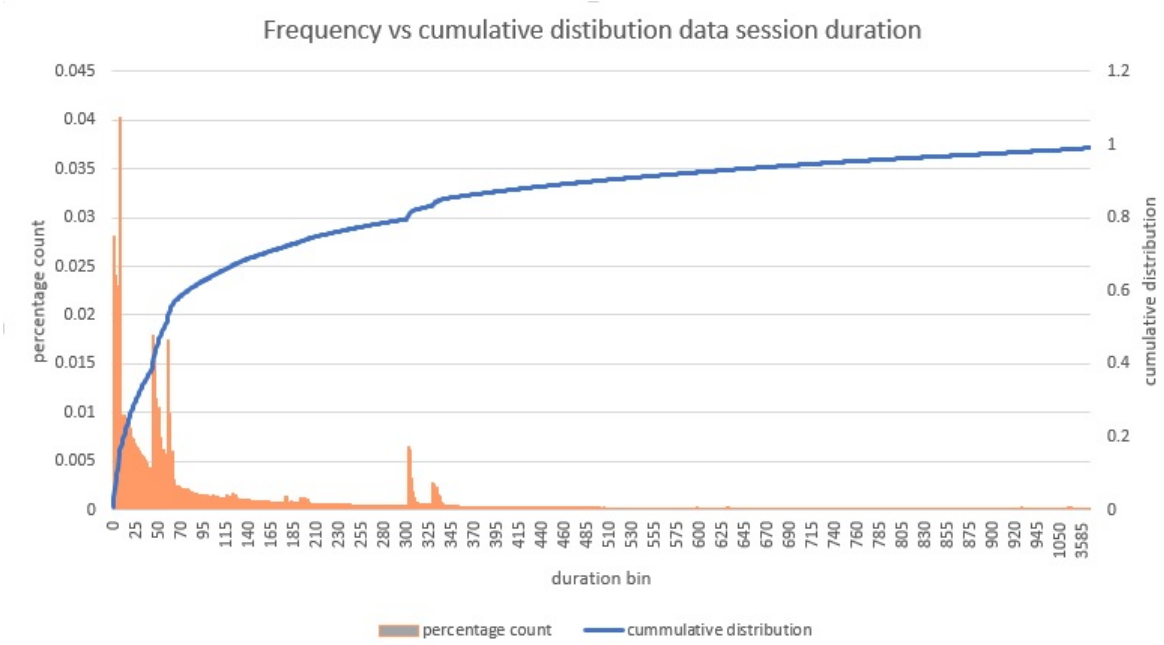


Figure 3.2: 80% of the records are below the 5 minute threshold.

3.3.1 Trajectory interpolation through cell tower data

We first investigated the possibility to snap the cell tower location data to the road infrastructure mesh depending from which direction the vehicle is coming. Various APIs are available to retrieve nearest road from an input geographical location[3]. We investigated how cell tower transmission is configured. There are many settings which determine the range of the cell tower including frequency, rated power, height of tower etc. Given that such information was not available, transmission range of the cell towers was an unknown variable. Another alternative to roughly estimate the range was to calculate the average distance from the nearest k neighbouring towers for all cell towers. However from a sample taken from the available dataset of the cell towers across Malta, the variance seemed quite high, ranging from an inter-distance of 150m in urbanized areas to several kilometres in rural areas (comprehensive cell tower locations map across the country not being

[3]Google Snap to Road API, Bing Map API and OSRM nearest service are examples: https://developers.google.com/maps/documentation/roads/snap (accessed January 8, 2018), https://microsoft.com/en-us/maps/snap-to-road (accessed January 8, 2018), http://project-osrm.org/docs/v5.5.1/api/#nearest-service (accessed January 8, 2018)

shown since it is commercial sensitive information). Furthermore, it was decided to plot the cell towers' on the map and check if their distribution pattern would make it feasible to snap a data record cell tower location to the nearest road or area polygon. Thus here it was assumed that the area around cell towers will have transmission strength with equal range from each tower. Allowance for overlapping was also taken into consideration.

A typical example of how many road sections there are within an area covered by a number of cell towers can be seen in Figure 3.3. One can easily appreciate that a lot of roads are associated to a particular cell tower which makes it difficult to devise an algorithm to derive trajectories and traffic flow counts from cell tower location data. Given that there are a lot of unknowns including how handover procedure is handled in specific areas and the actual range of cell towers, the solution path of snapping to nearest roads depending on EDR coordinates was discarded. This approach would have been impractical to assign traffic to junctions, roads or polygon areas and the probability of inaccurate results was high.

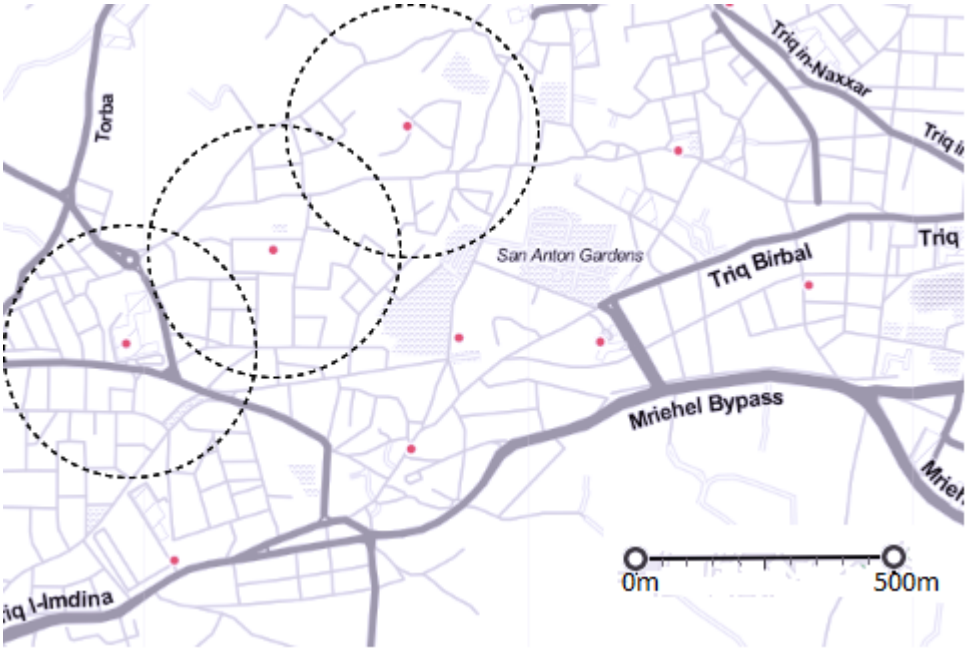


Figure 3.3: Cell tower range distribution. Red dots are cell tower locations and dotted line is the estimated range.

3.3.2 Traffic simulation from OD matrix

Another approach that was considered was to simulate traffic by using statistical information based on travel patterns extracted from the dataset. In Section 2.4 we described how Iqbal et al. optimized OD matrices through simulation. An OD matrix can be used as input to a simulation based traffic model. We discuss how an OD matrix was generated in Section 3.4.

Toledo et al. [34] mentions that OD flows are an important input to simulation models but an accurate OD matrix is difficult to acquire. An example of an implementation of a simulation based on an OD matrix can be found in [35]. In this study, electronic toll collection data is used to form an initial OD-Matrix. This OD matrix is further improved by optimizing a model that gets observed detector data and a simulation based on current OD, computes the least cost difference and optimizes the OD matrix depending on result. This process is iterated until an acceptable coefficient of determination is achieved. Simulation models generally give two types of outputs namely a visual simulation of traffic flow on a map and textual statistical output that can include metrics such as traffic delays, gap distances, speed and overall trip distance travelled by vehicles.

While macroscopic traffic simulators seem promising for motorways environments they were found to be less suitable for urban scenarios [36]. Urban environments have a lot of conflicting traffic flows caused by the numerous junctions and small roads that feed and attract traffic from the road network. Also such simulators require accurate OD matrices. This cannot be achieved by using our dataset due to displacement error created by distance separation between cell tower location and actual location.

In our research we tried to focus on both the macroscopic and on the microscopic level since we had a dataset that has ample coverage especially in urban areas.

3.3.3 Traffic flow detection by trip generation assigned traffic

The method we adopted to detect traffic on the road network involved first the generation of an OD matrix that contains main stay locations for users in a time series. Then a trip is generated between each main location for each user as will be explained in Section 3.5. The trip includes turn by turn directions with longitude and latitude coordinates. Traffic load assignment is then assigned to junctions and turns depending on the time retrieved from OSM (Open Street Maps) data (see section 3.6). The major challenge here proved to be the traffic assignment, given that there is an interaction of a lot of vehicles at a given point in time with a complex structure of roads and unexpected events such as weather, accidents and road blockages.

3.4 Main activity hubs extraction through clustering

One of the main steps of the proposed algorithm is to derive main areas of activity from the mobile data usage of subscribers. This can be achieved using clustering. Clustering was also used to remove noise caused by displacement through frequent oscillations by finding a centroid of activity. The removal of the displacement error through triangulation has been ruled out. There are missing dataset features that are required to get a more accurate location with this process such as strength of signal from every cell tower that the user connects to. Moreover, simple geometrical triangulation does not have the aggregation characteristics that clustering has. Grouping of similar locations have to be implemented on top of triangulation. Triangulation is more suited to remove noise or displacement error caused by cell tower oscillations or handovers. These are caused either because the signal from a tower is weaker from another that can provide better service or there is momentary

offloading causing a user to switch his connection to another tower with less load.

In Section 2.4 it was discussed how certain authors employed various techniques to smooth sudden location change of mobile users because they often switch cell towers in very short time intervals that cannot be attributed to movement.

Clustering is a machine learning unsupervised technique used to classify entities which have similar features. Clustering is done depending on the chosen algorithm and calibration hyper-parameters that control the grouping process. Two clustering techniques that were considered for their appropriateness to this research were k-means and DBSCAN [37, 38].

3.4.1 K-means clustering

k-means algorithm is highly popular especially for first analysis of datasets because it is simple to implement and highly efficient. The main drawback of k-means clustering is the requirement to select the number of clusters before running the algorithm. Then a number of expected centroids equal to the number of targeted clusters are randomly chosen. The algorithm starts to find the nearest neighbours based on a distance metric until finally the clusters are formed. This process can be run iteratively until the ideal set of centroids with the least root mean square error are found. Also something important to note is that clusters tend to be spherical in nature. This would be highly visible if 2D clusters are plotted on a graph. The only advantage of using k-means clustering over using DBSCAN is its speed. k-means performs better than DBSCAN especially for incremental version of the algorithms when datasets are frequently updated [37].

3.4.2 DBSCAN clustering

DBSCAN (density based spatial clustering of applications) has an edge on k-means and is mostly suited to our research since it does not need to set the number of clusters that we are after for each user at the outset. Moreover it finds clusters

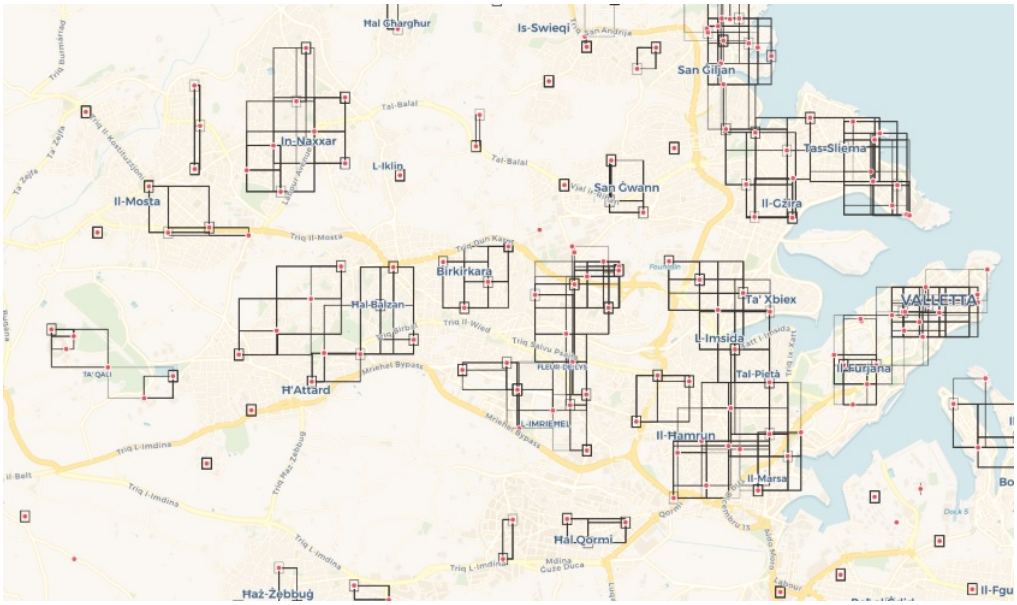


Figure 3.4: DBSCAN clustering to find main user activity hubs. (Sample illustration)

of non-spherical nature and leaves noisy elements out of the computed clusters [37]. DBSCAN has three main hyper parameters to set namely minimum points, ϵ (radius of area within which density is measured) and a distance metric. The algorithm is more sensitive to density rather than to aggregate distance of surrounding points. Basically the algorithm finds core points that have the required minimum points in its neighbourhood dictated by the distance metric. Other non core points that are within core points' radius range (i.e. they are not surrounded with the minimum number of points) are referred to as boundary points. If clusters formed by the core points overlap each other they are grouped together into one single cluster, hence the non-spherical shape of the clusters.

DBSCAN was the preferred candidate for clustering since the aim was to find dense clusters of mobile data usage activity and random locations visited by users are of no interest and need to be filtered out. The curse of dimensionality does not apply here since there are only two dimensions with the same scale. The values of hyper-parameters were 500m for radius, minimum required points was set to 3 and euclidean distance was chosen as the distance metric. The mean distance

between a sample of cell tower locations taken randomly from the whole dataset was calculated to be 350m. The radius was chosen to be 500m to allow for overlapping but not include too many cell towers accept for the shouldering ones. By choosing an excessive ϵ the centroid location coordinates was being too inaccurate and was clustering a wide range of subscribers' activity. With a smaller ϵ minimal clustering (every cell tower will be start to be considered a cluster) was being attained since cell tower location areas will not overlap.

The OPTICS algorithm, which does away with the ϵ parameter, iterates until it finds the optimal ϵ and orders its clusters in a hierarchical result. However this algorithm is more computationally expensive and we opted to use the non-generalized DBSCAN version of the algorithm. The implementation used[4] was integrated into Apache Spark processes that output clusters of usage patterns for every user. The output of the implementation we used was in the form of coordinates that outlined the rectangular boundaries of the cluster. The final geographic coordinates that denote the main activity clusters were those of the centroid. The centroid for each rectangular cluster had to be determined with readily available libraries (esri was the used library)[5] within Apache Hive.

In Figure 3.4 clustering of cell towers can be observed. The centre of the quadrilaterals would mark the centre of activity for the mobile usage. When there are no cell towers nearby the coordinates of the cell tower itself becomes the centre of activity for the mobile data usage.

3.5 OD Matrix trip Generation

OD matrix trip generation consists of a sequence of steps. A high-level overview is given in the form of pseudo-code in Algorithm 1. The main modules of the algorithm will be discussed in detail in the following subsections. The interaction

[4]https://github.com/scalanlp/nak (accessed November 10, 2017)

[5]https://github.com/Esri/spatial-framework-for-hadoop, https://github.com/Esri/geometry-api-java (accessed December 10, 2017)

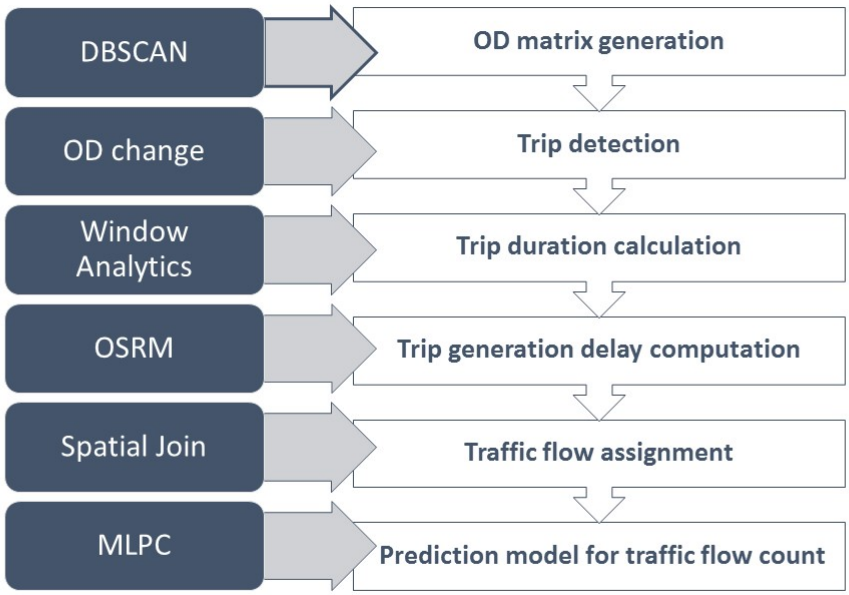


Figure 3.5: This diagram summarizes the data processing required in phases that build one whole pipeline underlying the proposed algorithm.

between these main modules is depicted in Figure 3.5.

3.5.1 OD Matrix computation

In our research we decided to focus on two main areas of activity per user as the basis of our OD matrix generation, namely home and work location. Inclusion of more areas of activity is left for future studies (refer to Section 6.1). It is assumed that most of the trips happen between home and work and vice versa. This is based on conclusions encountered in related literature (see Section 2.4). The top two mobile data usage activity clusters per user were retrieved from the resulting users' clusters created with DBSCAN algorithm run (see Section 3.4). We considered these top two clusters as the origin and destination of trips including returns. Then the user EDRs that have geographical coordinates located in the two main activity cluster areas are filtered into a new dataset through the spatial join technique [27]. This process results in a dataset containing all data usage records

that have a location in either of the top two clusters for any user in a times series. This resulting dataset is substantially the OD matrix.

3.5.2 Trip generation, route choice and traffic assignment

Basic OD matrices, on their own, do not give information on how traffic flow is distributed on the roads because they only represent home and work locations. Therefore we had to further enrich the OD matrix by detecting when trips happen by recording change of user cluster location events. This was achieved by ordering OD matrix entries by user and timestamp. The dataset was then scanned and when location of activity of a given record is found to be different from the previous record, the previous record is tagged as a departure and the current one is set as an arrival. We used Apache Hive's window analytic functions for the computations because it offers an sql-like syntax which we were already familiar with and processing is done on top of Hadoop. This made processing of huge amounts of data faster through parallel, distributed computing[6]. Hive is a data warehouse infrastucture tool running on Hadoop that abstracts a lot of java api calls to get data from distributed file systems managed by Hadoop[7]. Hive has a specific SQL dialect HiveQL(HQL) that can retrieve data from hdfs (hadoop disributed file system) without implementing the mapreduce calls.

There is a caveat on the accuracy of the actual duration of the trip. Records of mobile data usage are generated depending on actual usage at a given location. The frequency of generation of such records has a direct effect on the accuracy of departure and arrival times for any given trip. The higher is the temporal resolution, the more accurate are the departure and arrival times. If on the other hand records are generated at a lower frequency it cannot be determined with confidence and with a low margin of error. For example the user might arrive at his work location but he takes too much time to start his first mobile data

[6]https://cwiki.apache.org/confluence/display/Hive/LanguageManual+WindowingAndAnalytics (accessed February 15, 2018)

[7]https://en.wikipedia.org/wiki/Apache_Hive (accessed November 20, 2017)

session. This would add extra trip delay with the result that the trip duration is less accurate depending on the gap of time between the actual arrival time and the first generated mobile usage record timestamp. Therefore users with more frequent usage of mobile data have trips with durations with a narrower error margin. Departure times are more precise since these are computed by adjusting the EDR timestamp and adding the data session duration. This gives an accurate timestamp that represents when users leave their main clusters (home or work location).

We inferred the routes between origins and destination from the OSM in a similar manner to the work of Toole et al. [10]. A route was assigned for each entry in the OD matrix together with duration information from the trip. The user's routing choice was assumed to be the fastest one given by the Open Source Routing Machine[8] (OSRM). The OSRM api provides the possibility to request alternative routes. However in the approach taken the route selection was always considered to be static and user does not change route depending on traffic or due to unexpected events on the road network such as accidents or blockages caused by various other reasons. The fastest route is attributed to each trip done by each user. This may not always be the case since route selection can differ depending on traffic perception and arbitrary route selection made by users. This is a limitation of this research and introduces inevitable bias. It should be noted however that in the urban scenario in Malta the different routes to take towards work and back are limited due to the small scale of the road network infrastructure. In other words there are few possible routes which users can choose from or that enable detouring, making it highly probable that the fastest path is the preferred choice. In Section 4.3, it is discussed how to measure a level of confidence in the traffic assignment model.

Another important information that was extracted from the route selection is the trip delay. The total duration for each trip per user was retrieved from the

[8]http://project-osrm.org/docs/v5.15.2/api/#route-service (accessed January 12, 2018)

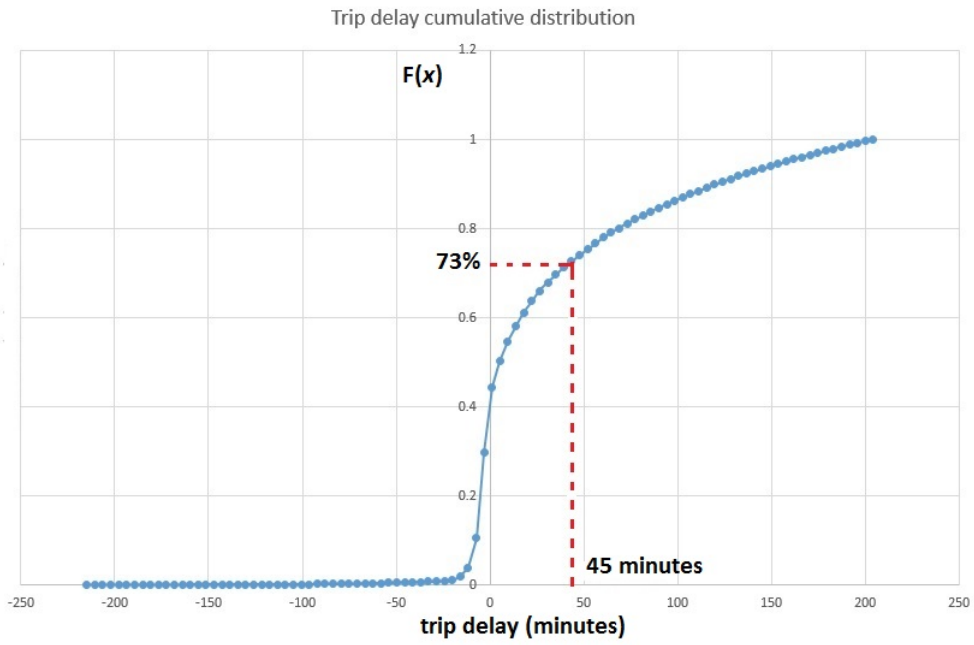


Figure 3.6: Cumulative distribution of trip delay. This figure shows how negative trip delay instances are a very small percentage. Cutoffs of -5 and 45 minutes were chosen to select the trips for the learning model.

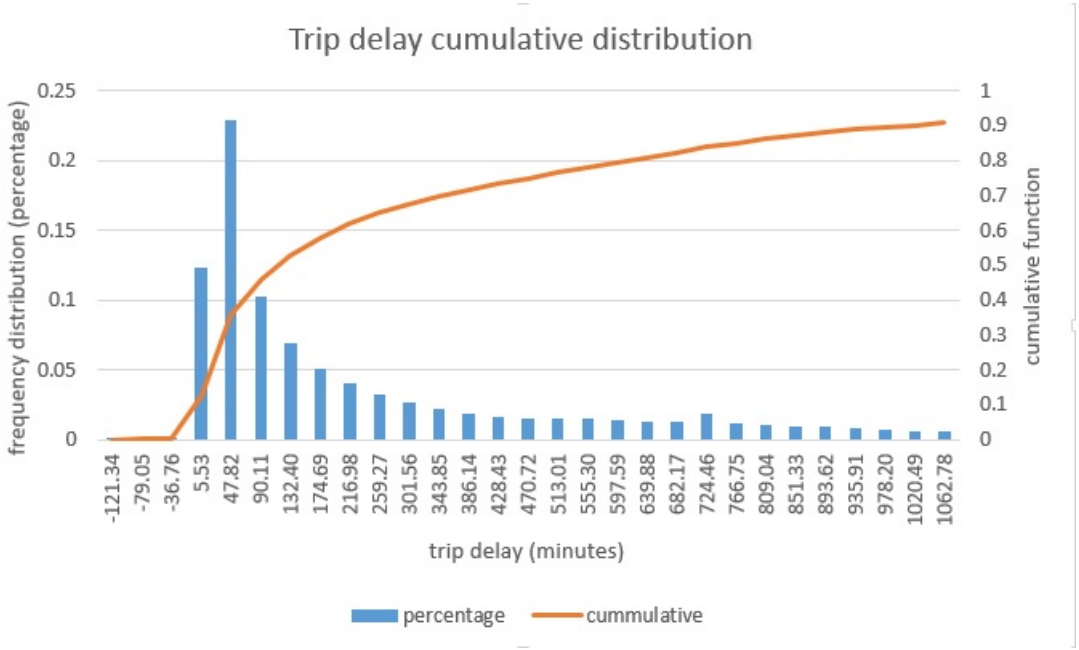


Figure 3.7: Trip delay probability distribution presents a heavy tail on the right.

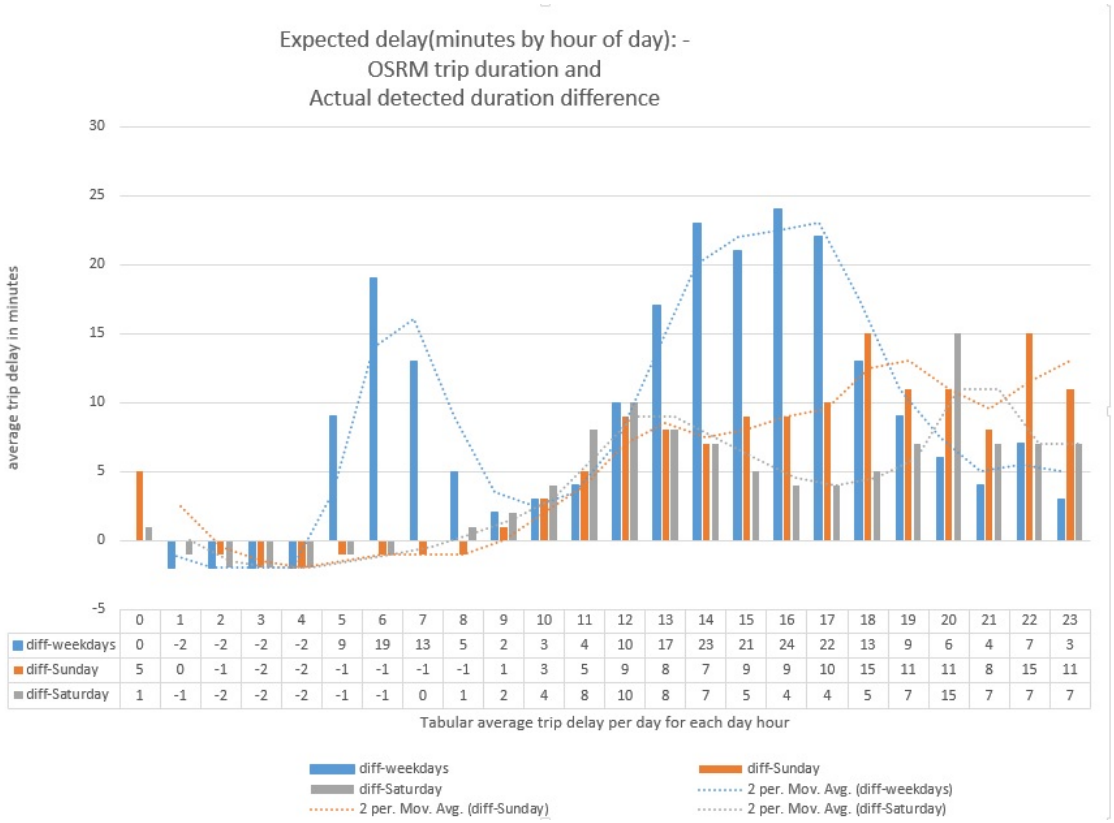
OSRM. The derived route duration does not account for delays. The difference between the actual trip duration retrieved from observed departures and arrivals per user and the OSRM derived trip duration was considered to be the global trip delay. After computing delays for each trip per user, aggregate statistics were collated to describe typical delays at different hours both in weekdays and weekends. Trip delay difference is evident even between Saturdays and Sundays but is highly regular for weekdays as seen in Figure 3.8.

A peculiar observation is the negative trip delays. This can be accounted for by actual trips that were faster than expected and estimated by OSRM. Such negative trip delays were observed during the night when people tend to arrive earlier due to almost in-existent traffic. Average trip delay peaks happen between 6:00 a.m. and 7:00 a.m. and 4:00 p.m. and 5:00 pm. for every weekday. Saturdays and Sundays peak trip delays are observed later in the day where usually during weekdays average trip delays are smaller. This can be attributed to the fact that people go out later during the day on these two days. Also, it is clearly noticeable that, for Saturdays and Sundays, only one distinct peak can be seen in the distribution, and average trip delays per hour are much lower in general.

Trip delay data had to be further investigated to remove outliers and data that was not suited for the traffic flow count and the machine learning model had to be filtered out. The data model fitted a heavy tailed distribution as seen in Figure 3.7. Data was skewed to the right because of long trip delays attributed to pauses in trips that are likely caused by intermediate location visits between the main areas of activity. Similarly, trip delays of less than -5 minutes[9] were mainly attributed to sudden location displacement caused by cell tower switching (see section 2.4 in chapter 2). Cut-off points were set to -5 minutes and 45 minutes for the lower and upper bounds respectively. Consequently, 50% of the data was maintained.

In OSM a route consists of steps and these in turn contain manoeuvres. Manoeuvers encapsulate geographical coordinates data and duration property after

[9]The negative trip delays are caused by trips with duration less than the one retrieved from the OSRM



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
diff-weekdays	0	-2	-2	-2	-2	9	19	13	5	2	3	4	10	17	23	21	24	22	13	9	6	4	7	3
diff-Sunday	5	0	-1	-2	-2	-1	-1	-1	-1	1	3	5	9	8	7	9	9	10	15	11	11	8	15	11
diff-Saturday	1	-1	-2	-2	-2	-1	-1	0	1	2	4	8	10	8	7	5	4	4	5	7	15	7	7	7

Figure 3.8: Average trip delay patterns are different between weekdays and weekends. Peaking of average trip delay on weekdays happen between 6:00 a.m and 7:00 a.m. and between 4:00 p.m. and 5:00 p.m. Peaks for weekend days happen later in the day.

which the driving decision should be taken. The manoeuvres' timestamp was computed by accumulating the duration of the previous steps and adding the final total offset to the trip departure timestamp. A new dataset was created with records including previous data structure and steps' information having the timestamp and the coordinates. Therefore, the new dataset, in addition to the coordinates of mobile users at their origin and their destination had trip geolocation information in the form of trip steps.

Algorithm 1 Experimental Overview

A. Filter Data:
    data_mobile_usage ← filter data usage records from raw *dataset*

B. Main activity locations clustering:
    data_users_clusters ← run DBSCAN on *data_mobile_usage*
    data_users_top_2_clusters ← filter top two user clusters from *data_user_clusters*
    data_users_top_2_clusters_geo_data_points ← get *data_users_top_2_clusters* left spatial join *data_mobile_usage*

C. OD Matrix generation:
    data_users_top_2_clusters_geo_data_points_sorted ← sort by user and datetime *data_users_top_2_clusters_geo_data_points*
    for each user group *ug* in *data_users_top_2_clusters_geo_data_points_sorted* **do**
        for each user record *ur* in *ug* **do**
            if ur_t cluster id $\neq$ ur_{t-1} cluster id (where t is timestamp) **then**
                ur_{t-1} departure flag ← true
                ur_{t-1} destination coordinates ← ur_t location coordinates
                ur_t actual trip duration ← ur_{t-1} timestamp - ur_t timestamp
    data_users_departures_arrivals ← store final resulting dataset

D. OD Matrix based trip generation:
    for each user departure arrival record *udar* in *data_user_departures_arrivals* **do**
        if *udar departure flag* = true **then**
            udar route ← derive OSRM route from *udar origin,destination coordinates*
            udar OSM route duration ← derive OSRM route duration from 1.5em*udar origin, destination coordinates*
            udar trip delay ← udar actual trip duration - udar OSM route duration
    data_users_trips ← store final resulting dataset
    data_users_trips_steps ← new empty dataset
    for each user trip record *utr* in *data_users_trips* **do**
        for each user trip step *uts* in *utr route* **do**
            data_users_trips_steps ← add new record with step coordinates and timestamp details

E. Traffic flow spatial binning:
    data_bin_traffic_flow_time_series ← count traffic flow group by bin id and step timestamp from *data_users_trips_steps*

F. Traffic flow prediction:

data_distinct_time_window_end ← get distinct time window ends from *data_bin_traffic_flow_time_series*

data_distinct_bin_ids ← get distinct bin ids from *data_bin_traffic_flow_time_series*

data_distinct_time_window_ends_bin_ids ← *data_distinct_time_window_end* cross join *data_distinct_bind_ids*

data_sparse_bin_count_time_series ← *data_distinct_time_window_end_bind_ids* left join *data_bin_traffic_flow_time_series*

data_time_windows_bin_count ← two dimensional pivot on *data_sparse_bin_count_time_series* by bin_ids

data_bin_count ← *data_time_windows_bin_count*

sample_locations ← *location_array[a,b,c,d]*

window_frames ← *window_frames_array[30 min,60 min,180 min,1 day]*

for each each bin for location *bin-loc* in *sample_locations* **do**

  for each global prediction at *t-ahead* time ahead in *window_frames* **do**

    for each each record with bin counts *bin-count-record* for time *t* in *data_bin_count* **do**

      t-ahead-bin-loc-count ← get bin count for *bin-loc* at time *t-ahead*

      bin-count-record-with-label ← attach *t-ahead-bin-loc-count* to *bin-count-record*

data_labelled_points ← store final resulting dataset

data_labelled_points_reduced ← take first 1000 components of PCA dimensionality reduction of *data_labelled_points*

data_training ← split *data_labelled_points_reduced* and get 60% of data

data_testing ← split *data_labelled_points_reduced* and get 40% of data

multilayer_perceptron_classifier_model ← fit model on *data_training*

multilayer_perceptron_classifier_prediction_result ← run model on *data_testing*

report result metrics

3.6 Traffic flow aggregation through spatial binning

To get aggregate statistics on traffic distribution, hadoop spatial binning was used as proposed in Eldawy et al.
[27]. Spatial Hadoop was used due to its highly efficient processing of geolocation data because it uses MapReduce[10] and a 2-level spatial index. MapReduce executes tasks with a level of parallelism and computation is distributed. Spatial Hadoop uses a special algorithm to partition data in Hadoop and maintains a spatial index for fast querying and fast spatial joins [27].

A Hive user defined function (UDF) from esri (esri is the company that owns the ArcGIS solution) is used within the Hive query language (HQL) syntax to interact with Hadoop and count traffic flow by spatial bin[11]. A spatial bin is a computational geometry that can be used to numerically describe features in a specific region. In our case we used 0.0005 degrees bins to count traffic flow 'steps' derived from OSM routes. 0.0005 degrees bins approximately equate to 50 by 50 metres bins. We are stating that dimensions are not precise when computing the geometrical bin because dimensions are not strictly universal and vary according to map position. These tend to be more of an elongated rectangle near the poles and squarish near the equator. This happens because latitudes get narrower for bins near the poles due to the fact that the earth is not a perfect sphere but an oblate spheroid[12]. Not withstanding this, the bins in the spatial area under investigation are of the same size since Malta does not cover a wide area. We chose $50m^2$ spatial bins to aggregate traffic flow data. In this way we do not have too much wide geometries that can aggregate traffic coming from two roads. Bins with sizes that are less than $50m^2$ make aggregations less meaningful since aggregation is more

[10]https://hortonworks.com/apache/mapreduce (accessed January 20, 2018)
[11]https://github.com/Esri/spatial-framework-for-hadoop (accessed December 10, 2017)
[12]http://www.longitudestore.com/how-big-is-one-gps-degree.html (accessed December 12, 2017)

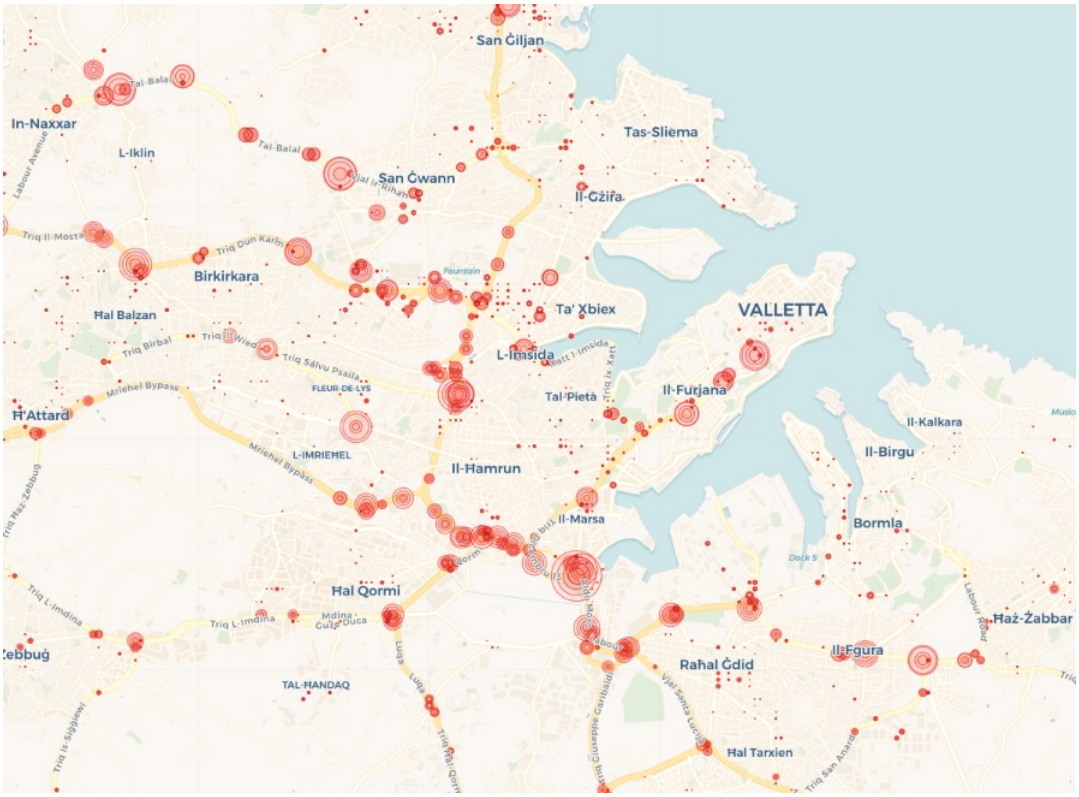


Figure 3.9: Traffic flow count through spatial and temporal binning.

near to data points rather than grouping polygons.

The centroid for each bin was calculated in order to attain the central coordinates of the polygon delineating the bin. Aggregation was not only done spatially but also temporally with intervals of 5 minutes each. This choice of interval's size is quite subjective in nature but it has been decided that it is both granular enough and not too wide to describe traffic flow temporally. The choice of time ahead window sizes would affect the prediction results. A larger time window would not permit finer prediction in a time series. For example a non-sliding 10 minute time window would allow predictions 10 minutes ahead and a 20 minute sliding window would allow for predictions 20 minutes ahead and so on.

We did not use a sliding window since this would have resulted in less data points for training since training of neural networks would be more costly in terms of computation and would have made analysis more time consuming and complex.

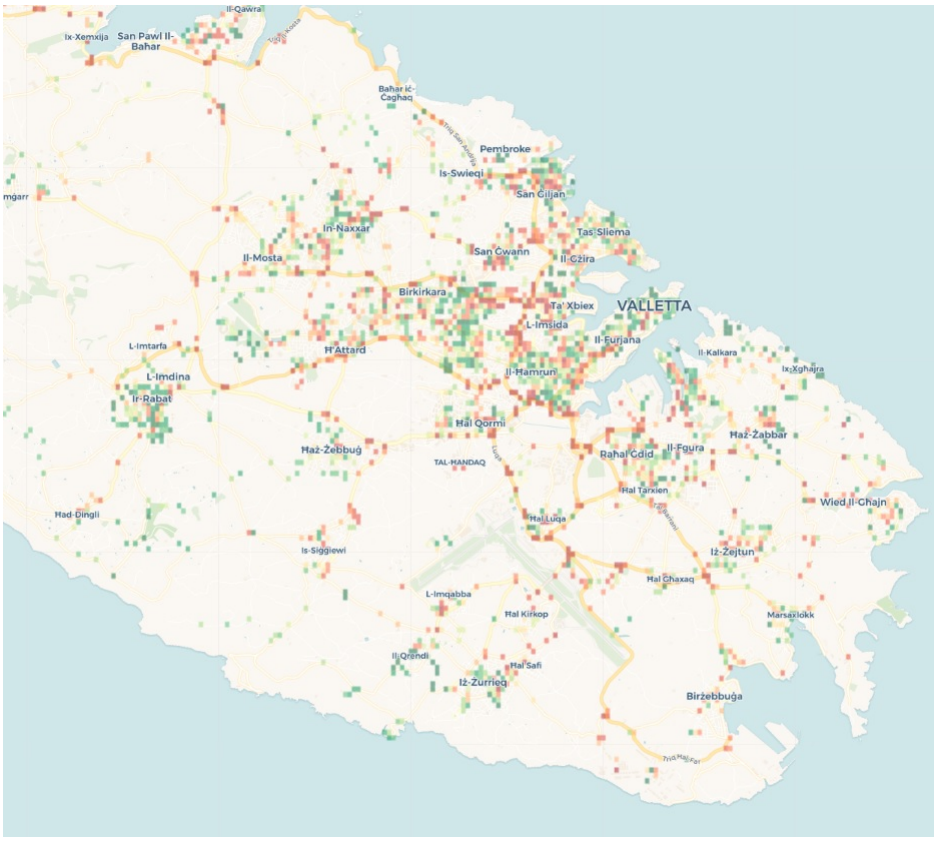


Figure 3.10: Traffic flow count categorization through a colour coding. Traffic flow count intensity is represented with a colour scheme ranging from dark green (low recorded traffic) to dark red (heavy traffic). The range stretches on 11 quantiles.

Having a sliding window on the other hand makes it more flexible to decouple the averaging window size from the time ahead distance. The prediction time ahead parameter would not need to be a multiple of the averaging window.

Visual tools such as CartoDB[13] where used to illustrate aggregation of traffic flow count in spatial bins. Figure 3.9 shows which areas attract most traffic in Malta. Locations such as the Santa Venera tunnels and Southern Harbour area around the four lane roads in Marsa have bigger distinctive spherical markers indicating that these roads are very busy. The tool allows to select specific date and time to analyse traffic temporally. A similar visualization (figure 3.9) depicts average traffic intensity across the whole month under investigation. CartoDB allowed us to configure pop ups that can display relative information to the bubble such as location coordinates and average month traffic.

Another illustration (see figure 3.10) uses a colour scheme in a categorical manner to display the intensity of traffic flow. This visualization technique makes it easier to categorize traffic flow count than the technique used to display traffic in figure 3.9.

CartoDB was a fundamental tool to analyse how traffic flow count changed with time and where. Methods that have been devised in this research to aggregate traffic flow on the road network, can effectively have the generated results illustrated temporally by moving a time window slider in the CartoDB UI. For instance the traffic at 7:00 a.m shown in Figure 3.13 is busier than the traffic at 6:00 a.m. in Figure 3.12. Figure 3.14 is a zoomed in image of the figure shown in 3.13.

3.7 Traffic flow modelling and prediction

The hypothesis that traffic flow in all areas is directly correlated to how traffic in a specific given area will be in the immediate future determined how the prediction model was structured. More specifically traffic flow at any particular bin b_i at

[13]https://carto.com (accessed May 10, 2018)

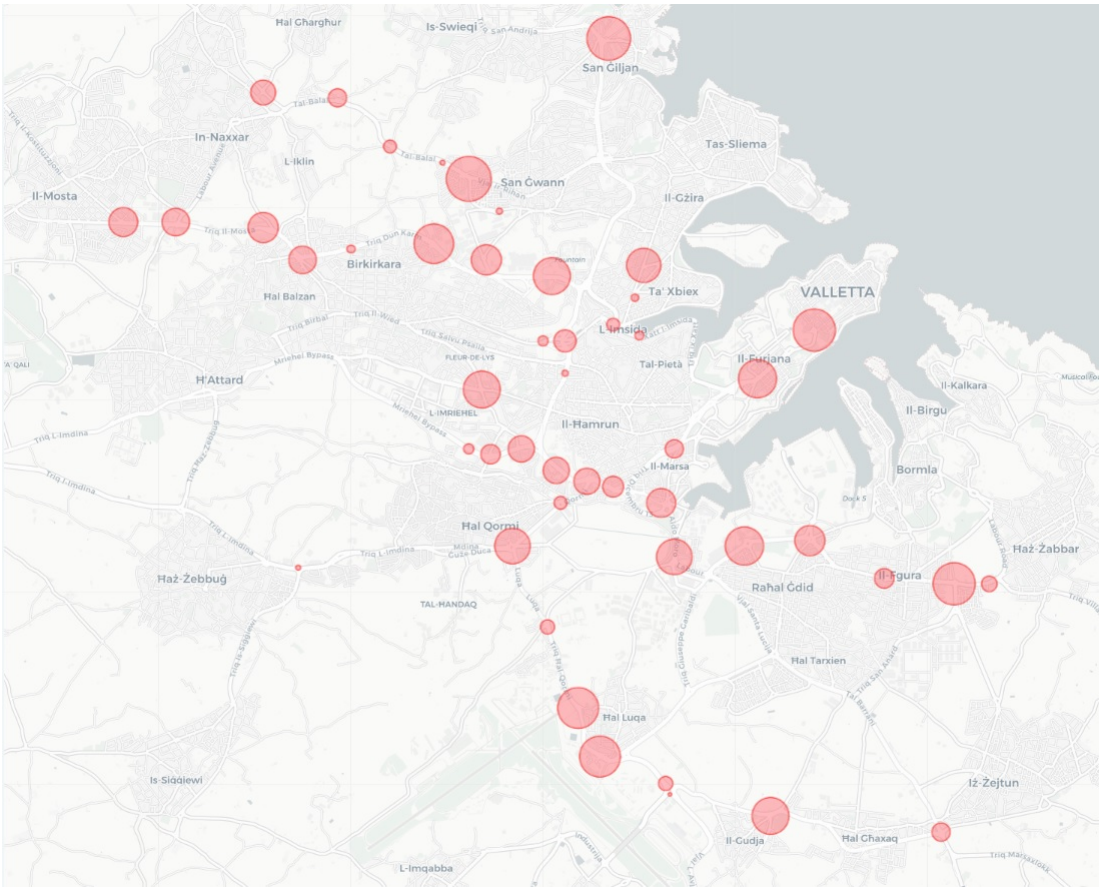


Figure 3.11: Traffic flow count through spatial and temporal binning.

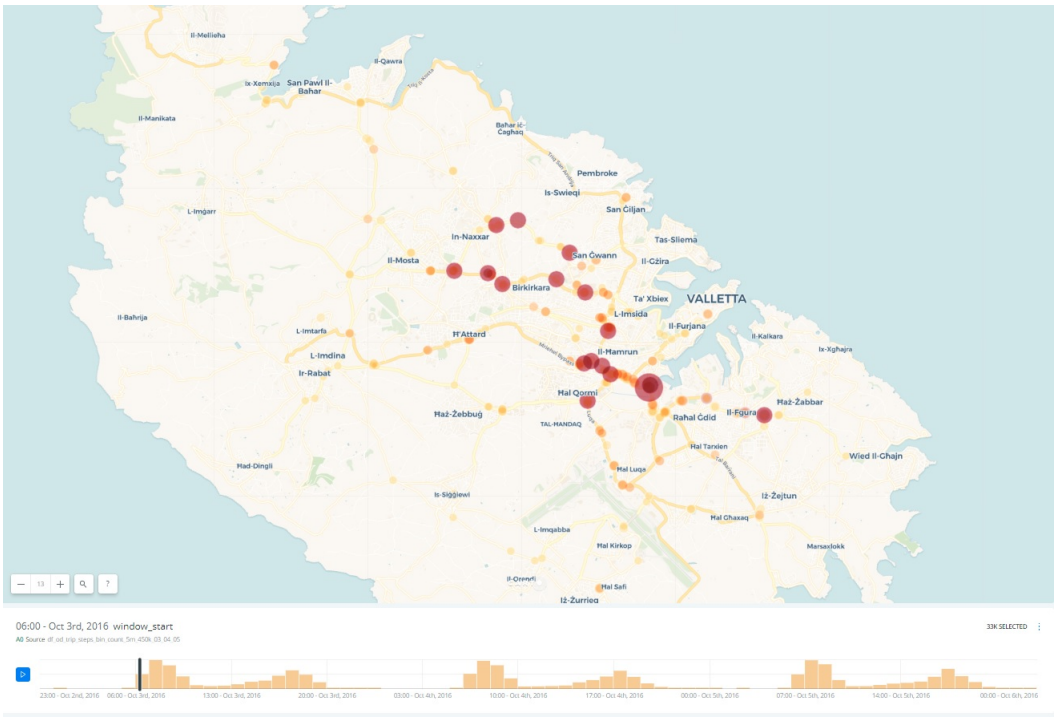


Figure 3.12: Traffic flow count at 6:00 a.m. illustrated through CartoDB temporal mapping

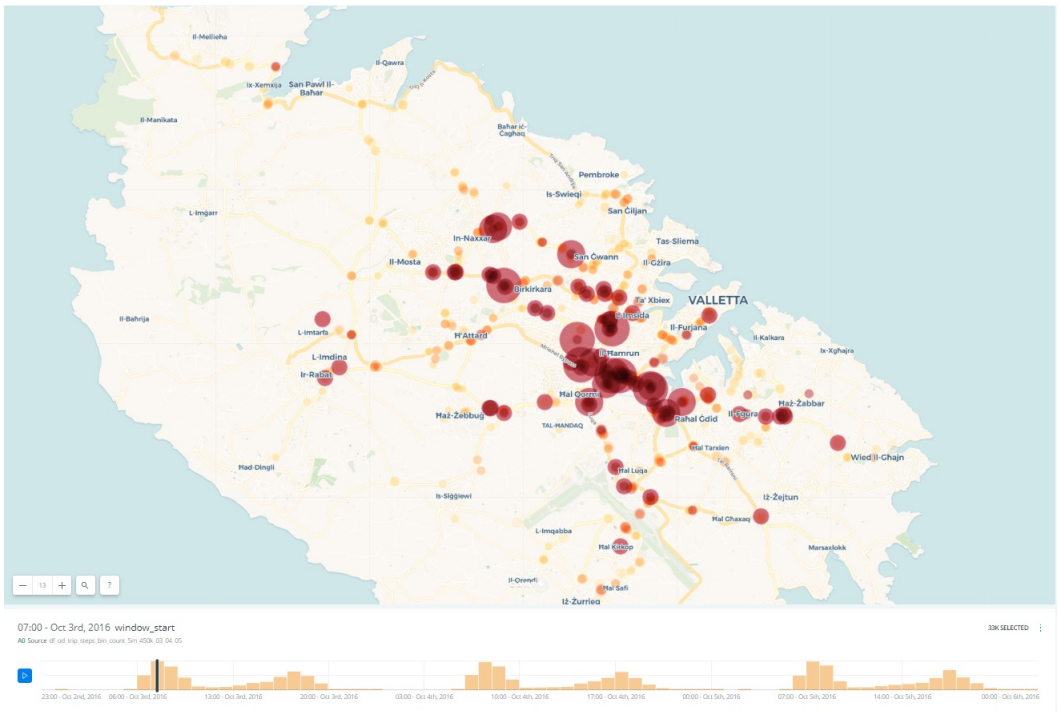


Figure 3.13: Traffic flow count at 7:00 a.m. illustrated through CartoDB temporal mapping.

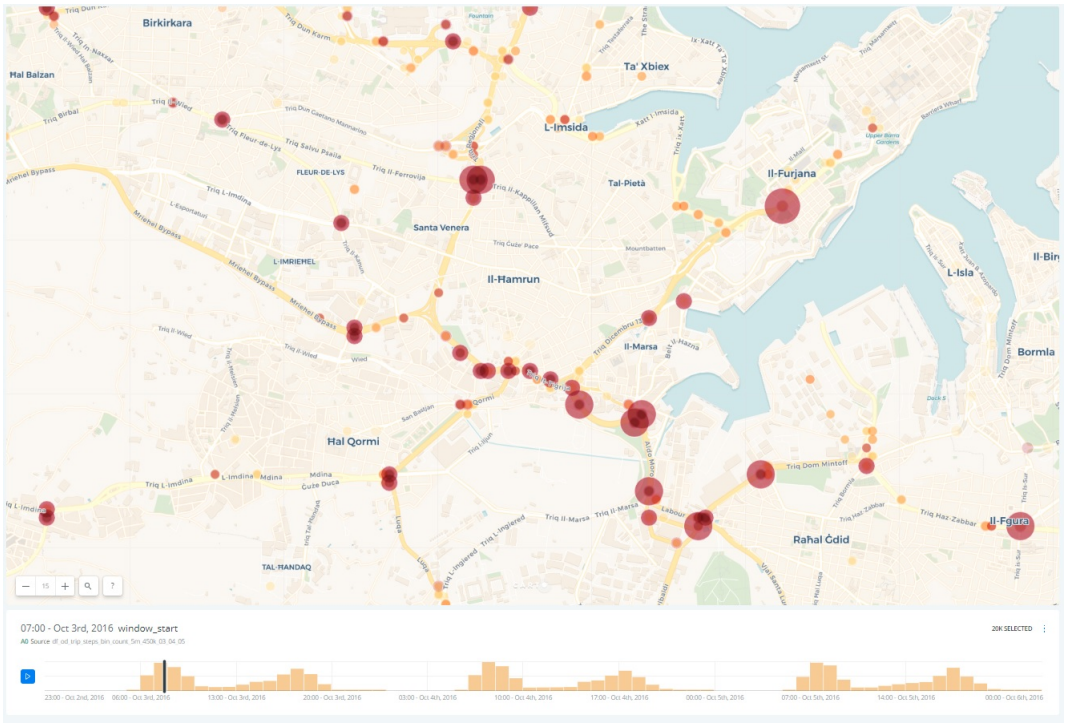


Figure 3.14: Traffic flow count at 7:00 a.m. illustrated through CartoDB temporal mapping (zoomed in).

time t influences traffic at bin b_j at time $t + n$. The traffic intensity at any given location is a function of other traffic flow counts from other preceeding bins. The computation of traffic in a location that is based on past traffic cannot be achieved by deriving a mathematical function. A process of artificial intelligence learning is needed in order to build a model. Black box predictive modelling was obtained by training a neural network. Neural networks are known for their ability to generalize and to be able to learn and handle unexpected inputs [39]. There were various steps needed to fit a neural network model which was then used to predict the traffic flow. The aim was to select a sample of locations and predict their traffic. In the following subsections, it will be explained how data was processed prior to training of the model, how the model hyperparameters were chosen, how the model was eventually trained and then how the model was validated.

3.7.1 Preprocessing data for the prediction model

The first data mining exercise was to build a dataset where each record contains all Malta traffic flow count for every 5 minutes for the month of October 2016 (see Table 3.4). This dataset was then ordered by time. This dataset was to be derived from the generated dataset in Subsection 3.5.2. This dataset consisted of aggregated traffic flow for each spatial geolocation in a time series with 5 minutes bins. The final resulting dataset from preprocessing was used to train and validate the artificial neural network (ANN). The features selected to build the model are the traffic flow counts at every location geofenced by a spatial bin. The aggregation process of traffic counts within specific spatial bins was explained in Section 3.6.

We used two configurations for traffic classification. Experimentation with the ANN training and validation was done primarily with four labels. Experimentation was also done with eight labels to test how it would perform in comparison. Classification of traffic was done using 4 labels since traffic is labelled similarly in available traffic applications such as Google Maps Traffic[14] and Tomtom navigation software[15]. These labels describe different classes of traffic speed ranging from slow to fast. In Google Maps traffic speed is visualized through colour coding. Green for example is used to indicate slow traffic speeds and dark red is used to notify about heavy traffic congestion.

In our research Label 1 would indicate very low traffic flow count per minute and Label 4 would represent high traffic flow count per minute. The assigning of the labels consisted first of classifying the traffic count of the bin location for every 5 minute window for which prediction modelling was carried out. Then the resulting classification was assigned to the $t - 5n$ record where n is an integer denoting the number of fixed time intervals to predict ahead. Therefore after this operation, we introduce another column to Table 3.5 with future classification of traffic count for a given bin location.

[14]https://www.google.com/maps (accessed May 20, 2018)
[15]https://mydrive.tomtom.com (accessed May 20, 2018)

Four datasets were prepared for training. These datasets were differentiated by the label assigned. The classification label for each dataset record was retrieved from time $t + 5n$ ahead through window analytical functions for n with values of 3, 6, 12 and 288. Since we used a 5 minute binning that sums the traffic flow count, 3,6,12 and 288 time ahead windows would represent 15 minutes, 30 minutes, 1 hour and 1 day ahead predictions respectively.

bin id	longitude	latitude	time window start	time window end	traffic flow count
4611467925420319675	14.4320000001052	35.905499999918	2016-10-01T00:00:00.000Z	2016-10-01T00:05:00.000Z	12
4611467846458306816	14.489500000100501	35.918499999953397	2016-10-01T00:00:00.000Z	2016-10-01T00:05:00.000Z	2
4611468259490374713	14.506000000011101	35.850499999983199	2016-10-01T00:00:00.000Z	2016-10-01T00:05:00.000Z	3
4611468311119383155	14.485500000027001	35.841999999986903	2016-10-01T00:00:00.000Z	2016-10-01T00:05:00.000Z	2

Table 3.4: A sample of traffic flow count by bin for every 5 minute window.

The resulting dataset had traffic flow count for each bin with 5 minute temporal resolution. Traffic counts of zero were not yet present before preprocessing. The main features of this dataset included *bin id*, traffic flow count and time window start and end timestamps.

Further processing was however needed to generate a dataset with records that give a snapshot of all traffic count for Malta for every 5 minutes. First all distinct bin ids where extracted and these amounted to 4134. All the possible time windows of 5 minutes in the month of October were generated, and these amounted to 8928. By performing a Cartesian product between all time series values and all possible bin ids a new dataset with all possible bin id and time window combinations was created. A left join between the original aggregated traffic flow with the latter produced dataset resulted in a new dataset with records that comprehensively describe traffic flow for every 5 minute window for the whole region under study. This data structure was not suitable to be programmatically inputted to the neural network training and further reorganization was necessary. A dataset with data record format where each row contains all traffic flow for all Malta was needed. The columns would be the bin ids that describe all the traffic in all areas. The rows would contain traffic flow count values at a particular 5 minute interval for all these bin ids. To achieve this, a two dimensional pivot was used to transform

data and traffic per bin. In the resulting dataset, the traffic count per bin is stored column-wise. The pivot operation based on the data from 3.4 resulted in the data that is shown in 3.5.

time window timestamp	**bin id 1**	**bin id 2**	**bin id 3**	**...**	**bin id 4134**
2016-10-01T00:00:00.000Z	0	0	2	...	0
2016-10-01T00:05:00.000Z	0	1	1	...	1
2016-10-01T00:10:00.000Z	0	0	0	...	0
2016-10-01T00:15:00.000Z	1	0	0	...	0

Table 3.5: Sparse traffic flow matrix

After all the feature data were organized in a format that could be fed to the model, a label for each data row was assigned. An ANN is a supervised machine learning type which requires output that can be mapped from input data during the training phase. The output in our case is classification of the level of traffic flow count for a sample location (spatial bin) for which we need to determine the traffic at time $t + 5n$. The output label was not chosen to be the traffic count but the logarithmic function:

$$y = \lfloor log(x + 1)/log(max_x/n + 1) * b \rfloor + 1 \tag{3.1}$$

Where x is the actual traffic flow count, y is the final classification label, n is the step size coefficient (the higher it is the smaller the steps)and b is the number of classification levels (bins).

If a step function with equally spaced intervals was used to classify traffic count, the function label outputs would almost all fall under the first class without meaningful differentiation (see Figure 3.15). The skewness towards low counts of traffic is highly decreased with logarithmic binning. The use of logarithmic 'step' function, defined in Equation 3.1, squeezes indicators in the low traffic flow label bin and widens the range for high level traffic. Note how in Figure 3.17 the logarithmic step function with n=2.36 manages to classify low traffic counts that happen to have high frequency more evenly than step functions with smaller n values. In our experimentation traffic count labelling was done with step size coefficient of 2.36

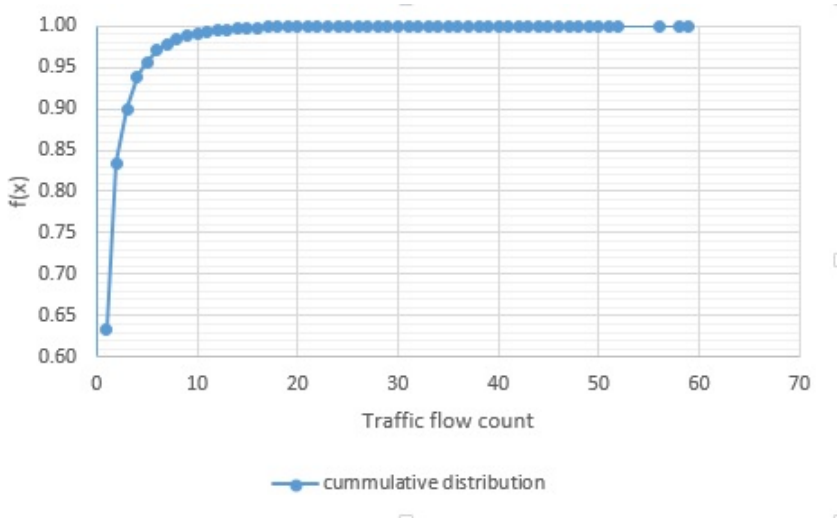


Figure 3.15: Traffic count cumulative distribution for a chosen location.

which proved to give better results for prediction evaluation. Classification of what is easy, moderate or heavy traffic flow is rather subjective. The coefficient of 2.36 value was chosen by changing the coefficient value and visually inspect plots like the one shown in Figure 3.17 to decide that the binning represents proportionally the cumulative distribution of traffic flow count.

From Figure 3.17 it can be seen that traffic flow count that ranges from 0 to 1 is classified with Label 1 (zero or minimal traffic flow count), traffic flow that ranges from 2 to 3 is classified with Label 2, traffic flow count that ranges from 4 to 10 is classified with Label 3 and traffic flow count that is equal or greater than 11 is classified with bigger labels. More than 60% of traffic flow count records are classified with Label 1 and more than 20% are classified as Label 2 traffic flow.

3.7.2 Dimensionality Reduction

The dataset acquired from the original mobile data usage dataset is huge. The features that describe the data amounted to 4134 as already mentioned in Subsection 3.7.1. The planned computational complexity depended on how quickly the model converges. However for each iteration carried out to reduce the cost and undergo gradient descent the magnitude of the computations to be performed depended on

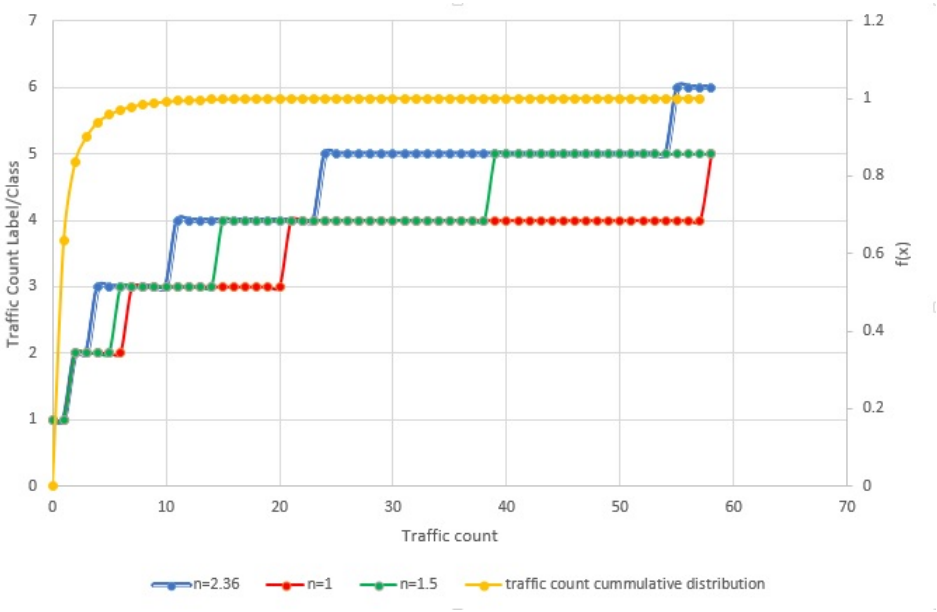


Figure 3.16: Traffic count logarithmic step function.

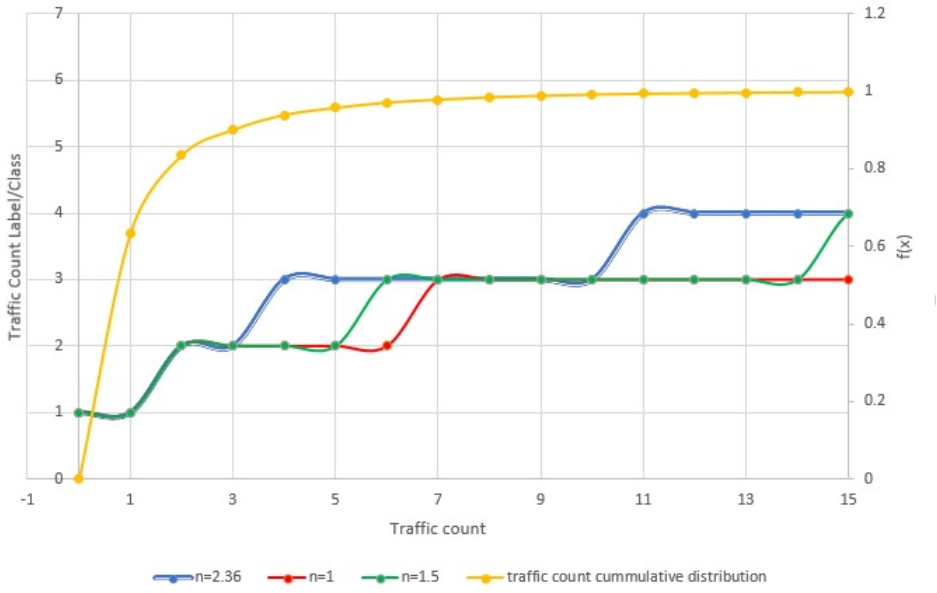


Figure 3.17: Traffic count logarithmic step function showing the lower traffic counts mapping.

the number of training examples multiplied by the number of features multiplied by in turn by the number of neurons in each hidden layer. The number of hidden layers was chosen depending on how much complex the problem is but tends to fall victim of overfitting the model if too many layers are inserted in architecture. As mentioned in [40] ANN design parameters such as number of neurons and number of hidden layers need a trial and error approach to get an architecture that yields better results. Therefore it was important to optimize computation times in order that experimentation that leads to an optimal architecture is less time consuming. Also, the final model is simpler and more practical in terms of getting a prediction after an acceptable amount of time.

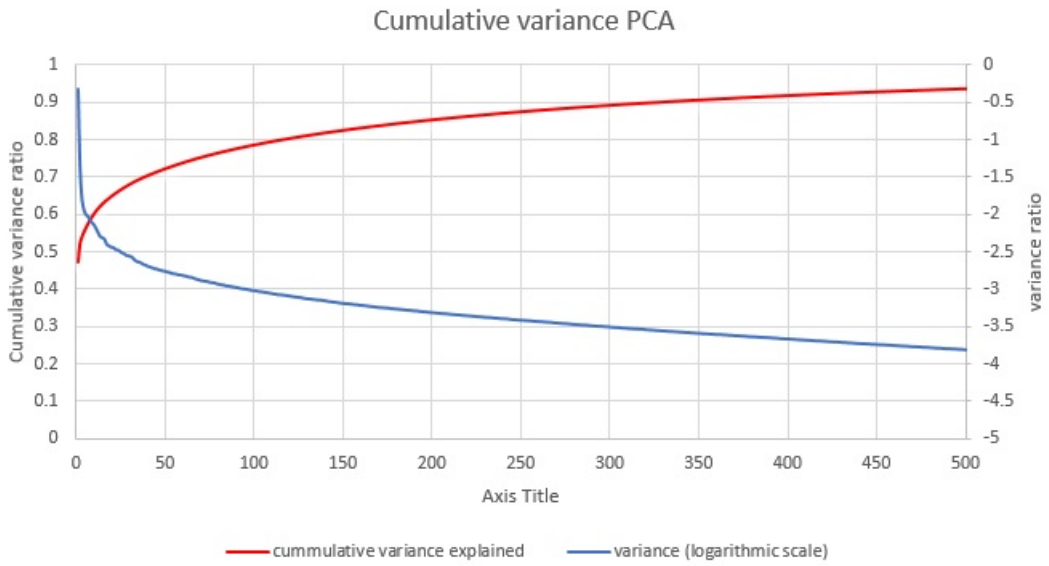


Figure 3.18: First components contain a higher percentage of the variance. First 324 components explain 90% of the variance

One way how to lower computation time was to reduce the number of features. This entails mapping an n-dimensional space to a smaller dimensional space which reduces the number of features in the process. Raschka [41] states that by reducing dimensionality in data there is less risk of overfitting and thus the model can generalize better to testing data. In [42] experimentation is done by keeping 95% and 99% of the total variance. We chose to keep 324 components that explained

90% of the total variance. By trial and error it was found that similar results are attained by using 90% and 98% of the variance. Original data features showed to be highly correlated, so a high degree of compression was possible through PCA. As explained in the Formula below 3.2,

$$\min_{k} \left\{ k : \frac{\Sigma_{i=1}^{k} \lambda_i}{\Sigma_{i=1}^{n} \lambda_i} \geq r \right\} \tag{3.2}$$

we selected k to be 324 so that the preserved variance ratio r is 90%. In Equation 3.2 eigenvalues λ_i were ordered in decreasing variance. n represents the original dimensionality of the reduced dataset. In Figure 3.18, it can be observed that the first 324 components explain more than 90% of the data.

3.7.3 Prediction through Multilayer Perceptron Classifier

The next step in the data processing pipeline consisted in predicting traffic from a stipulated time ahead for a given location point. This prediction had to be based on data that was harvested some time before. All of the original datasets had records with timestamps set in the past, so we simulated prediction of traffic flow by trying to forecast traffic at a certain point in time which is ahead of a given timestamp. Evaluation was carried out with a different number of first PCA components, prediction multi-steps ahead and number of possible classes.

As already explained in Subsection 3.7.1, the multi-step time series prediction was evaluated with a variable amount of steps ahead. Each step was already defined to be 5 minutes long. The experimentation was done with 3,6,12 and 288 steps that reflect 15 minutes, 30 minutes, 1 hour and 1 day. These particular prediction time intervals were selected because practically an individual would need to know traffic in certain locations just before he leaves home. Traffic information 3 hours in advance would prove to be irrelevant for a commuter that just leaves home. This applies especially for Malta based trips, where distances are relatively short and surely any journey is less than 3 hours. Even transport authorities might not

find 3 hours beforehand information useful for management purposes. Individual users leaving at 7.00 am in the morning would contribute not information for traffic status at 10.00 am where traffic flow would have eased by then. Therefore we opted to analyse and predict the impact of traffic at 15 minutes, 30 minutes, 1 hour and 1 day before respectively.

One of the first decisions was to choose what type of approach for machine learning to take in order to build a model. The problem at hand was complex, both because of the number of features and the relation they have with each other. A Multilayer Perceptron Classifier (MLPC) is a highly non-linear model that can adapt to problems with high complexity. An MLPC is a specific form of ANN in which perceptrons are feed forward neurons and are interconnected with weights. Layers with a different number of perceptrons define the architecture of the MLP. The first layer is the input layer and the last layer is the output layer. Hidden layers that are optionally inserted in between the input and output layer apply any function to the previous layer and output to the following layer [40].

The ANN approach is stated to have the universal approximation property which underlines how an ANN of MLPC type can represent any bounded continuous function to a given arbitrary degree of accuracy [43]. However, it is considered to be a black box that is difficult to control and monitor while learning during the training stage. Spark ML MLPC implementation contains intermediate layer neurons that use the logistic function and output nodes that use softmax function[16].

The Spark 2.3.0 implementation that was used makes use of back-propagation to learn the model. It employs the logistic function as an activation function with Limited-memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) to minimize error[17].

The topology choice was determined after carrying out a grid search configured with a set of different hyperparameters. Different configurations for architecture

[16]https://spark.apache.org/docs/latest/ml-classification-regression.html#multilayer-perceptron-classifier (accessed April 29, 2018)

[17]https://dzone.com/articles/deep-learning-via-multilayer-perceptron-classifier (accessed May 10, 2018)

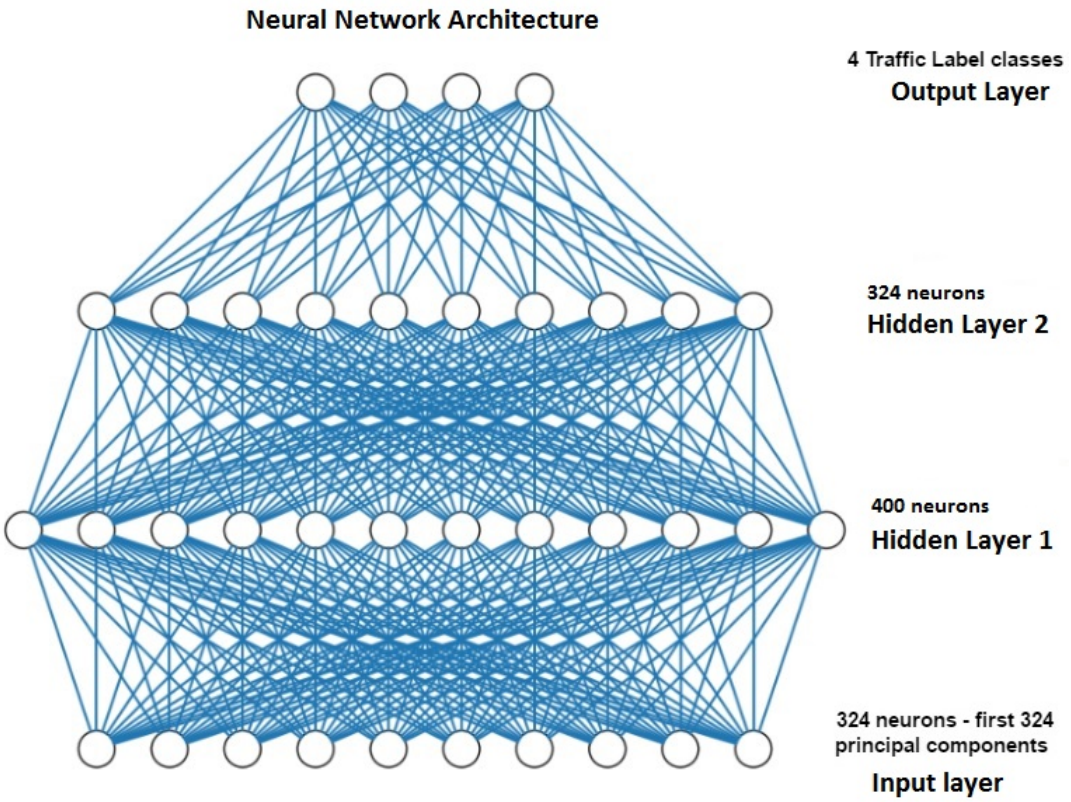


Figure 3.19: Multilayer Perceptron Classifier topology (324,400,324,4) - generated with python matplotlib library

structure were tried out and evaluated until the best performing architecture was chosen. The input and output layers size (number of neurons) are respectively dictated by the number of input PCA extracted features and output classes which are the possible traffic levels indicated by model. Two hidden layers were added to the overall topology. After testing out different possible architectures, the final configuration consisted of 324 neurons for both the input layer and for the second hidden layer, 400 neurons for the first hidden layer and 4 neurons for the output layer which defined the output classes (see Figure 3.19). All architecture layers are fully connected to the successive layer.

The available labelled data points dataset was split into 60% training and 40% testing. The model with maximum iterations parameter set to 200 showed that

after a set of runs it converged consistently. The first phase of setting up the model consisted of the training part where the weights settle to a final value that lead to a minimal error in its classification within the parameter of tolerance set in the configuration. Training outputs a model which is then fitted on the testing data. In the testing phase the prediction efficacy of the model built during training is checked by retrieving certain metrics.

4. Evaluation and Results

In this chapter we discuss the systematic evaluation approach that was adopted to get vehicular traffic related information and to build a prediction model from mobile usage data, several stages were involved. The stages in this machine learning pipeline are highly dependent on previous stages' results, mainly, because each stage feeds its output to the next stage.

Therefore inaccurate results error introduction in early stages would trickle down the pipeline. For example interpretation of error ratio at the prediction stage, which is the last stage, must be primarily done in the context of results observed in preceding stages. If traffic counts are not accurately measured, the training data itself would not lead to predictions that can be put to practical use. As part of the evaluation process that will be described in detail in the following sections we evaluate four main experimental procedures:

1. Average trip counts per hour for weekdays and weekdays
2. Average trip delay per hour for weekdays and weekends
3. Traffic flow count in a selection of locations
4. Traffic count prediction for a selection of locations

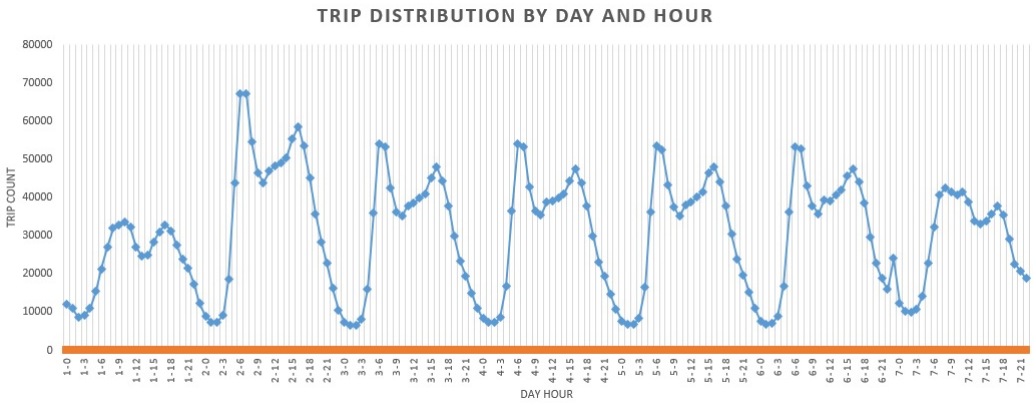


Figure 4.1: OD matrix generated trip count patterns per day week. To note how Saturdays and Sundays which are outlined by the last and first two peaks respectively have unique traffic delay patterns.

4.1 Trip counts per hour evaluation

After determining the users' daily routes between different origins and destinations two sets of important information were derived. These are namely, trip count and trip delay per route.

The former represents the average hourly trips done while the latter represents the average delay for every route or for all trips in general. Distributions for both were further derived by using information such as trip departure and arrival timestamp which is available for every route.

In Figure 4.1, we show that the trip distribution derived from mobile usage EDRs' generated OD Matrix is strikingly similar with the trip distribution as reported in a National Household Travel survey (NHTS) done in 2010 [1] (see Figure 4.2).

As illustrated in figure 4.3, trip count peaks for both reported distributions are observed at 7:00 a.m. and 5:00 p.m. NHTS data was collected on a Wednesday and has no car trip distribution for Saturdays and Sundays. Since figure shown in 4.1 shows that weekdays' OD generated trips count distribution is similar in shape the average was taken on all weekdays of the whole month. However, it should be noted that trip counts on a Monday are distinctively slightly higher than the other

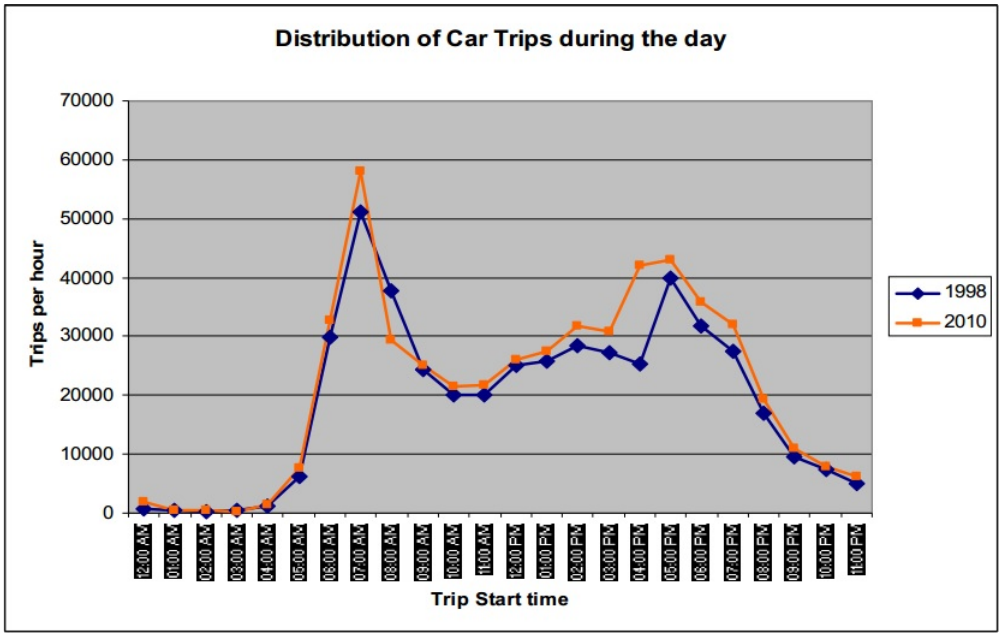


Figure 4.2: Car trip distribution as reported in Transport Malta 2010 National Survey [1].

Linear regression statistics		
Pearson correlation coefficient (r)	**Degrees of freedom**	**p-value**
0.94	22	$1.13628e-11^{*}$

Table 4.1: NHTS and OD trip distributions proved to be highly correlated. $^{*}p < 0.001$

weekdays (see figure 4.1).

The dynamics of the plot show how there is a sudden decrease of trips per hour after 7:00 a.m. that continues till 11:00 a.m. when trips start to rise again to gradually ramp up. The gradient suddenly increases again at 3:00 p.m. The increase and decrease of trip rate around the 5:00 p.m. peak is smoother than then the one observed in the morning peak for both distributions as shown in Figure 4.3).

NHTS [1] reports 11% in car trips from 1998 to 2010 and as shown in Figure 4.2 the distribution again shows resemblance. Both the relationship between HTS datasets gathered in 1998 and 2010 and the relationship of these to the existent OD generated trip dataset demonstrate that trip distribution increases evenly with

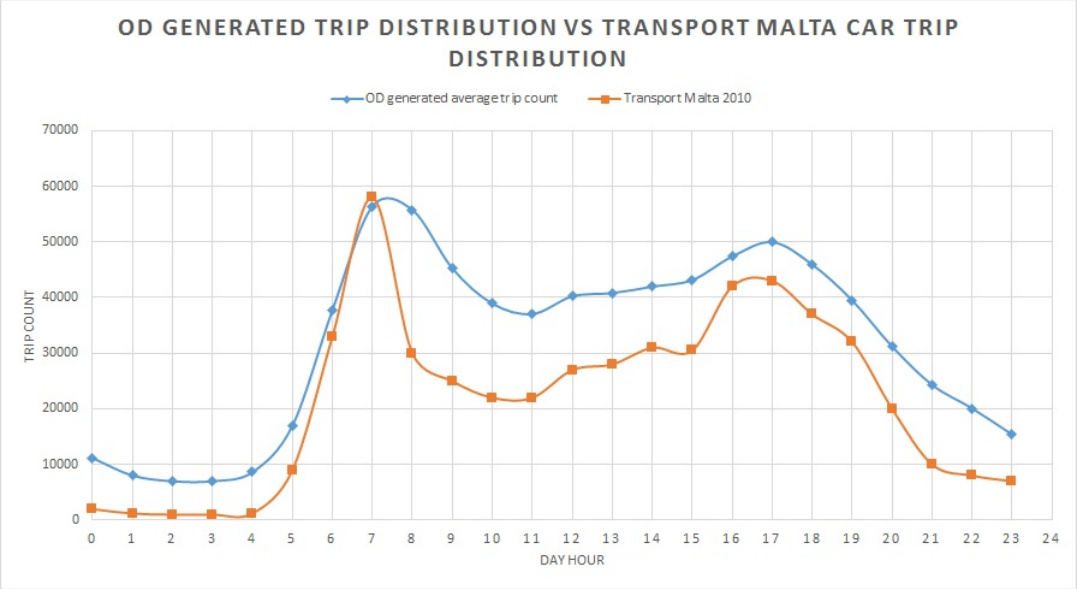


Figure 4.3: Comparison between OD average trip distribution over a month and Transport Malta 2010 survey results ([1])

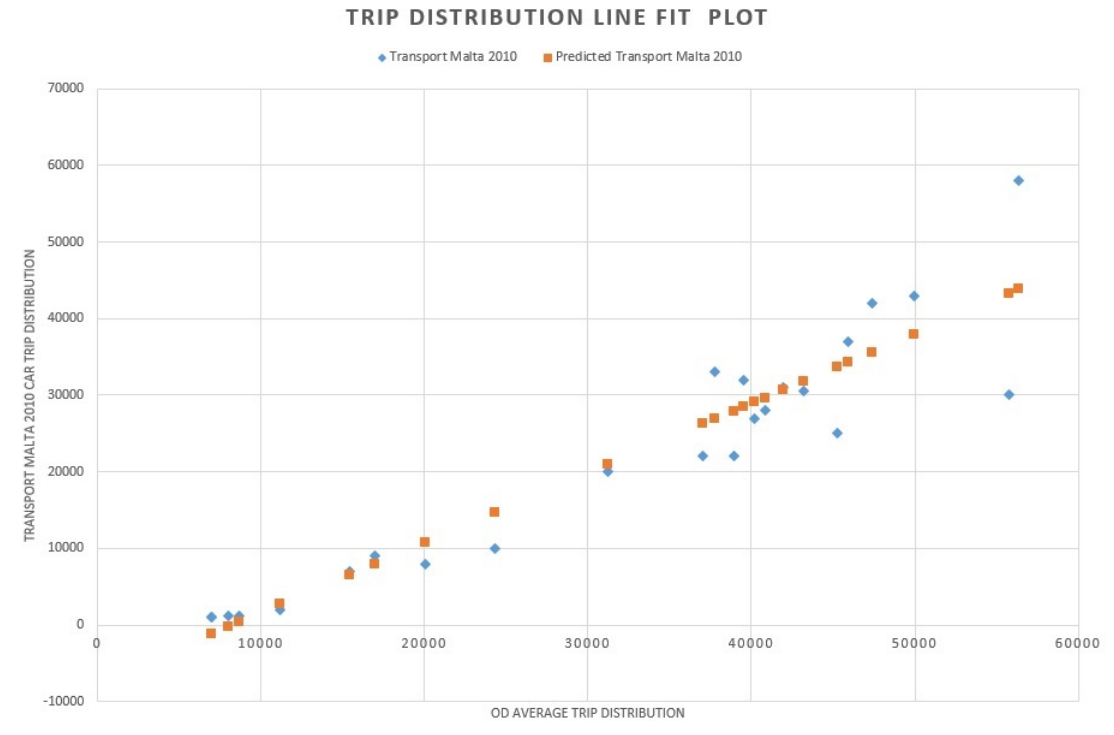


Figure 4.4: Linear relationship between OD average trip distribution over a month and Transport Malta 2010 survey results [1]

a specific scale factor across the hours as years go by.

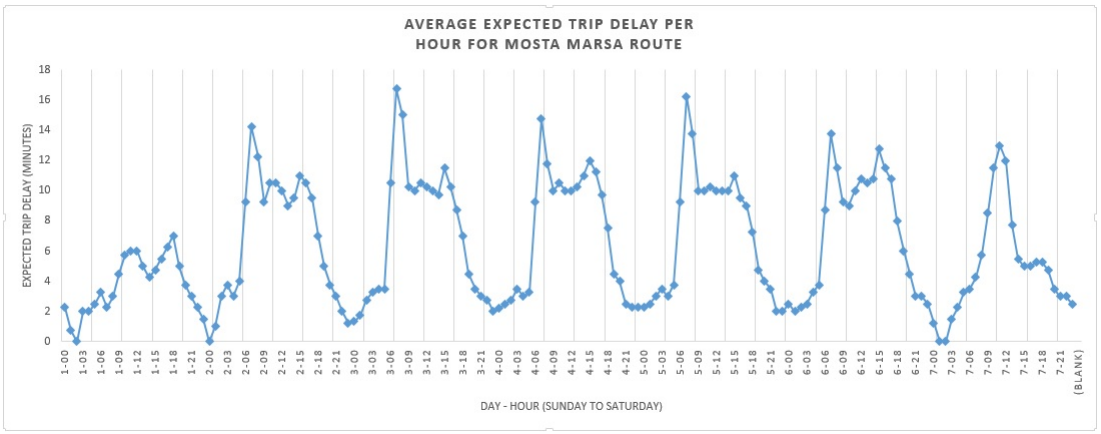


Figure 4.5: Expected trip delay 7 day distribution retrieved with Google distance matrix API for Mosta to Marsa route. Note how Sunday and Saturday delay pattern is different from the one observed for weekdays.

A linear regression model was devised to express the scaling up from trip distribution in 2010 and the one registered in 2016 in this study. A correlation statistical analysis would have sufficed to establish a linear relationship. There is definitely no causality relationship between these two variables. However, a regression model was fitted to the data to express how scaling up of counts can be done from the OD generated one to actual data that is collected through surveys (see Figure 4.4) . Results are reported in Table 4.1 and these show that there is a significant positive relationship between trip distributions.

4.2 Trip average delay per hour evaluation

Evaluation of average global trip delay results computed from OD and OSRM generated trips (OD-OSRM) proved to be challenging, because ground truth data could not be found in literature that considered similar research type and in reports from local transport authorities. In the NHTS [1] in addition to trip count statistics it is mentioned that a detailed matrix with trip information including departure and finish time was compiled. Correspondence with Transport Malta to attain such data or similar information proved to be futile up to the date of completion of this

work.

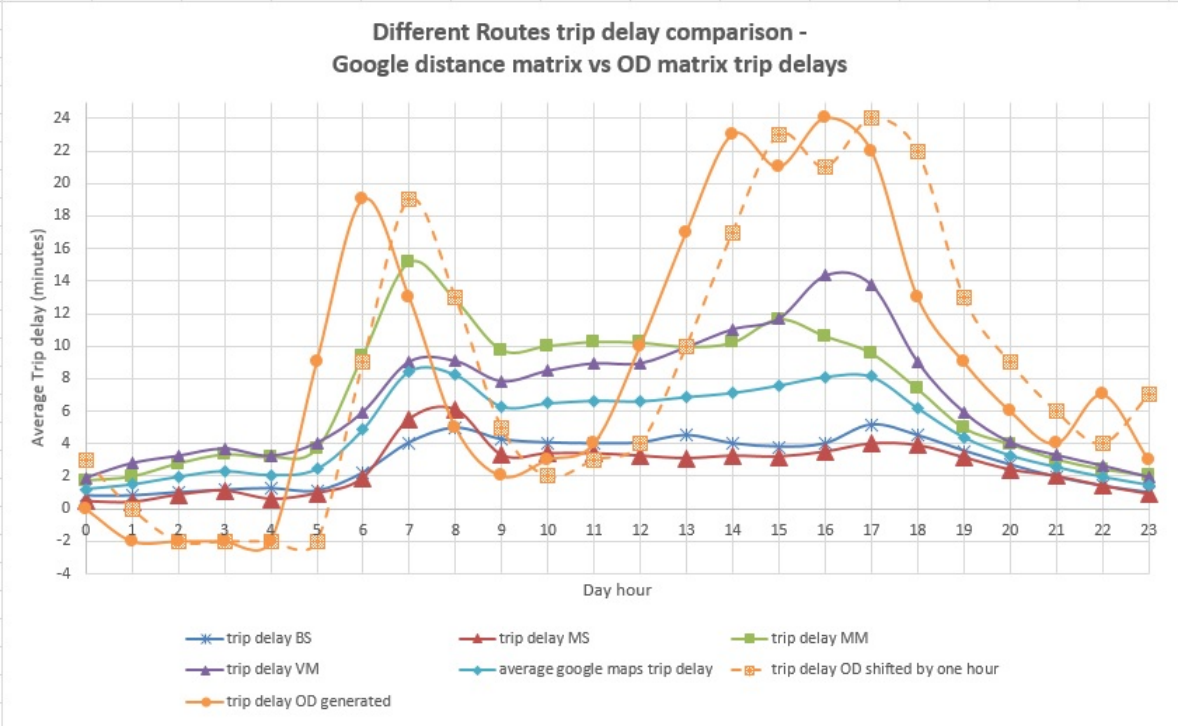


Figure 4.6: DMAPI expected average trip delay comparison with OD-OSRM computed average trip delays for routes: Mosta to Marsa (MM), Mellieha to Swieqi (MS), Birkirkara to Sliema (BS), Valletta to Mgarr (VM)

At the end of April 2018 Google maps made available traffic information overlay on its maps. In addition to this Google Cloud distance matrix API (DMAPI) exposed a web service that gives duration and duration in traffic of trips that are defined with origin and destination for Malta as well.

The trip delay model built through the OD matrix is a basic statistical one that gives average trip delay per hour. Google distance metric API gives estimated duration information (with traffic and without) by specific route. In order to compare the estimations of our model with the Google one for the local traffic a data-mining process that scraped a data set of trip delay through the DMAPI web service was carried out. Figures 4.7 and 4.8 show traffic status for the same zone at the same hour for OD-OSRM traffic flow count and Google traffic status. Note the similarity in how traffic hotspots' locations are shown.

Distance Matrix API and OD matrix OSRM generated trips delay correlation	Distance Matrix API					OD - OSRM	
	avg trip delay BS (DMAPI)	*avg trip delay MS (DMAPI)*	avg trip delay MM (DMAPI)	avg trip delay VM (DMAPI)	avg overall trip delay (DMAPI)	avg trip delay OD generated	avg trip delay OD generated shifted
avg trip delay BS (DMAPI)	1						
avg trip delay MS (DMAPI)	0.92	1					
avg trip delay MM (DMAPI)	0.86	0.88	1				
avg trip delay VM (DMAPI)	0.89	0.77	0.85	1			
avg overall trip delay (DMAPI)	0.95	0.91	0.96	0.95	1		
avg trip delay OD generated	**0.60**	**0.50**	**0.61**	**0.78**	**0.69**	1	
avg trip delay OD generated shifted	**0.70**	**0.69**	**0.61**	**0.76**	**0.72**	0.82	1

Table 4.2: Correlation statistics between DMAPI routes' average trip delay and DMAPI routes' correlation with OD-OSRM computed trip delay. Note that correlation is being done between data retrieved in June for DMAPI and data retreived in October for OD-OSRM

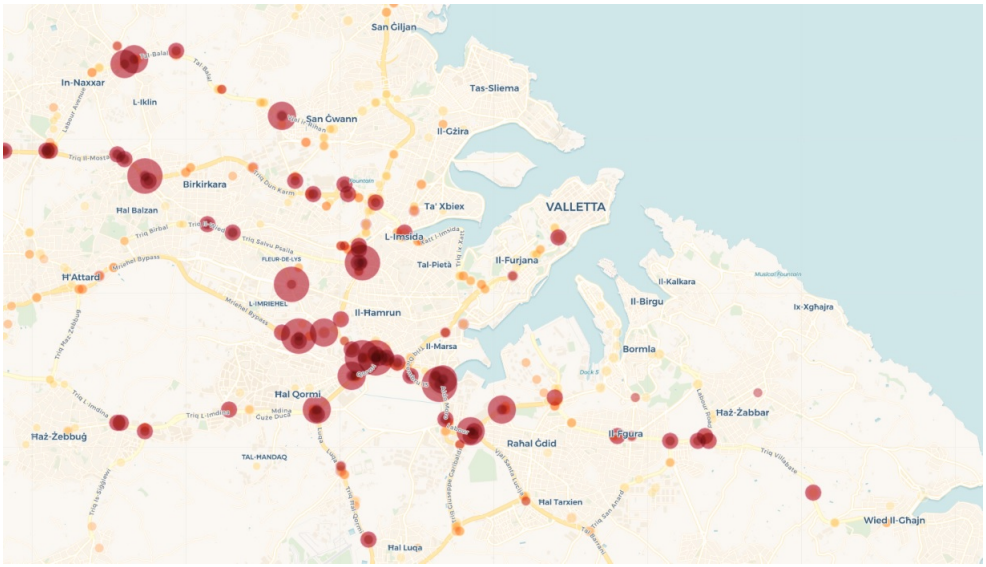


Figure 4.7: OD-OSRM traffic flow count mapping at 8:00 a.m. on a weekday

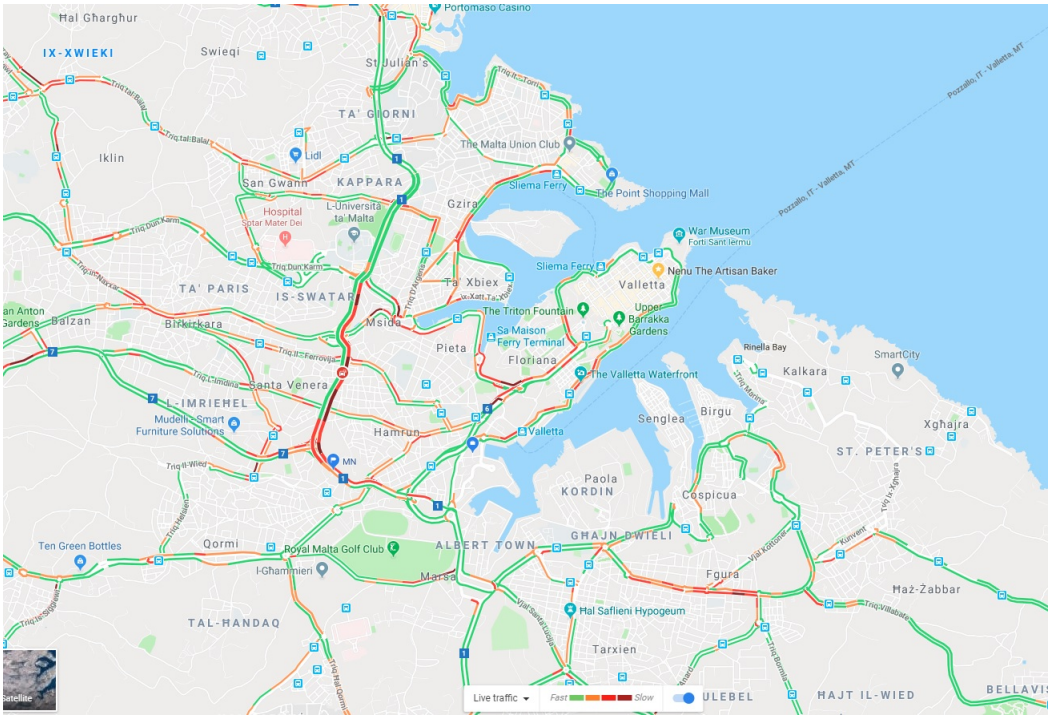


Figure 4.8: Google Map Traffic mapping at 8:00 a.m. on a weekday - to compare with OD-OSRM traffic flow count mapping

Seven whole days of Google DMAPI data from June 2018 was scraped by retrieving duration information for every quarter of an hour (see Figure 4.5 for an example). Estimated trip delay is calculated by subtracting estimated trip duration from trip duration in traffic. Average trip delay was then computed per hour. This process was done for four different routes namely Mosta to Marsa (MM), Mellieha to Swieqi (MS), Birkirkara to Sliema (BS) and Valletta to Mgarr (VM) and the overall trip delay average was calculated on these routes. These routes were chosen for two reasons. Firstly these were chosen because they represent routes that are really varied in type in terms of direction and areas covered. Secondly because the average time of the selected route trips which is retrieved from Google API (24 minutes) approximates the global trip average time which is reported for a car trip in [1] (20 minutes). To be noted however that from 2010, trip delays likely increase was due to further loading of traffic on the road infrastructure.

Correlation results showed that there is a strong linear relationship between the

routes' trip delay pattern which were investigated with the DMAPI. Correlation between DMAPI and OD-OSRM trip delay estimation is less but still considerable. Between DMAPI average overall trip delay and OD-OSRM non shifted expected trip delay data there is a correlation of 0.69 (see table 4.2).

When comparing trip expected delays we noticed some distinctive features that differentiate OD-OSRM trip delay plot and the DMAPI ones. Trip delays are higher in OD-OSRM data than for DMAPI. We also noted that patterns for plots based on DMAPI retrieved trip delay data tend to have delays peaking distinctively higher in the morning rather than in the afternoon. Furthermore, OD-OSRM trip delay morning peak comes 1 hour earlier. The fact that more trip delay is observed for the OD-OSRM dataset can be attributed to the fact that in October there is much more traffic. It is known that in Malta, October is one of the most chaotic months for traffic because schools and colleges would have just started. In June, the University of Malta semester is almost closing (no more lectures are being held) and primary and secondary students finish in the early afternoon. Government department work till midday from mid June as well. The earlier peak observed in the OD-OSRM data can be explained in the light that, in October, to cope with the heavy delays on the roads, commuters leave earlier to avoid traffic congestion. When shifting the OD-OSRM data by one hour (see dotted line in fig. 4.6) a higher correlation of 0.78 is observed with (DMAPI) overall average trip delay.

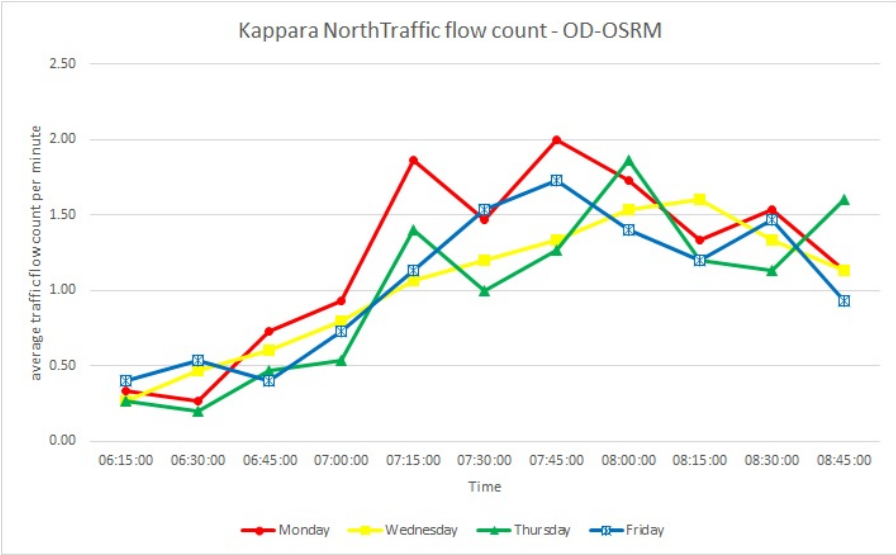
The fact that the data has different seasonality is a serious limitation in the evaluation of the OD-OSRM trip delay model with DMAPI data as ground truth. Given that at the time of writing results could not be recomputed for OD-OSRM for June it was attempted to get DMAPI webservice estimations data for October 2018. DMAPI does not give responses for data queries that request past data. However, it can give expected trip duration data in the future even if is queried from months ahead. The data retrieved for 7 days in October 2018 with DMAPI from four months in advance was exactly the same as the one retrieved for the same route in June. Therefore there was not the possibility to evaluate our model with

DMAPI data from the same month which was expected to a have a more similar pattern. Still, from the perspective of knowing how traffic in June and October is different, and the strong correlations between the trip delay estimation models, there is a good confidence level in relying on the OD-OSRM trip delay estimation data.

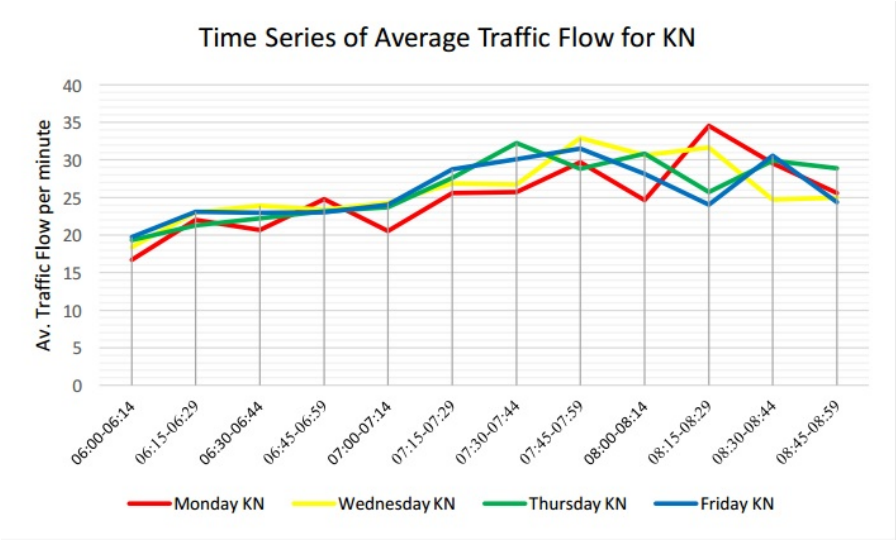
4.3 Traffic flow count evaluation

Route information was retrieved from OSRM with the OD matrix as input. Thus with a matrix of origins and destinations each trip done by any user was assigned a route (refer to method discussed in Section 3.5.2). The fastest route was retrieved from the OSRM and alternative routes were not considered. This is decisive together with OD matrix computation to characterize the traffic assignment model. The average OD-OSRM computed traffic flow count was compared with average actual traffic flow counts at the same locations and at the same exact date and time in order to analyse how accurately the traffic assignment distributes traffic flow with fastest route as default selection.

The ground truth data that was used, came from work done by Nigel Pace in his dissertation submitted in 2017 [44]. Directional traffic flow counts were manually gathered from web camera streams recorded from four locations. These were gathered from Kappara and Marsa roadways for traffic which is both northbound and southbound. The Marsa roadway is referred to as the Marsa-Hamrun bypass, which is the road leading to and from the Santa Venera tunnels. These roads are known for heavy traffic loads and congestion in Malta. The Kappara roadways get and feed traffic to the old Kappara roundabout which today has been replaced by a flyover. The dates for the data collection were from Monday 17th October to Friday 21st October. Data for the day of Tuesday 18th October was missing from the dataset and there was no particular reason specified why this was missing. The traffic flow count consisted of an average traffic flow count per minute taken over

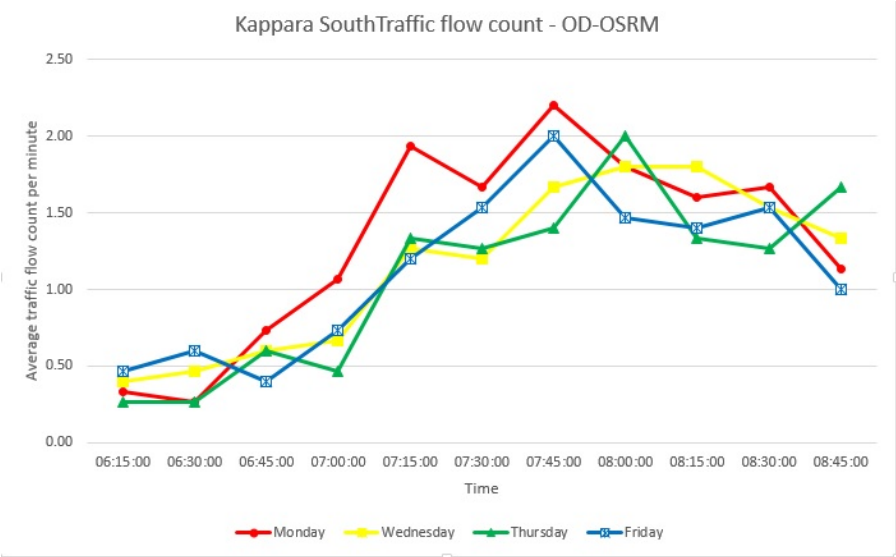


(a) Traffic flow Count Kappara North - OD-OSRM

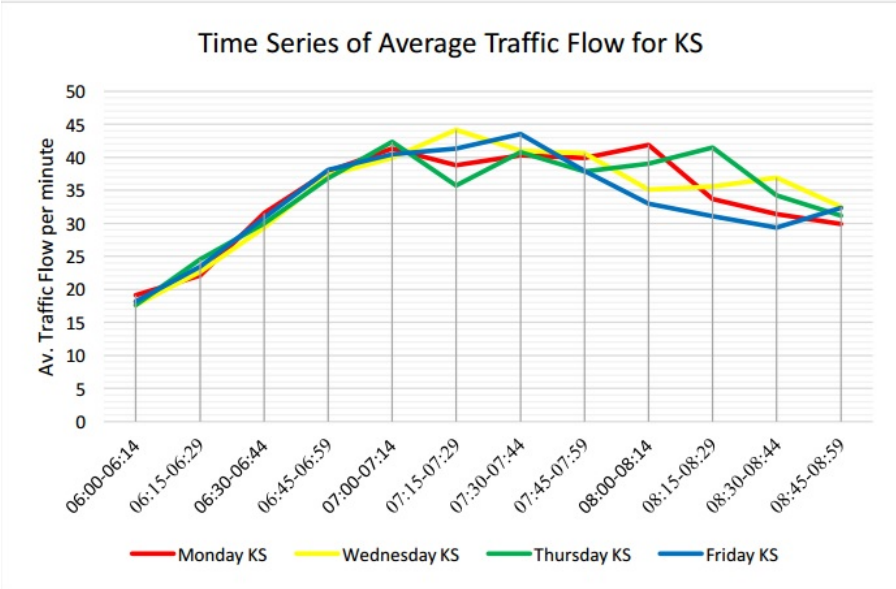


(b) Traffic flow Count Kappara North - video stream count

Figure 4.9: Kappara North Traffic flow counts from OD-OSRM and video stream count comparison. Video stream count graphic has been reproduced from [44].

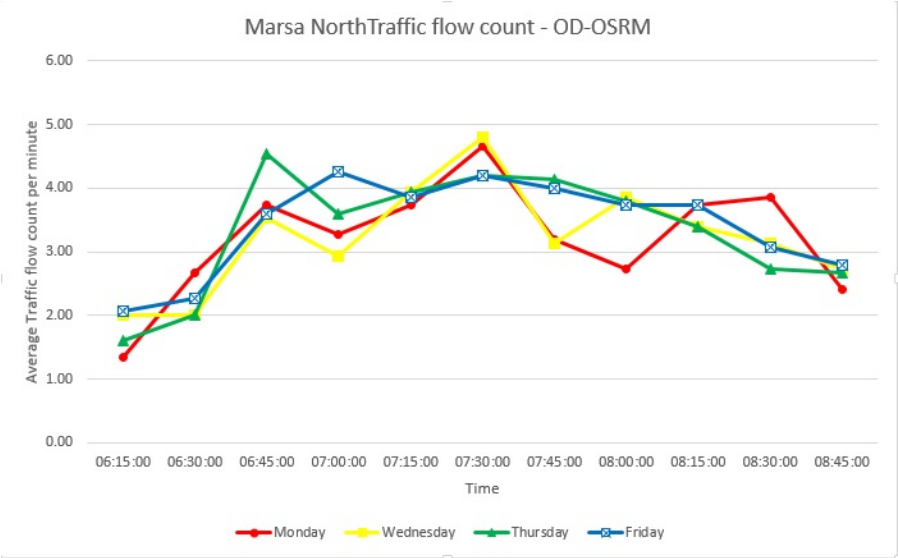


(a) Traffic flow Count Kappara South - OD-OSRM

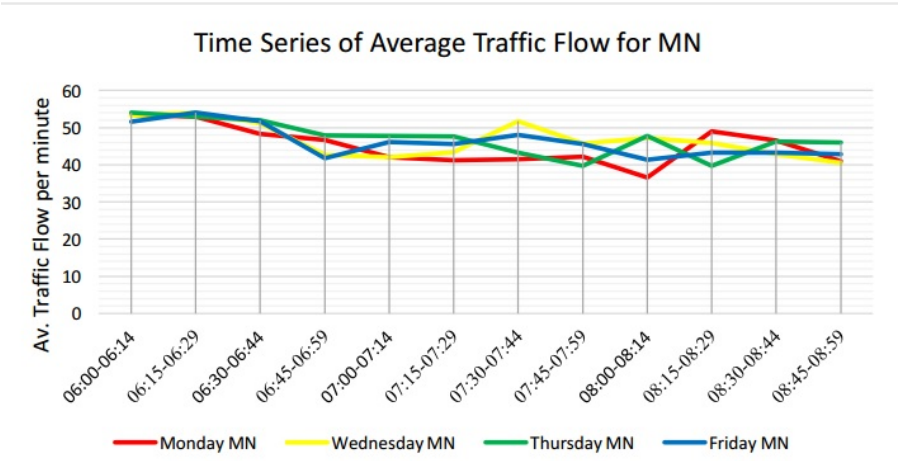


(b) Traffic flow count Kappara South - video stream

Figure 4.10: Kappara South Traffic flow counts from OD-OSRM and video stream count comparison. Video stream count graphic has been reproduced from [44]

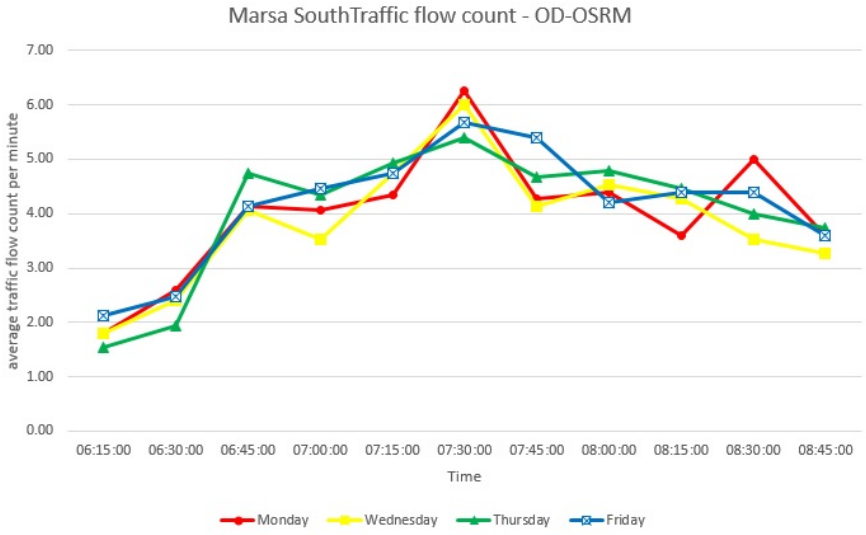


(a) Traffic flow count Marsa North - OD-OSRM

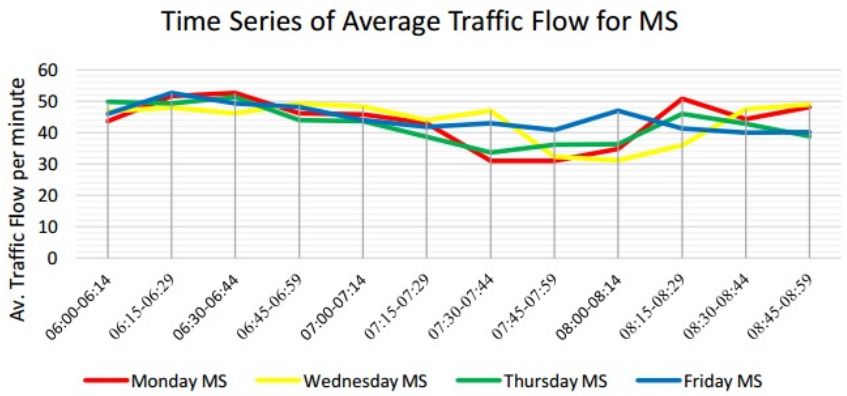


(b) Traffic flow count Marsa North - video stream

Figure 4.11: Marsa North Traffic flow counts from OD-OSRM and video stream count comparison. Video stream count graphic has been reproduced from [44].



(a) Traffic flow count Marsa South - OD-OSRM



(b) Traffic flow count Marsa South - video stream

Figure 4.12: Marsa South Traffic flow from OD-OSRM and video stream counts comparison. Video stream count graphic has been reproduced from [44].

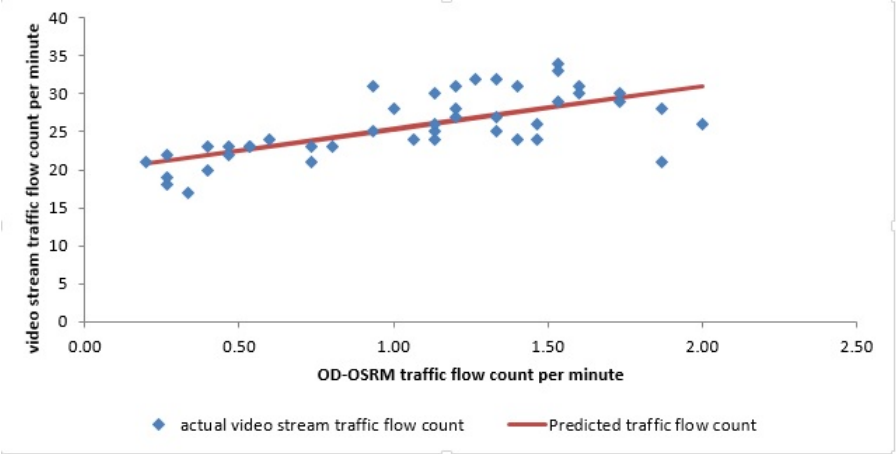
linear regression model OD-OSRM traffic counts vs video stream traffic counts	regression result output parameters			
	Pearson's Coefficient	R^2	p-value *	degrees of freedom
Kappara North	0.68	0.46	4.13E-07	43
Kappara South	0.75	0.56	3.85E-09	43
Marsa North	-0.46	0.22	0.0013	43
Marsa South	-0.31	0.10	0.04	43

Table 4.3: Correlation statistics for linear regression models. OD-OSRM traffic flow count is the predictor variable and video stream traffic count is the dependent variable. *Results are all significant with $p < 0.05$.

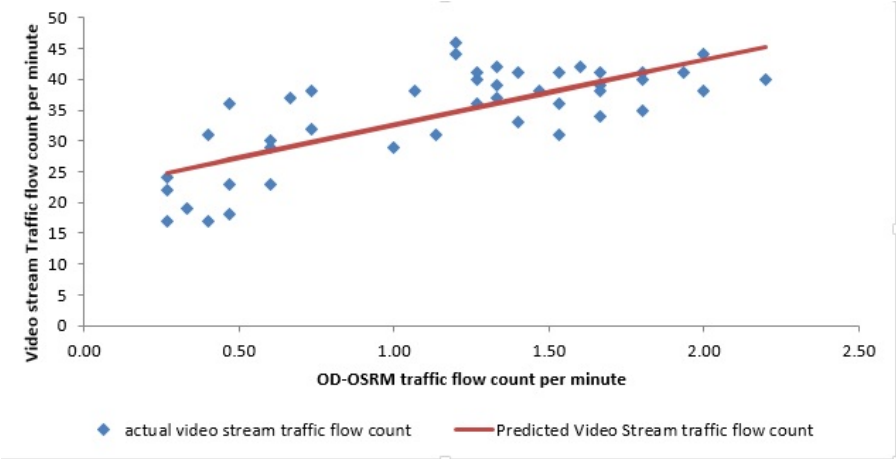
intervals of 15 minutes. This resulted into 44 samples for data gathered from 6.00 a.m to 8.45 a.m. for every day. A daily average for every quarter of an hour was then taken for both the actual data and the one generated with OD-OSRM. The traffic flow count data generated by OD-OSRM is only representative of a sample of the travelling population (i.e. those who have their 2G/3G/4G data switched on while travelling). This explains why the traffic flow counts generated by OD-OSRM are much smaller than those which were manually recorded in Pace [44]. This can clearly be observed in Figures 4.9a to 4.12b.

A simple linear regression model was fitted for each location to analyse the type of relationship between the OD-OSRM traffic flow counts and actual traffic flow count data used from [44]. There is no implied cause and impact relationship. We attempted to determine whether a true actual traffic flow count can be determined with a linear regression model from OD-OSRM traffic flow data. The resulting models were evaluated to determine how the independent variables which are location OD-OSRM traffic flow counts explain the variance of actual traffic flow. The null hypothesis here was that there is no significant functional mapping of actual traffic counts by OD-OSRM traffic counts for any specific road section.

Results include Pearson's correlation coefficient, R^2 which indicates the explained variance and p-value which shows that all results are statistically significant. Degrees of freedom value was 43 for each directional flow under study since there was one independent variable and 44 sample data points were available for

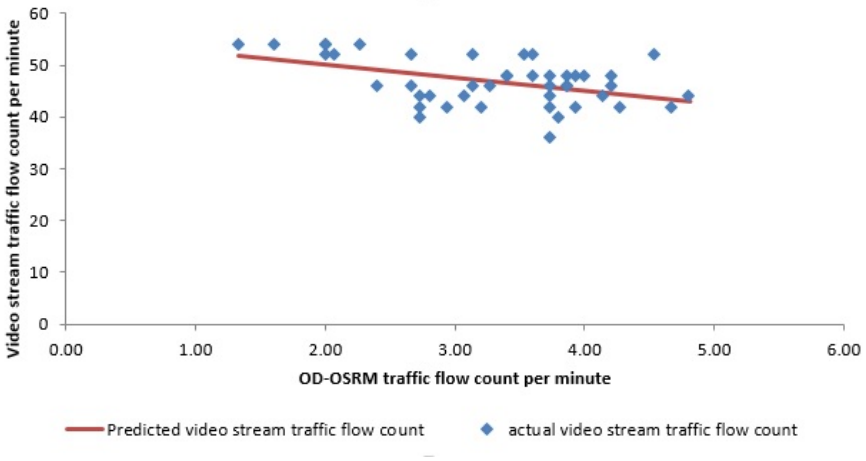


(a) OD-OSRM and video stream traffic flow Count linear plot - Kappara North

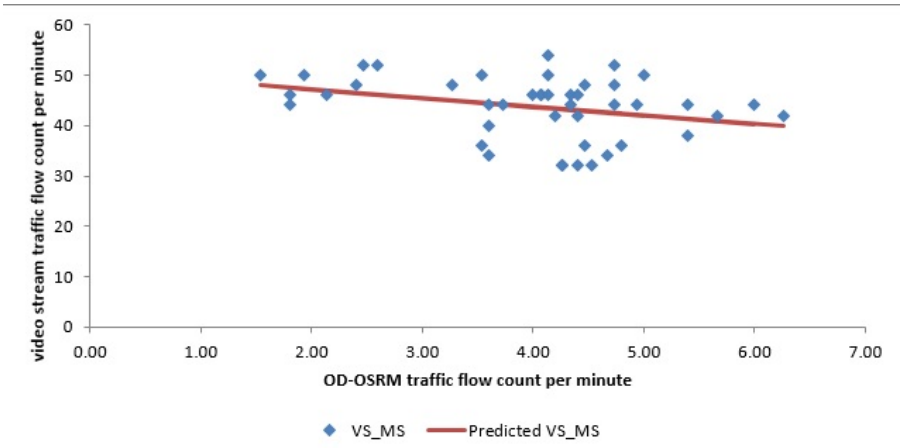


(b) OD-OSRM and video stream traffic flow Count linear plot - Kappara South

Figure 4.13: OD-OSRM and video stream traffic flow Count linear plot for Kappara traffic flow points.



(a) OD-OSRM and video stream traffic flow Count linear plot - Marsa Northbound



(b) OD-OSRM and video stream traffic flow Count linear plot - Marsa Southbound

Figure 4.14: OD-OSRM and video stream traffic flow Count linear plot for Marsa Northbound and Southbound traffic flow points.

each traffic flow point under examination. These results are shown in Table 4.3.

From the results' table one can conclude that there is a strong correlation between OD-OSRM and actual traffic flow counts for both Kappara carriage ways. The strong correlation is well illustrated with line plot shown in Figure 4.13. Line charts presented in Figure 4.9 and Figure 4.10 that illustrate traffic count for both Kappara carriageways show that traffic fluctuation patterns are very similar. On the contrary there is a weak negative correlation for Marsa traffic flow points (see Table 4.3 and line plots in Figure 4.14). In fact, when we analysed traffic flow charts in Figure 4.11 and Figure 4.12, we found that while actual traffic flow count starts high at 6.00 a.m. and gradually slows down up to 9.00 a.m. in OD-OSRM traffic flow charts shows that traffic increases constantly and peaks at 7.30 a.m. and then it starts to decrease for the subsequent later averaged samples. Traffic flow in Marsa traffic points does not necessarily mean that traffic flow is slowing down because there is less traffic load. It could be the case that traffic is slowing down because of an increase in traffic congestion [44].

Our explanation as to why there is strong correlation with Kappara traffic flows but a weak negative one with Marsa located traffic flows can be based on the fact that traffic tends to be slower in Marsa traffic flow points when compared with the Kappara traffic flow points. OD-OSRM measurements are based on trips that have been detected but if actual vehicular traffic slows down due to congestion the OD-OSRM traffic flow count does not reflect actual traffic counts. Therefore, two conclusions are derived from this. A first conclusion is that reliable regression models can be trained on actual traffic data for traffic flow road sections which do not experience heavy traffic slow down. Secondly the regression model mapped traffic flow counts gives a reliable account of what flow capacity is expected to be serviced at any given point in time from a given road section in order that traffic flows smoothly.

4.4 Traffic flow count prediction for a selection of locations evaluation

In Section 3.7, an approach on how to predict traffic flow count for specific road sections was presented. The model proposed is an MLPC that as a function takes the traffic flow count from each recorded location as input and predicts an approximated traffic flow count for a specific location/road section for a date time in the immediate future. The traffic flow counts are binned using specific labels through a logarithmic function.

Traffic flow count is represented with 4 bins which are equivalent to the model's classes. These classes range from class one to class four with class one being the lowest indicator of traffic flow count and class four being the highest. The MLPC model was devised by dedicating 60% of the data for training and 40% of the data for testing. Since Neural networks require a lot of data to train properly no data was dedicated for validating the models when searching the optimal hyper parameters such as neural network topology layout and PCA's first k components. A trial and error approach was used to check how the model would perform when changing such hyperparameters. Testing was done only for the application phase to evaluate how the model would perform with real-world data.

Collected performance metrics included accuracy and weighted precision, recall and F1-score. Accuracy gives a very basic picture of how the model is performing, however it does not provide clear information how the model is performing across all traffic flow count classes. The locations chosen from the available dataset possess the property of unbalanced classes. For example if a given location has 95% of classes of type one a model which always predicts class one will be 95% accurate on testing. Weighted precision and weighted recall further describe the performance of the model. When having a high precision and a low recall the model is more appropriate for exactness in classification (false positives are kept at a minimum at the cost of a high number of false negatives). A high recall and low precision model

Prediciton evaluation metrics for a given traffic flow point with variable prediction ahead		Evaluation results' metrics			
Traffic flow section	**Prediciton time interval ahead**	***Accuracy***	***weighted Precision***	***weighted Recall***	***F1-Score***
Kappara south bound	**15 minutes**	0.67	0.66	0.67	0.66
	30 minutes	0.67	0.66	0.67	0.66
	60 minutes	0.64	0.64	0.64	0.64
	1 day	0.61	0.60	0.61	0.61
Imsida skate park	**15 minutes**	0.70	0.70	0.70	0.70
	30 minutes	0.68	0.70	0.68	0.69
	60 minutes	0.68	0.69	0.68	0.68
	1 day	0.62	0.62	0.62	0.62
Hamrun - Valletta	**15 minutes**	0.91	0.89	0.91	0.90
	30 minutes	0.91	0.89	0.91	0.90
	60 minutes	0.90	0.89	0.90	0.90
	1 day	0.90	0.88	0.90	0.89
Marsa roadway leading to Aldo Moro	**15 minutes**	0.67	0.68	0.67	0.67
	30 minutes	0.66	0.66	0.66	0.66
	60 minutes	0.64	0.65	0.64	0.65
	1 day	0.58	0.58	0.58	0.58

Table 4.4: Classification evaluation metrics for 4 traffic flow road sections with 4 label classification and PCA set to extract 324 first components. Testing was done with 4 sizes of prediction time window ahead for each prediction location.

is better in identifying a higher percentage of classes correctly but can output a relatively high number of false positives in the process.

In machine learning sometimes high recall is more important than high precision or the other way round and in most of the times there is a trade-off. The more tuning is made to any one of the metrics to improve it, the riskier it is to get a lower performance in another metric. The ideal is to have both high recall and high precision. In the case of this study recall for high level of traffic classes is very important since knowledge of high traffic count is important and noise would be acceptable. Weighted F1-score was used to portray a balanced measure between recall and precision.

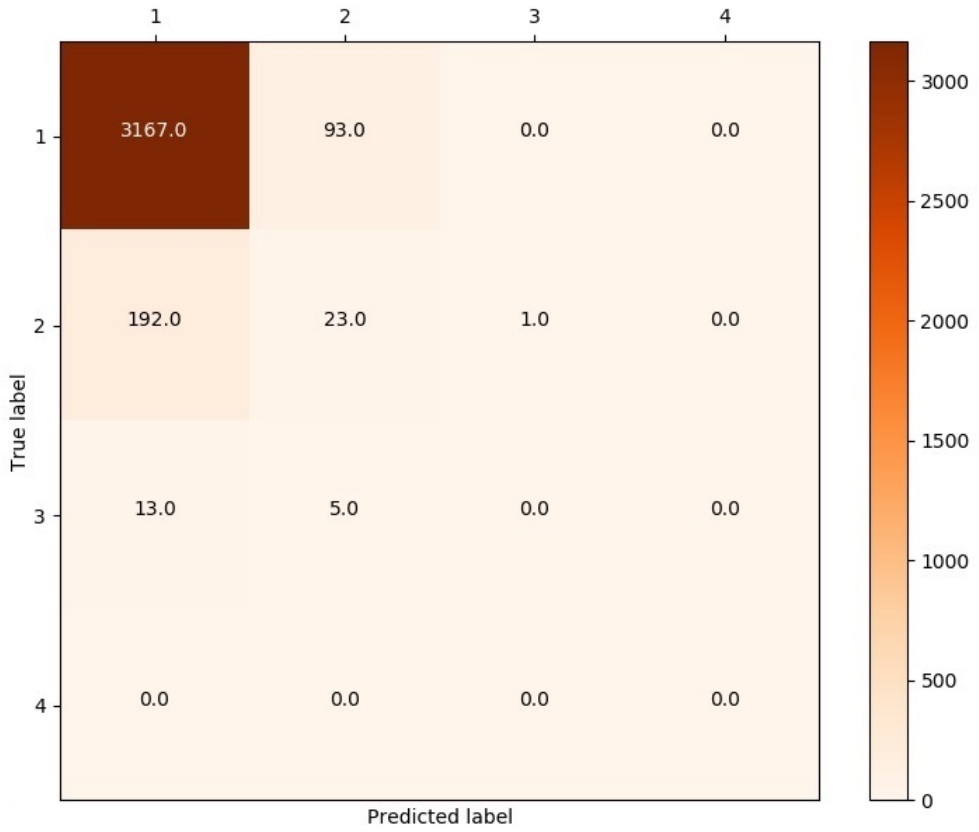


Figure 4.15: Confusion matrix for Hamrun to Valleta road traffic flow count for 15 minutes ahead prediction interval.

Table 4.4 shows the evaluation results described in terms of the metrics just discussed. It can be seen that models trained to predict for smaller time ahead intervals generally perform better than models that are trained with a lengthier prediction time interval for the same location. Models all have proven to have highest recall and precision for class one traffic flow counts. It appears from table that the best overall classification metric scores were attained for Hamrun-Valletta roadway. However on examination of the confusion matrix (shown in Figure 4.15) for classification results per label we noted that the model performed very badly for high traffic flow count classes. There were no results for class four and for classes two and three the precision and recall metrics are very low. In fact, when computing the F1-score for class two and class three, both result to be low at 0.14

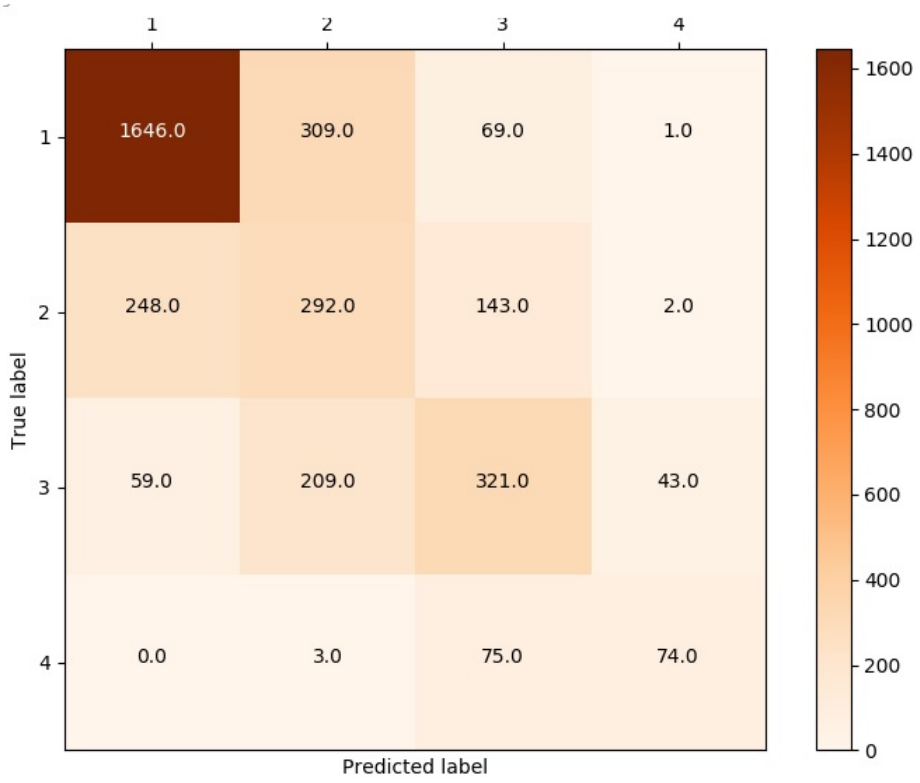


Figure 4.16: Confusion matrix for Marsa to Aldo Moro road traffic flow count for 15 minutes ahead prediction interval.

and 0.0 respectively. This was found to happen as well in literature. In [45] it is stated how trained ANN model does not perform well when traffic counts are low. Relative error in evaluation is much bigger when traffic flow is small. Results are only being quoted by Lv et al. when traffic flow is 450 vehicles or more for a 15 minute time window [45].

In contrast predictive overall results for Marsa road that leads to Aldo Moro are less promising than those for Hamrun-Valletta arterial road. Still, the predictive efficacy results are very good, especially when examined in the perspective of the confusion matrix shown in fig. 4.16. Class four cases, which are classified as class one or class two cases are very few and, even if almost half of class four test values were predicted as class three, in practice, this would still make the model useful and offer guidance to describe the level of high traffic flow counts.

4.5 Conclusion

This chapter demonstrated how the proposed techniques were evaluated. We have seen how there is a very strong correlation of 0.94 between global average trip count per hour distribution which was determined with our method and the one reported in NHTS. This may indicate that our model which is derived from mobile usage data is highly relevant for the real world with assured performance. Our model has an edge on surveys similar to NHTS because the illustrated trip count per hour statistics have seasonality and the model can be updated frequently.

Global trip delay was evaluated against data collected from Google's DMAPI. 4 routes were used for testing and correlation statistics were compiled. A very strong correlation of 0.78 exists between the average trip delay computed with our method and DMAPI average trip delay.

When building models to express a linear regression between our method and manual data collected from video streams by Nigel Pace we saw that there was a strong correlation between models for Kappara South and Kappara North roadways but a week negative correlation with Marsa Hamrun bypass North and South roadways. We concluded that a linear regression model can be used to accurately upscale traffic flow count data from our method for a given road section, only if the road section does not experience traffic congestion frequently.

We evaluated the prediction with a MLPC by dedicating 40% of the data for testing. Performance metrics were different for the 4 locations for which prediction testing was done. The highest F1-scores were achieved for Hamrun to Valletta route traffic flow prediction but on further scrutinizing the confusion matrix a heavy imbalance of classes towards class 1 was the reason why good prediction results were observed. The Hamrun to Valletta predictive model in fact did not perform well for classes that represent higher traffic flow counts such as Class 2 and Class 3. For other locations promising results were observed with a generally strong F1-score for all the classes. The models performed better when predicting from a shorter interval before.

5. Conclusion

In this dissertation we described a systematic approach from which a data processing pipeline was devised to extract vehicular traffic patterns from mobile usage data. A cascading data processing pipeline methodology was used which consisted in building models for each pipeline phase that processed input data and fed next phases down the chain. The ensuing ensemble of techniques can be applied to deduce traffic analytics from time series mobile usage data.

The aforementioned pipeline models include users' activity hubs clustering, OD matrix based trip generation, trip delay information and traffic flow measurement and prediction. Density based clustering was used to create an OD matrix containing data on visits done to the two top locations, defined as the two places where users made most use of mobile data. Trips were generated between every recorded departure from origin and arrival to destination and the resulting trip duration. OSRM was used to retrieve routes for these trips and get the actual estimated trip duration without traffic. The difference between the OSRM trip duration and the actual trip duration recorded from OD trip generation was considered to be the trip delay statistical information. OSRM was used as well to collect time series data that indicated where the traffic flow is being distributed on the road infrastructure in order to build aggregate statistical models. The traffic flow counts' dataset then was used to train an MLPC to build a model that predicts traffic flow count at varied time intervals ahead.

Clustering was the only part of the methodology that could not be evaluated directly. Ground truth was not available to test how accurately the resultant main locations for each user were matching the actual ones. Its evaluation relied on traffic flow count evaluation which is highly dependent on the OD-OSRM generated trips which are defined by the users' main two clusters.

Since the mobile users population does not match the number of commuters that use their own vehicle for transportation it was expected that trip distribution statistics would give lower values than those reflected by actual values. A scaling up method was needed to extrapolate real world trip distribution statistics. A very strong correlation of 0.94 was found between average NHTS 2010 trip distribution data and trip distribution generated by the OD-OSRM method devised in our research. A linear regression model was built to map OD-OSRM figures to scaled up numbers. One should underline the fact that a lot of human resources are needed to collect survey data and that surveys become eventually outdated. The method proposed in this research can use readily available data, can have its model dynamically updated and can be configured with a 5 minute temporal resolution. Survey responses may not contain exact departure and arrival times since people possibly tend to round these when replying. Therefore, as a conclusion our method is more practical than surveys to get information on user trips. However, it needs to be calibrated and scaled up by modelling on actual surveys through regression. This, however, does not need to be done frequently.

Correlation statistics were carried out for trip hourly average delay for four different routes. A promising overall correlation of 0.72 was found between trip delay data originating from OD-OSRM and Google's DMAPI. However one must note that DMAPI data was collected for a week in the month of June 2018 and OD-OSRM data was collected in October 2016. Our approach seems to work better to give predictive models for the distant future because Google's model for the distant future gives identical results to those given for the immediate future. Our method can be easily modelled with seasonality to forecast distant traffic delay if a year of

data is available.

Traffic flow distribution linear regression models were built to map OD-OSRM traffic flow distribution to actual ones. The linear relationship with data collected manually for four locations was analysed. It was concluded that the linear relationship is statistically significant and tends to have high correlation if the road traffic flow capacity is not exceeded. Otherwise for locations that tend to have high traffic congestion, correlation is both low and negative during traffic congestion time. Therefore, it was concluded that negative correlations in traffic flow distribution linear models indicate that locations are experiencing slowing down of traffic due to congestion.

Machine learning techniques were employed to predict traffic flow counts at a given time for a given location from previous traffic flow counts at an earlier time for all location data points. An MLPC was trained for four locations and prediction intervals ranged from 15 minutes to 1 day. Satisfactory results were attained and from these results it is concluded that users or information support systems can make well informed decisions on predicted traffic flow data for selected locations.

The main strength of this dissertation was to give an accurate measure and an effective prediction of traffic flow demand (not traffic congestion) on the road infrastructure. The insight gained could help transport agencies' administrators to tackle infrastructure problems before they eventually happen. Improvement is mainly needed on dynamic traffic assignment algorithms.

5.1 Major contributions of this dissertation

This research posed questions on whether it is feasible to get vehicular traffic descriptive and predictive analytics from mobile usage data. We showed how it is possible to retrieve top activity locations for users. It is possible also to achieve accurate results in getting global trip counts and trip delays. From mobile data usage as well, we have shown that it is possible to collect trip data for all users.

The trip data was then used to actually map traffic flow demand on the road grid. However, it was found that traffic flow mapping gave accurate results for low traffic congestion roads, whereas for high traffic congestion roads our model did not give accurate results. Finally, a MLPC was found really efficient to predict traffic flow for a set of locations. The confidence level given by the prediction results is high and if traffic flow input used to train the predictive model is accurate the method we devised could be used in the real world to forecast traffic in real-time. We did not experiment with processing of real-time streams but the model we devised can be easily adapted for real-time processing since it uses window analytics which are widely used for stream processing.

5.2 Discussion

In this research machine learning was used to train an MLPC with all locations' traffic flow count as data features. All the traffic flow count would be mapped by a built model to a level of traffic representing the future traffic flow count for a specific location. A similar predictive model can be built to predict estimated trip delay for every given route for any given required time. A comprehensive used routes database that exists in Malta for all users based on the OD matrix has to be compiled. Then statistical information on trip delays per 5 minutes for same routes are aggregated. One should investigate the possibility of using routes' trip delays as training input features to a machine learning model. The final predicted classification would be a trip delay class for certain give routes. The model basically would be a function that maps trip delays for a set of routes to a classification of trip delay for a given route.

In order not to increase the evaluation combinations only two main clusters' location were retrieved per user. These two clusters were considered as being the home and work locations. However the main relevant assumption was that most trips were made between these two main clusters. This is not representative for all

trips made and would incur phantom trip delays that could only be explained by stops at locations frequently visited by users that are not one of the main activity clusters. A fact that heavily impacts daily traffic in Malta are extra trips whose destinations are concentrated around schools. School run trips can be detected by extracting the third most common location that is located within a school geofencing boundary and by time boxing with school starting and finishing hours. A lot of trips have been discarded with the used methodology because they had excessive delays. Many of these trips can be retained to better explain the traffic dynamics if trips are further divided with the insertion of a third location.

Other datasources such as social media and ANPR video streams could be used to further dynamically assign traffic to the road network. From social network feeds one can extract for example accidents location and time. This data could be used to analyse the accident impact on traffic flow and can be used as an input feature for machine learning models. Also correlation between weather and traffic flow can be done by using available weather APIs.

This dissertation's methodology made extensive use of Open source maps (OSM) and Open Source Routing Machine (OSRM). An interesting project would be to edit the OSM maps under observation (opening of a new road) with map editing software and check the impact on the traffic flow of routes which are adjacent or near to the modification done in the map. OSRM generates route information based on the OSM data and when the OSM data file is modified the change is reflected in the routing information. This would be very useful to simulate and analyse the impact on traffic flow before the actual alteration is made in the road infrastructure.

6. Future Works

This dissertation created a lot of experimental data while developing and evaluating techniques which can be the background for future projects. Features suggested in this chapter were not investigated due to time constraints. Future works are categorized into two namely, future improvements and future research.

6.1 Future improvements

The challenge in such projects is to find ground truth to properly evaluate findings and achieved results. One way to achieve this is to develop or use off the shelf software that collects GPS points and collaborate with a group of people to gather the data. The resulting sample of data would be ideal to carry out correlations with experimental outcomes.

DBSCAN was used to find density based clusters to extract the most common origin and destination locations for users. As discussed in Section 3.4, ϵ was set to 500m. OPTICS is an algorithm which is similar to DBSCAN but it does not require ϵ parameter as an input and is more efficient to find meaningful clusters for data with varying density [46]. Clusters generated on data used in this research and that had a weak density might have not been captured with a radius of 500m. DBSCAN was furthermore computationally expensive and consumed a lot of time in the experimentation phase. An optimization for the performance of the processing of

the whole pipeline would be running DBSCAN on data in parallel in a distributed manner. An implementation based on the Spark platform is proposed in [38].

One limitation in this dissertation was to assume that users always take the fastest route and do not detour for whatever reason along the trip. Traffic assignment therefore is static and does not adapt to the traffic events on the road infrastructure. This does not give optimal results when gathering real-time and historical analytical statistics on routes, trips and distributed traffic load on the road infrastructure. Other methods must be combined with the proposed solution to rectify route selection and give a more realistic picture of traffic flow based on actual user route taking.

One way how to improve the correct route selection rate is by further polling mobile usage data during a user trip and snap the user to the nearest road route with snap to road software such as Roads API from Google[1]. This type of software takes a set of coordinates as input and returns a similar set of data that most likely define the route outlined by the set of data. Route selection would not be necessarily computed for each user every time a user trip is detected. A statistical model can be built to learn what are the most likely routes taken by each user on a given date and time context. This model can then be used in the application phase to analyse traffic loads which are related to the predicted routes.

An averaging moving filter could be used to remove noise from the traffic flow count and improve results by applying a smoothing function. When getting a sliding average more data points can be used for prediction. In our approach to reduce computational costs, an average with binning size of 5 minute each was calculated. This reduced however, a 60 data point resolution to 12 data points in an hour. When building a machine learning model that predicts traffic flow counts, a fixed interval was used to get the future classification label (refer to section 3.7). This fixed interval had to be a multiple of the 5 minute window and there was no flexibility for more granular tuning. Having less data points for the reasons just

[1]https://developers.google.com/maps/documentation/roads/snap (accessed January 8, 2018)

mentioned provided less data for training to the MLPC.

Evaluation for average trip delay computation (see section 4.2) was done by comparing results based on a June dataset with data retrieved through DMAPI and an October mobile data usage dataset. This evaluation although it gave promising results should be repeated with seasonality of datasets removed from the equation to compare like with like. Malta traffic in October is very different from Malta traffic in June.

References

[1] T. Malta, "National household travel survey 2010. transport malta," 2011.

[2] Directorate General for Mobility and Transport, "Urban mobility," 2018.

[3] S. Çolak, L. P. Alexander, B. G. Alvim, S. R. Mehndiratta, and M. C. González, "Analyzing Cell Phone Location Data for Urban Travel," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2526, pp. 126–135, 2015.

[4] M. Attard, P. von Brockdorff, and F. Bezzina, "The external costs of passenger and commercial vehicles use in malta," 2015.

[5] U. M. Scorecard, "The texas a&m transportation institute and inrix," *Inc., USA*, vol. 9, no. 2015, p. 10, 2015.

[6] Cambridge Systematics Inc. and Battelle Memorial Institute, "An Initial Assessment of Freight Bottlenecks on Highways," no. October, p. 191, 2005.

[7] E. Al Nuaimi, H. Al Neyadi, N. Mohamed, and J. Al-Jaroodi, "Applications of big data to smart cities," *Journal of Internet Services and Applications*, vol. 6, no. 1, pp. 1–15, 2015.

[8] G. Atkinson and S. Mourato, *Cost-Benefit Analysis and the Environment*. OECD Publishing, 2015.

[9] P. Naess, M. S. Nicolaisen, and A. Strand, "Traffic Forecasts Ignoring Induced Demand: a Shaky Fundament for Cost-Benefit Analyses," *European Journal of Transport and Infrastructure Research*, vol. 12, no. 3, pp. 291–309, 2012.

[10] J. L. Toole, S. Colak, B. Sturt, L. P. Alexander, A. Evsukoff, and M. C. González, "The path most traveled: Travel demand estimation using big data resources," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 162–177, 2015.

[11] J. Liu, F. Liu, and N. Ansari, "Monitoring and analyzing big traffic data of a large-scale cellular network with Hadoop," *IEEE Network*, vol. 28, no. 4, pp. 32–39, 2014.

[12] G. Leduc, "Road Traffic Data : Collection Methods and Applications," *EUR Number: Technical Note: JRC 47967*, vol. JRC 47967, p. 55, 2008.

[13] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 3 – 19, 2014. Special Issue on Short-term Traffic Flow Forecasting.

[14] F. Calabrese, M. Diao, G. Di Lorenzo, J. Ferreira, and C. Ratti, "Understanding individual mobility patterns from urban sensing data: A mobile phone trace example," *Transportation Research Part C: Emerging Technologies*, vol. 26, pp. 301–313, 2013.

[15] S. Hoteit, S. Secci, S. Sobolevsky, C. Ratti, and G. Pujolle, "Estimating human trajectories and hotspots through mobile phone data," *Computer Networks*, vol. 64, pp. 296–307, 2014.

[16] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.

[17] S. Hoteit, G. Chen, A. Viana, and M. Fiore, "Filling the gaps: On the Completion of Sparse Call Detail Records for Mobility Analysis," *Proceedings of the Eleventh ACM Workshop on Challenged Networks - CHANTS '16*, no. October, pp. 45–50, 2016.

[18] J. Steenbruggen, E. Tranos, and P. Nijkamp, "Data from mobile phone operators: A tool for smarter cities?," *Telecommunications Policy*, vol. 39, no. 3-4, pp. 335–346, 2015.

[19] L. Alexander, S. Jiang, M. Murga, and M. C. González, "Origin-destination trips by purpose and time of day inferred from mobile phone data," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 240–250, 2015.

[20] F. Calabrese, G. Di Lorenzo, L. Liu, and C. Ratti, "Estimating origin-destination flows using opportunistically collected mobile phone location data from one million users in boston metropolitan area," *IEEE Pervasive Computing*, no. 4, pp. 36–44, 2011.

[21] M. S. Iqbal, C. F. Choudhury, P. Wang, and M. C. González, "Development of origin-destination matrices using mobile phone call data," *Transportation Research Part C: Emerging Technologies*, vol. 40, pp. 63–74, 2014.

[22] M.-h. Wang and S. D. Schrock, "Feasibility of Using Cellular Telephone Data to Determine the Truckshed of Intermodal Facilities," *Cell*, no. August 2009, 2012.

[23] R. Ahas, M. Tiru, E. Saluveer, and C. Demunter, "Mobile telephones and mobile positioning data as source for statistics : Estonian experiences," *Presentation for NTTS*, 2011.

[24] M. H. Wang, S. D. Schrock, N. Vander Broek, and T. Mulinazzi, "Estimating Dynamic Origin-Destination Data and Travel Demand Using Cell Phone Network Data," *International Journal of Intelligent Transportation Systems Research*, vol. 11, no. 2, pp. 76–86, 2013.

[25] P. Locke, "Cell tower triangulation - how it works," Jun 2012.

[26] H. Wu, T. Zhang, and J. Gong, "GeoComputation for Geospatial Big Data," *Transactions in GIS*, vol. 18, no. S1, pp. 1–2, 2014.

[27] A. Eldawy and M. F. Mokbel, "Spatialhadoop: A mapreduce framework for spatial data," in *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*, pp. 1352–1363, IEEE, 2015.

[28] L. Alarabi, A. Eldawy, R. Alghamdi, and M. F. Mokbel, "TAREEG : A MapReduce-Based Web Service for Extracting Spatial Data from OpenStreetMap *," pp. 0–3, 2014.

[29] J. K. Laurila, D. Gatica-Perez, I. Aad, J. Blom, O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, and M. Miettinen, "The mobile data challenge: Big data for mobile computing research," *Proceedings of the Workshop on the Nokia Mobile Data Challenge, in Conjunction with the 10th International Conference on Pervasive Computing*, pp. 1–8, 2012.

[30] H. Shin, J. Vaidya, V. Atluri, and S. Choi, "Ensuring Privacy and Security for LBS through Trajectory Partitioning,"

[31] W. Inoubli, S. Aridhi, H. Mezni, M. Maddouri, and E. M. Nguifo, "An experimental survey on big data frameworks," *arXiv preprint arXiv:1610.09962*, 2016.

[32] A. M. Kurien, G. Noel, K. Djouani, B. J. Van Wyk, and A. Mellouk, "A subscriber classification approach for mobile cellular networks," *Simulation Modelling Practice and Theory*, vol. 25, pp. 17–35, 2012.

[33] Y. Zheng and X. Xie, "Learning travel recommendations from user-generated GPS traces," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 1, pp. 1–29, 2011.

[34] T. Toledo, M. Ben-Akiva, D. Darda, M. Jha, and H. Koutsopoulos, "Calibration of Microscopic Traffic Simulation Models with Aggregate Data," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1876, pp. 10–19, 2004.

[35] S. Hirai, J. Xing, R. Horiguchi, T. Shiraishi, and M. Kobayashi, "Development of a Network Traffic Simulator for the Entire Inter-urban Expressway Network in Japan," *Transportation Research Procedia*, vol. 6, no. June 2014, pp. 285–296, 2015.

[36] A. Bazghandi, "Techniques, Advantages and Problems of Agent Based Modeling for Traffic Simulation," *International Journal of Computer Science Issues*, vol. 9, no. 1, pp. 115–119, 2012.

[37] S. Chakraborty NKNagwani Lopamudra Dey, "Performance Comparison of Incremental K-means and Incremental DBSCAN Algorithms," *International Journal of Computer Applications*, vol. 27, no. 11, pp. 975–8887, 2011.

[38] F. Huang, Q. Zhu, J. Zhou, J. Tao, X. Zhou, D. Jin, X. Tan, and L. Wang, "Research on the parallelization of the dbscan clustering algorithm for spatial data mining based on the spark platform," *Remote Sensing*, vol. 9, no. 12, p. 1301, 2017.

[39] M. Sommer, S. Tomforde, and J. Hähner, "Using a neural network for forecasting in an organic traffic control management system.," in *ESOS*, 2013.

[40] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *Computer*, vol. 29, no. 3, pp. 31–44, 1996.

[41] S. Raschka, *Python machine learning*. Packt Publishing Ltd, 2015.

[42] K. Yang and C. Shahabi, "A PCA-based similarity measure for multivariate time series," *Proceedings of the 2nd ACM international workshop on Multimedia databases - MMDB '04*, p. 65, 2004.

[43] K. Hornik, M. Stinchcombe, and H. White, "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," *Neural networks*, vol. 3, no. 5, pp. 551–560, 1990.

[44] N. Pace, "Investigating the Potential of Big Data in the Management of Traffic in Malta," 2017.

[45] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[46] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure," in *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, SIGMOD '99, (New York, NY, USA), pp. 49–60, ACM, 1999.