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

Maurice Saliba

Supervisor: to insert

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

Faculty of ICT

Declaration

I, the undersigned, declare that the dissertation entitled:

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

submitted is my work, except where acknowledged and referenced.

Maurice Saliba

28-06-2018

Acknowledgements

your acknowledgments

Abstract

Multi-modal originated vehicular traffic flow data can be obtained with various techniques. To what extent this data is reliable, complete, timely and readily available requires a thorough analysis of past work and currently available solutions. A novel approach consisting of an ensemble of machine learning and data-mining techniques is being proposed. A mobile phone usage dataset from a telecommunications provider in Malta is used first to carry out basic traffic analytics. Then an origin-destination (OD) matrix based on the largest two clusters of activity per user will be computed to infer user trips between these clusters across time. Routes for these trips are retrieved with open source routing tools and obtained data pertaining to way nodes along these routes further enrich trip information. Spatial binning is then used to deduce the aggregate distribution of traffic load on the traffic network. The OD matrix and grid network load are subsequently used to build a Neural Network predictive model. Several previous works [23, 18] that carried out invaluable research in this field lacked on-line data in quality and quantity. They were at times compelled to devise corrective measures and carry out simulations to cater for such shortcomings. Having the luxury to avail of mobile call and data historical records will make it more possible to fine tune a better predictive model and evaluate it. Industry standard visualization tools were used to portray AI generated traffic patterns together with flow intensity projected in the geospatial dimension.

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 required for displacement of people for work, leisure and other purposes. Transport is important as well to deliver goods and services. Land transportation has undoubtedly evolved with 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.

Economic development and urbanization comes 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 private car is the preferred mode of travelling. [12] mentions how traffic congestion in urban areas in the EU is costing 100 billion every year which amounts to 1% of the EU GDP. [11] ellaborates on the crippling effect on the economy because of traffic congestion. 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 [5]. 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 [29]. 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 impact direly also the shipment industry. Travel costs increase when travel time increases. Pick-up and delivery times are more approximative. 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 [29] [9].

1.2 Addressing Traffic congestion

Car users and even public transport users (since buses cannot avoid traffic although use of it alleviates it) tend to get rather frustrated from lost time on the road. This time is deducted from a healthy lifestyle or from productivity hours.

Driver self adaptation measures help to smartly mitigate delay times. Individual drivers can hear radio adverts or check CCTV to enquire the traffic situation before departing or even while driving for better planning. Use of software such as Google maps, Apple maps or Waze help to have an informed decision how to schedule trips and decide what route to take. These applications might even suggest to take other transport modes because it is more convenient especially in terms of less time to get to destination.

Addressing traffic delays should be addressed within a wider scope. Efficient traffic management should be on top of government transportation agencies agenda. Traffic management is multi-faceted especially the urban one. Possible measures that can be taken by transport authorities include making different modes of trans-

port available and encourage the public to use it. Smart technology is a good candidate to alleviate travelling frustration by giving information,instructions and control parameters that have a say on traffic. 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 go for any generic modal shift. But these are rather unpopular measures and policy making agents are susceptible to pressure in such a way that they are less proactive to implement such measures.

Other measures to tackle traffic problems is by enforcing traffic laws. This would diminish road accidents or casual road blockages that can cause flow disruptions. CCTV road network monitoring would be helpful to inform drivers to take alternative routes. CCTV could be used also for deployment of traffic management personnel in problematic areas. License number plate recognition 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 may shift away concentration of traffic from urban centres. [2] for instance suggests how concerted efforts can lead to smart cities by analysing static data and make infrastructural changes by opening or modifying roads. Dynamic data then would be used to manage traffic lights to alleviate congestion, inform the public through their smart phones about the traffic status and orchestrate shipping movement for the supply chain.

Investment in the transport infrastructure to expand capacity is difficult to directly justify with a simple cost benefit analysis model. 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 not 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 [27]. When formulating return on investment functions induced traffic should not be ignored.

1.3 Traffic information and management systems

Traffic management would primarily consist of a systematic approach to monitor accurately, with wide coverage what is the traffic status in the road network. In order that this information is kept relevant it needs to be constantly updated and gathered in a reliable fashion. Once such information is acquired intelligent traffic management systems architectures can be designed around static data and traffic control is based on constant input feed stream processing. Obviously the latter is more challenging 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 [34]. 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.

Traffic information Systems may be based on mobile data as main source of information. These type of systems would require that mobile data collection and its processing has wide coverage, is reliable and accurate and updated with high frequency [24]. Less coverage is to be expected in rural areas where base stations are highly dispersed when compared to urban areas. Mobile vehicle geolocation cannot be used for traffic flow counts on lanes as it can be done with inductive loops. Frequency of updates refers to collection of data that probes the traffic situation (counts) and also to the amount of fresh updates users get from the

traffic information systems in real time.

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 at large scale of all the vehicles in a time series is difficult to model and then predict how traffic is affected along the course of the day. Traffic sensors, cameras and induction loops are all sources of information that can be lead 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. Crowd-sourced information that gives information on mobility traces enables new approaches how to make the road infrastructure management, both inter-towns and intra-cities, smarter. Vehicles and people that are on the move become traffic measurement mediums.

Long before the information era started, spatio-temporal data on human mobility was collected in various forms and modalities. There are various reasons that raise interest in the scientific community for gathering such information. One of the methods used to gather such information is to do straightforward surveys[8, 11]. However these are expensive in terms of manual work needed to carry out and a lot of human resources are needed to gather information. Besides they could only give a snapshot of reality at a given point in time. Generally these type of surveys are done every five to ten years [34]. 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. Given that telecommunications came into the picture and there was a wide adoption of its services at the turn of the millennium one could gather data more frequently in vaster amounts and in an automated fashion from mobile devices. The sample domain got even wider. A limitation which

comes with mobile phone data is the lack of background known on the travellers. Surveys gather such information and make stratified sampling possible in order to have a more representative sample [11].

1.5 Application of mobile traces analytics

Primarily mobile traces would lead to location based services that have a wide application spectrum that go beyond solving mobility issues [18, 8, 14, 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.

[8] went even further to emphasize that such 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.

Mobile device geolocation data surely proved to be useful to setup a platform to predict how traffic/commuting patterns evolve during different time-frames such as weekdays in contrast with weekends [32]. Prediction of traffic patterns would also include jam detection [18]. 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 [32].

In this dissertation we will focus on the topic of measuring traffic flow and predict how traffic increase along time by using mobile data usage. A combination of data mining and machine learning techniques will be used to devise a data processing pipeline. This pipeline will consume raw event data records containing cell tower locations and date time and then it zooms into the main areas of activity of users, plots routes between these areas and collects spatial grid aggregate data from daily trips done along these routes from thousands of users. The dataset which is produced from this pipeline is used to train and validate a predictive model using artificial neural networks.

1.6 Aims and objectives

The problem to be tackled by this research will be traffic congestion detection and also its prediction within a specific time window. Traffic congestion can be measured through aggregate functions exercised on areas with well defined geo-fences. Traffic hotspots' data is more static unlike the location of mobile users which is less accurately traceable. The trip trajectory of an individual when compared with traffic congestion at a given point is far more non-deterministic [21]. Data aggregation of multiple users against time will produce more accurate results when predicting waiting times at a traffic hotspot then when trying to predict the trip for a given time interval and specific mobile user.

Traffic congestion analysis is tightly linked to a long list of factors. These factors are related in some way or another to mobile network determined location. It is not however excluded that the dataset is further augmented. Such factors or 'features' as most often referred to in applied artificial intelligence jargon might include but are not limited to are: number of exits at a junction point, distance from the nearest busy(a standard threshold is to be chosen on what defines 'busy') junction point, aggregate statistics of currently moving mobile users, historical trajectory data for drivers in the area, actual day of week, seasonality (whether it

is a holiday or schools are closed), current infrastructural works which might skew the analysis, accidents records to correlate anomalies etc. Techniques that will be used as a traffic congestion metric is count of moving users per spatial bin at a given point in time[3]. Spatial bins are geo-fenced areas in a rectangular format that enclose geospatial information. Frameworks such as Spatial Hadoop facilitate parallelized processing on large datasets in order to group data points in spatial bins for further analytics [3].

We propose a systematic approach how to address the problem often stated in literature related to mobility patterns [8, 34, 18, 4]. We are aiming to devise an accurate metric of traffic congestion and be able to forecast traffic through a model trained and tested with available mobile usage data. The real challenges resides in achieving granularity when modelling traffic given that mobile usage records' geolocation dataset is sparse and reveals the position of users with a considerable margin of error [18, 14].

1.7 Dissertation outline

This dissertation started with a section that introduces the reader to the vehicular traffic problematic nature. 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 across time. The following chapter "problem definition" will discuss how the problem at hand of measuring traffic and predicting from mobile usage data is not trivial. It will show where the main challenges reside in order to arrive to a viable solution. Background on traffic flow detection and prediction and an overview of related literature will be given in chapter **??** "Background and literature review". The proposed method to showcase selected implementations of certain concepts inspired by literature will be elaborated in the "Methodology" chapter. Validity and versatility of the created model will be evaluated in the

"Evaluation and Results" chapter. Finally the "Conclusion and Future works" chapter will summarize what has been achieved in this work and to what extent. In this chapter shortcomings of the proposed solution will be discussed and possible improvements and prospects for the future will be listed.

2. Problem definition

2.1 Problem statement

From this research it is required to demonstrate that it is possible to attain an accurate measure of traffic and predict traffic for different amounts of time 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 **??**, 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. An algorithm must be devised to deduce the actual path travelled by the user for his most common trips. The predictive model must possibly predict the traffic in a reasonable amount of time since a prediction that take a long time to compute will become futile in its purpose.

2.2 Research Questions

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

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

2. What is the best approach to analyse traffic flow over time in space? Is the resolution of mobile data usage cell tower location fit for purpose to measure vehicular traffic flow on the road network?

3. Is it possible to model traffic flow over time with machine learning techniques that use mobile data usage or processed data derived from it? From 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?

3. 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.

3.1 Mobile location data sources

Mobile location retrieval include various sources. User locations have to be recorded from cell towers mainly for billing purposes. Other records are generated when there is a location update and hand over information [7]. Geographical coordinates of the cell tower are added with billing information. Call data records (CDR) or event data records (EDR) as they are generically referred to when they comprise other forms of activity other then calling 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 decide whether the user may continue to make use of the service. 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 [8]. A call data record 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 technology of mobile data transfer (2G, 3G or 4G) the user device makes use of is negotiated depending on the strength of the signal and load of cell tower [25]. The technology used will fail-over to a less faster one but stronger in signal strength if reception experienced by user weakens (example changeover from 4G to 3G and so forth). 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 with surveys they lack demographics [8, 11] 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 [7]. Many studies mentioned that it is important to remove bias in preprocessing of such datasets before any further processing is done [19, 34]. Passive data gathered in the

form of CDRs are not suited to extract different modes of travel, route assignment and classify detailed activity types [11].

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 on 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 really fast especially because of long signal acquisition time [35]. Thus users would be reluctant to have such service running in the background on their mobiles all the time [1].

CDRs were the most commonly mobile location data source used in recent research [17]. The intention of our research is to use data usage EDRs since these can have a higher temporal resolution. A CDR 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. To our best knowledge up till today the trend is to use CDRs. There are very few research projects that rely on mobile data usage to detect vehicular traffic or predict it. Few examples in literature are [18, 7].

Other sources of geolocation include social mobile application recorded events such as check-ins in facebook [18]. Such data can be accessed by available APIs.

3.1.1 Data format and sample structure used in literature

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 modelling travel on mobile generated data. Hoteit et al utilized mobile data coming from 1 million users between

July and October 2009 [18]. The data consisted of calling and messaging parties' anonymous id together with data of when users make a data connection. Interestingly in [8] together with data collected from the CDRs (a sample of 1 million mobile users in Massachusetts) which contain calling id, time of when call/sms is done or received and when a data session is initiated, vehicle safety inspection data is used as part of the study. This is later used to verify approximately the kilometres covered when inferred from the trajectory computed based on the user data points. Time window is 3 months long and area covered is metropolitan Boston. [8, 11] stressed several reasons why surveys have 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.

In [14] two datasets are used. First sample is of 100,000 individuals sub-sampled from a wider dataset population of 6 million anonymized phone users. Again data used was id of device from which calls or sms originated or terminated and location of tower projected over time. Average area covered by cell tower was 3 km^2 with 30% of cell towers having a coverage of 1 km^2 or less. The other dataset consisted of 206 mobile users whose location was traced every two hours for a week. The second dataset individuated irregular calling patterns noticed in the first one. Then displacements were recorded for consecutive calls in order to construct a distribution model.

[17] use two datasets which have GPS location of 86 mobile users in various places in the world. 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.

Two different examples of tools have been found to be employed to aggregate location data. Airsage datasets were found often to be used in literature [18, 36, 8, 24, 35, 11]. Basically airsage does not simply just record the tower cell sector but depending on a refined triangulation algorithm it gives a more precise location.

[17] makes use of 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 much more on a minor scale than that collected from raw cdrs in other studies.

3.1.2 Mobile position inference from Floating cellular data

The method that makes use of mobile 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 [18]. 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 Inc. Airsage got accurate classification of congested traffic 91% of the time [35]. No technical background was made available how these algorithms get a more precise location of mobile users and this is most likely attributed to the fact that algorithms are patented. It is stated that Airsage location computation accuracy is within 200 to 300 metres in [11]. [8] states that the degree of precision reported by AirSage is an average of 320m and median of 220m. As already aforementioned AirSage has been used in [18] as well. In comparison [14] 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 retrieved from cellular location data would have a maximum error of around 500m.

Therefore antenna/mobile tower location to which the mobile users connect with, may not be useful for all intents and purposes since it 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 local mobile network operators and currently there are no such agreements with the local providers. 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 would prove to be more efficient in accurately measuring levels of traffic an essential topic to treat in traffic congestion research is spatial binning. Spatial binning basically provide geographical aggregate statistics. MapReduce frameworks such as Spatial Hadoop exist so the expensive temporal geospatial analytics are done within an acceptable time window [37, 13]. Such tools would even 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 [3].

3.2 Privacy and data anonymization

Mobile subscribers location is highly sensitive so anonymization has to come into play if legal issues are to be avoided when handling data for research purposes. For instance [23] exposes facts regarding potential privacy breach risks within datasets that have unique identifiers hashed. Methods that adds to anonymization efficiency listed in [23] are contractual binding of data users to not reverse-engineer identity together with truncation of data if in a given area enough data is available. Also [23] gives a detailed account of techniques used to hash unique identifiers. [30] 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.

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 [18] and 2 days in [8]. It is the norm to assign a hashed anonymous identifier to each mobile user in such method as well.

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

Computing systems could not hold the pace of the vast increase in storage requirements [25]. The bottleneck have been always IO reads and while cpu processing power and disk read speeds increased, data volume related to big data problems increased at a faster rate.

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 computation since data will be redundant in other blocks of data replicated on other nodes in the cluster. Replacement of failing nodes is also a smooth operation in big data infrastructure. The main shift of the high performance architectural change was not how to distribute data in the network because this alone would increase network communication latency. It 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.

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 effective? When you have the 3 V's which are volume, velocity and variety in the data its 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 150G 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 [25] 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. [25] might not be that related to the analysis of mobility behaviour but describes well how to process mobile device generated data traffic. 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 [25, 23, 22]. [34] states that dataset size can be an issue for computation when determining the OD matrix. In this same study parallelization is used to assign routes to trips.

3.4 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[19]. Many research articles confronted the problem posed by traffic congestion detection by first deducing the OD matrix [34, 19, 4, 7, 8, 11]. In [4] OD matrices are used to generate trips and hence also give an accurate analysis of travel patterns. in [19] the OD matrix extracted from mobile usage data is scaled up to generate an more representative OD matrix and a simulation is carried out in order to compare with traffic counts readings collected in surveys.

ODs are used to extract main activity hubs. [14] states that 40% of the time users are at their two preferred locations. 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 [7, 11]. In [7] home location is detected for user by checking which 500 metres square cell has the most activity during the night for every specific user.

[11] labels zones such as home and work and tries to find purpose behind other types of trips. [11] mentions how ODs are analysed in terms of stays and trips.

Frequency of calls and time of day determine the labelling of these stays. It was not possible to categorize other types of stays other than home and work. So these types of stays are generally labelled as other. [7] puts forward the concept of virtual location which is derived from fused visited locations by the user. Calabrese devises an algorithm which localizes the centroid of important locations in a user trip that are to be labelled as the origin or destinations of particular users [7]. The method analyses which points are in the proximity of others within a 1 km radius.

A common occurrence in literature is to remove users that do not make enough usage. The behaviour of these is less predictable and its more difficult to generate trips from OD data. In [34] users that do not make enough calls are filtered out from dataset and [11] 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. [19] reduces false displacements 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 than used to detect trips. In [7] a low pass filter is used to minimize localization errors. To reduce sudden movements due to cell tower handover clustering is used. [11] raises the concern in a similar fashion and says that CDR data contains jumps or oscillations which is noisy. [11] mentions 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 long period of time it is possible to get more clear patterns where the stays are actually visited by users or not. [34] removes noise from mobile phone calls deduced trajectories by using the stay algorithm proposed in [39]. A 'stay' location is recorded whenever user makes a set of calls with a time window greater than a given threshold. Centroid is then calculated for set of locations that are close to each other in order to compute a better approximate location of the user. [19] mentions that Estimation of OD

matrices can be unreliable because of sampling bias. Equally [34] 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 statical purposes. Very few information has been found in literature on this. In [11] census tracts and town boundaries are chosen for OD resolution. No justification for this choice is given tough.

Scaling methods are often used to get OD matrices that reflect reality better. In [19] scaling factor Beta is used to get the actual OD matrix for traffic flow. The scaling factor is obtained by inputting optimization formulas, route choice probabilistic models, network data and the OD matrix in a simulation engine. The scaling is then distributed as shown in 3.1

$$OD_{ij} = \sum_{ij}(t - OD_{ij}) * \beta_{ij} \tag{3.1}$$

Problem is then simplified to calculate scaling factor for a set of groups based on dataset analysis rather than for every single OD.

[11] uses the iterative proportional fitting (IPF) upscaling method. 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.

Route selection is tightly knit as a product of OD matrices and when linked they are commonly referred to as OD trips. In [19] route is mostly determined by a function of least travel time path. In [34] Open Street Maps (OSM) which is an open source map building 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. Two consecutive 'stays' that are not more than 1 day apart would constitute a trip [11, 34]. 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. [34] for instance imple-

ments incremental traffic assignment (ITA) in which trips are added to network incrementally. Then on each iteration routes are assigned according to capacity saturation of roads. It is admitted however that WardropâĂŹs equilibrium adapts better since routes are changed dynamically depending on congestion. However ITA algorithm is chosen because it is simpler to implement. [11] relies on probabilistic model for traffic assignment. Departure times for trips are set according to pre-set distribution of departures.

Traffic flow metrics 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 [34]. 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 [34] 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).

Validation related to OD matrix generation is generally done by correlating the generated locations and trips to survey data. In [34] survey data traffic load on road network is compared with that generated through OD matrices formed from mobile CDRs. Simulation generated routes for the latter have been produced with the ITA approach. [34] states however that other methods should be further explored to removed uncertainty from the proposed techniques.

In [19] traffic count was collected on a spread of 3 days in 13 locations and this data was used for calibration of the system. 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 [7] evaluation was done against a tract by tract census. Euclidean distance was calculated and the distribution of the trip distance confirms Gonzalez affirmation that trips follow a random walk [14]. See equation **??**

$$P(x) = (x + 14.6) - 0.78^{-x/60} with R2 = 0.98 \tag{3.2}$$

Here euclidean distance added error and to portray visually, 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 [7] 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.

[11] carries out validation against traffic surveys and already available OD matrices from department of transportation. However only morning sample was used for validation. [11] boasts 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. [11] reports OD matrix limitations. Suitability of CDRs to determine ODs is only good at a certain resolution. Best to have 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.

3.5 Graph databases and Parallel graph processing

In traffic congestion research cross-sectional snapshots of data are of little use. Historical data hoarding is imperative in order to chisel out patterns of commute and classification of traffic patterns in every region within a set of given boundaries.

Nowadays the data encountered in many IT systems' scenarios got too voluminous in such a way that normal traditional RDBMSs could not handle any more in terms of both model and performance. NoSql entered the scene in the last ten years to cater for new challenges and together with Big data it helped to address issues such as requirement to store unstructured and semi-structured data in a schema-less form, need for high-scalability, low-latency and high performance. NoSql however has its drawbacks as well. Most of the NoSql solutions do not support ACID transactions in order to keep data consistent for example.

Graph databases use native storage and native graph processing. They were designed to process data mining related to graph better than relational databases. Relational Databases are most suited for problems that are well defined at the start of a project. A clear sign when to use graph database is when designing the schema it appears that a lot of joins will be needed. Graph-parallel computation gives an edge when processing is shifted on each data node of the graph. Referential integrity although useful for data integrity as the name implies comes at a dear cost. Queries and data manipulation that involves joins are slower. Mining of highly relational entities such as mobile users location in the traffic network and the streets map make graph databases such as Neo4j an essential tool. Destination matrices should also be stored in Graph databases. Offline graph processing frameworks such as Giraph or Spark Graphx might then be useful to carry out scheduled jobs for collating information such as shortest paths and estimations of travel duration from any point A to any point B in a map grid.

3.6 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 [18]. These are namely linear-interpolation, nearest-neighbour inter-

polation 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 the recorded data sample are distant in what is time interval. As for the nearest-neighbour method location is placed to the previous recorded value or to the subsequent depending which is nearest on the time axis. The cubic interpolation is best explained when contrasted with the linear one. This method as perfectly stated in [18] 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 [14] both the variation of displacements for consecutive 'steps' (call location) 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. 3.1 and equation 3.3 for illustration of displacement distribution modelling).

$$P(\Delta r) = (\Delta r + \Delta r_0)^{-\beta} exp(-\Delta r/\kappa) \tag{3.3}$$

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

This mathematical model is cited and verified in [8]. Methodology adopted in [17] suggested approaches how to determine home and work locations, span of movement and complete trajectory. Two datasets were compiled. The second one is a sub-sample of the first which is 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 [18, 14]. As for the actual movement trajectory error was calculated by calculating the euclidean

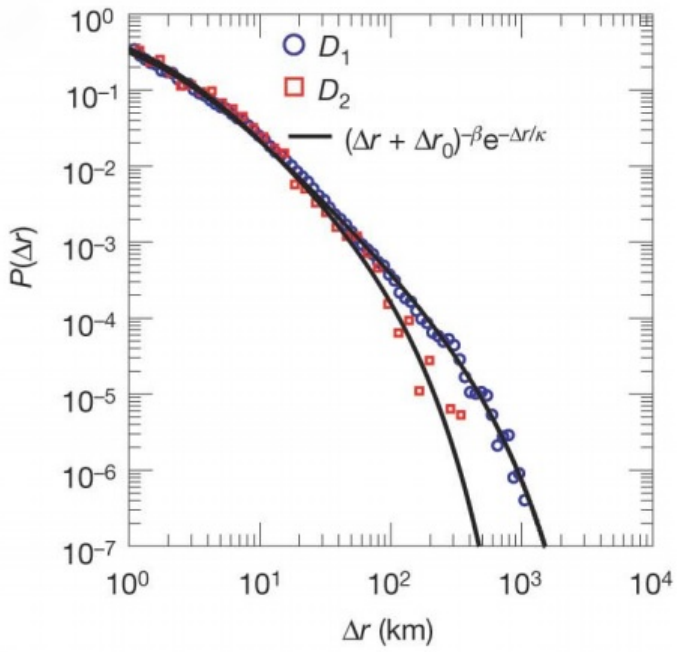


Figure 3.1: Truncated levy flight human motion modelling. Reproduced from [14]

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”.

An issue have been raised in [8] about detecting a lot of trips in very short distance which do not tally with 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 [8] ’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 limits static location detection to the home location and the 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 highway exits. These urban features were derived from US Census of 2000 and activity travel surveys.

Estimation of load error is proportional to concentration if users in a given block [18]. When error is 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 the probability of being inside radius of gyration given error is high at 40.25% and that of being outside is 59.75%.

[8] boasts of 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 [14] 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 [18]. 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 [18]. Interpolation methods are found to be most suited depending on distance from centre of mass. 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 are contributed in [8] 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 radius. 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. [17] affirms 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. [18] segments mobile users depending on how much stretched is the radius of gyration (r_g). The different distinguished categories of users are listed as sedentary, urban, peri-urban users and commuters. Classification boundary was decided upon steep changes in the cumulative distributed function of the radius of gyration. Respectively they fall in the ranges $r_g <= 3km, 3km < r_g <= 10km, 10km < r_g <= 32km, 32km < r_g$. This radius of gyration (see eq. 3.4) 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 [18, 14, 17].

$$r_g = \sqrt{\sum_{i=1}^{n} (\vec{p}_i - \vec{p}_{centroid})^2} \tag{3.4}$$

where

$$\vec{p}_{centroid} = \sum_{i=1}^{n} \vec{p}_i \tag{3.5}$$

In [17] the hypothesis that an individual tends to be found with high probability at his home or place of work makes the authors to come out with so-called 'stop-by' categories. The stop-by categories are stop-by home which is demarcated by the

night time interval where a user is expected to be at home. stob-by-flexhome is a refinement over and above stop-by-home were night time interval varies per user. Stop-by-spothome fills data lacunas or corrects errors when there are exceptional errors where user is expected to be in home location as indicated by previous category.

3.7 Prediction methods

works in progress

Prediction of traffic results have to take in consideration where the model is being used for forecasting. [31] states for example that it is easier to predict traffic in highways rather than in urban areas since traffic tends to be smoother.

4. Methodology

One of the main objectives 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 construct an algorithm to use in order to map raw data and translate 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 would have proven to be too sparse, both temporally and spatially, augmentation of mobile usage dataset by tapping into other datasets would have been necessary. 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 information on human mobility patterns. In this chapter we will elaborate how we investigated certain approaches and how we decided to dig deeper with some chosen techniques rather than others depending on how practical the solution was and how it would give a better result.

4.1 Mobile data collection and structure

The dataset was provided by GO Plc Malta which is one of the main data providers. 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. 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. Number of distinct cells amount to several thousands but distinct cell locations amount to only a few hundreds since a cell tower shares antennas for different technologies. Precise figures cannot be disclosed due to commercial sensitivity.

As stated by the provider of the dataset, EDR/CDR records are generally buffered to file on the network element. Files are closed periodically, generally every 1 to 5 minutes. 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. It normally takes around 10 minutes for an EDR/CDR to be within the data-warehouse, hence available for further processing. However mobile operator engineers stated that polling frequency can be increased as needed. A smaller polling interval would allow to have predictions closer to reality.

Unique data usage users amount approximately to 108 thousand for the month of October 2016. This month was chosen for its' heavy traffic characteristics because schools start in their full sway and university students start to travel with their cars adding to the load of traffic. This sample would be roughly representing half of the provider's subscriber base which is just over 200 thousand. This figure was derived from all the distinct users that make calls or use data. Therefore the number of data users that are in the sample is roughly half of the whole set of users. One must take note however that all these data users in this study might not necessarily contribute in a directly proportional manner to the number of vehicles on the move at a certain point in time. There might be 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. All these facts must be taken in

consideration when setting up the proposed solution and evaluating results. The records' data structure is shown in table 4.1.

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 4.1: Data Dictionary of mobile usage raw dataset

4.2 Dataset preliminary analysis

Total number of records of any given type for the month of october 2016 was 125 million. 78% of these records are data usage records. This gives an indication that there is a four fold higher frequency data usage type record generation when compared with calling data records generate rates. 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 would reflect better spatial resolution and both of these in combination will conduce to better results both when measuring traffic flow counts and when extracting main user activity hubs. Lower sampling rates lead to interpolation error.

Table 4.2 shows some summary statistics about the main unprocessed data set. Minimum and maximum timestamps show that data stretches from the very first minute to the last of the month being analysed. The total count of data usage

records is 97 million. The basic data session duration mean 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 real time immediate 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 4.2: Basic summary statistics of main EDR dataset.

When a frequency diagram is plotted an interesting fact comes out (see figure 4.1). 15% of the EDRs have a data session duration value 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 exercise 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 removing 1 hour duration EDRs newly calculated summary statistics show that the mean and standard deviation are decreased down to 8 minutes and 14 minutes respectively. This is a 50% gain with respect to previous statistical

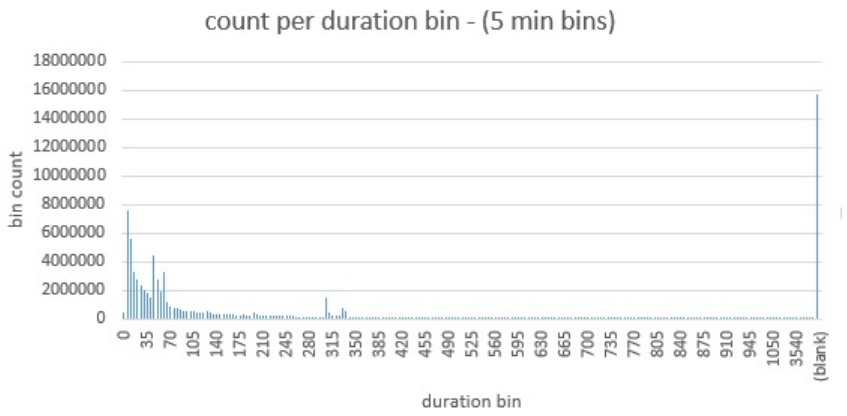


Figure 4.1: At the right of the plot one can notice a spike of records on with data session duration of 1 hour.

data. Further looking at the data, by overlaying a cumulative distirbution it is shown that 80% of the records are below the 5 minute mark. These data facts have to be all taken in consideration when assessing the usefulness of the prediction results in evaluation and results chapter 5.

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

4.3 Algorithm Selection

One of he most challenging tasks involved in this study was to assign vehicular traffic to the road network depending on surrounding cell tower traffic in a time series.

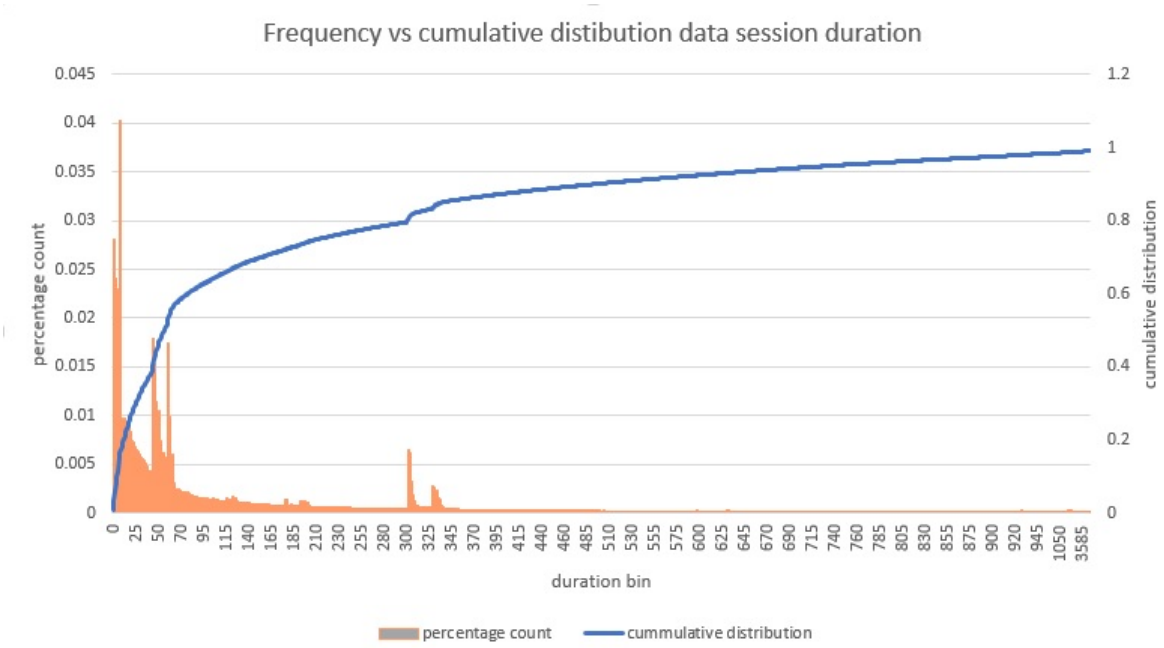


Figure 4.2: 80% of the records are below the 5 minute threshold.

4.3.1 Trajectory interpolation through cell tower data

The first approach that we tried to investigate was to relate the cell tower data by snapping it 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[1]. We had no prior knowledge on how cell tower transmission is configured. There are many settings which might determine the range of the cell tower including frequency, rated power, height of tower etc. Given that such information was missing transmission range of the cell towers was an unknown variable. A possible alternative to roughly estimate range was to calculate the average distance from the nearest k neighbouring towers for all cell towers. However from a sample taken from the dataset available 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 for all the country not being shown since it is sensitive commercial data). Further

[1]Google Snap to Road is an example - https://developers.google.com/maps/documentation/roads/snap

to this from the analytical perspective 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 taken as an assumption that the area around cell towers will be be given service with equal range from each tower. Allowance for overlapping was also taken in consideration. It is evident from figure 4.3 that a lot of roads would be included in the range of a cell tower so it would have been 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. Such an approach would might have made it too impractical to assign traffic to junctions, roads or polygon areas and the probability of inaccurate results was high.

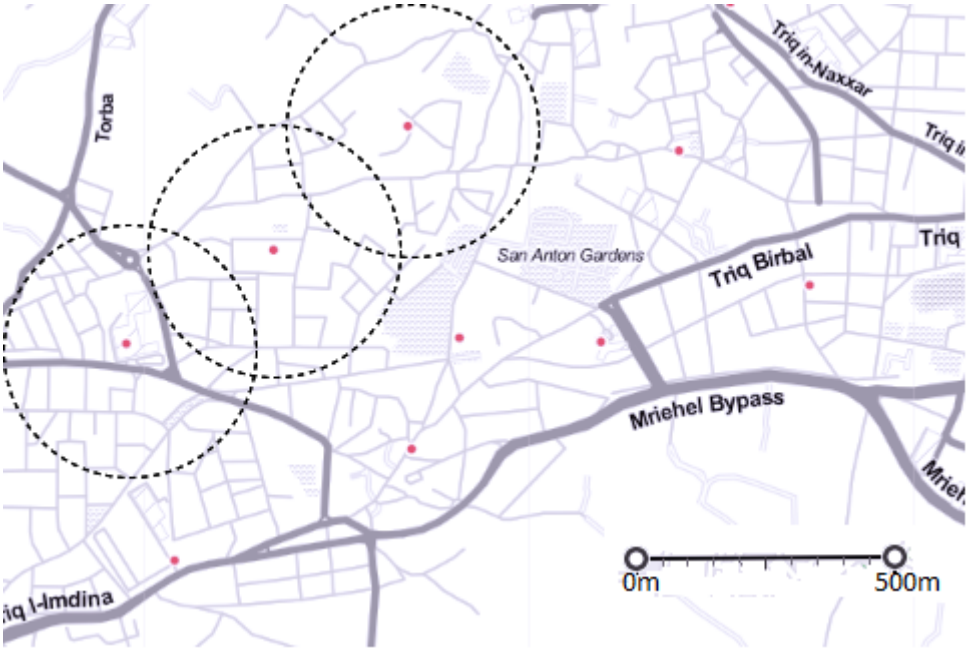


Figure 4.3: Cell tower range distribution. Red dots are cell tower locations and dotted line is the estimated range.

4.3.2 Traffic simulation from OD matrix

Another approach would have been to simulate traffic by inputting statistical information on travel patterns retrieved parameters from the dataset. Possible input to the simulation based traffic model would have been an OD matrix that will be discussed in section 4.4. Toledo et al. [33] mentions that OD flows are an important input to simulation models but an accurate OD matrix is difficult to acquire. An example of such implementation of a simulation based on an OD matrix can be found in [15]. In this study electronic toll collection data is used to form an initial OD-Matrix. This OD matrix is further optimized 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 are found to be less suitable for urban scenarios [6]. 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 which cannot be achievable from our dataset due to displacement error created by cell tower location and actual location inevitable considerable distance difference. In our research we tried to focus on both the macroscopic and on the microscopic level since we had a dataset that have ample coverage at our disposition especially in urban areas.

4.3.3 Traffic flow detection by trip generation assigned traffic

The method chosen to detect traffic on the road network was to first generate 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. The trip would include 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. The major challenge here would prove 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 for whatever reason.

4.4 Main activity hubs extraction and displacement error removal 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. One possible way to do this from the described dataset is by clustering. Clustering is also used to remove noise caused by displacement through frequent oscillations by finding a centroid of activity. Removal of displacement error through triangulation has been ruled out. There are missing dataset features that are needed 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 **??** we have seen 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 checked for their appropriateness to this research were k-means and DBSCAN.

4.4.1 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 its requirement to select the number of clusters you need to find 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.

4.4.2 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 of non-spherical nature and leaves noisy elements out of the computed clusters [10]. 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 form one single cluster, hence the non-spherical shape of the clusters.

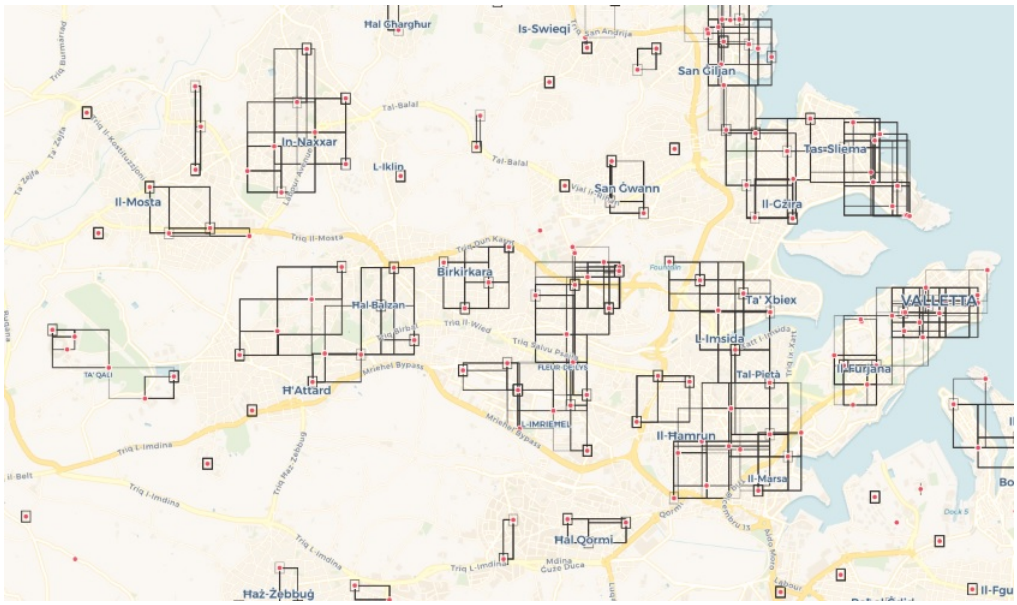


Figure 4.4: DBSCAN clustering to find main user activity hubs. (Sample illustration)

DBSCAN was the preferred candidate for clustering since our aim is 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 over-

lapping 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 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[2] 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 would denote the main activity clusters would be those of the centroid. Therefore the centroid for each rectangular cluster had to be determined with readily available libraries (esri was the used library)[3] within Apache Hive.

4.5 OD Matrix trip Generation

To simplify and present in a summarized way 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.

4.5.1 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. Inclusion of more areas of activity is left for future studies (refer to chapter 6). 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 3.4). The top two clusters per user

[2]https://github.com/scalanlp/nak

[3]https://github.com/Esri/spatial-framework-for-hadoop, https://github.com/Esri/geometry-api-java

where retrieved from the resulting users' clusters created with DBSCAN algorithm run. 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. This process will produce 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.

4.5.2 Trip generation, route choice and traffic assignment

OD matrices on their own would not give information how traffic flow is distributed on the roads. We needed to detect 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. The computations done depending on previous and current rows while scanning a dataset were made by using Apache Hive window analytical functions [4].

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 would have a direct effect on the accuracy of departure and arrival times for any given trip. The more the temporal resolution is high the more accurate are the departure and arrival times since if the record is generated at a low frequency it cannot be determined with confidence and low margin of error. For example the user might arrive to his work location but he takes too much time to start his first data session. This would add extra trip delay and the resulting trip duration 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

[4]https://cwiki.apache.org/confluence/display/Hive/LanguageManual+WindowingAndAnalytics

with a narrower error margin. Departure times are more precise since these are computed by adjusting the EDR timestamp by adding the data session duration. This will give an accurate timestamp that denote when users leave their clusters.

As similarly done in Toole et al. [34] routes between origins and destinations were inferred from OSM. 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[5] (OSRM). It is 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. This is a limitation of our 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.

When determining the daily users' routes between different origins and destinations done there are two sets of important information that are derived. These are namely trip count and trip delay per route. Distributions for both were further derived given that information such as trip departure timestamp is available for every route. One can notice how the trip distribution derived from mobile usage EDRs (see figure 4.5) generated OD Matrix is strikingly similar with the trip distribution as reported in a National Travel survey done in 2010 [26] (see figure 4.6). The figure reported in [26] does not include Saturday and Sunday trip distributions.

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

[5]http://project-osrm.org/

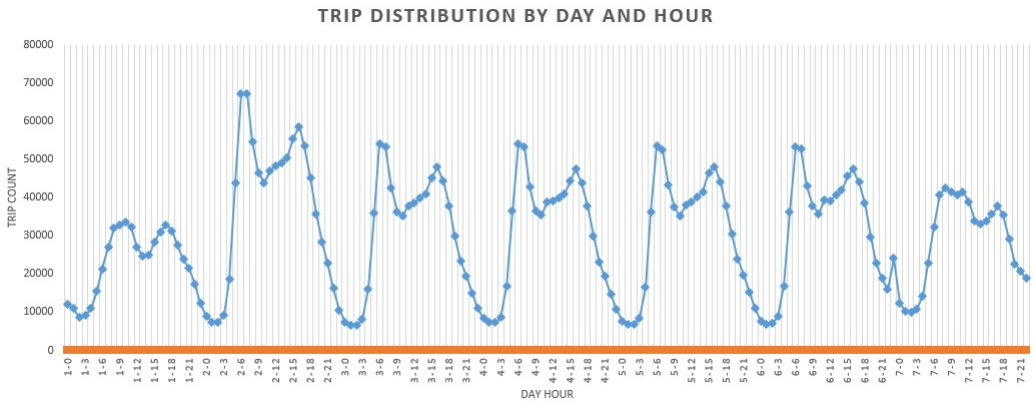


Figure 4.5: Trip delay patterns per day week. To note how Saturdays and Sundays which are outlined by the last and first two peaks respectively have unique traffic delay patterns.

difference is evident even between Saturdays and Sundays but is highly regular for weekdays as seen in figure 4.9. 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 are 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. Only one peak can be observed for weekend days. 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 4.8. 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 less than -5 minutes were mainly attributed to location sudden

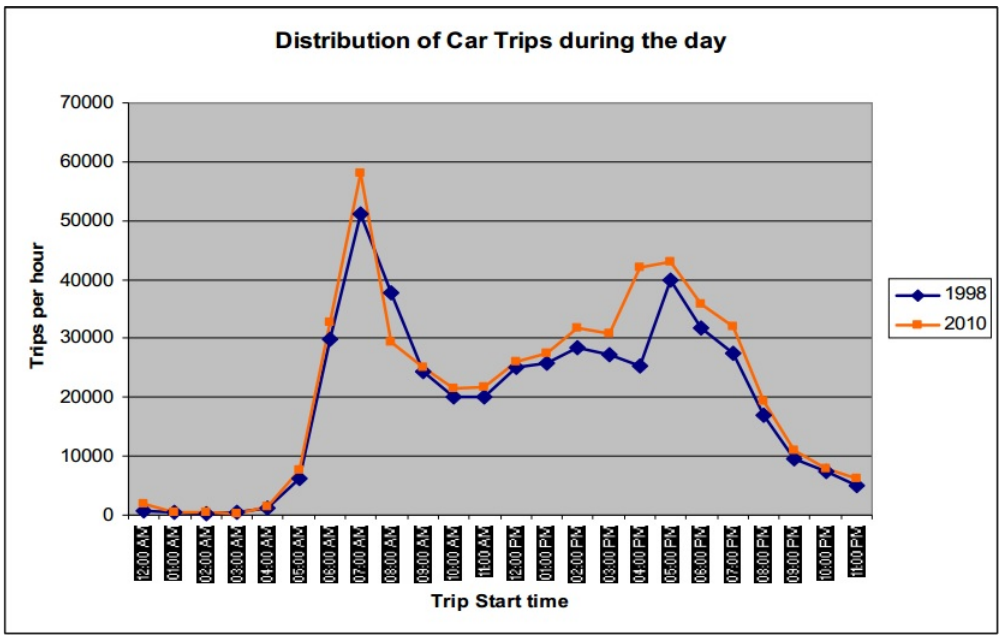


Figure 4.6: Trip distribution as reported in Transport Malta 2010 National Survey [26].

displacement caused by cell tower switching (see section 3.4 in chapter 3). 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 technical jargon that describe map objects the route consists in steps that contain manoeuvres. These manoeuvres fields encapsulate geographical co-ordinates data and duration property after which the driving decision should be taken. The manoeuvres' timestamp was computed by accumulating previous steps duration and add 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.

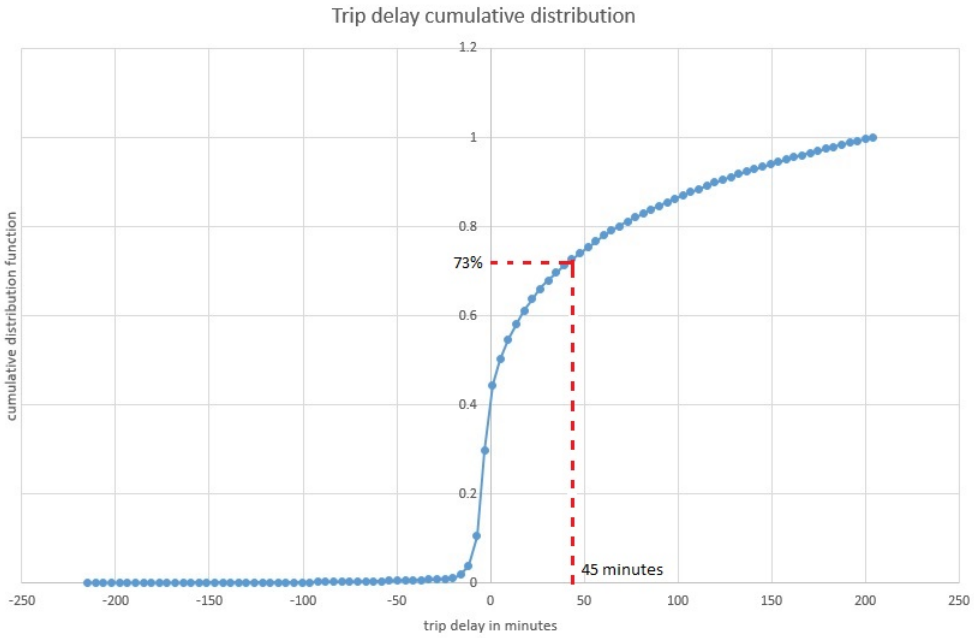


Figure 4.7: Cumulative distribution of trip delay. This figure shows how negative trip delay instances are a very small percentage. Cutoffs of -5 and 45 minutes were chosen to select the trips for the learning model.

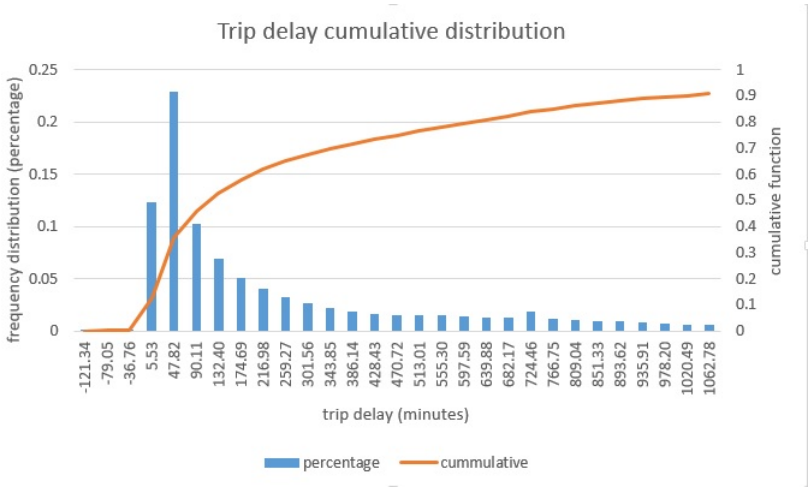


Figure 4.8: Trip delay probability distribution presents a heavy tail on the right.

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

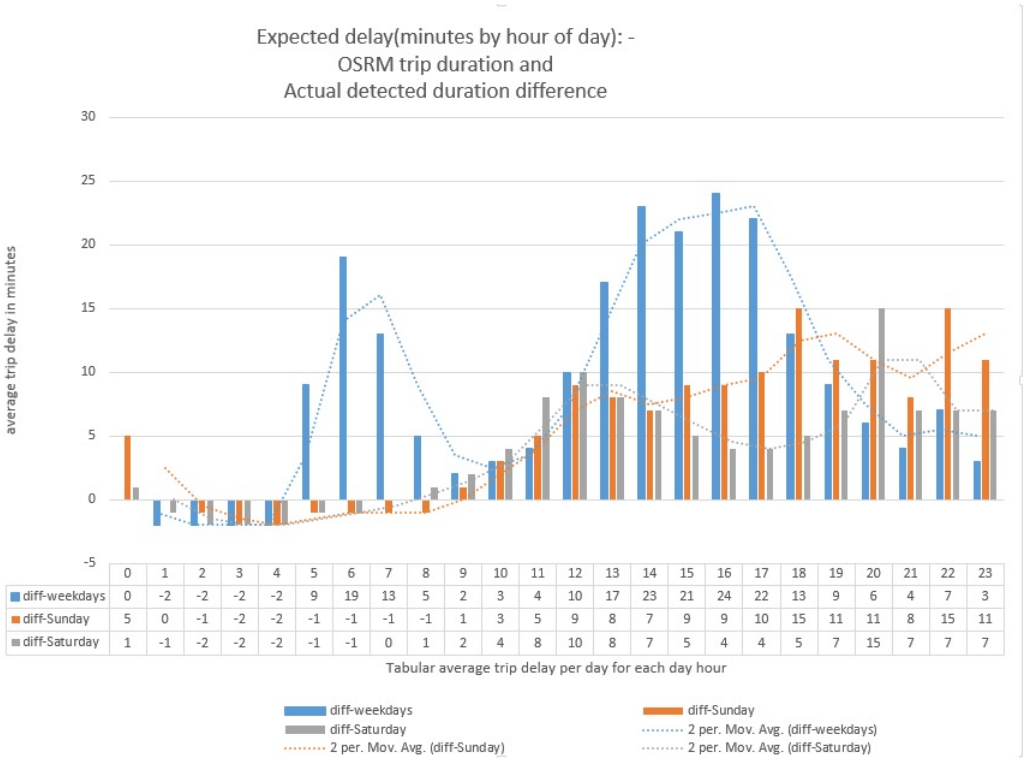
D. OD Matrix based trip generation:
    for each user departure arrival record *udar* in *data_user_departures_arrivals* **do**
        if *udar departure flag* = true **then**
            udar route ← derive OSRM route from *udar origin,destination coordinates*
            udar OSM route duration ← 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*

29: ***F. Traffic flow prediction***:
30: *data_distinct_time_window_end* ← get distinct time window ends from *data_bin_traffic_flow_time_series*
31: *data_distinct_bin_ids* ← get distinct bin ids from *data_bin_traffic_flow_time_series*
32: *data_distinct_time_window_ends_bin_ids* ← *data_distinct_time_window_end* cross join *data_distinct_bind_ids*
33: *data_sparse_bin_count_time_series* ← *data_distinct_time_window_end_bind_ids* left join *data_bin_traffic_flow_time_series*
34: *data_time_windows_bin_count* ← two dimensional pivot on *data_sparse_bin_count_time_series* by bin_ids
35: *data_bin_count* ← *data_time_windows_bin_count*
36: *sample_locations* ← *location_array[a,b,c,d]*
37: *window_frames* ← *window_frames_array[30 min,60 min,180 min,1 day]*

38: **for each** each bin for location *bin-loc* in *sample_locations* **do**
39: **for each** global prediction at *t-ahead* time ahead in *window_frames* **do**
40: **for each** each record with bin counts *bin-count-record* for time *t* in *data_bin_count* **do**
41: *t-ahead-bin-loc-count* ← get bin count for *bin-loc* at time *t-ahead*
42: *bin-count-record-with-label* ← attach *t-ahead-bin-loc-count* to *bin-count-record*

43: *data_labelled_points* ← store final resulting dataset
44: *data_labelled_points_reduced* ← take first 1000 components of PCA dimensionality reduction of *data_labelled_points*
45: *data_training* ← split *data_labelled_points_reduced* and get 60% of data
46: *data_testing* ← split *data_labelled_points_reduced* and get 40% of data
47: *multilayer_perceptron_classifier_model* ← fit model on *data_training*
48: *multilayer_perceptron_classifier_prediction_result* ← run model on *data_testing*
49: report result metrics



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

4.6 Traffic flow aggregation through spatial binning

To get aggregate statistics on traffic distribution hadoop spatial binning was used as proposed in (Eldawy et al,2015) [13]. Hive, which is a data warehouse infrastucture tool running on Hadoop, abstracts a lot of java api calls to get data from distributed file systems managed by Hadoop [6]. Hive has a specific SQL dialect HiveQL(HQL) that can retrieve data from hdfs (hadoop disributed file system) without implementing the mapreduce calls.

Another used tool was Spatial Hadoop which is highly efficient to process geolocation data because it uses MapReduce [7] and a 2-level spatial index. MapReduce

[6]https://en.wikipedia.org/wiki/Apache_Hive
[7]https://hortonworks.com/apache/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 [13].

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 count traffic flow by spatial bin[8]. 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[9]. Not withstanding this the bins in the spatial area under investigation are of the same size since Malta does not relatively cover a wide area. We chose $50m^2$ spatial bins to aggregate traffic flow data in order not to have too much wide geometries that can aggregate traffic coming from two roads. Smaller bins than 50 metres square make aggregations less meaningful since aggregation is more 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 within intervals of 5 minutes each. This choice of interval 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. Prediction of traffic could then be more practical in terms of time windows ahead relative to the prediction to be done. A larger time window would not permit finer prediction in a time series. For example a non-sliding 10 minute time window which would allow predictions 10

[8]https://github.com/Esri/spatial-framework-for-hadoop
[9]http://www.longitudestore.com/how-big-is-one-gps-degree.html

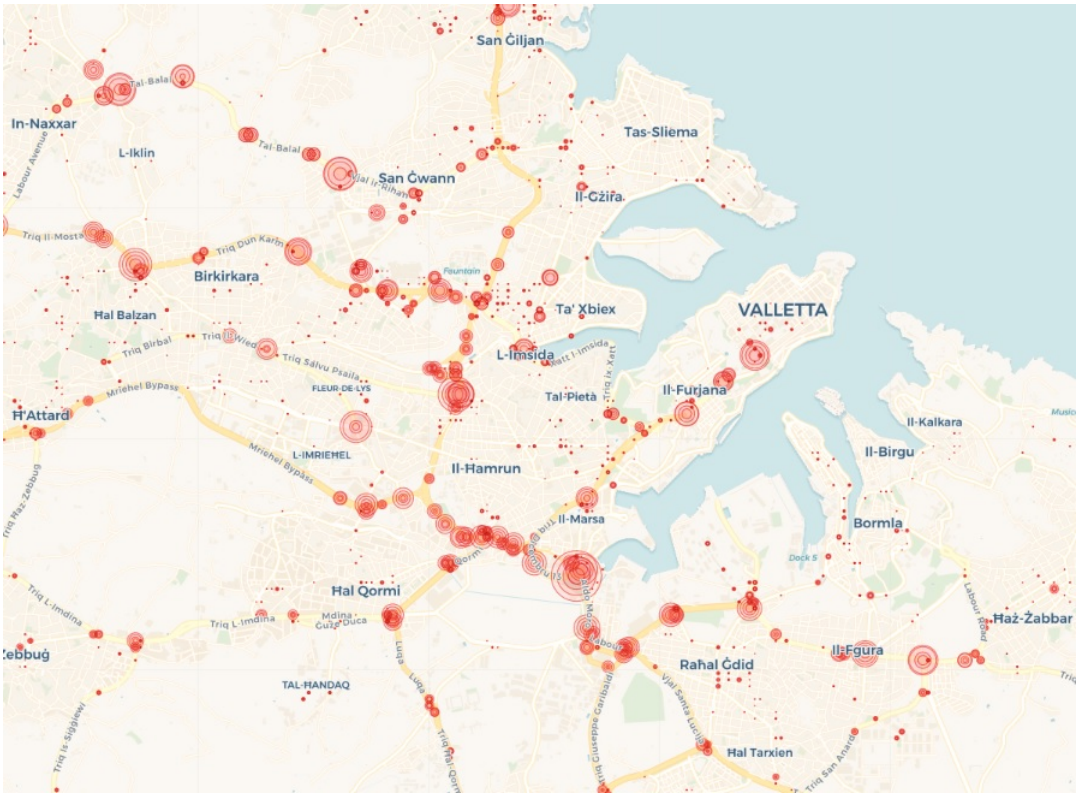


Figure 4.10: Traffic flow count through spatial and temporal binning.

minutes ahead, 20 minutes ahead and so forth. We did not use a sliding window so that we have 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. Having a sliding window would make 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[10] where used to illustrate aggregation of traffic flow count in spatial bins. Figure 4.10 shows which areas attract most traffic in Malta. Well known to be busy traffic flow Locations such as Santa Venera tunnels and Southern Harbour area around the four lane roads in Marsa have bigger distinctive spherical markers. The tool allows to select specific date and time to analyse traffic temporally. A similar visualization depicts average traffic intensity

[10]https://carto.com

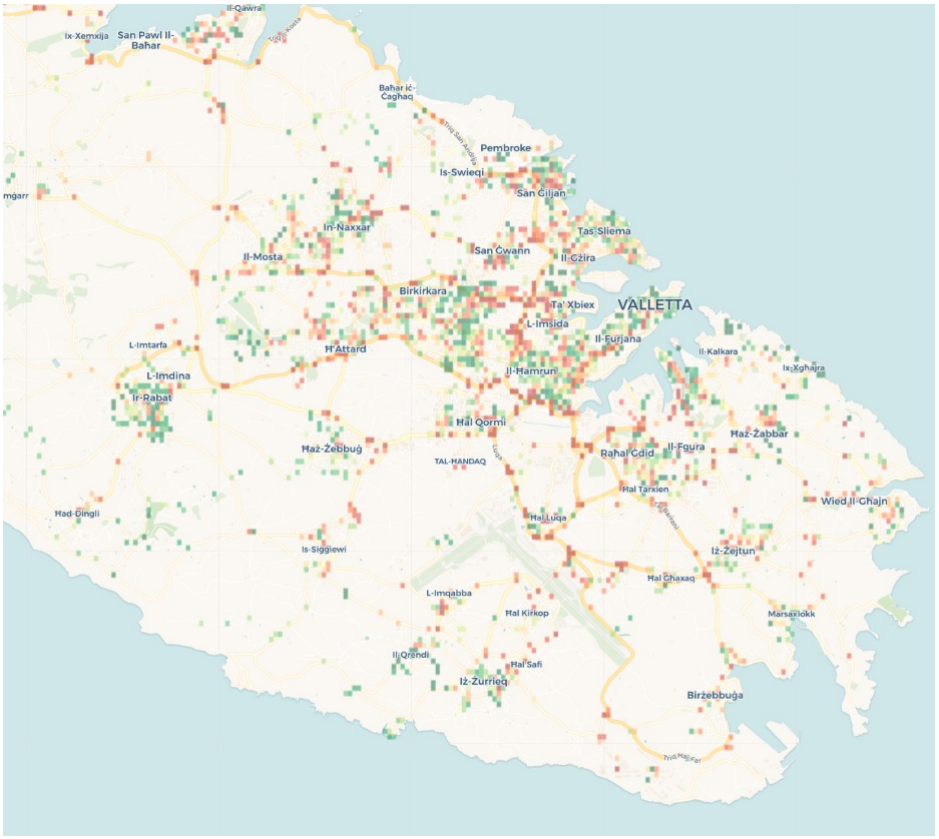


Figure 4.11: Traffic flow count categorization through a colour coding. Traffic flow count intensity is represented with a colour scheme ranging from dark green (low recorded traffic) to dark red (heavy traffic). The range stretches on 11 quantiles.

across the whole month under investigation. The tool allows to configure pop ups that can display relative information to the bubble such as location coordinates and average month traffic.

Another illustration (see figure 4.11) shows through a colour scheme in a categorical manner the intensity of traffic flow. This method makes it easier to categorize traffic flow count than the method used to display traffic in figure 4.10.

4.7 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

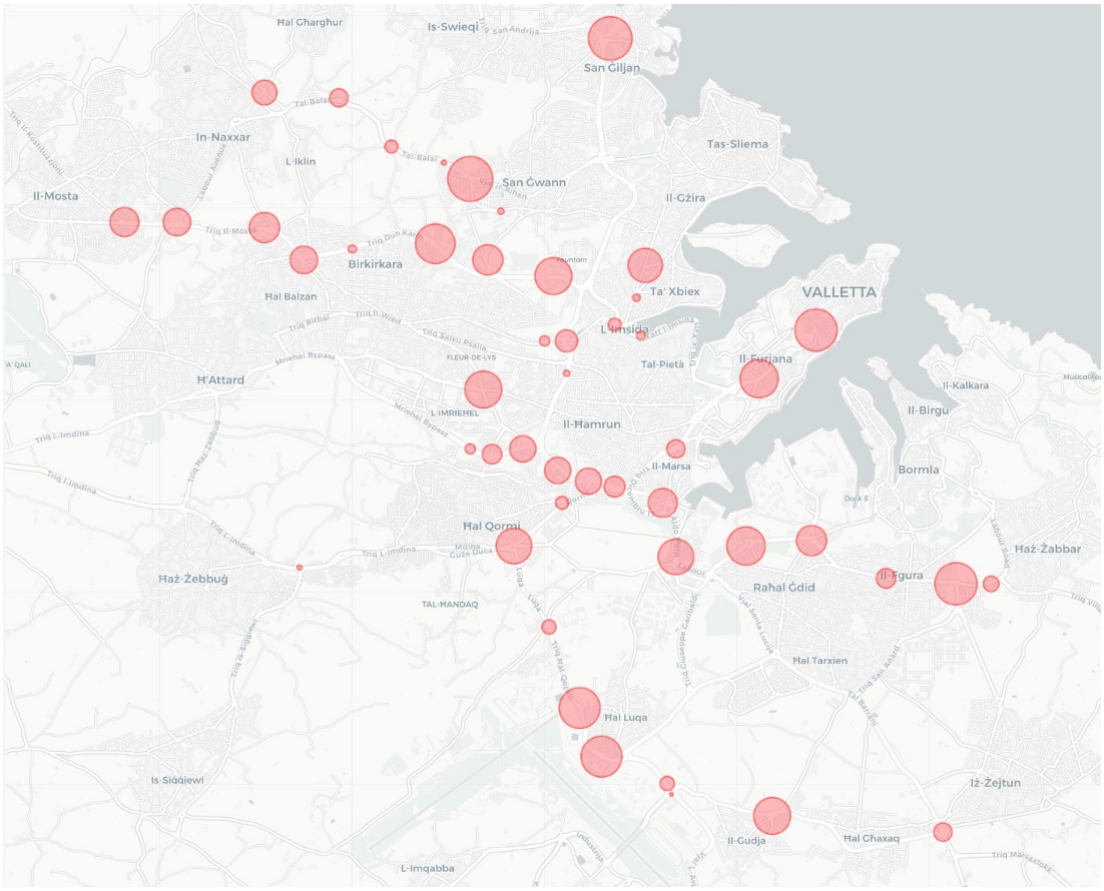


Figure 4.12: Traffic flow count through spatial and temporal binning.

model has been structured. More specifically traffic flow at any particular bin b_i at 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 count from other bins in preceding time. The computation of traffic in a location based on traffic in the past cannot be achieved by deriving a function based on a mathematical approach. Black box predictive modelling was obtained by training a neural network. 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 training of the model, how the model hyperparameters were chosen, how the model was eventually trained and then how the model was validated.

4.7.1 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 4.4). This dataset was then ordered by time. This dataset was to be derived from the generated dataset in subsection 4.5.2 which has aggregated traffic flow for each user in a time series with 5 minutes bins. The final resulting dataset from preprocessing would be 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 4.6.

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 in 4 labels since traffic is labelled similarly in available traffic applications such as Google Maps Traffic [11] and Tomtom navigation software [12]. The assigning of the label consisted first of classifying the traffic count of the bin location for every 5 minute window for which prediction modelling is to be carried out. Then the resulting classification is 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, table 4.5 would have another column with future classification of traffic count for a given bin location. Four datasets were prepared for training. These datasets were differentiated by the label assigned. The classification label for each dataset record would be retrieved from time $t + 5n$ ahead through window analytical functions for n with values of 3, 6, 12 and 288 which given the 5 minute coverage of each record they reflect time windows ahead of 15 minutes, 30 minutes, 1 hour and 1 day respectively.

This dataset had traffic flow count for each bin with 5 minute temporal reso-

[11] https://www.google.com/maps
[12] https://mydrive.tomtom.com

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

lution. Traffic counts of zero were not yet present before preprocessing. Therefore the main features of this dataset would be bin id, traffic flow count and time window start and end timestamps. In order to generate a dataset with records that give a snapshot of all traffic count of Malta for every 5 minutes further processing was needed. First all distinct bin ids where extracted and these amounted to 4134. Then all the possible time windows of 5 minutes in the month of October were generated and these amounted to 8928. By performing a cross product between all time series values and all possible bin ids a new dataset with all possible bin id and time window combinations is 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 input to the neural network training and further reorganization was necessary. A data structure 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 attained data with format as shown in table 4.5 from data with format as illustrated in 4.4.

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 4.5: Sparse traffic flow matrix

After all the feature data have been organized in a format that can be processed as input data for the model a label for each data row had to be assigned. ANN is a supervised machine learning type which requires output that can be mapped from input data during the training phase. The output is in our case 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 a logarithmic function thereof (see Eq. 4.1).

$$y = \lfloor log(x + 1)/log(max_x/n + 1) * b \rfloor + 1 \tag{4.1}$$

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).

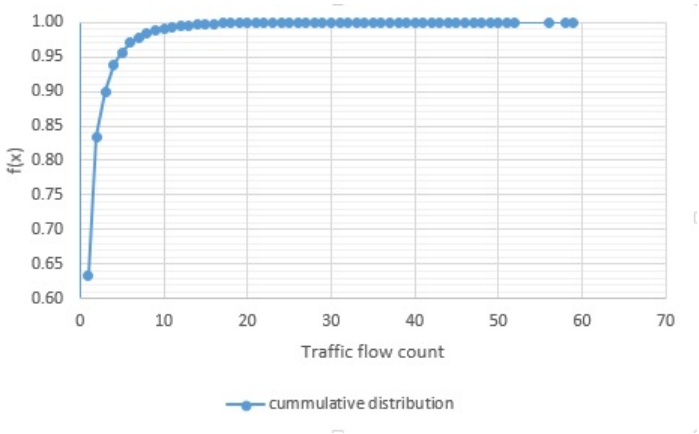


Figure 4.13: Traffic count cumulative distribution for a chosen location.

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 fig. 4.13). The skewness towards low counts of traffic is highly decreased with logarithmic binning. The use of logarithmic 'step' function defined in equation 4.1 squeezes indicators in the low traffic flow label bin and widens the range for high level traffic. Note how in figure 4.15 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 which proved to give better results for prediction evaluation.

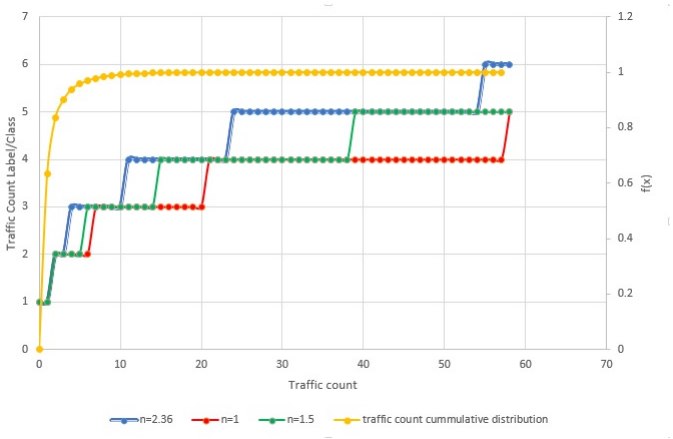


Figure 4.14: Traffic count logarithmic step function.

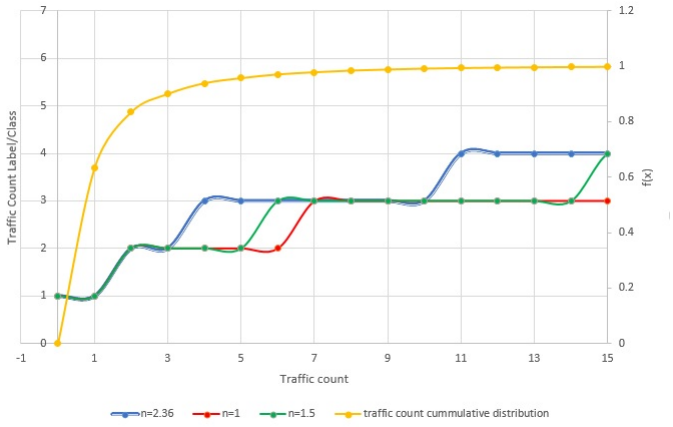


Figure 4.15: Traffic count logarithmic step function. Lower traffic count mapping.

4.7.2 Dimensionality Reduction

The dataset acquired from the original data usage mobile is huge. The features that describe the data amount to 4134 as already aforementioned in subsection 4.7.1. The planned computational complexity depends 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 depends on 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 is 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 [20] and other literature on 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 is 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 practical in terms of getting a prediction after an acceptable amount of time.

One way how to lower computation time is 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 [28] states that by reducing dimensionality in data there is less risk of overfitting and thus model can generalize better to testing data. In [38] 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 4.2

$$\min_{k} \left\{ k : \frac{\Sigma_{i=1}^{k} \lambda_i}{\Sigma_{i=1}^{n} \lambda_i} \geq r \right\} \tag{4.2}$$

we selected k to be 324 so that the preserved variance ratio r is 90%. In equation 4.2 eigenvalues λ_i are ordered in decreasing variance. n represents the original dimensionality of the reduced dataset. In figure 4.16 it can be observed that the first 324 components explain more than 90% of the data.

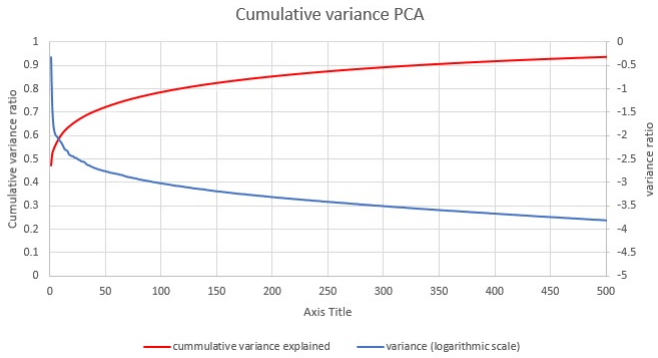


Figure 4.16: First components contain a higher percentage of the variance. First 324 components explain 90% of the variance

4.7.3 Prediction through Multilayer Perceptron Classifier (MLPC)

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 is harvested some time ahead. All of the original dataset 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 different first PCA components, prediction multi-steps ahead and number of possible classes.

As already explained in subsection 4.7.1 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 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.

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. MLPC is a highly non-linear model that can adapt to problems with high complexity. 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 [16]. 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 [13].

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 [14].

The topology choice was determined after carrying out a grid search configured with a set of different hyperparameters. Different configurations for architecture 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. The final configuration after testing different possible architectures consisted of 324 neurons for both the input layer and for the second

[13]https://spark.apache.org/docs/latest/ml-classification-regression.html#multilayer-perceptron-classifier

[14]https://dzone.com/articles/deep-learning-via-multilayer-perceptron-classifier

hidden layer, 400 neurons for the first hidden layer and 4 neurons for the output layer which defined the output classes (see figure 4.17). All architecture layers are fully connected to the successive layer.

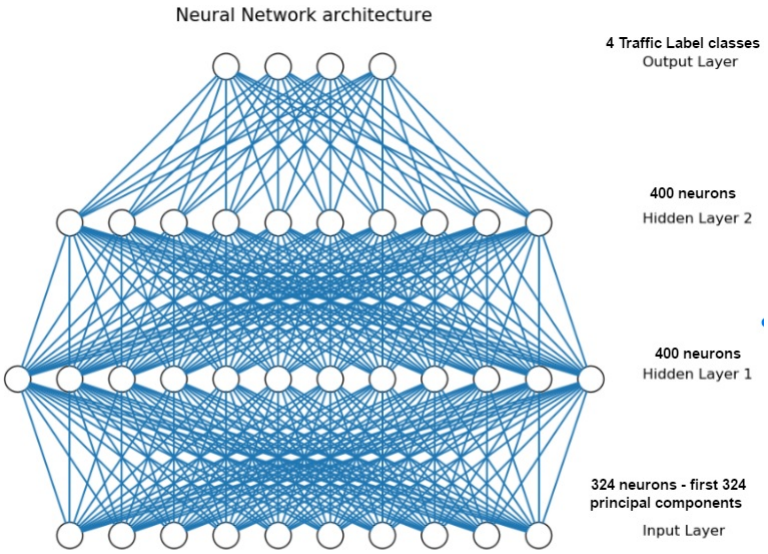


Figure 4.17: Multilayer perceptron classifier topology

The available labelled data points dataset was split into 60% training and 40% training. 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 fit on the testing data. In the testing phase the prediction efficacy of the model built during training is checked by retrieving certain metrics.

5. Evaluation and Results

5.1 Section Name

Mobile users averaged location calculation, estimated path trajectory and predicted traffic congestion points are basically the targets aimed for in this research project which have to be evaluated. The pivotal point here is to have a ground truth to be able to evaluate properly the obtained results. As already aforementioned in section **??** this ground truth can be gathered from tailor made applications that collect GPS data from voluntary users. Other data sources to contrast with are actual travel diaries taken by users and traffic counts compiled from video camera captures. All the three areas required to be evaluated need to have a uniform way how to positively assess as a good prediction or bad prediction. A way how to do is is to break down the used geographical map in a grid of arbitrarily placed cells with a specific stipulated resolution which should not be neither too big and nor too small. A root mean square cost or a cross-entropy cost function may be used in order to calculate how off-mark is the prediction when testing. A confusion matrix would be useful to visualize in a tabular fashion were the models are getting it wrong in terms of particular grid cells. Metrics used for evaluation could include an F-Measure which is a summary statistic of precision and recall and is parametrized in such a way to give different importance to precision and recall as required.

Finally after training with a batch of data a traffic flow prediction model is

created for each particular spatial bin in the chosen sample. The different locations chosen where

6. Future Work

6.1 Section Name

7. Conclusion

The approach taken is systematic so that the research passes through gradual stages in such a way that we build on top of previous analyses and prototypes. Targets of this research include extraction of behavioural patterns of traffic encountered on the level of the isolated individual, subset of individuals, locality, specific time events and specific traffic hotspots. As described at the outset the main aim is not to trace the mobility of users but rather to predict estimated traffic congestion points and computation of duration for a travelling path given a starting point and a destination. Ideally if the users are highly predictable and stick to a regular travelling pattern they might be automatically notified when they are actually going to travel, how much its going to take them in terms of duration and suggestions are given to take different alternative routes which are less costly in terms of business.

A. This chapter is in the appendix

A.1 These are some details

```
this is some code;
```

Make sure to use this template.

References

[1] 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.

[2] 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*, 6(1):1–15, 2015.

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

[4] 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*, 58:240–250, 2015.

[5] M. Attard, P. Von Brockdorff, and F. Bezzina. The External Costs of Passenger and Commercial Vehicles Use in Malta. 2015.

[6] A. Bazghandi. Techniques, Advantages and Problems of Agent Based Modeling for Traffic Simulation. *International Journal of Computer Science Issues*, 9(1):115–119, 2012.

[7] C. Calabrese, F. Giusy, D. Lorenzo, L. Liu, C. Ratti, F. Calabrese, and G. D. Lorenzo. Estimating Origin-Destination flows using opportunistically collected mobile phone location data from one million users in Boston Metropolitan Area Terms of Use Estimating Origin-Destination flows using opportunistically collected mobile phone location da. *IEEE Pervasive Computing*, 10(4):36–44, 2011.

[8] 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*, 26:301–313, 2013.

[9] Cambridge Systematics Inc. and Battelle Memorial Institute. An Initial Assessment of Freight Bottlenecks on Highways. (October):191, 2005.

[10] S. Chakraborty NKNagwani Lopamudra Dey. Performance Comparison of Incremental K-means and Incremental DBSCAN Algorithms. *International Journal of Computer Applications*, 27(11):975–8887, 2011.

[11] 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*, 2526:126–135, 2015.

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

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

[14] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008.

[15] 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*, 6(June 2014):285–296, 2015.

[16] K. Hornik, M. Stinchcombe, and H. White. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural networks*, 3(5):551–560, 1990.

[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*, (October):45–50, 2016.

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

[19] 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*, 40:63–74, 2014.

[20] A. K. Jain and J. Mao. Artificial Neural Network: A Tutorial. *Communications*, 29:31–44, 1996.

[21] O. Järv, R. Ahas, E. Saluveer, B. Derudder, and F. Witlox. Mobile Phones in a Traffic Flow: A Geographical Perspective to Evening Rush Hour Traffic Analysis Using Call Detail Records. *PLoS ONE*, 7(11), 2012.

[22] 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*, 25:17–35, 2012.

[23] 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*, pages 1–8, 2012.

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

[25] J. Liu, F. Liu, and N. Ansari. Monitoring and analyzing big traffic data of a large-scale cellular network with Hadoop. *IEEE Network*, 28(4):32–39, 2014.

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

[27] 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*, 12(3):291–309, 2012.

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

[29] D. Schrank., B. Eisele., T. Lomax., and J. Bak. 2015 Urban Mobility Scorecard, 2015.

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

[31] M. Sommer, S. Tomforde, and J. Hähner. Using a Neural Network for Forecasting in an Organic Traffic Control Management System. In *Presented as part of the 2013 Workshop on Embedded Self-Organizing Systems*. USENIX, 2013.

[32] J. Steenbruggen, E. Tranos, and P. Nijkamp. Data from mobile phone operators: A tool for smarter cities? *Telecommunications Policy*, 39(3-4):335–346, 2015.

[33] 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*, 1876:10–19, 2004.

[34] 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*, 58:162–177, 2015.

[35] M.-h. Wang and S. D. Schrock. Feasibility of Using Cellular Telephone Data to Determine the Truckshed of Intermodal Facilities. *Cell*, (August 2009), 2012.

[36] 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*, 11(2):76–86, 2013.

[37] H. Wu, T. Zhang, and J. Gong. GeoComputation for Geospatial Big Data. *Transactions in GIS*, 18(S1):1–2, 2014.

[38] 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*, page 65, 2004.

[39] Y. Zheng and X. Xie. Learning travel recommendations from user-generated GPS traces. *ACM Transactions on Intelligent Systems and Technology*, 2(1):1–29, 2011.