Privacy Preserved Model Based Approaches for Generating Open Travel Behavioural Data

Godwin Badu-Marfo

A Thesis

In the Department

of

Geography, Planning & Environmental Studies

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Geography, Urban and Environmental Studies) at

Concordia University

Montréal, Québec, Canada

January 2021

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: **Mr. Godwin Badu-Marfo**

Entitled: **Privacy Preserved Model Based Approaches for Generating Open Travel Behavioural Data**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Geography, Urban and Environmental Studies)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. David Kwan

_____ External Examiner
Dr. Sebastien Gambs

_____ External to program
Dr. Yann-Gaël Guéhéneuc

_____ Examiner
Dr. Nizar Bouguila

_____ Examiner
Dr. Greg Butler

_____ Thesis Co-Supervisor
Dr. Bilal Farooq

_____ Thesis Co-supervisor
Dr. Zachary Patterson

Approved by _____
Dr. Norma Rantisi, Graduate Program Director

January 18 2021

_____
Dr. Pascale Sicotte, Dean
Faculty of Arts and Science

Abstract

Privacy Preserved Model Based Approaches for Generating Open Travel Behavioural Data

Godwin Badu-Marfo, Ph.D.

Concordia University, 2021

Location-aware technologies and smart phones are fast growing in usage and adoption as a medium of service request and delivery of daily activities. However, the increasing usage of these technologies has birthed new challenges that needs to be addressed. Privacy protection and the need for disaggregate mobility data for transportation modelling needs to be balanced for applications and academic research. This dissertation focuses on developing modern privacy mechanisms that seek to satisfy requirements on privacy and data utility for fine-grained travel behavioural modelling applications using large-scale mobility data. To accomplish this, we review the challenges and opportunities that are needed to be solved in order to harness the full potential of "Big Transportation Data". Also, we perform a quantitative evaluation on the degree of privacy that are provided by popular location anonymization techniques when undertaken on sensitive location data (i.e. homes, offices) of a travel survey. As a step to solve the trade-off between privacy and utility, we develop a differentially-private generative model for simultaneously synthesizing both socio-economic attributes and sequences of activity diary. Adversarial attack models are proposed and tested to evaluate the effectiveness of the proposed system against privacy attacks. The results show that datasets from the developed privacy enhancing system can be used for travel behavioural modelling with satisfactory results while ensuring an acceptable level of privacy.

.

Acknowledgments

I would like to thank all people who supported me during my work on this dissertation. Especially, I would like to express my gratitude and appreciation to my supervisors Bilal Farooq and Zachary Patterson for their utmost support and advice. I am grateful for their contributions on time, encouragement, ideas and funding to make my Ph.D. experience worthwhile and inspiring.

Again, I express thanks to the members of my Ph.D. committee, Prof Sebastien Gambs, Prof. Yann-Gaël Guéhéneuc, Prof. Nizar Bouguila, Prof. Gregory Butler and Prof Norma M. Rantisi for their beneficial suggestions and advice, as well as their time and consideration. Special thanks to my colleagues I worked with at the TRIP Lab, Concordia University, who contributed valuable suggestions and support: Ali Yazdizadeh, Kyle Fitzsimmons, Moshen Rezeai and Marshall Davey. I would also like to thank the staff and faculty at the Department of Geography, Planning and Environment at Concordia University for their support through the four years especially Jennifer Srey for the warmth of her smile. A big thank you to the great friends I made at Concordia: Jessica Hewitt and Chloe Boone for their utmost support and encouragement.

My heartfelt appreciation goes to my wife, Rebecca Commey, for her relentless support, inspiration and prayer. Thanks to my parents: Prof Kwame and Joana Agyei Frimpong, and family for their prayers and support. I would want to express gratitude to Valerie Gueye through whom I got to know about the PhD opportunity, and Patience Boni for their continuous advice and support. My final thanks goes to my team at DexAfrica: Daniel, Rammy, Kobby, Eugene and Gare for encouraging me to follow my dreams.

Contribution of Authors

Godwin Badu-Marfo carried out the data processing, data analysis, and development of the machine learning algorithms in all of the four articles in this thesis. He also wrote the manuscript of the thesis. Prof. Bilal Farooq and Prof. Zachary Patterson supervised Godwin Badu-Marfo during his PhD studies. They reviewed all of the methods and manuscript text. All authors discussed the results and contributed to the final manuscript.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Travel demand on transportation systems is dramatically increasing in urban cities due to the growth in populations and the adoption of public transport as a means to reduce emissions towards environmental sustainability. Contemporary economic activities are generally accompanied by a significant increase in the mobility and accessibility of a population, hence travel demand models are adopted by transportation planners to study the behavior of individuals in their choice of decisions regarding the provision and use of transport. Travel demand models are highly dependent on household survey data composed of household characteristics, personal information and trip details of travellers. Previously, data collection was administered through dedicated, self-reported surveys (e.g., household surveys, on-board surveys, etc.), and through technologies concentrated on vehicle flow counts like loop detectors. The emergence of smart mobile sensing devices equipped with GPS chips and used in combination with wireless technologies have dramatically increased the potential sources and volume of data that is collected in transportation system applications, referred to as "Big Transportation Data" [2]. The Pew Research Center [3] has reported that one-third of Americans live in a household with three or more smartphones while 84% of households contain at least one smartphone. Earlier research by the same institution in 2013 [4] estimated that 74% of the adult population aged beyond 18 years, use their phones to retrieve directions to points of interest (POI) based on their current location through location-aware technologies. Location-aware technologies (e.g. GPS, WiFi) are capable of determining their own location. These technologies are used by smartphone applications (e.g. Google Maps, OkCupid, Facebook, and many others)

to provide convenience to users by retrieving current user location and providing services close-by. These sets of application services are referred to as Location-Based Services (LBS). The works of [5, 6, 7] collected data on pedestrian counts using Wi-Fi logs, activity detection with location-aware social media data [8], travel time forecasting with Bluetooth [9], measurement of vehicle speed with CCTV [10] and dedicated smartphone applications [11].

While pervasive technologies provide high volumes of data, requirements on the capacity to store, manage and process all the data in a timely manner at a large operational scale needs to be met in order to extract relevant information for transportation decision making. We categorize these requirements into two broad challenges namely technical and non-technical. The technical challenges are focused on emerging computing frameworks and system architecture implementations that are suitable to handle the large quantity, diverse formats, and rapidly generated location data. As an example, CCTV and road-side cameras generate high definition video frames at faster rates. Generating large amounts of data at such high rates could come up against storage and processing constraints of traditional computing systems. The non-technical challenge is the concern for privacy, disclosure of information that could occur by integrating multiple data sources or when data is "published" for the public in the push to make data "open." This dissertation reviews large-scale operational challenges for implementing passive data collection from multiple sources and provides recommendations for overcoming the challenges to harness the full potential of big data in transportation decision making.

Typically, service providers of LBS applications (e.g. Google[12], Waze[13], Itinerum[14], Uber[15]) collect and store a user location over time (i.e. spatio-temporally). Transportation data on users are used to analyze mobility patterns and study the travel behaviour of a population in the private and public sectors. Though the potential of exploiting big transportation data is appealing for research, the release of personal information raises concern on privacy violations that cannot be overlooked. For instance, a service provider can be malicious by using data for unauthorized intent, while an adversary could approximate a user's precise home location to stalk them. An incident occurred in California in 2007 [16] where records from automatic toll booths on bridges were used in divorce proceedings to prove claims about suspicious movement of spouses. These incidents on privacy violations have made the public increasingly aware and concerned about the

risk of privacy on their personal information shared on smartphone applications and social media networks, especially in the exposé of the Facebook and Cambridge Analytica scandal in 2018 [17]. Similarly, governments and public institutions have adopted "Open Data" initiatives, a protocol allowing data to be published for free re-use, share and redistribution by the public without any restrictions of use [18]. This public release of information could raise concerns about privacy leaks when an adversary with enough background information can infer an individual participated in a survey.

Badu-Marfo et al. [19] suggested that personal privacy in terms of transportation data is as much a social challenge as it is a technical challenge, hence, the need for greater attention to technical, legal, and regulatory frameworks. Technically, transportation data contains personally identifiable information (PII) that requires protection through anonymization mechanisms. Similarly, the ability to maintain data privacy is threatened by de-anonymization (i.e., re-identification) by an adversary who has enough background knowledge of a user's mobility traces and is able to infer the identity of a user from published anonymized data. Thus arises the need to deploy effective protection mechanisms that are robust to adversarial attacks with less, or enough background knowledge. A common method to anonymize data, perturbation, adds a level of random noise to distort original values [20, 21]. Another approach to anonymization, generalization, changes data values from specific to more generalized information to achieve a level of indistinguishability for every data record. With these anonymization techniques, there is a trade-off between the guarantee of privacy the loss of utility [20]. In other words, when a high privacy level is achieved, the anonymized data is not useful for precise or accurate predictions.

To solve this privacy-utility trade-off, population synthesis is used to reconstruct new members of a population having similar features to the true population, but with no simulated members having characteristics of members of the real population. This technique accepts two inputs: aggregated census data, and sample microdata. The technique is capable of generating data surrogates with properties that conform to the underlying distribution of the population. Notable population synthesis approaches are re-weighting, matrix fitting and simulation-based methods. While these techniques are usable for providing a level of privacy with acceptable utility, they suffer the drawback of scalability due to the "curse of dimensionality" and computational complexity [22, 23].

A recent, novel approach, deep generative models address the concerns on scalability and bring with them the promise of computational effectiveness than the traditional population synthesis approaches. Notable generative models are Generative Adversarial Networks (GANs) [24] and Variational Autoencoders (VAE) [25]. GANs have shown promising results in generating realistic images which were hitherto intractable for large training sets and complex data types. The GANs framework work by training two model networks simultaneously namely: a generative model and discriminator model. The generative model is set against an adversary (i.e. the discriminator) that learns to distinguish fake samples produced by the generative model from real data samples. While GANs can reproduce realistic representations of a population, information leaks could occur when generated samples exhibit exact properties like real samples. The works of Abadi et al. [26] demonstrated model training in a differentially private manner using the Differentially Private Stochastic Gradient Descent (DP-SGD), such that the gradient of training data points are clipped and "noised" to limit its impact on the learning gradient. Using this approach, model training controls the confidence with which an adversary can infer further information about an individual.

In this dissertation, we explore privacy protection mechanisms for anonymizing sensitive attributes in detailed travel behaviour data. We demonstrate the performance and efficiency of the anonymized results for travel data analysis. We also experiment using the deep generative modelling frameworks (i.e. GAN and VAE) to synthesize complete travel behaviour data composed of a snapshot of socio-economic attributes and longitudinal location and activity sequences of a large population sample. The generative model is capable of simultaneously generating tabular and sequential location attributes in a differentially private manner as a means to protect sensitive information found in training data. The synthesized sets guarantee privacy protection for individual data points used in the training of the network such that an adversary cannot identify with a high confidence whether a member of the population participated in the training.

1.1 Contributions

In this dissertation, our key contributions are in the following aspects:

Big Transportation Data: We present a technical resource and positioning paper that highlights

the opportunities and challenges to harness the full potential of big data in transportation. As part of the discussion, we evaluate existing and emerging technology frameworks that could be adopted for an implementation of big data architectures for behavioural modeling.

Location privacy protection: We experimented different privacy algorithms to test on privacy protection for sensitive travel attributes for open data publication. We showed several techniques for measuring the utility and privacy guarantee provided by popular algorithms on travel activity diaries.

Population synthesis: We developed a novel deep learning framework to jointly synthesize sociodemographic (e.g. age, gender, income, etc.) and sequential travel behaviour (e.g. trajectory, activity schedule, etc.) data that ensures the representativity, privacy, and utility of the simulated data.

1.2 Dissertation outline

The major contributions of the dissertation are broken into four articles, each focusing on different aspects of travel data synthesis, big transportation data, location privacy, and generative modelling into behaviour analysis. An overview of the dissertation is shown in Fig 1.1.

Chapter 2 presents a methodological background on travel behavioural modelling, privacy protection algorithms, population synthesis, deep learning, and Generative Adversarial Networks. The chapter discusses relevant literature reviews and recent developments within the dissertation areas and highlights the existing drawbacks with existing approaches.

Chapter 3 introduces the concept of Big Transportation Data for behavioural data management and analysis. This chapter presents the system-level challenges, opportunities and future directions for deploying large-scaled transportation applications. The chapter establishes detailed resource information on diverse architectural designs, hardware components and scalable data infrastructure design.

Chapter 4 focuses on privacy protection for geographic points of interests in trip data using the most common population anonymization algorithms. Using a controllable random perturbation, this chapter experiments with the privacy protection of sensitive home locations while maintaining

the spatial fidelity of data to maximize the utility of anonymized data points. The chapter shows fundamental evaluation techniques for measuring location privacy and utility.

Chapter 5 introduces generative modelling with Generative Adversarial Networks for complete travel diaries. The chapter proposes a novel (CTGAN) approach for synthesizing travel population data that is capable of synthesizing synthetic agents having tabular (sociodemgraphic) and sequential mobility (location) data. A number of metrics are used to evaluate the performance and similarity between the true and synthesized populations.

Chapter 6 proposes an improved composite travel generative adversarial network with a shared layer that is capable of using a single model to synthesize multiple outputs in diverse formats and which is trained in a differentially private manner. The chapter demonstrates the effect of different privacy parameters and sampling on synthetic generation, which in effect decays the utility of the synthesized results. Also, the chapter evaluates various metrics for privacy guarantee and similarity of distributions.

Chapter 7 summarizes the findings from the dissertation articles in the afore-mentioned chapters and discusses the limitations of the developed methodologies. The chapter recommends new research interests for future work and possible improvements to build privacy-by-design generative modelling frameworks for travel behaviour.

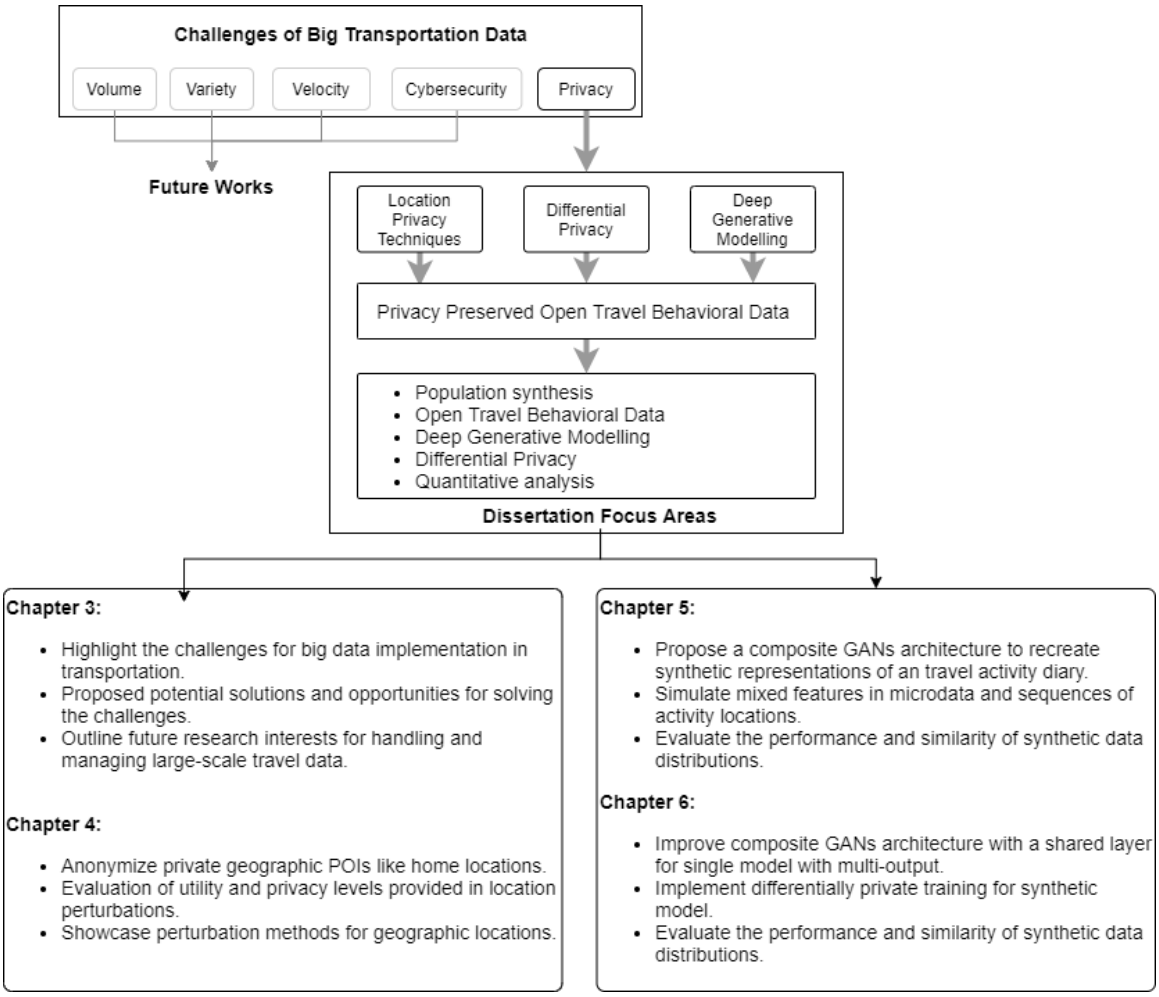


Figure 1.1: Dissertation Overview

Chapter 2

Background

This chapter reviews the literature and relevant contributions on the focus areas for this dissertation including big transportation data, travel behaviour modelling, population synthesis, privacy protection and deep generative architectures. In the next sections, we discuss the details for each focus topic.

2.1 Big transportation data

In a fast changing and complex urban environment, urban planners rely on data collected on mobility and traffic to plan transport systems. This data is key to solving urban mobility problems (e.g., road congestion) and also allowing better travel patterns, origins and destinations, and observe, operations and drawbacks in real time. Recently, the evolution of Internet of Things (IOT) sensors and smartphones have dramatically increased the size and diversity of formats of data collected for travel purposes referred to as "big transportation data." This term characterizes the high-volume of data collected, the large variety of data formats and the velocity at which data is generated and processed for transportation decision making [27].

High volume: defines the growing size of transportation data that continues to increase exponentially from zettabytes to yottabytes daily. With a recent estimate from McKinsey & Company[28], cars connected to the internet create up to 25 gigabytes (GB) of data per hour, which equates to dozens of movies stored in High Definition(HD) every 60 minutes.

High velocity: defines the increasing rate of data that is generated from transportation systems and devices. Emerging technologies like sensors, traffic cameras and autonomous vehicles produce data at high speeds and this data must be processed for further action and decision making. According to Forbes[29], there are 95 million photos and videos shared on instagram daily. Also, Google processes 40,000 searches every second.

High variety: defines the heterogeneity of transportation data exhibited in its numerous formats including structured and unstructured data. Traffic cameras generate high-definition videos and images of traffic flow. Vehicle onboard sensor devices log on events that mostly exist in unstructured data exchange formats like extensible markup language (XML) for third party devices to feed and act on the data provided. GPS units and smartphones with guidance software generate enormous volumes of locational data. All of these data sources are stored in different formats.

While these three characteristics underlie the primary composition of big transportation data, there are other expanding characteristics that are also well discussed in literature of Kitchin [30] and Gandomi [31]. Though these three defining characteristics are not only characteristics of transportation data, the distinguishing characteristic which is of much relevance to the field of transportation is that of high variety, transportation systems mostly generate location-aware and location-ignorant datasets in both structured and unstructured formats. Location ignorant datasets are gathered from devices (e.g. Bluetooth, WiFi, GSM) that are not able to ascertain their own location but can collect information and sense the presence of other devices. For instance, in the work of Poucin et al. [6], data from WiFi networks has been used to study pedestrian travel behavior based on connection histories to wireless routers. Bluetooth receivers have also been used to assess automobile route choice and travel times on alternate routes [32].

Location-aware datasets, on the contrary, are generated from devices and technologies (e.g. smartphones, navigation GPS) that are capable of determine their own location. They derive coarse and fine grained precision on location coordinates. Transportation planning, operations and research depend heavily on these devices for precise spatio-temporal data in their decision-making and analysis. The remarkable growth in usage of smartphones in recent years has made it a great technology for collecting location information from users who seek services through location-aware applications. For example, a user can search for the nearest restaurants in her proximity by providing her

current location to location-based services (LBS) like Google Maps. Through LBS applications, it is now easier to request a rideshare pickup by a user sharing his location information with car-sharing service providers like Uber [15]. A lot of academic research work has been done in the use of location-aware data from GPS and smartphones for transportation purposes. In a study of the diversity of mobility patterns for children [33], GPS-tracking devices and mobile phones were used to conduct a survey on child mobility and the geographic inter-dependency of child mobility. In other work, Patterson et. al. [34] used a location-aware smartphone travel survey application to collect mobility data on participants from Concordia University, Montreal in 2016, and the experiment reduced the burden of responding to origin-destination surveys as compared to traditional self-reported surveys.

In the next subsection, we discuss the application of open travel behavioural data for travel demand modelling with relevant literature.

2.2 Travel behaviour modelling

Transportation planners and decision-makers have the responsibility of making informed choices on the performance of transportation systems in an environment of rapid change in economic, demographic, and land use of a population. Generally, travel demand models are used as a decision making framework to provide quantitative information about travel demand and supply alternatives that are optimal in providing an efficient transportation system for different input assumptions. Travel demand refers to: the activities that people travel to (trip purpose), where the activities are located (location), when the activities take place (time), by what mode of transport and route people make these trips (mode and itinerary). Given these dimensions, travel models aim to predict the behaviour, patterns, and preferences of commuters, and provide objective assessments of the benefits and costs of transportation dependent variables such as policies, land use, capital investments, socio-demographic assumptions and many others. The most popular travel modeling approaches are trip-based, activity-based, and tour-based models [35, 36].

Trip-based models (i.e., the Four Step Model) basically estimate the trips but trips are abstracted from the individual that make them. These models are structured into four steps: trip generation,

distribution, mode choice, and assignment (route choice). Trip generation estimates the number of trips leaving and destined to each geographic or Traffic Analysis Zones (TAZs). Trip distribution links trip origins and destinations from and to TAZ in the zone system. Mode choice determines the mode of travel for trips between TAZs. Finally, assignment predicts route itineraries for each trips between zones [37].

Unlike trip-based models, activity-based models (ABMs) estimate each person's activity pattern and travel choices across the entire day while reflecting the types of activities, the individual desires to participate in and also sets the priorities for scheduling. Also, ABMs are generally based on behavioural theories postulating how people decide on the activities based on the time, location and mode of travelling and sociodemographic characteristics. Comparatively, activity-based models focus on disaggregate person-level information rather than an aggregate zone-level information of the trip-based models.

Most transportation research works have focused on estimating traveler origin, transfer and destination. For example, Zhang et al. [38] analyzed the distribution and transfer of passenger flow of platform metro passenger at operational stages. Similarly, Shouhua et al. [39] proposed classification models to classify urban rail, to demonstrate the characteristics of entering station, exiting station, and transferring and waiting of metro passengers. Horowitz [40] presented a trip frequency, destination and mode choice model that incorporates inter-trip dependence and is implemented in a trip-based model. classification for urban rail transit passages based on passenger perceptions

Activity-based models incorporate significantly more detailed information and thereby produce more detailed outputs than trip-based models. Also, ABMs present a more theoretically consistent representation of trips made jointly by household members and also include explicit details of time-of-day choices, such as the amount of time spent when participating in activities, arrival and departure times [37].

Trip-based and activity-based models are generally dependent on household travel surveys. Typically, travel surveys collect detailed information about individuals with attributes containing personal sensitive information (e.g. identity). Disclosing the identity of households or individuals could pose a potential threat hence there is a need for privacy protection on datasets collected for

travel demand models. In this study, we develop privacy-by-design approaches to generate privacy-guaranteed travel behaviour data that maintains utility for travel modelling. In the next section, we discuss approaches for implementing privacy protection.

2.3 Privacy protection

The transit ridership report of the American Public Transportation Association (APTA) for the first quarter of 2018 [41] recorded an approximate ridership of 96 million trips through the public transportation rail system in Montreal, Canada. Smart card readers are deployed in metro stations and buses that charge transit fares when passengers board using their smart cards. The metro system operator maintains a database of all transactions, associated with the personal identities of passengers especially when credit cards are used to recharge smart cards. While these systems are proprietary and highly secured, in a hypothetical scenario where such personal identifiable transit information is gained by an adversary or inappropriate access is achieved by an employee, the credit card details could be linked to the income or financial information of an individual.

Additionally, it is known that public and private research organizations (i.e., travel authorities) usually collect large amounts of identifiable information during data collection. Often, much of this data is not relevant to the purpose of study but is considered valuable to augment the possible dimensions of research. Following the recent public outcry on privacy violations (e.g. the case of Facebook and Cambridge Analytica [17]), data collectors are required to limit the amount of personal identifiable data collected or implement mechanisms to privatize personal identifiable information. Transit authorities could suffer losing money through legal suits and reputation if they violate the privacy of their passengers. In order to implement privacy protection on travel data, anonymization models are used. The most popular anonymization models are; k-Anonymity [42] and Differential privacy [43].

2.3.1 *k*-Anonymity

k-Anonymity is a widely used privacy model, first proposed by Samarati and Sweeney [42, 44]. The notion assumes that for any record in the table with a quasi-identifier (QID) attribute value, there exist at least *k*-1 other records with the same QID. This means that the minimum group size is at least *k*. As an objective, *k*-Anonymity, ensures that each record is made indistinguishable from at least *k*-1 other records in the table. This removes uniqueness of each individual through generalization and suppression, and prevents record linkages through quasi-identifiers (QI). An example is in the case of the Governor of Massachusetts whose health records were identified by linking identifier fields (i.e. age, sex, zipcode) from the voter registry to medical data[42]. Its implementation removes explicit fields and modifies QI but maintains sensitive attributes. However, it has been shown that anonymizing a table for *k*-Anonymity does not sufficiently provide privacy protection for individuals, successive works have built on the notion of *k*-Anonymity to address some of the weaknesses unearthed in the original model. The emerging flavours of *k*-Anonymity which guarantee a better privacy of individuals are discussed below.

The work of Machanavajjhala et al. [45] show that *k*-Anonymity cannot prevent an adversary from uniquely identifying an individual without considering its sensitive attributes. This means that while k-anonymity can protect identity disclosure by modifying the QI values, an adversary with background knowledge about a sensitive attribute of an individual, such as cash tips on taxi fares, will be able to uniquely identify an individual. The authors introduced the model of *l*-diversity, which considers both the QI attributes and sensitive attributes. In this improved model, an equivalence class (i.e. a collection of records in a table that have same attribute value with respect to a quasi-identifying attribute) is said to satisfy *l*-diversity if the probability that any record in the group is associated with a sensitive value is at most 1/*l*.

As follow-up work on *l*-diversity, Li et al. [46] observed that in scenarios where the overall distribution of a sensitive attribute is skewed, the *l*-diversity model cannot prevent linkage attack on the attribute. For example, when travel data has 90% of records having a trip mode of train and 10% with the mode of bus, after generalization and suppression on the QI attributes, it could be inferred that a passenger travels on train with a high confidence. To overcome the problem of skewness in

sensitive attributes, the authors introduced *t*-closeness, as an improvement on the *l*-diversity model. In this model, an equivalence class is said to satisfy *t*-closeness if the distribution of sensitive values within the group is nearly equal to the distribution of the sensitive attribute in the overall table. Though *t*-closeness proves robust in normalizing the distribution of sensitive attributes, it has its own limitations and weaknesses. As a weakness, it lacks the ease of defining varying protection levels for different sensitive values. Also, it degrades data utility when it normalizes the distribution of the sensitive values within the equivalence classes.

Most of the models previously discussed provide privacy protection with respect to categorical or discrete-type sensitive attributes. To apply the same to numerical sensitive attributes such as tips paid, the work of Zhang et. al [47] provides a privacy model named (*k*, *e*)-anonymity. The objective of this model is to partition individual records into groups such that each group contains a minimum *k* different sensitive values with a range of at least *e*.

(*k*, *e*)-Anonymity does not consider the proximity of sensitive values in any equivalence class, which can violate individual privacy. For example, in travel data with many sensitive values (e.g., travel budget, tips paid) are close to each other like a tip of $50 is close to a tip of $51. When these sensitive values (e.g., tips) are grouped into equivalence classes, an adversary has high confidence to infer that an individual in an equivalence class is associated to a close sensitive value.

Lastly, another privacy model that considers both QI and sensitive attributes is (α, *k*)-anonymity. This model is developed by Wong et al. [48], and suggests that an equivalent class is said to satisfy (α, *k*)-anonymity if the number of records in the equivalent class is at least *k* and the frequency of each sensitive value in equivalent class is at most α.

As discussed in section 2.1, academic researchers and transport planners collect mobility data to understand travel characteristics, patterns, and choices of commuters. Travel demand models basically consider trip sequences with socio-economic information (e.g. household size, income, automobile ownership) generated for a sample population to reflect the travel behaviour of the general population [49].

2.3.2 Differential privacy

While the k-Anonymity technique protects a record by creating at least k-1 records with similar attributes, Dwork et al. [43] created differential privacy as an anonymization technique that promises protection for a participant whose addition or removal from a survey does not affect the computed outcome of the survey. The outcome of statistical computation on the data will not have a significant impact by the removal or addition of a single record. Imagine a survey on the routes used daily from home to work collected on 100 workers. As part of survey, data on workers who made a stop at a bar were also collected. The outcome of the survey gave 60 out of the 100 workers who use a route without making a stop at a bar while the remaining 40 made stops at a bar. In studying the outcome of the survey by a researcher, when the record of a participant, Joseph is excluded from the survey, the statistics gave a count of 39 for workers who visits a bar. The researcher can confidently deduce that Joseph visits a bar on his route to work.

The primary mechanism of differential privacy is to address this weakness of aggregation by adding random noise to the computed outcome such that the addition or removal of a participant like Joseph does not influence the statistical outcome. This implies that the probability for a query to return a value v when applied to a database with the route record of Joseph, will be similar to the database without his route record. The differing values in both databases should be within a bound of a privacy budget or epsilon (ϵ).

Critical to the implementation of differential privacy is the addition of random noise. In relation to the route example, if the reported outcome for workers who make a stop at a bar is given as 39 or 41 then the removal of a participant will not change the outcome. An inference by an adversary will require strong background knowledge of the context. The amount of random noise added to the statistical outcome must be robust to differentiated attack, defining the knowledge an adversary might have to make deductions. The amount of random noise is selected from a Laplace or Gaussian distributions to produce a result that masks the details of a given record. Differential privacy is effective in cases where an aggregation of several records are to be published. This raises a weakness and poor suitability for applications where a single record is involved. The addition of noise reduces the precision of statistical outcomes thus in applications where noise is unacceptable such

as safety and life-threatening observations, this technique cannot be used [50]. Lastly, the technique makes use of a global sensitivity function (e.g. sum, average, count) to summarize statistical outputs. Though these query functions are appealing, in cases where functions like Max and Min are used, the results can be misleading. As earlier mentioned, privacy protection is becoming a primary requirement in analysing travel data. This requirement is driving research work towards the design of efficient anonymization techniques that are adaptive and suitable for numerous scenarios of data utility with a promise of privacy protection. Hitherto, transportation planners have relied on traditional population synthesis approaches that allows for recreating members of population by integrating aggregate data from one source with disaggregate data from another source such as census data [51]. This approach prevents information disclosure risk. In the next section, we introduce the concept of population synthesis and existing methodologies of population synthesis.

2.4 Population synthesis

Planners study the interactions between the social behaviour of population agents (i.e., individuals and households) and the environment and urban systems like transportation. Evolving travel alternatives such as car-sharing, park-and-ride cannot be simulated with traditional travel models rely on aggregate population data, that is not capable of reproducing the sophisticated spatial behaviour of agents and respond timely to growing urban travel demands. These drawbacks have recently increased research interest in activity-based modelling using disaggregate agent-based information. In activity-based models, researchers use detailed information on agents to understand human behavioural patterns, preferences, habits and perceptions usually at a neighbourhood-scale resolution to understand transportation patterns, referred to as microsimulation.

Microsimulation models are built with micro data (i.e., individual or household level information) that generally are difficult to acquire due to the high cost of data collection, legal restrictions in most countries, and diverse data variety existing in multiple sources. These models have been used in studies of transport behaviour, demographic and household dynamics [52, 53]. Miller et al. [54] used the technique to develop an Integrated Land Use, Transport and Environment model at Canadian universities under the leadership of the University of Toronto. Given its flexibility to model

simulations and processes that cannot be modelled with aggregate data, microsimulation models have been used for urban land use and transportation in North America including the California Urban Futures (CUF) Model at the University of California at Berkeley [55, 56]. While micro-data are difficult to gather, synthetic micro data are used as replacement for developing microsimulation models.

Population Synthesis is an approach to generating synthetic representations of a population that have similar statistical distributions to the original population. It was first developed by Beckman et al. [51]. This procedure is the most widely used in integrated models that generate synthetic populations. A variant of synthetic population simulation, known as Iterative Proportional Fitting (IPF) was first introduced in the transportation literature by Duguay et al. [57] to synthesize data based on household surveys. The IPF method transforms one-dimensional distributions that are generally available as statistical input to multidimensional distributions required to generate a synthetic population. As an example, given a statistical one-dimensional input data of populations by age, or households by land-size, IPF could be used to generate multidimensional distributions of this input such as persons by age, their gender, education, nationality and others. IPF transforms the data elements of a matrix such that the aggregate sums of rows and columns are equal to its one-dimensional input data with minimal deviation from the initial values of the matrix. IPF is also applicable for two or multidimensional matrices.

While iterative proportional fitting is appealing for the generation of sets of microdata, it has limitations in dealing with complex sets of micro data thus requires very high standards of reliability of input data for the cells of data matrix. Typically, micro data like PUMS are not readily available as input and there is a limitation on extracting features that are not contained in the initial micro data. Another variant of population synthesis, referred to as Monte-Carlo sampling addresses these drawbacks. Monte-Carlo allows the generation of an infinite set of features (multidimensional) from a one-dimensional distribution of administrative registers [58]. Moeckel et al. [59] suggest that with Monte-Carlo sampling, many features can be selected to run a microsimulation model and this can only be limited by determining correlations between the selected features.

Both approaches use an aggregated population usually from the census or administrative registers as input to extract detailed information at the agent level. These techniques do not learn from raw

disaggregate data to generate the synthetic populations, a level of utility is lost by features that are not covered in the published micro data. Microsimulation models are bound by the features available in the underlying input data. However, population synthesis suffers drawbacks of scalability to high dimensional features that under-perform because of computational challenges. Deep generative models have advanced in recent times and have shown successes in reproducing new members of population by estimating the underlying joint distributions, while scalable to high dimension data with computational efficiency.

2.5 Deep generative modelling

Generative models are unsupervised learning tasks in machine learning, capable of learning the full probability distributions (i.e. patterns and regularities) of disaggregate input data to predict new samples that fit the learnt distribution of the original dataset. Typically, generative modelling includes discriminative and generative components. The objective of the discriminative component is to learn a function (f) to predict the label (y) of a given feature of an input data-point (x). In this sense, the discriminator performs a task of supervised classification for input variables. The generative component learns the joint probability distribution and generates new samples from the distribution. Traditional machine learning models that are capable of generating new samples are Naive Bayes [60], Latent Dirichlet Allocation [61] and the Gaussian Mixture Model [62].
Similarly, generated modelling implemented with deep neural networks is referred to as "deep generative modelling". Examples of deep generative models include Restricted Boltzmann Machines (RBM) [63], Deep Belief Network (DBN) [64], Variational AutoEncoders (VAE) [65] and Generative Adversarial Networks (GAN) [24]. The VAE and GAN are modern approaches which have shown success in reproducing realistic samples of images, sound and video.

2.5.1 Variational Auto-Encoders

Kingma et al. [65] proposed the use of Variational Auto-Encoders (VAE) consisting of two neural networks; an encoder and a decoder. The encoder compresses data into a latent space (z) while the decoder reconstructs data from the latent representation. The input of the encoder is a set of

features of a data point (an observation), its output is a hidden representation z, and has weights and biases. Mathematically, the encoder approximates $q(z|x)$ where input z conditioned on the data x. New samples are drawn based on the posterior distribution $p(x|z)$ of the latent representation, the output from the encoder. The encoder and decoder networks are trained to maximize a lower bound of the log-likelihood of the data.

2.5.2 Generative Adversarial Networks

The Generative Adversarial Network (GAN) is a deep generative model first introduced by Goodfellow et al. [24]. It's basic utility is to create samples after learning the distribution of underlying data. GANs consist of two models that compete against each other in a zero-sum game framework namely a discriminator, D and a generator G as shown in Figure 6.1. The task of the discriminator is to discriminate between real and fake input data while the generator is to generate fake data after learning the distribution of the real data. The interaction between the generator and the discriminator is illustrated by the generator acting as a counterfeiter trying to make fake money while the discriminator acts like a police officer checking the legitimacy of money and detecting counterfeit money. The zero-sum game between the two models helps to develop skills and improve their performance of both the generator and discriminator. Recently, Ouyang et al.[66] proposed a GAN-based approach to generate trajectories.

GANs is primarily used as a generative model to generate data samples through labeling the data into fake and real [67]. The work of Mirza and Osindero [68] introduced conditional GANs, which conditions GANS on class labels. This improvement allows for samples to be generated based on classes rather than the vanilla categories of fake and real labels. In a similar improvement, Springenberg [69] developed a categorical GANs (CatGANs) in which the discriminator separates data into K categories while assigning a label y to each example x. While the model output of a neural network can be transformed into a multinomial distribution by applying the softmax layer, sampling from the distribution is not a differentiable operation hence the back propagation is blocked during the training of generative models for discrete samples. To solve this drawback, the Gumbel-Softmax [70] and the Concrete-Distribution [71] were proposed in the domain of the VAE. Similarly, Kusner et al. [72] adapted the approach and implemented a GAN for generation of sequences of discrete

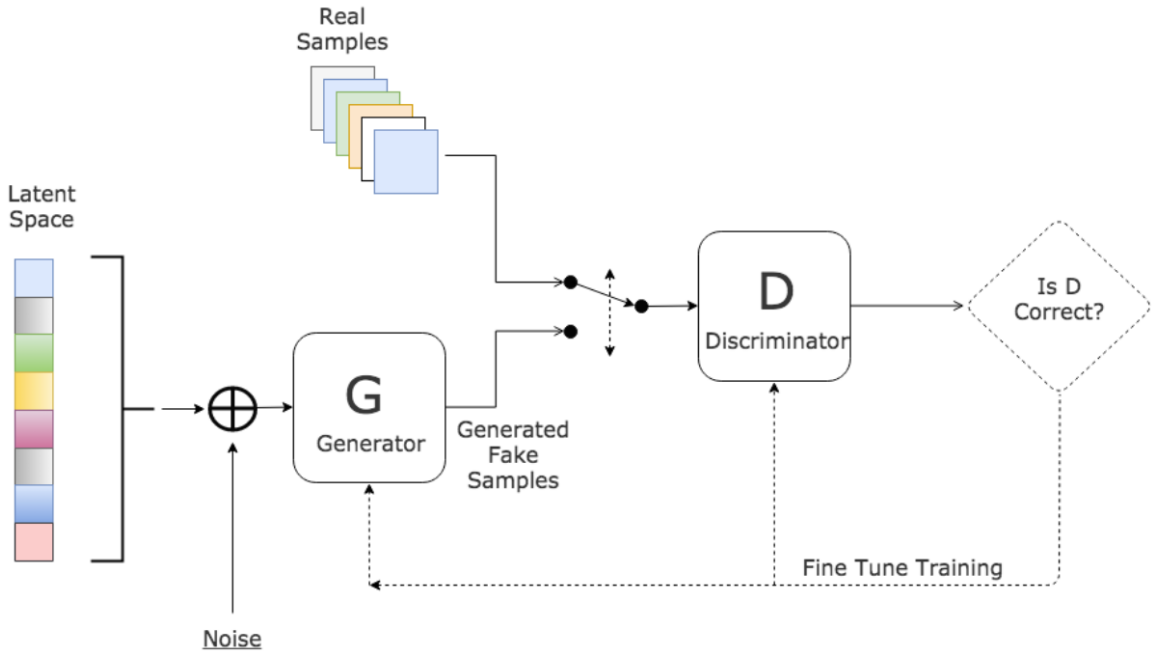


Figure 2.1: Generative Adversarial Networks.

elements. Sequence generation is essential in travel diary synthesis to simulate longitudinal representation of travel activities.

Given these research backgrounds, this dissertation adopts and extends the enlisted approaches and discussions to develop novel methodologies to provide privacy protection for travel data publication. In the next section, we present the challenges and opportunities of managing big data in transportation.

Part II

Dissertation Articles

Chapter 3

A Perspective on the Challenges and Opportunities for Privacy-Aware Big Transportation Data

3.1 Preamble

This chapter presents a review on the concepts related to and the characteristics of Big Data and its application to Transportation planning and decision making. We discuss the opportunities and challenges for harnessing the potential of Big Transportation Data.

This research article was published in the Springer Journal of Big Data Analytics in Transportation:

Badu-Marfo, G., Farooq, B. Patterson, Z. A Perspective on the Challenges and Opportunities for Privacy-Aware Big Transportation Data. J. Big Data Anal. Transp. 1, 1–23 (2019).
https://doi.org/10.1007/s42421-019-00001-z

It was also presented at the Transportation Research Board (TRB) 97th Annual Meeting in January 2018, at the Walter E. Washington Convention Center, in Washington, D.C.

3.2 Abstract

In recent years, and especially since the development of the smartphone, enormous amounts of data relevant for transportation have become available. These data hold out the potential to redefine how transportation system (i.e., design, planning and operations) is done. While researchers in both academia and industry are making advances in using this data to transportation system ends (e.g., information inference from collected data), little attention has been paid to four larger scale challenges that will need to be overcome if the potential for Big Transportation Data is to be harnessed for transportation decision-making purposes. This paper aims to provide awareness of these large-scale challenges and provides insight into how we believe these challenges are likely to be met.

3.3 Introduction

Transportation system (i.e. design, planning and operations) has been a quantitative discipline highly dependent upon data at least since the birth of modern travel demand modeling in the 1950s. Until recently, data collection has been done through dedicated, often self-reported surveys (e.g. household surveys, on-board surveys, etc.), and through various methodologies and technologies concentrated on vehicle flow counts (e.g. loop detectors). Recently, a combination of devices and technologies have dramatically increased the number of potential sources, as well as the amount of data that can be collected with urban transportation system applications, what we refer to as Big Transportation Data. Examples of this data include Bluetooth and CCTV traffic counts [9, 10], pedestrian counts with WiFi [5, 7, 73], activity detection with social media location data [8], dedicated travel survey smartphone applications [11] and smartphone data aggregators [74].

The potential for this data in transportation systems have not been overlooked, with many researchers in academia and the public and private [75, 76, 77, 78] sectors investigating ways in which to use it in their processes. Until now, the academic literature has been primarily preoccupied with two aspects of big data in transportation. First there has been research on how to go about collecting relevant data with these new technologies (e.g., [79],[80], [11]). Second, there has been research focusing on methods (statistical, machine learning, etc.) using collected data and inferring transportation relevant information from it (e.g. mode, trip purpose, etc.) [8, 81, 82].

While the successful collection of data, and inference of information relevant to transportation system presents many challenges to the routine incorporation of Big Transportation Data in design, planning and operations, little attention has been paid to the impending challenge of actually being able to store, manage and process all the data on large and operational scale, not to mention the challenge of protecting privacy of the people providing the data. We divide these large-scale implementation challenges into four dimensions. The tautological fact that there is a large quantity of Big Data presents challenges in storing it. Second, the need to compute algorithms on large scale data presents a challenge in processing. Third, Big Data comes in many different formats, making the ability to take advantage of data collected from different sources challenging. Fourth is the challenge of protecting personal privacy.

While the quantity of data and diversity of formats are primarily technical challenges, personal privacy is a political as well as technical challenge. The political nature of the challenge was recently evidenced by the controversy around Facebook and Cambridge Analytica [17] and public reaction to it. The issue of privacy and Big Data is multifaceted. Most obviously, much Big Data is sufficiently detailed (e.g. geographically and temporally precise GPS data) that it could reasonably be used to identify individuals. A less obvious challenge to privacy is the ability to combine information about individuals across data sources thereby making the identification of individuals possible with individual "quasi-identifying" information. Another less obviously personal challenge relates to who can access private data, and how to control access in the most secure way.

All of these challenges will need to be met before the potential for Big Data in transportation can be harnessed. As such, this paper aims to provide an in-depth awareness of the large-scale implementation challenges currently facing the use of Big Transportation Data in design, planning and operations of transportation. It also provides insights into how we believe these challenges are likely to be met.

The paper continues with a section describing the scope of this paper and moves on to define Big Data, Big Transportation Data and from where they come. The next section describes the current state of the transportation literature as it relates to Big Data. This is followed by a background section on system architecture needed to understand the sections on the four main challenges to the widespread use of Big Transportation Data in transportation planning. A concluding section sketches our understanding of how the challenges of Big Transportation Data are likely to be overcome in the future.

3.4 Scope of this Work

The four large-scale challenges to the widespread use of Big Transportation Data identified in this paper have resulted from a thorough literature review. Since there is very little attention to this question in the transportation literature, most of the literature reviewed has come from computer science, computer engineering, and fields the most advanced in the use of Big Data, such as health and agriculture. The primary Google Scholar search terms used were"big data implementation

challenges" and "big data technologies." Relevant papers from articles resulting from these searches were then included in the literature, and this process was done iteratively. The more than one hundred and fifty papers resulting from this process were placed into four categories of challenges: storage, processing, integration, and data privacy. These challenges concentrate on those relating directly and uniquely to Big Data. While other challenges such as data security, integrity and transfer are relevant to Big Data, they are not unique to Big Data, and so we don't concentrate on them here. Interested readers can consult the vast literature on these topics elsewhere [83, 84, 85]. We continue by defining both Big Data and Big Transportation Data, as well as from where they come.

3.5 Key Characteristics of Big (Transportation) Data

Big Data has been described, characterized and defined in both academic and non-academic (traditional media, trade press, etc.) sources. Across these sources, there is a great variety in how Big Data has been defined and characterized [86, 87, 88, 89]. Often, Big Data are characterized by words beginning with the letter "v." One problem with such"v-words" is that there is often variation in how they are defined from one author to another. Also,"v-words" do not necessarily define characteristics of only Big Data but of "non-Big-Data" as well. Finally, there are some concepts critical to understanding the challenges for the widespread use of Big Data that are not easily described with"v-words." Given the confusion around definitions and the fact that we are most interested in the characteristics of Big Data as they relate to the challenges of using it, we discuss two types of characteristics, not all of which are "v-words." As such, below we discuss "defining" and "non-defining" characteristics of Big Data.

3.5.1 Defining Characteristics of Big Data

Defining characteristics of Big Data are those that are unique to Big Data as opposed to data in general. Those critical to understanding the challenges of widespread use of Big Data are those from the most-cited definition of Big Data by the Information Technology (IT) advisory firm Gartner. According to Gartner:

"Big data is high-*volume*, high-*velocity* and/or high-*variety* information..."■[27].

Volume refers to the size of individual datasets. Already in 2011, there were 2.5 quintillion bytes of data created every day [90], and this number keeps increasing exponentially [91, 92] so that"Big" datasets currently typically range from zettabytes (10^{21} bytes) to yottabytes (10^{24} bytes) [93]. It is often said that "Big" datasets are too large to be handled by an individual computer [94].

Velocity refers to the rate at which data are being generated. As with volume, the figures on rates of data being produced and received can be staggering. It was reported in March 2018 that over 900 million photos were uploaded to Facebook [95]. In addition to being related to the rate with which data are generated, velocity also encompasses a notion sometimes referred to in the literature as *variability* [31]. Whereas velocity refers to the rate at which data are generated, variability refers to variance over time in data flow rates.

Variety refers to the structural heterogeneity of data. That is, data provided in different formats, some structured and others not. Structured data are mostly in the form of tabular schema-imitating spreadsheet and relational database systems. Text, audio, images and videos are examples of unstructured data, with Extensible Markup Language (XML), being an example of semi-structured format [31]. Unstructured data is more difficult to process, store and integrate and is becoming more common [96, 97, 98].

3.5.2 Non-defining Characteristic of Big Data

A non-defining characteristic of Big Data is simply one that applies to other types of data as well. Such characteristic creates an important challenge for the widespread use of Big Data, as Big (and non-big) Data is either by nature personal, or can be personal. By personal we mean that an individual's identity is explicit, or can be revealed. That data *can* be personal we mean that different data sources can be combined to identify an individual and other information about them. While this is not a new problem (e.g., it has been a concern for a long time with traditional census data [99]), it becomes compounded with Big Data. This is so because of the many potential different sources of data available on people [98] and also because of the very personal nature of some Big Data (e.g. precise location data, medical records, etc.) [100].

Recently, large data collection organizations (i.e. government, institutions and non-governmental organizations) have begun adopting "open data"" initiatives that allow for data to be freely available,

shared, redistributed and reused by the public without restrictions of use [18]. As such, open data can serve as a resource for private, public and academic research. The availability of such data means privacy has become of even greater concern.

Big Transportation Data

We characterize Big Transportation Data (BTD) simply as Big Data (as characterized above), but with potential transportation system applications. That is, data that could be used in areas in the traditional purview of transportation design, planning and operations, such as travel demand forecasting, infrastructure planning, transit network planning, operation optimization,etc.

3.6 Where Does Big Transportation Data Come from?

BTD comes from the combination of three types of technologies. We begin with two broad categories of devices that collect BTD; location-ignorant and location-aware devices. Location-ignorant devices are able to sense the presence of other devices, although they are not explicitly aware of their own locations. These include technologies such as Bluetooth[101], Wireless Fidelity (WiFi)[101], Global System for Mobile (GSM)[101] and Closed-circuit Television (CCTV)[102].

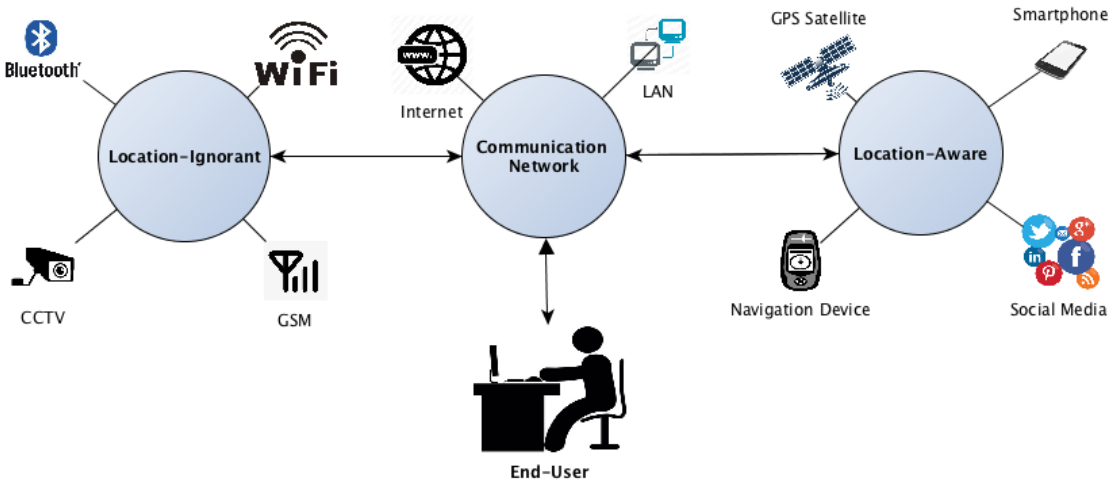


Figure 3.1: Ecosystem of Big Transportation Data

The second are devices that can determine their own whereabouts, i.e. they are location-aware.

These devices typically derive their locations based on the location of other devices such as WiFi routers, GSM towers, or satellites part of various Navigation Satellite Systems, such as the Global Positioning System (GPS). They include GPS units, GPS navigators and most importantly smartphones.

While devices that collect data are critical for being able to use BTD, its potential can only be harnessed if the devices are connected to a communications network, such as the Internet[103], private Local-Area Networks (LAN)[103] or Wide-Area Networks (WAN)[103]. These networks allow the transfer of data from collecting devices to database storage systems from where they will be accessed for processing and analysis by end-users. Figure 3.1 provides a schema of the BTD Ecosystem.

3.7 The Current State of BTD in Transportation

The combination of location-ignorant, location-aware and communications networks has led to the birth of Big Transportation Data. Academia as well as the public and private sectors have not overlooked the potential for BTD in transportation.

3.7.1 Research with Data Collected with Location-ignorant Devices

In recent years, academic research has been conducted with the use of data from location-ignorant devices in public transit planning and operations. Transit smartcard data has been at the forefront of this to understand travel behavior [104, 105] and transit user loyalty [106], also state that Smart cards can be used to ascertain the loyalty of transit users in a network.

WiFi network data has also been used to understand (primarily pedestrian) travel behavior based on connection histories to wireless routers [6, 107]. Similarly, Bluetooth receivers have been used to assess automobile route choice and travel times on alternate routes [32].

3.7.2 Research with Data Collected with Location-aware Devices

Location-aware technologies have been developed to determine their own location. Location sensors derive precise locations through the use of GSM, WiFi and GPS [108, 109]. Transport operations,

planning and research heavily rely on these devices for precise spatio-temporal data in analysis and decision making. Location-aware technologies is discussed in two categories namely GPS and Smartphones.

GPS

Navigation GPS devices have long been used for finding the location of Point Of Interest (POI). Transportation fleet operations rely heavily on navigation GPS systems that provide mobility trajectories of fleets. Much academic research has been done to cover the application of navigation GPS devices in transportation. Davies et al.[110] evaluated the use of GPS devices for providing location-aware visual and auditory prompts for people with intellectual disabilities to enable them in navigating busroutes. Handheld GPS devices have been extensively used for travel mobility surveys in research [111, 112, 113]. A study on children's mobility using GPS-tracking device and mobile phone survey was conducted in Copenhagen [33]. The research shown diversity of mobility patterns for children and the geographic interdependency of child mobility. Surveying and data collection with Navigation GPS devices are becoming phased out due to the emergence of location-aware Smartphones, that assure precise location from satellites and can augment location from cell phone towers in places with poor satellite signals.

Smartphones

Pervasive Smartphone devices have gained popularity recently for mobile and internet communication. Many mobile applications (e.g. social media, maps, dating apps, locations and others) are used daily on smartphones by their users. Location-aware applications are common in smartphones, they observe the location of the user and report to a location based service (LBS). Location Based Services provide queries of point of interest within a defined proximity of the user as reported by the smartphone. As an example, a Smartphone user can ask (query) for restaurants near-by or within a distance of his/her current location to receive a list of matched restaurants. Smartphones have in-built Assisted-GPS sensor for precise location tracking to satellites, in cases where cloud visibility is achieved. At places with less cloud visibility, Smartphones can gain location by connecting to nearest cellphone towers or WiFi access points. A large body of literature has contributed to the use

of Smartphone in transportation studies.

Patterson et al. [34] conducted an experiment on participants from Concordia University, that used a smartphone travel survey developed to collect passive data on human mobility whilst minimizing the respondent burden. Respondent burden is reduced in such surveys relative to traditional self-reported surveys. An enormous amount of location-sensitive data is gathered on social media platforms like Facebook, Twitter, Instagram and others.

Information Inference from BTD

Another research area receiving attention in the transportation literature is that related to the development of methods allowing the inference of the main aspects of transportation demand required for traditional trip-based transportation demand forecasting. As such, data inference methods have been developed in the following areas. The inference of trip ends was one of the earliest questions to be broached in the literature (e.g. [112]), but one which continues to have (primarily rule-based methods) methods developed (e.g. [34, 114]). Mode detection has received the greatest amount of attention in the literature with methods evolving from rule-based (e.g. [115]) to discrete choice (e.g. [116]) and machine-learning approaches [117, 117, 118].

Purpose detection, has turned out to be the most difficult to infer. Initial rule-based (e.g. [119] continue to be used (e.g. [120]) but are being replaced with machine learning algorithms(e.g. [121, 122]) increasingly using data collected from various BTD sources such as social media (e.g., [8]).

Finally, itinerary inference has evolved from simple map matching methods (see [123]) to more sophisticated probabilistic approaches [116]. Itinerary inference have been applied primarily to road networks and particularly to automobiles (e.g. [116]) and bicycles (e.g. [124]). Less common are methods for inferring transit itineraries combining smartphone and GTFS data [125].

Future sources of BTD

In addition to current sources of BTD, we also have to include the coming addition of autonomous vehicles as a data source. According to Intel[126], the evolution of Autonomous Vehicles (AV) with their on-vehicle sensors and cameras will generate and require enormous amounts of data. AV cameras alone will generate 20 to 40 Mbps per vehicle, while radars will generate between 10 and

100Kbps with an estimated average of 40 terabytes of data for every eight (8) hours of driving [126].

Summary of Current BTD Research

As can be seen from the rest of this section, there is a great deal of research being done on BTD in transportation. Collectively this work can be divided into three broad categories. The first category relates to the use of various technologies in actual data collection [127, 128]. The second category concentrates on challenges related to methods that process BTD and seek to infer information from it that can be useful in transportation [8]. The third category focuses on the evolving technologies that present opportunities for the successful implementation of BTD. While this work is clearly necessary for BTD to be effectively used in transportation, there has been little emphasis on the importance of system architectural components necessary for large-scale adoption of BTD.

3.8 System Architectural Components

Critical to understanding the challenges of BTD is an understanding of data system architecture more generally. Data Management Architectures (DMAs) organize the flow of data from collecting devices to the storage systems with which data is managed. DMAs can be split into three essential elements. First is the physical infrastructure (i.e., hardware) needed to be able to store data. Second are file systems with which files (and their underlying data) are organized on hard drives. Third are database management systems.

3.8.1 Hardware

We begin with the hardware side of data management systems and with data retrieval. Data retrieval typically, and traditionally, involves an in-between step; data must be read from long-term storage on hard drives into active memory. The speed with which this happens is dependent upon three elements: computer processor (CPU), disk characteristics, and disk connection to active memory. The faster the processor, the faster data can be read into active memory [129, 130]. Disks themselves vary in the speed with which data can be accessed from them. Traditional spinning Hard Disk Drives (HDDs) have slower transfer speeds than Solid State Drives (SDDs), from which data can

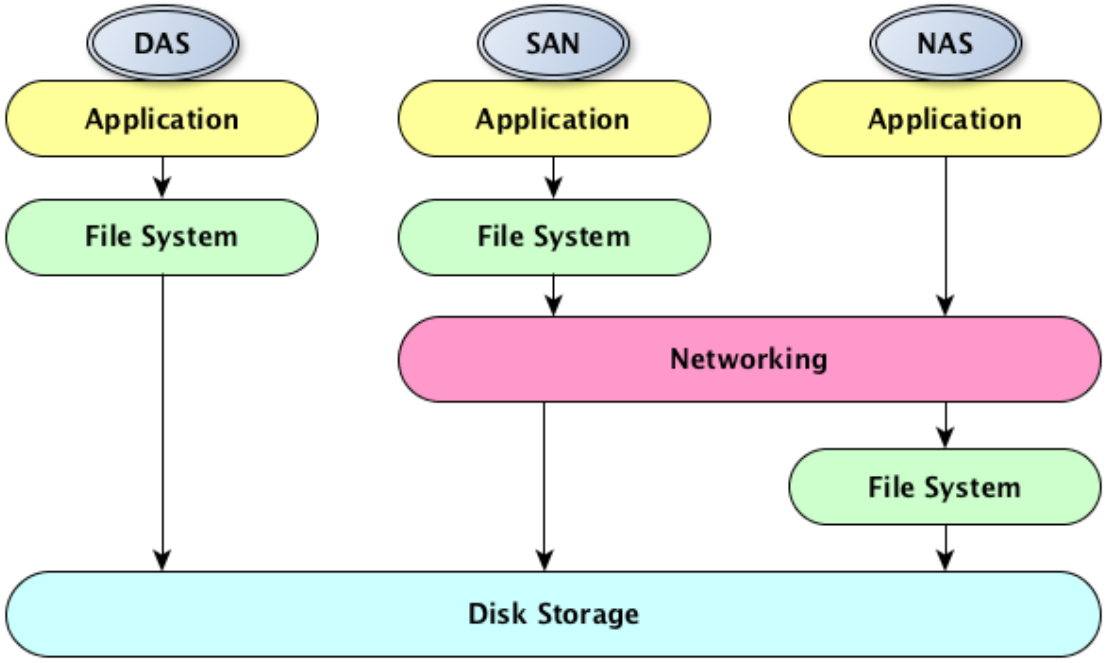


Figure 3.2: Disk Storage Drives

be accessed directly from its storage sector [131]. Finally, the connection between hard drives and active memory plays a critical role in the speed with which data can be accessed, See Figure 3.2. Transfer speeds are fastest from directly attached storage (DAS) (i.e. hard-drive on a single node, such as a server or other standalone computer). Speeds decrease with a greater separation of where the data is stored and active memory with network attached storage (NAS) (i.e. connected through a local area network (LAN) having slower speeds than DAS, and storage area networks (SANs) (e.g. storage on remote networks) potentially taking even longer than LANs [129, 132]. The writing of data to storage involves the reverse process, i.e. from active memory to final storage.

3.8.2 File Systems

On hard drives, data is stored hierarchically. At the lowest level, data is stored in a binary format as bytes with a location on a hard drive [129, 130]. Bytes are grouped together as "data" (e.g. the content of a spreadsheet cell) and data are grouped together into files. There are different underlying logical systems by which bytes can be organized into data, and data into files. These logical systems are known as "file systems," that are a subsystem within the operating system (e.g. Linux, Windows,

MacOS, etc.) [130, 133]. There are many file systems that exist, but the most common are NTFS, VFAT, EXT3 and HPFS [133].

3.8.3 Database Management Systems

While file systems hierarchically organize data and files on hard drives, database management software uses the file system to make data available for processing. This is done with database software. The traditional and most popular database software products are based on Structured Query Language (SQL). SQL resulted from the work of E.F. Codd who introduced the“Relational Model” in the 1970s [134]. As a result, these products are also known as Relational Database Management Systems (RDBMS) of which there are many examples (e.g., MySQL, PostgreSQL, Microsoft SQL Server, Oracle DB). RDBMSs, now typically referred to as“legacy“ systems, have proven very efficient for intensive amounts of data storage, retrieval and processing for many decades [135]. RDBMSs are organized into databases containing tables, with tables related to each other by common identifier constraints (i.e., keys). Database table schemas are strictly defined. That is, data can only be read into them if it adheres to the structure defined in the schema (e.g. text data cannot be read into a variable defined as an integer). The structure placed upon the data is a primary factor making such systems so efficient at saving and accessing data. Also, RDMBSs are typically“centralized” meaning they are deployed on one node and cannot be easily scaled to multiple nodes.

Finally, RDBMSs are “transactional“ [136, 137] which means they also demonstrate the following properties. First, Atomicity guarantees that all transaction operations are executed “all-or-nothing”; if one part of a query fails, the entire query fails and none of it is executed. Second, transactional Consistency guarantees every transaction will bring the database from one valid state to another. Third, Isolation ensures concurrent transactions (e.g. from multiple users) will be executed sequentially. Fourth, Durability ensures that once a transaction has been committed, databases remain the same in the event of a power loss, system error, crash, etc. Collectively these four characteristics are known as “ACID“ properties of a transaction [137].

3.9 Challenges and Opportunities in "Storing-It-All"

The first challenge identified in the literature is to actually being able to store and manage all the BTD. This concerns the "v-word" "volume." The volume of data that will need to be stored is a challenge for using Big Data in general, but is clearly also a challenge in transportation in particular with the many new sources of data (described in Section 3.6) available with transportation applications. As an example, it is now possible to record mobility traces collected by cell phone operators, traffic information, transaction systems (integrated ticketing, road user charging, car park payment, electronic fee collection), cameras, in-vehicle GPS, social media and smart phone geolocation technologies [93]. The rich data gathered from these sources will help to improve on transport modelling and planning to deliver accessibility, efficiency and economic performance potential which hitherto was not possible. Ultimately, this boils down to adding capacity; faster CPU, hard drives with more storage capacity from which data can be accessed (and written) more quickly, and enabling software. Such capacity can be added in two ways; vertically, or horizontally (see Figure 3.3).

3.9.1 Vertically Scaled Systems

The traditional approach to increasing data storage and management capacity is "vertical scaling". This involves improving the capacities of a single node (i.e., a standalone computer). Since traditional RDBMSs were designed for deployment on such systems, there are few software implications and as a result, vertically-scaling concerns primarily hardware. As such, it entails the use of faster CPUs, the increase of active memory (RAM) and the addition of larger and faster disk drives (e.g. converting from HDD to SSD) as shown in Figure 3.3.

While hardware improvements lead to vertical scaling, there are limitations to just how "high" such systems can be scaled. While Moore's law suggests increasing improvements in CPU speeds, we are limited to the available chip technology at any given time [130, 138], even when considering the possibility of multiple cores on the same node. Secondly, there is no guarantee that Moore's law will continue into the future [130, 139]. Similarly, capacities are limited by available active and long-term storage technologies. Moreover, it may be possible to scale up to required capacity with available technology in some circumstances, but component cost increases dramatically with

improvements at the cutting-edge of performance. Finally, vertically scaling a single node amounts to putting all of your eggs in one basket, the downside of which is that if there is a problem with the vertically-scaled node (e.g., it crashes), data cannot be read or written. In other words vertical integration increases the risk of greater downtime.

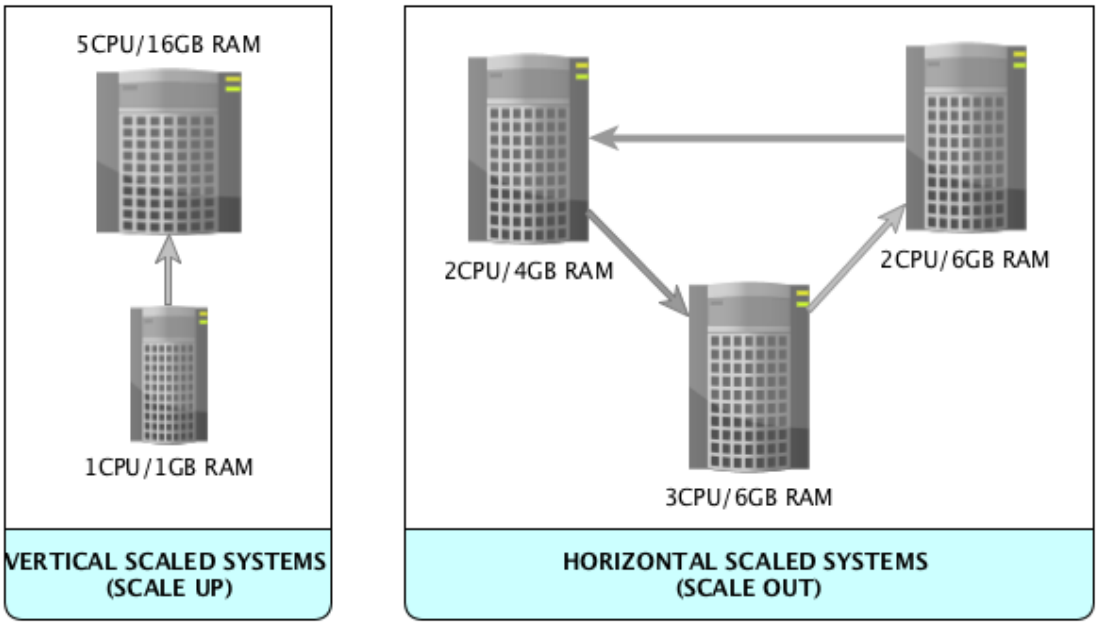


Figure 3.3: Scaled Systems (*Systems sizes are for illustration purposes only*)

3.9.2 Horizontally Scaled Systems

Instead of increasing the capacity of a given node, horizontal"scaling-out" involves the combination of different nodes into a "cluster." That is, a "distributed" storage system. As illustrated in Figure 3.3, nodes with similar (homogeneous) or varying (heterogeneous) capacities are added to the cluster to meet storage and computing needs. Distributed systems have the following advantages compared to single-node systems. First, it is possible to add resources (CPU, active and long-term memory) in a cost-effective manner since capacity can be increased almost limitlessly without the skyrocketing costs associated with performance increases in a single-node.

Second, distributed systems typically store redundant copies of data across multiple nodes, which decreases the risk of data not being available at any given time. The storage of multiple copies

is done in the following ways. The same data can be stored on different nodes. This, referred to as “redundancy,“ means that if one node goes offline, the data is still available on another node. Additionally, data can be be “sharded.” This means that different parts of the same dataset can be stored on different nodes. For example the columns (or rows) from the same database table can be stored separately, thus increasing the speed at which data can be accessed and written.

As with vertically scaled systems, database software is required for the proper functioning of horizontally scaled systems. At the same time, the limitations of traditional RDBMSs make them inappropriate for horizontally scaled systems. A key characteristic of horizontally scaled systems, is that data is synchronized across nodes within the system. Traditional RDBMSs were not initially designed with this in mind, so they remain relatively inflexible in this respect making synchronization with them inefficient and arduous [140, 141]. This inflexibility is ultimately due to the reliance of RDBMSs on traditional, centralized file systems (see Section 3.8.2). Such file systems do not easily allow the management of files across multiple nodes.

As a result, horizontal scaling requires both hardware in the form of nodes and networks, as well as Distributed Database Management Systems (DDMS) that are designed to seamlessly synchronize data across nodes. In order to do this, DDMSs themselves rely on non-centralized distributed file systems. DDMSs and files systems make up the software component of horizontally-scaled systems [141].

Distributed File Systems

The logical hierarchy of centralized file systems locates bytes on a single hard drive and groups the bytes into data and files. Distributed file systems on the other hand use a slightly deeper hierarchy. Bytes are stored on a hard drive, organized into data, data are organized into “chunks“ and chunks into files [130]. Chunks themselves, however, do not have to be stored on the same hard drive. So, in addition to a deeper logical hierarchy, the key feature of distributed file systems is that they can also locate data across different hard drives. While several distributed file systems exist, the most common are the the Google File System (GFS) and the Hadoop File System (HDFS).

GFS, developed by by Google Inc. [12] supports large-scale and data-intensive applications [142]. It can be deployed on any standard node thus making it desirable from a cost perspective when

scaling-out a system. The distribution of chunks across hard drives with GFS is orchestrated by one "master" node to the subnodes ("slaves") of the system. This organization means that if the master node goes offline, access to data on the master and slave drives becomes impossible. As such, GFS is said to have a single point of failure.

The Hadoop File System (HDFS) [143], designed by Apache like GFS also runs on any standard node and is suitable for data-intensive applications. It is also based on a "master-slave" architecture, and as a result also has a single point of failure. Compared to GFS, HDFS has become much more common in industry application, and has had a series of DDMSs built using the underlying HDFS [143, 144, 145].

Distributed Database Management Systems

In addition to specialized file systems, and due to the limitations RDBMS, horizontally-scaling also requires dedicated database management systems (DDMSs). A number of such systems exist and fall into broad categories; structured and unstructured. Basically, such systems are distributed versions of RDBMSs. That is, they allow for the distribution and synchronization of data across multiple nodes, but they remain structured database management systems. The most common such systems in use are Google Big Table [146] and Apache HBASE [147]. Another increasingly common DDMS is noSQL, which in addition to being designed for horizontally-scaling is also unstructured.

3.9.3 Characteristics of Horizontally-Scaled Systems

In order to be effective, horizontally-scaled systems need to be planned well. Key characteristics of effective distributed systems have been summarized in Brewer's CAP Theory [148, 149] (see Figure 3.4):

Consistency (C): While redundancy means having multiple copies of the same data in different locations,"Consistency" means that all copies of redundant data are identical [150]. This ensures that the most up-to-date data is available even if there are server or network failures.

Availability (A): Distributed Systems operate on multiple nodes that run concurrently in the implementation of a task. As a result, individual nodes can stop operating (e.g., due to a crash failure).

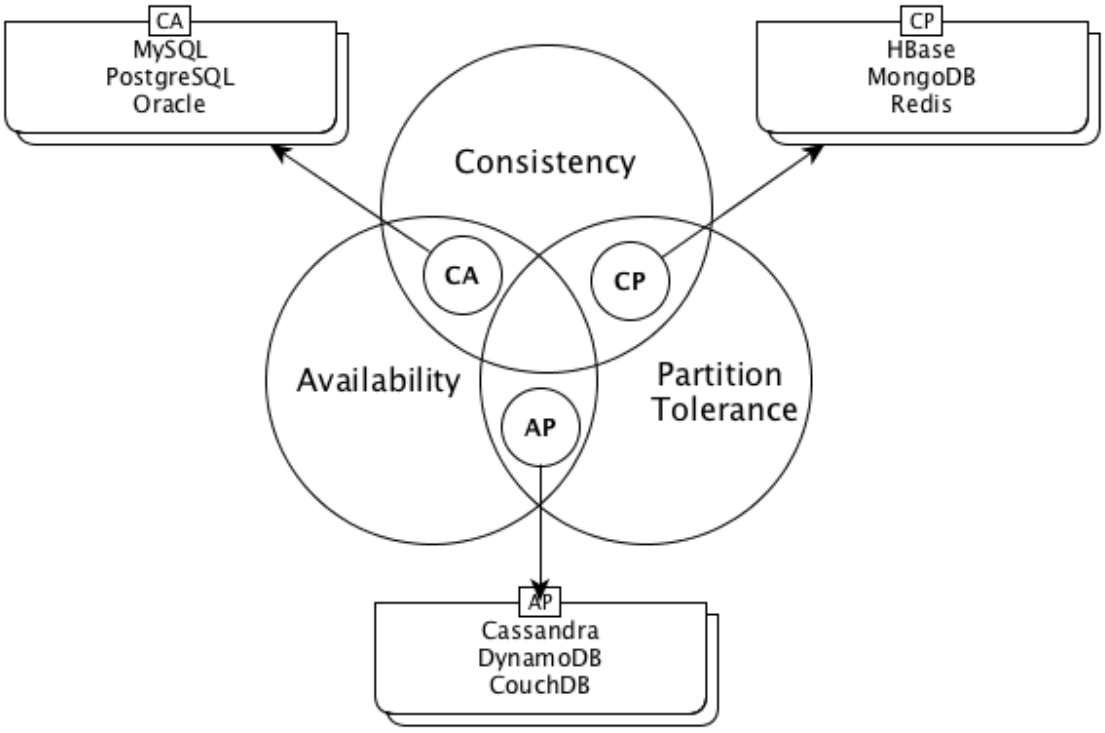


Figure 3.4: CAP Theorem

Such failures are common and inevitable in networked systems. Availability means there is a sufficient number of nodes with redundant data that all data can be accessed at all times, even if one or multiple nodes crash [150].

Partition Tolerance (P): Partition tolerance is similar to availability in that it describes systems where redundant data can be accessed at all times. With Partition Tolerance, however, the concern is not with nodes themselves, but with the network connectivity of the nodes [151]. This can be seen as"network availability."

While ideally, distributed systems would have all three of these characteristics, in practice they are typically characterized by two at most, with system design amounting to trading-off between the characteristics [152]. While systems that are not distributed over different networks exist, discussion on distributed systems is typically limited to those that are. As a result, we describe only systems demonstrating Partition Tolerance that is AP and CP systems.

AP systems are characterized by Availability and Partition Tolerance. Such systems are made up

of multiple networks (P) with a node (or cluster of nodes) (A) on each network. Additionally each node (or cluster) would be able to operate without communication to the others. If communications between the nodes/clusters were interrupted, updates to data would be out of sync and as such, the system is not always consistent (i.e., it does not demonstrate Strong Consistency). Once all networks are functioning, data will become synchronized again but with delays (Eventual Consistency). Well known "AP" systems include CouchDB and Cassandra (see Section 3.10.1 "NoSQL and NewSQL" below).

CP Systems are characterized by Consistency (C) and Partition Tolerance (P). Such systems are made up of multiple networks (P), but with only single nodes on each network. CP Systems maintain multiple copies of the same data and therefore are "Strongly Consistent." Unlike AP systems, if there is a network failure, there is always sufficient network redundancy, that the data across all nodes remains consistent. At the same time, since there is only one node per network if one of the nodes fails, there is no node redundancy, and as a result, the system is not "Available." Well known "CP" systems include MongoDB and Redis (see Section 3.10.1 "NoSQL and NewSQL" below)

As such, the volume of BTD presents a major challenge to the potential to use it effectively in transportation in the future. At the same time, new approaches and technologies, namely the use of scalable distributed systems appear to be the most probable solutions to meeting this challenge, with the design of the systems requiring choices and trade-offs to be made between Consistency, Availability and Partition Tolerance.

3.9.4 Data Storage Opportunities for Transport Systems

The recent advent of sensor-based technologies such as infrared detectors, video detectors, induction coils at bayonet points, laser detectors and others, for real-time traffic monitoring and the passive data collection of mobile user trip data for transport modelling (i.e. mode and activity inference) contribute a rich dataset for real time analytics and decision making by transport stakeholders. In this regard, Damaiyanti et al. [153] presented a novel system that collects traffic data and represents speed values of all road segments of Busan. Their system stores traffic data and supports traffic congestion queries in a distributed NoSQL document database system that is deployed on a MapReduce framework. The rapid rate at which transport data is ingested in an ITS ecosystem, as earlier

discussed, makes an adaptation of a distributed database system a requirement to achieve an effective and performing transport system. The United States Department of Transport [154] has stated the data streams rate within 10 and 27 petabytes per second of connected vehicle Basic Safety Messages (BSM) will be generated, and thus that connected vehicle-to-vehicle (V2V) infrastructure is being implemented in test tracks. These implementations require a large volume of distributed data warehouse capacity. Amini et al. [155] proposed a comprehensive and flexible architecture based on a distributed computing platform for real-time traffic control. Using a mapReduce framework, their distributed architecture is based on systematic analysis of the requirements of an existing traffic control systems and analytics engine that informs the control logic.

3.10 Challenges and Opportunities in Unstructured Data Storage

The second challenge identified in the literature is being able manage BTD of many different data formats. This concerns the v-word, "variety." As with data volume, this is a challenge to using Big Data more generally, as well as BTD.

In general, data can be formatted on a continuum between structured and completely unstructured data. Structured data (described in Section 3.8.3) is highly organized and format schema are defined before data is even collected (i.e., before it is stored in a database). In fact, if structured data is expected for a relational database but the data is not sent in the pre-determined format, it will typically not be stored at all. On the other end of the spectrum is unstructured data. Unstructured data is negatively defined as that not adhering to any predefined data schema. It comes in two main types; text and non-text. Examples of unstructured text data are email messages, text documents, etc. Examples of non-text unstructured data are satellite images, CCTV videos, etc. In between structured and unstructured data there also exists semi-structured data. Semi-structured data encapsulates unstructured data within a meta-structure using semantic tags and marking. Common semi-structured formats include mark-up languages (e.g. HTML, XML) and JSON (Java Script Object Notations). Different formats present two major challenges. First, mechanisms are required to be able to save and access the data in an efficient manner. Recall that structure in traditional RDBDMSs is what allows them to efficiently manage large amounts of data. Second, taking advantage of BTD also

means taking advantage of different sources of data, typically in different formats, so integration of the different data sources is a challenge. Being able to use data of different formats ultimately requires the use of software that can accommodate a variety of formats in a structured manner that also allows efficient retrieval. The most common DDMSs rely on frameworks based on NoSQL [140, 156] with NewSQL being a more recent and quickly evolving framework.

3.10.1 NoSQL and NewSQL

NoSQL databases (i.e., non formally structured relational databases) are becoming more popular for big data storage. NoSQL databases are much more flexible allowing the following features that are impossible in RDBMSs: the ability to add new variables and modify existing variables within tables, without the need to drop and recreate tables; support for copying and pasting data into and from tables; more flexible integration of different programming platforms through Application Programming Interfaces (APIs); eventual consistency (see Section 3.9.3), and supports the management of data across nodes and in quantities too large for one node. At the same time NoSQL systems are not transactional, and as a result do not demonstrate ACID properties (see Section 3.8.3). NoSQL databases are becoming the core technology for big data and can be characterized according to one of four data models: key-value, column-oriented, document-oriented, and graph. We describe these models below.

In Key-value databases each observation (row) is stored as a dictionary, with each key defining a variable. Queries can be made directly according to keys. Such databases are characterized by high expandability (easy to add or remove variables without having to create new tables) and shorter query response time than those of relational databases. These databases have suitable storage structure for continuously growing, inconsistent values of big data for which faster response of queries is required. Key-value databases provide support to large-volume data storage and concurrent query operations. Popular examples of Key-value NoSQL DDMSs are MongoDB [157], Cassandra[158] and DynamoDB[159].

Column-oriented databases store columns of data separately, unlike RDBMSs where data are stored in the form of complete records. They are suitable for vertically partitioned, contiguously stored, and compressed storage systems. Reading of data and retrieval of attributes in such systems is

quite fast and less resource intensive than RDBMSs, as only the relevant column is accessed and concurrent process execution is performed for each column [129]. Column-oriented databases are highly scalable and Eventually Consistent. Examples of Column-oriented DDMSs are HBase [160] and HyperTable [161].

Document-oriented database are similar to key-value DBs and store data in the form of key and value as reference to a document (i.e., a file). However, document databases support more complex queries and hierarchical relationships. This data model typically uses the JSON format and offers very flexible schema [162]. Although the storage architecture is schema-less for structured data, indexes are well defined in document-oriented databases. SimpleDB is the only database that does not offer explicitly defined indexes [163, 164]. Document-oriented databases extract metadata to be used for further optimization and store it as documents. CouchDB [165] and SimpleDB [166] are two examples of Document-oriented DBs.

Graph databases are extensions of Key-value databases. As such, each observation (row) is stored as a dictionary or a series of nested dictionaries (primarily in JSON format). The nested dictionaries contain relational structure. Graph databases offer persistent storage of objects and relationships and support simple and understandable queries with their own syntax [167]. This allows data to be linked together directly, which can be accomplished with one operation making querying more efficient. Modern enterprises are expected to implement graph databases for their complex business processes and interconnected data, as this relational data structure offers easy data traversal [168]. The most common Graph DB is Neo4J [168].

Finally, NewSQL is an emerging DDMS technology that extends NoSQL approaches while building upon attractive features of traditional RDBMSs. Whereas NoSQL does not provide ACID guarantees for database transactions, NewSQL approaches do. As a result, NewSQL approaches combine the best of traditional RDBMSs and NoSQL approaches. At the same time, NewSQL are rapidly evolving and do not always have extensive support. As a result, we mention them as an avenue of considerable potential, but which remain in development and an interest for research[78, 169, 170]. The most popular NewSQL frameworks are NuoDB [171], VoltDB [172], Google Spanner [173] and CockroachDB [173].

3.10.2 Opportunities for Unstructured Transport Data

Evolving transport systems ingest data in the formats of images, videos, audio and various other unstructured data formats. As a result, ITS architectures need schema-free databases to store non-related data provided by traffic surveillance and traffic sensor systems, which hitherto could not be stored in traditional RDBMSs. Orru et al. [174], however, built an ITS application with a back-end of a NoSQL database to create a public access of public transport information (of GTFS files) all over the world and also search for geotagged photos. NoSQL systems allow for the storage of such schema-less files, which would be difficult to implement in a traditional database. Typically, travel mobility datasets are designed with varying questions (i.e. fields) based on the purpose of the survey that can contain unstructured formats like audio and images. NoSQL databases allow for the efficient storage of travel mobility data. Vela et al. [175] focused on the design and storage of accessible transport routes, obtained by means of crowd-sourcing techniques, in a NoSQL graph-oriented database. The authors adopt a graph NoSQL database to address the integration of accessibility data from three sources, namely; existing open data, private data concerning actual accessible routes obtained through crowd-sourcing, and data from existing traffic sensors. NoSQL databases embrace the capability of a seamless integration of varying and non-related data, which is common in transport systems.

3.11 Challenges and Opportunities in Processing

The third challenge identified in the literature is being able process all of the BTD. This concerns the v-word, "velocity." While processing is required in the management of data (i.e., storage), the main processing challenge is making use of collected data. The methods used to process data are a function of how quickly the processing is required, i.e. whether information is required in real-time or not. There are in general, two approaches to processing BTD: Batch (*ex post*) Processing, and Stream (real-time) Processing. These approaches require implementation using different Processing Engines, or Frameworks. Below we describe the approaches as well as the most common implementing Frameworks.

3.11.1 Batch Processing

Batch processing is the processing of large, complete, static or historical data sets, and provides information after the entire dataset has been collected [93, 140, 176]. In other words, results are not provided in real-time. As an example is OD surveys are conducted until completion of data collection before processing of data aggregation is done.

This approach is mostly adopted when processing finite (or bounded) datasets that are complete, whose size can be estimated, and that are persistently stored on a hard drive. That is, the dataset is unchanging when it is analyzed and includes information for a given period of time (e.g. data from a regional OD survey). The data needs to be complete because the types of calculations done on them require having all of the relevant data, such as when calculating totals and averages. In such situations, datasets must be treated holistically instead of as a collection of individual records. Also, the operations require that the dataset be unchanged for the duration of the calculations. Most common framework for batch processing is Apache Hive [177].

3.11.2 Stream Processing

Whereas Batch Processing requires datasets to be complete and static, Stream Processing systems operate on data immediately as it arrives [93, 178]. As such, the data being processed does not need to be complete or static. Moreover, the size of the "entire" dataset is unknown at any given time until data is no longer collected, i.e. it is "infinite," and its size is irrelevant for Stream Processing. To understand Stream Processing, it is useful to understand Stream Processing workflow.

Typically in a Stream Processing environment, data is received continuously (although not necessarily at a continuous rate), and the data contains information that is not required for the immediate analysis for which results are sought. As such, a first stage of processing is to retain only data relevant to the processing goal. None of the other data is kept or stored. Once data is filtered, processing operations are done on individual observations "one at a time." Stream Processing is well-suited to situations in which results are required in real-time.

An excellent example of situations requiring Stream Processing is Uber, the peer-to-peer ridesharing company [15]. Uber needs to analyze the location of its riders and to match them with the nearest

drivers. They also need to determine the most efficient itinerary for the driver to the rider's origin and destination once picked-up. Moreover, information on the location of the driver needs to be provided to the waiting rider. Once a trip is completed, Uber needs to calculate the cost of the trip and send this information to the rider. All of this requires processing to be done in real-time. An emerging technology for which Stream Processing is already required, and for which it will be required in greater amounts in the future is that of Autonomous Vehicles. While Uber needs to be able to process streamed data quickly, Autonomous Vehicles need to process information (read in data, react) instantaneously.

As with Batch Processing, specialized Processing Frameworks are required for Stream Processing. Also, as with Batch Processing, many such frameworks exist, with the most common being Apache Storm [179], Kafka [180] and S4 [181].

3.12 Challenges and Opportunities in Cyber-Security

The fourth challenge relates to the fact that BTD infrastructure needs to be secured from unauthorized access by an attacker. This challenge is related to ensuring transportation system components are securely protected to avoid vulnerabilities exposed for an adversary to exploit and also protect data as it is transmitted on communication channels. We continue to discuss the context of cyber-security in transportation and known vulnerabilities to be considered.

3.12.1 Cyber-Security of BTD

Recent dominance of high-resolution information gathering devices (i.e. Cameras, transponders, wireless routers) and social systems are on a path of fully connectivity known as "Internet Of Things". A large body of research and standards have evolved on mining rich data ingested by these interconnected devices. Intelligent transport systems, gain access to a wealth of information from interconnected data from GPS location tracking to traffic logs, that aid in public safety, disaster recovery and emergency response. As modern transport devices contain a network of networks made up of embedded communication methods and scope, issues of cyber-security are raised.

Whilst discussion on IT Security is a fundamental challenge to core IT implementation and not limited to Big Data implementation, a scope of Cyber-Security is worth considering as it can impact on the veracity (truthfulness) of the data harnessed on large-scale integrations. Cyber-Security protects against illegal or unauthorized access to information sources and their communication channels which can disrupt service availability for interconnected devices. There is a need for devices and generated data to be adequately secured against attacks, vulnerabilities and exploits. Potential vulnerabilities that could be exploited in transportation include unsecure vehicle-to-vehicle communication, unauthorized vehicle data interception, seizure of control systems like brakes or accelerators. As an example, a group of civic hackers deciphered and exposed the bus location system of Baltimore in 2015 [182]. In 2016, San Francisco transit was hacked to give unpaid access to commuters for two days [183]. It is evident that uncontrolled attacks and vulnerabilities can defame the purpose of intelligent transport systems and incur unforeseen losses that can destroy system implementation. Key vulnerabilities that are of concern in Big Transportation Data implementation are discussed below.

Vulnerabilities of Software Applications: Most common threat to security for Big Transportation Data is exploits undertaken in software libraries and bugs. Software packages and Operating system (or firmware) kernels usually expose vulnerabilities or system bugs that hackers can exploit to gain unauthorized access to control the system. Software updates, patches or fixes are periodically developed by software manufacturers to update known vulnerabilities mostly through automatic system updates. As transportation information systems encompass a wide suite of software components (i.e. web server, database, application framework), it is required system updates from trusted manufacturers are allowed and enforced to ensure a robust secured platform for information share.

Vulnerabilities of Field Devices: BTD ingest high-volume data from dispersed sensor and pervasive devices which are mostly located in remote areas and far from routine supervision. These remote field devices such as traffic lights, cameras, road counter equipments are often in isolated public places and remain susceptible to tampering. Isolated field devices are vulnerable to tampering thus an adversary, who can alter the physical configuration of devices can compromise a system by gaining illicit access to its information source. It is important a level of surveillance is provided for field devices which are deployed in isolated environments.

Vulnerabilities of Communication Networks: Communication devices create an enabling environment for data exchanges between interconnected devices. Network vulnerabilities are well known within wired and wireless network service. Such vulnerabilities allow an attacker to eavesdrop on data packets which are exchanged in the communication channel. Cellular networks, mostly wireless services, are known to be vulnerable to signal intercepts and other threats. Wi-Fi network vulnerabilities are very common in hacker communities, who gain access and exploit the network including devices that are connected. A network map is a sensitive information to an adversary who might be interested in exploiting a transportation system thus its detail should be treated with high confidentiality. Data Encryption and cryptographic algorithms such as Data Encryption Standard (DES) algorithm, Rivest-Shamir-Adleman(RSA) are applied to data packets to perturb the data content as they are transmitted over network channels. The underlying transport layer is made secured by adopting secured communication protocols such as Transport Layer Security(TLS) and Secure Sockets Layer(SSL) which provides privacy and data integrity between communication nodes.

3.13 Challenges and Opportunities in Privacy Protection

Until now we have focused on challenges related to defining characteristics of Big Transportation Data, namely the three Vs. The fifth challenge relates to the fact that BTD often contains personal data explicitly, or personal information that could be revealed by combining or analyzing data that is not strictly-speaking personal, i.e., "Personally Identifiable Information" (PII) [184, 185]. In other words, the challenge is related to ensuring the protection of individual privacy with the use of BTD. This is not a challenge uniquely for BTD, and the challenge of privacy protection in the face of PII has been an issue for a long time (see e.g. Sweeney [42] who experiments identifying personal information by linking voter registration data sets to medical records). As a result, we do not concentrate on the general question of privacy protection with PII as it has received a great deal of former attention (see e.g. [185, 186]). What is unique about BTD is the large amount of temporally and geographically precise location data that can be collected on people. As such, this discussion focuses on the protection of privacy in the context of what we refer to as "Personally Identifiable Location Information" (PILI).

An example is given by Anthony Tockar [187], a summer intern at Neustar, an information-analytics who showed how to extract the exact location and time that celebrities used cabs in New York City extracted from open New York City Taxi and Limousine Commission (TLC) data. By joining the two data sets, Tockar found the cash tips paid by celebrities [187].

Transportation planning agencies have had access to both PII and PILI in the past through routine data sources collected for planning purposes such as Origin-Destination surveys. As a result they have used techniques to protect privacy both internally, as well as when such data is shared with third parties such as consultants and academic partners.

With greater amounts of or more detailed information about people, these methods will need to be adapted. Such adaptation is becoming increasingly important with open data policies (see e.g., [188]), which are becoming more common and which by their nature impose much less control on who and the number of people who have access to potentially identifiable information. An understanding of the techniques used for privacy protection in the context of PII, and available for use with PILI, requires an understanding of underlying data Anonymization Operations. We begin with these and then continue with a description of Anonymization Techniques as they have been applied with PII and how they are applicable to PILI.

3.13.1 Data Privacy and the Need for Anonymization

Information collected for transportation planning and operations purposes can contain "micro-data," i.e., detailed information on individuals and households (addresses, age, sex, etc.) [189, 190]. Data attributes (or variables) that identify individuals are referred to as "Explicit Identifiers." Attributes that do not explicitly identify individuals or households can, in combination with other attributes, potentially identify record owners uniquely [42]. Such attributes (e.g., zip code, sex, date of birth, etc.) are referred to as "Quasi Identifiers." While being able to identify individuals is an issue in itself, it becomes even more critical when "Sensitive Attributes" (e.g, disease, income, etc.) [190] are available.

Another issue affecting privacy protection and concerns is to whom data is available. To best understand the issues surrounding this, we define what we refer to as the Data Chain of Custody (DCC) that describes how data passes from the individual on whom it is collected to the end user of the

data. The Data Chain of Custody is an adaptation of Xu et al.'s [100] data "User Roles."

The chain begins with the Data Owner (same term is as Xu et al.) who is the person on whom data is being collected. The Owner's information is recorded by the "Data Recorder" typically a device, such as a smartphone. The Data Collector arranges the collection, stores and curates the data for the Data Analyst. It can be an individual researcher, a governmental institution (e.g. regional planning authority) or a private company. Data Collectors can collect data for their own purposes, or on behalf of others. Data Analysts process, analyze and integrate collected data for the End User. Multiple roles can be played by the same individual or institution, so that for example the Data Collector might also be the Data Analyst and End User. Sometimes the Data Owner (in the case of Location Based Services) can also be the End User. We quickly provide three examples of BTD and the DCC.

The first example relates to the smartphone travel survey platform, Itinerum [11]. This platform allows researchers to develop and administer their own customized smartphone travel surveys (see e.g. [191]). While the platform also allows some data processing, in this example, we assume that the survey administrator only uses it to collect data and does analysis in-house. As such, this example involves a municipality that undertakes a smartphone travel survey that it will use for analysis of their local transportation system, as the City of Montreal did in 2016 [192]. In such a circumstance, the Data Owner is the respondent with their smartphone being the Data Recorder. The Data Collector is the Itinerum project that is collecting the data on behalf of the municipality. The municipality peforms analysis on the data and therefore is the Data Analyst. Because the municipality will use the analytical results from the collected data, it is also the End User.

The second example is a someone requesting a list of nearby restaurants through Google Maps on their smartphone, also known as a Location Based Query. In this case, the Data Owner is the person searching for restaurants and their phone the Data Recorder. Google is the Data Collector since it developed the app and infrastructure and stores the owner's location data. Google is also the Data Analyst since it processes the request and returns a list of nearby to the User. As such, the Owner is also the "End User". See Figure 3.5.

Lopez and Farooq [193] propose a transportation blockchain system to protect the personal travel information and improve the privacy of respondents to passively solicited data. The proposed system

Figure 3.5: Dataflow across data agents

protect users by making them the data owners and controllers of their personal information and is secured by a private key which can be accessed through smart contracts. The blockchain performs the role as a data collector by assigning keys, maintaining a transactional ledger and smart contracts to the information which the data owner seeks to share. Data Analyst mostly third parties require a smart contract to access travel information.

Data privacy risks are related to the DCC and in particular who, and under what circumstances, has access to the data. A data privacy breach results when someone's identity (possibly associated with Sensitive Attributes) is revealed in a dataset when it is not supposed to be. This can happen unintentionally and with no malicious intent. When it happens intentionally and with malicious intent, it is referred to as "Adversarial." [194, 195]

As the number of people accessing data, and the number of people accessing data whose identities are not known, increases, so does the risk of adversarial data privacy breaches. When data is available to few known individuals (e.g. to data analysts in a municipal planning agency), privacy risks are limited. This is because the people with access are known and typically employees operating under regulations. Also, fewer people accessing data implies lower probabilities of discovery of private information that could be revealed when combining data sources, Quasi-identifiers, etc. This situation is at one end of the privacy risk spectrum with open data being at the other.

With Open Data there are unknown numbers of unknown people accessing data. So, the characteristics of data and the degree to which data is available to known or unknown users determines the risk of the revelation of private information. Privacy protection with PII is implemented with a number of different anonymization operations, which are applied in different combinations in Anonymization Techniques (or Anonymization Models). We first discuss Anonymization Operations and then Anonymization Techniques.

3.13.2 Anonymization Operations

The most popular anonymization operations used in application are: generalization, suppression and perturbation. Generalization performs anonymization on data by replacing some values of an attribute with a taxonomy of its parent value [196, 197]. A set of attributed values are replaced by a general categorical description value (e.g. replacing language spoken at home with English or Other). Generalization operations are mostly applied to quasi-identifiers and sensitive attributes, and reduce the probability of uniquely identifying a record owner. A numerical interval or range is typically used to generalize numerical attributes. Specialization is achieved when generalization is reversed by returning the detail of specific values.

Whereas Generalization works with taxonomies, Suppression also replaces values of an attribute with a special key [197, 198], typically an asterisk (*). As data is suppressed, identifiable values are replaced by special keys to make values non-identifiable. Suppression is generally applied to explicit-identifiers and quasi-identifiers. Suppression ensures personal information is not disclosed. On the other hand, Perturbation performs anonymization by distorting the original data with the addition of noise, data swapping, value aggregation and generation of synthetic data. Statistical approaches are used to perturb data values [198]. Perturbation generally replaces real data values as well so that data does not correspond at all to the original value associated with the individual. When statistical methods are used to perturb data, while attribute values are not those of the original individual, the aggregate characteristics of the attributes are the same as for the entire dataset.

3.13.3 General Anonymization Techniques

Anonymization Techniques use combinations of the Anonymization Operations described above to anonymize PII. The most popular techniques used to limit disclosure of identifiable information are are K-anonymity-based techniques and Differential Privacy. The anonymization techniques address privacy protection under different circumstances of access to data.

K-anonymity-based Techniques

K-anonymity-based techniques are relevant in the following data access circumstances. The original dataset is contained in one or multiple tables and all Explicit Identifiers have been removed. K-anonymity requires that after removal of Explicit Identifiers, each record must be indistinguishable from at least another k-1 records with respect to any given quasi-identifier [42, 198]. For example, when k-anonymized, if a given record has a given value for an attribute, there will be at least k-1 other records with that same value. As such, k-anonymization removes the uniqueness of distinct values for a quasi-identifier through generalization and suppression operations.

While k-anonymity protects against identity disclosure, it is insufficient to prevent attribute disclosure (being able to associate a unique attribute value to a given record). L-diversity on the other hand is a concerned not so much with identity disclosure, but with the ability to associate Sensitive Attributes to a given record Machanavajjhala et al. [45]. An equivalence class (i.e., a set of records that are indistinguishable from each other with respect to a given quasi-identifying attribute) is said to have l-diversity if there are at least l "well-represented" values for the sensitive attribute. As such, this is fundamentally k-anonymity but for the special case of Sensitive Attributes [45]. A table is said to have l-diversity if every equivalence class of the table has l-diversity. As a result, and like k-anonymization, l-diversity removes the uniqueness of distinct values but for a sensitive attribute through generalization and suppression operations.

A common problem of both k-anonymity and l-diversity is that they cannot guarantee the protection of private data if information about the global distribution of an attribute is known, e.g., if someone had access to the entire table containing any given k-anonymized or l-diversified attribute. This problem is particularly acute if the distribution of the attribute in question has few values and/or

is highly skewed towards a few values, e.g. if 90% of the values of tips given to drivers (see example in Section 3.13) in a given dataset were 0, it would be straightforward to infer that a given individual did not leave tips. To address this problem, the t-closeness anonymization technique has been developed.

T-closeness [46] itself is a measure of the degree to which a distribution is skewed towards a few values. As t-closeness increases, a distribution becomes more skewed towards a few variables. The t-closeness technique amounts to adjusting the distribution of sensitive attributes to assure that the global distribution does not have few values and is not highly skewed towards any, or a few, of those values. An equivalence class is said to have t-closeness if the distance between the distribution of a sensitive attribute in this class and the distribution of the attribute in the whole table is no more than a given threshold *t*. A table is said to have t-closeness if all equivalence classes have t-closeness. T-closeness is ensured through generalization and suppression operations.

3.13.4 Differential Privacy

Strictly speaking, Differential Privacy is not a technique, but rather a property of the anonymization process. The concept of Differential Privacy was originally introduced by Dwork et al. [43] and is relevant in the following data access circumstances. There is an original database (D) with Explicit- and/or Quasi-identifiers and Sensitive Attributes. There are also two agents accessing the data either indirectly or directly. The data User wants to learn about the characteristics of the original dataset by making queries to it, but does not have direct access to the original data. The Curator (a software component between other software layers, or middlewear) has direct access to the original data, but has the role of modifying it thus creating a new dataset (D') to which the User has direct access.

Critical to understanding Differential Privacy is the notion of Privacy Degradation. Privacy Degradation describes the fact that as queries are made to a database, the results of each additional query provide information that can be compared with previous results. As such, it is possible, all else equal, to learn about individual observations in a modified database by comparing results made with different queries.

Ultimately, the Curator's role is two fold. First, remove Explicit identifiers from the original database and perform modifications (Perturbations) on the Quasi-identifiers and Sensitive Attributes.

These perturbations are typically created by adding noise drawn from a Laplace distribution to Quasi-identifiers and Sensitive Attributes. It is important to note that $D\prime$ itself is dynamic, so that it might not be the same for subsequent queries from the User. The degree to which $D\prime$ is different from D is referred to as epsilon (ϵ). With Differential Privacy ϵ is also dynamic and is a function of the number of queries from the user.

3.13.5 Location Privacy

Anonymization methods discussed so far have been developed and applied primarily to PII [185, 186] data. The large amounts of temporally and spatially precise BTD can be thought of as Quasi-identifying data, but the techniques mentioned above are not suitable to ensure privacy protection with this PILI data. There are two broad categories of circumstances under which anonymization of PILI can take place. The first relates to when data is transferred from the Data Recorder to the Data Collector. The second is when data is transferred from the Data Collector to the Data Analyst. The first category is referred to as Location Based Query (LBQ) anonymization [199, 200]. This might happen for example if the true location of the Data Owner is anonymized or obscured by the Data Recorder before being sent as part of an LBQ, such as a search for nearby restaurants. The second category is when data is transferred from the Data Collector to the Data Analyst. While the first type of anonymization is important, we believe it to be less of a challenge to the use of BTD than the second. This is because with LBQs the Data Collector and Analyst will typically be known and presumably trusted if the Data Owner is willing to share their information with them. Of greater practical concern is what happens as data is transferred from the Collector to the Analyst, since the identity of analysts may not be known, and there may be many, particularly in the case of Open Data. As a result, in this paper we concentrate on techniques relevant for anonymization that takes place between the Data Collector and the Data Analyst, i.e. to data "publishing."

There are four main techniques available for location anonymization appropriate for PILI data when it is published. The techniques differ along three dimensions: whether a Data Owner ID persists or not; the approach used for obfuscating location; and whether or not the anonymization is done real time.

Spatial Cloaking

Spatial Cloaking [201, 202] is used when data is static (i.e. when it has already been recorded and stored). When location data are spatially cloaked the Data Owner IDs persist across observations, but locations reported to the Data User are adjusted. In particular, and instead of providing the original location data (i.e., latitude and longitude), the data are spatially aggregated so that the Data User is provided a spatial buffer known as the Anonymized Spatial Region (ASR) [203]. The size of the buffer is dynamic and is a function of the number of other Data Owners on whom data is reported. In particular, the ASR is large enough to encompass the data of at least k-other Data Owners. As such, it can be seen as a spatial k-anonymization. Since ASRs are dynamic this technique is also computationally intensive. This technique could be used with trip-end location data or trajectory data.

Mixed Zones

Mixed Zone (MZ) [204] anonymization is used when data is static. With MZ-anonymization, it is Data Owner (pseudonym) IDs that are obfuscated and not their locations. This is done first by defining zones through which the Data Owner passes. As the Data Owner passes through zones, their IDs are modified so that it is not possible follow an individual as they pass through the different zones. Mixed Zones is a more general approach that encompasses the special case of the Vehicular Mix Zone approach.

Dummy Trajectories

As with Spatial Cloaking and Mixed Zones, the Dummy Trajectories technique [205] is used to anonymize static spatial data, particularly static trajectory data. As with Spatial Cloaking, Data Owner (pseudonym) IDs persist throughout the data and unlike MZ does not involve the creation of zones. This method amounts to perturbing location data over the course of a trajectory.

Path Confusion

As with Spatial Cloaking and Dummy Trajectories with Path Confusion [206] Data Owner IDs persist. Like the Dummy Trajectory approach, the location data is perturbed directly. Unlike these other methods, the data in question is not static but is arriving in real time. The key concern with this approach is to make it impossible to predict a future location based on the dynamic data. As such, this is a more statistically involved approach that not only perturbs location data directly, but also associated bearing and speed data. Due to the the statistical complexity and the need to treat each data point in real time, it is computationally intensive.

3.14 Cross-Cutting Opportunities and Challenges

The previous sections have focused on the primary challenges facing the widespread use of BTD and the opportunities to overcome these challenges. The opportunities in these sections have included those that are applicable to one of the challenges at a time. In this section we discuss opportunities (and challenges associated with their implementation) that will help in overcoming more than one of the "3-V" challenges. In one way or another the approaches required to overcome the 3-V challenges amount to being able to add computing resources, constrained ultimately by hardware. Cloud computing [207] involves adding resources virtually. That is to say that instead of adding physical resources (e.g. servers), it is possible to add resources through software that mimic the behaviour of physical hardware. This can be done "privately" on infrastructure managed directly or "publicly" by going through Cloud computing providers such as Amazon Web Services (AWS), Google Cloud, Microsoft Azure, Rackspace, etc. Cloud computing allows the possibility to quickly add resources, and thereby scale systems in near real time and even automatically. Using Cloud resources reduces the requirements for internal expertise and allows granular addition of resources where the addition of physical infrastructure is "lumpier."

The costs of using Cloud computing services need to be traded-off with the costs of managing physical infrastructure, but is becoming increasingly competitive for almost all typical computing requirements. It is likely to become even more competitive over time making the choice of using Cloud computing somewhat easier on a cost-only basis. Another issue with Cloud computing,

however, is the loss of control over where data is physically stored (i.e., where physical servers are located). This can be an issue for transportation authorities that have traditionally operated under circumstances where all data is stored "internally." Of course, Cloud computing can be done "privately" although this still requires a great deal of internal resources (more than managing infrastructure directly) and is likely only viable for large organizations.

Cloud resources [208] can be added through three service models: Infrastructure as a service (IAAS), Software as a service (SAAS) and Platform as a service (PAAS). IAAS is the most direct model for adding additional resources. It involves the addition of virtual infrastructure (e.g. computers) that are managed by the service user. As such, software required by the user is installed and maintained on the additional virtual resources. SAAS is the most limited model with users subscribing to particular application software and databases. Microsoft Office online, SQL Server web, ArcGIS online are all examples of this. PAAS is the most involved of the three models. PAAS solutions are designed primarily for technology developers and as a result provide all necessary elements of a development environment. That is PAAS comes with pre-packaged operating system, web server, database and programming languages. PAAS examples include IBM Cloud, Microsoft Azure, Blockchain [193], etc.

3.15 The Future of BTD in Transportation

The rapid emergence of different tools for data collection has led to an unprecedented potential not only to collect, but to integrate data from many sources and potentially to revolutionize how transportation planning and operations are done. This potential has not been lost to transportation researchers, but current research has focused on techniques for collecting data or on inferring relevant transportation information from this data. While critical to fulfilling the potential, we define four existing, higher-level challenges and opportunities to the large-scale use and integration of BTD for planning and decision making purposes. Three of the challenges (and opportunities) are related directly to the 3-Vs of Big Data more generally. A fourth relates to all of the 3-V challenges collectively, and a fifth concentrates on the challenge of privacy protection. This is particularly relevant to BTD due to the large amount of temporally and spatially precise data collected. In our

view, BTD will not be able to fulfill its promise if these challenges are not met. We consider the challenges related to the 3-Vs (Volume, Variety and Velocity) first and continue with those related to privacy.

3.15.1 Challenges Associated with the 3-Vs

The challenge associated with the sheer Volume of BTD that are, and that will continue to be, available in ever-larger quantities will continue to place pressure on traditional vertically-based Database Management Systems. The ability to vertically scale these systems is already at its limits and as a result the future will increasingly (and perhaps eventually entirely) require horizontally scaled systems deployed using distributed architectural approaches. While current approaches are dominated by CA architecture, there is a gradual drift towards AP architecture ensuring high Availability, Fault Tolerance and Delayed or Eventual Consistency. This pattern will continue into the future and AP architectures are likely to become the dominant approach in the near, and for the foreseeable, future. While large Volumes of data present their own challenge, being able to process data coming in at different rates and increasingly in real time is the challenge of Velocity. Traditional Batch Processing methods are ill-adapted to the onslaught of real-time data that also needs to be processed in real time. As a result, in the future the need to increasingly devote resources to Stream Processing methods will become more prominent. While Stream Processing will undoubtedly make up a larger proportion of processing, Batch Processing, when appropriate will continue to play an important role. Batch Processing will remain the mainstay of processing for static datasets and analysis requiring access to a finite dataset. In the future, processing will not simply take place as Batch or Stream processing, but is likely to involve techniques that take advantage of both approaches, such as emerging "Lambda" architectures [209].

The Variety (different data and file formats from different sources) of BTD represents another key challenge. Traditional structured Relational Database Management Systems that require defined data schemas are incapable of handling and integrating data from different sources; something necessary but also which provides one of the most important aspects of the potential of BTD. As such, the move away from traditional RDBMSs and towards more flexible non-relational DB systems will need to continue to cope with the many different formats. The most common production-ready

flexible systems are NoSQL-based and such systems are set to become more commonplace and the *de facto* standard in the near future. At the same time, new approaches are already evolving to overcome the constraints of NoSQL systems and in particular new flexible systems that are also ACID compliant with NewSQL-systems being the most likely to replace NoSQL.

Finally, Cloud computing will be key to meeting all of the 3V challenges. It will provide the possibility to granularly, quickly and automatically add computing resources necessary to cope with increased Volumes, Velocity and Variety of data. The economic case for Cloud computing seems undeniable, but its use will likely involve the necessity to give up the ability to store data internally for most organizations. As such, in order for it to play its facilitating role in allowing BTD to revolutionize planning, organizations will need to be convinced that collected data is stored sufficiently safely. This will likely happen through a combination of attempts on the part of Cloud service providers to convince organizations of the safety of data and an eventual institutional acceptance of using these services.

3.15.2 Challenges Associated with Cyber-Security and Privacy

The last major challenge is that of security and privacy. The first three challenges are essentially technical in nature and if not met, it will simply not be possible to take advantage of the potential of BTD. Privacy on the other had is both a technical, as well as social/political challenge. The social/political challenge is that of Data Owners (the public) being willing to share their data with Data Collectors and subsequently to Data Users. Ensuring this willingness has three elements. The first is that related to security, which is a challenge facing all IT. Network threats have not been dominant in the transport industry as compared to other sectors. Notwithstanding, there is a rising need to build robust and secured transportation infrastructure that is protected from wide range of system vulnerabilities and exploits. As a step to improve security, network threats to existing systems need to be assessed and reported. Network assessment tools (e.g. Wireshark[210]) have become popular for network monitoring to gain better visibility of vulnerability to cyber threats. Enterprise architectures deploy network security systems such as firewalls and proxy servers, which monitors incoming and outgoing network traffic by adhering to strict security rules. A final step to achieve secured computing is to improve communication between trusted cyber-security experts

and operators at national and local transportation agencies. Such communication notifies on active security threats and provide relevant information on managing such threats.

The second relates to the knowledge of Data Users with respect to the nature of their data that is being shared as well as with whom. This has been prominent lately with the necessity for companies to comply with European GPDR regulations. We believe an important challenge related to this is also in the simple and clear explanation to the Data Owners of what data is being collected and shared, something not easily accessible through typical Terms of Use and Consent Forms. The third, is that related to privacy protection and more specifically privacy protection in the context of "published" data. That is, data that is shared with Data Users. Traditionally data was published to relatively few people whose identities were known. The advent of Open Data has resulted (and will increasingly result in the future) in many more people whose identities are not known having access to published data. Moreover, with the anonymity of Data Users, the risk of the adversarial use of such data, particularly with increasing "background" knowledge, and therefore the threat of privacy breaches will only increase. As such, ensuring willingness on the part of Data Owners will increasingly involve assurances around the protection of privacy from collected data. These assurances will be based upon methods of anonymization. As a result, anonymization is critical to ensure the trust of Data Users.

There are already many anonymization techniques that have been developed for the purposes of privacy protection with both tabular as well as geographic data, and this is a lively area of academic and private sector research. At the same time, this is an evolving field and one that will have to continue to evolve as more data becomes open. The primary reason for this is the growth of Open Data for two reasons. First, as more data become open, there will be more people able to access it anonymously and as a result a greater threat of adversarial use of the data. Compounding is the fact that as more data becomes open, more "background" knowledge will also become open further expanding the threat of potential privacy breaches. As a result not only will it be necessary for anonymization techniques to evolve, but caution related to the data that is made open will need to be taken.

3.16 Acknowledgements

This research has been funded by the Social Sciences and Humanities Research Council of Canada (SSHRC) and Canada Research Chairs program.

3.17 Author Contribution Statement

The authors confirm contribution to the paper as follows:

Study conception and design: Godwin Badu-Marfo, Bilal Farooq, Zachary Patterson
Literature review and analysis on frameworks: Godwin Badu-Marfo;
Draft manuscript preparation: Godwin Badu-Marfo, Bilal Farooq, Zachary Patterson;

All authors reviewed the results and approved the final version of the manuscript.

Chapter 4

Perturbation Methods for Protection of Sensitive Location Data: Smartphone Travel Survey Case Study

4.1 Preamble

In this chapter, we implement privacy protection for sensitive home locations using most popular location protection algorithms. We compared the level of privacy provided by both approaches and measure the utility of the anonymized location sets for transportation planning and decision making. This research article was published in Transportation Research Record: Journal of the Transportation Research Board:

Badu-Marfo, G., Farooq, B., Patterson, Z. (2019). Perturbation Methods for Protection of Sensitive Location Data: Smartphone Travel Survey Case Study. Transportation Research Record, 2673(12), 244-255. https://doi.org/10.1177/0361198119855999

It was also presented at the Transportation Research Board (TRB) 98th Annual Meeting in January 2019, at the Walter E. Washington Convention Center, in Washington, D.C.

4.2 Abstract

Smartphone based travel data collection has become an important tool for the analysis of transportation systems. Interest in sharing travel survey data has gained popularity in recent years as "Open Data Initiatives" by governments seek to allow the public to use these data, and hopefully be able to contribute their findings and analysis to the public sphere. The public release of such precise information, particularly location data such as place of residence, opens the risk of privacy violation. At the same time, in order for such data to be useful, as much spatial resolution as possible is desirable for utility in transportation applications and travel demand modeling. This paper evaluates geographic random perturbation methods (i.e. Geo-indistinguishability and the Donut geomask) in protecting the privacy of respondents whose residential location may be published. We measure the performance of location privacy methods, preservation of utility and randomness in the distribution of perturbation distances with varying parameters. It is found that both methods produce distributions of spatial perturbations that conform closely to common probability distributions and as a result, that the original locations can be inferred with little information and a high degree of precision. It is also found that while Achieved K-estimate anonymity increases linearly with desired anonymity for the Donut geomask, Geo-Indistinguishability is highly dependent upon its privacy budget factor (ϵ) and is not very effective at assuring desired Achieved K-estimate anonymity.

4.3 Introduction

Transportation demand modeling helps governments and researchers to better understand human mobility in the delivery of an efficient, intelligent and secure transport system and is highly dependent on quality travel demand data. Traditionally, travel demand surveys (Origin-Destination, regional, household, etc.) actively engage respondents in the collection of travel and personal information. In such contexts, collected data is available to relatively few institutions and individuals who are typically employed and under contract not to use the data for any purposes apart from those strictly related to their responsibilities (e.g. transportation planning). These institutions have also been responsible for ensuring the protection of privacy when any of the data is provided to other institutions, or the public, i.e. when the data are "published."

Nowadays, there is a proliferation of pervasive devices and technologies (e.g. smartphones, tablets, wearable devices, etc.), with location-sensing capabilities. Travel Surveys are now delivered on these technologies that can collect personal and sensitive information (i.e. biographical data, credit card information, location traces, etc.) passively even without respondents being aware. The mobility of a respondent is typically recorded as trajectories and processed by location based services (e.g. Google[12], Uber[15]), location-aware applications (e.g. Waze[13]) or dedicated travel survey apps [11]). Whilst governments, public and private transport researchers exploit the potential of passive large-scale transport data to understand mobility patterns and travel demand, the threat of personal information disclosure cannot be overlooked.

As witnessed in recent years, numerous high-profile privacy breaches have taken place. There was, for example, an enormous public outcry around the privacy controversy related to Facebook and Cambridge Analytica in 2018 [17]. Another example was provided by Anthony Tockar [187], a summer intern at Neustar (an information-analytics company) who showed how to extract the exact location and time that celebrities used cabs in New York City based on publicly available New York City Taxi and Limousine Commission (TLC) data. By joining the two data sets, Tockar was even able to find the cash tips paid by celebrities [187] to their drivers. These examples have given rise to an interest in data privacy violations and the need for "data agents" [100] (data collectors and analyst individuals or organizations) to protect personally identifiable information.

At the same time, governments have been eager to adopt "Open Data Initiatives" that make data available for free reuse and republishing to everyone, without restrictions related to copyright or patents [30, 211]. Open data agreements between governments, transport operators and travel application developers have been witnessed in the sharing of information for improving transportation service delivery. The City of Toronto in 2017 entered into an agreement with Waze [212] to share and use its real-time traffic and road conditions data, to improve service delivery and navigation in the city of Toronto. Uber also launched "Uber Movement" [213] a platform that shares travel information with cities and transport planners with the aim of helping them make informed decisions in the design of transport infrastructure.

Since there are no controls on who can access Open Data, any sensitive data provided by people

on whom data is collected i.e. “Data Owners” [100], such as respondent identity needs to be protected to prevent privacy breaches by untrusted users with malevolent intentions, or “adversaries.” Geographic points of interest (POIs) can be extracted from trip data and inferences can be made on characteristics (i.e. semantic data) such as religious affiliation, health conditions and political interests of respondents. In line with this, the disclosure of sensitive location information poses a risk and could violate a respondent‘s confidentiality if known to an adversary or untrusted party. Thus as governments adopt “Open Data“ policies, it is important for the location privacy (or geoprivacy) of a subject to be protected to ensure the identity of an individual is not disclosed through location information.

Location protection mechanisms [214] such as spatial cloaking, aggregation and random perturbation are used to protect–what we refer to as–the Personally Identifiable Location Information (PILI) of a subject [19]. Random perturbation techniques endeavor to deliver better privacy, while maintaining spatial fidelity of data to maximize the utility of anonymized spatial data [215, 216], while protecting privacy. Random perturbation methods are used to add noise that displaces/masks point locations in a random distance and direction. A popular random perturbation method, geomasking, is used for preserving location privacy by creating a circular buffer at a specified distance around the location to be masked, from which the perturbed location is selected. Geomasking is the most common method of perturbing an individual‘s location for privacy protection[216, 217]. However, the quantity of displacement applied to a location for masking can at once reduce the utility of the data and, if displacement is small, provide little privacy protection. Among the various random perturbation techniques, two have received the greatest amount of attention and are the most commonly used in practice.

The first, the “Donut Geomask”, which is an implementation of a k-anonymity location privacy protection mechanism (LPPM) [42] and achieves privacy protection by using the underlying neighborhood population density of a point location to determine the obfuscation distance. This geomask technique has been used extensively in the protection of patient health information [216, 217] and crime data [215], both of which require high spatial resolution in order to anonymize data.

The second, Geo-indistinguishability (Geo-I) [218], is an implementation of differential privacy for

location data. It guarantees a respondent‘s location is protected within a specified protection distance with a level of added noise that decreases with the distance, at a rate depending on the desired level of privacy. In other words, the original location is highly indistinguishable from locations that are close to it, and gradually becomes more distinguishable from locations that are farther away [219]. This is intended to maintain anonymity, while at the same time maintaining the utility of the underlying data.

In this paper, we have three main contributions. First, we apply the most common geographic anonymization techniques to the case of residential location of respondents in a large-scale smartphone travel survey, MTL Trajet [220]. Second, we evaluate both techniques with respect to their ability to provide location privacy while maintaining utility of the data. Third, we analyze the distribution of the perturbation distances for their degree of randomness to evaluate the degree to which it would be possible to infer original location by an adversary with prior knowledge of only the distribution of disturbances.

4.4 Problem Statement

The objective of this paper is to protect respondent residential locations collected in a travel survey before data are published. We compare and evaluate the two most commonly used random perturbation techniques (the Donut Geomask and Geo-I) to measure and their efficiency of privacy protection and their effectiveness of data utility. We aim at evaluating the degree of protection offered by both techniques by studying the probability distribution of the achieved perturbation distances.

To achieve this objective, we consider respondents who took part in the MTL Trajet smartphone survey of 2016. MTL Trajet respondents were asked to report their home location(latitude and longitude) as part of the survey. We treat home locations as sensitive, independent points that need to be protected to prevent violation by an adversary who has access to the published travel data. (Note that the City of Montreal never published the residential location data).

4.5 Literature Review

Extensive literature exists on the geographic perturbation of location addresses for privacy protection. This research has been undertaken by numerous disciplines (e.g. computer science, geography, transportation, etc.) that are engaged in dealing with personal identifiable location information (PILI). We discuss a few examples of this in the following section.

Zhang et al. [221] developed a geomasking technique referred to as location swapping. Their technique replaces an original location with a masked location that is selected from all possible locations with similar geographic characteristics within a specified neighborhood. Their technique provided greater anonymity than other random methods by achieving higher k values. (K is the population around the sensitive location that could be associated with equal probability to the perturbed location. If K, were for example 10, then the perturbed location could be equally attributable to 10 different households.)

Allhouse et. al. [216], used the Donut method to provide privacy for sensitive health data using household data for Orange County, North Carolina. The authors determined the actual k-anonymity (the number of households that could be associated with perturbed points) by revealing household locations contained in the county database. They achieved an approximate privacy standard for the households at 99.5% (i.e., 99.5% of perturbed points represented k-anonymities of above a desired threshold).

Abul et al. [222], proposed a technique that creates cylinders within which users move such that at every instant of time, there exists at least k users walking a given distance from others.

Ma et al., [223] implemented Geo-I in protecting the privacy on nearby friend-request location-based services (LBS for short) from stalkers. The authors combined the location approximation technique and the homomorphic cryptography to achieve formal privacy guarantees for LBS users, and achieved a satisfactory quality of the reported location to be used by the LBS. That is, the query results were relevant to the original location of the user, even though only perturbed data was provided to the LBS.

Finally, Chatzikokolakis et al. [219], protected the privacy of LBS users using the principle of Geo-I. Using the foundations of Differential-Privacy, their work protected exact user location, while

providing sufficiently accurate location information to allow satisfactory results to be provided by the LBS.

4.6 Definitions

In this section, we provide explicit definitions of technical concepts and key terms that will be used in the analysis of the paper.

Sensitive location refers to any residential point locations represented in its Cartesian coordinates as latitudes and longitudes that needs to be protected to prevent the identification of a user.

k-anonymity refers to the population within a buffer region of the outer radius around the original point prior to displacement, from which a de-identified cluster case cannot be reversely identified. K is the population around a sensitive location that could be associated with equal probability to a perturbed location. If K, were for example 10, then the perturbed location could be equally attributable to 10 different households.

Protection Radius refers to a circular region around a sensitive location within which other location points existing should be made indistinguishable from the sensitive point.

Location-privacy protection mechanism. These are mechanisms that modify datasets to offer privacy guarantees by adding a level of noise to displace the sensitive location to distances away from their true location. Protected datasets are also referred to as *geomasked datasets*.

Adversary. This is an agent seeking to re-identify true residential location of the user by inferring from sanitized dataset.

4.7 Background on Anonymization Techniques Considered

As described above, the two most widely used classes of anonymization techniques are k-Anonymity and Differential-Privacy. This section describes them in greater detail.

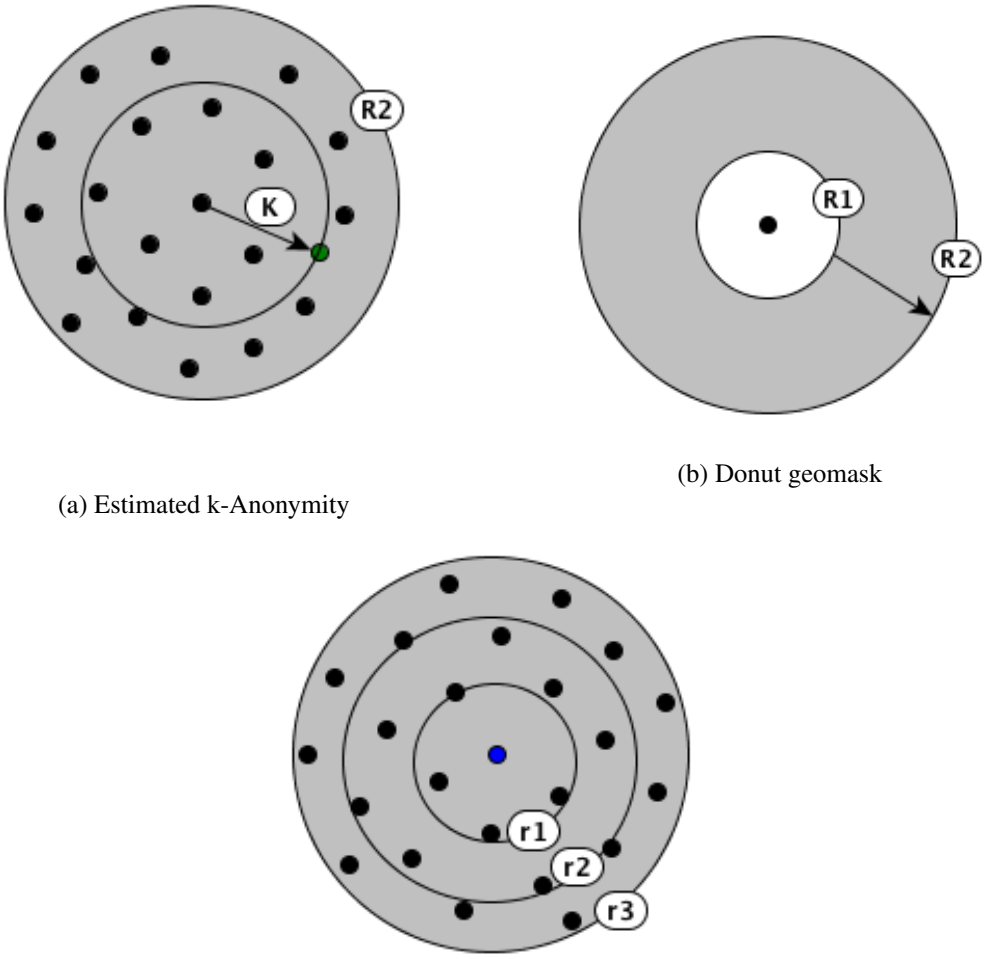
4.7.1 K-Anonymity

K-Anonymity is the most widely used class of privacy protection technique for location-based systems existing in literature. The notion of k-anonymity was introduced by Sweeney in 2002 [42].

Many implementations of k-anonymity aim at protecting a subject‘s identity, requiring that an adversary cannot identify an individual record, among a set of k indistinguishable subjects (i.e. any query result in no less than k observations). k-Anonymity has been used in protecting location (l-diversity), that requires that a set of k-points are spatially indistinguishable. This technique of achieving location privacy using k-anonymity can be implemented through the use of dummy locations [224, 225], where k-1 dummy points are generated and returned as a location-based query result [225]. Another implementation to achieve k-anonymity in location privacy is through the use of spatial cloaking [201]. This approach creates a cloaking region around the real location point with k other points. The cloaking region is then returned as the result of a location-based query and protects the original location by making it indistinguishable among a set of k points in its cloaked region [201]. The first technique we compare examines the random perturbation of sensitive locations using a new adaptive geomasking technique, referred to as the Donut method which has the property of k-anonymity.

Donut Geomask

The Donut method is a geomask technique that protects the privacy of locations by transposing real locations to random displacements within an inner circle radius (i.e. a minimum limit of perturbation) and an outer circle radius that is the maximum limit of the perturbation distance. As illustrated in Figure 4.1b, *R1* is the inner circle representing the minimum displacement from the original location. This method prevents negligible displacements that are close to the original location. On the other hand, *R2* is the outer circle which sets the maximum distance to of random displacement [216]. Random displacements of perturbed points are inversely proportional to the underlying population density, and this guarantees privacy protection of point locations while minimizing spatial error [216, 221]. As an example, while point locations in urban high-density communities will only need to be perturbed small distances, locations in low-density (e.g. rural) areas will need to be perturbed larger distances to achieve the desired level of privacy. The method provides a robust privacy guarantee as maximum and minimum thresholds of displacements are used to prevent negligible or outlier perturbation distances.



(a) Estimated k-Anonymity

(b) Donut geomask

(c) Varying protection distance, r

Figure 4.1: Examples of: (a) Calculating the estimated k-anonymity of a location. (b) Generating protection distance by the Donut approach, and (c) Varying protection distances with Max K.

4.7.2 Differential Privacy

Differential privacy [43] has gained popularity as a new privacy model for protecting an individual without disclosing the data of a subject when the subject participates in a database, and similar disclosure occurs with same probability when the subject does not participate in the database. This ensures that the removal or addition of any record about an individual in a database does not modify the results of a query. Intuitively, the concept of differential privacy requires that the distribution of the characteristics of two datasets (i.e. the original dataset, D and adjacent dataset, D‘) differing by

only one observation should not be noticeable. This is explained by the notion that an addition or removal of a single record in an adjacent dataset does not significantly affect the outcome of a query to the two datasets.

It can be illustrated by a scenario where the probability of a query returning a value v when applied to a database D is similar when compared to the probability of reporting the same value to an adjacent database D‘, differing by only one observation. The amount of difference between D and D‘ is parameterized as epsilon (ϵ), or the “privacy budget.” In order to achieve differential privacy, a controlled random noise is drawn from a Laplace distribution and added to a query output. Differential privacy has been applied in the context of location privacy, as observed in [226]. There a differentially private region quadtree is used for both de-noising the spatial domain and identifying the likely geographic regions containing the sensitive locations. The quadtree spatial decomposition enables one to obtain a localized, reduced sensitivity to achieve the differential privacy goal and accurate outputs. The most recent form of this technique is Geo-indistinguishability [218] and is the second technique we include in our comparison.

Geo-indistinguishability

Geo-indistinguishability (Geo-I) is a property similar to that of differential privacy [43]. This privacy model is an implementation of differential privacy to address location privacy protection. Geo-I works with the notion that within a radius $r > 0$, a respondent is protected within r such that the level of privacy is proportional to the radius. This is illustrated by a basic scenario where a real location, *li* is obfuscated by using some random noise to an approximate location that lies in radius r1 as shown in Figure 4.1c. At radius r1, a high level of privacy is achieved, making the real location indistinguishable among the nearest point locations (there are 3). At radius r2 and r3, the level of noise added to obfuscate *li* decreases at a rate that is dependent on the desired level of privacy, “epsilon.“ As an example, an adversary may be able to make a confident guess of the area where a respondent is located, but would not be able to predict the exact location of a respondent within the area [219]. The random noise of perturbation for Geo-I can be implemented from a Laplacian distribution with respect to perturbation distance from original location. This approach is intended to ensure robustness with respect to the composition of attacks, as the level of privacy decreases in

a controlled way (linearly) [219]. We implement an experimental simulation that achieves Geo-I by perturbing home locations of respondents in the MTL Trajet survey.

4.8 Methodology

We assume a sensitive residential location, L_r of a respondent that needs to be protected by adding a random noise to displace the original location to a new location. We refer to the "noised" location as the "Perturbed Location" of the respondent. The distance and direction to which a sensitive location is displaced to guarantee protection is implemented by a location-privacy protection mechanism (LPPM). In this paper, we employ two LPPMs namely GeoI and Donut.

A protection radius, r, is a required parameter for perturbation by both mechanisms and this sets the minimum distance to which a sensitive location will remain indistinguishable among a set of other locations nearby. As illustrated in Figure 4.1b, the Donut geomask method defines its protection radius as *R1*, which is the minimum distance a sensitive location is displaced to ensure indistinguishability among a set of k points, referred to as "k-Anonymity". For the GeoI, the protection radius is the defined circular region around a sensitive location to which other locations within the radius are made indistinguishable by adding a level of noise. We employ a numerical set including 100, 200, 300, 400, 500 as the protection radius (or Max K) for both mechanisms. E.g. 100 is the radius required (the Max K) to ensure indistinguishability with 99 other locations, 200 is the radius required for 199 other locations, etc.

For the Donut geomask, an outer radius *R2* is also defined to be the maximum distance that a sensitive location can be displaced. This limits the extent of perturbation for the Donut method. We experiment by varying predefined k-anonymity levels in calculating sets of outer radii for each point as shown in Figure 4.1c. This allows us to experiment with how different perturbation radius sizes affect the output of desired anonymity results (i.e. the Achieved K-estimate). We undertake the Donut perturbation method of selecting random distance and angles within *R1* and *R2* using a random number generator built into the perturbation algorithm 4.10. The distortion of the perturbation is guided by the region boundary such that a new geomask point does not fall outside the region of the original location. Using the desired k-anonymity level and population density, the outer radius

R2 is calculated and inner radius, *R1*, is estimated as 10% of *R2* in this paper. The outer radius, R2 varies from point to point since it depends on population density. As an example, for low density regions points are displaced at farther distances than in high density areas.

For the Geo-I technique, we maintain the sets of outer radii *R2* for the varying k-anonymity levels as the protection distance within which perturbation should occur. We undertake this approach to examine how a changing width of the perturbation region will affect the results of the geomasked points. To determine the sensitivity of the privacy budget epsilon, we employ varying privacy budget values (0.10, 0.20, 0.30, 0.40, 0.50) that are repeated for each protection distance. This range is typical of what is used in the literature.

We compare the protection levels achieved from both techniques using the Achieved K and Average error distance metrics. We then calculate the Euclidean distances between the original and geomasked points and summarize them in histograms to which we fit the following probability distributions (normal, lognormal, gamma, exponential, Weibull). Since there is a trade-off between privacy protection and the utility (or usefulness) of the perturbed data, we also evaluate the utility of the perturbed points. To do this, we use average spatial error (defined below).

4.9 Evaluation Metrics

In this section, we present a set of metrics that we use in evaluating the effectiveness of the perturbation techniques used to protect true respondent residential location. We discuss the evaluation metrics under three main indicators: distribution of perturbation, location privacy and data utility. We discuss each of the metrics, the steps involved in its execution and the desired output of measure below.

4.9.1 Location Perturbation Distribution

To understand the effectiveness of the perturbation techniques used in this paper, we study the randomness in displacement for each perturbed point by calculating the euclidean distances between the original points and the obfuscated points for each technique. We refer to this as "perturbed distance." The perturbed distances are tested for randomness of distribution by fitting five continuous

distributions (Normal, Lognormal, Gamma, Weibull and Exponential), to assess if they follow a known distribution pattern. We consider perturbed distances that follow particular distributions as lacking randomness and showing weak privacy since, as shown by Farooq et al. [22] an adversary could reversely identity a subject‘s residential location knowing only the parameters of the distribution. Distances which do not fit any distribution are considered to be strong against re-identification by an adversary. The fitted log likelihood values of the different distributions are presented to compare which distributions have the best fit.

4.9.2 Location privacy

In order to ensure a reliable guarantee of protection of a respondent‘s location, we measure the success of privacy achieved in obfuscating such locations using two approaches: the achieved K-estimate and geographic distance. The first metric, Achieved K-estimate is a commonly used measure of privacy protection performance [217]. It is inspired from k-Anonymity [42] and evaluates the accuracy of location privacy by measuring the number of households among whom a specific de-identified subject cannot be reversely identified [216, 227]. This is illustrated by the population of households that can be counted within a circular region with its radius defined by the euclidean distance from original location to its obfuscated location as shown in Figure 4.1a.
The estimated k-anonymity for a sensitive location is calculated as:

$$k_{est,i} = \pi \times D_i^2 \times \left(\frac{N_i}{A_i}\right)$$

where D_{i} is the measure of Euclidean distance between the original household location and its perturbed location, N_{i} is the population of the neighborhood block and A_{i} is the area of the geographic block of the neighborhood. The estimated k-anonymity metric replaces the exact location of a subject with an anonymized spatial region that contains at least K-1 other subjects preventing an adversary from distinguishing a unique subject at a probability of 1/K [228]. Achieving higher Achieved K-estimate values guarantees a higher degree of privacy protection. In our analysis, we compare the derived Achieved K-estimate for each perturbed location and evaluate which perturbation technique provides the highest location privacy protection across a range of protection radii

and varying privacy budgets. We consider a minimum of ten households to be the smallest to ensure privacy protection.

In our second approach, we use the geographic distance metric which evaluates the guarantee of privacy protection by the extent of perturbed distance. In using the euclidean distance between the original and perturbed location, we assume a small geographic distance offers a low level of privacy with weak protection guarantees, whereas for a stronger privacy guarantee, requires higher geographic distances.

4.9.3 Data utility

Discussions on privacy protection generally assume a trade-off between privacy and the usefulness of data after perturbation. In other words, when a higher privacy is achieved, the potential usefulness of the data is degraded especially in a transportation planning context. Consider an example where our application interest is transit assignment for metro users, and that we have information on a respondent‘s home location that we would like to protect/perturb. The greater the perturbation distance to the home location, the more likely the perturbation is to protect the respondent‘s identity. At the same time, the greater the perturbation distance, the more likely it will be that the respondent would be assigned to the wrong metro access station and therefore anonymized assignment results would not correspond with the original results. As such, we assess the utility of anonymized data for the purpose of travel modeling by comparing the amount of spatial error introduced by the perturbation techniques. In our analysis, a minimum observed average spatial error defines a high utility of sanitized data. We calculated the average spatial error as:

$$ASE = \frac{1}{n} \times \sum_{i=1}^{n} D_{i,j}$$

where n is the total number of point locations, D_{ij} is the euclidean distance from original location, *i*, to perturbed location, *j*.

4.10 Experimental Setup

For this analysis, we used training datasets as discussed in the section below. We built on algorithms and source codes that had been developed for both perturbation methods. For Geo-I, we enhanced the differential privacy algorithm which is implemented by Chatzikokolakis et al. [219] in the Location Guard browser extension [229]. Our enhancement provided capabilities for varying epsilon and choosing an attribute field to derive protection distances of perturbation. The algorithm was deployed on Quantum GIS 2.18 [230] running on Microsoft Windows 10 desktop computer. On the other hand, we used and improved existing algorithms for the Donut Geomask, which is built by the Bayesian Maximum Entropy Lab of the University of North Carolina [231]. Our modification updated the libraries for the algorithm to be deployed on the Esri ArcGIS 10.4 platform. The complete source code is available at `http://github.com/gbmarfo/geoperturbation`.

4.11 Experimental Results and Analysis

4.11.1 Training Datasets

We conducted analysis on both perturbation methods (i.e. Donut and Geo-I) using the MTL Trajet data of 2016. MTL Trajet 2016 was a large-scale smartphone travel survey conducted by the City of Montreal using the app MTL Trajet developed by the Concordia University TRIP Lab. The study took place in October and November 2016 [192]. The residential location data used in the analysis came from the questionnaire asked to respondents after installation of the app. Altogether, the home locations of 7,985 respondents were included in the analysis. Whereas the data sets contain trajectory information of trips by users that we could infer sensitive origin destinations, we focused on working with the user‘s reported data that had been definitively labelled as place of residence. Notwithstanding, the techniques and algorithms are applicable to sensitive trip origins and destinations as well.

4.11.2 Analysis on Privacy Protection

In measuring the amount of privacy protection achieved, we calculated the Achieved K-estimate for perturbed points over varying protection distances for values as shown in Figure 4.1c, and evaluated the population that made them indistinguishable if reversely identified by an adversary. This is illustrated in Figure 4.1a. We experimented with varying protection distances to measure the relationship between privacy and perturbation distances. Our analysis on achieved privacy protection is categorized and discussed as follows.

Achieved K-estimate Measurement

We observe for the Donut approach, that the estimate of Achieved K-estimate increases linearly with an increase in protection distance. Thus, higher Achieved K-estimate values are obtained when perturbation distances increase. A higher Achieved K-estimate value guarantees a stronger privacy as a larger indistinguishable population is created around the perturbed point [221] (see Section 4.8).

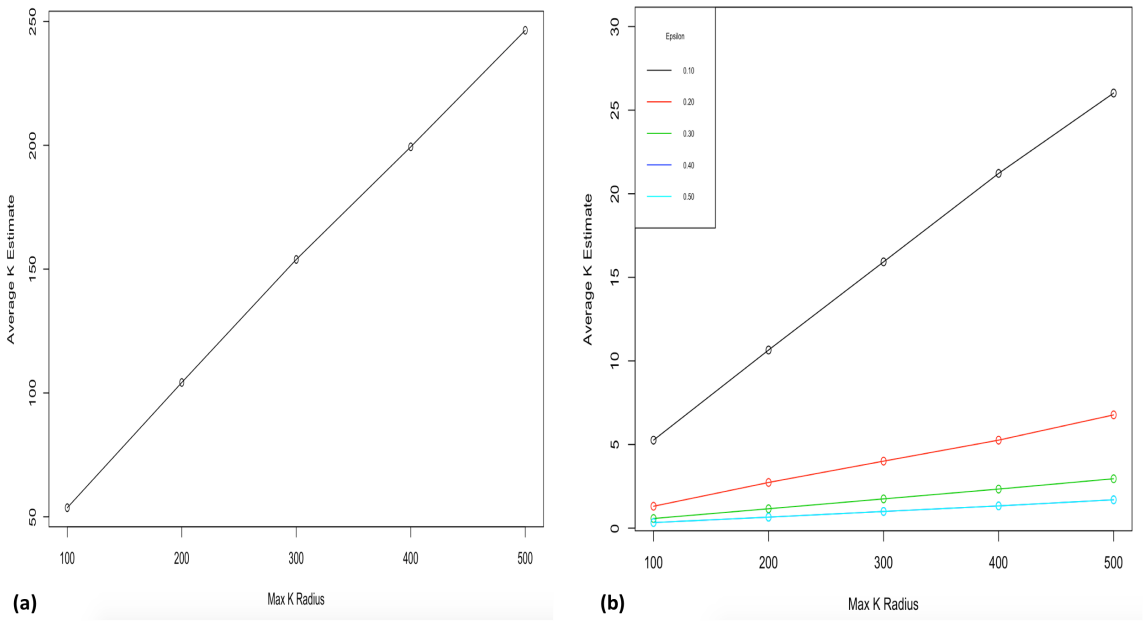


Figure 4.2: A plot of the Achieved K-estimate showing average k achieved vs average radius at Max K (Section 4.8.). The diagram shows (a) Achieved K for the Donut method and (b) Achieved K for the Geo-I method.

The Donut approach using an inner ring radius, R1 as shown in Figure 4.1b, sets the minimum distance of perturbation to prevent perturbation of too small distances. As can be seen this has the effect of ensuring relatively high Achieved K measures that increases linearly with Max K. On the other hand, the Geo-I approach did not guarantee stronger privacy protection over widened protection distances in our analysis. Unlike the Donut approach where Achieved K-estimate values increase linearly with perturbation distances, the Geo-I approach provides privacy that correlates to a minimized privacy budget (i.e. epsilon). As shown in Figure 4.2, the lowest Achieved K estimates (0 to 5) was recorded for epsilons at 0.3 to 0.5 whereas a steep rise to 25 was observed for epsilon at 0.1 which suggest an improved privacy with smaller privacy budgets.

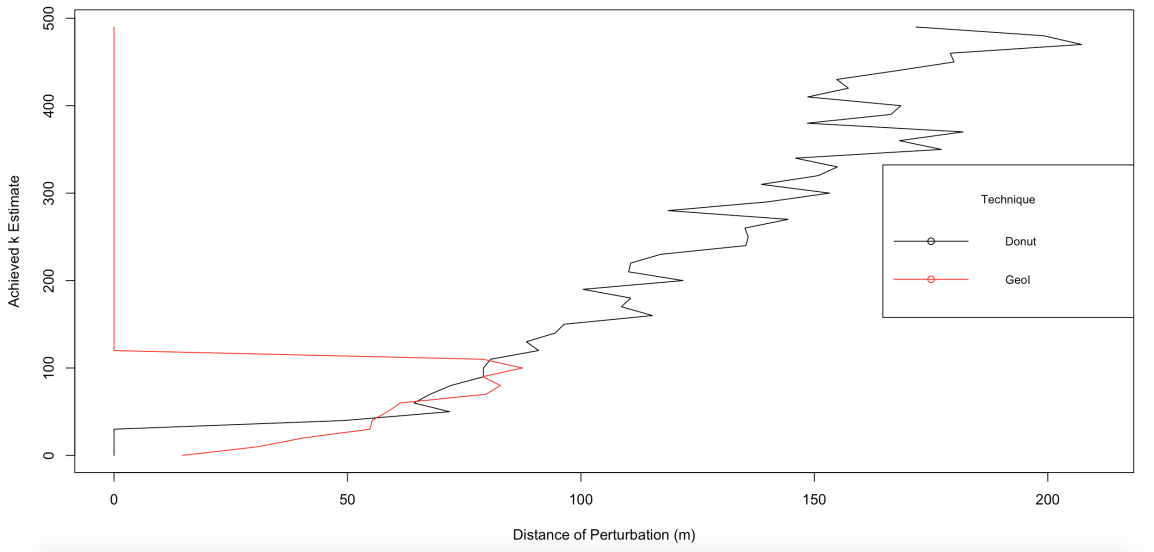


Figure 4.3: A plot of Average Achieved K against perturbation distances.

Our continued analysis incrementing protection distances (with larger Max K) for Geo-I did not impact on Achieved K-estimates whereas the Donut method steadily increased with distances. Our analysis ended with a minimum privacy budget of 0.1 which performed the best of the chosen privacy budget values. Lowering the privacy budget improves privacy protection [232, 233]. The absence of an inner ring in the Geo-I, however, allowed for the negligible distances of perturbation resulting in weak levels of privacy. As such, when considering all levels of Max K and epsilon, the Donut method performed better in the delivery of privacy protection over the Geo-I.

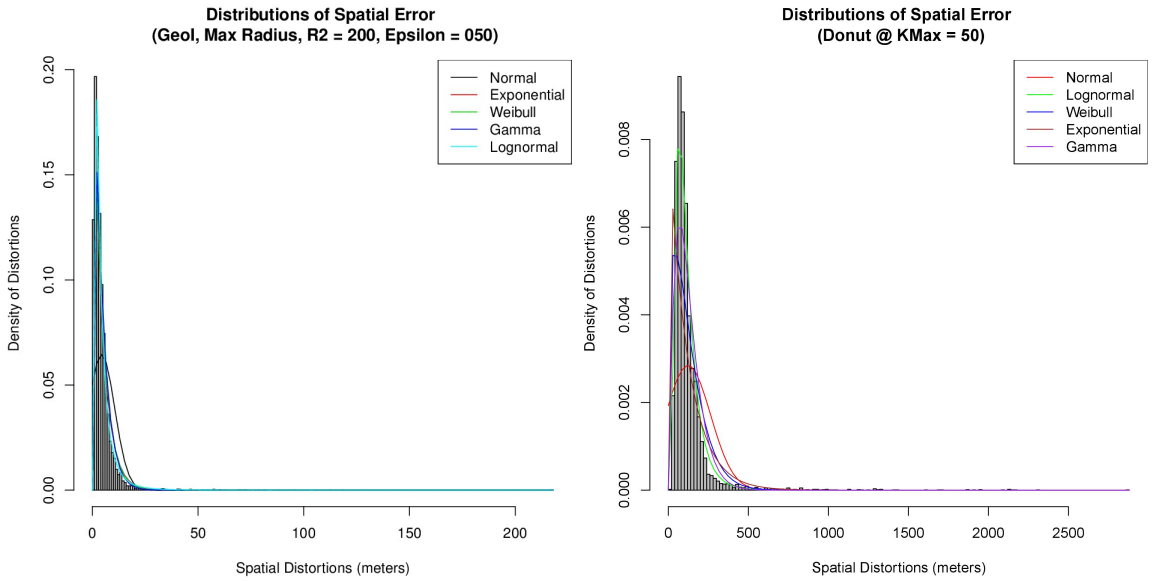
Average K-estimate and Perturbation Distance

We use the optimal performing parameters for Donut at Max K of 500 and Geo-I with Max K of 500 and a privacy budget of 0.1, to evaluate the correlation of perturbation distances and Achieved K-estimates. We break Achieved K-estimates into bins of 10 and aggregate the mean perturbation distances with both techniques. Geo-I records its maximum Achieved K-estimate at 112 and its highest perturbation distance at 87 meters as shown in Figure 4.3. Notice that despite choosing a K-Max value of 500, the Geo-I technique never produced Achieved K-estimate values near 500. At its lowest end, an Achieved K-estimate of 0 is recorded with an average distance of about 10 meters. This denotes very weak privacy protection as negligible distances are observed with an Achieved K-estimate of less than 10, i.e. the locations are highly distinguishable. The Donut method on the other had never produces Achieved K-estimates of less than 10 and results in perturbation distances from 60 to 200 meters. In other words, the Donut does a much better job of ensuring privacy.

4.11.3 Analysis of Perturbed Distance Distribution

As mentioned before, each perturbation technique seeks to protect a respondent‘s location by transposing to a random distance away from its original location. We studied the distribution of perturbed distances to evaluate whether these distances conformed to known probability distributions. If they do, then the original locations can be inferred with knowledge only of the probability distribution and its parameters (see [22] for details).

To do this, we first calculated the euclidean distances between the original location and its generated obfuscated location for each perturbation method. In order to evaluate whether the distributions conformed to known distributions, we fit common continuous distributions (i.e. Normal, Lognormal, Weibull, Gamma, Exponential) using maximum likelihood estimation to the perturbed distance distributions. We also recorded the maximum log-likelihood values for each of the distributions for different values of K-Max and epsilon for distances achieved by both the Donut and Geo-I methods. The lognormal distribution recorded the highest maximum log likehood values for both techniques. The distribution has its greatest density centered about the mode value, where the mode value represents a positive linear skew. We illustrate the empirical anonymization distributions of spatial



(a) Geo-I (b) Donut

Figure 4.4: The plot of density distributions of spatial distortions for both methods.

distortions observed for both perturbation methods as shown in Figures 4.5 and 4.6. The distribution of randomness in the spatial error fits better in a lognormal distribution as shown in the plots.

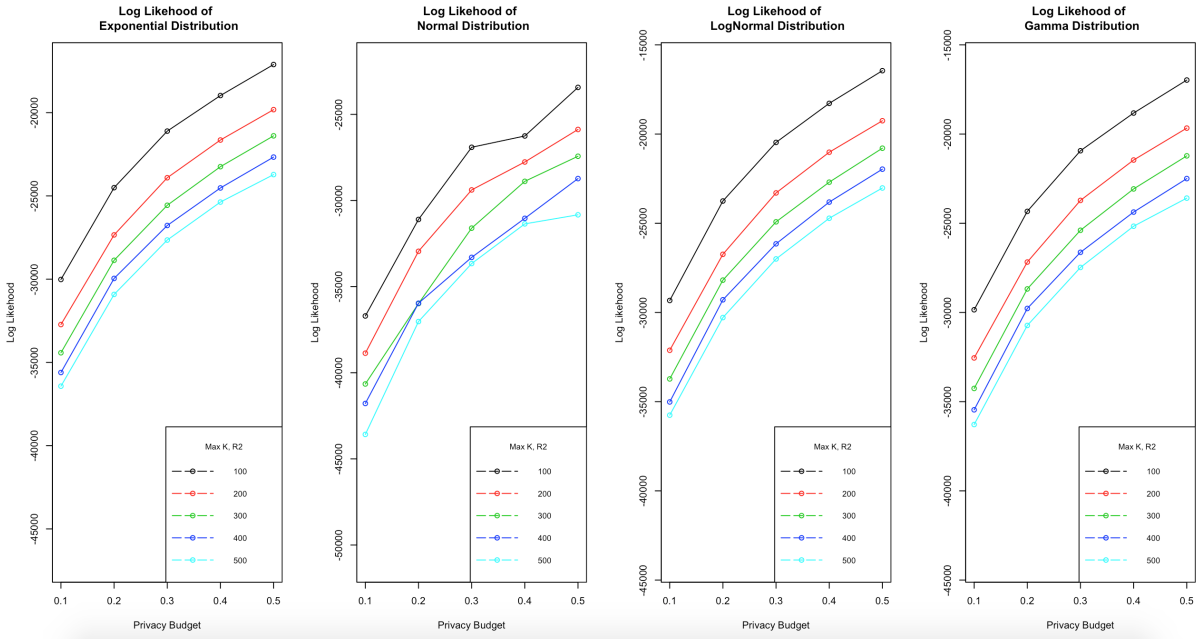


Figure 4.5: Log Likelihood plots for continuous distributions on Geo-I

In order to attest that the randomness of perturbation distances generated by the two approaches are

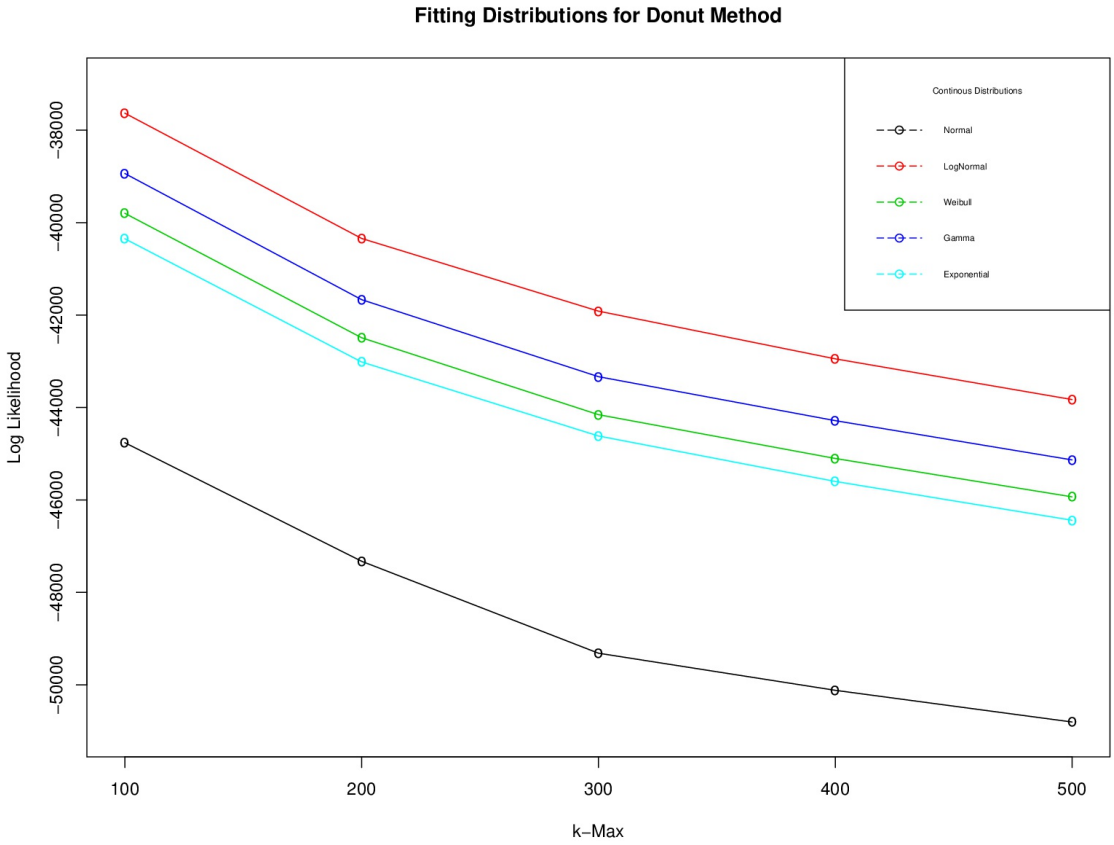


Figure 4.6: Log Likelihood plots for continuous distributions on Donut

closest to a lognormal distribution, we report the average maximum log likelihood statistics of the distributions of fit to the anonymized data sets for both techniques.

Table 4.1: A table showing average maximum likelihood values of continuous distributions fitted on anonymized data for perturbation methods.

Method	**Normal**	**Exponential**	**Weibull**	**Gamma**	**Lognormal**
Donut	-48464.978	-44003.698	-43495.976	-42671.105	-41333.105
Geo-I	-32442.575	-26170.354	-26144.260	-26002.949	-25521.902

4.11.4 Analysis of Data Utility

Finally, we calculate the average spatial error for sets of perturbation distances achieved by varying K Max (i.e. 100, 200, 300, 400, 500) and epsilon. As shown in Figure 4.7, the Donut method shows

a steady increase in the spatial error with an increase in Max-K. The magnitude of spatial error degrades the utility of the anonymized data, however. This implies that to ensure a high utility of anonymity for the Donut approach, the protection distance should be reduced so as to ensure privacy protection, while maintaining the utility of the perturbed data.

The Geo-I method exhibits a high utility on anonymized data as average spatial error decreases gradually with an increase in the privacy budget as shown in Figure 4.7. A high average spatial error was observed at an estimate of 35 meters for the smallest value of epsilon (i.e. 0.1) to a mean protection distance of K Max at 500. Meanwhile, at a high epsilon of 0.5 applying the mean of the same K Max values, an estimate of average spatial error of only about 10 meters was observed. With this observation, the Geo-I provides a high utility relative to the Donut approach, but clearly limited privacy protection.

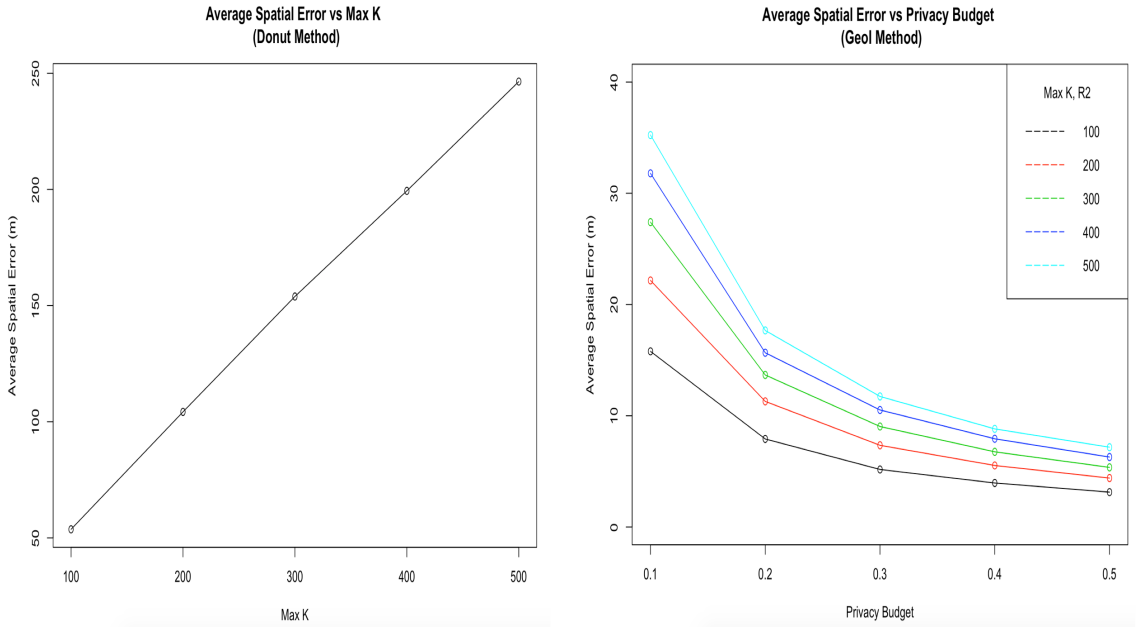


Figure 4.7: Spatial Error observed for Donut (left graph) and Geo-I method (right graph)

4.12 Concluding Remarks

In this paper, we have evaluated two popular geographic perturbation techniques (Geo-indistinguishability and Donut Geomask) to anonymize residential location data from a large-scale smartphone travel

survey. Our results showed that the Donut method performs better for anonymizing location data than the Geo-I method. The degree of privacy resulting from the Donut method increased linearly with an increase of the protection distance thereby making the method sensitive to desired K-anonymity levels. The inner radius used for the Donut method that determines a minimum distance of perturbation provides a great improvement to output perturbation. As observed in our analysis, the inner radius prevented negligible perturbed distances thus this technique guarantees a strong privacy protection of sensitive locations.

On the other hand, Geo-I, which has drawn a lot of attention recently, showed much worse privacy, but did however show promise for preserving the utility of the data. Unlike the Donut method, Geo-I is not sensitive to increased protection distance, but rather to lower privacy budget (i.e. epsilon) values. This was evident in our analysis, where we experimented with a range of epsilon values. At the smallest epsilon value of 0.1, we achieved the best perturbation distance for the Geo-I at an average of 25 meters. As explained by Oya et al. [233], the poor performance of Geo-I is attributed to the fact that counting queries in differential privacy has low sensitivity. This means that an addition or removal of a single record does not significantly affect the outcome of a query thereby ensuring high privacy achievement without introducing much noise. On the contrary, queries implemented in Geo-I demonstrate high sensitivity and therefore require large noise to achieve a high level of privacy.

Notwithstanding, while analyzing the distribution of perturbed distances, we observed that the distances of perturbation closely resembled lognormal distributions for both approaches. This means that true randomness in the resulting displacements from perturbation is not achieved. With this in mind, an adversary with prior knowledge of the perturbation distance distributions would be able to reversely identify a real location from its perturbed point within a high degree of precision.

We find interest in advancing our research into improving the protection efficiency of Geo-I by introducing the concept of inner radius as implemented in the Donut method. A potential solution to improve Geo-I privacy performance which we seek to investigate, is to design its location queries to have lower sensitivity. Further investigation of Geo-I using smaller protection budgets might also prove fruitful. Also, for the Donut method, we would like to work on incorporating location semantics in determining the optimal radius of perturbation as an addition to the existing computation by

population density.

We also acknowledge the scope of this work is focused on the anonymization of independent point samples and that there is also a potential interest for protecting privacy in the context of multi-point data such as trajectories. We hope to further our research into location-privacy protection mechanisms that address multiple trips points and trajectories of mobile users in the future.

4.13 Acknowledgements

This research has been funded by the Social Sciences and Humanities Research Council of Canada (SSHRC) and Canada Research Chairs program.

4.14 Author Contribution Statement

The authors confirm contribution to the paper as follows:

Study conception and design: Godwin Badu-Marfo, Bilal Farooq, Zachary Patterson
Development of generative model and testing: Godwin Badu-Marfo;
Analysis and interpretation of results: Godwin Badu-Marfo, Bilal Farooq, Zachary Patterson;
Draft manuscript preparation: Godwin Badu-Marfo, Bilal Farooq, Zachary Patterson;

All authors reviewed the results and approved the final version of the manuscript.

Chapter 5

Composite Travel Generative Adversarial Networks for Tabular and Sequential Population Synthesis

5.1 Preamble

In this chapter, we present a novel Generative Adversarial Network (GANs) to simultaneously synthesize tabular socio-economic variables and sequential location variables in a travel survey. The GANs developed in this chapter is composed of multiple generators and discriminators operating concurrently to generate the complete trip activity diary. The experiment and methodology highlighted in this chapter extends contribution to the broader subject of population synthesis using deep generative models.

This research article is under review in *Transportation Research Part B: Methodological*.

5.2 Abstract

Agent-based transportation modelling has become the standard to simulate travel behaviour, mobility choices and activity preferences using disaggregate travel demand data for entire populations, data that are not typically readily available. Various methods have been proposed to synthesize population data for this purpose. We present a Composite Travel Generative Adversarial Network (CTGAN), a novel deep generative model to estimate the underlying joint distribution of a population, that is capable of reconstructing composite synthetic agents having tabular (e.g. age and sex) as well as sequential mobility data (e.g. trip trajectory and sequence). The CTGAN model is compared with other recently proposed methods such as the Variational Autoencoders (VAE) method, which has shown success in high dimensional tabular population synthesis. We evaluate the performance of the synthesized outputs based on distribution similarity, multi-variate correlations and spatio-temporal metrics. The results show the consistent and accurate generation of synthetic populations and their tabular and spatially sequential attributes, generated over varying spatial scales and dimensions.

5.3 Introduction

Agent-based transportation microsimulation models study the interaction between the mobility of travel agents and how urban systems operate and evolve through an individual's daily activities [22, 234, 235, 236, 237]. These models help to understand and predict future travel demand, which subsequently impacts transportation networks, environmental sustainability, land and energy usage. Traditionally, individual level data have been collected through phone surveys, household or individual travel diaries and paper questionnaires administered by Census agencies. The proliferation of pervasive technologies (i.e. smartphones, mobile devices, GPS) with high computing power and data connectivity capacities in recent times have influenced the volume, variety and velocity of travel data collected [2]. While data collection technologies are advancing, the availability of microdata still remains relatively limited owing to the high cost of acquiring reliable data and also the threat to privacy of the collection of spatially- and temporally-detailed information on individuals. In practice, government bodies (e.g. census agencies) conduct travel surveys on a sample of a population whose statistical characteristics are used to represent the behaviour of the entire population. Using sample data and other information (i.e. partial views) as base population information, researchers can reconstruct representative members of a population using synthesis techniques such as Iterative Proportional Fitting (IPF) [58, 238], combinatorial optimization (CO) [239], or Markov chain Monte Carlo (MCMC) simulation [22].

Deep generative models have evolved recently and shown the ability to estimate the joint probability distribution of data using deep neural networks and have had success in regenerating high resolution images [24, 65, 240]. Well known deep generative models, such as Variational Auto-Encoders (VAE) [65] and Generative Adversarial Networks (GANs) [24] have gained considerable attention recently for their potential to generate synthetic representations from latent space that estimate the underlying data distributions. GANs have exhibited flexibility in generating high-quality synthetic images and natural language processing [241, 242]. VAEs use a probabilistic graphical formulation of creating models into latent space thus inherently reducing most dimensions into compressed latent representations. This allows VAEs to train efficiently, but their synthetic outputs can be blurry due to drawing from low dimensional latent space. GANs are explicitly optimized for synthetic

generation, and don't have the dimension collapse issues of VAEs. The advantage of GANs is in reproducing realistic synthetic outputs using their adversarial objectives. In this paper, we develop GANs models for population synthesis to estimate combinations of high dimensional synthesized output.

While traditional population synthesis techniques are mostly used for the generation of point estimates and cross tabulations of tabular data, travel behaviour data require spatial and temporal sequences of travel-related activities. Deep neural networks such as Recurrent Neural Networks(RNN) and Long Short Term Memory (LSTM) models [243] have proved successful in generating sequences through modelling the conditional probability distributions of input sequences. Another contribution of our work is to simultaneously recreate the location sequence of a synthesized population using LSTM, while studying the underlying distribution of the trajectory of the sample. To the best of our knowledge, this is the first effort in the population synthesis literature that recreates disaggregate microdata with sequences of locations.

In this paper, we present a novel composite GANs model following the Coupled GANs architecture by Liu et al. [244], having multiple generative and distributive models to learn the joint distribution of multi-domain travel diary data having tabular socio-economic variables as well as sequential trajectory locations. This model is capable of learning the joint distribution by drawing samples from the marginal distributions of variables. In summary, our contributions expand on the current literature on population synthesis as follows:

(1) We propose a composite GANs architecture to simultaneously recreate synthetic representations of tabular microdata *and* sequential locations of travel diary data.

(2) In tabular microdata synthesis, we synthesize mixed features i.e. numerical as well as categorical.

(3) We showcase synthetic sequences of locations inspired by the SeqGAN [1].

(4) We compare and evaluate the performance and similarity of synthesized tabular data distributions to synthesis using Variational Auto-encoders [245].

The paper is organized as follows. In Section 2, the literature review is provided. Section 3 formalizes the problem and introduces the proposed methodology. In Section 4, a case study, evaluation procedure, results and discussion are provided. Section 5 provides a conclusion and some directions for future work.

5.4 Literature review

Traditional population synthesis approaches have been inherently mathematical and can be used to estimate synthetic members of a population having spatial and aspatial characteristics. The aggregate summary of population members corresponds to published aggregates of the entire population. These synthesis approaches are broadly classified into three categories namely, re-weighting, matrix fitting, and simulation-based approaches [234]. First of all, re-weighting methods adopt different techniques to adjust the weight factor of surveys such that the sample represents sub-regions rather than the entire summation of the population aggregates. In this sense, re-weighting applies non-linear optimization to estimate weights and are not scalable to high dimensions [235, 246, 247]. Matrix fitting method evoke expansion factors that are expressed by the ratio between a starting solution and the final matrix. Common implementations of matrix fitting are the Iterative Proportion Fitting (IPF) proposed by Deming et al. [248] and the Maximum Cross-Entropy [249]. It is worth noting that these two methods known as deterministic models, do not produce agent-based samples but rather a sample of prototypically weighted agents [245]. Lastly, simulation-based approaches model the joint distribution of population data with its full set of attributes. New members of the population can be recreated by sampling from the joint distribution. This approach addresses the drawbacks of the deterministic models and is capable of estimating agent-based samples while being scalable to high dimensional datasets. A notable simulation-based approach is the Bayesian Network proposed by Sun and Erath [250]. This method uses a graphical representation of a joint probability distribution, encoding probabilistic relationships among a set of variables in an efficient way. While the bayesian network outperforms the deterministic models, the learning of its graph structure can be computationally challenging [245].

More recently, deep generative models have become popular in the academic literature because

of their outstanding performance and computational effectiveness in producing realistic images and machine translation [251, 252]. Well-known deep generative models are the Variational Autoencoder (VAE) [65], restricted Boltzmann Machines (RBM) [253], and Generative Adversarial Networks (GANs) [24]. These generative models have shown promising results in reproducing the structural and statistical representations of original data by sampling from the estimated joint probability distribution of the underlying data. While GANs have been used extensively for image, sound and sequential text generation, little attention has been paid to its applications in terms of structured tabular data that is mostly composed of numerical and categorical features.

Choi et al. [254] proposed a model that combines auto-encoders with GANs to synthesize private electronic health records. Their method focused on the generation of binary and count variables in health datasets. The authors assert that the original "vanilla" GANs formulation [24] is susceptible to the "mode collapse" problem and difficult to train [251]. Similar work by Park et al. [255] proposed a *table-GAN* to synthesize tabular data using a hinge-loss privacy control mechanism. Their method showed a compatible model for anonymization as sensitive attributes are maintained without change. Recently, Borysov et al. [23] presented a generative model to synthesize micro-agents from a large Danish travel diary to learn the joint distribution of the training data using a Variational Autoencoder (VAE) model. In our approach, the GANs architecture will be optimized for high performance throughput, making it capable of learning all training data records; even those with many zeros representing agents that are omitted from the training samples but exist in the real population.

Generative models have been used in the generation of sequence discrete data, such as text and language translation. Sequence prediction models are typically trained to maximize the log-likelihood (Maximum Likelihood Estimation, or MLE) of the next token (character or word) based on the current token. GANs has had little progress in generating sequence discrete data [256] because the generator network is designed to output continuous gradient updates, which does not work on discrete data generation [240]. In an attempt to solve this discrepancy, Bengio et al.[257] proposes Scheduled Sampling builds on MLE by randomly replacing ground-truth tokens with model predictions as the input for decoding the next-step token. Another approach is to use the concept of Reinforcement Learning named SeqGAN [1]. The SeqGAN approach models the generator as a

stochastic policy where the state is the tokens generated so far and the action is the next token to be generated. The presence of a stochastic policy, REINFORCE [258] algorithm, allows different actions to be sampled during training and derive a robust estimate of the policy. We adopt the SeqGAN approach in our model for the sequential component of the CTGAN architecture whose purpose is to synthesize trip sequences.

5.5 Methodology

The problem definition is introduced, which establishes the objective of this research. As a base case, we briefly present the variational auto-encoder method, which has been recently used for population synthesis of tabular data only [245]. An overview of the Generative Adversarial Networks and subsequently a detailed description of our proposed composite architecture of GANs for synthesizing tabular and location sequences follow.

5.5.1 Problem definition

We assume a dataset on mobility of *N* population agents (i.e. households, families or individuals) characterized by a set of basic attributes $X = (X_1, X_2, X_3, ...X_m)$ where *m* is the number of attributes, and their sequence of time-ordered trips to locations drawn from the universe of locations, U_L. The universe of locations, without loss of generality, consists of geographic positions of all route intersections and road vertices within the study area. Formally, the trip chain is defined by $T = L_1 \rightarrow L_2 \rightarrow \cdots \rightarrow L_{|T|}$ where $\forall 1 \leq i \leq |T|, Li \in U_L$. It is worth noting a location may occur multiple times in a sequence of trip chain especially for home based trips. Table 5.1 shows an example of such a dataset. Typically, the joint distribution between attributes in a true population are not accessible hence partial views such as samples are used to estimate the joint distribution of the population [22]. In this regard, we present a novel generative framework using deep learning methods to estimate the joint distribution of a true population using sample partial views having tabular and sequential attributes, from which we can draw synthetic agents with tabular and sequential characteristics simultaneously.

5.5.2 Variational Auto-encoders

The Variational Auto-Encoder (VAE) was proposed by Kingma et al. [65], as an alternate deep learning approach to estimate a population distribution into a compressed lower dimensional latent space using a neural network called the "encoder" that is supported by an auxiliary neural network named the "decoder", acting as a generator by drawing random samples from the distribution of the latent space. During training, the encoder network receives an input vector of the size of the training data and outputs a latent representation. The decoder network receives the latent representation as input and generates new synthetic agents from the prior distribution of the latent space. Using VAE, Borysov et al. [245] developed a scalabe population synthesis method for tabular data and showed that it outperforms IPF and simulation based methods. Thus we will use VAE as our base case for comparison for tabular data synthesis (Columns 1–4 in Table 5.1). For further reading about the VAE, readers are referred to [65, 245].

5.5.3 Generative Adversarial Networks

Goodfellow et al. [24] proposed Generative Adversarial Networks (GANs), which have gained prominence in the deep learning literature because generative modelling has shown promising results in synthesizing realistic images and sequences for natural language processing. Intuitively, GANs simulates a two player game composed of Generator and Discriminator networks. The goal of the Generator is to generate samples from latent space that are equivalent to real samples while the Discriminator acts as a police officer to distinguish real samples from synthesized ones. Models of the generative and discriminator are both realized as multilayer perceptrons. During model learning, the Discriminator gets better at discriminating real samples from fake, while the generator improves on generating samples that are close to the real samples until a Nash equilibrium [259, 260] is achieved, where each model reaches its peak ability to thwart the other's goal. The

Age ($\mathbf{x}_1$)	**Sex ($\mathbf{x}_2$)**	**Status ($\mathbf{x}_3$)**	**Permit ($\mathbf{x}_4$)**	**Trips (T)**
21	m	student	y	$L1 \rightarrow L2 \rightarrow L3 \rightarrow L4$
30	f	worker	n	$L1 \rightarrow L3 \rightarrow L4$
45	m	not employed	y	$L1 \rightarrow L2 \rightarrow L3 \rightarrow L4$

Table 5.1: A preview of mobility data on travel agents comprising structured and sequential features.

objective function of GANs is defined as:

Definition 1 (Objective function):

The *objective function* of the Generative Adversarial Networks [24] is:

$$G_{min}D_{max}V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[log\ D(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1 - D(G(z)))] \quad (1)$$

Equation 4 explains the objective function of the Discriminator, which seeks to maximize the output of D(x) to 1 when the input is from the true data distribution of the real samples. If the input is generated from the Generator, then D(G(z)) should minimize the output of the objective function. In the training process, both networks simultaneously learn parameters using Stochastic Gradient Descent. The training process halts when a Nash equilibrium is reached so that the Discriminator is unable to distinguish probability from true or fake samples.

5.5.4 Coupled generative adversarial network

The Coupled generative adversarial network (CoGAN) proposed by Liu et al. [244] addresses the problem of learning a joint distribution of multi-domain images from data. While other multi-modal learning approaches exist [261, 262, 263], CoGAN has shown successes in overcoming correspondence dependency [244] which makes it challenging to build a dataset of corresponding images. CoGAN is built on the GANs framework [24] and extends the capability of learning joint image distribution tasks. CoGANs consist of multiple GANs networks each defined for a single image domain. While CoGAN naively learns the marginal distributions of its input data, the authors enforced a weight-sharing constraint to achieve joint distribution learning between the networks and showed its effectiveness in application to multi-image domains, unsupervised domain adaptation and image transformation. We refer readers to the literature [244] for a thorough discussion on the architecture and applications of the CoGAN.

5.5.5 Composite Travel Generative Adversarial Network

The Composite Travel Generative Adversarial Networks (CTGAN) is designed for learning the joint distribution of tabular travel attributes *and* sequential trip chain locations of an agent in a

simultaneous manner, drawing inspiration from the CoGAN [244]. CTGAN as shown in Figure 5.1, consist two GAN networks - GAN_1 referred as the Tabular model, and GAN_2 as the Sequence model.

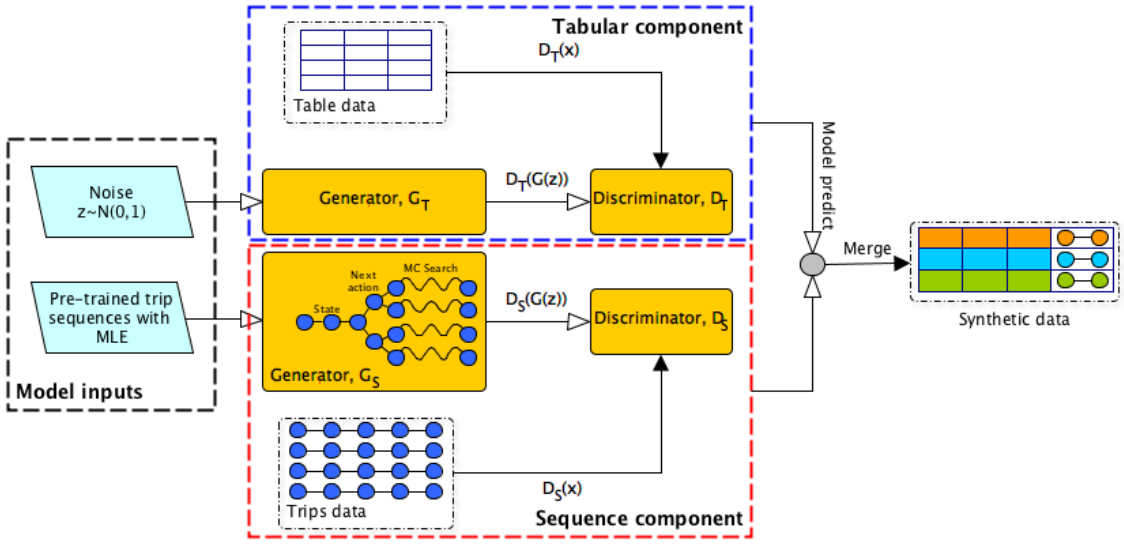


Figure 5.1: The architecture diagram of Composite Travel Generative Adversarial Networks (CTGAN).

The CTGAN has a *tabular* component whose objective is to learn the joint distribution of the basic socio-demographic attributes in the travel diary and a *sequential* component with an objective to learn the distributions of the trips undertaken in a day by an agent. During the training, each component is implemented as an independent network and learns its parameters based on the underlying data distribution. CTGAN then learns to synthesize pairs of tabular attributes with sequential locations of an agent in a population.

GAN_1-Tabular Component

The purpose of the *Tabular* component in the CTGAN is to synthesize the table of records on an agent's socio-demographic and economic attributes (i.e. Age, Sex, Status, Income) which exist in numerical as well as categorical types. GAN_1 is able to synthesize both types of tabular attributes. The *tabular* component shown in Figure 5.2 is composed of an independent GANs architecture having a single Generator denoted G_T and Discriminator, D_T. The Generator, G_T, is made up of

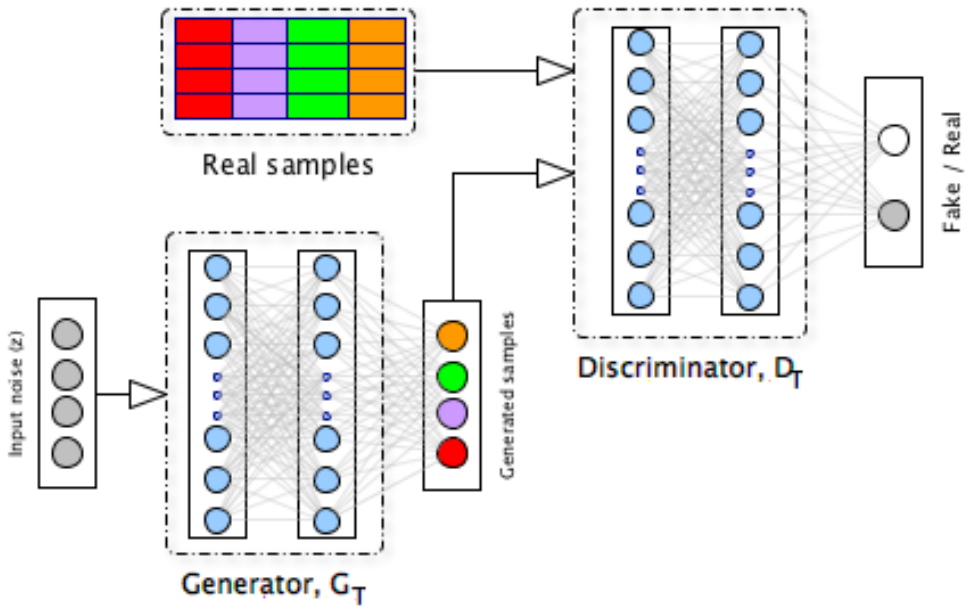


Figure 5.2: The structure of the Tabular component of CTGAN

a Multi-Layer Perceptron (MLP) with neurons for each layer connected to the neurons of the next layer. It takes as input a fixed set of vectors and processes them through three (3) hidden layers to compute a higher level representation of the inputs. A final output layer returns a prediction of a last representation for the corresponding inputs. Similar to the Vanilla GAN [24] implementation, the input layer of the Generator accepts a random noise sampled from a Gaussian distribution with a dimension size equivalent to the size of the real data. In order the depth of features learnt in the neural network, we exploit multiple hidden layers in the network. Each layer has a bias with a Rectified Linear Unit (ReLU) [264] activation applied to its output. The ReLU activation is used because it is computationally efficient which allows the network to convert faster, and easily allows for backpropagation. Due to the diverse nature of the data types (i.e. numerical and categorical), the final output layer is split into categorical and numerical vectors. For the categorical vectors, the Softmax activation is applied while the Linear activation function is applied to the numerical vectors. Subsequently, the activated output layers are merged together as a final output of the generator network. We consider age a continuous numerical feature unlike previous work of Borysov et al. [245] that bins into age group categories using count aggregates. An arbitrary size of 200 neurons

are defined for the first hidden layer, followed by 100 neurons and 50 neurons for the last hidden layer. The choice of neuron sizes was done randomly and the best choice was based on the training performance of the network and distribution of the final output layer.

The Discriminator of the tabular component, D_T, is designed with an aim to distinguish between true data and synthetic data from the Generator, G_T. The Discriminator is made up of Multi-Layer Perceptron with neurons for each layer connected to the neurons of the next layers. The input layer of D_T receives a matrix with the size of the true data shape equivalent to the size of the generated data from G_T. The real data samples are pre-processed prior to being fed into the input layer. The numeric features are normalized to a range between -1 and 1, a recommended approach for optimizing effective learning in neural networks [240]. The binary and categorical features were encoded with one-hot vectors [265] because of the low cardinality of categorical unique values. Each hidden layer is composed of matrix multiplication of nodes with bias and a ReLU activation function. The last hidden layer is activated with a Sigmoid activation function with output of 1 for real samples and 0 for fake samples.

GAN_2-Sequential Component

In the second component of the CTGAN architecture, the objective is to synthesize sequences of location distributions traveled by population agents. As earlier mentioned, the CTGAN is composed of multiple generators and discriminator networks hence for the second network of generator and discriminator, we adopt and integrate the SeqGAN model shown in Figure 5.3 proposed by Yu et al. [1] that has been successful in the generation of text sequences. This network cluster is referred as the "Sequential component of CTGAN." We extend the implementation of this architecture towards synthesizing location sequences knowing that previous work has used the same in text and sentence generation [266, 267, 268].

It is worth noting that GANs have proven difficulty in the training and generation of sequences and discrete data types. By design, the standard GANs were designed to work with continuous or real-valued data, thus the gradients propagated from the discriminator exist as floating or real-valued losses sent to the generator. This implementation limits the suitability of training with gradient descent on discrete data types. Another is in how the discriminator evaluates gradient loss on a

sequence. The discriminator is designed to only classify and evaluate gradient loss on an entire sequence. For instance, only a complete sentence of text can be classified as real or fake by the discriminator but not an incomplete sentence with parts of text. This implies that the loss of a partial sequence cannot be evaluated on how good the partial sequence is until the entire sequence is fully generated.

This scenario cannot be applied to discrete types as they cannot be updated with continuous or real-valued losses. In order to address the drawback of evaluating partial sequences, we adapt the SeqGAN approach to employ an intermediate score mechanism built using Reinforcement Learning [269]. The intuition of Reinforcement learning is illustrated by an agent (a baby) who takes a set of actions (like walking) in an environment based on the state (or thinking) of the agent. When the outcome of the actions of the agent is successful, the agent is given a reward. The objective of this approach is to optimize the actions of the agent and adversely maximize the future expected rewards to the agent. In this regard, the Generator, G_S is modelled as an agent of Reinforcement Learning as discussed. As an RL agent, the state **s** is defined as the tokens generated so far, the action **a**, as the next token to be generated and a Reward **r** gives an intermediate feedback or score to guide G_S by D_S on evaluating the location sequence generated. The gradients from the Discriminator, D_S, cannot pass back to G_S since the outputs are discret. To overcome this, we implement an algorithm of reinforcement learning called "Policy Gradient" which is a stochastic parameterized policy. As a stochastic parameterized policy, the action (next token) may be sampled from a normal distribution whose parameters (i.e., mean and variance) are predicted by the policy. When the samples drawn are evaluated by the policy, subsequent samples can be drawn by moving mean closer to samples that lead to higher rewards, or farther away to samples leading to lower reward. The underlying objective of the generator model is to generate a sequence starting from a state S_O in a way to maximize the expected end reward. We discuss the definition of the end reward in the next section.

Definition 2 (End Reward):

The *expectation of the end reward* is defined by:

$$J(\theta) = \mathbb{E}[R_T|s_0, \theta] = \sum_{y_1 \epsilon Y} G(y_1|s_0).Q_D^G(s_0, y_1) \tag{2}$$

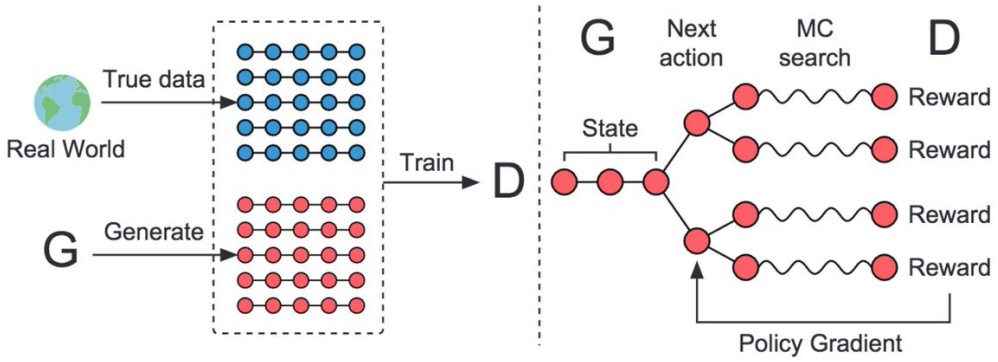


Figure 5.3: The sequential architecture diagram from SeqGAN [1]

The expectation of the end reward R_T given in Equation 2 is derived as the product of possible values of the reward (i.e. the action-value function) and the probability of the value occurring when given a start state s_0, and Generator with parameter of θ. The action-value function $Q_D^G(s_0, y_1)$ estimated by the discriminator returns the reward value for taking an action from the state s_0 following the policy G. The objective of the Generator, G_S is to generate sequences of location destinations from the start state $\mathbf{s}_0$ in a way to maximize the end reward, $\mathbf{R}_T$ determined by D_S. While D_T only rewards the end of a finished sequence, it is important for every action predicted at each timestep of a state be evaluated for fitness. Intermediate scores are thus required. To achieve this, the Monte Carlo search with roll-out policy is used as in SeqGAN. This approach samples the unknown tokens and estimates the state-action value at each intermediate step.

The Monte Carlo search is a tree-search algorithm having a root node, $\mathbf{s}_0$. The root node is expanded while trying all possible actions belonging to the set of action states as a way to construct child nodes for each state. The value for each child node is determined while the remaining tokens are rolled out with a policy until the entire sequence is generated. The Discriminator gives a score accumulated on each node of the MC tree when the end of sequence is reached.

5.6 Data and case study

The experimental evaluation of CTGAN is based on travel data from the 2013 Montreal Origin-Destination (OD) survey conducted in 2013. The data contains the travel diary of 139,354 individuals and includes socio-economic variables such as age, employment status, gender, etc., and other trip related variables such as origin and sequence of trip destinations [270].

5.6.1 Data Pre-processing

Dealing with the mixed data types and complex geospatial types, especially for generating travel survey data poses two challenges: numeric representation, and reversibility. Neural networks work efficiently with floating precision numbers, making it necessary to translate all variables into low-cardinal dimension floating representations and to ensure the uniqueness of each sample represented. Binary and categorical variables are indexed numerically and one-hot encoded [271]. Numeric variables are scaled and normalized within a range from negative one (-1) and positive one (+1). These pre-processing techniques derive a numeric representation of the input data. Unlike regression and classification algorithms that usually have a single output, generative modelling of tabular data requires the vectors of the final output layer to be easily reversible to readable formats synonymous with the raw input data. Thus, encoding techniques of input data to numeric representations must be easily reversible with the ability to be decoded to the format of the input data. In our work, we used Scikit-Learn [272] label encoding and OneHot encoders which have reverse encoding capabilities. The geographic coordinates (i.e. latitudes and longitudes) of spatial locations are transformed into one-dimensional spatial representation using the Google s2 [273] library. The travel routes were generated using the shortest distance path between origin and destination points. This was implemented using the Open Source Routing Machine (OSRM) api available at http://project-osrm.org.

5.7 Evaluation metrics and results

We evaluate the fitness of the synthesized population using similarity benchmarks on the statistical and spatial distribution. As a base case for comparison we also synthesized a population using VAE

with the same input data.

5.7.1 Similarity in statistical distribution

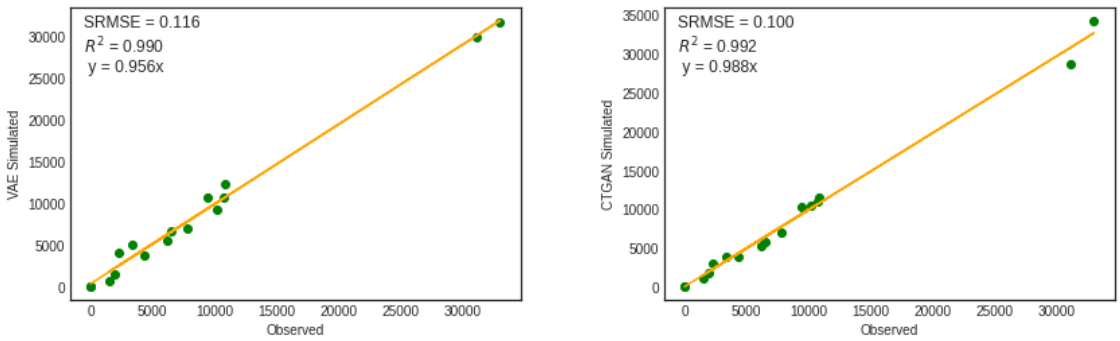
The purpose of this benchmark is to evaluate the statistical similarity between the true and synthetic representations of the data. An efficient approach to guarantee the utility of synthetic reconstruction is to compare its statistical properties to the true distribution whose results should be identical or near-identical. We assume that the synthetic data is fit for microsimulation estimations when aggregate queries on both true and synthetic distributions are equivalent. We evaluate the similarity of statistical properties using three (3) metrics. First, we observe the full joint distribution of all possible combinations of data variables. While this approach is efficient for low dimensional tabular data as used in this paper, an implementation to high dimensional data could be complicated. Partial and conditional joint distributions should be used in such cases. Secondly, we derive and compare the marginal distributions for all domains in data variables for the true and synthetic representations. Using this benchmark, the success of the synthesized output is measured by the high score in similarity of the probabilities of values of variables in both datasets without reference to the values of other variables. Finally, we quantify the empirical distributions between the synthetic and true distributions with the Standard Root Mean Square Error (SRMSE) [274], the accuracy and fitness of the synthetic reconstruction using a measure of the Pearson correlation coefficient(*corr*) and the coefficient of determination(R^2). The standardized root mean squared error is defined by:

$$SRMSE(\hat{\pi}, \pi) = \frac{RMSE(\hat{\pi}, \pi)}{\bar{\pi}} = \frac{\sqrt{\sum_i \cdots \sum_j (\hat{\pi}_{i \cdots j} - \pi_{i \cdots j})^2 / N_b}}{\sum_i \cdots \sum_j \pi_{i \ldots j} / N_b} \tag{3}$$

where N_b is the total number of agents; $R_{i..j}$ is the number of agents with attribute values i...j in the synthesized population, $\hat{\pi}$ and π is the synthetic and true distribution respectively.

5.7.2 Similarity in spatial distribution

To evaluate the utility of the synthetic reconstruction on sequential location data, we evaluate with metrics: *trip length, segment usage* and *origin-destination distribution*. Trip length distribution measures the similarity in distances traversed on trip segments, segment usage distribution measures



(a) VAE (b) CTGAN

Figure 5.4: Fit between true and synthesized population.

the frequency of trips on a routes and the origin-destination measures the agent count on each zone for trip origin and destinations. These metrics quantify the accuracy and fitness of spatial characteristics in the synthesis model.

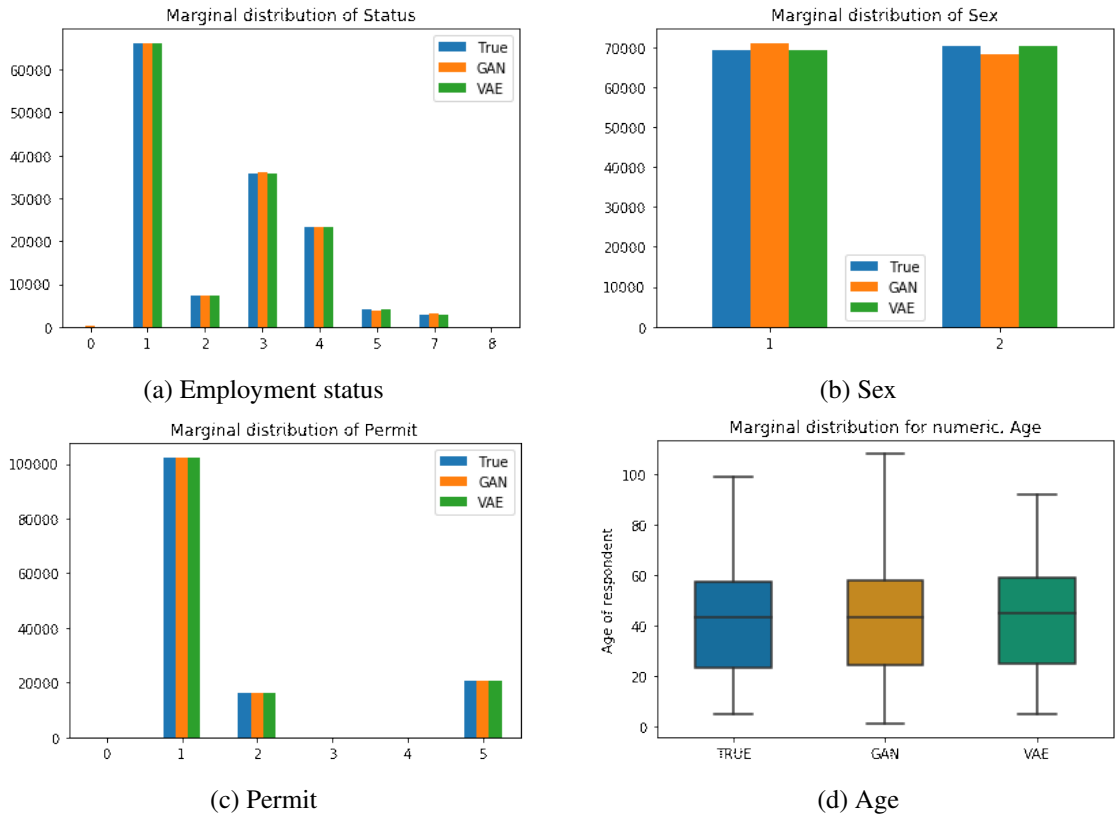
5.8 Experiments and evaluation results

In this section, we discuss the experiment setup and the results achieved on the model implementation using the metrics stated. The model was built and implemented with Python Keras with Tensorflow backend support on a MacBook Intel Core i5-4258U and GPU Intel Iris Graphics 5100.

5.8.1 Statistical distribution comparison

In this experiment, we focus on comparisons of population-synthesis-based approaches on tabular data between CTGAN and VAE. The experiments were designed such that both models were provided with the same amount of data and dimensions about the sample population. The output of each model is subsequently analyzed to evaluate how good the full joint and marginals of the true population are reproduced. To assess the goodness of fit, the Standardized Root Mean Square Error is performed on the output of each model.

For comparative analysis on the full joint distribution, we consider a combination of all attributes in the sample data for Age Group (the age variable is discretized into groups of child, young, adult, old), Sex, Employment status and Permit. We construct a contingency table on all combinations



(a) Employment status

(b) Sex

(c) Permit

(d) Age

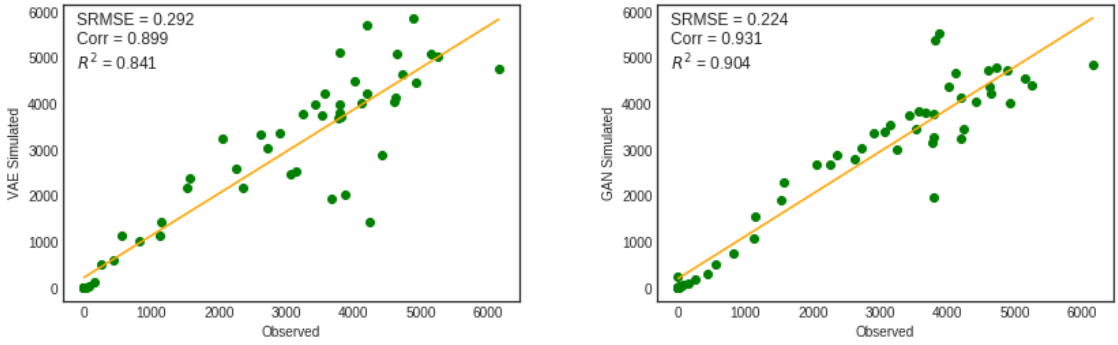
Figure 5.5: Comparison of marginals for attributes for True, CTGAN and VAE data.

of attributes using frequency counts. As observed in Figure 5.4, while both models give a good synthetic representation of the true data distribution, the simulated observations from CTGAN exhibit a better fit with a lower SRMSE of 0.010 while the VAE results in an SRMSE of 0.116. Also, CTGAN results in a strong correlation 0.996 compared to 0.988 for the VAE. The minimal loss in approximation of the VAE could be attributed to the low latent dimensional representation adopted by the VAE thus there is a loss of resolution in the synthetic reconstruction. Similarly as can be seen in Figure 5.4, the VAE shows a slight dispersion along the line of fit that could be attributed to the same low representation.

The marginal distributions of the tabular variables are shown in Figure 5.5, and depict the similarity of representation for both the VAE and CTGAN approaches to the True distribution. Obviously, the synthetic population perfectly reproduces the marginals of the training data. The representation from the VAE marginal distribution gives a better similarity to the true distribution than the CTGAN

though the model does not memorize the input data. This could be a cost of vanishing gradients suffered by the use of sigmoid activation functions [275, 276] on the last output layer of the generator network for binary types, as seen by the slight imbalance in the marginals of sex variable.

We extend the experiment to compare the fitting and correlation patterns in the marginal distributions of the numeric variable, age. As shown in Figure 5.6, CTGAN exhibits a better fit with a lower SRMSE of 0.224 compared to SRMSE of 0.292 of the VAE. At an R^2 of 90%, the CTGAN model explains the true distribution with minimum variation relative to the 84% of the VAE. Finally, it is evident that the simulated agents of the VAE show spread along the best line of fit while agents remain clustered along the line of fit for the CTGAN. In this sense, the CTGAN model presents a reliable agent representation that has a better fit to the true distribution and clearly outperforms the VAE.

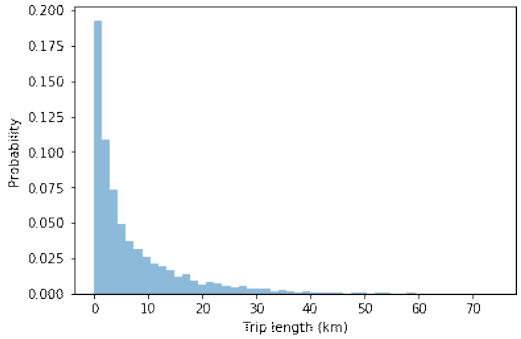


(a) VAE (b) CTGAN

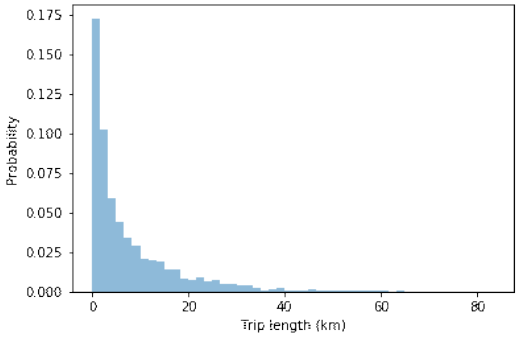
Figure 5.6: Fitting and correlational analysis for marginal distribution on numeric variable, Age.

5.8.2 Spatial distribution comparison

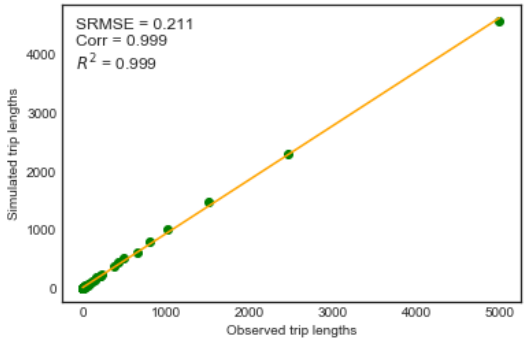
In order to ensure the consistency in the spatio-temporal behaviour of synthetic agents is retained after synthetic reconstruction of the trip sequences, we evaluate the similarity in trip length distributions and the spatial distributions of error in route segment usage.



(a) Trip length distribution of True data



(b) Trip length distribution of Synthetic data



(c) Line of fit for trip length counts

Figure 5.7: Histogram of trip length distributions for true (a) and synthetic (b), and best line fitting for true and synthetic trip lengths.

Trip length distribution

Trip lengths are defined by the movement of an agent from one location (origin) to another geographic location (destination). The length of trips is estimated using the euclidean distances between two points. Typically, an agent embarks on a sequence of trips (i.e. trip segments) based on the purpose at the time of the day until a complete trip ends at the start origin. We consider the lengths of all trip segments and compare the frequency distribution of travel distances between the true and synthetic sequential representations.

In Figure 5.7a and 5.7b, the CTGAN simulated trip lengths show a near equivalence in distribution to the real sequences. It is observed that there is a high count of short trips within distances of two (2) kilometers for both distributions, though a slight imbalance of 19% of trip length is estimated for real trips as compared to 17% for synthesized trips. There is a steep decline of trips whose distances are beyond 5 kilometers in both real and synthetic representations. These statistical estimations are

expected because travels within urban communities like in the case of our study region are relatively shorter than rural areas. The synthetic sequences present a near perfect fitting on trip lengths to the real sequences as shown in Figure 5.7c having an SRMSE of 0.211 and a correlation coefficient of 0.99 and an adjusted R^2 of 99%.

Route segment usage distribution

The purpose of this metric is to evaluate the similarity in the frequency of trip routes taken by agents. While the model outputs sequences of trip destinations, we assume the shortest possible distance using the Dijkstra Algorithm [277] to derive the route itinerary from Montreal road network [278]. We compare the frequency of trip counts travelled on each route for both true and synthesized data. The efficiency of the synthesized trip sequences is evaluated by the similarity or equivalence in route usage counts observed on both true and synthetic trips.

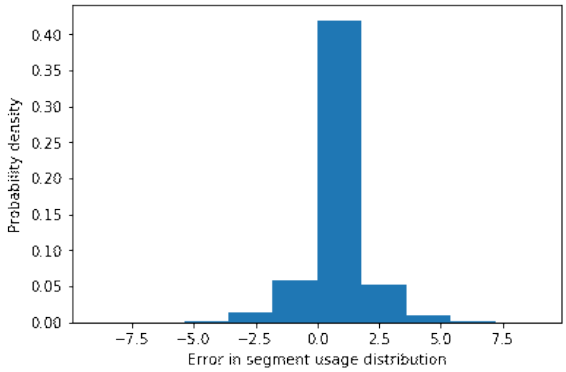


Figure 5.8: Distribution of differences in route segment usage for true and synthetic trips.

In Figure 5.8, a high proportion of routes show equivalent similarity on route usage for both true and synthesized trips. The model shows remarkable success in generating similar route usage frequencies at a probability density above 40% recording the difference in usage counts between the true and synthesized. Route usage probabilities less than 5% of the total routes exhibit variances in frequency within range of 1 to 5 counts symmetrically. We illustrate the error distribution of route usage for the Greater Montreal Area shown in Figure 5.9. A majority of the routes give a perfect fit of synthetic reconstruction marked by differences close to zero, colored in magenta on the route map.

Figure 5.9: Route usage distribution of error in the simulated sequential trips of Greater Montreal Area

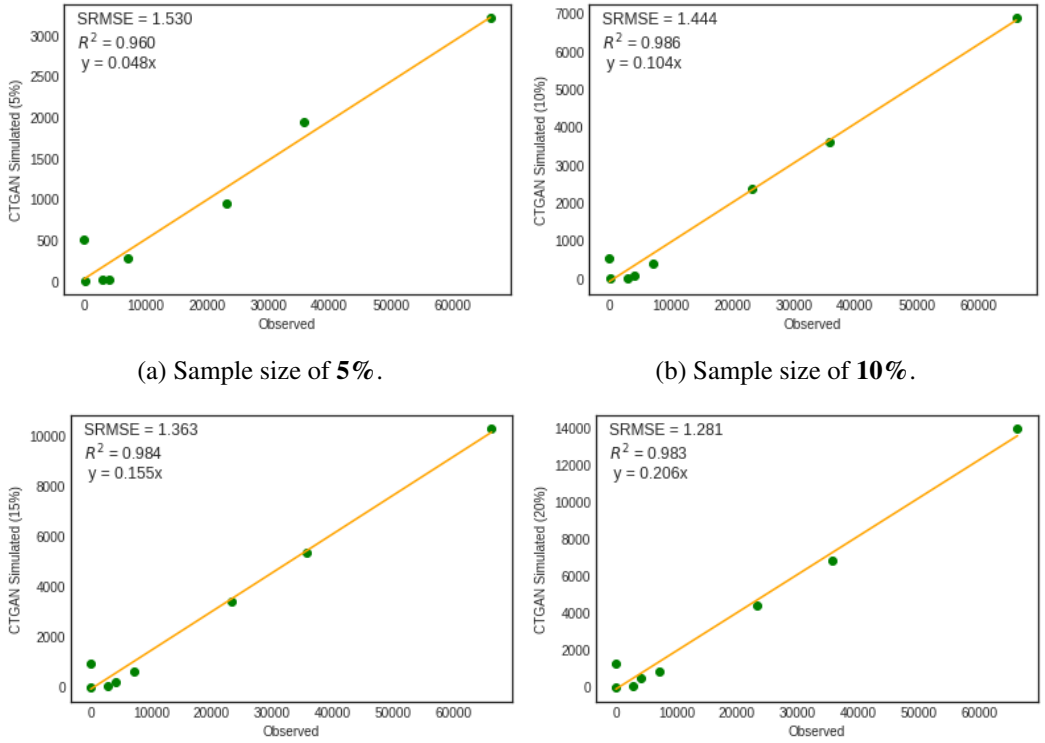
5.8.3 Sensitivity Analysis

The aim of this analysis is to critically and systematically evaluate the performance, accuracy and elasticity of CTGAN for varying sample and categorical sizes when synthesizing individual level attributes or populations. The outputs are assessed using the Standard Root Mean Square Error, calculated by comparing the sample to the simulated population and the coefficient of determination, denoted by R^2.

Varying input sample size

In this approach, random samples are selected from the original sample with sizes of 5, 10, 15 and 20%. The varying selected samples were independently trained as inputs to CTGAN. Scatter plots are shown in Figure 5.10 to depict the relationships between the observed and simulated for dimensions using different sampling sizes.

With a sample size of 5%, we observe a spread along the line of fit with an SRMSE of 1.530. Subsequently an improvement is observed as the sample size is increased to 10% with declining SRMSE of 1.444. It is observed that the fit improves while minimizing spread when sample sizes are increased. This suggests the model performs better with an increase in sample size and smooths towards the distribution of the sample population with incremental sample ranges. Table 5.2 gives a summary of the performance for all simulated dimensions. As expected, a decline in the mean squared error for all synthesized dimensions is observed as sample sizes are increased.



(a) Sample size of **5%**.

(b) Sample size of **10%**.

(c) Sample size of **15%**.

(d) Sample size of **20%**.

Figure 5.10: Uni-dimensional distribution of varying sampling sizes between observed and simulated observations.

Sample	Status	Gender	Permits	Age
5	1.530	0.950	1.762	1.202
10	1.444	0.900	1.666	1.123
15	1.363	0.850	1.575	1.052
20	1.281	0.800	1.465	0.994

Table 5.2: Standardized Root Mean Square Error (SRMSE) on varying samples of synthetic generation on varying sizes

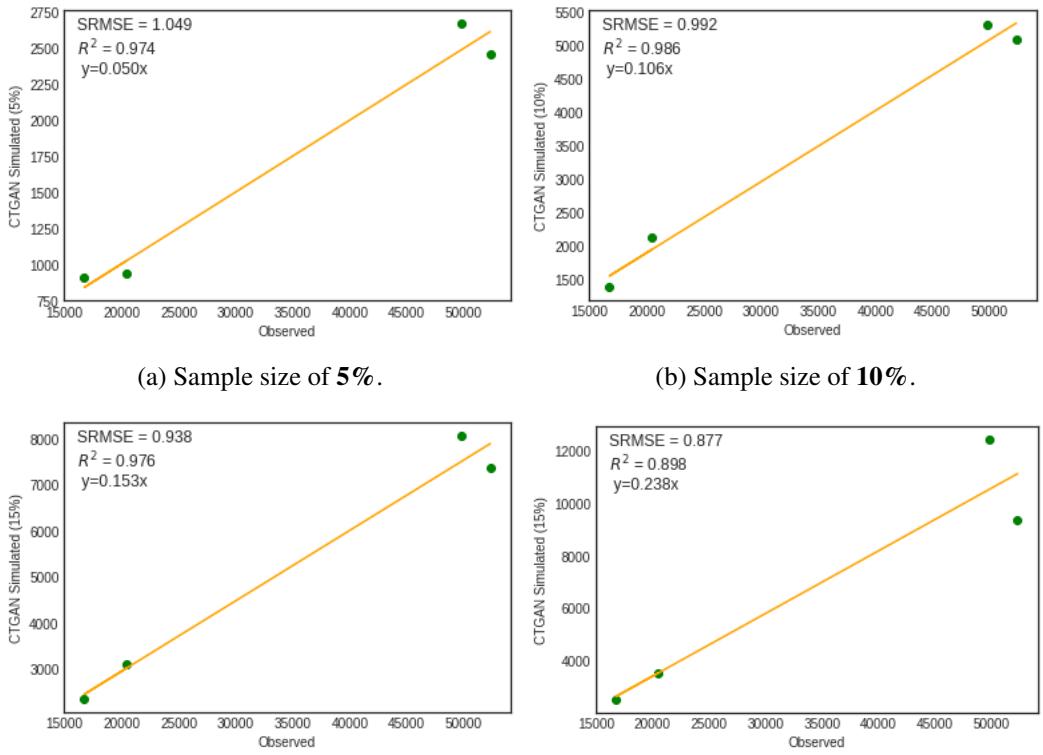
.

Inter-attribute relationships

This analysis considers how well the synthetic model recreates the observed relationships between attributes in the original sample population for varying sample sizes (i.e. 5, 10, 15, 20%). The results in Figure 5.11 show the performance of the conditional probabilities for Permit by Gender attributes and Age Group by Gender attributes. The line of fit exhibits a balance population between counts of the conditionals. As observed, the increase in sample sizes reduces the mean square errors from 1.049 for a 5% sample size to SRMSE of 0.992 for a 10% sample size, these steadily decline in SRMSE values for increasing sample sizes. This suggests the model improves on learning a fit of the conditional distributions between attributes and subsequently smooths the distribution of the increasing sample sizes toward the distribution of the sample population. Similarly, we evaluated the full joint distribution for all variables between the sample population and synthesized population. The output observations were re-sampled and evaluated. Using a 5% sample size as shown in Figure 5.12, there is a wider distribution spread between observed and synthetic of SRMSE at 1.457, while the line of fit shows a spread of points along it. This suggests an imbalance in the population summaries between observed and synthesized observations with a weaker distribution fit compared to the sample population depicted by the spread. It can be seen from the analysis that the model shows consistency in learning the inter-attributes relationships and full joint distributions between all attributes when the sample sizes are increased.

Varying categorical sizes

In the final experiment, we evaluate the performance of CTGAN for varying categorical sizes. For this purpose, the attribute "Age" is converted from numerical to categorical input and subsequently discretized into bin sizes of 5, 10, 15 and 20 categories of age groups. The model is retrained with the discretized categories and the output is represented in Figure 5.13. At a category size of 20, we observe a weaker correlation along the line of fit suggesting an imbalance between population counts of observed and simulated observations having a high SRMSE of 0.716. The output of trained samples on category size of 10 shows a better improvement of fit with a wide spread along its perfect line of fit. We observe a sequential improvement with a reduction to size of categories



(a) Sample size of **5%**.

(b) Sample size of **10%**.

(c) Sample size of **15%**.

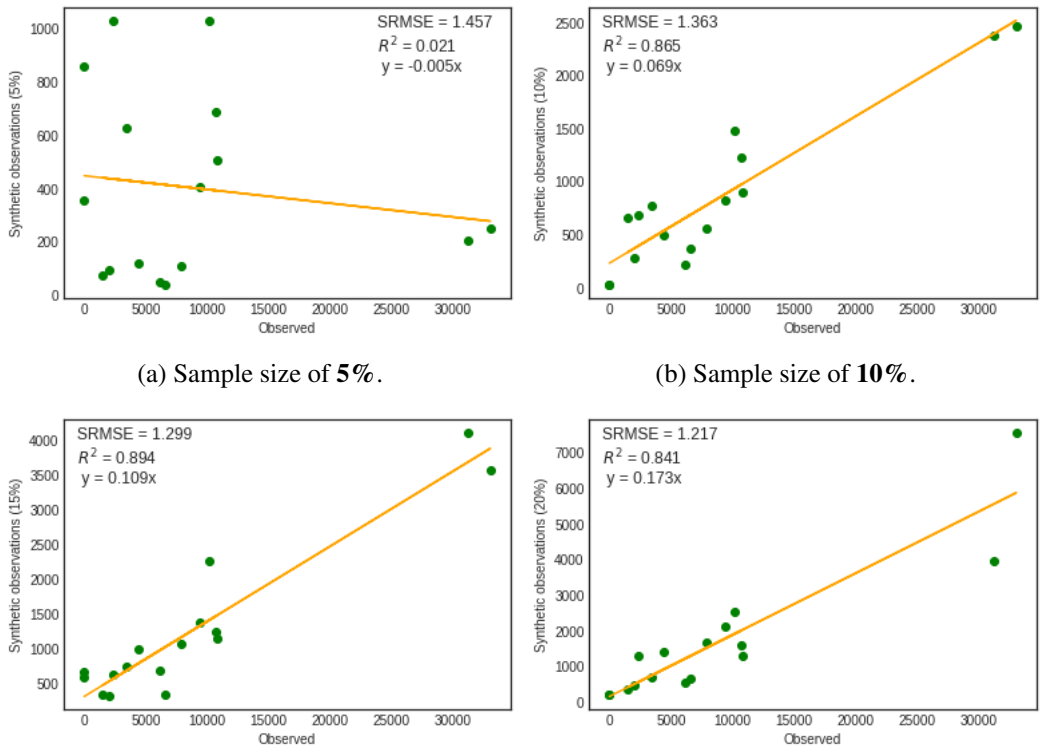
(d) Sample size of **20%**.

Figure 5.11: Conditional distributions for permit by gender between observed and simulated counts.

for 7 and 5 categories. This suggest the model is able to smoothen the distributions of minimal categories or modes. This could have arisen because of the lack of diversity/mode dropping and non-convergence that is notable limitation in GANs [279, 280].

5.9 Discussions and conclusions

A novel deep learning generative model for reconstructing synthetic agents having tabular and sequential location-based travel information is presented. Specifically, we combine two generators and two discriminators to design the Composite Travel GAN (CTGAN) architecture that outputs both tabular and sequential attributes simultaneously. The work compared the statistical similarities of the synthetic tabular results of the CTGAN with synthetic results from the VAE. The models were tested with sample population data from the origin-destination survey of the region of Greater Montreal (Canada) in 2013. The CTGAN outperformed the VAE in terms of synthetic generation



(a) Sample size of **5%**.

(b) Sample size of **10%**.

(c) Sample size of **15%**.

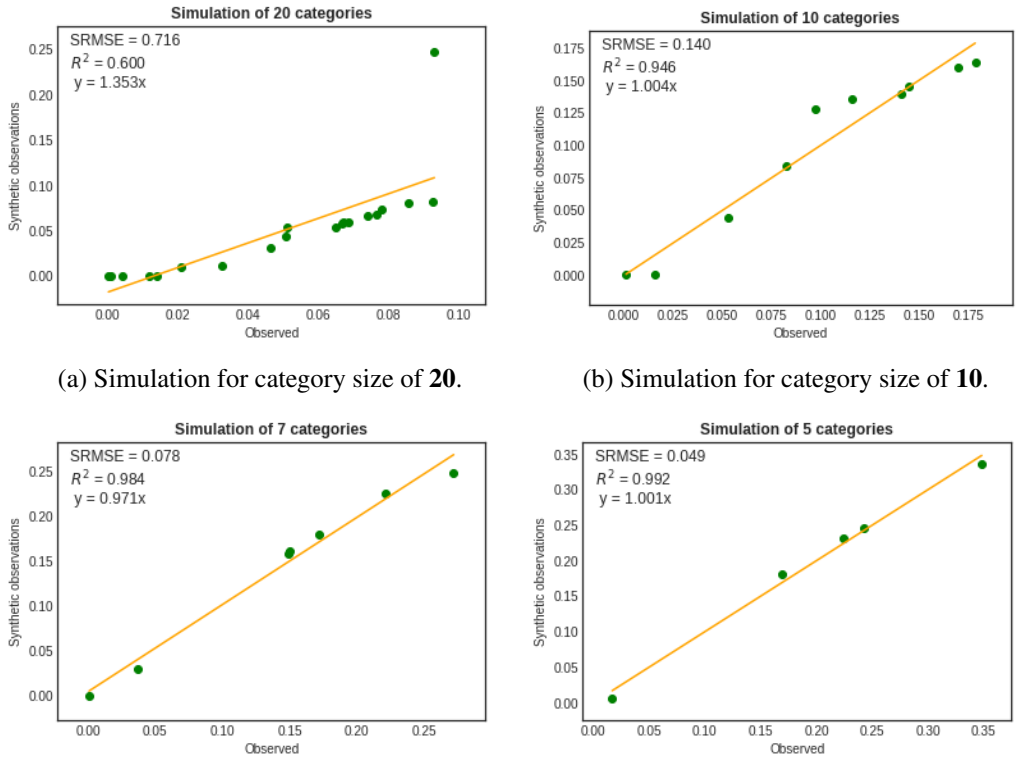
(d) Sample size of **20%**.

Figure 5.12: Full joint distributions for all variables between observed and simulated counts.

of tabular data.

Our results show the capability and success of CTGAN to recreate the marginals of attributes for both the tabular and sequential samples while maintaining inter-attribute relationships. We observed improvement of the performance of the model through scaling of different sample sizes with a better output for the large sample sizes that smoothens the learning distribution to the underlying distribution of the sample population. Sampling variation has a significant impact on the representation of the attributes and inter-attributes relations as evident in the analysis of the varying sizes. Based on this trend, it can be concluded that the model will perform better when a larger sample population is provided.

When implementing CTGAN, we observed the following drawbacks. There was significantly longer training time to synthesizing both tabular and sequential data simultaneously. 12 hours were required to train and synthesize 100,000 simulated household samples. Also, CTGAN showed difficulty in training sequences of more than 5000 complete trips hence samples had to be batched



(a) Simulation for category size of **20**.

(b) Simulation for category size of **10**.

(c) Simulation for category size of **7**.

(d) Simulation for category size of **5**.

Figure 5.13: Distribution of varying categorical sizes (age discretized).

for training. These drawbacks limit the adaptation of the model on real travel datasets which could have millions of travel records. In this regard, future work will consider deploying the model in a distributed computing framework and parallelized training on multiple nodes to improve on the training time and increase capacity for optimal model training. We also seek to consider improving the generative framework with losses to control the level of privacy that can be achieved. We will be able to control the expected privacy, especially in cases of releasing data to non-trusted data agents. While this paper is one of the first studies using generative models on travel data, we plan to explore methods that will be needed to improve the utility and privacy of the models when publicly releasing the synthetic datasets. We will work to extend this research on the generation of synthesized continuous mobility trajectories. We will explore the use of federated learning and Blockchain for Smart Mobility Data-markets (BSMD) framework proposed by Lopez et al. [281] to estimate CTGAN without directly accessing the sample, which may result in compromising the privacy of the individuals in the sample.

5.10 Acknowledgements

This research has been funded by the Social Sciences and Humanities Research Council of Canada (SSHRC) and Canada Research Chairs program.

5.11 Author Contribution Statement

The authors confirm contribution to the paper as follows:

Study conception and design: Godwin Badu-Marfo, Bilal Farooq, Zachary Patterson
Development of privacy-preserved generative model and testing: Godwin Badu-Marfo;
Analysis and interpretation of results: Godwin Badu-Marfo, Bilal Farooq, Zachary Patterson;
Draft manuscript preparation: Godwin Badu-Marfo, Bilal Farooq, Zachary Patterson;

All authors reviewed the results and approved the final version of the manuscript.

Chapter 6

Privacy versus Accuracy in Activity Diary Synthesis: A Differentially Private Multi-Output Deep Generative Networks Approach

6.1 Preamble

In this chapter, we develop a differentially-private GANs architecture which uses a multi-task approach through a shared layer in a single Generator-Discriminator, and outputs tabular and sequential (location) features simultaneously. Unlike the model in Chapter 5 using a multiple generators-discriminators network, this improvement uses a single generator-discriminator that is computationally efficient to train. The model implements a "privacy-by-design" approach to ensure privacy guarantee for training points.

This paper was accepted for presentation in Transportation Research Board Conference, but will be submitted for review at a journal.

6.2 Abstract

Population synthesis approaches in transportation planning applications are used to generate synthetic populations from sample information to be used for transportation demand generation. Recent population synthesis approaches are capable of reproducing accurate synthetic representations, but with the risk of revealing compromising personal information on members of the samples on which they are based, potentially causing privacy violations. The importance of privacy has motivated research seeking to develop newer population synthesis approaches designed explicitly to accurately reproduce populations yet guaranteeing privacy protection. This work extends literature on population synthesis by contributing novel deep learning approaches to the development and application of synthetic travel data while guaranteeing privacy protection for members of the samples on which the synthetic populations are based. First, we show a complete de-generalization of activity diaries to simulate the socioeconomic features and longitudinal sequences of geographically and temporally explicit activities. Second, we introduce a differential privacy approach to control the level of resolution disclosing the uniqueness of survey participants. Finally, we experiment using Generative Adversarial Networks that promise scalability in handling high dimensional variables for this synthesis work. We analyse the statistical distributions, pairwise correlations and measure the level of privacy guaranteed on simulated datasets for varying noise. The results of the generative model show privacy-protected travel populations having tabular and sequential attributes that are generated over varying privacy noise levels produced from the generative adversarial network framework. While the proposed model can generate trip information, we do not concentrate the analysis of trip information in this paper because newer benchmarks are required to be defined for measuring the efficiency and performance of simulated geographic and temporal representations which will be considered in future research.

6.3 Introduction

Activity-based travel demand models have become commonplace in the academic transportation modeling literature and increasingly in transportation decision-making in recent times. Using these models, transportation planners and stakeholders study the behaviour of agents (i.e. households and individuals) that influences their choices of daily activity participation and travel. Activity-based models require spatially and temporally granular representations of a person's trip activities including where and when activities take place, and how they get to the activity (travel mode) locations. These microsimulation models necessarily rely heavily on disaggregate individual-level information (i.e. microdata). In practice, it is difficult to obtain disaggregate travel information of an agents because of the high cost of data collection for large populations and the potentially privacy compromising effects of inadvertent releases or intentional publishing of such data.

As a solution to address the lack of accessibility to and completeness of microdata, population synthesis is used to reproduce synthesized representations of true populations based on samples of disaggregate that are characterized by the same joint distribution of variables of the real population [22, 239]. This technique is appealing for generating data surrogates with properties that conform to the underlying distribution of the population but which only require data from a small sample from the population of interest. Popular methods of population synthesis including re-weighting and matrix fitting do not produce agent-based samples, but rather a sample of prototypical weighted individuals, and hence a post-simulation is required in cases where population synthesis is linked to agent-based samples where individuals are drawn from the weighted samples [239, 282]. An increasingly popular approach to population synthesis, simulation, solves some of the drawbacks of the re-weighting and matrix fitting models. Simulation-based methods have proved effective for high-dimensional synthetic generation and provide a systematic way for imputing or interpolating data [22, 283]. Farooq et al. 2013 [22] used this approach to generate a synthetic micro population for Brussels, Belgium where complete data for the population was not available. All of these methods of population synthesis suffer the drawback of scalability due to the "curse of dimensionality" and computational complexity [22, 23].

These traditional population synthesis approaches have shown success in recreating weighted samples from aggregate census data, but have not been used to generate representations of complete travel diaries due to issues around computational complexity and scalability. Outside of transportation, Generative Adversarial Networks (GANs) have proved capable of estimating complex joint distributions, hitherto intractable for large training sets and complex data types like video, images, sound, etc. In the GANs framework, a generative model is set against an adversary, or a "discriminative" model that learns to distinguish fake observations produced by the generative model from real data observations. In the transportation literature, Borysov et al. [23] demonstrated the use of GANs for simulating the socio-demographic characteristics of synthetic agents for a travel model. The capability of GANs to reproduce faithful representations of a population, however, could lead to information leakage [284, 285] on training data points that threatens privacy on respondents. This potential for compromising the privacy of synthesis seed sample members raises research interest in developing newer synthesis approaches that exhibit privacy-in-design capabilities. To this end, recent work [26, 286, 287] has focused on developing deep learning approaches that protect sensitive information by training in a differentially private manner such that the privacy of synthesis seed sample members is not compromised. Abadi et al. [26] have demonstrated the possibility of training a model in a differentially private way that relies on the Differentially Private Stochastic Gradient Descent (DP-SGD). This privacy-sensitive training can control the the confidence with which an adversary could learn or infer information about an individual from a sample, and can indeed control this through a parameter, epsilon (ϵ) that defines the level of privacy guaranteed.

In this paper, we leverage the potential of previous work to solve three problems: First, we want to synthesize a complete activity diary (based on socioeconomic attributes and a snapshot of longitudinal activity sequences of a sample) to synthesize travel diaries for a synthetic population. Second, we explore training the generative model in a differentially private manner as a step to protect sensitive information of individuals in the underlying training data. Finally, we want to build and deploy a novel generative mechanism that adopts state-of-art deep learning techniques like Generative Adversarial Networks.

As such, the key contributions of our work include:

(1) We present a novel generative model that is capable of estimating the joint distribution of socio-economic variables of travel agents and simultaneously learn the agent activity sequences from an Origin-Destination (OD) survey while incorporating parameters to guarantee privacy of information leakage about a person who participated in a survey.

(2) We experiment a differential private training of the generative model with varying degrees of noise to evaluate the effect on the statistical distribution of the synthesized representations.

(3) To the best of the authors' knowledge, this is the first work using Generative Adversarial Networks to synthesize a complete activity diary of agents having multiple outputs of socio-economic characteristics and a complete activity chain in a single model.

It should be noted that while the proposed generative model can generate trip information, we do not concentrate the analysis of trip information in this paper because newer benchmarks are required to be defined for measuring the efficiency and performance of simulated geographic and temporal representations which will be considered in future research work. The paper is organized as follows: the next section presents review of the relevant literature, followed by a section that describes the framework architecture of the generative model. A methodology section describes the data processing steps and we then define the evaluation metrics before presenting an analysis of results. We finish the paper by explaining our conclusions and future directions for the research.

6.4 Literature Review

Population synthesis approaches have been dominant in trip-based modelling over the years to estimate synthetic members of a population in cases where data on travel agents (i.e. individuals and households) are not available. Using data inputs of census aggregates and a sample of microdata on agents in a study region, new members of the population can be simulated to possess similar travel characteristics of the true population. Synthesis approaches are broadly classified into three categories; re-weighting, matrix fitting and, simulation based approaches [234]. The re-weighting methods adjust weight factors of surveys to create samples that represent subregions rather than the entire summation of the population aggregates, in effect, applies non-linear optimization to estimate weights [235, 246, 247]. The methods of matrix fitting evoke expansion factors expressed by the ratio between a starting solution and the final matrix. The Iterative Proportion Fitting (IPF) proposed by Deming and Stephan [248] and the Maximum Cross-Entropy [249] are known implementations of the matrix fitting method and referred as *deterministic models*. These deterministic models do not produce agent-based samples but rather a sample of prototypically weighted agents [245]. Duguay et al. [57] first introduced IPF method to synthesize households survey data in transportation literature. Similarly, Beckman et al. [51] created synthetic population for TRANSIMS [288] using census cross tabulations and sample. Adopting fitting methods for large dimensional data becomes computationally and memory-wise expensive. Simulation-based methods solve some of the drawbacks of the deterministic models and is capable of estimating the joint distribution of population data with full set of attributes from which new members can be recreated through sampling. Sun and Erath [250] proposed the Bayesian Network, a popular implementation of the simulation-based approach. Similarly, Sun et al.[289] shown the Bayesian network is an effective approach for traffic flow modelling and forecasting while performing experiments on urban vehicular traffic flow data of Beijing. However learning of the graph structure of the bayesian network for large datasets can be computationally expensive [23].

Deep generative models have evolved lately in reproducing realistic and near-true synthetic representations that perform effectively in dealing with complex computation of synthesizing agents. Most popular variants of deep generative models are the Generative Adversarial Networks (GANs)[24]

and Variational Autoencoder (VAE) [65]. Similar to the simulation-based approaches, these models are capable of estimating the joint probability distribution of the underlying data and newer members of the population can be sampled from the joint distribution. Choi et al. [254] proposed a generative model that combines auto-encoders with GANs to synthesize private electronic health records in generating binary and count variables in health datasets. Park et al. [255] proposed a *table-GAN* to synthesize tabular data using a hinge-loss privacy control mechanism. In their approach, they showed a compatible model for anonymization where sensitive attributes are maintained without change. Neural sequence generation has been well studied since the advent of Recurrent Neural Networks (RNN) [290] and Long Short Term Memory (LSTM) [243]. RNNs have shown incredible results in capturing long-term dependencies but as Bengio [257] discussed, fitting the distribution of observed data does not mean generating satisfactory text because of the exposure bias [291]. Solutions proposed to address this limitations included the concept of reinforcement learning and GANs to generate acceptable sequences. SeqGAN [1] was proposed as a language model for the generation of sequence using the concept of reinforcement learning. In their approach, the generator used the stochastic policy where the state is defined as the tokens generated and an action being the next token to be generated. The presence of a stochastic policy, REINFORCE [258] algorithm, allows different actions to be sampled during training and derive a robust estimate of the policy. Both the generator and discriminator are pretrained on real and fake data prior training with policy gradients. During training they implement Monte Carlo rollouts in order to get a useful loss signal per word. Subsequent work demonstrated text generation without pretraining with RNNs [292].

While GANs and other generative models have been successful for reproducing identical copies of the true population, there is a risk of information leakage to an adversary who could infer if a person partook in the training data points. Such concerns have motivated recent research work into developing privacy-by-design techniques such as differentially private training in deep learning [26]. The authors studied a gradient clipping method that imposed privacy during training of the neural network. Shokri and Shmatikov [286] proposed a multi-party privacy preserving neural network with a parallelized and asynchronous training procedure. In the work of Phan et al.[293], the authors developed a private convolutional deep belief networks(CBDNs) by leveraging the functional mechanism to perturb the energy-based functions of conventional CBDNs.

In the next sections, we provide a brief definition of relevant topics including: deep generative modelling and differential privacy.

6.4.1 Deep Generative Modelling

Deep generative models have evolved out of artificial neural networks [294] where they have been used successfully to reproduce realistic images and translations, while exhibiting outstanding performance and computational effectiveness. Notable deep generative models are the Variational AutoEncoder (VAE) [65] and Generative Adversarial Networks (GANs) [24]. Both generative models have shown promising results in estimating the joint distribution of underlying data, a property that is important for simulation-based population synthesis techniques like Bayesian Networks.

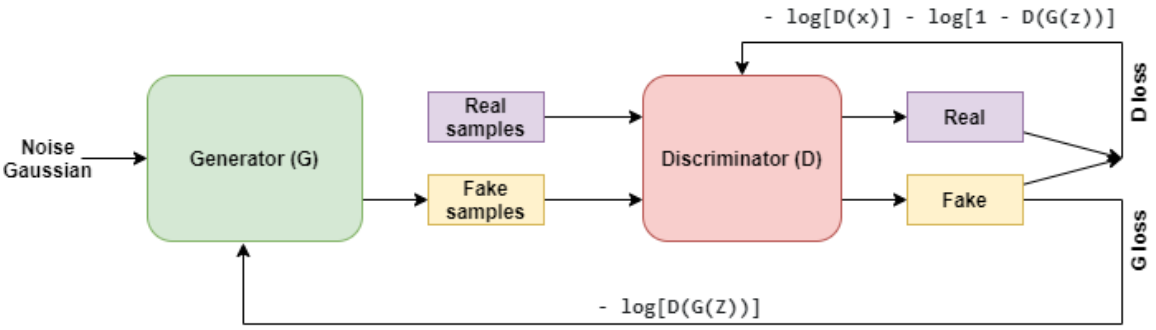


Figure 6.1: The architecture of GANs

Ian Goodfellow [24] proposed the Generative Adversarial Network that simulates a two player game composed of "Generator" and "Discriminator" networks. The generator network learns to generate samples from latent space that corresponds to the real samples. The Discriminator network is programmed to distinguish between synthesized and real sample data, with updated weights being sent back to the generator. Models for both networks are implemented as multi layer perceptrons [295]. During model training, the Discriminator gets better at distinguishing real samples from fake generated samples, while the generator improves on generating samples that are close to the real samples until a Nash equilibrium is achieved where each model reaches its peak ability to thwart the other's goal. The objective function of GANs is defined by:

Definition 1 (Objective function):

The *objective function* of the Generative Adversarial Networks [24] is:

$$\min_{G} \max_{D} V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[log\ D(x)] + \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z)))] \tag{4}$$

where *V (G, D)* is the value function of a two-player minimax game, *D(x)* represents the probability that x came from the true data, p_g is the generators distribution over the real data *x*, p_z(z) is a prior on noise variable (z) that maps to data space as G(z), where G is a differentiable function represented by a multilayer perceptron.

Equation 4 derives the objective function that suggests that if the input of the Discriminator is sampled from the true distribution then maximize the output of D(x) to 1 whereas if the input is generated from the Generator then D(G(z)) should minimize the output of the objective function. In this regard, the network seeks to maximize the parameters of the Discriminator using Gradient Ascent while minimizing the parameters of the Generator using Gradient Descent. The training process halts when a Nash equilibrium is reached so that the Discriminator is unable to distinguish true or fake samples.

In the work of Choi et al. [254], the authors proposed a model that combines auto-encoders with GANs to synthesize private electronic health records. The results of their model simulated binary and count variables in the context of health datasets. Similar work by Park et al. [255] proposed a *table-GAN* to simulate tabular data using a hinge-loss privacy control mechanism that is suitable for anonymization of sensitive attributes. Borysov et al. [23] has shown a simulation of micro-agents from a large Danish activity diary to estimate the joint distribution of the underlying data using the VAE model. In our approach, the GANs architecture will be optimized for high performance throughput making it capable of learning all training data records in order to avoid the challenge of sampling zeros, referring to agents that are omitted from the training samples but exist in the real population.

6.4.2 Differential Privacy

The concept of differential privacy [296] assures nothing new can be learnt on the statistical output of a query mechanism given that a record of information on the individual is added or removed from

the statistics of a survey. In this sense, a privacy guarantee is provided in the query function, which is not the case for other anonymization techniques like k-Anonymity [42]. Differential privacy limits a constraint on the processing of data such that the output of executing a query mechanism on two adjacent databases are approximately similar. The functional mechanisms of achieving differential privacy include the approach of adding Laplacian noise [43], the exponential mechanism [297], and the functional perturbation approach [298]. According to Dwork et al. [43], a randomized algorithm *M* fulfills ϵ-differential privacy if, for any adjacent databases *d* and *d'* differing at most one element, and for any output O of *M* is formally defined by:

Definition 2 (ϵ-Differential Privacy):
The formal definition of *ϵ-Differential Privacy* is given by:

$$Pr(M(d) = O) \leqslant e^{\epsilon} Pr(A(d') = O) \tag{5}$$

The privacy budget defined by parameter epsilon(ϵ) defines the difference between adjacent databases d and d', differing by only one observation. A controlled random noise is sampled from a Laplace distribution that is added to the query output of the function mechanism to achieve differential privacy.

6.4.3 Deep learning with differential privacy

As a step towards implementing differentially private training, we adopt the approach by Abadi et al. [26] in our work. Here, the authors developed a technique to train deep learning models in a differentially private manner. In their approach, random noise is sampled from a Gaussian distribution and added to the gradients of parameters of the neural network. The addition of noise to the computed gradients limits the influence that any particular input data can have on the final model. The steps for differential privacy training are as follows:

- Sample a minibatch of training data (x, y) where x is the input and y is the label.
- Compute loss L(θ, x, y) defined as the difference between the model's prediction $\theta(x)$ and label y where θ represents the parameters of the model.

- Compute the gradient of the loss L(θ, x, y) with respect to the parameters θ.
- For each training example, clip gradients in the minibatch to an upper bound defined by the maximum euclidean norm.
- Add random noise sampled from a Gaussian distribution to the clipped gradients and update parameters.

The model training with differential privacy gives a sanitized model gradient in which the influence of input data is bounded thus achieving privacy. The bounded gradients are used to train the model while update the weights.

6.4.4 Membership Inference Attacks Against Generative Models

Membership inference attack (MIA) was proposed by Shokri [284], as a privacy attack to measure the robustness of machine learning algorithms against adversarial attacks. The attack evaluates the prediction score of a model when the input data point is sampled from the training set rather than the validation set. The MIA comes in two forms: *black-box* and *white-box* attacks. The black-box attack assumes the adversary can only make queries to the target model under attack but has no access to the internal parameters of the model [284]. Contrarily, the white-box attack assumes the adversary has the parameters of the trained model at disposal and can make queries to it. We adopt the white-box attack approach in this work because is simple to implement and efficient. In a GAN setting, the adversary only is given access to the discriminator of the trained GAN model and consider a setting where the model parameters are leaked following a data breach. The trained model determines if a record was part of the training set, consequently the attack analyzes the danger in identifying with high confidence if a sample was used in the training. The adversary is assumed to have knowledge of the proportion of the dataset that is used for training but no other subsequent information is known about the training set. The attack is implemented by obtaining the probability score when the discriminator of the trained GAN predicts on each sample of the dataset. In a non-private trained model, the output of the attack should score lower probabilities (i.e., close to 0) for validation sets and high probabilities (close to 1) for training sets. On the other hand, private trained models should not output scores that distinguish training sets from validation sets.

6.5 Methodology

In this section, we first introduce the problem definition to establish the goal of the research. We continue in the subsequent subsections to give a detailed description of the proposed architecture for synthesizing tabular socio-economic variables and longitudinal activity sequences of location.

6.5.1 Problem definition

In this work, we assume, *X* to be the training data containing sensitive travel information of individuals. The training data is comprised of structured socio-economic variables characterized by a set of basic attributes $X = (x_1, x_2, x_3, x_4, ...x_n)$ where n is the number of variables, and a longitudinal sequence of time-ordered trip activities including trip purpose, departure time, and geographic coordinates of origins and destinations.

A generative model, *M* is trained on the private data and new data, X', is sampled from the model. In practice, the true data distribution of the population, $p_{data}(X)$ is unknown hence it is estimated empirically on a sample population. For the purpose of data synthesis, we use GANs as a framework to estimate the $p_{data}(X)$ and subsequently draw samples from it. In order to maintain privacy protection for participants in a travel survey, the generative model will be required to prevent an adversary from recovering with a high degree of confidence that an individual participated in the training data of the generative model, or prevent the adversary from inferring sensitive information about an individual based on the output of the model. In this sense, the goal of the proposed differentially private generative model is to synthesize a complete activity diary with high utility while guaranteeing privacy protection on training data.

6.5.2 Differentially Private Composite Travel Generative Adversarial Network

The proposed Differentially Private Composite Travel Generation Adversarial Network (DP-CTGAN) is a novel generative model that is designed to accept input from multiple data types (i.e. tabular and sequences) and is capable of estimating the joint distribution of data inputs through a shared hidden layer, and subsequently generate new private samples from the generative model trained in a differentially private manner. The DP-CTGAN is composed of two neural networks; the Generator

network, G and discriminator network, D.

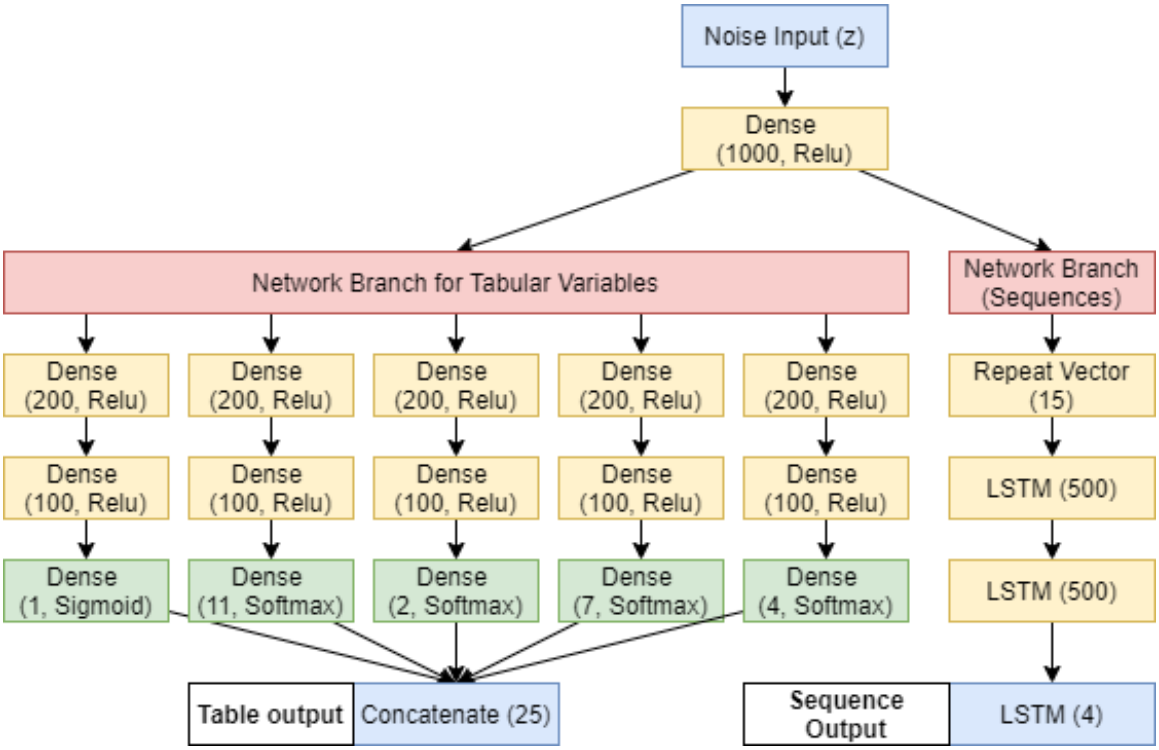


Figure 6.2: The Generator network of DP-CTGAN

The Generator, G, accepts as input a random noise that is sampled from a Gaussian distribution that is fed into two branches of neural hidden layers. The first branch of the generator (G_1), purposed for structured data learning is made up of a series of multi-layer perceptrons (MLP) for each training variable that connects neurons for each layer to the neurons of the next layer. Each hidden layer is activated by a Rectified Linear Unit (ReLU) [299] function which sets a lower bound of zero for negative inputs but returns same output for positive inputs. We apply a Sigmoid activation [300] to the output layer for numeric variables (i.e. Age), and a Softmax activation [301] to the last hidden layer for categorical variables.

The second branch of the generator, G_2 is designed for sequential data learning. The first layer of G_2 is made up of a Keras [302] Repeat Vector layer to repeat the incoming inputs in order to get hidden features for 20 future time-steps, the maximum length of each sequence. The output of the Repeat-Vector is fed to an LSTM layer with node size of 500 to extract features of previous time-steps. Two subsequent LSTM layers with node size of 500 are applied to the output of the

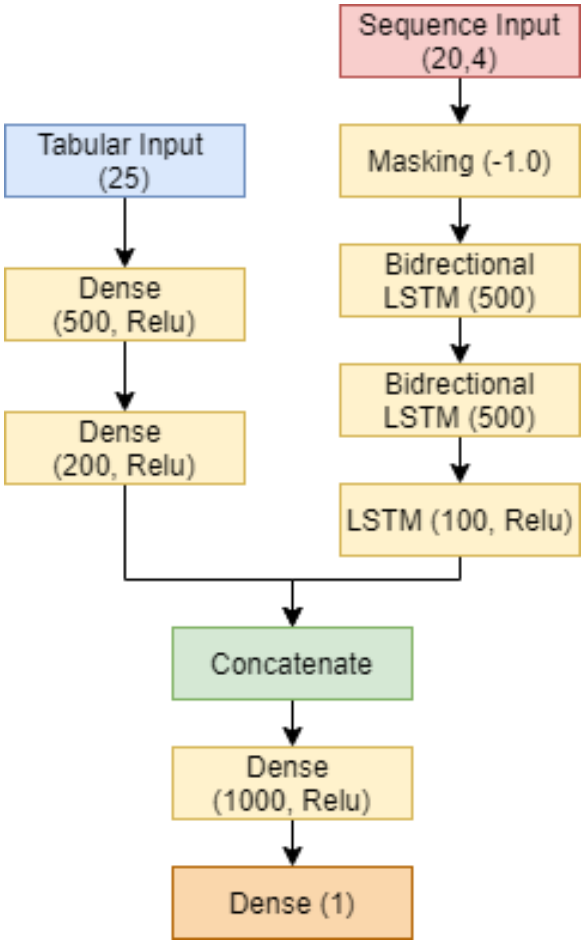


Figure 6.3: The Discriminator network of DP-CTGAN

time-steps such that the probabilities are well learnt for future sequence predictions. The last layer of the this branch is an LSTM with a size of 5, which is the size of the features for each time-step. The outputs of both branches G_1 and G_2 are merged into a shared output as the final output of the Generative model.

The discriminator D accepts inputs of the real tabular and sequential data as matrix of vector inputs. Similarly, D is also made up of two branches. The first branch of the network, D_T is an MLP made of two hidden layers with node sizes of 500 and 200 respectively. This branch accepts the input of the real tabular data and is purposed to learn the joint distribution of its input. The goal of the second branch, D_S is to estimate the distributions of sequences by learning its weights. It is made up of two Bidirectional LSTM models with node sizes of 500 to learn the probabilities of sequences in both directions. The outputs is fed into an LSTM with node of 100. The outputs of both branches

are merged into a shared layer. The shared output is consequently fed into a Dense layer with node size of 1.

In the design of GAN, the generator does not have access to the real data but accepts an input of Gaussian noise. This makes it intractable to implement differential private training in the Generator. Contrarily, the Discriminator accepts the real data as its inputs hence making it suitable for training in a differentially private manner. To achieve privacy, we use the differentially private Stochastic Gradient proposed by [26] to train the discriminator network as suggested by Xie et. al [303] and the RMSProp Optimizer to train the Generator network. First, we introduce a clipping parameter to act as an upper bound on the L2-norm of each gradient update observed through the training. We also introduce a noise multiplier to control the ratio between the clipping parameter and the standard deviation of noise that is applied to each gradient update of the discriminator after clipping. In this paper, we use a range of noise multipliers from 0 to 4 as shown in table 6.1. The differentially private discriminator propagates its parameters to train a standard generator whose computed weights also become differentially private. In effect, any new samples predicted out of the generator guarantees differential privacy.

Model	**Noise Multiplier**	**Description**
WGAN (0.0)	0	Model with no privacy noise
WGAN (1.0)	1	Model with noise multiplier of 1
WGAN (2.0)	2	Model with noise multiplier of 2
WGAN (3.0)	3	Model with noise multiplier of 3

Table 6.1: Noise multiples for differential private training

6.5.3 Case Study

In this work, we evaluated Dp-CTGAN on data from the 2013 Montréal Origin-Destination (OD) Survey [270]. The training data contained the activity diary of 10,000 individuals that were sampled out of the OD survey. The data included individual and household socio-economic variables such as gender, age, economic status, etc., and trip activity details such as activity location, time of departure, trip mode and purpose of travel. A list of the data available from the Montréal OD Survey is shown in Table 6.2.

Column	Type	Description
P_AGE	numeric	Age of the respondent
P_SEXE	binary	Gender of the respondent
P_MOBIL	categorical	Whether the respondent is mobile
P_STATUT	categorical	Occupation status of respondent
PERMIT	categorical	Driving permit type of respondent
M_DOMXCOOR	geospatial	X coordinate of residence
M_DOMYCOOR	geospatial	Y coordinate of residence
D_ORIXCOOR	geospatial	X coordinate of trip origins
D_ORIYCOOR	geospatial	Y coordinate of trip origins
D_DESXCOOR	geospatial	X coordinate of destinations
D_DESYCOOR	geospatial	Y coordinate of destinations
D_MOTIF	categorical	Trip purpose

Table 6.2: Description of variables to be synthesized from 2013 Montréal OD Survey

Mean	Standard deviation	Min	Max
43	20	5	95

Table 6.3: Summary statistics for numeric variable "Age"

6.5.4 Data Pre-Processing

The OD survey data is made up of tuples of households and individuals socio-economic variables as well as sequences of individual trips denoted by the coordinates of trip origins and destinations. The socio-economic variables have a fixed number of features for each individual comprising numerical (i.e.Age), binary (Sex) and categorical variables. On the other hand, trips of individuals have varying sequences having a minimum of three (3) locations and maximum of fifteen (15) location points. In this paper, we focus on generation of home based trips, typically made up a minimum of 3 locations (i.e., Origin-Destination-Origin). As a first step towards training in neural networks, all variables are converted into normalized numeric representations that is recommended for achieving efficient training with neural networks. Binary and categorical variables are first encoded to integer indices and one-hot encoded [271]. Similarly, numeric variables (i.e. age, geographic coordinates) are scaled and normalized within a range from negative one (-1) and positive (+1). The statistics of variable "Age" is shown in Table 6.3, reports a minimum of 5 years and maximum of 95 years for respondents that partook in the survey. While the objective of generative modelling is to recreate a synthetic copy of the true data, the encoding technique should be capable of being reversed or

decoded to the initial state. In this work, we used Scikit-Learn algorithms [304] label encoding and OneHot encoding which have reserve encoding capabilities.

6.5.5 Evaluation metrics and results

In this section we empirically evaluate the performance of the generated synthetic representations of the population and their travel characteristics. We vary different noise levels of privacy to assess the private performance of the synthesized travel data. The evaluation is done using the following benchmarks.

6.5.6 Similarity in statistical distribution

Using this benchmark, we compare the statistical properties of the generated output to the training set to verify that their distributions are similar. A generated output should be appropriate for microsimulation estimations if aggregate queries on distributions are identical to the true distribution. To achieve this, we first sample from marginal distribution of each variable $\pi(x_i)$ independently to verify that the marginals were perfectly reproduced. We also evaluate the conditional dependence of each attribute over other attributes, in effect deriving counts by category for each attribute. Finally, we measure the joint distributions on all possible combinations of data variables. This measure is applicable in low dimensional data but can be computationally intensive for high dimensional data. In such instances, marginal and conditional joint distributions are recommended. We evaluate the success of the synthetic approach by the similarity score in probabilities of the distributions. We quantify the empirical distributions between the synthetic and true distributions with the Standard Root Mean Square Error (SRMSE), the fitness of the synthetic reconstruction using a measure of the Pearson Correlation Coefficient(corr) and the coefficient of determination (R^2). The standardized root mean squared error is defined by:

$$SRMSE(\hat{\pi}, \pi) = \frac{RMSE(\hat{\pi}, \pi)}{\bar{\pi}} = \frac{\sqrt{\sum_i \cdots \sum_j (\hat{\pi}_{i \cdots j} - \pi_{i \cdots j})^2 / N_b}}{\sum_i \cdots \sum_j \pi_{i \cdots j} / N_b} \tag{6}$$

where N_b is the total number of agents; $R_{i..j}$ is the number of agents with attribute values i...j in the synthesized population, $\hat{\pi}$ and π is the synthetic and true distribution respectively.

6.5.7 Pattern Analysis

In this analysis, we adopt Principal component analysis (PCA) to measure the trends and patterns retained when synthesized data are reduced to fewer dimensions. The objective of PCA is to find the best summary of the data by reducing data using a geometric projection into a lower dimension. Using PCA, we can measure the variance of projected points and correlations between principal components. The output of the generative model should exhibit similarity in the variance and correlations between projected points. For each of the attributed sets in the synthesized data, numerical variables are normalized and categorical variables are converted to one-hot encoded representations. PCA is performed on all tuples of the generalized data.

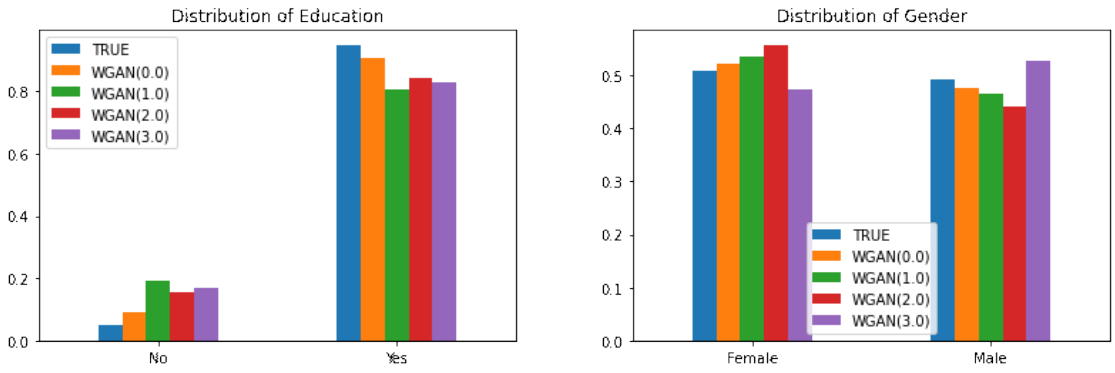
6.6 Evaluation results

In this section, we discuss the results achieved on performing the evaluation analysis. The generative model was developed and implemented with Python Keras with Tensorflow [305] backend support on a Windows 10 PC with Intel Core i7-2600 (8 Cores) and G-Force GTX 950.

6.6.1 Statistical distribution comparison

In this analysis, we compared the summary statistics on marginals, conditional and joint distributions for combinations of variables in the training and synthesized set. First, the marginals of the synthesized variables reproduced from the generative model produce the best approximation to the marginals of the true population. Figure 6.4 shows the marginals for 2 selected attributes from the true and synthesized population with varying privacy noise levels. It can be seen at WGAN(0.0) (no privacy noise added) that the simulated marginals sampler precisely reproduces the marginals of the training set though a low error is observed due to the sampling bias persistent in the random selection of samples during training of the generative model. With an incremental addition of noise, the reproduced marginals of the simulated sampler are less precise compared to marginals of the training set and exhibit randomness in the error of prediction, which is not deterministic nor does it follow a monotonic pattern. As an example in Figure 6.4(a), while the prediction of "Yes" values is under-predicted, it can be seen at noise level of 1.0, 2.0 and 3.0, the under-prediction increases, but at noise level of 4.0, the under-prediction decreases. This exhibits the level of randomness expected in the addition of noise such that an adversary cannot quantify if there will be a monotonic under-prediction or over-prediction. Marginals on the variable "Gender" shown in 6.4(b) show similar characteristics.

We study the goodness of fit by measuring the SRMSE of marginals observed between the true population and simulated populations at varying noise levels. In Figure 6.5, we observe a monotonic pattern that depicting an increase in the SRMSE of predictions as privacy noise levels increase. An SRMSE of 0.356, 0.362, 0.390, 0.455 is observed at noise levels of 0.0, 1.0, 2.0 and 3.0 respectively. These results affirm the elastic nature of noise addition, a promise of differential privacy to control the difference between the distribution of the true and simulated by privacy budget.



(a) Education status

(b) Gender

Figure 6.4: Comparison of marginals for attributes for True, WGAN and private WGAN representations

Similarly, we compare fitting of empirical distributions on possible combinations of bivariate distributions on possible pairs of variables. The distributions of the conditional probabilities are computed as frequency tables where each bin corresponds to a specific combination of values between two data variables. We measure the SRMSE between true and simulation distributions for bivariate combinations including "Permit vs Gender", "AgeGroup vs Gender", "AgeGroup vs Employed" and "Employed vs Gender".

NOISE LEVELS	**WGAN (0.0)**	**WGAN (1.0)**	**WGAN (2.0)**	**WGAN(3.0)**	σ_{mse}
p(Permit | Gender)	0.432	0.576	0.464	0.497	0.053
p(AgeGroup | Gender)	0.614	0.703	0.527	0.588	0.063
p(AgeGroup | Employed)	0.902	0.996	0.833	1.007	0.0714
p(Employed | Gender)	0.372	0.384	0.419	0.462	0.035

Table 6.4: SRMSE measured on bivariate conditional probabilities for synthetic agents using varying privacy noise levels. σ_{mse} denotes the variance between SRMSE.

As can seen in table 6.4, introducing noise impacts prediction errors of the conditional probabilities of synthesized agents. Adding a noise of 1.0, 2.0 and 3.0, the SRMSE of p(Permit |Gender) increased from 0.432 to 0.576, 0.464 and 0.497 respectively. Similar to random perturbations in the marginals, the SRMSE does not exhibit a monotonic pattern hence suggesting randomness in the model prediction which makes it difficult for an adversary to estimate the pattern of prediction. For an example, at a noise of 1.0, SRMSE of p(AgeGroup | Employed) increases to 0.996 but drops to 0.833 at noise of 2.0.

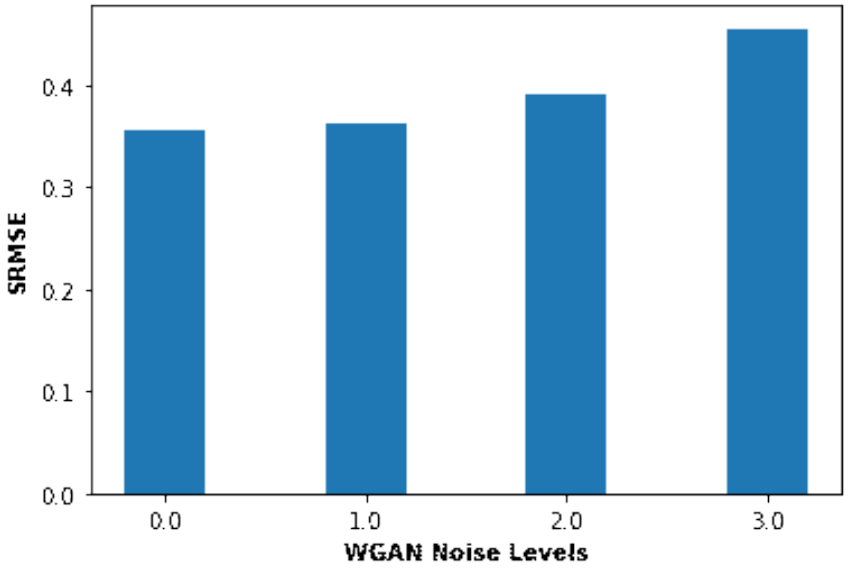
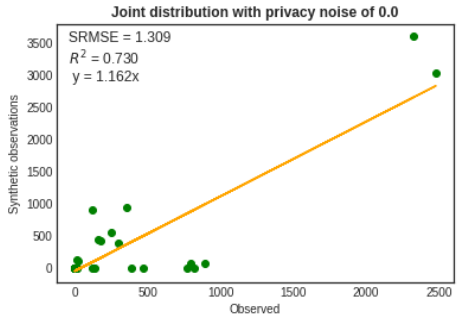
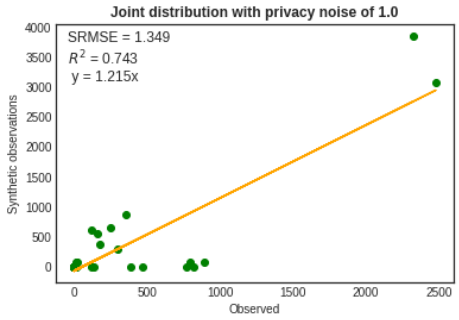


Figure 6.5: SRMSE on predictions for marginal distributions of synthetic agents using varying privacy noise levels

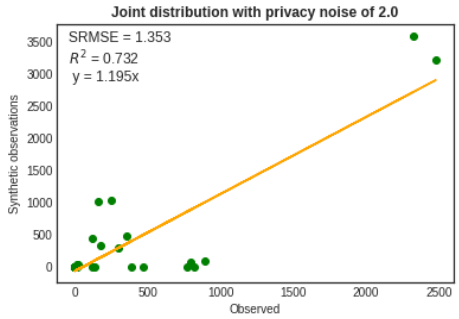
Finally, the joint probabilities were evaluated on all possible combinations of values on all data variables. We computed the frequency bins of all combinations for variables, p(AgeGroup, Employed, Gender, Educated). We show the fitting of the joint distributions for synthetic agents in Figure 6.6. The results of the generative model show a low performance in reproducing the joint probabilities of the synthetic agents. This inaccuracy of prediction can be attributed to the shared latent space that reduces the resolution of network parameters where multiple node branches having different dimensions are merged or compressed, as seen in the generator network of DP-CTGAN where the tabular and sequential branches are concatenated. This is evident in the joint distribution outputs in Figure 6.6 predicting synthetic frequency counts of 0 where true population counts are about 800. The line of fit exhibits a population balance observed between frequency bins of joint combinations for the synthetic agents. The mean square error of the prediction output increases with the magnitude of privacy noise added. For example, training at noise 1.0, SRMSE increases from 1.309 to 1.349. These marginal increases of SRMSE are consistent for larger noise additions as seen in Figure 6.6 (c) and (d). The model exhibits the capability of maintaining a good joint distribution even with the introduction of noise.



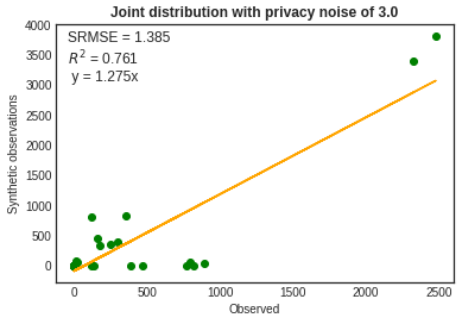
(a) Full joint distribution of at privacy noise level of **0.0**



(b) Full joint distribution of at privacy noise level of **1.0**

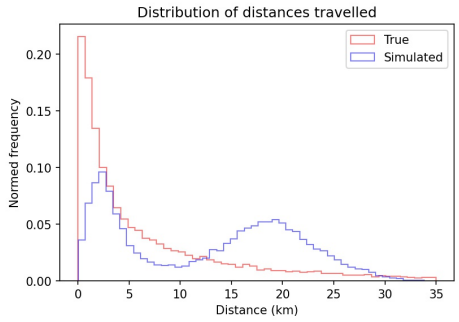


(c) Full joint distribution of at privacy noise level of **2.0**

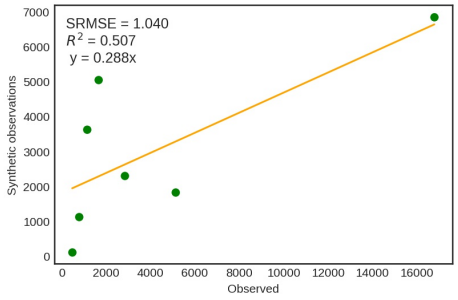


(d) Full joint distribution of at privacy noise level of **3.0**

Figure 6.6: Full joint distributions for all variables between observed and simulated counts.



(a) Marginal distribution on trip distances



(b) Line of fit on counts of trip distances

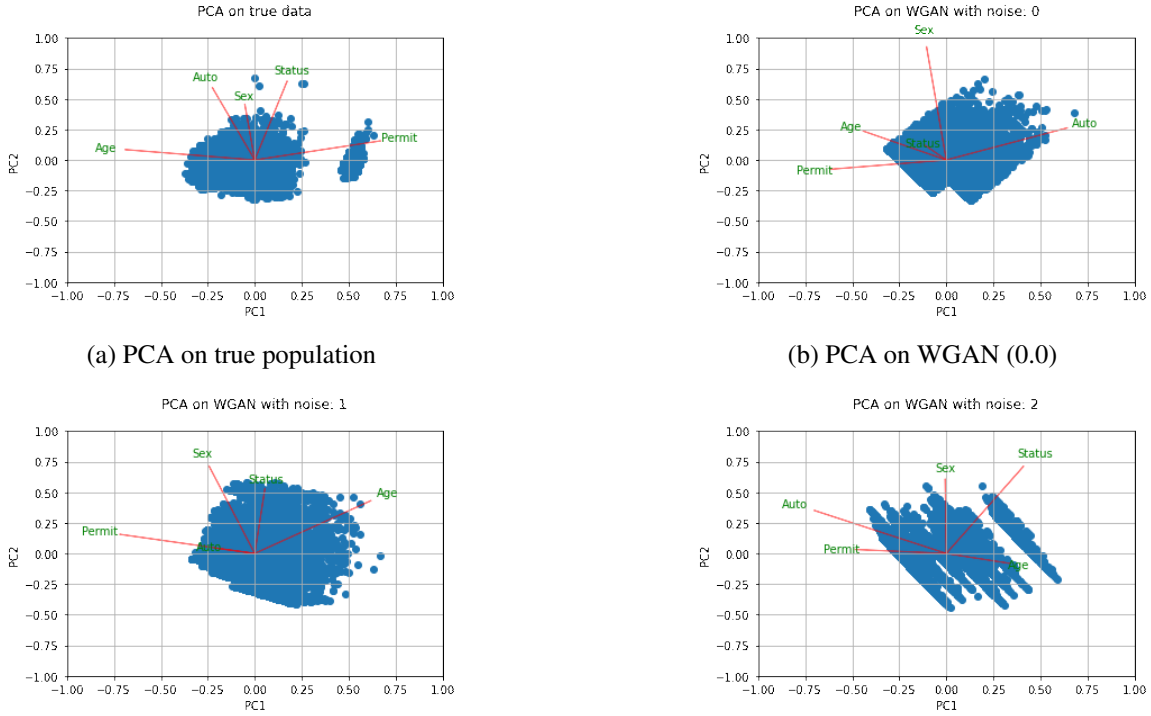
Figure 6.7: Comparison of distributions and fitting analysis between true and synthetic trip counts.

6.6.2 Trip length distribution

In this analysis, we assess the similarity in sequence representations simulated by the generative model by calculating the euclidean distances between sequences of origin and destination geographic coordinates for true and simulated set. We assume an agent embarks on a trip composing of a sequence of location destinations based on an activity preference within time periods in a day. We evaluate the lengths of all trip segments made up of an origin and destination, and compare the joint distributions between the true and synthetic representations. In Fig 6.7a, we show the marginal distributions of computed trip distances for both the true and synthetic representations. It can be observed that the model under-predicts trip distances between 0km to 3km and 5km to 11km. Contrarily, the model over-predicts trip distances from 13km to 29km. While the memory capacity of the LSTM [306] promises of learning correlations and representations for longer sequences, it can be observed that the model suffers the complexity of learning higher order correlations and long-range temporal dependencies needed for multiples features in longer timesteps. This drawback makes it difficult to learn longer sequences in complex generative architectures that could involve two or more networks learning with back-propagation. Oord et al. [307] have recently proposed a dilated convolution approach to address this drawback in generative longer sequences. To the best of our knowledge, this is the first attempt to implement such architecture in a multi-output with variable trip sequences thus this drawback needs further research to improve the prediction accuracy for long temporal dependencies. Similarly, the line of fit for trip length counts in Fig 6.7b shows an imbalance in the prediction, and recording a SRMSE of 1.040 and adjusted R squared of 50

6.6.3 Dimension reduction on principal components

In this section, we explore analysis using PCA to identify the main axes of variance within the synthetic agents and evaluation on how the orthogonal variables correlate with the principal components. PCA constructs relevant features through linear combinations of the original variables. This construction is implemented by linearly transforming correlated features into a lower dimensions of uncorrelated features using the eigenvectors of the correlation matrix. In this sense, the PCA undertakes an orthogonal transformation of the data into a reduced PCA space such that its derived components explain the most variance in the data.



(a) PCA on true population

(b) PCA on WGAN (0.0)

(c) PCA on WGAN (1.0)

(d) PCA on WGAN (2.0)

Figure 6.8: Principal component analysis on true and synthesized agents with varying privacy noise levels.

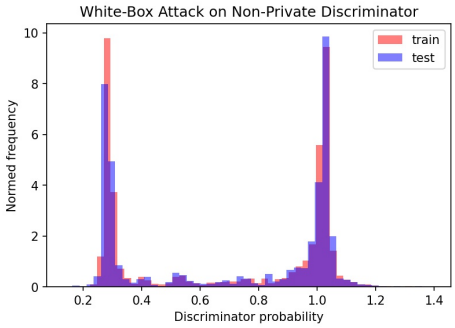
In Figure 6.8, the biplots and loadings plot show the association between the orthogonal variables and their clusters. Variables *Age, Sex and Status* are highly associated and form a cluster in representations of the true set and synthesized with noise of 0. With an introduction of noise with magnitude 1 and 2, the angles between these 3 variable vectors widen suggesting lower associations between them. Similarly, Principal component 2 (PC2) shows a strong correlation with variables

Age, Sex and Status in Figure 6.8(a) and (b), and exhibiting a high positive loadings suggesting PC2 will increase when the scores of the three variables increase. As noise is introduced shown in Figure 6.8(c) and (d), these correlations decay.

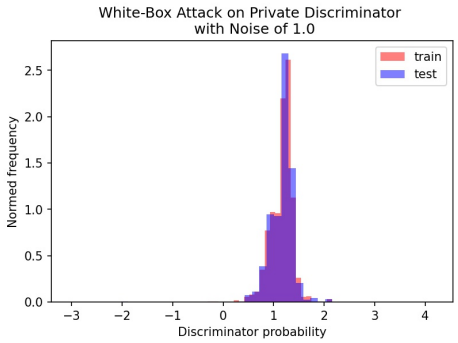
In summary, the PCA analysis has shows that the introduction of noise distorts associations and correlations in representations of synthetic agents. Therefore, the magnitude of noise controls the level of distortions that influence the correlations in synthesized representations.

6.6.4 Adversarial predictions on target models with knowledge on parameters

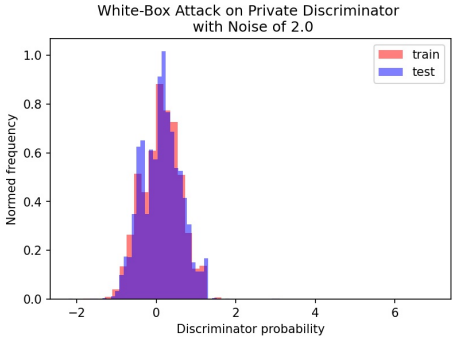
The adversary assumes knowledge on the model parameter of the target model, *the trained discriminator*. Fundamentally, the objective of the discriminator is to distinguish between true or fake samples. This means samples in the training set of the model should have higher predictive score than validation sets. As observed in Fig 6.9a, a bimodal distribution with two peaks having means approximating to zero (0) and one (1) are recorded for prediction scores from the non-private trained discriminator model. The bimodal distribution affirms the accuracy of classification outputs from the target model. This interprets that when an adversary has data on the entire population including the training sets at his disposal, he can predict with high confidence level on whether a sample data point was used in the training of the model. It can also be seen that the discriminator does not perfectly classify but shows traces of proportions of validation sets also having a high score and vice versa. This occurs because of similarity in features for both training and validation sets thus members of the sample population have near similar attributes. In Fig 6.9b, a unimodal distribution centered around 1 is derived for both the training and validation sets. The two peaks of the bimodal distribution as expected for the classification by a discriminator diminishes into a unimodal. This means the target model fails to classify between the samples that were used for training and validation. In this sense, the adversary cannot exploit the target model to infer if a data point was used in the training. This is the promise of differential private training by stochastic gradient descent [26]. We perform sensitive test on differing noise levels to privately train the target model as shown in 6.9c and d. The results show a consistency with unimodal prediction score suggesting the failure of the adversary to correctly classify training or validation sets. In summary, privacy protection is guaranteed on the differential-private trained models against any attacks in case the adversary has



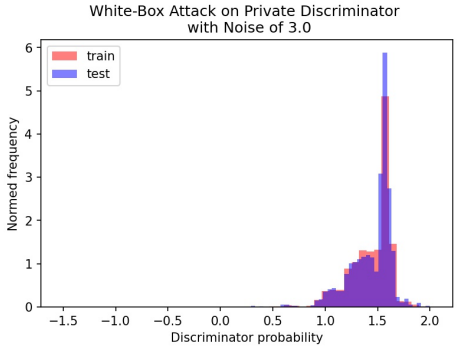
(a) Attack with no private training



(b) Attack with private training at noise level of **1.0**



(c) Attack with private training at noise level of **2.0**



(d) Attack with private training at noise level of **3.0**

Figure 6.9: White-box attacks on trained discriminator model with varying noise levels.

access to the parameters of the target model. In this sense, any record of a person that participated in the training of the model cannot be confidently identified by an adversary who has access to the model parameters.

6.7 Discussions and Conclusions

We developed and demonstrated the use of the novel Differentially-Private Composite Travel Generation Adversarial Network (DP-CTGAN) for activity diary synthesis, accounting for both accuracy of the population synthesized and privacy of the individuals. This generative model shows success in simulating activity diaries composed of multiple outputs including structured socio-economic features and sequential trip activities in a differentially private manner. To implement privacy-by-design, the generative model was trained by bounded gradients of data points with added Gaussian

noise propagated from the discriminator. The outputs of the synthetic agents shown appreciable similarity in statistical properties to the true population such that the synthetic agents proved suitable for microsimulation modelling while protecting the privacy of data points in the training set.

We evaluated the similarity in statistical properties by comparing the marginals, conditionals and joint probabilities of synthetic representations of varying privacy noise levels to the true distribution. The results shown a consistent output which gives a level of randomness influenced by the addition of noise such that the query probabilities differed by the specified level of noise. Observed root mean square error increased consequentially with the addition of more noise on a line of fit of synthetic probabilities to true distribution.

To the best of our knowledge, this is the first work that adopts a deep learning approach to simulate population synthesis of trip data having multiple outputs with different dimensions (i.e., structured and sequential) and guarantees privacy protection. While the model promises of reproducibility of sequential activities, our approach suffers limitations on training multiple features in long sequence generation using LSTM. The LSTM approach was not effective in producing accurate samples for higher dimensional and longer sequences. In practice, most existing literature have implemented similar sequence generations with one dimensional data. We foresee this drawback as a research interest that needs to be further studied to develop models that are robust and efficient in generation of multiple features and lengthy sequences as required by travel trajectories. Also, the approach of training deep learning with differential privacy as implemented in this paper makes it intractable to sample geographic location points whose positional accuracies are of high priority in transportation modelling. During training with differential privacy, the magnitude of random noise injected could perturb the normalized location coordinates making them less useful for microsimulation. In our future research, we will perform detailed sensitivity analysis on hyper-parameters that can control the injection of noise on accuracy and precision demanding variables like positional coordinates when used in synthesis approaches.

Further research will be done to improve on the fitting for true population especially in the context of the conditional and full joint probabilities of the empirical distributions. Also, we will perform state-of-art adversarial attacks [285, 308] on the generative model to test on its robustness against attacks such where an adversary with enough background knowledge seeks to infer whether information

of an individual was used in the training data or when the adversary seeks to learn something new about an individual from the synthesized outputs.

Finally, we will extend this research into designing model frameworks that support privacy-by-design techniques in generating usable location and activity diary sequences for microsimulation. While little research is available on benchmarks for evaluating multidimensional sequences, we will further this work into defining and developing such metrics in the context of location-aware privacy protection.

Chapter 7

Conclusion

This dissertation presents a series of papers to address the trade-off of using big data for transportation planning and the requirement of privacy protection for data owners. In the first paper, we highlight the rapid advances in transportation technologies propelled by the birth of location-aware mobile sensing technologies and smartphones that are capable of collecting high volumes of mobility and traffic data, hitherto was not possible with traditional data collection approaches in Chapter 2.1. While these modern data gathering techniques are promising, they come with challenges to overcome to harness the full potential of Big Transportation Data. We identified and discussed opportunities that are to be adapted to meet these challenges for transportation decision makers.

In Chapter 4, the paper connects with the previous chapter but with an emphasis on the challenge of ensuring privacy protection as it experiments with existing anonymization techniques to perturb sensitive location data (i.e., home, work) in an open travel survey. The chapter presents quantitative metrics to evaluate the privacy guarantee, and also the utility trade-off of using privacy-enhanced location data for behavioural modelling. As a step to address the limitations of the privacy and utility trade-off in location perturbation mechanisms, Chapter 5 presented a novel deep generative model as an extension of population synthesis to simultaneously synthesize tabular and sequential features of travel data. This approach used a multiple generators-discriminators network, the Composite Travel GAN (CTGAN) architecture that operate concurrently but is computationally expensive for large-scale travel data. In Chapter 6, we improve on the CTGAN architecture in the previous chapter to develop a multitask generative adversarial network using a shared layer, accepting multiple

inputs and generates both tabular and sequential representations. We embed a privacy-by-design mechanism in the overall design to protect privacy from adversarial attacks while satisfying the data needs of fine-grained behavioural modelling.

7.1 Key research findings

Among the numerous challenges to the use of Big Data transportation applications as highlighted in Chapter 3, we focused this dissertation on addressing the challenge of privacy protection due to the recent public outcry on information leakages and privacy violations. To accomplish privacy protection in the release of sensitive geographic location information, we experimented with the most common population anonymization mechanisms: *k-anonymity* and *differential privacy* on location data. From our analysis, the k-anonymity approach perturbed the location points in a deterministic fashion and influenced by the size of K, the expected indistinguishable population. This approach was found therefore vulnerable to an adversary who has some background knowledge of the study region. The differential privacy technique was stochastic and shown to be robust against adversarial attacks. While both approaches were found to provide a level of protection, there is an inherent trade-off of achieving privacy at the expense of losing utility and vice versa.

Population synthesis is a traditional approach addressing the concerns of privacy while gaining satisfactory utility on data needs for fine-grained behaviour modelling. We developed a deep generative adversarial network architecture (CTGAN) composed of multiple generators-discriminators to synthesize complete travel diaries for synthetic populations that included outputs of tabular and sequential features. The model resulted probability distributions closely matching the original data, with a better (lower) SRMSE of 0.224 when compared to results using a Variational Auto-encoder (VAE) approach of 0.292. Sequences of activity location points were also encoded into one-dimensional categorical representation and the model showed the ability to model the temporal dependencies of sequential locations.

While the GAN approach samples from the distribution of the latent space, the work of [284] has shown the generative model could leak information with similar features to the true distribution. In

this regard, we revised and improved the CTGAN architecture against adversarial attacks. As an improvement, we used a multi-task generative approach instead of multiple generators-discriminators to efficiently reduce the computation time of model training, and introduced differential privacy in the model training process such that the gradients learnt on each data point were clipped and Gaussian noise added. This approach limited the influence of the gradient of a single data point on the learning process hence making it difficult for an adversary to infer if a member of the population was used in the training. We performed adversarial attacks on the model and the results showed robustness against predictions on the target discriminative model by an adversary. Consequently, the model was capable of producing underlying distributions similar to the true distribution for the tabular features. However, the model could not reproduce the temporal dependencies for longer sequences with multiple features (i.e. real-valued sequences of location coordinates). We evaluated the correlations across variables using PCA on transformed vectors. The vector of principal components sustain relative associations in both the true and synthetic sets thus ensuring correlations are not decayed.

7.2 Study limitations

In this dissertation, we have shown the capability of implementing deep generative models to synthesize multi-input and varying dimensions (i.e. tabular, sequences) through a differentially-private manner while given multiple outputs synchronously. While this development is notable, it presents new challenges that need further research work to address. First, as noted in the discussion of Chapter 6, the generative model is not efficient in learning the temporal dependencies for longer, variable sequences having multiple features. For the variability in sequence length of travel activity diaries, the proposed technique adopts padding with zeros to achieve a fixed sequence dimension as required by the network. While this works for text generation and other time-series data using the fixed window approach, location trajectories are unique because they can be composed of variable destination lengths that cannot be easily augmented by dummy locations. This could lead to distortions in the pattern and correlation of associated variables —especially in the inference of purpose or mode of a trip.

Second, geographic coordinates of location points are highly precise in nature and this precision needs to be sustained in the synthetic sets. The process of gradient optimization through back-propagation across multiple networks could decay the level of granularity and precision of the co-ordinates making them less useful for transportation modeling. The discriminator could be adjusted through an objective function with promise of high accuracy to achieve high precision of generated coordinate vectors. Also, the intuitive objective of the generator network is to approximate the distribution of the true population from which new samples are generated. Samples are selected from an approximation of the distribution, the desired precision of location coordinates decays given that the true distribution exhibits a stochastic nature having uncertainties while the latent distribution is smoothened. This requirement will require further research to find an optimal methodology for generating high precision vector sequences.

Finally, while the differentially-private model training approach guarantees a level of privacy, the clipping of gradients and noise addition largely distorts the spatial resolutions making generated location sets less useful for behaviour modelling. A prudent methodology to solve this drawback is to use spatial encodings like Google S2 [273] to use a one-dimensional encoding to represent the location pairs. Using this technique, we assume the encodings as discrete for the network, which will subsequently suffer the inherent drawbacks of generation of discrete sequences in GAN architecture [1]. Perturbing the discrete latent representations with noise additions could be intractable to achieve a privacy-enhancement while satisfying utility for transportation modelling.

7.3 Practical implications

The research presented in this dissertation focuses on privacy techniques to ensure mobility data in travel behavioural analysis does not leak sensitive information about members of a population. The proposed differentially-private generative architecture is an example of a "privacy-by-Design" technique, which ensures privacy protection and also satisfies the data needs of fine-grained transport modelling. In this regard, the methodologies developed in the dissertation are well suited to traffic and microsimulation modelling applications on populations with privacy requirements. Given the robustness of the model against adversarial and linkage attacks, it is difficult for an adversary

with any amount background knowledge to infer needed information on members of the population. The proposed model can be adopted by transportation planners to recreate data surrogates of the raw data in scenarios where privacy requirements are needed. Given this approach, custodians of travel data will not be required to anonymize or perturb data points with random noise which in effect decays the utility of the anonymized data. Periodically, the model can be trained on new data and subsequently its parameters can be shared with researchers who can predict newer privacy-enhanced samples for travel behaviour modelling and analysis. It should be noted that the model can confidently reproduce tabular attributes as well as for sequential activity location coordinates. For the sequence generation of location coordinates, the model obfuscates by adding random noise to the generated location points. This will require further map matching to trace routes on generated sequences. We will extend research on this drawback to present a generalization-based synthetic sequences such that the spatial resolutions are sustained while providing privacy guarantee.

7.4 Future works

In subsequent work, we will extend this work to improve the model's learning and prediction of higher-order representations of multiple features in long temporal dependencies for generative models. We will experiment on the stacking of multiple layers and varying node sizes to test the impact on the stability of model parameters and prediction output. Second, to the best of the author's knowledge, there is no work that has looked into the generation of sequences with varying lengths. This is a needed requirement for activity diary synthesis which generally consist of varying lengths of trip trajectories. We will explore the best approaches to adopt to generate multiple features with varying sequence lengths. Finally, the proposed system needs to be tested and improved using sophisticated adversarial models, and more traffic modelling applications.

Glossary

Trajectory: A trajectory (or GPS trajectory) refers to a sequence of time-stamped points, usually recorded with some other information about latitude, longitude, altitude, speed, and acceleration, etc. The trajectories are typically recorded by location-aware devices such as smartphones, smart watch, handheld GPS and others.

Transit Itinerary: A transit itinerary is the details of scheduled events relating to a transit trip, generally including the bus number or metro line taken by the traveler at specific times and locations.

Trip: A trip is considered as a single journey between two points made by a specific mode of transport and has a defined purpose. As an example, a trip can be done from home to work by car.

Travel Time: While travel time is usually considered as the time it takes to travel door-to-door, in this thesis travel time is defined as the time between the first and last GPS point along a detected trip trajectory.

Sensitive attribute: This is a information that a respondent will not want to be disclosed. In this thesis, we consider the home and work locations as sensitive attributes.

Personal identifiable information: This is any data that could potentially reveal the identity of a specific person. For example, the full name, driver's license, social security number and email address.

Utility: defines how well a task could be performed on either the true dataset or an anonymized version to obtain similar or near-similar results. It must not be confused with the concept in microeconomics with similar name.

Quasi-identifiers: are variables or pieces of information that are independently not unique identifiers but are sufficiently well correlation with other variables such that its combination with other

variables could reveal the identity of a person.

Sensitive location: refers to any residential point locations represented in its Cartesian coordinates as latitudes and longitudes that needs to be protected to prevent the identification of a user.

k-anonymity: refers to the population within a buffer region of the outer radius around the original point prior to displacement, from which a de-identified cluster case cannot be reversely identified. K is the population around a sensitive location that could be associated with equal probability to a perturbed location. If K, were for example 10, then the perturbed location could be equally attributable to 10 different households.

Protection Radius: refers to a circular region around a sensitive location within which other location points existing should be made indistinguishable from the sensitive point.

Location-privacy protection mechanism: or LLPM refers to mechanisms that modify datasets to offer privacy guarantees by adding a level of noise to displace the sensitive location to distances away from their true location. Protected datasets are also referred to as geomasked datasets.

Adversary: This is an agent seeking to re-identify true residential location of the user by infer-ring from sanitized dataset.

Adversarial modelling: refers to the technique of identifying attackers based on malicious intent and suspicious behaviors, versus only searching for specific indicators of an attack. This model demands knowledge of the model parameters to identify outliers and suspicious patterns.

Generative modeling refers to unsupervised learning approaches used in machine learning to discover and learn irregularities or patterns in an input data such that the model can reproduce newer samples by estimating the joint distribution.

Bibliography

[1] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[2] Godwin Badu-Marfo, Bilal Farooq, and Zachary Patterson. A perspective on the challenges and opportunities for privacy-aware big transportation data. *Journal of Big Data Analytics in Transportation*, 1(1):1–23, 2019.

[3] Pew Research. A third of americans live in a household with three or more smartphones. `https://www.pewresearch.org/fact-tank/2017/05/25/a-third-of-americans-live-in-a-household-with-three-or-more-smartphones/`, 2017. Accessed: 2019-02-26.

[4] Kathryn Zickuhr. Pew research center: Internet technology – location-based services. `http://www.pewinternet.org/2013/09/12/location-based-services/`, 2013. Accessed: 2019-02-26.

[5] Antonin Danalet, Bilal Farooq, and Michel Bierlaire. A bayesian approach to detect pedestrian destination-sequences from wifi signatures. *Transportation Research Part C: Emerging Technologies*, 44:146–170, 2014.

[6] Guilhem Poucin, Bilal Farooq, and Zachary Patterson. Pedestrian activity pattern mining in wifi-network connection data. Technical report, 2016.

[7] Bilal Farooq, Alexandra Beaulieu, Marwan Ragab, and Viet Dang Ba. Ubiquitous monitoring of pedestrian dynamics: Exploring wireless ad hoc network of multi-sensor technologies. In *SENSORS, 2015 IEEE*, pages 1–4. IEEE, 2015.

[8] Ali Yazdizadeh, Zachary Patterson, and Bilal Farooq. An automated approach from gps traces to complete trip information. Submitted to the *International Journal of Transportation Science and Technology*, 2018. Submitted: February, 2018.

[9] Jaume Barceló, Lidin Montero, Laura Marqués, and Carlos Carmona. Travel time forecasting and dynamic origin-destination estimation for freeways based on bluetooth traffic monitoring. *Transportation Research Record: Journal of the Transportation Research Board*, (2175):19–27, 2010.

[10] FW Cathey and DJ Dailey. A novel technique to dynamically measure vehicle speed using uncalibrated roadway cameras. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 777–782. IEEE, 2005.

[11] Zachary Patterson and Kyle Fitzsimmons. The Itinerum open smartphone travel survey platform. Technical Report, Concordia University TRIP Lab, Montreal, Canada, July 2017. TRIP Lab Working Paper 2017-2, available at: itinerum.ca/documents.html.

[12] Google. Google. `https://www.google.com/`, 2018. Accessed: 2017-06-12.

[13] Waze. Waze, 2018. URL `https://www.waze.com/`.

[14] Zachary Patterson, Kyle Fitzsimmons, Takeshi Mukai, and Stewart Jackson. Itinerum: The open smartphone travel survey platform. *SoftwareX*, 10, July-December 2019.

[15] Uber. Uber. `https://www.uber.com/`, 2018. Accessed: 2017-06-12.

[16] John Simermany. Fastrak to courthouse. east bay times, 2007. `http://www.eastbaytimes.com/2007/06/05/fastrak-to-courthouse/`, 2007. Accessed: 2019-02-26.

[17] Olivia Solon. Facebook says cambridge analytica may have gained 37m more users' data, 2018. URL `https://www.theguardian.com/technology/2018/apr/04/facebook-cambridge-analytica-user-data-latest-more-than-thought`.

[18] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[19] Godwin Badu-Marfo, Bilal Farooq, and Zachary Patterson. A perspective on the challenges and opportunities for privacy-aware big transportation data. Accepted for publication in the *Journal of Big Data Analytics in Transportation*.

[20] Godwin Badu-Marfo, Bilal Farooq, and Zachary Patterson. Perturbation methods for protection of sensitive location data: Smartphone travel survey case study. *Transportation Research Record*, 2673(12):244–255, 2019.

[21] Bugra Gedik and Ling Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing*, 7(1):1–18, 2008.

[22] Bilal Farooq, Michel Bierlaire, Ricardo Hurtubia, and Gunnar Flötteröd. Simulation based population synthesis. *Transportation Research Part B: Methodological*, 58:243–263, 2013.

[23] Stanislav S Borysov, Jeppe Rich, and Francisco Camara Pereira. Scalable population synthesis with deep generative modeling [arxiv]. *ArXiv*, 2019.

[24] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[25] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. Variational autoencoder for deep learning of images, labels and captions. In *Advances in neural information processing systems*, pages 2352–2360, 2016.

[26] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM, 2016.

[27] Gartner. Gartner IT Glossary. http://www.gartner.com/it-glossary/big-data/, 2012. Accessed: 2017-03-25.

[28] kdespagniqz. Connected cars will send 25 gigabytes of data to the cloud every hour. https://qz.com/344466/connected-cars-will-send-25-gigabytes-of-data-to-the-cloud-every-hour/, 2015. Accessed: 2019-02-26.

[29] Bernard Marr. How much data do we create every day? the mind-blowing stats everyone should read. https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/#18c8283460ba, 2018. Accessed: 2019-02-26.

[30] Rob Kitchin. *The data revolution: Big data, open data, data infrastructures and their consequences*. Sage, 2014.

[31] Amir Gandomi and Murtaza Haider. Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, 35(2):137–144, 2015.

[32] Alexander Hainen, Jason Wasson, Sarah Hubbard, Stephen Remias, Grant Farnsworth, and Darcy Bullock. Estimating route choice and travel time reliability with field observations of bluetooth probe vehicles. *Transportation Research Record: Journal of the Transportation Research Board*, (2256):43–50, 2011.

[33] Miguel Romero Mikkelsen and Pia Christensen. Is children's independent mobility really independent? a study of children's mobility combining ethnography and gps/mobile phone technologies. *Mobilities*, 4(1):37–58, 2009.

[34] Zachary Patterson and Kyle Fitzsimmons. Datamobile: Smartphone travel survey experiment. *Transportation Research Record: Journal of the Transportation Research Board*, (2594):35–43, 2016.

[35] Juan de Dios Ortúzar and Luis G Willumsen. *Modelling transport*. John wiley & sons, 2011.

[36] John L Bowman, Mark Bradley, Joe Castiglione, and Supin L Yoder. Making advanced travel forecasting models affordable through model transferability. Technical report, 2014.

[37] Joe Castiglione, Mark Bradley, and John Gliebe. *Activity-based travel demand models: a primer*. Number SHRP 2 Report S2-C46-RR-1. 2015.

[38] C Zhang. Analysis on the characteristics of urban rail transit passenger flow [ph. d. thesis]. *Southwest Jiaotong University, Chengdu, China*, 2006.

[39] CAO Shouhua, YUAN Zhenzhou, Chiqing Zhang, and ZHAO Li. Los classification for urban rail transit passages based on passenger perceptions. *Journal of transportation systems engineering and information technology*, 9(2):99–104, 2009.

[40] Joel Horowitz. A utility maximizing model of the demand for multi-destination non-work travel. *Transportation Research Part B: Methodological*, 14(4):369–386, 1980.

[41] APTA. Apta transit ridership report, 2018. URL `https://www.apta.com/resources/statistics/Documents/Ridership/2018-Q1-Ridership-APTA.pdf`. last accessed on March 05, 2019.

[42] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[43] Cynthia Dwork. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*, pages 1–19. Springer, 2008.

[44] Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, Technical report, SRI International, 1998.

[45] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, pages 24–24. IEEE, 2006.

[46] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 106–115. IEEE, 2007.

[47] Qing Zhang, Nick Koudas, Divesh Srivastava, and Ting Yu. Aggregate query answering on anonymized tables. In *2007 IEEE 23rd international conference on data engineering*, pages 116–125. IEEE, 2007.

[48] Raymond Chi-Wing Wong, Jiuyong Li, Ada Wai-Chee Fu, and Ke Wang. (α, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 754–759. ACM, 2006.

[49] Xin Wang, Asad J Khattak, and Sanghoon Son. What can be learned from analyzing university student travel demand? *Transportation research record*, 2322(1):129–137, 2012.

[50] Frank Kargl, Arik Friedman, and Roksana Boreli. Differential privacy in intelligent transportation systems. In *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks*, pages 107–112. ACM, 2013.

[51] Richard J Beckman, Keith A Baggerly, and Michael D McKay. Creating synthetic baseline populations. *Transportation Research Part A: Policy and Practice*, 30(6):415–429, 1996.

[52] Ali Yazdizadeh, Zachary Patterson, and Bilal Farooq. An automated approach from gps traces to complete trip information. *International Journal of Transportation Science and Technology*, 8(1):82–100, 2019.

[53] Zachary Patterson and Kyle Fitzsimmons. MTL Trajet. Working Paper 2017-2, Concordia University, TRIP Lab, Montreal, Canada, 2017.

[54] Eric J Miller, Bilal Farooq, Franco Chingcuanco, and David Wang. Historical validation of integrated transport–land use model system. *Transportation Research Record*, 2255(1): 91–99, 2011.

[55] John Landis and Ming Zhang. The second generation of the california urban futures model. part 1: Model logic and theory. *Environment and Planning B: Planning and Design*, 25(5): 657–666, 1998.

[56] John Landis and Ming Zhang. The second generation of the california urban futures model. part 2: Specification and calibration results of the land-use change submodel. *Environment and Planning B: Planning and Design*, 25(6):795–824, 1998.

[57] Gerald Duguay, Woo Jung, and Daniel McFadden. *SYNSAM: A methodology for synthesizing household transportation survey data*. Urban Travel Demand Forecasting Project, Institute of Transportation Studies, 1976.

[58] Alan Geoffrey Wilson and Carol E Pownall. A new representation of the urban system for modelling and for the study of micro-level interdependence. *Area*, pages 246–254, 1976.

[59] Rolf Moeckel, Klaus Spiekermann, and Michael Wegener. Creating a synthetic population. In *Proceedings of the 8th International Conference on Computers in Urban Planning and Urban Management (CUPUM)*, pages 1–18, 2003.

[60] Andrew Y Ng and Michael I Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems*, pages 841–848, 2002.

[61] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[62] Carl Edward Rasmussen. The infinite gaussian mixture model. In *Advances in neural information processing systems*, pages 554–560, 2000.

[63] Geoffrey E Hinton. A practical guide to training restricted boltzmann machines. In *Neural networks: Tricks of the trade*, pages 599–619. Springer, 2012.

[64] Geoffrey E Hinton. Deep belief networks. *Scholarpedia*, 4(5):5947, 2009.

[65] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[66] Kun Ouyang, Reza Shokri, David S Rosenblum, and Wenzhuo Yang. A non-parametric generative model for human trajectories. In *IJCAI*, pages 3812–3817, 2018.

[67] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[68] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[69] Jost Tobias Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390*, 2015.

[70] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[71] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[72] Matt J Kusner and José Miguel Hernández-Lobato. Gans for sequences of discrete elements with the gumbel-softmax distribution. *arXiv preprint arXiv:1611.04051*, 2016.

[73] Guilhem Poucin, Bilal Farooq, and Zachary Patterson. Activity patterns mining in wi-fi access point logs. *Computers, Environment and Urban Systems*, 67:55–67, 2018.

[74] StreetLight. StreetLight Data. `https://www.streetlightdata.com`, 2018. Accessed: 2017-06-15.

[75] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang. Traffic flow prediction with big data: a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):865–873, 2015.

[76] Honghui Dong, Mingchao Wu, Xiaoqing Ding, Lianyu Chu, Limin Jia, Yong Qin, and Xuesong Zhou. Traffic zone division based on big data from mobile phone base stations. *Transportation Research Part C: Emerging Technologies*, 58:278–291, 2015.

[77] Xinhu Zheng, Wei Chen, Pu Wang, Dayong Shen, Songhang Chen, Xiao Wang, Qingpeng Zhang, and Liuqing Yang. Big data for social transportation. *IEEE Transactions on Intelligent Transportation Systems*, 17(3):620–630, 2016.

[78] Cynthia Chen, Jingtao Ma, Yusak Susilo, Yu Liu, and Menglin Wang. The promises of big data and small data for travel behavior (aka human mobility) analysis. *Transportation research part C: emerging technologies*, 68:285–299, 2016.

[79] Guillaume Leduc. Road traffic data: Collection methods and applications. *Working Papers on Energy, Transport and Climate Change*, 1(55), 2008.

[80] Qi Shi and Mohamed Abdel-Aty. Big data applications in real-time traffic operation and safety monitoring and improvement on urban expressways. *Transportation Research Part C: Emerging Technologies*, 58:380–394, 2015.

[81] Philippe Nitsche, Peter Widhalm, Simon Breuss, Norbert Brändle, and Peter Maurer. Supporting large-scale travel surveys with smartphones–a practical approach. *Transportation Research Part C: Emerging Technologies*, 43:212–221, 2014.

[82] Jianting Zhang, Simin You, and Le Gruenwald. High-performance spatial query processing on big taxi trip data using gpgpus. In *Big Data (BigData Congress), 2014 IEEE International Congress on*, pages 72–79. IEEE, 2014.

[83] Colin Tankard. Big data security. *Network security*, 2012(7):5–8, 2012.

[84] Brian Tierney, Ezra Kissel, Martin Swany, and Eric Pouyoul. Efficient data transfer protocols for big data. In *E-Science (e-Science), 2012 IEEE 8th International Conference on*, pages 1–9. IEEE, 2012.

[85] Carl Lagoze. Big data, data integrity, and the fracturing of the control zone. *Big Data & Society*, 1(2):2053951714558281, 2014.

[86] Andrew McAfee, Erik Brynjolfsson, Thomas H Davenport, DJ Patil, and Dominic Barton. Big data: the management revolution. *Harvard business review*, 90(10):60–68, 2012.

[87] Ibrahim Abaker Targio Hashem, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, and Samee Ullah Khan. The rise of "big data" on cloud computing: Review and open research issues. *Information Systems*, 47:98–115, 2015.

[88] Paul Zikopoulos, Chris Eaton, et al. *Understanding big data: Analytics for enterprise class hadoop and streaming data*. McGraw-Hill Osborne Media, 2011.

[89] Xindong Wu, Xingquan Zhu, Gong-Qing Wu, and Wei Ding. Data mining with big data. *IEEE transactions on knowledge and data engineering*, 26(1):97–107, 2014.

[90] Martin Hilbert and Priscila López. The world's technological capacity to store, communicate, and compute information. *science*, 332(6025):60–65, 2011.

[91] HV Jagadish, Johannes Gehrke, Alexandros Labrinidis, Yannis Papakonstantinou, Jignesh M Patel, Raghu Ramakrishnan, and Cyrus Shahabi. Big data and its technical challenges. *Communications of the ACM*, 57(7):86–94, 2014.

[92] Scott D Kahn. On the future of genomic data. *science*, 331(6018):728–729, 2011.

[93] CL Philip Chen and Chun-Yang Zhang. Data-intensive applications, challenges, techniques and technologies: A survey on big data. *Information Sciences*, 275:314–347, 2014.

[94] Avita Katal, Mohammad Wazid, and RH Goudar. Big data: issues, challenges, tools and good practices. In *Contemporary Computing (IC3), 2013 Sixth International Conference on*, pages 404–409. IEEE, 2013.

[95] David Gewirtz. Volume, velocity, and variety: Understanding the three v's of big data. *DIY-T*, 2018.

[96] Imran R Mansuri and Sunita Sarawagi. Integrating unstructured data into relational databases. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, pages 29–29. IEEE, 2006.

[97] Arthur Choi, Todd L Leyba, Beate Porst, and Amit Radheshyam Somani. Real-time aggregation of unstructured data into structured data for sql processing by a relational database engine, December 5 2006. US Patent 7,146,356.

[98] AnHai Doan, Jeffrey F Naughton, Raghu Ramakrishnan, Akanksha Baid, Xiaoyong Chai, Fei Chen, Ting Chen, Eric Chu, Pedro DeRose, Byron Gao, et al. Information extraction challenges in managing unstructured data. *ACM SIGMOD Record*, 37(4):14–20, 2009.

[99] Pierangela Samarati. Protecting respondents identities in microdata release. *IEEE transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[100] Lei Xu, Chunxiao Jiang, Jian Wang, Jian Yuan, and Yong Ren. Information security in big data: privacy and data mining. *IEEE Access*, 2:1149–1176, 2014.

[101] Paolo Perego, Giuseppe Andreoni, and Giovanna Rizzo. *Wireless Mobile Communication and Healthcare: 6th International Conference, MobiHealth 2016, Milan, Italy, November 14-16, 2016, Proceedings*, volume 192. Springer, 2017.

[102] Vishal Ashok Gadhe, Ashwini Subhash Nimse, Amruta Satish Tanpure, and PD Sinare. Networking smartphones for disaster recovery using teamphone. 2017.

[103] Mark Stamp. *Information security: principles and practice*. John Wiley & Sons, 2011.

[104] Mousumi Bagchi and Peter R White. The potential of public transport smart card data. *Transport Policy*, 12(5):464–474, 2005.

[105] Marie-Pier Pelletier, Martin Trépanier, and Catherine Morency. Smart card data use in public transit: A literature review. *Transportation Research Part C: Emerging Technologies*, 19(4): 557–568, 2011.

[106] M Trépanier and C Morency. Assessing transit loyalty with smart card data. In *12th World Conference on Transport Research, July*, pages 11–15, 2010.

[107] Neveen Shlayan, Abdullah Kurkcu, and Kaan Ozbay. Exploring pedestrian bluetooth and wifi detection at public transportation terminals. In *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, pages 229–234. IEEE, 2016.

[108] Alfred Leick, Lev Rapoport, and Dmitry Tatarnikov. *GPS satellite surveying*. John Wiley & Sons, 2015.

[109] Frank Stephen Tromp Van Diggelen. *A-GPS: Assisted GPS, GNSS, and SBAS*. Artech House, 2009.

[110] Daniel K Davies, Steven E Stock, Shane Holloway, and Michael L Wehmeyer. Evaluating a gps-based transportation device to support independent bus travel by people with intellectual disability. *Intellectual and Developmental Disabilities*, 48(6):454–463, 2010.

[111] Geert Draijer, Nelly Kalfs, and Jan Perdok. Global positioning system as data collection method for travel research. *Transportation Research Record: Journal of the Transportation Research Board*, (1719):147–153, 2000.

[112] Peter R Stopher and Stephen P Greaves. Household travel surveys: Where are we going? *Transportation Research Part A: Policy and Practice*, 41(5):367–381, 2007.

[113] Lara Montini, Sebastian Prost, Johann Schrammel, Nadine Rieser-Schüssler, and Kay W Axhausen. Comparison of travel diaries generated from smartphone data and dedicated gps devices. *Transportation Research Procedia*, 11:227–241, 2015.

[114] Fang Zhao, Ajinkya Ghorpade, Francisco Câmara Pereira, Christopher Zegras, and Moshe Ben-Akiva. Stop detection in smartphone-based travel surveys. *Transportation Research Procedia*, 11:218–226, 2015.

[115] Wendy Bohte and Kees Maat. Deriving and validating trip purposes and travel modes for multi-day gps-based travel surveys: A large-scale application in the netherlands. *Transportation Research Part C: Emerging Technologies*, 17(3):285–297, 2009.

[116] Michel Bierlaire, Jingmin Chen, and Jeffrey Newman. A probabilistic map matching method for smartphone GPS data. *Transportation Research Part C: Emerging Technologies*, 26:78–98, 2013.

[117] Paola A Gonzalez, Jeremy S Weinstein, Sean J Barbeau, Miguel A Labrador, Philip L Winters, Nevine L Georggi, and Roxana Perez. Automating mode detection for travel behaviour analysis by using global positioning systems-enabled mobile phones and neural networks. *IET intelligent transport systems*, 4(1):37–49, 2010.

[118] Sasank Reddy, Min Mun, Jeff Burke, Deborah Estrin, Mark Hansen, and Mani Srivastava. Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks (TOSN)*, 6(2):13, 2010.

[119] Jean Wolf, Randall Guensler, and William Bachman. Elimination of the travel diary: Experiment to derive trip purpose from global positioning system travel data. *Transportation Research Record: Journal of the Transportation Research Board*, (1768):125–134, 2001.

[120] Li Shen and Peter R Stopher. A process for trip purpose imputation from global positioning system data. *Transportation Research Part C: Emerging Technologies*, 36:261–267, 2013.

[121] Patrick Tracy McGowen and Michael G McNally. Evaluating the potential to predict activity types from GPS and GIS data. In *Proceedings of Annual Meeting of the Transportation Research Board*, Washington, DC, January 2007. Transportation Research Board. Reference Number: 07-3199.

[122] Terry Griffin and Yan Huang. A decision tree classification model to automate trip purpose derivation. In *The Proceedings of the ISCA 18th International Conference on Computer Applications in Industry and Engineering*, pages 44–49, 2005.

[123] Christopher E White, David Bernstein, and Alain L Kornhauser. Some map matching algorithms for personal navigation assistants. *Transportation research part c: emerging technologies*, 8(1):91–108, 2000.

[124] Jeffrey Hood, Elizabeth Sall, and Billy Charlton. A GPS-based bicycle route choice model for san francisco, california. *Transportation letters*, 3(1):63–75, 2011.

[125] Seyed Amir H Zahabi, Ajang Ajzachi, and Zachary Patterson. Transit trip itinerary inference with GTFS and smartphone data. *Transportation Research Record: Journal of the Transportation Research Board*, (2652):59–69, 2017.

[126] Brian Krzanich. Data is the new oil in the future of automated driving, 2016. URL `https://newsroom.intel.com/editorials/krzanich-the-future-of-automated-driving/`.

[127] Theo Arentze, Harry Timmermans, Frank Hofman, and Nelly Kalfs. Data needs, data collection, and data quality requirements of activity-based transport demand models. *Transportation research circular*, (E-C008):30–p, 2000.

[128] Dimitrios Efthymiou and Constantinos Antoniou. Use of social media for transport data collection. *Procedia-Social and Behavioral Sciences*, 48:775–785, 2012.

[129] Daniel Abadi. Optimizing disk io and memory for big data vector analysis, 2016. URL `http://blogs.teradata.com/data-points/optimizing-disk-io-and-memory-for-big-data-vector-analysis/`.

[130] John Ousterhout and Fred Douglis. Beating the i/o bottleneck: A case for log-structured file systems. *ACM SIGOPS Operating Systems Review*, 23(1):11–28, 1989.

[131] Dimitris Tsirogiannis, Stavros Harizopoulos, Mehul A Shah, Janet L Wiener, and Goetz Graefe. Query processing techniques for solid state drives. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 59–72. ACM, 2009.

[132] Pooja Tanaji Patil. A study on evolution of storage infrastructure. *International Journal*, 6 (7), 2016.

[133] Andrew S Tanenbaum and Albert S Woodhull. *Operating systems: design and implementation*, volume 2. Prentice-Hall Englewood Cliffs, NJ, 1987.

[134] Edgar F Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.

[135] Chad Vicknair, Michael Macias, Zhendong Zhao, Xiaofei Nan, Yixin Chen, and Dawn Wilkins. A comparison of a graph database and a relational database: a data provenance perspective. In *Proceedings of the 48th annual Southeast regional conference*, page 42. ACM, 2010.

[136] Jim Gray and Andreas Reuter. *Transaction processing: concepts and techniques*. Elsevier, 1992.

[137] David Maier. *The theory of relational databases*, volume 11. Computer science press Rockville, 1983.

[138] Robert R Schaller. Moore's law: past, present and future. *IEEE spectrum*, 34(6):52–59, 1997.

[139] Laszlo B Kish. End of moore's law: thermal (noise) death of integration in micro and nano electronics. *Physics Letters A*, 305(3-4):144–149, 2002.

[140] ABM Moniruzzaman and Syed Akhter Hossain. Nosql database: New era of databases for big data analytics-classification, characteristics and comparison. *arXiv preprint arXiv:1307.0191*, 2013.

[141] Luis M Vaquero, Luis Rodero-Merino, and Rajkumar Buyya. Dynamically scaling applications in the cloud. *ACM SIGCOMM Computer Communication Review*, 41(1):45–52, 2011.

[142] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. *The Google file system*, volume 37. ACM, 2003.

[143] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler. The hadoop distributed file system. In *Mass storage systems and technologies (MSST), 2010 IEEE 26th symposium on*, pages 1–10. Ieee, 2010.

[144] Dhruba Borthakur. The hadoop distributed file system: Architecture and design. *Hadoop Project Website*, 11(2007):21, 2007.

[145] Jeffrey Shafer, Scott Rixner, and Alan L Cox. The hadoop distributed filesystem: Balancing portability and performance. In *Performance Analysis of Systems & Software (ISPASS), 2010 IEEE International Symposium on*, pages 122–133. IEEE, 2010.

[146] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C Hsieh, Deborah A Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E Gruber. Bigtable: A distributed storage system for structured data. *ACM Transactions on Computer Systems (TOCS)*, 26(2): 4, 2008.

[147] Mehul Nalin Vora. Hadoop-hbase for large-scale data. In *Computer science and network technology (ICCSNT), 2011 international conference on*, volume 1, pages 601–605. IEEE, 2011.

[148] Eric A Brewer. Towards robust distributed systems. In *PODC*, volume 7, 2000.

[149] Seth Gilbert and Nancy Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *Acm Sigact News*, 33(2):51–59, 2002.

[150] Oracle. Managing consistency with berkeley db- ha (white paper), 2015. URL `http://www.oracle.com/technetwork/products/berkeleydb/high-availability-099050.html`. last accessed on May 5, 2015.

[151] Madjid Fathi. *Integration of Practice-Oriented Knowledge Technology: Trends and Prospectives*. Springer, 2013.

[152] Seth Gilbert and Nancy Lynch. Perspectives on the cap theorem. *Computer*, 45(2):30–36, 2012.

[153] Titus Irma Damaiyanti, Ardi Imawan, and Joonho Kwon. Querying road traffic data from a document store. In *Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*, pages 485–486. IEEE Computer Society, 2014.

[154] U.S. Department of Transportation. Some observations on probe data in the v2v world: A unified view of shared situation data, 2013.

[155] Sasan Amini, Ilias Gerostathopoulos, and Christian Prehofer. Big data analytics architecture for real-time traffic control. In *Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2017 5th IEEE International Conference on*, pages 710–715. IEEE, 2017.

[156] Jaroslav Pokorny. Nosql databases: a step to database scalability in web environment. *International Journal of Web Information Systems*, 9(1):69–82, 2013.

[157] Michael Dirolf and Kristina Chodorow. *MongoDB: the definitive guide*. O'Reilly Media, Incorporated, 2010.

[158] Avinash Lakshman and Prashant Malik. Cassandra: a decentralized structured storage system. *ACM SIGOPS Operating Systems Review*, 44(2):35–40, 2010.

[159] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, and Werner Vogels. Dynamo: amazon's highly available key-value store. In *ACM SIGOPS operating systems review*, volume 41, pages 205–220. ACM, 2007.

[160] Lars George. *HBase: the definitive guide: random access to your planet-size data*. " O'Reilly Media, Inc.", 2011.

[161] Ankur Khetrapal and Vinay Ganesh. Hbase and hypertable for large scale distributed storage systems. *Dept. of Computer Science, Purdue University*, pages 22–28, 2006.

[162] Kristina Chodorow. *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage*. " O'Reilly Media, Inc.", 2013.

[163] Rick Cattell. Scalable sql and nosql data stores. *Acm Sigmod Record*, 39(4):12–27, 2011.

[164] Andre Calil and Ronaldo dos Santos Mello. Simplesql: a relational layer for simpledb. In *East European Conference on Advances in Databases and Information Systems*, pages 99–110. Springer, 2012.

[165] J Chris Anderson, Jan Lehnardt, and Noah Slater. *CouchDB: The Definitive Guide: Time to Relax*. " O'Reilly Media, Inc.", 2010.

[166] Prabhakar Chaganti and Rich Helms. *Amazon SimpleDB Developer Guide*. Packt Publishing Ltd, 2010.

[167] Borislav Iordanov. Hypergraphdb: a generalized graph database. In *International conference on web-age information management*, pages 25–36. Springer, 2010.

[168] Neo4J Developers. Neo4j. *Graph NoSQL Database [online]*, 2012.

[169] Michael Stonebraker. Newsql: An alternative to nosql and old sql for new oltp apps. *Communications of the ACM. Retrieved*, pages 07–06, 2012.

[170] Katarina Grolinger, Wilson A Higashino, Abhinav Tiwari, and Miriam AM Capretz. Data management in cloud environments: Nosql and newsql data stores. *Journal of Cloud Computing: advances, systems and applications*, 2(1):22, 2013.

[171] Barbara Brynko. Nuodb: Reinventing the database. *Information Today*, 29(9):9–9, 2012.

[172] Michael Stonebraker and Ariel Weisberg. The voltdb main memory dbms. *IEEE Data Eng. Bull.*, 36(2):21–27, 2013.

[173] James C Corbett, Jeffrey Dean, Michael Epstein, Andrew Fikes, Christopher Frost, Jeffrey John Furman, Sanjay Ghemawat, Andrey Gubarev, Christopher Heiser, Peter Hochschild, et al. Spanner: Google's globally distributed database. *ACM Transactions on Computer Systems (TOCS)*, 31(3):8, 2013.

[174] Michele Orru, Riccardo Paolillo, Andrea Detti, Giulio Rossi, and Nicola Blefari Melazzi. Demonstration of opengeobase: the icn nosql spatio-temporal database. In *Local and Metropolitan Area Networks (LANMAN), 2017 IEEE International Symposium on*, pages 1–2. IEEE, 2017.

[175] Belén Vela, José María Cavero, Paloma Cáceres, Almudena Sierra-Alonso, and Carlos E Cuesta. Using a nosql graph oriented database to store accessible transport routes. In *EDBT/ICDT Workshops*, pages 62–66, 2018.

[176] Changqing Ji, Yu Li, Wenming Qiu, Uchechukwu Awada, and Keqiu Li. Big data processing in cloud computing environments. In *Pervasive Systems, Algorithms and Networks (ISPAN), 2012 12th International Symposium on*, pages 17–23. IEEE, 2012.

[177] Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Suresh Anthony, Hao Liu, Pete Wyckoff, and Raghotham Murthy. Hive: a warehousing solution over a map-reduce framework. *Proceedings of the VLDB Endowment*, 2(2):1626–1629, 2009.

[178] Rajiv Ranjan. Streaming big data processing in datacenter clouds. *IEEE Cloud Computing*, 1(1):78–83, 2014.

[179] Apache Storm. Storm, distributed and fault-tolerant realtime computation. *Google Scholar*, 2013.

[180] KMM Thein. Apache kafka: Next generation distributed messaging system. *International Journal of Scientific Engineering and Technology Research*, 3(47):9478–9483, 2014.

[181] Leonardo Neumeyer, Bruce Robbins, Anish Nair, and Anand Kesari. S4: Distributed stream computing platform. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pages 170–177. IEEE, 2010.

[182] Kevin Rector. Mta real-time bus data 'hacked,' offered on private mobile application, 2015. URL `http://www.baltimoresun.com/business/bs-bz-mta-tracker-hack-20150224-story.html`.

[183] The Guardian. Ransomware attack on san francisco public transit gives everyone a free ride, 2016. URL `https://www.theguardian.com/technology/2016/nov/28/passengers-free-ride-san-francisco-muni-ransomeware`.

[184] Omer Tene and Jules Polonetsky. Privacy in the age of big data: a time for big decisions. *Stan. L. Rev. Online*, 64:63, 2011.

[185] Paul M Schwartz and Daniel J Solove. The pii problem: Privacy and a new concept of personally identifiable information. *NYUL rev.*, 86:1814, 2011.

[186] Erika McCallister, Timothy Grance, and Karen A Scarfone. Guide to protecting the confidentiality of personally identifiable information (pii). Technical report, 2010.

[187] Neustar Research. Riding with the Stars: Passenger Privacy in the NYC Taxicab Dataset. `https://research.neustar.biz/2014/09/15/riding-with-the-stars-passenger-privacy-in-the-nyc-taxicab-dataset/`, 2018. Accessed: 2018-05-14.

[188] Ville de Montréal. Montreal's Open Data Policy. `http://donnees.ville.montreal.qc.ca/portail/city-of-montreal-open-data-policy/`, 2018. Accessed: 2018-05-14.

[189] Gabriel Ghinita, Panagiotis Karras, Panos Kalnis, and Nikos Mamoulis. Fast data anonymization with low information loss. In *Proceedings of the 33rd international conference on Very large data bases*, pages 758–769. VLDB Endowment, 2007.

[190] Graham Cormode and Divesh Srivastava. Anonymized data: generation, models, usage. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 1015–1018. ACM, 2009.

[191] Zachary Patterson, Kyle Fitzsimmons, Michael Widener, Jessica Reid, and David Hammond. Designing smartphone travel surveys: Recruitment, burden, incentives and participation. Submitted to the *Journal of Urban Management*, 2018. Submitted: May, 2018.

[192] Zachary Patterson. MTL Trajet 2016. Paper presented at the 11th International Conference on Travel Survey Methods, Esterel, Quebec, 2017. Available at: itinerum.ca/documents.html.

[193] D. Lopez and Bilal Farooq. A blockchain framework for smart mobility. Submitted to the *Blockchain Technology Symposium (BTS'18) - from Hype to Reality, The Fields Institute, Toronto*, September, 2018.

[194] Omar Hasan, Lionel Brunie, Elisa Bertino, and Ning Shang. A decentralized privacy preserving reputation protocol for the malicious adversarial model. *IEEE Transactions on Information Forensics and Security*, 8(6):949–962, 2013.

[195] Yehida Lindell. Secure multiparty computation for privacy preserving data mining. In *Encyclopedia of Data Warehousing and Mining*, pages 1005–1009. IGI Global, 2005.

[196] Mehmet Ercan Nergiz, Maurizio Atzori, and Yucel Saygin. Towards trajectory anonymization: a generalization-based approach. In *Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS*, pages 52–61. ACM, 2008.

[197] Roberto J Bayardo and Rakesh Agrawal. Data privacy through optimal k-anonymization. In *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, pages 217–228. IEEE, 2005.

[198] Gagan Aggarwal, Tomás Feder, Krishnaram Kenthapadi, Rajeev Motwani, Rina Panigrahy, Dilys Thomas, and An Zhu. Anonymizing tables. In *International Conference on Database Theory*, pages 246–258. Springer, 2005.

[199] Panos Kalnis, Gabriel Ghinita, Kyriakos Mouratidis, and Dimitris Papadias. Preventing location-based identity inference in anonymous spatial queries. *IEEE transactions on knowledge and data engineering*, 19(12):1719–1733, 2007.

[200] Gabriel Ghinita, Panos Kalnis, Ali Khoshgozaran, Cyrus Shahabi, and Kian-Lee Tan. Private queries in location based services: anonymizers are not necessary. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 121–132. ACM, 2008.

[201] Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 31–42. ACM, 2003.

[202] Chi-Yin Chow, Mohamed F Mokbel, and Xuan Liu. A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*, pages 171–178. ACM, 2006.

[203] Manolis Terrovitis and Nikos Mamoulis. Privacy preservation in the publication of trajectories. In *Mobile Data Management, 2008. MDM'08. 9th International Conference on*, pages 65–72. IEEE, 2008.

[204] Alastair R Beresford and Frank Stajano. Mix zones: User privacy in location-aware services. In *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, pages 127–131. IEEE, 2004.

[205] Tun-Hao You, Wen-Chih Peng, and Wang-Chien Lee. Protecting moving trajectories with dummies. In *Mobile Data Management, 2007 International Conference on*, pages 278–282. IEEE, 2007.

[206] Baik Hoh and Marco Gruteser. Protecting location privacy through path confusion. In *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*, pages 194–205. IEEE, 2005.

[207] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, et al. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.

[208] Sushil Bhardwaj, Leena Jain, and Sandeep Jain. Cloud computing: A study of infrastructure as a service (iaas). *International Journal of engineering and information Technology*, 2(1): 60–63, 2010.

[209] James Serra. What is the lambda architecture?, 2018. URL `http://www.jamesserra.com/archive/2016/08/what-is-the-lambda-architecture/`.

[210] Angela Orebaugh, Gilbert Ramirez, and Jay Beale. *Wireshark & Ethereal network protocol analyzer toolkit*. Elsevier, 2006.

[211] Michael B Gurstein. Open data: Empowering the empowered or effective data use for everyone? *First Monday*, 16(2), 2011.

[212] David Rider. Toronto and waze app agree to trade traffic data. `https://www.thestar.com/news/city_hall/2017/11/20/toronto-and-waze-app-agree-to-trade-traffic-data.html/`, 2017. Accessed: 2018-07-28.

[213] Uber. Uber Movement. `https://movement.uber.com/`, 2018. Accessed: 2018-06-12.

[214] John Krumm. A survey of computational location privacy. *Personal and Ubiquitous Computing*, 13(6):391–399, 2009.

[215] Ourania Kounadi and Michael Leitner. Spatial information divergence: Using global and local indices to compare geographical masks applied to crime data. *Transactions in GIS*, 19 (5):737–757, 2015.

[216] William B Allshouse, Molly K Fitch, Kristen H Hampton, Dionne C Gesink, Irene A Doherty, Peter A Leone, Marc L Serre, and William C Miller. Geomasking sensitive health data and privacy protection: an evaluation using an e911 database. *Geocarto international*, 25(6): 443–452, 2010.

[217] Kristen H Hampton, Molly K Fitch, William B Allshouse, Irene A Doherty, Dionne C Gesink, Peter A Leone, Marc L Serre, and William C Miller. Mapping health data: improved privacy protection with donut method geomasking. *American journal of epidemiology*, 172(9):1062–1069, 2010.

[218] Miguel E Andrés, Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 901–914. ACM, 2013.

[219] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Marco Stronati. Location privacy via geo-indistinguishability. *ACM SIGLOG News*, 2(3):46–69, 2015.

[220] MTL Trajet. Mtl trajet, 2018. URL `https://ville.montreal.qc.ca/mtltrajet/en/`. last accessed on July 27, 2018.

[221] Su Zhang, Scott M Freundschuh, Kate Lenzer, and Paul A Zandbergen. The location swapping method for geomasking. *Cartography and Geographic Information Science*, 44(1): 22–34, 2017.

[222] Osman Abul, Francesco Bonchi, and Mirco Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 376–385. Ieee, 2008.

[223] Changsha Ma and Chang Wen Chen. Nearby friend discovery with geo-indistinguishability to stalkers. *Procedia Computer Science*, 34:352–359, 2014.

[224] Reynold Cheng, Yu Zhang, Elisa Bertino, and Sunil Prabhakar. Preserving user location

privacy in mobile data management infrastructures. In *International Workshop on Privacy Enhancing Technologies*, pages 393–412. Springer, 2006.

[225] Bhuvan Bamba, Ling Liu, Peter Pesti, and Ting Wang. Supporting anonymous location queries in mobile environments with privacygrid. In *Proceedings of the 17th international conference on World Wide Web*, pages 237–246. ACM, 2008.

[226] Shen-Shyang Ho and Shuhua Ruan. Differential privacy for location pattern mining. In *Proceedings of the 4th ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, pages 17–24. ACM, 2011.

[227] Christopher A Cassa, Shaun J Grannis, J Marc Overhage, and Kenneth D Mandl. A context-sensitive approach to anonymizing spatial surveillance data: impact on outbreak detection. *Journal of the American Medical Informatics Association*, 13(2):160–165, 2006.

[228] Gabriel Ghinita, Keliang Zhao, Dimitris Papadias, and Panos Kalnis. A reciprocal framework for spatial k-anonymity. *Information Systems*, 35(3):299–314, 2010.

[229] Nicolás E Bordenabe. *Measuring privacy with distinguishability metrics: Definitions, mechanisms and application to location privacy*. PhD thesis, Ph. D. dissertation, École Polytechnique, Palaiseau, France, 2014.

[230] GIS Quantum. Development team.(2013). quantum gis geographic information system. open source geospatial foundation project, 2013.

[231] Marc Serre. The university of north carolina, bayesian maximum entropy lab for space/time geostatistics in exposure, disease and risk mapping. `http://www.unc.edu/depts/case/BMElab/`, 2018. Accessed: 2018-07-28.

[232] Vincent Primault, Sonia Ben Mokhtar, Cédric Lauradoux, and Lionel Brunie. Differentially private location privacy in practice. *arXiv preprint arXiv:1410.7744*, 2014.

[233] Simon Oya, Carmela Troncoso, and Fernando Pérez-González. Is geo-indistinguishability what you are looking for? In *Proceedings of the 2017 on Workshop on Privacy in the Electronic Society*, pages 137–140. ACM, 2017.

[234] Robert Tanton et al. A review of spatial microsimulation methods. *International Journal of Microsimulation*, 7(1):4–25, 2014.

[235] Kirk Harland, Alison Heppenstall, Dianna Smith, and Mark H Birkin. Creating realistic synthetic populations at varying spatial scales: A comparative critique of population synthesis techniques. *Journal of Artificial Societies and Social Simulation*, 15(1), 2012.

[236] Bilal Farooq and Eric J Miller. Towards integrated land use and transportation: A dynamic disequilibrium based microsimulation framework for built space markets. *Transportation research part A: policy and practice*, 46(7):1030–1053, 2012.

[237] Justin Ryan, Hanna Maoh, and Pavlos Kanaroglou. Population synthesis: Comparing the major techniques using a small, complete population of firms. *Geographical Analysis*, 41(2): 181–203, 2009.

[238] Stephen E Fienberg and Michael M Meyer. Iterative proportional fitting. Technical report, Carnegie-Mellon University, Pittsburgh PA, 1981.

[239] Kirill Müller and Kay W Axhausen. Population synthesis for microsimulation: State of the art. *Arbeitsberichte Verkehrs-und Raumplanung*, 638, 2010.

[240] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[241] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537, 2011.

[242] Gobinda G Chowdhury. Natural language processing. *Annual review of information science and technology*, 37(1):51–89, 2003.

[243] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9 (8):1735–1780, 1997.

[244] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016.

[245] Stanislav S Borysov, Jeppe Rich, and Francisco C Pereira. How to generate micro-agents? a deep generative modeling approach to population synthesis. *Transportation Research Part C: Emerging Technologies*, 106:73–97, 2019.

[246] Hillel Bar-Gera, Karthik Konduri, Bhargava Sana, Xin Ye, and Ram M Pendyala. Estimating survey weights with multiple constraints using entropy optimization methods. In *88th Annual Meeting of the Transportation Research Board, Washington, DC*, 2009.

[247] A Daly. Prototypical sample enumeration as a basis for forecasting with disaggregate models. In *Transportation planning methods. Proceedings of European transport conference*, volume P423, pages 14–18, 1998.

[248] W Edwards Deming and Frederick F Stephan. On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *The Annals of Mathematical Statistics*, 11(4):427–444, 1940.

[249] Jessica Y Guo and Chandra R Bhat. Population synthesis for microsimulating travel behavior. *Transportation Research Record*, 2014(1):92–101, 2007.

[250] Lijun Sun and Alexander Erath. A bayesian network approach for population synthesis. *Transportation Research Part C: Emerging Technologies*, 61:49–62, 2015.

[251] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[252] David Berthelot, Thomas Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[253] Melvin Wong and Bilal Farooq. A bi-partite generative model framework for analyzing and simulating large scale multiple discrete-continuous travel behaviour data. *Transportation Research Part C: Emerging Technologies*, 110:247–268, 2020.

[254] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F Stewart, and Jimeng

Sun. Generating multi-label discrete patient records using generative adversarial networks. *arXiv preprint arXiv:1703.06490*, 2017.

[255] Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data synthesis based on generative adversarial networks. *Proceedings of the VLDB Endowment*, 11(10):1071–1083, 2018.

[256] Ferenc Huszár. How (not) to train your generative model: Scheduled sampling, likelihood, adversary? *arXiv preprint arXiv:1511.05101*, 2015.

[257] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179, 2015.

[258] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

[259] John F Nash et al. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.

[260] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

[261] Nitish Srivastava and Russ R Salakhutdinov. Multimodal learning with deep boltzmann machines. In *Advances in neural information processing systems*, pages 2222–2230, 2012.

[262] Shenlong Wang, Lei Zhang, Yan Liang, and Quan Pan. Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2216–2223. IEEE, 2012.

[263] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. 2011.

[264] Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines vinod nair. 2010.

[265] Cheng Guo and Felix Berkhahn. Entity embeddings of categorical variables. *arXiv preprint arXiv:1604.06737*, 2016.

[266] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1587–1596. JMLR. org, 2017.

[267] Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. Adversarial feature matching for text generation. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 4006–4015. JMLR. org, 2017.

[268] William Fedus, Ian Goodfellow, and Andrew M Dai. Maskgan: better text generation via filling in the_. *arXiv preprint arXiv:1801.07736*, 2018.

[269] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[270] Agence métropolitaine de transport. Enquête origine-destination 2013: La mobilité des personnes dans la région de Montréal - Faits Saillants. Technical report, Agence métropolitaine de transport, Montréal, ND.

[271] Kedar Potdar, Taher S Pardawala, and Chinmay D Pai. A comparative study of categorical variable encoding techniques for neural network classifiers. *International Journal of Computer Applications*, 175(4):7–9, 2017.

[272] Ekaba Bisong. Introduction to scikit-learn. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, pages 215–229. Springer, 2019.

[273] Google. S2 Geometry. `http://s2geometry.io/`, 2019. Accessed: 2019-07-29.

[274] Müller Kirill and Kay W Axhausen. Population synthesis for microsimulation: State of the art. In *Transportation Research Board 90th Annual Meeting*, 2011.

[275] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[276] Boris Hanin. Which neural net architectures give rise to exploding and vanishing gradients? In *Advances in Neural Information Processing Systems*, pages 582–591, 2018.

[277] Donald B Johnson. A note on dijkstra's shortest path algorithm. *Journal of the ACM (JACM)*, 20(3):385–388, 1973.

[278] Statistics Canada. Montreal Road Network, 2020. URL `https://carto.com/dataset/montreal_road_network`.

[279] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*, 2017.

[280] Sanjeev Arora and Yi Zhang. Do gans actually learn the distribution? an empirical study. *arXiv preprint arXiv:1706.08224*, 2017.

[281] David López and Bilal Farooq. A multi-layered blockchain framework for smart mobility data-markets. *Transportation Research Part C: Emerging Technologies*, 111:588–615, 2020.

[282] Danqing Zhang, Junyu Cao, Sid Feygin, Dounan Tang, Zuo-Jun Max Shen, and Alexei Pozdnoukhov. Connected population synthesis for transportation simulation. *Transportation research part C: emerging technologies*, 103:1–16, 2019.

[283] Ismaïl Saadi, Ahmed Mustafa, Jacques Teller, Bilal Farooq, and Mario Cools. Hidden markov model-based population synthesis. *Transportation Research Part B: Methodological*, 90:1–21, 2016.

[284] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[285] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019(1):133–152, 2019.

[286] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321, 2015.

[287] Mahawaga Arachchige Pathum Chamikara, Peter Bertok, Ibrahim Khalil, Dongxi Liu, and Seyit Camtepe. Local differential privacy for deep learning. *arXiv preprint arXiv:1908.02997*, 2019.

[288] S LaRon, R Beckman, K Baggerly, D Anson, and M Williams. Transims transportation analysis and simulation system project summary and status. *NASA Open Source Agreement Version*, 1, 1996.

[289] Shiliang Sun, Changshui Zhang, and Guoqiang Yu. A bayesian network approach to traffic flow forecasting. *IEEE Transactions on intelligent transportation systems*, 7(1):124–132, 2006.

[290] Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*, 2013.

[291] Sidi Lu, Yaoming Zhu, Weinan Zhang, Jun Wang, and Yong Yu. Neural text generation: Past, present and beyond. *arXiv preprint arXiv:1803.07133*, 2018.

[292] Ofir Press, Amir Bar, Ben Bogin, Jonathan Berant, and Lior Wolf. Language generation with recurrent generative adversarial networks without pre-training. *arXiv preprint arXiv:1706.01399*, 2017.

[293] NhatHai Phan, Xintao Wu, and Dejing Dou. Preserving differential privacy in convolutional deep belief networks. *Machine learning*, 106(9-10):1681–1704, 2017.

[294] Geoffrey E Hinton. How neural networks learn from experience. *Scientific American*, 267 (3):144–151, 1992.

[295] Geoffrey E Hinton. Learning multiple layers of representation. *Trends in cognitive sciences*, 11(10):428–434, 2007.

[296] Cynthia Dwork. Differential privacy. *Encyclopedia of Cryptography and Security*, pages 338–340, 2011.

[297] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103. IEEE, 2007.

[298] Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. In *Advances in neural information processing systems*, pages 289–296, 2009.

[299] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[300] Yu-nong Zhang, Lu Qu, Jun-Wei Chen, Jin-Rong Liu, and Dong-sheng Guo. Weights and structure determination method of multiple-input sigmoid activation function neural network. *Application Research of Computers*, 29(11):4113–4116, 2012.

[301] M Shahidur Rahman et al. Towards optimal convolutional neural network parameters for bengali handwritten numerals recognition. In *2016 19th International Conference on Computer and Information Technology (ICCIT)*, pages 431–436. IEEE, 2016.

[302] Antonio Gulli and Sujit Pal. *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[303] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*, 2018.

[304] Ekaba Bisong. *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Springer, 2019.

[305] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016.

[306] Alex Graves. Long short-term memory. In *Supervised sequence labelling with recurrent neural networks*, pages 37–45. Springer, 2012.

[307] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[308] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Gan-leaks: A taxonomy of membership inference attacks against gans. *arXiv preprint arXiv:1909.03935*, 2019.

[309] William McGinnis, Chapman Siu, S Andre, and Hanyu Huang. Category encoders: a scikit-learn-contrib package of transformers for encoding categorical data. *Journal of Open Source Software*, 3(21):501, 2018.

[310] Holger Orup. On-the-fly one-hot encoding of leading zero count, October 26 1999. US Patent 5,974,432.

[311] Peilu Wang, Yao Qian, Frank K Soong, Lei He, and Hai Zhao. Word embedding for recurrent neural network based tts synthesis. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4879–4883. IEEE, 2015.

Appendix A

Study Area

The map in Figure A.1 shows the geographical extent of the study area. The map states the boundaries of the census metropolitan areas within the Greater Montreal Area.

Maps showing the geographical suburbs of the Greater Montreal Area (Study Region).

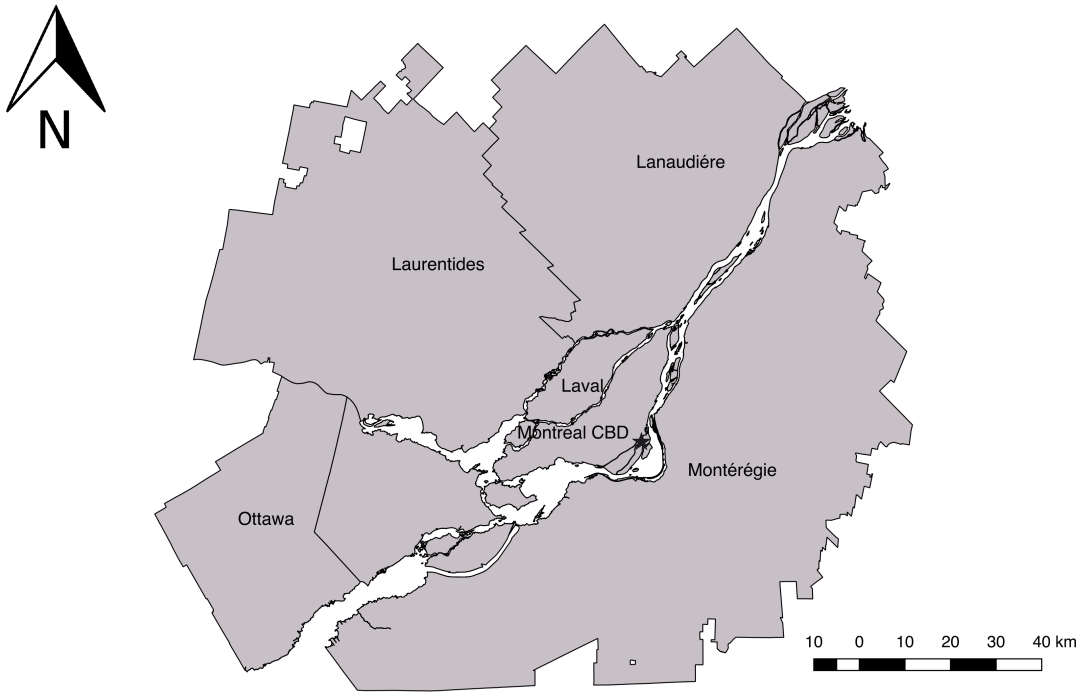


Figure A.1: Map of geographic areas of the Greater Montreal Area

Appendix B

Data Preparation

In this section, we discuss the procedures that were adopted to prepare the data for the generative modelling. The trip data for this project composed on numerical, categorical and location geographic variables as well as location sequences.

B.0.1 Numerical attributes

The objective of processing numerical attributes is to normalize and scale within range of -1 to 1. The approaches of Scaling and Normalization standardizes numeric inputs into data points that are suitable for Neural Networks. Standardizing data points transform that into a resulting distribution with a mean of 0 and a standard deviation of 1. Normalization is defined by:

$$x^{|} = \frac{x - x_{mean}}{x_{max} - x_{min}} \tag{7}$$

where X is the feature vector, X_{mean} is the mean of the feature vector, X_{min} is the minimum of the feature vector and X_{max} is the maximum of the feature vector. We implemented the normalization using the Scikit-learn [309] Pre-processing framework available in Python. The package presents two libraries: MinMaxScaler and StandardScaler. The MinMaxScaler library normalizes a feature to range of 0 to 1 while the StandardScaler library standardizes the data points to a mean of 0.

B.0.2 Categorical attributes

When processing categorical attributes, we consider two categories namely low and high cardinality. Low cardinality refers to variables with a minimum of 20 unique variables while High cardinality referes to variables with 20 or more unique variables. For low cardinal variables, we apply the one-hot encoding technique. One-hot encoding [310] converts categorical variables to binary combinations of values with a single high (1) bit and all the others low (0). This encoding technique derives an integer representation for category values with a length of the encoded vectors equivalent to the number of unique values of the variable. This technique becomes inefficient when implemented on high categorical values since larger matrices are created with a drawback on computation. On the other hand, we employ feature embeddings [265, 311] to encode high cardinal values to fixed dimensional real values. Feature embeddings derive unique real-valued vectors to represent each category. We employ Keras layer embeddings for generation of feature embeddings for high cardinal categories.

B.0.3 Route Itinerary

For the purposes of trip sequences, the model demand complete route itineraries between origin and destination geographic points. The travel routes were generated with the shortest distance path between an origin and a destination data points. The Open Source Routing Machine (OSRM) allows a public accessible Application Programming Interface (API) available at http://project-osrm.org. The API endpoint returns a sequence of geographic points stating the complete geographical route itinerary.