UNIVERSITY OF CALIFORNIA
SANTA CRUZ

UMAD: CLASSIFICATION, GENERATION AND ANALYSIS OF USER MOBILITY AND ACTIVITY DATA.

A dissertation submitted in partial satisfaction
of the requirements for the degree of
DOCTOR OF PHILOSOPHY
in
COMPUTER SCIENCE
by

Sinjoni Mukhopadhyay King

March 2023

The Dissertation of Sinjoni Mukhopadhyay King
is approved:

Professor Katia Obraczka, chair

Professor Faisal Nawab

Professor Yang Liu

Peter Biehl

Vice Provost and Dean of Graduate Studies

To my near and dear ones.

Contents

List of Figures

List of Tables

List of Acronyms

uMA User Mobility and ActivityHuman movement and activities like travelling, communication, network usage, health patterns etc.

uMAD User Mobility and Activity Data. Datasets containing uMA data

CLI Command Line Interface. A text-based user interface used to run programs, manage computer files and interact with the computer.

DGM Deep generative models Refer to unsupervised machine learning models, usually a combination of generative models with neural networks.

HMM Hidden Markov ModelA class of probabilistic graphical model that allow us to predict a sequence of unknown or hidden variables from a set of observed variables.

RBM Restricted Boltzmann MachinesStochastic two layered neural networks categorized under energy based models that can detect inherent patterns automatically in the data by reconstructing the input

VAE Variational AutoencodersIn consists of an encoder, a decoder, and a loss function. The encoder is a neural network that compresses data into a latent space while the decoder reconstructs the data given the hidden representation.

GAN Generative Adversarial NetworkGenerative models that are able to produce or generate new content from noise.

GPS Global Positioning System satellite-based radionavigation system owned by the United States government and operated by the United States Space Force.

Acknowledgments

This endeavor would not have been possible without my advisors Katia Obraczka and Faisal Nawab. I am grateful for their enthusiasm and constant encouragement towards the project. I could not have completed any of this without their ongoing patience and continuous guidance. Thanks to my committee member Yang Liu for all his valuable feedback on my research. I would like to express my deepest appreciation to the members of Baskin School of Engineering Graduate Advising and the Department of Computer Science and Engineering, especially Roberto, Alicia, Theo for making the entire process so smooth for me. I am deeply indebted to my husband Justin for patiently dealing with all my moods, while encouraging me to cross the finish line. I would like to extend my sincere thanks to both sets of my parents. Nina and Saibal thank you for always providing for me, never asking for anything in return, and for believing in me. Lynn and Robin thank you for your support and for cheering me on throughout the process. Special thanks to my friends Akash, Ushnik, Oceane, Kenneth, Yash and Matt for being my constant support over the last 6 years and for all the extremely useful life advice that I have received from all of you! Lastly I would like to acknowledge all my other wonderful friends and family for their constant encouragement, without whose support this high pressure academic journey would feel incomplete.

Abstract of the dissertation

UMAD: CLASSIFICATION, GENERATION AND ANALYSIS OF USER MOBILITY AND ACTIVITY DATA.

by
Sinjoni Mukhopadhyay King
University of California, Santa Cruz
March 2023

Access to user mobility and activity data (uMAD) is crucial for researchers and practitioners in various areas of technology and infrastructure planning. It reveals a number of aspects of user behavior and trends at different spatio-temporal scales which in turn provide invaluable information to guide the design, operation, and management of critical infrastructure, services and applications. However, previous academic/industry efforts to collect user mobility and activity (uMA) information face important challenges raised by issues such as uMA data diversity, privacy and protection concerns. Consequently, even if uMA data is collected successfully, it cannot be generalized and/or shared publicly. To address these challenges, there has been significant work on the generation of synthetic uMA datasets as well as work on data anonymization. Prior work in these areas, however, target specific applications and datasets, and thus make it harder to generalize them for use across different scenarios.

Our aim is to fill these gaps by providing an uMA ecosystem that manages classification, generation, evaluation and analysis. As part of this goal, our pipeline uMAD aims to include the following features: ***Classification***: enabling existing or new uMA data to be classified into our proposed taxonomy buckets; ***Generation***: allowing users to capture patterns and generate realistic uMA datasets by leveraging well known Machine learning generation models like Generative Adversarial Networks (GANs); ***Trace Analysis***: helping users analyze and visualize patterns in existing and new uMA datasets; and ***Model Analysis*** providing users with a broad understanding of the ML model resource consumption and parameters. uMAD's open source command line interface (CLI) is eventually meant to generate realistic synthetic uMA datasets that mimic existing traces for a range of user-configurable parameters and provide users with existing datasets that can be selected based on the users' specific needs.

Chapter 1

Introduction

According to a recent Mobile Data Traffic Outlook report by Ericsson [106], global mobile data traffic reached a total of 58EB per month at the end of 2020 and is forecast to exceed 300EB per month in 2026. To be able to cope with this unprecedented growth and still be able to provide adequate service to users demands a deeper understanding of how users move, connect, as well as generate and consume data. Understanding *user mobility and activity* in access networks—including wireless and fixed broadband—is essential to be able to scale and accommodate future connectivity and traffic demands as well as design systems and applications that are able to adapt to user mobility and activity patterns. Furthermore, better understanding mobile user behavior and activity can also greatly contribute to improve urban planning, such as transit, transportation, housing infrastructure, and emergency response (including public health emergency situations like the COVID-19 pandemic), as well as other services such as shopping, entertainment, and others.

The importance of understanding user mobility and activity (uMA) data has led to extensive efforts from academia and industry to collect [211, 212], analyze [127, 199, 216, 187], and synthetically generate [203, 128] such data with the purpose of studying it and enable further exploration. These prior efforts, however, face some significant challenges as follows: **(1) Diversity of uMA data:** uMA data is extremely diverse such that small changes in the environment—e.g., the driving application, collection technology, geographic location, user demographics—can trigger significant change in the data. For example, existing taxi traces from Rome focus on temporal patterns using timestamps to categorize locations

visited based on dates, while taxi traces collected in San Francisco focus on spatial patterns categorizing locations based on unique cab identifiers. This challenge places considerable burden in efforts that aim to improve access, analysis, and generation of uMA data. On one hand, real datasets will likely not apply to different scenarios, but, on the other hand, uMA synthetic data generation tools rely on the user to configure the activity and mobility patterns that are not typically known, which lead to imprecision and inaccuracies [161]. **(2) Storage and Computation Costs:** uMA datasets are typically large consisting of millions of datapoints and their sizes can be in the range of Gigabytes. This makes storing them reliably a challenge and also raises computation costs. **(3) Data privacy and protection:** uMA data is typically personal identifiable information (PII). This means that collecting, storing, processing, and sharing such data is restricted due to data privacy and protection regulations such as GDPR, CCPA, and HIPAA. This places considerable challenges in the whole pipeline of uMA data collection, generation, and exploration. User consent is needed to collect, store, and process uMA data, and sharing uMA data needs to be performed in a manner that does not reveal any PII data. This has significantly constrained access and collection of uMA data and has led organizations that have uMA data to not be able to share it without anonymization and summarization techniques that limit the information that can be shared and thus insights that can be derived. **(4) Absence of a holistic approach:** To our knowledge, there is no current tool that integrates uMA data collection, storage, generation, and analysis. Most solutions cover only one or at most two of these services, leaving researchers and practitioners who use uMA data with the arduous and often tedious task of having to piecemeal different solutions together.

To overcome the above challenges, we propose a holistic, end-to-end tool, called **uMAD**, that will eventually manage the collection, storage, analysis and generation of uMA data. uMAD provides, in a single, integrated pipeline, uMA data collection, storage, generation, evaluation, and analysis capabilities.

1. *Collection:* uMAD's Collection component will provide methods to enable users to add new uMA datasets to the framework for future study and analysis. This can be accomplished by adding new datasets that have already been collected and processed. Also, we will develop end-user and device collection software that can be deployed on

various platforms, including mobile and IoT devices to collect and report uMA data in real time or offline. uMAD's Collection component is augmented with solutions for data cleaning to help overcome the inaccuracies and errors that are common in raw uMA data. It will also include mechanisms to ensure collected uMA data adheres to data privacy and protection policies and regulations while mitigation performance overhead introduced by privacy regulation compliance.

2. *Storage:* The uMAD pipeline will also handle the storage of collected datasets for future access and querying. An important feature of the storage component is to enable annotating datasets to include various features/labels such as mobility mode, collection infrastructure, and measurement medium. These features can be used to help users select datasets that match their interests and needs. Furthermore, as discussed below and later in the proposal, these features will also be used by uMAD's synthetic data generation and analysis components. Some features can be derived automatically from the collected dataset—such as statistics about the dataset and collected items. Other features, however, cannot be easily inferred from the data and requires domain knowledge. For example, the demographics of data subjects might not be part of the dataset, but can be annotated by the domain expert who deployed and conducted the collection.

3. *ML-based generation:* One of uMAD's core components is its uMA data generation stage which handles the generation of synthetic datasets. In uMAD, we aim to overcome the challenges faced by existing synthetic dataset generation solutions. In particular, uMAD's synthetic data generation allows users to control and specify the desired features and parameters of the generated dataset. For example, a user may choose the type of subject demographics, geographic location, collection infrastructure, and other features and parameters that are common in existing datasets. New features and parameters can also be added as needed. Additionally, uMAD synthetic dataset generation contains methods to ensure compliance with data privacy and protection regulations. uMAD is also able to flexibly control the features and parameters of generated datasets by utilizing a machine learning (ML) pipeline that we will design and develop. This pipeline learns the features and parameters of uMA datasets that

have been stored and collected previously (including a bank of datasets that we have collected from public sources).

After learning the features and parameters of selected uMA datasets, the ML pipeline utilizes generative ML models that allow generating datasets that mimic real traces based on the parameters that the user specifies. For example, consider a user interested in a realistic dataset that has properties A and B. If no real- or synthetic dataset has both properties A and B, then the user will have to rely on ”suboptimal” datasets, i.e., datasets that do not completely match the user's needs. To address this limitation, uMAD automatically learns features A and B from the pool of existing datasets, in which there exists datasets that either have feature A or B. Then, using the generative model, datasets containing both features plus all other realistic patterns of the datasets would be generated. The flexibility and parametrization capabilities of uMAD's ML-based dataset generation are especially beneficial when users are interested in a large number of features and testing them with different parameters.

4. *Analysis:* uMAD's Analysis component will be used to perform various analytics on either real- or synthetic datasets, including different types of pre-processing, visualization, trend analysis, similarity studies, and anomaly detection. It will also conduct evaluation of the generated traces according to relevant performance metrics. We will augment and utilize the wealth of tools and solutions that provide such functionalities and adapt them to uMA data and analysis.

For the purpose of this thesis we will focus briefly on Number 1, where we propose a taxonomy to classify existing open source traces, that can be extended in the future to include newer traces as well; and then heavily focus on Numbers 3 and 4, where we create a detailed analysis report for the generated datasets, comparing them against the ground truth, and the models that are being used. More detailed analysis of Numbers 1 and 2 will be studied as part of future work.

1.1 uMA data collection, classification and analysis tools

A number of surveys have outlined challenges raised by different aspects of uMA datasets, notably: collection and analysis of uMA traces [122, 223, 70, 225, 236, 207], location-based pattern prediction [250, 268, 196, 271, 114, 257], and wireless/mobile networking issues and their effect on human mobility [73, 249, 135]. Additionally, advances in positioning and localization technologies like GPS, cellular radio tower and WiFi have enabled collection of larger feature representations of UMA traces, which in turn have motivated feature analysis techniques like vector analysis [88, 55], feature granularity studies [189, 221, 154, 251], spatio-tremporal characterisation [170, 138, 175, 179], and machine learning based feature analysis [105, 243, 118, 191]. New developments in machine learning have enabled a host of new models for uMA trace creation [193, 111, 209, 224], analysis [110, 134], prediction [256, 240, 185, 210], and pattern recognition based personalization [262, 206, 115].

Even though there has been considerable work on uMA dataset services, there are still critical gaps that need to be addressed. More specifically, current approaches do not provide a general, holistic, end-to-end solution that focuses on the entire uMA dataset lifecycle. Our project aims to bridge this gap by developing from the ground up a simple, extensible, integrated tool that encompasses the basic stages of the uMA dataset pipeline, namely collection, storage, creation, and analysis.

Existing uMA traces can be loosely classified into 4 categories: location, lifestyle, connectivity/networking, movement and health, with applications ranging from communication, urban planning, vehicular and network traffic analysis, social/community services (e.g., community centers, libraries, shopping and entertainment, etc). ***Location*** traces typically include latitude-longitude information which when combined with *UMA tags* like user/vehicle identifiers, can provide insight into frequency of visits to a location, population density along different times of the day/year, popular locales in a specific region etc. When combined with connectivity-based tags, location traces can help identify optimal placement of WiFi access points, help manage network traffic, etc. ***Lifestyle or Geosocial*** traces use check-in information, which reveals a personality of a neighborhood in a city, for urban planning and retail real estate investments. Geosocial traces can also be used to target online applications

to specific communities of people and be applied for marketing, merchandising and consumer goods. ***Connectivity and Network*** traces use information on signal strengths, packet transfer details, network usage details and timestamps to analyze and improve performance across the network stack. ***Movement and Health*** traces include tracking information from sensors like heart rate monitors, oximeters, accelerators and magnetometers. This information can be powerful in developing customized health solutions specific to different categories of users, classified by pre-determined behaviors. These traces can also be used for indoor and outdoor localization and positioning applications.

The goal of uMAD and this thesis are to provide researchers, practitioners, and citizen scientists with a holistic, end-to-end integrated tool that will manage the entire uMA data life-cycle including:

- *uMA data collection* which allows users to add new datasets (whether collected actively or offline) as training data to our model pipeline.
- *uMA data storage* which contains a database of stored trained generative models that can be manipulated to generate new/old uMA data.
- *uMA data generation* which allows uMAD users to generate synthetic, yet realistic uMA datasets based on the user's needs.
- *uMA dataset analysis and evaluation* including data pre-processing, cleaning, as well as evaluation of generated datasets.

uMAD's integrated pipeline for managing the entire uMA dataset lifecycle has the potential to make significant contributions to Internet measurement research by providing (1) a database of real- and synthetically-generated, yet realistic uMA traces, (2) uMA dataset generation and (3) uMA dataset analysis and evaluation capabilities. uMAD will be made publicly available and accessible through GitHub.

The research and education outcomes of this project will be transformative in several aspects. First, it will support researchers and practitioners in a range of disciplines by providing them with an integrated tool that allows them to collect, access, manipulate, and generate realistic uMA datasets containing features that match their needs and applications. By democratizing access to uMA data, our research will also inform and guide the design,

operation, and management of critical infrastructure (e.g., access networks, power grid, transportation and transit systems) and services (e.g., healthcare, education, community and social services) and, as a result, promote equity and inclusion of under-served communities.

1.2 Thesis chapter overview

Chapter 2 discusses the need and challenges associated with uMA data in more detail, the landscape of uMA applications; and a background of Deep Generative Models with special emphasis on the working, mathematical intuition an challenges associated with Generative Adversarial Networks (GANs). Chapter 3 goes over existing state of the art in the areas of uMA feature analysis, uMA modeling and prediction, uMA Surveys, generative Models for uMA data generation and evolution of public accessibility of uMA datasets. Chapter 4 goes over the taxonomy and its different categories along with classifying existing uMA traces into the taxonomy categories. Chapter 5 goes over the ML based generative models and a comparison of the traces generated by each model, going to select the model with the best fidelity across uMA categories. Chapters 6 and 7 discusses the experimental methodology and results to support our proposed pipeline. Chapter 8 concludes the thesis and points to several future directions in which this research can be applied.

Chapter 2

Background

In today's modern information-rich world, better understanding of human movement and activity has become increasingly essential in various areas such as network, communication provisioning and deployment, urban planning, health care delivery optimization, localizing technology improvements, geo-spatial environmental updates etc. As part of the background we will look at the need and challenges associated with uMA data; elaborate on their landscape and application areas broadly dividing them into (4) umA application categories: Connectivity, Location, Health and Lifestyle; learn about Deep Generative Models and their applications with special focus on Generative Adversarial Networks (GANs).

2.1 Need for uMA traces and its challenges

Several industry sources have publicly acknowledged the growth in the volume of mobile data traffic, i.e., data traffic generated by mobile network and user communication, as well as the different applications of uMA datasets. The Mobile Data Traffic Outlook report by Ericsson [106] forecasts a global mobile data traffic growth from a total of $65EB$ per month at the end of 2021 to $370EB$ per month in 2027. Another published article [3] talks about coping with this unprecedented growth and still being able to provide adequate service to users. A report by Deloitte [23] discusses applying public vehicular mobility analysis results to increase public transport convenience to the point where people consciously choose to use public over personal vehicles. Arity, a company that focuses solely on mobility analysis, has their own blog outlining various applications of both vehicular and user mobility [45].

These reports provide hard evidence that analyzing uMA datasets is vital not only for adequate dimensioning of the underlying network infrastructure but also for a diverse set of applications in different disciplines, such as (1) Access networks, including wireless and fixed broadband which can help scale and accommodate future connectivity and traffic demands as well as design systems and applications that are able to adapt to user mobility and activity patterns; (2) Urban planning such as transit, transportation, housing infrastructure, and emergency response (including public health emergency situations like the COVID-19 pandemic), as well as other services such as shopping, entertainment, and others.

The importance of understanding uMA data has led to extensive efforts from academia and industry to collect [211, 212], analyze [127, 199, 216, 187], and synthetically generate [203, 128] such data with the purpose of studying it and enabling further exploration. These prior efforts, however, face some significant challenges as follows:

- **Diversity in data:** uMA data is diverse to the point where small changes in the environment—e.g., the driving application, collection technology, geographic location, user demographics—can trigger significant change in the data. For example, existing taxi traces from Rome focus on temporal patterns using timestamps to categorize locations visited based on dates, while taxi traces collected in San Francisco focus on spatial patterns categorizing locations based on unique cab identifiers. This challenge places considerable burden in efforts that aim to improve access and analysis of uMA data.

- **Data privacy and protection:** uMA data is typically personal identifiable information (PII). This means that collecting, storing, processing, and sharing such data is restricted due to data privacy and protection regulations such as GDPR, CCPA, and HIPAA. User consent is needed to collect, store, and process this data, and sharing data needs to be performed in a manner that does not reveal any PII data. This significantly constrains access and collection of uMA data and has led organizations to not be able to share it without anonymization and summarization techniques, which in turn limit the information that can be shared and insights that can be derived.

2.2 Landscape of uMA traces

Public uMA traces have a multitude of applications including communication, urban planning, vehicular and network traffic analysis, social management and in some cases even healthcare. Examples of social management would be structuring community needs like library, shopping complexes based on mobility footprints. uMA traces across the above mentioned applications can be loosely classified into 4 categories: location, lifestyle, connectivity/networking, movement and health.

- ***Location Traces*** Traditional GPS traces typically include latitude-longitude information which when combined with *uMA tags* like user/vehicle identifiers, can provide insight into frequency of visits to a location, population density along different times of the day/year, popular locales in a specific region etc. When combined with connectivity-based tags, location traces can help identify optimal placement of WiFi access points, help manage network traffic, etc. Other uMA tags can include any type of information tied to location like behavioural patterns, foot traffic, choices, network usage etc. Vehicular GPS information when combined with timestamps can provide insights on routes taken by specific modes of transportation at different times of the day. Frequency of travel using different modes of transportation at different locations can also be derived, which provide insights into mobility patterns of different communities. For example, if we have a trace that contains information of all buses along with their locations and timestamps for a particular city, we can identify the major hotspots or centrally located spots in the city based on locations that are visited most frequently by buses of different routes. GPS traces when combined with connectivity traces can provide insight into how well-connected some regions are. For example a trace that gives locations and their corresponding RSSI values can help identify placement of WiFi access points, which can in turn help offloading/managing network traffic within a specific location. GPS traces can also provide useful migration information. For example location information combined with unique person IDs, date and timestamps can help us identify patterns in how a person is moving between two locations.

- ***Lifestyle Traces*** Geosocial traces, referred as lifestyle traces in the paper, use check-in information to derive pedestrian/user patterns to aid with urban planning and retail real estate investments. Lifestyle traces can also be used to target online applications to specific communities of people and be applied for marketing, merchandising and consumer goods. The unique style of lifestyle traces data has the potential to reveal the personality of neighborhoods in a city. Building a park near neighborhoods that have a strong healthy living, nature or dog loving segments might be a source of support. Whereas building a shopping center in that same space instead might encounter resistance from the same group of people. Retail business success and geosocial segments are also closely related. A low priced, high traffic region may seem like a good place to build a store, but the most important factor contributing to the success of the store would be the social dynamics of the people around the store. Retail property owners can also use lifestyle data to determine social segments of people around their property, which will help them lease the property to stores that are more likely to do well in the longer term in a particular area. Lifestyle traces can also be used to target online applications to specific communities of people and be applied for marketing, merchandising and consumer goods. We can identify what people are doing and talking about in various locations. This kind of information can help with placement of billboards, local radio spots, or location-targeted mobile advertisements, as it is more effective to advertise in an area which has a social segment that has been predicted to be more receptive to those ideas.As a bonus these traces can be applied in planning healthcare facilities. Such data can be used to identify age group of people in different locations, and based on the age determine if a particular location has more children who require pediatricians, or have an older population who require elder care physicians. We can also use such data to plan other specialized healthcare solutions, like chiropractors for locations where people have more of a sedentary lifestyle e.g. software engineers.

- ***Connectivity and Network Traces*** Connectivity and network traces provide information on signal strengths, packet transfer details, network usage details and timestamps. Mobility performance metrics like user pause probability, user arrival, departure prob-

abilities heavily impacts the performance of 5G cellular networks. Optimizations can be performed by analyzing these metrics [117, 95, 124, 237, 195, 155]. Understanding user mobility characteristics, predicting network usage, can also help determine performance of routing protocols and feasibility of running an application over a vehicular ad hoc network [53]. Caching files based on popularity to reduce pressure on back-haul networks relies on user mobility pattern studies to provision storage allocation [188], model cost optimal device to device networks [95, 124] and improve data offloading [237, 195, 155]. Other applications of connectivity traces include analysis of spatial and temporal properties of pedestrian smart device based mobility datasets to enhance operations of wireless sensor networks [244].

- ***Movement and Health Traces*** Movement and health traces include tracking information from sensors like heart rate monitors, oximeters, accelerators and magnetometers. One important application of these traces is in the healthcare field. For example traces with information about a user's orientation and displacement can be used to predict whether the user is about to fall; this kind of information can be useful to enable independent living for older adults. Another important application of these traces is positioning and localization for users. For example navigation traces collected from sensors over time can help build a map for a particular area, complete with obstacles. This map can later be used for several applications like, video gaming using augmented reality, and accessibility applications like creating navigation tools for mobility-challenged users.

Even though there is a large collection of publicly-available uMA traces (as discussed in our survey paper [143]), challenges raised by uMA data diversity combined with privacy and protection concerns underscore the need for an open-source tool like uMAD to support systematic and integrated uMA data collection, storage, generation and analysis.

2.3 Deep generative models

Deep generative models (DGM) refer to unsupervised machine learning models, usually a combination of generative models with neural networks. Unsupervised data

provides no labels to the model during training and have applications like density estimation, clustering, feature learning, and dimension reduction. DGMs can be used to extract patterns and abnormalities in a data distribution and then create a new instance of the same data. This data instance creation is commonly a result of a combination of: The *Learning* process, where the model aims to minimize the value of some form of a distance metric between the model and the data distributions and; The *Inference* process, where the trained model identifies the likelihood of a model datapoint belonging to the data distribution and the potential generation capacity of the model. Some examples of commonly used generative models are:

1. Restricted Boltzmann Machines (RBMs): Involves learning a probability distribution from an original dataset and using it to make inferences about never before seen data.

2. Variational Autoencoders (VAE): Given a bunch of random variables that can be sampled easily, random data samples following other distributions can be generated, through a complicated non-linear mapping.

3. Hidden Markov Models (HMM): A class of probabilistic graphical model that allow us to predict a sequence of unknown (hidden) variables from a set of observed variables.

4. Generative Adversarial Networks (GANs): Generator tries the best to cheat the discriminator by generating more realistic data, while the Discriminator tries the best to distinguish whether data is generated by a model or not.

Different variations of ***RBMs*** have been used for reconstruction [222] and analysis [199, 133] of time series datasets. ***VAEs*** have been used extensively for pattern recognition [261, 112, 205] and generation [147]. RBM models were an excellent choice for prototyping datasets with smaller feature sets, but were inefficient for larger feature sets. Most previous VAE and HMM models have recently been replaced by ***GANs*** due to the improved resconstruction advantages of the double feedback between GAN's generator and discriminator [107]. To leverage this feature, we will initially use GANs for uMAD's dataset generation stage.

2.4 Generative Adversarial Networks

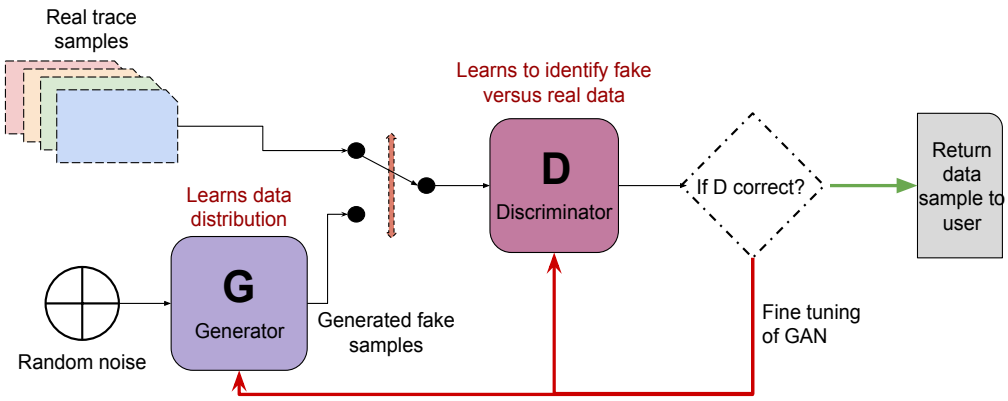

Figure 2.1: GAN basic components: the Discriminator and Generator in adversarial feedback loop with each other.

Generative adversarial networks (GANs) have a simple pretense: if you train a model on all available training data, and train a second model to try to come up with examples that the first model mis-classifies, the second model will eventually converge to produce synthetic data indistinguishable from the data the first model was trained with [126]. For GANs, the models are artificial neural networks, known as the *discriminator* and the *generator*, and they are placed at odds as described in Figure 2.1. In the last five years GANs have been used for different applications [200, 148, 113, 137, 248, 219, 131, 100, 181, 173], all of which are related to either image/video synthesis or emulating human behavior.

2.4.1 Mathematical Intuition

Since GANs are a combined back and forth between two different models: the generator and the discriminator; each model will have its own loss function. The math has already been explained using a binary cross entropy loss function by Jake Tae [217] and a min-max function by Ian Goodfellow [126] in their works and we are adding the explanation here for context to the dissertation. Notations that we will be using throughout this section are:

$$x: \ Real\, data, \ z: \ Latent\, vector, \ G(z): \ Fake\, data,$$

$$D(x): \ Discriminator's\, evaluation\, of\, real\, data,$$

$$D(G(z)) : Discriminator's evaluation of fake data,$$

$$Error(a, b) : Error between a and b$$

The *discriminator's* goal is to label generated samples as **fake** and the true data points as **real** using a loss function. In both cases the unspecific notation for Error can be replaced with a well known loss function.

$$L_D = Error(D(x), 1) + Error(D(G(z)), 0) \qquad (1)$$

The *generator's* goal is to confuse the discriminator as much as possible such that it mislabels generated samples as being **real**, and minimize the difference between the label for real data, and the discriminator's evaluation of the generated fake data.

$$L_G = Error(D(G(z)), 1) \qquad (2)$$

Lets take an example loss function, the binary cross entropy:

$$H(y, \hat{y}) = -\sum y\, log(\hat{y}) + (1 - y) log(1 - \hat{y}) \qquad (3)$$

The Binary cross entropy loss function is appropriate in this case as it measures how different two distributions are in the context of binary classification of determining whether an input data point is real or fake. Applying (3) to the loss functions in (1) and (2) we get (4) and (5):

$$L_D = -\sum_{x \epsilon \chi, z \epsilon \zeta} log(D(x)) + log(1 - D(G(z))) \qquad (4)$$

$$L_G = -\sum_{z \epsilon \zeta} log(D(G(z)) \qquad (5)$$

The two loss functions are used to train the generator and the discriminator such that for the loss function of the generator, the loss is small if D(G(z)) is close to 1, since log(1)=0. Once the loss functions have been defined we mathematically try to solve the optimization problem, that is try to find the parameters for the generator and the discriminator such that the loss functions are optimized. Goodfellow presented the mathematical logic by framing (4) and (5) as a combined min-max game, where the discriminator seeks to maximize the given quantity whereas the generator seeks to achieve the reverse:

$$min_G max_D[log(D(x)) + log(1 - D(G(z)))] \quad (6)$$

Training of GANs is typically performed one model at a time. Assuming the quantity of interests as a function of G and D, the value function V(G,D) can be defined as:

$$V(G, D) = E_{x \sim p_{data}}[log(D(x))] + E_{z \sim p_z}[log(1\ D(G(z)))] \quad (7)$$

Laying more emphasis on the distribution modeled by the generator, p_z can be rewritten as a new variable $y = G(z)$ and use this substitution to rewrite the value function (8):

$$V(G, D) = E_{x p_{data}}[log(D(x))] + E_{y p_g}[log(1 - D(y))]$$

$$= \int_{x \epsilon \chi} p_{data}(x) log(D(x)) + p_g(x) log(1 - D(x)) dx \quad (9)$$

The discriminators task is to maximize (9). Taking a partial derivative of $V(G, D)$ with respect to $D(x)$, we see that the optimal discriminator, denoted as $D^*(x)$, occurs when

$$\frac{p_{data}(x)}{D(x)} - \frac{p_g(x)}{1 - D(x)} = 0 \quad (10)$$

Rearranging (10), we get

$$D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \quad (10)$$

Explaining the intuition behind the new equation (10), for a sample x, if it is very close to real data, we would expect $p_{data}(x)$ to be close to 1 and $p_g(x)$ to be close to 0, in which case the optimal discriminator would assign 1 to that sample. On the other hand, for a generated sample $x = G(z)$, we expect the optimal discriminator to assign a label of 0, since $p_{data}(G(z))$ should be close to 0.

Applying the same value function while training the generator, we assume the discriminator to be fixed. We plug in the result from (10), into the value function:

$$V(G, D^*) = E_{x \sim p_{data}}[log(D^*(x))] + E_{x \sim p_g}[log(1 - D^*(x))]$$

$$= \quad E_{x \sim p_{data}}[log \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}] + E_{x \sim p_g}[log \frac{p_g(x)}{p_{data}(x) + p_g(x)}] \quad (11)$$

The next steps involves a logarithmic manipulation of the equation (11)

$$V(G, D^*) = \quad E_{x \sim p_{data}}[log \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}] + E_{x \sim p_g}[log \frac{p_g(x)}{p_{data}(x) + p_g(x)}]$$

$$= \quad -4 + E_{x\sim p_{data}}[log \quad p_{data}(x) - log\frac{p_{data}(x)+p_g(x))}{2}]$$

$$+E_{x\sim p_g}[log \quad p_g(x) - log\frac{p_{data}(x)+p_g(x))}{2}] \qquad (12)$$

Equation (12) can now be re-interpreted with the Kullback-Leibler Divergence(KLD) [48] equation:

$$V(G, D^*) = -log \quad 4 + D_{KL}(p_{data}||\frac{p_{data}+p_g}{2}) + D_{KL}(p_g||\frac{p_g+p_g}{2}) \qquad (13)$$

Taking into account the Jensen Shannon Divergance(JSD) [46], defined as:

$$J(P, Q) = \frac{1}{2}(D(P||R) + D(Q||R))$$

$$where, R = (P + Q)/2$$

Equation (13) can be rewritten using JSD as:

$$V(G, D^*) = -log4 + 2.D_{JS}(p_{data}||p_g)(14)$$

Equation (14) proves the GAN intuition, where the goal of training the generator is to minimize the value of the function $V(G, D)$, by keeping the value of the JS divergence between the distributions of the real and generated data as small as possible. Both KLD and the JSD are methods to measure similarity between two probability distributions.

2.4.2 Challenges with GANs

There are several challenges associated with GANs. Training a single neural network can be difficult due to the incredible number of choices involved: architecture, activation functions, optimization method, learning rate, and dropout rate, among others. GANs double all of those choices and add new complexities [83]. Both the generator and the discriminator may forget samples they used earlier in their training, which can lead to the two networks getting caught in a stable cycle of solutions that do not improve over time. One network may overpower the other network, such that neither can learn anymore. The generator may not explore much of the possible solution space, only enough of it to find realistic solutions [2]. This last situation is known as *mode collapse*. Mode collapse is when the generator only learns a small subset of the possible realistic modes. For instance, if the

task is to generate images of a house, the generator could learn to create only images of huts. The generator would have missed all of the other modes consisting of house of other sizes or shapes. We believe we can get past these challenges by providing the GAN with a steady flow of novel uMA enterprise traces. This will allow the GAN's discriminator to update itself by learning from the characteristics of the new workloads. Batch normalization and mini-batch discrimination can also be used to handle model divergence and modal collapse [1]. Despite the cons, GAN has several pros like their ability to generate high quality realistic data, generate samples from a training dataset without the need for density estimations, and their ability to generate a wide variety of samples, which makes them ideal for distribution generation. Specifically with respect to uMA data, GANs applications are more likely in cases of data-limited uMA scenarios, because of their self-training semi-supervised learning ability to create new realistic data from noise.

Chapter 3

State of the Art

As wholly evident from Chapter 1 and Section 2.1, a better understanding of human activity and mobility in today's information-driven world has become increasingly essential in various areas such as network and communication provisioning, network and communication service deployment, urban planning, health care delivery, to name a few. Existing efforts in user and mobility activity (uMA) research can be roughly grouped into 5 categories: uMA feature analysis, uMA modelling and prediction, uMA surveys, uMA data generation using Generative Models and Evolution of publicly accessible uMA datasets.

3.1 *uMA Feature Analysis*

Positioning and localization technologies like GPS, cellular tower based geo-positioning, WiFi positioning and other technologies used to track motion have enabled additional, more sophisticated approaches to collecting human mobility data and mining patterns of interest. Feature vector studies using uMA datasets were introduced in the late $1990's$ with the goal of analyzing human interactions in various environments that could provide information about cultural group formation [119].

Between 2010 and 2015, uMA feature studies took off again using GPS traces to mine geo-locations [269] and geocommunities [270]. There are also studies that focus on coarse- versus fine granularity of uMA datasets [68], location-dependent versus location-independent datasets [87], periodic transitions between locations affecting human activity [189], and city-wide GPS logs from taxis [221]. Other references conduct compara-

tive studies of different GPS-based trace analysis techniques [154]. Other kinds of uMA trace analysis involve data from location-based social network (LBSN) platforms to: extract and infer the purpose of travel, or the activity at the destination of a trip in daily life scenarios [268]; or study the impact of location history collection on uMA features [196]; investigate human movement among points-of-interests (PoIs) [271, 114, 257]; exploit information on transitions between types of locations, mobility flows between locations, and spatio-temporal characteristics of user check-in patterns [170]. Datasets captured from applications like Twitter are information-rich, e.g., they can indicate diversity in movement modes among individuals as well as movement within and between cities [138]. Some references also talk about using multiple sources of data from both cellphones and transit [175], and extracting uMA patterns using tensor decomposition techniques [251]. Others discuss inferring human activity patterns from anonymized mobile communication usage [220].

In the last five years, with advances in data mining and data analysis techniques, several references have talked about the importance of Point of Interests (PoIs) and temporal distance to understand mobility patterns [179], using mobile and sensing data to analyze human habits and living environments [249, 135], mining human behavior and patterns from geo-socially tagged data [88], and learning mobility patterns with minimal user intervention [55]. With the rampant usage of deep learning (DL) techniques, DL-based feature extraction approaches have been used to analyze trajectory and transportation based mobility traces [105, 243]. Additionally, recent fog and edge computing technology has paved paths for healthcare [118] and transit [191] based mobility feature extraction.

More recently, user activity and mobility feature analysis has expanded to include Mobility-as-a-Service which allows paid access to mobility services like digital transportation management, environmental and health impacts of uMA patterns and choices, economic trends affected by social analysis etc. We divide these analysis techniques by application categories: connectivity, location, health and lifestyle.

Lifestyle based feature analysis included spatial, temporal analysis of geo-tagged Twitter data of Singapore residents to reduce crowds [259]; social media based opinion and pattern analysis to discover user mobility patterns, estimate polarized political opinions and tag interesting social media discussion topics [75]; improving energy efficiency in location by

studying direct impact of building user mobility on operational and transport energy [56]; using social analysis to improve social exchange in uMA industry with respect to daily habits of adapting to connected vehicles, electrical motorization [227]; analyzing determinants of active mobility choices to compare the demographic, socio-economic and cultural factors that influence it [186].

Health based feature analysis included Mobility and Trajectory based Technique for Monitoring Asymptomatic Patients (MTT-MAP) [51] that used time-ordered spatial and temporal trajectory and uMA records of Asymptomatic patients towards reducing the stress of socio-economic complications in the case of pandemics; using Meta's user mobility database to identify the role of infection threats and containment policies, through labor commuting flows and business travels [99], spatial analysis of changes in urban uMA patterns and the modal distribution of transport to correlate with the evolution of environmental air quality indicators in the city of Spain [116].

Location based feature analysis included analysis of various service attributes of transportation modes (car-sharing, private car, and taxi) along with socio-demographic attributes of users, to optimize car-sharing strategies [198]; Mobility-on-Demand and Mobility-as-a-Service applied to fixed-route and on-demand service policies for low income communities [238]; Analyzing user mobility and activity patterns from GPS data to study differences between non-work/non-home locations of working/non working users on workdays/offdays [145]; Using difference between prediction of trajectories among two different locations to optimize travel paths among the locations [52].

Connectivity based feature analysis included studying activity of a sensor device and its effects on high latency, which can result in low quality of services [140]; Privacy preserving, uMA supported federated learning vehicle algorithms [235].

3.2 *uMA Modeling and Prediction*

uMA modeling in the $90's$ focused heavily on communication systems applications. Examples of uMA models include: using residence time distributions to analyze channel holding time [272]; using features of asynchronous point-to-point communication like distribution of processes to locations, routing of messages, failure to reach locations and

their detection, to extract uMA patterns of processes [59]; and supporting activity in $IPv6$ without loss of connectivity [60]. uMA models studying mobility and activity of elderly people and their quality of life was also briefly studied [163]. uMA prediction between 2010 and 2015 started branching out into several new areas. Data from location based sensing networks, like a person's GPS trajectory, were used to predict current and future locations visited by users, how frequently they were visited [98, 256, 233, 240] and to find additional points of interests [79]. Communication-based uMA modeling considered opportunistic networks and used data shared by short range devices to predict user communication patterns [185]; it also targeted uMA-aware personalization and resource allocation for mobile cloud applications [262]. Transportation-based uMA models use bus/taxi travel requests to predict bus travel demand for different routes as well as locations for potential future customers [153]. There has also been some psychology-based human mobility and activity studies on regularity and predictability of human movements [87, 206], predicting uMA in response to a large-scale disaster [208, 210], and predicting long-term activity associating location information with contextual features like days of the week [197]. More recently, with uMA prediction riding the machine learning wave, there have been several references to DeepMove [110] that uses recurrent neural networks (RNNs) to predict human trajectory data, hidden Markov models to predict user movement [193], federated learning as a privacy-preserving mobility prediction framework [111], Deeptransport to predict user's future movements and transportation mode for a period of time [209], DeepUrbanMomentum for prediction of short-term urban mobility [134], variational trajectory convolutional networks to predict point of interests [115], and Neural Turing machine with Stacked RNNs to predict neighborhood human mobility patterns [224].

More recent trends in mobility modeling and prediction have focused on topics like tourist choices, network resource scheduling, situation based activity trend prediction, edge computing optimizations etc., we try to highlight some of those topics again dividing them based on the information based taxonomy layers.

Lifestyle based modeling and prediction included PredicTour which performs uMA modeling via social media profile extraction, to predict tourist activity [93].

Health based modeling and prediction included analyzing factors like number of new cases,

social distancing, stay-at-home orders, domestic travel restrictions, mask-wearing policy, socioeconomic status, unemployment rate, transit mode share, percent of population working from home, and percent of older (60+ years) and African and Hispanic American populations, to predict user motion and activity within USA in early days of the pandemic [82].

Location based modeling and prediction included analyzing spatio-temporal correlations and multi-type urban transition flows to predict individual traveling behaviors [101]; predicting the supply/demand of transport systems for efficient traffic management, control, optimization, and planning [169]; privacy-aware human trajectory prediction using adversarial networks [260]; group-based multi-features move (GMFMove), that constructs a uMA prediction model based on factors like the sequence of location, the category of location, and the geographic relevance of human mobility and activity [141]; Prediction by Partial Matching (PPM) to forecast each vehicle's path and cluster the vehicles with similar future path, moving direction, and moving speed into one group [252].

Connectivity modeling and prediction included optimizing system performance with wireless resource scheduling methods for high activity cases by predicting traffic volume [247]; propagation delay prediction using energy-efficient mobility based localization scheme [168]; Lightweight uMA prediction and offloading framework (LiMPO) that optimizes latency and energy consumption while improving the resource utilization of mobile edge computing servers [132]; MoSaBa, a wireless crowd charging method which leverages uMA prediction and social information for improved energy balancing [174].

3.3 *uMA Surveys*

Over the years there have been various surveys outlining the state-of-the-art of user mobility and activity research. We have grouped the surveys by their application type.

Lifestyle surveys, where Thorton et al surveyed user characteristics and their effect on the uMA, especially reactions to environmental change [223]; Barbosa et al surveyed geolocation data to study individual versus collective uMA patterns [70]; Lin et al surveyed data mined from GPS trajectory data focusing on locations significant for prediction of future moves, detecting modes of transport, mining trajectory patterns and recognizing location-based activities [154].

Location surveys for example, Palmer et al surveyed various gathering and analyzes techniques for spatially-rich demographic data using mobile phones [178]; Asgari et al surveyed datasets representing population flow in transportation networks along with their data types and various applications [63]; Toch et al analyzed large scale uMA datasets using machine learning techniques [225] focusing on the data's positioning characteristics, the scale of the analysis, the properties of the modeling approach, and the class of applications.

Connectivity surveys where Karamshuk et al analyzed challenges associated with uMA in Opportunistic Networks research and also reviewed uMA analysis and models [139]; Becker et al studied uMA characterization with respect to cellular network data [73]; Hess et al described steps for creation and validation of mobile networking based uMA models [122]; Yang et al surveyed wireless indoor localization using inertial sensors [250].

Other uMA based surveys include Solmaz et al discussing commonly used metrics and data collection techniques for various models and also proposed a taxonomy to classify uMA models based on their main characteristics [207]; Wang et al surveyed uMA prediction models derived using multi-source datasets [236]. So far there have been no surveys on health specific uMA modeling.

3.4 Generative Models for uMA data generation

The state of the art in generation of uMA datasets have been captured in detail by surveys [158], which describes traditional versus Deep Learning based techniques and [204], which describes GANs specific to uMA data. [158] divide their taxonomy into the uMA prediction and generation categories. Examples of traditional uMA generation techniques:

- Flow Generation: Gravity based and radiation based models, which failed to capture variability of real flows and were limited in capturing non-linear relationships between features.
- Trajectory Generation: Data driven human mobility modeling Exploration and Preferential (EPR) based random walk process models and temporal/spatial extensions which suffer from limited realism.

Limitations mentioned above can be tackled by deep generative models. Deep generative models refer to machine learning based models, usually a combination of generative models with neural networks, that can be used to extract patterns and abnormalities in a data distribution and then create a new instance of the same data. Some examples of commonly used generative models are Restricted Boltzmann Machines (RBMs), Variational Autoencoders (VAE), Hidden Markov Models (HMM) and Generative Adversarial Networks (GANs). Different variations of RBMs have been used for reconstruction [222] and analysis of tabular datasets, while VAEs have been used extensively for pattern recognition and generation [147]. RBM models were an excellent choice for prototyping datasets with smaller feature sets, but were inefficient for larger feature sets and most previous VAE models have recently been replaced by GANs due to the advantages of the double feedback between GAN's generator and discriminator [107].

GANs run on relatively simple logic: if one model is trained on all available training data, and a second model is trained to try and come up with examples that the first model classifies incorrectly, the second model will eventually converge to produce synthetic data that is indistinguishable from the data the first model was trained with [126]. GANs have two models, which are artificial neural networks, known as the *discriminator* and the *generator*, and they are placed at odds, improving their performance respectively until they produce a high quality dataset. In the last five years GANs have become the de facto standard for being used for different applications [57], all of which are either related to image/video/audio synthesis or in some cases generation of datasets representing human activity like healthcare datasets. Starting in 2019, there have been attempts at trying to generate and analyze time series data using LSTMs, Wasserstein GANs, conditional GANs and other generative models. Most of these applications were to generate trends in financial markets [218, 146], emulate ECG and PPG [94], rhythm generation [267] and anomaly detection [150]. With respect to user activity data, GANs have been used for semi-supervised learning of activity recognition, where learning from raw datasets was computationally intensive like in CsiGAN [246]. In [167], GANs were used to generate synthetic human activity datasets to improve classification of activity recognition. The work reported in [128] used GANs to generate differentially private healthcare data. [204] discuss four GAN

architectures specific to uMA categories:

- SocialGAN, Non-Parametric trajectory GAN: based on the SGAN, RGAN, DC-GAN and WGAN architectures, which generate realistic pedestrian walking trajectories conforming to social laws and geographically accurate visiting behavior.
- GAN based Location Density Matrix Generator, trajgan: WGAN and CGAN based pedestrian and vehicular trajectory generation models.

Some common technical challenges associated with GANs are: (1) No evaluation models/metrics to improve model training or for testing fidelity of complex generated data points. (2) Lack of a good equilibrium finding algorithm causing unstable training pipeline that can result in issues like modal collapse for un-optimised training parameters. (3)Resource utilization variability across different GAN models, as we will explore in section 7.3. But the pros outway the challenges. Unlike traditional approaches that can only capture a single mobility aspect, like spatial dimension, at a time; DL based approaches like GANs can simultaneously capture mobility aspects like spatial, temporal and social dimensions. In addition GANs can also capture complex and non-linear relationships across features within a dataset, making the model ideal for generating datasets that are more realistic than the traditional approaches. To leverage these features and given the current success rate of using GANs for application specific use cases, we have decided to use GANs for our uMAD's dataset generation module.

3.5 Evolution of public accessibility of uMA datasets

Historically, uMA datasets have been open sourced through university based servers and websites like CRAWDAD [211] by Dartmouth, Google's subsidiary Kaggle [214], data.world [212] etc. These websites have a large trove of existing datasets, millions of subscribers, along with added capability to add new datasets into the database. For example, Kaggle allows users to upload datasets as large as 100 gigabytes. They also provide data preprocessing on the raw datasets to make them more readable to the average user. Despite all of these benefits, there are two major challenges with these websites. First, they require you to be registered, albeit free, with the companies, which means the company has access

to your name and email address. Second, locating uMA data within the various categories of datasets can be challenging, even with relevant keywords, which ends up with the users manually going through each data summary to pick the relevant dataset. Our tool focuses only on uMA data, therefore reducing the pool of data that users will need to comb through to find their requested datasets. Another category of tools available for public uMA datasets are companies that provide data as a service.

Among open-source programming interfaces available to users there is RFDatafactory [42], which is a platform that allows users to access and create custom WiFi datasets. RFDataFactory currently has 31 data sets and 15 contributors. They provide APIs for data preprocessing, visualization, feature extraction etc. Another example of an open source dataset generation tool is Synthea [43], developed by the MITRE Corporation which is an open-source, synthetic patient generator that models the medical history of synthetic patients. Our tool is inspired by these open source projects that provide generation tools for datasets and we use these projects as a baseline to provide an adapted uMA specific tool.

From this summary of related work, it is clear that user mobility and activity is studied across an extremely broad scope of topics. However, to the best of our knowledge, our work is the first to provide comprehensive documentation and taxonomy classification of the various open source uMA datasets. Using this thesis, the researcher's have access to a high level classification of a wide range of popular uMA traces along with granular details about their publishing source and privacy preserving properties. They can also use our taxonomy to guide classification of a newly generated uMA trace, which will help them scope out potential application areas for the new trace.

Chapter 4

Classifying Popular Open Source Mobility Traces

The current state-of-the-art in user mobility and activity (uMA) research has extensively relied on open-source traces captured from pedestrian and vehicular activity through a variety of communication technologies as users engage in a wide-range of applications, including network resource planning, connected healthcare, localization, social media, e-commerce, etc. As we discussed in Chapter 1 and Section 2.1, most of these open source traces are feature-rich and diverse, not only in the information they provide, but also in how they can be used and leveraged. This diversity poses two main challenges for researchers and practitioners who wish to make use of available mobility datasets. *First*, it is quite difficult to get a bird's eye view of the available traces without spending considerable time searching and inspecting them. *Second*, once the trace types have been found, determining whether the identified datasets are adequate to a users application needs is typically labor intensive and time consuming. The purpose of this chapter is three-fold. It proposes a taxonomy to classify open-source mobility traces including their mobility mode, data source and collection technology. It then uses the proposed taxonomy to classify popular open-source uMA traces, along with providing their publishing source, licensing, and anonymization strategy. Finally, it highlights three case studies using popular publicly available uMA datasets to showcase how our taxonomy can be used to tease out feature sets in traces to help determine their applicability to specific networking, health, lifestyle and location based use-cases.

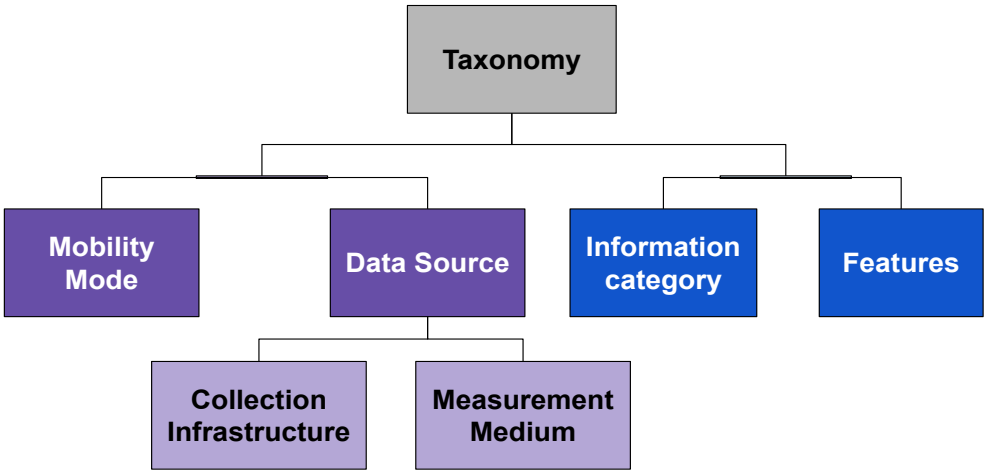

Figure 4.1: Taxonomy overview

4.1 Taxonomy Overview

Coming up with a representative taxonomy for uMA traces is not trivial due to their feature and application diversity. To create our main taxonomy we used a bottom up approach, starting from the data source or technology used to collect the traces, all the way up to the uMA mode i.e., pedestrian or vehicular, being represented by these traces. Another challenge we faced was finding a representative collection of traces to define our taxonomy. Most current state of the art uMA modeling techniques generate their own traces, only a fraction of which are put on public domain. Our strategy was then to select datasets that have been widely used with the goal of creating a classification scheme that is broad enough so that existing or new traces can be categorized using our taxonomy. Additionally, our study also analyzes potential applications of these traces to identify significant gaps in availability of real uMA traces, and thus motivates the need for realistic uMA trace generators.

An overview of our taxonomy is illustrated in Figure 4.1. Under the left branch we have:

- **Mobility Mode**, which refers to the user's mode of movement, namely stationary, pedestrian or vehicular.
- **Data Source** considers how the trace was collected and is further subdivided into:
  - **Collection infrastructure:** systems that host the devices used to collect data.

  - **Measurement Medium:** actual device/technology that generates the different measurements used to populate the datasets.

Stationary mobility represents information from sources like a group of access points that can be used to triangulate the movement of an user. Pedestrian mobility typically represents movement within a limited geographic region, while vehicular mobility involves movement using various modes of transportation usually spanning larger geographic regions. Vehicular mobility includes personal vehicles and public modes of transportation like buses, trains, shared scooters/bikes/cabs, ships, airplanes, etc. In the case of pedestrian mobility, data is usually collected through smartphones, laptops, tablets, wearable devices or through network infrastructure gear. In vehicular mobility, data is either gathered through end-user devices like smartphones, laptops, tablets and wearable devices, in which case it is usually generated when these devices are inside a moving vehicle; or through smart-vehicle hosted devices. A more detailed example collection infrastructure and measurement media are illustrated in Figure 4.2.

As illustrated in Figure 4.1, under the right branch we have:

- **Information Category:** Application groups created by studying existing open source mobility traces.
- **Features:** Raw and derived information types generated in open source mobility traces.

Based on the existing set of open-source traces, we have identified four main information categories:(1) *Connectivity* traces are typically used to optimize network performance, e.g., provisioning, redistributing resources to better manage network traffic, etc [61, 35, 28, 30, 15, 27, 12, 19, 17, 16, 9, 11]; (2) *Location* traces can be used for location-based optimizations like improving waiting area around a specific business that has a heavy footfall [6, 61, 29, 34, 18, 28, 30, 12, 17, 16, 14, 11, 22]; (3) *Health*-related traces are applied to improve health solutions like adding features for call-for-help services; there aren't any current open source health traces because of HIPAA compliance, but we include this class as a future categorization possibility; (4) *Lifestyle* traces are used to draw patterns in user behavior like sleep cycles, downtime etc. Note that the taxonomy also lists, under every in-

formation category, some examples of features that may be present in the respective category of traces [29, 34, 18, 5, 10, 7, 33].

4.2 Taxonomy

Keeping in mind the taxonomy overview, we dig deeper into the criteria used to classify open source mobility traces.

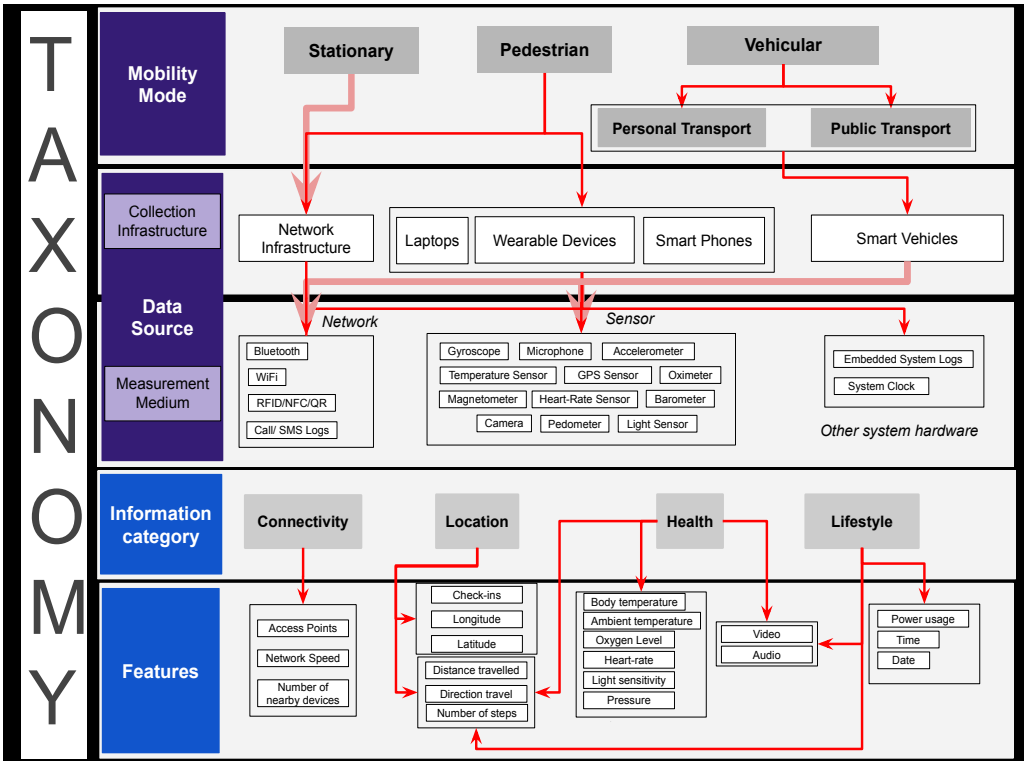

Figure 4.2: Mobility trace classification based on mobility mode and data source.

4.2.1 Mobility Mode

The Mobility Mode layer organizes traces into two major categories: pedestrian and vehicular.

- ***Stationary Mobility*** Stationary mobility datasets can provide an understanding of how users or vehicles move with respect to stationary sources, like access points or servers located in specific locations. A user's location can be estimated by using triangulation techniques on data from access points. Applications of this type of data are in areas of

network optimizations based on usage patterns of access points at different points of time. Another common application of triangulation is in GPS research, Cellphones and car GPS units use triangulation to find your location relative to the radio towers in an area.

- ***Pedestrian Mobility*** Pedestrian mobility datasets can provide an understanding of how people move in certain areas according to various aspects like traveling distance, locales where people tend to congregate, and trends related to places visited, when they are visited, and for how long [10, 7, 21, 13]. Lately, there has been a lot of emphasis on understanding crowd management, e.g., identifying attractors and detractors, determining and optimizing wait times in various situations, localizing congestion and bottlenecks in crowded localities, all of which provide invaluable insights on how people move in different places and situations. Other examples of applications that can benefit from better understanding of pedestrian mobility include reduction of crowd-based carbon-dioxide emissions, population density control in crowded areas, and optimization of city infrastructure. The COVID-19 pandemic has made the importance of being able to model pedestrian mobility patterns even more critical so that it can be used to perform contact tracing and manage public spaces in order to better enact local policies and restrictions.

- ***Vehicular Mobility*** Vehicular mobility traces can be used to characterize movement of various modes of transportation [6, 22, 14]. Before automobiles, cities were limited in terms of area, population and business prospects. The advent of different motorized transportation modes have fundamentally transformed the way cities are planned and expanded. Vehicular mobility can be broken down into personal and public transport. Public transport can further be differentiated based on mobility medium, namely: ground, sea, or air. Irrespective of whether publicly shared or personally owned, vehicular mobility captures various aspects of a given region like traffic during different times of the day, road congestion, favorable modes of transport, usage based on location specific infrastructure, etc. Understanding vehicular movement can also be insightful to understand trends among social communities. Other applications of vehicular mobility modeling include managing electrical vehicles (e.g., provisioning

of charging stations), supporting autonomous driving, managing shared modes of transportation and supporting smart transportation services.

Examples of vehicular mobility datasets include:

- *Personal Transport*, which are getting smarter by the day with new advances in 5G and IoT technology. With these automobiles being connected to the Internet, there is a treasure chest of uploaded data that can help us model useful mobility patterns. Add to that application integration, where applications like Google and Apple Maps can record common routes and locations frequently visited, we can derive a complete picture of user mobility.
- *Public transport* by companies like Uber, Lyft, and Bay Wheels, which also collect mobility information via data shared by user applications. This includes information like pick up/drop off location, route taken, and stops along the way. Shared transport by local governments including buses and taxis which also provides us with information similar to privately-owned shared public transport.

4.2.2 Data Source

In the second layer, traces are then classified according to their data sources, including the infrastructure where data is being stored/located, and how data was obtained/measured.

1. ***Collection Infrastructure*** Mobility data is typically collected by network infrastructure or end user devices.

   Network infrastructure includes devices that provide network connectivity to end users such as base stations and cell phone towers, that provide fingerprints which help identify times when a user has been around a certain location and for how long; as well as routers, switches, hubs, and wireless access points that provide network traffic exchange information which can be used to study mobility as a device switches between multiple components.

   End user devices can either directly be the source of mobility traces or traces can be extracted from applications hosted on these devices. Examples include:

- Smartphones/Tablets/laptops that have become increasingly common as information and communication sources. They can collect/store information such as location, social media check-ins, communication logs like calls/messages being exchanged and sensor based information like motion tracking and health conditions.
- Wearable devices, like smart-watches, with their sensing capabilities are ideal to monitor health conditions, and crowd density during ordinary and emergency scenarios.
- Smart Vehicles that provide network and location information.

Pedestrian traces can be extracted from public network infrastructure and end user devices like smartphones, tablets, laptops, wearable devices, etc. Vehicular traces can be captured using data from specialized end user devices like transportation smart cards in personal and shared vehicles or, from transportation / transit applications or websites. Vehicular mobility can also be collected using sensors embedded in vehicles.

2. ***Measurement Medium*** Here we consider how data was obtained or measured and divide traces into three categories: the *Sensor* category refers to traces collected via a variety of sensing devices; the *Network* category includes traces collected using devices that provide network connectivity (e.g., cellular, WiFi, Bluetooth, etc); and the *Other system hardware* group includes information derived from logs of on-device applications.

   Sensor-Based data is usually the output from a device that measures the physical environment. The output of sensors is usually used as raw information or to trigger other sensors or processes. Below, we include examples of sensor data commonly found in mobility datasets.

   - *Global Positioning Systems* or GPS sensors are receivers with antennas that use satellite based navigation to provide time and geolocation information usually in the form of latitude and longitude coordinates. In some cases, GPS sensors can also capture position in the form of velocity and orientation. These features

are most commonly used as unique location identifiers. The datasets we have explored include latitude and longitude coordinates only.

- *Light* sensors are devices that convert any form of light energy, visible or infrared, into electrical signal outputs. In the case of mobility datasets, information on when it is day versus night can be useful to monitor patterns in user habits.
- An*Accelerometer* measures acceleration using three axes, X, Y and Z. Such sensors mainly provide two kinds of information: first, the static force applied on the sensor due to gravity and orientation; second, the force and acceleration exerted on the sensor in motion.
- *Gyroscope* sensors calculate angular velocity, or change in rotational angle per unit of time, usually measured in degrees per second. In the datasets we consider, gyroscope sensors add an additional dimension to the accelerator data to determine the orientation of a device.
- *Magnetometers* measure the relative change in magnetic field at a given location.
- *Pedometers* are mechanical devices that use software to detect vertical movement at the hip, to count the number of steps taken by a user. This can indirectly be used to derive information like distance traveled and patterns of other physical activities.
- *Oximeters* or pulse oximeters use LEDs to emit two types of red light through human tissue in order to measure oxygen saturation levels in the blood along with the number of times our heart beats per unit of time.
- *Temperature* sensors are electronic devices that measure surrounding ambient temperature and convert that into electronic data, to measure changes in temperature.
- *Camera* produces records in-terms of images or videos that can be used to derive social patterns in different human communities. This information can be derived based on the contents of the picture/video, location where the record was made, people who were a part of the record, what the people in the record were doing, etc.

Datasets containing sensor information can be collected directly by sensors and can be classified either under pedestrian or vehicular mobility.

Network-Based The ever increasing popularity and availability of mobile communications has made ”anywhere, anytime connectivity” a reality. As such, end user mobility information that is collected through access network devices helps manage and provision network resources. Examples of network connectivity information contained in mobility traces include:

- *Bluetooth* technology targets short-range wireless communication. .
- *WiFi* is one of the most widely used wireless technologies for data communication in local-area networks. It is also widely used as Internet access technology. Information from WiFi networks like access point associations / dissociations and signal strength can be used to determine user location as well as mobility patterns and trajectories.

Other System Hardware Mobility datasets can also include information generated by Application based data, also known as data collected from other system hardware, is actively triggered by users utilizing an application. Most of this data collected is only restricted to when the application is running and does not include information from the application's rest time. This kind of data usually gives information in the form of timestamps. Some specific datasets combine these timestamps with other information like location, human movement and constraints in the digital space.

The *Clock* and the *Calendar* applications provide us with date and time, which can be useful if we want to model time series data, or analyze patterns and trends over specific time/date periods.

The *Map* application uses requests to derive information about frequency of trips, locations in-terms of latitude and longitude, start and end times, types of transportation used in different locations for different times, pedestrian population in different areas of a city. Information generated by this kind of application can be useful for city planning and vehicle traffic management.

Location based social network (LBSN) follows geosocial networking principles, where a social networking application has geographic capabilities like geotagging and geocoding to collect additional information about human social patterns. Location coordinates like latitude and longitude, added to uploaded pictures or social network check ins to cities, bridges the gap between the physical world and the online services, bringing social networks back to reality. Like the map datasets, this category of traces can also belong to the sensor group of datasets.

4.2.3 *Information Category*

Open source mobility traces can also be grouped in terms of their application and features these traces contain. Information category can be roughly divided into four categories, which we have derived from Section 2.2: Connectivity, Lifestyle, Location and Health. The features under these application brackets can be generated using the appropriate sensors, details of which we have already covered in the measurement medium section of the taxonomy. One thing to keep in mind when using the application based taxonomy trajectory is that due to the vast diversity in features within each application, there can be traces that belong to a subset of application categories instead of a single category.

4.3 Applying the Taxonomy

This section has two main goals. The first part of this section is used to categorize popularly used uMA traces using the buckets in our taxonomy. The second part of the section focuses on one pedestrian and one vehicular trace example, to showcase how our taxonomy can be used to tease out out communication technologies (data collected using communications between GPS and mobile devices), information category, features, in a trace, which then helps the users determine the trace's applications, resulting in analysis of the trace using popular uMA metrics.

4.3.1 Classifying Open Source uMA Traces

Analyzing uMA traces have a multitude of applications including provisioning communication infrastructure, urban resource planning, vehicular and network traffic analy-

Table 4.1: Popular uMA traces classified using our taxonomy

TRACES		EVALUATION		APPLICATION
	Mobility Mode		Data Source	Information category
		Collection Source	Measurement Medium	
T-Drive [6]	Vehicular	GPS	Sensor	Location
Crivello [61]	Pedestrian	Wearables/Smartphones	Sensor/Network	Connectivity/Location
Apple Maps [29, 34]	Pedestrian	Smartphones	Sensor/Other	Location/Lifestyle
Google Maps [39, 34]	Pedestrian	Smartphones	Sensor/Other	Location/Lifestyle
Descartes Lab [35]	Pedestrian	Smartwatches/phones	Sensor	Connectivity
KCMD-DDH [18]	Pedestrian/Vehicular	GPS/Smartphones	Sensor/Other	Location/Lifestyle
JRC(Europe) [5]	Pedestrian/Vehicular	GPS/Smartphones	Sensor/Other	Location/Lifestyle
GIM [32]	Pedestrian/Vehicular	GPS/Smartphones	Sensor/Other	Location/Lifestyle
Gowalla [7]	Pedestrian	Smartphones/Laptops	Sensor/Other	Location/Lifestyle
Brightkite [7]	Pedestrian	Smartphones/Laptops	Sensor/Other	Location/Lifestyle
Nsense [21]	Pedestrian	Smartphones/Laptops	Sensor/Other	Location/Lifestyle
Cabspotting [8]	Vehicular	GPS	Sensor/Other	Location
Geolife [13]	Pedestrian	Smartphones/Laptops	Sensor/Other	Location/Lifestyle
NYC Mobility [24]	Pedestrian/Vehicular	Smartphones/Laptops	Sensor/Other	Location/Lifestyle
GRID Bikeshare [33]	Vehicular	GPS	Sensor/Other	Location
Texas Mobility [26]	Pedestrian/Vehicular	GPS, Smartphones	Sensor	Location/Lifestyle
UILM [25]	Pedestrian	Smartphones	Sensor/Other	Location/Lifestyle
GSMC [28]	Pedestrian	Smartphones	Network/Other	Connectivity/Lifestyle
Flexran [30]	Pedestrian	Smartphones	Network/Other	Connectivity/Lifestyle
KTH [15]	Pedestrian	Smartphones	Network	Connectivity
BLEBeacon [27]	Pedestrian	Smartphones	Network	Connectivity
HYCCUPS [12]	Pedestrian	Smartphones	Network/Other	Connectivity/Lifestyle
Cambridge Haggle [19]	Pedestrian	Smartphones	Network/Other	Connectivity
Fire Dpt Asturius [17]	Pedestrian	Smartphones	Sensor/Network	Connectivity/Lifestyle
SocialBlueConn [16]	Pedestrian	Smartphones	Network/Other	Connectivity/Location
Rome Taxis [14]	Vehicular	GPS	Sensor/Other	Location
SIGCOMM 2009 [9]	Pedestrian	Smartphones	Network/Other	Connectivity
Commercial Seoul [11]	Pedestrian	Smartphones	Sensor/Network/Other	Connectivity/Lifestyle
Chicago taxi [22]	Vehicular	GPS	Sensor/Other	Location
Pedestrian Louisville [20]	Pedestrian	Smartphones	Sensor/Other	Location
MHealthDroid [69]	Pedestrian	Smartphones	Sensor/Other	Lifestyle
NetAAL [67]	Pedestrian	Smartphones	Sensor/Network	Connectivity

sis, social management and in some cases even healthcare. Examples of social management would be structuring community needs like libraries, shopping complexes based on mobility footprints. This section takes a set of popular traces and classifies them using our taxonomy, as shown in Table 4.1 and then elaborates on features in the traces and subsequently some of their existing applications.

T-drive [6], a vehicular dataset by Microsoft Research containing GPS trajectories, i.e., longitude and latitude of approximately ten thousand taxis in Beijing, has been used to derive optimized travel times, route and traffic prediction [258, 183]. This trace has also been used for non-mobility related applications like testing database management systems and large-scale data processing techniques [64], and testing density-based spatial clustering of applications with noise[202]. Other vehicular traces include Cabspotting [8] consisting of GPS latitude and longitude information collected from over 500 taxis; and mobility data collected from taxi cabs in Rome derived from GPS coordinates [14]. Such traces have been

Table 4.2: Feature list for Popular uMA traces classified using our taxonomy

TRACES	FEATURES
T-Drive [6]	Identifiers, Date, Time, GPS coordinates
Crivello [61]	IMU sensor data, WiFi and geo-magnetic field fingerprints
Apple Maps [29, 34]	Date, Time, Location, Transportation type, Usage
Google Maps [39, 34]	Date, Time, Location, Percentage increase/decrease in number of location visits
Descartes Lab [35]	travel information, dates
KCMD-DDH [18]	statistics from air passenger traffic and tourism
JRC(Europe) [5]	land and sea arrival informtion via travel portals
GIM [32]	land and sea arrival informtion via travel portals
Gowalla [7]	Latitude, Longitude, social network checkins, edge relations
Brightkite [7]	Latitude, Longitude, social network checkins, edge relations
Brightkite [7]	Latitude, Longitude, social network checkins, edge relations
Nsense [21]	Latitude, Longitude, social network checkins, edge relations
Cabspotting [8]	Taxi id, Date, Longitude, Latitude, Fare
Geolife [13]	GPS trajectories
NYC Mobility [24]	Survey for travel choices, user behavior
GRID Bikeshare [33]	GRID temperature, date, bike usage status
Texas Mobility [26]	GPS coordinates from different modes of transportation
UILM [25]	Census information, birth place, current home city
GSMC [28]	Device bluetooth encounters
Flexran [30]	Device bluetooth encounters
KTH [15]	User associations to their WiFi networks
BLEBeacon [27]	Device bluetooth encounters
HYCCUPS [12]	usage statistics, user activity, battery statistics
Cambridge Haggle [19]	Device bluetooth encounters
Fire Dpt Asturius [17]	WiFi, bluetooth, GPS information
SocialBlueConn [16]	Facebook friendships and interests
Rome Taxis [14]	Taxi id, Date, Longitude, Latitude
SIGCOMM 2009 [9]	Bluetooth encounters, opportunistic messaging, and social profile
Commercial Seoul [11]	GPS infor- mation, Wi-Fi fingerprints, user-annotated location information
Chicago taxi [22]	Taxi id, Date, Longitude, Latitude
Pedestrian Louisville [20]	GPS Trajectories
MHealthDroid [69]	IMU sensor, Physical Activity labels
NetAAL [67]	RSSI signatures, Room switch, Paths taken

used to study mobility patterns in urban taxis [123], route regularity [109], social analysis in vehicular ad hoc networks [157], and identify outlier mobility trajectories [192]. Other less known vehicular traces that contain sensor information are the GRID bikeshare dataset, which describes the main attributes of GRID temperature, including the feed, operator, hours, calendar, regions, pricing, alerts, stations, and bike status [33]; and the mobility dataset from the city of Austin, Texas which includes GPS sensor information from bicycles and other mean of transportation [37, 20, 26]. More recent taxicab datasets like the Chicago cabs trace collected location data from seven thousand licensed cabs operating within city limits has been used to route prediction and optimization [22].

Crivello refers to a pedestrian trace which includes sensor information from wearable devices and network connectivity information from smartphones; this trace has been used

to compare and evaluate indoor localization solutions [61, 201], and for health applications like sleep quality monitoring [72]. Microsoft's Geolife dataset [13] consists of approximately eighteen thousand GPS trajectories with a total distance of $1,292,951$ kilometers and a total duration of $50,176$ hours collected from GPS loggers and GPS-enabled phones. The Geolife dataset has been used for transportation related applications like identification of transportation modes to create sophisticated intelligent transport systems [152, 166, 215], prediction of transport mode choices [176]; Privacy related applications like secure data compression in cloud [182], privacy-preserving location preferences [160, 151]; and benchmarking performance of large datasets on prediction and generation models [86, 89, 90].

Application generated traces like COVID traces derived from request information in Google and Apple maps [31, 29] consists primarily of location information represented by countries, regions, sub-regions and cities, combined with lifestyle information like transport type and location category. Other application based traces include data from Location Based Social Networks (LBSNs) like Gowalla [10], Nsense [21] and Brightkite [7] which use social network check-ins as the main source of the mobility data. We will elaborate on their applications in the next subsection.

Application based traces can also include census and migratory information. Examples include the US Internal Lifetime Mobility (UILM) [25], which predicts mobility based on when and where a user is born and where they are currently located. The NYC Citywide Mobility survey of the New York City residents' travel choices, behaviors, and perceptions [24] collects mobility information via online surveys and phone surveys. The Knowledge Center on Migration and Demography, Dynamic Data Hub (KCMD-DDH) contains global transnational mobility data that provides us with information on country-to-country cross-border human mobility using global statistics on tourism and air passenger traffic [18]. The Knowledge Center on Migration and Demography, Dynamic Data Hub also highlights information on monthly air passenger flows, which can be synthesized into a set of indicators between countries worldwide; demography and mobility data collected by the Joint Research Centre (JRC) and the Directorate General for Regional and Urban Policy in European metropolitan regions in 2018 [5]; and region based mobility data collected via interactive maps publicly available on the Flows to Europe Geoportal, which provides statistical

updates on migrant and refugee land and sea arrivals and routes towards Europe [32].

Another important consideration when studying uMA datasets is how they were collected, i.e., what kind of collection infrastructure and measurement medium were used. For example Bluetooth networks provide information about nearby devices and their characteristics [9], low energy packets generated by BLE beacons from end user devices like smartphones and laptops also provided user mobility information [27], the Cambridge Haggle dataset that contains bluetooth encounters between 12 nodes for approximately 6 days [19]; Asturias (Spain) Fire Department mobility and connectivity traces generated by GPS devices embedded mainly in cars and trucks, but also in a helicopter and a few personal radios [17]; traces containing Bluetooth encounters, Facebook friendships and interests of a set of users collected through the SocialBlueConn application at the University of Calabria [16].

Other notable network traces, also collected using applications, include the Global System for Mobile Communications (GSMC) which gathers information from approximately 10 mobile smartphone (iPhones) users via the MySignals iPhone App [28]; data collected by Flexran from a platform for software-defined radio access networks [30]; data collected using the HYCCUPS Tracer, that contains availability and mobile interaction information such as usage statistics, user activity, battery statistics, or sensor data, a device's encounters with other nodes or with wireless access points [12]; and traces with Bluetooth encounters, opportunistic messaging, and social profiles of 76 users, collected using the MobiClique application at the SIGCOMM 2009 [9].

Mobility data collected by organizations include records of authenticated user associations to their WiFi networks [15] and fine-grained network mobility data from commercial mobile phones in Seoul, Korea, containing continuous GPS information combined with Wi-Fi fingerprints and user-annotated location information [11]. In the following subsections we apply our taxonomy to three traces from this non-exhaustive list followed by some mobility analysis and outline. The three traces we are choosing are the COVID mobility traces by Google, since they are of the pedestrian type, the Cabspotting traces, since they are of the vehicular type, and the Brightkite traces, since despite being a feature-sparse social networking dataset, they have several important applications.

4.3.2 Situational COVID analysis using data derived from mapping platforms

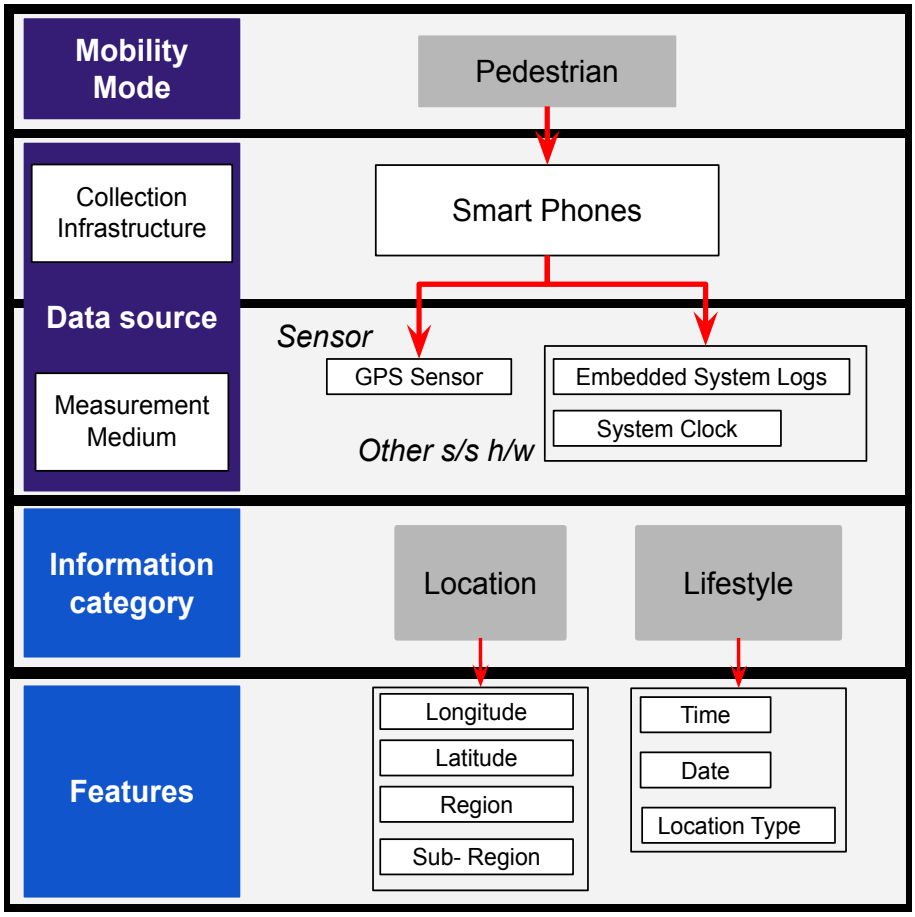

Figure 4.3: Classifying COVID Mobility Traces extracted from Google Maps.

Given the current ongoing battle the world is fighting against COVID-19, we apply our taxonomy to classify the very well known Google's COVID Mobility trend dataset, as shown in Figure 4.3, derived using data from applications like Google maps on smartphones. Companies like Microsoft, Google and Apple have extracted data from applications like Google and Apple Maps to analyze changes in mobility trends since the COVID-19 pandemic started in late 2019 [34]. We classify this under pedestrians since the information is requested on an application in a pedestrian smartphone. The raw maps dataset provides date-wise GPS locations and a percentage increase or decrease in mobility for various categories within each location. The categories include retail/recreation, grocery, parks, residential, workplace, and transit stations and are derived from location tags present in the map settings. The Google dataset, lays emphasis on public businesses and properties signaled in the requests. As such,

we use it to analyze trends based on the increase or decrease in number of requests for the specific classes of locations/businesses. We can generate charts for cities, states and/or at country levels. From the figure we observe that, as the months progressed from February to May, with increase in COVID threat, there has been up to a 38% decrease in visits to workplaces, a 35% decrease in retail and recreation Maps requests, a 30% decrease in public transportation usage, and a 25% increase in park visitations.

Real life applications of these traces are:

- ***Lifestyle:*** One real life application of this dataset from 2022 is combining it with the geo-spatial analysis of tweets in Singapore [259] to determine user visitation and amenities usage patterns at locations like parks, public links between parks and malls, taxi stands, residential areas, and shopping malls; Examination of the impact of early evening curfew on mobility by studying a shift in curfews from 9pm to 6pm in Greece using Google mobility data [230]; Studying the impact of mobility restriction strategies in the control of the COVID-19 pandemic to model the relation between COVID-19 health and community mobility data [54]; Using Google mobility reports combined with geotagged Twitter data to extract spatiotemporal human mobility patterns during this COVID-19 pandemic in New York City [136]; Description of the economic activity using internet during COVID-19 pandemic, aimed to show the relationship between the people mobility during COVID-19 with economic activity using internet [165].

- ***Health:*** Other real life applications for the Google Mobility Report include: mobility used as a representation for risky behavior, comparing exposure before and after mask mandates were imposed [232]; Study the impact of mobility restriction on reducing the COVID-19 effective reproduction number in the Kingdom of Saudi Arabia [58]; Assess the impact of contact tracing in middle-income countries to provide data to support the expansion and optimization of contact tracing strategies to improve infection control [229]; using Google mobility report to support government policies, national culture and promote social distancing during the first wave of the COVID-19 pandemic [239]; Comprehensive survey of human mobility open data that can guide researchers and policymakers in conducting data-driven evaluations and decision-making for the COVID-19 pandemic and other infectious disease outbreaks [125];

- ***Location:*** Combining data from google report and local agencies to compare the transport impacts of the COVID-19 in Germany and State of Qatar, based on the rates of infection and response measures [129]; Analysis of impacts of working from home on activity-travel behavior during the pandemic [194]; Combining Google mobility data and Apple maps data to track changes in community mobility and transport modes during the COVID-19 Alert levels [242]

- ***Connectivity:***Evaluation of the association between social distancing quantified by mobile phone data and the current prevalence of COVID-19 infections in the U.S. per capita [96];

4.3.3 Location based mobility pattern study using GPS data

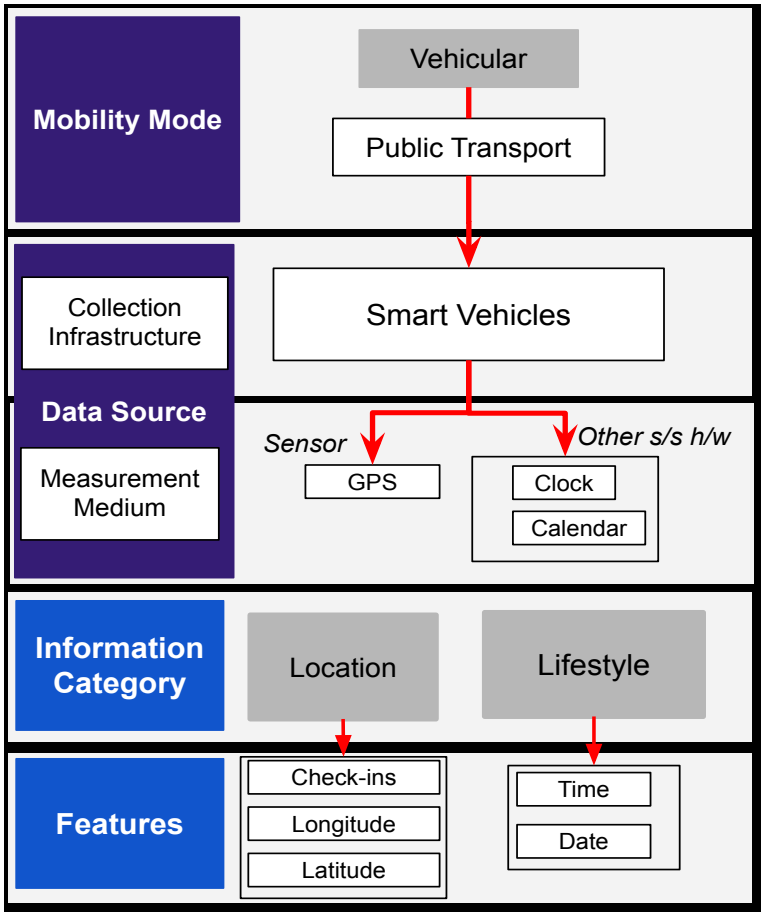

Figure 4.4: San Francisco GPS Taxi Traces.

We highlight mobility patterns in two different datasets:(1) Derived from GPS

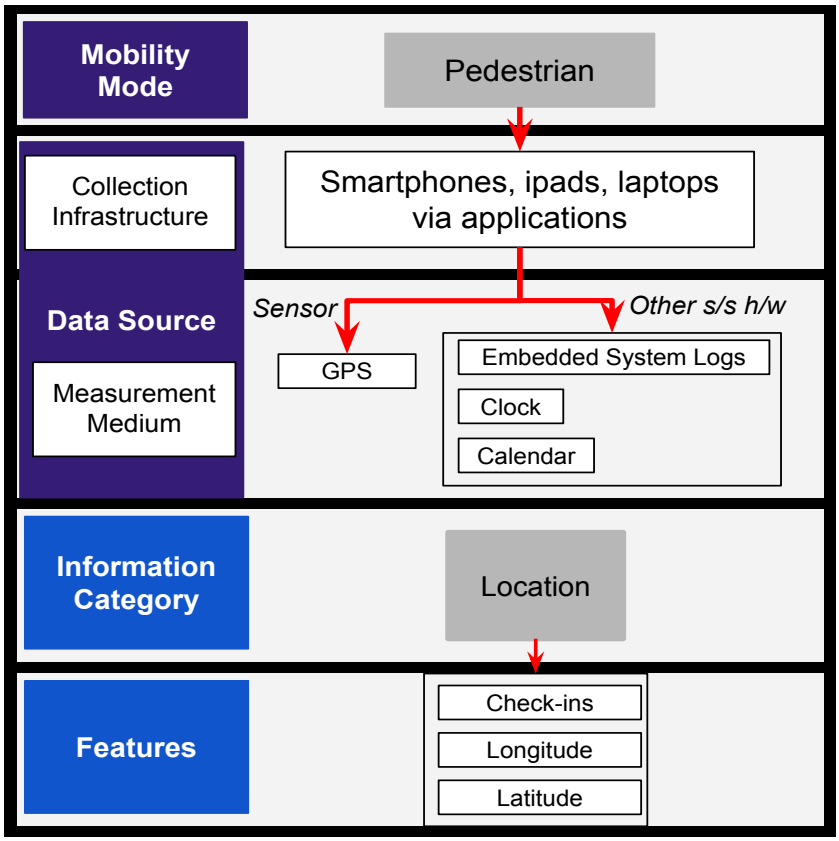

Figure 4.5: Brightkite Location Based Sensor Networks Traces.

traces from taxi cabs in San Francisco and (2) Derived from Brightkite, a location based social network application. Both datasets can be analyzed using the same mobility metrics. However, the taxi trace represents vehicular mobility patterns while the social network trace represents pedestrian mobility patterns. Figure 4.4 shows how the SF taxi dataset is classified using our taxonomy: It is a public transportation based vehicular trace. The data is collected using end user devices like smart GPS devices containing GPS locations, in the form of latitude-longitude tuples, and time/date information from the system clock. The San Francisco taxi trace associates each taxi with a unique taxi ID and a series of locations it visited over a period of time. The overall goal of the dataset is to provide locations that are commonly visited by the taxis in each region. The traces focus on spatial patterns categorizing locations based on unique cab identifiers. Such dataset graphs can provide us with various types of information. Figure 4.5 shows how the Brightkite traces are classified using our taxonomy. The dataset is pedestrian based, collected using applications running on end user devices like smartphones, iPads, or laptops. GPS location, region- and check-in

information are collected and used to identify social relations between various groups of people. The technology type used provides information about visited locations, along with their corresponding latitude, longitude, and timestamps.

Some recent real life use cases of the SF taxi traces in the last couple of years are: traffic forecasting using a temporal directed graph convolution network by studying temporal tendencies and periodicities in movement characteristics of vehicles [84]; Urban hotspot area detection using spatial clustering, mined from the taxi trajectory data, which directly represents an user's travel characteristics and the operational status of urban traffic [255]; optimizations to taxi-sharing and ride-sharing mechanisms and algorithms [241, 254, 234]; Optimizing passenger finding recommendation algorithms for taxis that suffer from load balancing issue [226]; Extracting the social/operational dynamics from taxi trips to study vehicle and passenger movement to and from its origin to improve road regulations and create new public transportation routes [171]; Optimizations to density blocking algorithms and trip demand merging strategies to propose an effective and scalable solution to the load-balancing problem [156]; studying historical trajectories to predict vehicle's next location [159]; Instantly discovering outlier trips from taxi trajectories [102].

Some latest use cases for the Brightkite dataset include: Analyze role of LBSN check-ins using social community detection methods to extract city structured communities (SoLoMo cities) to eventually detect behavioral events changing city's communities [103]; Human mobility prediction approach using movement patterns with k-Latest Check-ins (kLC) [92]; Friend relationship judgement methods based on improved gravity models, using residence distance and spatial temporal co-occurrence zone as an influence on friendship judgement [264]; Graph models like reconstruction graph model with fusion feature, designed for mining potential social connections with the help of users' spatial information, that will ultimately reduce the negative effect caused by the sparsity of social connection graph [263]; Personalized recommendation tool solutions for suggesting interesting and new locations to users by bridging preference-aware and social-based recommendations [74]; Naïve Bayes Prediction Model derived using Bayesian Theory for point of interest recommendations [121]; Creating relationship-protection algorithms based on location-visiting characteristics [245]; Markov chain based position prediction model using multidimensional

correction (MDC-MCM) [85]; Using advantages of regularity in human trajectories to model spatio-temporal information [80]; Identify social triad classes in a homophilic network to analyze the correlation between social triads and homophily [142].

Table 4.3: Open sourcing and privacy policies of popular uMA traces classified using our taxonomy

TRACES	SOURCE	LICENSE	ANONYMIZATION STRUCTURE	
T-Drive [6]	Kaggle	Attribution	Pseudo-anonymized	Non-aggregated
Crivello [61]	Unpublished	N/A	N/A	N/A
Apple Maps [29, 34]	data.world	Public Domain	Anonymized	Aggregated
Google Maps [39, 34]	Google	Public Domain	Anonymized	Aggregated
Descartes Lab [35]	GITHUB	Attribution	Anonymized	Aggregated
KCMD-DDH [18]	KCMD Data Portal	Public Domain	Anonymized	Aggregated
JRC(Europe) [5]	Urban Mobility Data Platform	Public Domain	Anonymized	Aggregated
GIM [32]	Flow Monitoring	Public Domain	Anonymized	Aggregated
Gowalla [7]	Snap Stanford	Attribution	Pseudo-anonymized	Non-aggregated
Brightkite [7]	Snap Stanford	Attribution	Pseudo-anonymized	Non-aggregated
Nsense [21]	Snap Stanford	Attribution	Pseudo-anonymized	Non-aggregated
Cabspotting [8]	CRAWDAD	Attribution	Pseudo-anonymized	Non-aggregated
Geolife [13]	Microsoft Research	Public Domain	Anonymized	Aggregated
NYC Mobility [24]	data.world	Attribution	Pseudo-anonymized	Non-aggregated
GRID Bikeshare [33]	data.world	Attribution	Pseudo-anonymized	Aggregated
Texas Mobility [26]	Austin Open Data portal	Public Domain	Anonymized	Aggregated
UILM [25]	data.world	Attribution	Pseudo-anonymized	Aggregated
GSMC [28]	CRAWDAD	Attribution	Pseudo-anonymized	Non-Aggregated
Flexran [30]	CRAWDAD	Attribution	Pseudo-anonymized	Non-Aggregated
KTH [15]	CRAWDAD	Attribution	Pseudo-anonymized	Non-Aggregated
BLEBeacon [27]	CRAWDAD	Attribution	Pseudo-anonymized	Non-Aggregated
HYCCUPS [12]	CRAWDAD	Attribution	Pseudo-anonymized	Non-Aggregated
Cambridge Haggle [19]	CRAWDAD	Attribution	Pseudo-anonymized	Non-Aggregated
Fire Dpt Asturius [17]	CRAWDAD	Attribution	Pseudo-anonymized	Aggregated
SocialBlueConn [16]	CRAWDAD	Attribution	Pseudo-anonymized	Non-Aggregated
Rome Taxis [14]	CRAWDAD	Attribution	Pseudo-anonymized	Non-Aggregated
SIGCOMM 2009 [9]	CRAWDAD	Attribution	Pseudo-anonymized	Aggregated
Commercial Seoul [11]	CRAWDAD	Attribution	Pseudo-anonymized	Aggregated
Chicago taxi [22]	Kaggle	Attribution	Pseudo-anonymized	Non-Aggregated
Pedestrian Louisville [20]	data.world	Public Domain	Anonymized	Aggregated
MHealthDroid [69]	UCI Machine Learning Repository	Attribution	Pseudo-anonymized	Non-Aggregated
NetAAL [67]	WNLAB	Attribution	Anonymized	Aggregated

4.3.4 Data privacy for open source datasets

Open source mobility data at the point of collection almost always contains personal identifiable information (PII) like the user's name, contact information, trip history with exact location precision , payment information etc. Before sharing this information with government agencies or research groups, it is important for the mobility operators to mask or remove such information from the datasets. An example of a privacy specification developed by the Open Mobility Foundation [41] is the Mobility Data Specification (MDS) [44, 38], which makes available a set of APIs to make anonymized mobility data available as an open source resource. MDS is specifically used for location data derived from vehicles and provides information like trajectories, popular visit points, wait times etc. Since this data is collected by the vehicle and not the user device, there is no PII revealed. An important aspect

of mobility data privacy is real-time data and its role in improving micro-mobility efficiency. Foundations that tackle mobility data privacy have collaborated with mobility operators to come up with standards and specifications to make such data available in anonymized or aggregated formats. Most of these standards come up with policies and non-infringement agreements that enforce good data-sharing practices without compromising on the privacy of PII.

Most of the open source datasets that we have accessed are from websites like CRAWDAD, which makes the users sign a nonexclusive, non transferable, data license agreement before getting access to any of the information, with the caveat that data is not redistributed [40]. The San Francisco taxi traces were part of CRAWDAD and so fall under their license. The Google Mobility Trend Report [39] follows the company's stringent privacy policy and provides information about the percentage rise and fall of Map requests to a given location, at no point making any PII available like an individual's location, contacts or movement. This dataset is derived from aggregated/anonymized sets of data from user's who have specifically turned on their Location History in the Google maps application. Google uses differential privacy to add noise to the datasets, which provides the same insights as real data without revealing any PII. The Stanford Brightkite dataset does not specifically talk about licenses, but the data is anonymized to the point where we can group a random set of user check-ins based on similarity in check-in patterns, but we cannot identify who each user in the group is.

Table 4.3 discusses the different sources, licenses, anonymization and aggregation standards of the various uMA traces. The *sources* column outlines the websites where the datasets are hosted. *Licenses* address the type of permissions required for distribution and reuse of these datasets. The different types of data licenses available are [50]:

- **Public Domain** The dataset has been dedicated to the public by waiving all rights to the research data worldwide under copyright law, including all related and neighboring rights, to the extent allowed by law.

- **Attribution** Appropriate credit is given where necessary by providing a link to the license or citations, and indicating if changes were made.

- **Share-alike** Remixing, transforming, or building upon the material, must include distributing your contributions under the same license as the original.
- **Non-commercial** Cannot use the material for commercial purposes.
- **Database Only** License applies to the database only and not its contents or data.
- **No Derivatives** No Derivative Works. Cannot alter, transform, or build upon this work.

The *anonymization* column describes if the data is completely or partially anonymized. Complete anonymization is when the patterns in a dataset cannot be traced back to the original users under any circumstances, while pseudo-anonymization means that even though the data is currently anonymized, they are ways to derive connections between the original user and the features, which can be a privacy concern in the case of datasets heavy on personal identifable information (PII). The *aggregation* column suggests if a dataset is aggregated or not aggregated. An aggregated dataset like the Google covid dataset, will have derived collective information for a group of individual records, while a non-aggregated dataset like Cabspotting will have datapoints for each individual user.

Chapter 5

Synthetic Generation using GANs

User mobility and activity (uMA) data reveals a number of aspects of user behavior and trends at different spatio-temporal scales which in turn provide invaluable information to guide the design, operation, and management of critical infrastructure, services and applications. Due to their diverse applications, spatio-temporal significance, and methodologies for collection, distribution and analysis, open source uMA datasets are riddled with a wide range of challenges including privacy preservation, computational and storage overhead, and representativity. As useful as it can be to have access to a large repository of user activity traces, catering to the different user needs could mean having to compliantly store different versions of the same dataset. To address these challenges, there has been significant work on heuristics and machine learning based uMA generative modeling along with data anonymization. However, existing models do not generalize well across uMA application categories and cannot be used to generate different types of uMA traces with different feature structures.

This chapter introduces uMAD, a Generative Adversarial Network (GAN) based pipeline for realistic uMA trace generation and analysis that is able to produce synthetic uMA datasets representative of different uMA application categories. The Highlights of this chapter include: **(a)** Comparison of existing GAN architectures to shortlist a high fidelity generative model across uMA application categories given reasonable computational resource utilization. **(b)** Evaluation of uMA trace generation performance using two types of metrics, namely application agnostic and application specific. As illustrated in Figure 5.1, uMAD

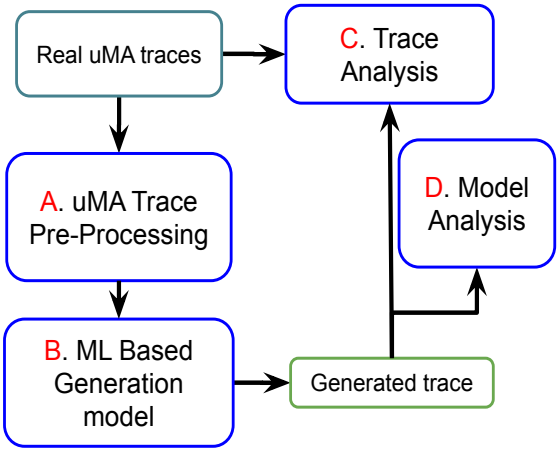

Figure 5.1: uMAD Overview

provides, in a single,integrated pipeline, uMA data collection, generation, and analysis capabilities.

There has been significant work on techniques to generate synthetic, yet realistic uMA datasets, including proposing heuristics and model based generation, as well as work on data anonymization. A recent and notable example is the use of Generative Adversarial Networks (GANs) [126] to generate uMA datasets. Due to their auto-tuning capabilities, less training data requirements and proven high accuracy, lately GANs are seen as the gold standard for generation of synthetic data, be it images or tabular data. While prior work in these areas target specific uMA categories like health and connectivity, to-date there is no research on GAN models that can generalize to different uMA applications and scenarios.

Our goal is to propose a GAN-based uMA data generation framework called uMAD, which can be used to generate synthetic, yet realistic uMA datasets for various classes of applications. To this end, we explore existing GAN architectures like Deep Regret Analytic Generative Adversarial Networks (DraGAN) [144], CramerGAN [76], and WGAN [62], TabGAN [65], TimeGAN [253] and CTABGAN Plus [265] to create synthetic realistic uMA data that can dynamically and accurately capture characteristics across the different uMA feature classes that the model is trained on. GANs have been extensively used in data-limited scenarios, because of their self-training semi-supervised learning ability to create new realistic data from noise. Our project uses open source real uMA datasets like the Google COVID lifestyle dataset, the San Francisco taxi location dataset, the Mhealth sensor

health dataset and an RSSI based connectivity dataset to train the GANs.

5.1 The uMAD Pipeline Overview

Studying spatio-temporal patterns in real user mobility and activity (uMA) datasets is an important part of being able to recreate real life user scenarios accurately. However, challenges associated with collection, distribution and generation of real uMA data, in addition to lack of access to a representative set of uMA datasets. make it an important problem to solve. Our pipeline aims to solve issues like (1)Extreme sensitivity of uMA data to change in experimental environment conditions, by allowing the user to generate a replica of the original dataset without having to simulate the experimental environment; (2) Vast diversity of uMA datasets, by providing a one model fits all approach for generation of data from various uMA categories; (3) Privacy associated with dissemination of personal identifiable uMA information, by generating datasets that preserve the patterns of the real data, while using random user identifiers.

1. *uMA Trace Pre-processing:* uMAD's dataset preprocessing component allows users to look at existing traces and categories that they can generate. This component is augmented with solutions for data cleaning and handling missing data, to help overcome the inaccuracies and errors that are common in raw uMA data. It will also include mechanisms to ensure collected uMA data adheres to data privacy and protection policies and regulations while mitigation performance overhead introduced by privacy regulation compliance.

   Another essential part of the pre-processing component is the storage of the collected uMA traces and their corresponding models for future access and querying. These traces are annotated to include various features/labels such as mobility mode, collection infrastructure, and measurement medium. These features help users select datasets that match their interests and needs. These features also guide uMAD's synthetic data generation and analysis components. In the future this module will also enable users to add new uMA datasets to the framework for future study and analysis.

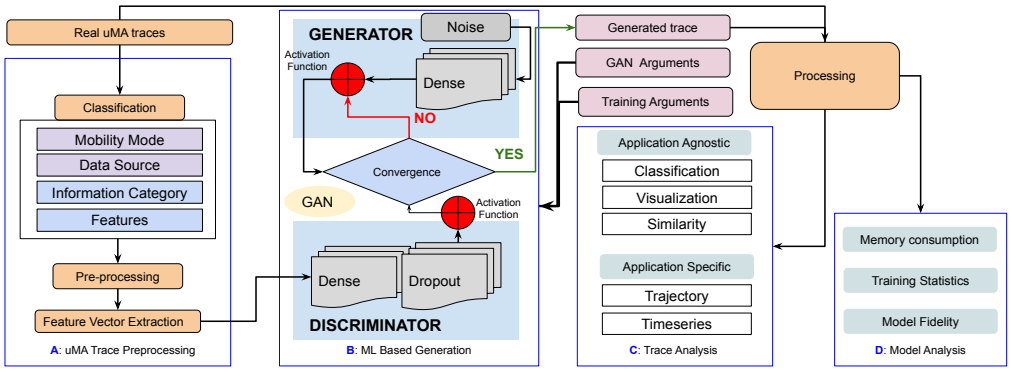

Figure 5.2: uMAD end to end framework pipeline

2. *ML-based generation:* One of uMAD's core components is its uMA data generation stage which handles the generation of realistic synthetic datasets. In uMAD, we aim to overcome the challenges faced by existing synthetic dataset generation solutions. In particular, uMAD's synthetic data generation will eventually allow users to control and specify the desired features and parameters of the generated dataset. For example, a user may choose the type of subject demographics, geographic location, collection infrastructure, and other features and parameters that are common in existing datasets. Additionally, uMAD synthetic dataset generation utilizes models that ensure compliance with data privacy and protection regulations. uMAD will also be able to flexibly control the features and parameters of generated datasets by utilizing a machine learning (ML) pipeline that we will design and develop. This pipeline learns the features and parameters of uMA datasets that have been stored and collected previously (including a bank of datasets that we have collected from public sources). After learning the features and parameters of selected uMA datasets, the ML pipeline utilizes GAN-based generative models to generate datasets that mimic real traces based on the parameters that the user specifies. As part of future work in this module, we plan to add capabilities such that a user can generate a combination of two traces, and also add new features and parameters as needed.. For example, consider a user interested in a realistic dataset that has properties A and B. If no real- or synthetic dataset has both properties A and B, then the user will have to rely on "suboptimal" datasets, i.e., datasets that do not completely match the user's needs. To address this limitation, uMAD will automatically learns features A and B from the pool of existing

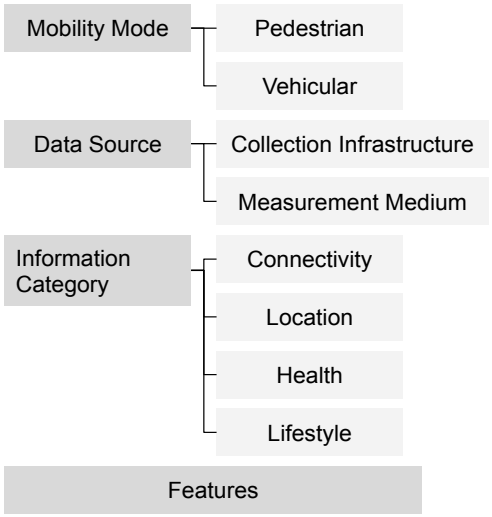

Figure 5.3: Classification Knobs

datasets, in which there exists datasets that either have feature A or B. Then, using the generative model, datasets containing both features plus all other realistic patterns of the datasets would be generated.

3. *Trace Analysis:* uMAD's Trace Analysis component will be used to perform various analytics on either real or synthetic datasets, including different types of visualization, trend analysis, similarity studies, and anomaly detection. This component is independent of the generation module and can be accessed by the user to analyze an existing or a new trace.

4. *Model Analysis:* This component conducts evaluation of the generative model and the generated traces according to relevant performance metrics. We augment and utilize the wealth of tools and solutions that provide information like memory utilization, size, processing capacity, statistical tests like KStest and CStest [47] and adapt them to uMA data and analysis.

5.2 uMA Trace Pre-Processing

The benefits of the vast diversity in open source uMA traces is followed by a very real challenge of determining a useful classification of these traces. uMA traces

have applications in fields ranging from urban planning [91], public health and healthcare [104], efficient transit and transportation [130], critical infrastructure planning and deployment [81], commerce, entertainment [108], network and connectivity [177, 228] etc. As mentioned in previous sections, open source uMA traces available on public websites like CRAWDAD [211], data.world [212], Kaggle [214] and GitHub [213] are feature-rich. Our framework uses a taxonomy proposed by King et al [143] to allow users to pick uMA categories or features types based on the granularity of the users preference.

Like the taxonomy, our framework will provide knobs with a bottom up approach starting from the features, data source or technology used to collect the traces, all the way up to the uMA mode i.e., stationary, pedestrian or vehicular, being represented by these traces. This feature is useful as it will allow users of different uMA expertise levels to be able to play around with the framework for their applications. For example a user that is new to the uMA field and wants to explore traces, can start by picking between pedestrian or vehicular traces, which would provide them with a condensed set of traces under those specific categories to begin with.

Details of our proposed uMA category knobs are provided in Figure 5.3.

- **Mobility Mode**, which refers to the user's mode of movement capture, namely stationary, pedestrian or vehicular.
- **Data Source** considers how the trace was collected and is further subdivided into:
  - **Collection infrastructure:** systems that host the devices used to collect data.
  - **Measurement Medium:** actual device/technology that generates the different measurements used to populate the dataset.
- **Information Category:** Application groups created by studying existing open source mobility traces.
- **Features:** Raw and derived information types generated in open source mobility traces.

The survey [143] uses these buckets to classify 31 well known uMA traces to then conclude that all these traces can be broadly categorized into two classes: Vehicular

and Pedestrian traces. Our reason for picking the taxonomy uMA classes is that they have extensive resources and metrics that can be used to measure quality of the original as well as generated traces. We have also extended the pedestrian category to include social networking traces to provide insights into datasets like Brightkite and Gowalla, with smaller feature sets.

- **Vehicular**: Vehicular uMA traces are traces collected using GPS and sensor devices connected directly to a mode of transportation like cars, bikes, buses, taxis etc. Under vehicular traces, we look at the San Francisco taxi trace, collected using end-user devices like a smartcar's GPS devices containing locations, in the form of latitude-longitude tuples, and time/date information from the system clock. The overall goal of the dataset is to provide locations that are commonly visited by the taxis in each region. The traces focus on spatial patterns categorizing locations based on unique cab identifiers. The taxi dataset has been divided into groups based on cab identifiers as labels.

- **Pedestrian**: Pedestrian uMA traces are traces collected using GPS and sensor devices on a user's device like smartwatches, smartphones and laptops. Companies like Microsoft, Google and Apple have extracted data from applications like Google and Apple Maps to analyze changes in mobility trends since the COVID-19 pandemic started in late 2019. The Google COVID Mobility Report, an example of pedestrian uMA traces, uses map requests generated via Google maps to showcase two months data with a day-wise percentage increase or decrease in visits to various categories of locations like parks, malls, residential areas, offices etc. For the Covid data, the labels are two different months

- **Social Networking**: This is an example of stationary traces purely based on check-in data collected via an application. Location based social network (LBSN) traces like Brightkite, Foursquare and Gowalla derive information from check-ins, providing users as edges and vertices as locations visited by that user. These datasets can be used to not only find out a user's travel network, but also to correlate similarities between likes and dislikes of users, based on similar trajectories.

For our experiments in this paper we have selected datasets from the four uMA information

categories:

- **Connectivity**: Connectivity and network traces provide information on signal strengths, packet transfer details, network usage details and timestamps. Mobility performance metrics like user pause probability, user arrival, departure probabilities heavily impact the performance of 5G cellular networks.

- **Location**: Traditional location or GPS traces provide basic latitude-longitude information and when combined with location specific mobility tags, have several applications in various spheres of mobility analysis. Mobility tags can include any type of information tied to location like behavioral patterns, foot traffic, choices, network usage etc.

- **Lifestyle**: Geosocial traces, referred to as lifestyle traces in the paper, derive human patterns using social network check-ins. The unique style of geosocial mobility traces can provide insightful information for urban planning and retail real estate to property owners and operators. Geosocial data has the potential to reveal the personality of neighborhoods in a city.

- **Health**: Movement and health tracking traces include information from sensors like heart rate monitors, oximeters, accelerators and magnetometers. One important application of these traces is in the healthcare industry to derive conditions from health features.

More details about the datasets are included in section 6.

The raw datasets were pre-processed using Python's pandas and sklearn libraries by performing data cleaning, handling missing values and outliers. Some of the challenges we faced with these real traces were managing categorical values through data encoding, dealing with missing values and outliers and model specific feature extraction/transformations we had to perform for each dataset category. Data preprocessing support includes:

- **Handling missing values**: One of the easiest ways to handle missing data is to remove the rows where there are values missing. This method is not ideal for uMA traces as we may lose important pattern information by omitting data points. Our pipeline

looks for missing values within the dataset and replaces those values with statistically equivalent data. For example, in the case of categorical data, the missing values are replaced with the mode value in the categorical column. While for numerical data, the missing values are replaced with the mean value of the numerical column.

- **Normalization**: We use sklearn's $normalize()$ function which rescales each column to have a unit norm. This scaling is independent of the distribution across the column. The function $normalize()$ provides a quick way to apply a vector space model [49] on a single array-like dataset, either using the l1, l2, or max norms. Given x is the vector of co-variates of length n and say that the normalized vector $y = x/z$, then the three options that can be used to describe z:

$$L1_{normalization} : z = \|x\|_1 = \sum_{i=1}^{n} |x_i|$$
$$\text{L2}_{normalization} : z = \|x\|_2 = \sqrt{\sum_{i=1}^{n} {x_i}^2}$$
$$\text{Max}_{normalization} : z = \|x\|_\infty = max|x_i|$$

- **Encoding categorical values**: There are several encoding functions available through sklearn like OneHotEncoder, OrdinalEncoder, LabelEncoder. We use the ordinal encoder to replace n unique column values with values ranging from 0 to $n-1$ values.

5.3 ML-Based Generation

We have included analysis of several GAN architectures like tabGAN, TimeGAN, CRAMERGAN, DRAGAN, WGAN and CTAB-GAN Plus etc, that have previously been used to generate tabular and time-series data like financial and healthcare data. We compared generation accuracy and model fidelity across these models for the different uMA categories like pedestrian, vehicular and social network. All of these models optimize different aspects of the GAN if we assume that we are looking for the equilibrium where the discriminator maximizes cost functions and the generator minimizes them. Our application-specific and application-agnostic experimental evaluation of uMAD generated traces confirms that model CRAMERGAN achieves both trace generality and fidelity with reasonable computation resource consumption.

More details about the GAN architectures are as follows:

- **TimeGAN** [253]: Time-series Generative Adversarial Networks was a novel framework proposed for generation of realistic time-series data in various domains by preserving temporal characteristics between features. In addition to the unsupervised adversarial loss on both real and synthetic sequences, they introduce a step-wise supervised loss using the original data as supervision, which forces the model to capture the step-wise conditional distributions in the data. They also introduce an embedding network to provide a reversible mapping between features and latent representations, thereby reducing the high-dimensionality of the adversarial learning space. The supervised loss is minimized by jointly training both the embedding and generator networks, such that the latent space not only serves to promote parameter efficiency—it is specifically conditioned to facilitate the generator in learning temporal relationships. The authors also generalize their framework to handle the mixed-data setting, where both static and time-series data can be generated at the same time. TimeGAN achieves consistent and significant improvements over state-of-the-art benchmarks like RCGAN, C-RNN-GAN, T-Forcing, P-Forcing, WaveNet and WaveGAN, in generating realistic time-series.

- **TGAN** [65] is a variation of a conditional GAN used to generate traces with uneven distributions, focusing on relational tables containing continuous and discrete variables. The authors use clustering on numerical variables to deal with the multi-modal distribution for continuous features. They add noise and KL divergence into the loss function to effectively generate discrete features. They observe that GANs can effectively capture the correlations between features and are more scalable for large datasets, and are able to show that the model can generate high-quality synthetic data to benefit data science.

- **Wasserstein-GAN (WGAN-GP)** [62, 36] was introduced as an optimization of traditional training techniques to improve the learning stability and avoid modal collapse. It does so by minimizing an approximation of the Earth Mover(EM) distance value, which helps with hyperparameter tuning. The original WGAN-GP architecture has been created for image generation and has been modified for tabular data. WCGAN-GP, is used for tabular data. It uses Wasserstein distance and Gradient Penalty to

reduce the occurrence of failure modes associated with GANs. WCGAN-GP is similar to WGAN-GP and the only change is where critic (discriminator) and generator are both conditioned on an extra information of class labels. In WCGAN-GP rather than classifying samples as real or fake, the critic predicts values that are large for real and small for fake samples.

- **DRAGAN** [144] views mode collapse as a result of the GAN min-max game converging to bad local equilibria. It studies the GAN training process as regret minimization, as opposed to the popular view that there is consistent minimization of a divergence between real and generated distributions. The authors hypothesize the existence of undesirable local equilibria in this non-convex game to be responsible for mode collapse and make an observation that these local equilibria often exhibit sharp gradients of the discriminator function around some real data points. They demonstrate that these degenerate local equilibria can be avoided with DRAGAN's gradient penalty scheme.I t solves these challenges using alternating gradient updates procedure (AGD). This enables faster training and improved model stability leading to generator networks with better model performance across different objective functions.

- **CRAMERGAN** [76]: Unlike the Kullback-Leibler (KL) divergence, which strictly measures change in probability, the Wasserstein metric reflects the underlying geometry between outcomes. The architecture relies on three properties of probability divergences that are essential requirements for machine learning modeling: sum invariance, scale sensitivity, and unbiased sample gradients in the GAN training phase. The Wasserstein metric possesses the first two properties but, unlike the Kullback-Leibler divergence, does not possess the third. Leveraging insights from probabilistic forecasting the authors propose an alternative to the Wasserstein metric, the Cramér distance, which they show that possesses all three desired properties, combining the best of the Wasserstein and Kullback-Leibler divergences. Both CRAMERGAN and DRAGAN have shown significant improvements over WGAN for certain image-based use cases, but WGAN is still widely used for tabular data.

- **CTAB-GAN Plus** [265] is an extension of WGAN-GP. It is a novel conditional

tabular GAN, that improves upon the current tabular generation state-of-the-art by adding downstream losses to conditional GANs for higher utility synthetic data in both classification and regression domains; using Wasserstein loss with gradient penalty for better training convergence; introducing novel encoders targeting mixed continuous- categorical variables and variables with unbalanced or skewed data; and training with DP stochastic gradient descent to impose strict privacy guarantees. The authors extensive evaluation of CTAB-GAN plus on data similarity and analysis utility against state-of-the-art tabular GANs, show that CTAB-GAN plus synthesizes privacy-preserving data with at least 48.16 percent higher utility across multiple datasets and learning tasks under different privacy budgets.

As part of this project, we will tweak hyperparameters for CRAMERGAN, which proves to generate the closest results to the original compared to the other older GAN models, like learning rate and training arguments like batch size, number of epochs for different datasets to try to further improve the accuracy of the models across different uMA application categories. We will also assess a newer 2022 model called CTAB-GAN Plus and its performance across uMA information categories.

5.4 Model and Trace Analysis

An important part of the uMAD generation pipeline is the analysis and validation of the quality of synthetic traces produced by uMAD's generative models. Work in this area has spanned various applications, for example in healthcare, wireless networks, location-based social networks, vehicular mobility etc. The generated traces go through a post processing stage where we perform normalization, dimensionality reduction and other identifier based datapoint transformations required during the analysis phase. We perform two types of analysis as part of the pipeline- Model analysis and trace analysis. We further divide the analysis into application specific metrics that are different for different trace categories like trajectory, network, health and time series analysis; and application agnostic metrics like classification, correlation, statistical similarity in distributions etc. The model analysis falls under the application agnostic analysis. We will read more about these analysis metrics in

Chapter 6 and Chapter 7.

Chapter 6

Experimental Methodology

This chapter outlines the experimental methodology to support our pipeline. We ran all our experiments using the Python 3.9 version and used standard GPUs assigned via google colab to train all our tabular GAN models. The standard GPU assigned by colab is NVIDIA's Tesla T4, which runs on a TU104 graphics processor. The TU104 graphics processor features 2560 shading units, 160 texture mapping units, and 64 ROPs, including 320 tensor cores which help improve the speed of machine learning applications. NVIDIA has paired 16 GB GDDR6 memory with the Tesla T4, which are connected using a 256-bit memory interface. The NVIDIA driver version for the T4 is 525.85.12 and the CUDA Version is 12.0.. All resource consumption results may change if the compute type used is changed.

6.1 Real uMA traces and their pre-processing

We have chosen 4 open source uMA traces, one from each information category and worked with smaller samples of the traces for the convenience of generating cleaner results.

	RSSI_1	RSSI_2	RSSI_3	RSSI_4	Target	Group	Paths
0	-0.90476	-0.48	0.28571	0.30	1	1	1
1	-0.57143	-0.32	0.14286	0.30	1	1	1
2	-0.38095	-0.28	-0.14286	0.35	1	1	1
3	-0.28571	-0.20	-0.47619	0.35	1	1	1
4	-0.14286	-0.20	0.14286	-0.20	1	1	1

Figure 6.1: The Ambient Assisted Living RSS Based pedestrian localization dataset.

These traces have been picked from open source websites like crawdad [8], snap [149] and the google covid mobility report[39] and have been cleaned, pre-processed and restructured using pandas libraries [97].:

- **Connectivity**: The Ambient Assisted Living RSS Based pedestrian localization dataset [67], referred as the **NetAll** dataset in the rest of the document, is a simple binary classification trace that provides an output of 1 when there has been a room change versus -1 if there has been no room change. Each datapoint contains temporal streams of radio signal strength (RSS) measured between the nodes of a WSN, comprising 5 sensors: 4 anchors deployed in the environment and 1 remote worn by the user. We used data from 100 out of the 314 users which came up to a size of 34 kilobytes.The **NetAll** dataset has a total of 314 csv files, each representing a temporal sequence of a single user trajectory. Preprocessing for this datasets included combining the RSSI information with its equivalent target (whether a specific sequence results in a room change for a user or not), group (different set of users) and path (where each group of users has 6 different paths to traverse). The dataset features are shown in Figure 6.1.

- **Location**: The epfl vehicular dataset [8], referred as the **SFTaxi** dataset in the rest of the document, contains mobility data of taxicabs in San Francisco, USA. They collect GP location information from around 500 taxis over thirty days. The dataset has four main features- 500 unique cab identifiers, co-ordinates in the form of a Latitude/Longitude tuple and Fare that is a binary feature that shows if a cab had a passenger at a particular location. The size of the dataset we are working with is approximately 460 megabytes The **SFTaxi** traces have approximately 800 Kilobytes of trajectory information associated with each identifier. The resultant dataset created from the original dataset contains a random number of trajectory data points each from a random subset of users. This was coded such that we can get a unique set of data points each time to train our models, so we can test the model fidelity across unseen groups of the datasets. The dataset features are shown in Figure 6.2.

	Cab_id	Latitude	Longitude	Fare	Timestamp
0	adkavy	37.61549	-122.38821	0	1213037028
1	adkavy	37.61562	-122.38849	0	1213036968
2	adkavy	37.61518	-122.39029	0	1213036903
3	adkavy	37.61393	-122.39508	0	1213036843
4	adkavy	37.60493	-122.38362	0	1213036783

Figure 6.2: The epfl vehicular dataset.

	sub_region_1	date	retail	grocery	parks	transit	work	residential
0	Alabama	2022-01-01	-41	-23	-15	-19	-39	11
1	Alabama	2022-01-02	-19	-3	-28	8	-14	7
2	Alabama	2022-01-03	-12	4	-24	-3	-33	13
3	Alabama	2022-01-04	-8	7	-20	0	-20	8
4	Alabama	2022-01-05	-5	11	-9	0	-19	7

Figure 6.3: The Google COVID Mobility Trend pedestrian dataset.

- **Lifestyle**: The Google COVID Mobility Trend pedestrian dataset [39], , referred as the **Google Covid** dataset in the rest of the document, is derived from traces generated by Google maps to highlight changes in movement trends across places like retail and recreation, groceries and pharmacies, parks, transit stations, workplaces, and residential. These changes show trends in response to changes in COVID policies. The dataset provides location wise percentage increase or decrease in hits/visits to the locations mentioned above. We are only working with data from the United States area and size of our training data is approximately 76 megabytes. The dataset features are shown in Figure 6.3.

- **Health**: The MHealthDroid based pedestrian dataset [69], referred as the **MHealth** dataset in the rest of the document, uses an agile development framework for mobile health applications to collect health data from $Shimmer2[BUR10]$wearable sensors like accelerometer, electrocardiogram, gyroscope and magnetometer. The data include body motion and vital sign recordings from users performing 12 different physical activities like standing still, sitting and relaxing, lying down, walking, climbing stairs, waist bends, elevating arms, bending knees, cycling, jogging, running and jumping. These movements are recorded via sensors attached to the user's right wrist, chest and left ankle. This dataset has 23 features, one for each sensor in every relevant axes and a label that contains values ranging from $0 - 12$ representing a unique activity. We have used data from only subject one and the size of the training data is around 31

	A_chest_x	A_chest_y	A_chest_z	electrocardiogram_1	electrocardiogram_2	\
0	-9.2153	-0.34416	0.958500	-0.121400	-0.079540	
1	-9.1456	-0.29805	0.457390	-0.092098	-0.075353	
2	-9.5118	-0.16988	0.405760	-0.071167	-0.050235	
3	-9.7535	-0.20278	0.423850	-0.066981	-0.041863	
4	-9.8728	0.20370	0.060438	-0.050235	-0.029304	

	A_ankle_x	A_ankle_y	A_ankle_z	Gyro_ankle_x	Gyro_ankle_y	...	A_rla_x	\
0	2.9926	-9.1750	1.3108	0.10575	-0.87054	...	-3.0175	
1	2.9275	-9.0989	1.0014	0.10575	-0.87054	...	-2.7585	
2	2.8605	-9.4745	1.0653	0.10575	-0.87054	...	-2.4723	
3	3.1992	-9.5925	1.1461	0.10946	-0.88180	...	-2.2207	
4	2.8502	-9.2746	1.2523	0.10946	-0.88180	...	-2.2331	

	A_rla_y	A_rla_z	Gyro_rla_x	Gyro_rla_y	Gyro_rla_z	Mag_rla_x	Mag_rla_y	\
0	-9.7849	1.5108	0.24118	-1.0575	-0.34267	-2.10980	-13.4960	
1	-9.4258	1.4245	0.21961	-1.0493	-0.34052	-1.22140	-14.4150	
2	-9.1188	1.1170	0.21961	-1.0493	-0.34052	0.04785	-13.1680	
3	-9.1493	1.0738	0.21961	-1.0493	-0.34052	0.60799	-11.0140	
4	-8.9392	1.0843	0.21961	-1.0493	-0.34052	0.62958	-8.8549	

	Mag_rla_z	label
0	9.4790	0
1	9.1450	0
2	8.0749	0
3	6.2595	0
4	4.4333	0

Figure 6.4: The MHealthDroid based pedestrian dataset.

megabytes. The dataset features are shown in Figure 6.4.

6.2 ML Model Implementations

A Brief comparison of the tabular GANs is summarized in Table 6.1. The TimeGAN, CramerGAN, DraGAN and WGAN architectures were selected from a Python library called ydata-synthetic [164], which has miscellaneous architectures created for generation of tabular data. TabGAN is a PyPi implementation of a conditional tabular GAN used to imbalance integer columns [66]. A more recent GAN architecture we explore, was CTAB-GAN Plus [266, 265] is an improved conditional GAN implementation that uses a combination of Wasserstein loss, gradient penalty and new continuous categorical encoders to deal with skewed data. For the analysis of the original and generated traces, we used several python libraries like sklearn [184], statsmodel [162], scipy [231], matplotlib [71] pyplot and table-evaluator [77] etc.

We used mobility metrics that are part of scikit-mobility [180] to perform trajectory analysis of the SF taxi traces, for the Timeseries COVID analysis we used the metrics from the Python library statsmodels.tsa [162], and for the RSSI based localization, we used simple

Table 6.1: Summary of all tabular GAN architectures that we tried for uMA.

MODEL	FEATURES	CURRENT APPLICATION	APPLIED TO UMA
WGAN [62]	Minimizes a reasonable and efficient approximation of the EM distance, Has the ability to continuously estimate the EM distance by training the discriminator to optimality	Images	Sensitive to hyperparameter tuning, lacks ability to capture modal distribution
WCGAN-GP [36]	Wasserstein distance/Gradient Penalty to reduce the occurrence of failure modes, critic (discriminator) and generator are both conditioned on an extra information of class labels	Finance, Census	Not privacy protected and cannot handle imbalanced data
DRAGAN [144]	New gradient penalty scheme to reduce modal collapse and imporve model performance across replacable objective functions	Images, Finance	Cannot handle imbalanced data
CRAMERGAN [76]	CRAMER distance instead of Wassertein distance, contains sum invariance, scale sensitivity, and unbiased sample gradients.	Images, Finance	Generated data does not map to real data
TIMEGAN [253]	Captures the stepwise conditional distributions, provide a reversible mapping between features and latent representations	Weather, Finance	Not generalizable across uMA applications, longer training times.
TGAN [65]	Variation of a conditional GAN with focus on relational tables containing continuous and discrete variables	Finance, Census	Longer training periods for larger datasets, Not privacy protected, Statistically inaccurate
CTAB-GAN Plus [265]	Uses additional classifier and information losses as added feedback to the generator	Finance	Privacy preserving, Handles both categorical and numerical values, as well as imbalanced datasets

plotting to visualize location of user within a 4D plane. All models used to test the synthetic data quality were compiled from Python's scikit-learn library. As part of our final open source framework, we will be compiling all the results generated using the aforementioned libraries into an analysis report of the original and generated datasets. Our experimental evaluation of the uMAD pipeline includes: (1) analysis of model accuracy with hyperparameter tuning, (2) application-agnostic evaluation of generated traces, and (3) application-specific evaluation of generated traces.

6.3 Parameters and Resources

We measure accuracy of the older CRAMERGAN model by tuning different hyperparameters like the number of epochs, learning rate, batch size, number of dense layers etc, to find the optimal model and then use the tuned model to compare against the latest CTBAGAN-Plus results. We use tensorflow's model.summary() and Python's memory profiler to provide a computation and storage estimate of the models before and after training, and CPU/GPU usage during training for the different applications. This information will

be derived from the number of trainable parameters, shape of outputs after each layer of up/down sampling. This information will be useful in optimizing the model to fit different computational needs for example heavy-weight servers vs IoT devices.

6.4 Application-Agnostic Analysis

These metrics can be used to analyze any given trace, irrespective of their uMA application type.

1. ***Correlations:*** Correlational analysis is a two variable statistical procedure that sets out to identify the mean value of the product of the standard scores of matched pairs of observations. This type of analysis is used to find out whether changes in one variable produce changes in another. It includes deriving linear correlations between the two distributions using metrics like the Pearson, Spearman and Kendall Correlation Coefficients. There are two kinds of correlation we are looking at in this project. First, we calculate correlations between the synthetic data and the real data distribution. Second, we calculate pairwise correlations between feature pairs in the real and synthetic datasets. The following are the types of correlation values we have looked at

   - **Pearson Correlation**: This test compares the mean value of the product of the standard scores of matched pairs of observations. It yields a number from range -1 to $+1$. Positive figures are indicative of a positive correlation between the two variables, while negative values indicate a negative relationship. Furthermore, the value of R_p represents the strength of the relationship. A Pearson's R_p near values 0 and 0.3 (or 0 and -0.3) indicates a weak relationship between the two variables. A Pearson's R_p near values 0.4 and 0.6 (or -0.4 and -0.6) indicates a moderate strength relationship between the two variables. A Pearson correlation coefficient of between 0.7 and 1 (or -0.7 and 1) indicates a strong relationship between the two variables. The Pearson correlation coefficient R_p is calculated using the following expression:

$$R_p = \frac{1}{n-1} \sum_{i=1}^{n} \left(\frac{X_i - \bar{X}}{s_x}\right)\left(\frac{Y_i - \bar{Y}}{s_y}\right)$$

Where x_i represents the values of the x variable in a sample, $x - bar$ indicates the mean of the values of the x variable, y_i indicates the values of the y variable, and $y - bar$ indicates the mean of the values of the y variable. S indicates the sum of squares of the x and y variables respectively, and n is the number of observations of x and y variables.

- **Spearman Correlation**: The Spearman's test is a non-parametric version of the parametric Pearson bivariate correlation coefficient. The Spearman's test is useful where the basic assumptions of linearity and continuous variables necessary to perform a Pearson's bivariate correlation analysis have not been met.The Spearman's test can be used to analyse ordinal level, as well as continuous level data, because it uses ranks instead of assumptions of normality. It follows similar rules for the range of values as Pearson Correlation. The following formula can be used to calculate Spearman's correlation:

$$R_s = 1 - (\frac{6\sum d^2}{n^3 - n})$$

Before calculating Spearman's correlation value, each column of data should be ranked by assigning the ranking 1 to the largest number in a column, 2 to the next largest value, 3 to the third largest and so on. Then, we find the difference in the ranks d. This is the difference between the ranks of the two values on each row, calculated by subtracting the ranking of the second value from the rank of the first. In the formula R_s is the Spearman's rank and n is the number of observations.

- **Kendell Correlation**: Kendall rank correlation, similar to Spearman's correlation coefficient is used to test the similarities in the ordering of data when it is ranked by quantities. Other types of correlation coefficients use the observations as the basis of the correlation, Kendall's correlation coefficient uses pairs of observations and determines the strength of association based on the pattern of concordance and discordance between the pairs. The formula used to denote the Kendell coefficient rank R_τ is:

$$R_\tau = \frac{2}{n(n-1)} \sum_{i<j} sgn(x_i - x_j) sgn(y_i - y_j)$$

The τ correlation coefficient returns a value of -1 to 1, where 0 is no relationship and 1 is a perfect relationship and -1 is negative relationship.

2. ***ML utility test:*** This uses well known classification and regression algorithms on the original and the synthetic traces. It uses original data as training and synthetic as test and vice versa to compare accuracy scores. We define the classification algorithms along with the results.

3. ***Similarity****:* Measuring the probability distribution of all input and output features helps identify the data patterns and trends. When training generative models, our goal is to minimize the error. The error can be minimized by knowing the current information loss by measuring the divergence, which will indicate how much information loss to be minimized when approximating a distribution. Some common techniques to calculate the similarity between two distributions are:

   - **Kullback–Leibler Divergence(KL Divergence)**: Called the relative entropy of distribution P with respect to distribution Q and quantifies the information lost when moving from P to Q. KL Divergence is measured as the expectation value of the logarithmic difference between the two probability distributions as computed with weights of P_i:

     $$K(P\|Q) = \sum_i P_i log(\frac{P_i}{Q_i})$$

     The range of values for the KL metric outcomes is from $[0, +\infty]$, where values near zero mean the outcomes are similarly distributed for the different facets and positive values mean the label distributions diverge—the more positive value, the larger the divergence.

   - **Jensen Shannon Divergence(JSD)**: A method, also known as Information radius or total divergence to the average, to measure the similarity between two probability distributions, P and Q. It is The symmetric version of the KL divergence. Based on KL Divergence, it is a bounded symmetrization of KL divergence but does not require the condition of absolute continuity. JSD can be represented using:

$$D_{JS}(P\|Q) = \frac{1}{2}(D_{KL}(P\|M) + D_{KL}(Q\|M))$$

$$M = (P + Q)/2$$

Where M is a mixed distribution. The values for JSD is bounded, and between $[0, 1]$ for log_2 and log_e, the value ranges from $[0, ln(2)]$, with values near zero indicating similarity between distributions and positive values indicating divergence in distribution. The bigger the value larger the divergence.

- **Kolmogorov-Smirnov (KS) Test**: It compares the actual and predicted cumulative densities of the data between two samples to determine if they come from the same one-dimensional distribution. Formula used to measure KS test:

$$D = Max|P(X) - Q(X)|$$

KS test continuous outcomes is $[0, +1]$, where values near 0 indicate the densities are evenly distributed between two samples and values near 1 indicate densities are not evenly distributed between two samples. The null hypothesis for KS test states that there is no difference between the two distributions and the alternative hypothesis states that the two datasets are from different continuous distributions. If the KS statistic is higher than the critical value, then the null hypothesis can be rejected and the two distributions are different or if $p - value$ is lower than a, where $a = 0.05$ or $a = 0.01$, then it is very probable that the two distributions are different. KS Test creates a test statistic using the maximum difference between the cumulative density function of two data distribution. The significance is calculated from that test statistic.

We can also use standard methods like Euclidean distance or Mean Absolute Percentage Error (MAPE).

4. ***Visualization:*** One of the most effective ways to depict any form of data is visually. Especially for large-scale uMA traces, it is easier to understand patterns from plots and images rather than from tabular data. All of the above information will be represented in the form of t-distributed stochastic neighbor embedding (TSNE) and Principal Component Analysis (PCA) plots wherever relevant. Both the above techniques are

dimensionality reduction techniques. Dimensionality is the major factor in any dataset. Humans arent good with visualization of more than 3-dimensional properly so we need to reduce the dimension of a dataset with a large feature set so we can then visualize them properly. Additionally, training an ML model on all the features would be computationally expensive. In our project we have used this technique to visualize the data in 3D as opposed to visualizing all the features of the real and generated dataset:

- **Principal Component Analysis (PCA)**: Most common dimensionality reduction for visualizing data. Converts n-dimensions of data into k-dimensions while maintaining as much information from the original dataset. Steps to calculate PCA are: calculate the mean of each column; center the value in each column by subtracting the mean column value; calculate covariance matrix of centered matrix; and calculate eigen decomposition of the covariance. Things to keep in mind with respect to PCA are in the case of uncorrelated features, the variance preserved is relatively low. Due to its global structure preservation property, neighborhood clusters may not be preserved. Finds it difficult to capture non-linear feature relationships. Cannot work for diverse scaled datasets, this can be overcome by standardizing the data. Is prone to outliers in the data.
- **T- distributed Stochastic Neighbor Embedding(t-SNE)**: It is a non-linear dimensionality technique. One major differences between PCA and t-SNE is that it preserves only local similarities whereas PCA preserves global similarities in a data distribution. Thing to bear in mind for TSNE are its non-deterministic nature makes it such that it has to be run multiple times with varying the values of parameters and pick what suits best for our dataset. Cluster sizes do not mean anything as the algorithm manipulates denser and sparser clusters to adjust into the lower dimensional space. Even if the parameter values suit our dataset, we should run the same algorithm with same value multiple times to make sure the shape does not change.

6.5 Application-Specific Analysis

Some uMA traces also have application specific analysis metrics like trajectory traces, timeseries traces etc.

1. **Trajectory Analysis:** is performed on mobility datasets that trace out paths taken by different identifiers. These are mathematical measures used to capture motion/travel patterns in users in uMA traces. Previous works summarize graph theory-based and velocity-based mobility metrics like speed angle rate, angle coefficient of variation, average trip length, degree of link changes, degree of network spatial distribution, degree of spatial accessibility, degree of temporal dependence and degree of node proximity. These metrics can provide insight to important patterns like similar behavior among user classes, points of interests, uniformity and randomness in visitation patterns. skmob [4] divides their trajectory metrics into individual and collective types, where individual metrics are per user basis and collective metrics are for a set of users. Examples of **Collective Metrics** are:

   - **Random Location Entropy**: Compute the random location entropy of the locations in a TrajDataFrame. The random location entropy of a location j captures the degree of predictability of j, if each individual visits it with equal probability, and it is defined as:

     $$LE_{rand}(j) = log_2(N_j)$$

     where N_j is the number of distinct individuals that visited location j.

   - **Uncorrelated Location Entropy**: Computes the temporal-uncorrelated location entropy of the locations in a TrajDataFrame. The temporal-uncorrelated location entropy $LE_{unc}(j)$ of a location j is the historical probability that j is visited by an individual u. Formally, it is defined as:

     $$LE_{unc}(j) = -\sum_{i=j}^{N_j} p_j log_2(p_j)$$

     where N_j is the number of distinct individuals that visited j and p_j is the historical probability that a visit to location j is by an individual u.

- **Mean Square Displacement**: Computes the mean square displacement across the individuals in a TrajDataFrame. The mean squared displacement is a measure of the deviation of the position of an object with respect to a reference position over time. It is defined as:

$$MSD = \langle |r(t) - r(0)| \rangle = \frac{1}{N} \sum_{i=1}^{N} |r^{(i)}(t) - r^{(i)}(0)|^2$$

  where N is the number of individuals to be averaged, vector $x^{(i)}(0)$ is the reference position of the i-th individual, and vector $x^{(i)}(t)$ is the position of the i-th individual at time t.

- **Visits per Location**: Computes the number of visits to each location in a TrajDataFrame.

- **Home per Location**: Computes the number of home locations in each location. The number of home locations in a location j is computed as:

$$N_{homes}(j) = |\{h_u | h_u = j, u \in U\}|$$

  where h_u indicates the home location of an individual u and U is the set of individuals.

- **Visits per time unit**: Computes the number of data points per time unit in the TrajDataFrame.

Examples of **Individual Metrics** are:

- **Radius of Gyration**: Computes the radii of gyration (in kilometers) of a set of individuals in a TrajDataFrame. The radius of gyration of an individual u is defined as:

$$r_g(u) = \sqrt{\frac{1}{n_u} \sum_{i=1}^{n_u} dist(r_i(u) - r_{cm}(u))^2}$$

  where $r_i(u)$ represents the n_u positions recorded for u, and $r_{cm}(u)$ is the center of mass of u's trajectory. In mobility analysis, the radius of gyration indicates the characteristic distance travelled by u.

- **K radius of Gyration**: Computes the k-radii of gyration (in kilometers) of a set of individuals in a TrajDataFrame. The k-radius of gyration of an individual u is defined as:

$$r_g^{(k)}(u) = \sqrt{\frac{1}{n_u^{(k)}} \sum_{i=1}^{k} (r_i(u) - r_{cm}^{(k)}(u))^2}$$

where $r_i(u)$ represents the $n_u^{(k)}$ positions recorded for u on their k most frequent locations, and $r_{cm}^{(k)}(u)$ is the center of mass of u's trajectory considering the visits to the k most frequent locations only. In mobility analysis, the k-radius of gyration indicates the characteristic distance travelled by that individual as induced by their k most frequent locations.

- **Random Entropy**: Computes the random entropy of a set of individuals in a TrajDataFrame. The random entropy of an individual u is defined as:

$$E_{rand}(u) = log_2 N_u$$

where N_u is the number of distinct locations visited by u, capturing the degree of predictability of u's whereabouts if each location is visited with equal probability.

- **Uncorrelated Entropy**: Computes the temporal-uncorrelated entropy of a set of individuals in a TrajDataFrame. The temporal-uncorrelated entropy of an individual u is defined as:

$$E_u nc(u) = -\sum_{j=1}^{N_u} p_u(j) log_2 p_u(j)$$

where N_u is the number of distinct locations visited by u and $p_u(j)$ is the historical probability that a location j was visited by u. The temporal-uncorrelated entropy characterizes the heterogeneity of u's visitation patterns.

- **Real entropy**: Computes the real entropy of a set of individuals in a TrajDataFrame. The real entropy of an individual u is defined as:

$$E(u) = -\sum_{T'_u} P(T'_u) log_2 [P(T^i{}_u)]$$

where $P(T'_u)$ is the probability of finding a particular time-ordered subsequence T'_u in the trajectory T_u. The real entropy hence depends not only on the frequency of visitation, but also the order in which the nodes were visited and the time spent at each location, thus capturing the full spatiotemporal order present in an u's mobility patterns.

- **Jump lengths**: Computew the jump lengths (in kilometers) of a set of individuals in a TrajDataFrame. A jump length (or trip distance) Δr is defined as the geographic distance between two consecutive points visited by u:

$$\Delta r = dist(r_i, r_{i+1})$$

where r_i and r_{i+1} are two consecutive points, described as a latitude, longitude pair, in the time-ordered trajectory of an individual, and $dist$ is the geographic distance between the two points.

- **Maximum Distance**: Computes the maximum distance (in kilometers) traveled by a set of individuals in a TrajDataFrame. The maximum distance d_{max} travelled by an individual $u$cis defined as:

$$d_{max} = max_{1 \leq i < j < n_u} dist(r_i, r_j)$$

where n_u is the number of points recorded for u, r_i and r_{i+1} are two consecutive points, described as a $(latitude, longitude)$ pair, in u's time-ordered trajectory, and $dist$ is the geographic distance between the two points.

- **Distance straight line**: Computes the distance (in kilometers) travelled straight line by a set of individuals in a TrajDataFrame. The distance straight line d_{SL} travelled by an individual u is computed as the sum of the distances travelled u:

$$d_{SL} = \sum_{j=2}^{n_u} dist(r_{(j-1)}, r_j)$$

where n_u is the number of points recorded for u, r_{j1} and r_j are two consecutive points, described as a $(latitude, longitude)$ pair, in $u's$ time-ordered trajectory, and $dist$ is the geographic distance between the two points

- **Waiting Times**: Computes the waiting times (in seconds) between the movements of each individual in a TrajDataFrame. A waiting time (or inter-time) by an individual that u is defined as the time between two consecutive points in $u's$ trajectory:

$$\Delta t = |t(r_i) - t(r_{i+1})|$$

where r_i and r_{i+1} are two consecutive points, described as a $(latitude, longitude)$ pair, in the time-ordered trajectory of u, and $t(r)$ indicates the time when u visits point r.

- **Number of Locations**: Computes the number of distinct locations visited by a set of individuals in a TrajDataFrame.
- **Home Location**: Computes the home location of a set of individuals in a TrajDataFrame. The home location $h(u)$ of an individual u is defined as the location u visits the most during nighttime:

$$h(u) = argmax_i|\{r_i|t(r_i)\epsilon[t_{startnight}, t_{endnight}]\}|$$

where r_i is a location visited by u, $t(r_i)$ is the time when u visited r_i, and $t_{startnight}$ and $t_{endnight}$ indicates the times when nighttime starts and ends, respectively.

- **Maximum Distance from Home**: Computes the maximum distance (in kilometers) traveled from their home location by a set of individuals in a TrajDataFrame. The maximum distance from home $dh_{max}(u)$ of an individual u is defined as:

$$dh_{max}(u) = max_{1 \leq i < j < n_u} dist(r_i, h(u)))$$

where n_u is the number of points recorded for u, r_i is a location visited by u described as a $(latitude, longitude)$ pair, $h(u)$ is the home location of u and $dist$ is the geographic distance between two points.

- **Number of Visits**: Compute the number of visits (i.e., data points) for each individual in a TrajDataFrame.
- **Location frequency**: Computes the visitation frequency of each location, for a set of individuals in a TrajDataFrame. Given an individual u, the visitation frequency of a location r_i is the number of visits to that location by u. The visitation frequency $f(r_i)$ of location r_i is also defined in the literaure as the probability of visiting location $r_i$by u:

$$f(r_i) = \frac{n(r_i)}{n_u}$$

- **Individual Mobility Network**: Computes the individual mobility network of a set of individuals in a TrajDataFrame. An Individual Mobility Network (aka IMN) of an individual u is a directed graph $G_u = (V, E)$, where V is the set of nodes and E is the set of edges. Nodes indicate locations visited by u, and edges indicate trips between two locations by u. On the edges the following function is defined:

  $$\omega : E \to N$$

  which returns the weight of an edge, i.e., the number of travels performed by u on that edge.

- **Recency Rank**: Computes the recency rank of the locations of a set of individuals in a TrajDataFrame. The recency rank $K_s(r_i)$ of a location r_i of an individual u is $K_s(r_i) = 1$ if location r_i is the last visited location, it is $K_s(r_i) = 2$ if r_i is the second-last visited location, and so on.

- **Frequency Rank**: Computes the frequency rank of the locations of a set of individuals in a TrajDataFrame. The frequency rank $K_f(r_i)$ of a location r_i of an individual u is $K_f(r_i) = 1$ if location r_i is the most visited location, it is $K_f(r_i) = 2$ if r_i is the second-most visited location, and so on.

2. **Time Series Analysis:** Time series properties include levels (best values, max, min for the series), trends(linear change in series levels over time) and seasonality (repeated patterns across different periods of time). Understanding these properties will help the user model specific uMA environments across different time scales (daily, monthly, yearly or over a few years). They also provide insight into the kind of pre-processing a dataset needs. For example, a dataset with a regulated increase/decrease in trend will not require as much smoothing as compared to a dataset with random trend patterns. Overall, time series analysis is quite important for uMA traces as they help predict future behavior under different environmental conditions.

We use trajectory analysis for the SF taxi traces and the time series analysis for the covid traces respectively. Because of the nature of our network and health traces, the best possible tests for them would be visualization and classification numbers.

Chapter 7

Experimental Results

The chapter gives an overview of the multi-GAN architectural results showing that among the older GAN architectures, that are part of ydata-synthetic, CRAMERGAN outperforms the others in overall performance across the different uMA categories. We then perform hyperparameter tuning on CRAMERGAN to further improve the generation performance. We also bring in a newer 2022 wild-card model CTAB-GAN Plus [265] and compare the quality of the tuned CRAMERGAN generated traces with CTABGAN Plus generated traces. CTABGAN Plus is a new conditional tabular GAN architecture that improves generated synthetic data quality by adding a protection from attacks on personally identifiable information. It has three enhancements: It introduces a new encoder that specifically deals with variables that are of mixed continuous and categorical types; It uses Wasserstein distance plus gradient penalty to improve stability of GAN training; It also reduces algorithm complexity introduced in multi-discriminator WGAN variants, created to overcome training challenges, by switching to a single discriminator solution. Because of its effective privacy handling of personal identifiable uMA information (PII) and improved statistical synthetic data quality, CTAB-GAN makes for a compelling choice for uMA GAN generation.

Model	Taxi-Lat	Taxi-Lon
Original	-122.4	37.8
WGAN	-126	36.5
CRAMER	-122.7	37.7
DRAGAN	-122.1	37.1

Table 7.1: Mean Feature Values

7.1 Older GANs

Since uMA data is represented in tabular formats, the preliminary analysis of the older GANs was performed on models in a Python library called ydata-synthetic [164] which has miscellaneous architectures created for generation of tabular data. We have higlighted esults that are similar across uMA categories are highlighted using the **SFTaxi** dataset, and for instances where the results are different, we use the **SFTaxi** and the **Google Covid** dataset. We calculate similarity in distributions by using correlation and general distribution statistics like mean value. Correlation is a basic mathematical metric that can quickly provide us with a comparison of relations between the features of the original versus generated traces.

Pearson correlation calculates the linear relationship, whereas Spearman correlation calculates the monotonic relationship between two features. Both correlation values lie between -1 to 1 with values closer to 0 indicating that the features are not related versus values closer to $+1$ or -1 showing positive or negative correlation. All correlation values for the **SFTaxi** datasets represent correlation between latitude and longitude values, while for the **Google Covid** timeseries dataset, the correlation is between time and the corresponding feature.

From Table 7.2, traces generated by CRAMERGAN have the closest correlation to the original dataset in case of the **SFTaxi** data, while in Table 7.3 for the **Google Covid** dataset, only certain features are highly correlated. The mean value of each feature from

Model	Pearson	Spearman
Original	0.47	0.64
WGAN	0.97	0.86
CRAMER	0.36	0.66
DRAGAN	0.27	0.21

Table 7.2: Correlation between features across real and synthetic datasets

Model	Retail	Grocery	Parks	Transit	Work	Home
Pearson						
Original	0.31	0.19	0.09	0.05	0.26	-0.2
WGAN	0.02	0.02	0.018	-0.01	0.02	0.02
CRAMER	-0.04	-0.05	-0.03	-0.01	-0.01	-0.017
DRAGAN	0.04	-0.03	0.08	-0.07	-0.001	-0.15
Spearman						
Original	0.32	0.18	0.1	0.06	0.27	-0.2
WGAN	0.01	0.01	0.02	-0.01	0.02	0.01
CRAMER	-0.04	-0.06	-0.03	-0.01	-0.001	-0.02
DRAGAN	0.03	0.02	0.1	-0.08	-0.01	-0.13

Table 7.3: Correlation between features across real and synthetic datasets: COVID traces.

the original and synthetic traces helps us identify if the values generated are in the same range as the original features. In Tables 7.1, CRAMERGAN mean values are closest to the values of the original trace. The most commonly used visualization techniques are t-SNE

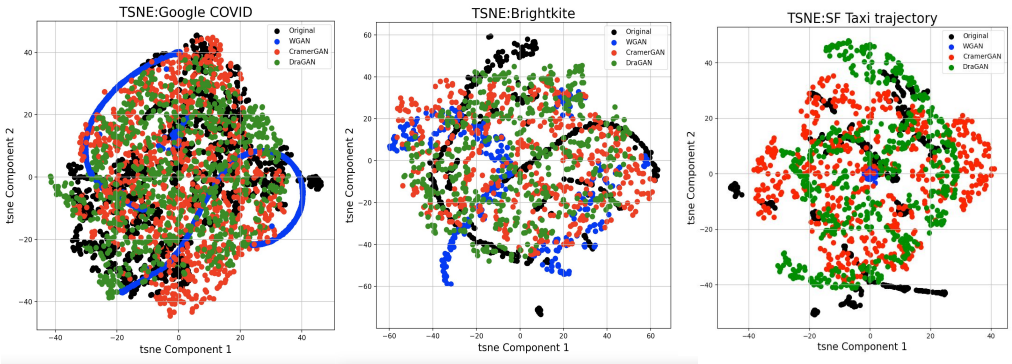

Figure 7.1: T-SNE visual comparisons between original and synthetic traces.

and PCA analysis on the original and generated traces, to study how close our generated data is to the original, in a two-dimensional space. Due to lack of space, we have only captured the TSNE plots in the paper, as seen in the figure 7.1. T-SNE captures the overall similarity between the original; and generated traces better than PCA, because of the KL divergence based objective function, which allows it to preserve local relationships between variables better. Additionally, it also handles outliers better than PCA. As we can see from the figures, for all of the trace categories, CRAMERGAN and DRAGAN, outperform WGAN.

From our experiments, we deduce that although CRAMERGAN outperforms the other GAN architectures overall, for time series based data it does not have the best performance. Since GANs focus heavily on trends within a timeseries, for features where

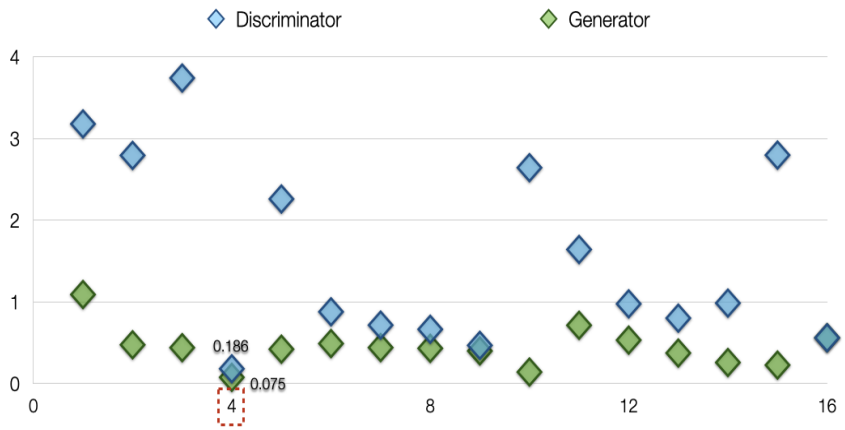

Figure 7.2: Performing Bayesian search to find the optimal set of hyperparameters.

the series contains a significant number of outliers, the results are inconsistent across the entire series. This motivates the need to explore hyperparameter tuning of CRAMERGAN and newer GAN models from 2022 to see if we can find a better fit across uMA categories.

7.2 Hyperparameter Tuning

Thinking of a machine learning model as a mathematical model which can learn several parameters from the internal structures of a given dataset, hyperparameters are values that can be tweaked to enhance the learning capabilities of the ML model. For example, a smaller learning rate value will create a highly sensitive model, which will produce unstable model results. Given a set of hyperparameters, finding the best combination of parameters that will create the best model is a search problem. Among existing search techniques [120]:

- GridSearch which derives the best hyperparameter combination by going over values in a grid. This can be computationally intensive for larger hyperparameter sets, since the model goes over every intermediate sets of parameters
- Randomised Search solves the issue with GridSearch by only optimizing the model for a finite set of parameter combinations.
- Bayesian Search's tuning algorithm bases its parameter selection on scores from the previous round by choosing only the relevant search space and discarding the ranges that will most likely not deliver the best solution.

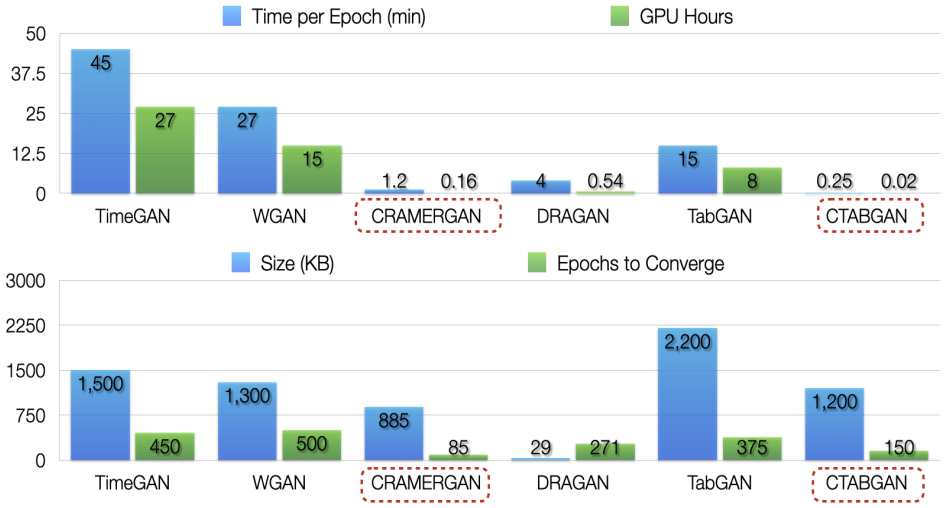

Figure 7.3: A comparison of resources used by the different GAN architectures.

For our pipeline, we used the Bayesian Search function that is part of Python's skopt library [190]. Figure 7.2 shows loss values on the y-axis and each number on the x-axis represents a unique set of hyperparameters. For a given combination we have plotted the critic and the generator loss for CRAMERGAN. The best model scores were generated by the combination 4 with values: Noise dimension 32, learning rate 0.0001, batch size 128, number of dense layers 1000. Since, there was no significant change in the CRAMERGAN generation performance after hyperparameter tuning, we proceeded to analyze generated data using CTABGAN Plus.

7.3 Multi-GAN resource comparison

Understanding the resource utilization of each of the GAN architectures plays a key role in the model selection process for our pipeline. Since the model is a part of a framework that will eventually be open sourced, we ideally want to select a model that achieves adequate trade-off between memory footprint, GPU utilization, train time per epoch and number of epochs combination, along with providing a high quality of synthetic traces. Figure 7.3 summarizes resource consumption for the different models. The memory size of each model before training is approximately 300 Kilobytes. The size in kilobytes, in the image, is the size of a model after training. For instance, at 29 Kilobytes, DRAGAN has the smallest trained model size out of all the models, with models CTAB-GAN plus

and CRAMERGAN having sizes around 1200 to 1300 kilobytes respectively. The time per epoch represents time taken to run a single epoch, with a set batch of training samples, to train the discriminator. Numbers of epochs in the Table represent the number of runs required for both the GAN models to converge. The memory footprint provides us with an idea of the space we would need to store these different models in our framework. GPU Hour represents time taken for the model to be trained in hours. Based on the results from Figure 7.3, DRAGAN yields superior computational resource performance of all the models, followed by CRAMERGAN and CTAB-GAN plus. In the previous sections, we already proved that CRAMERGAN outperforms DRAGAN when it comes to generation accuracy and synthetic trace quality. In the following sections, we will focus on results computed on traces generated by CTAB-GAN plus.

7.4 Application Agnostic Analysis

These metrics can be used to perform a quality check on the synthetic traces irrespective of the uMA application types.

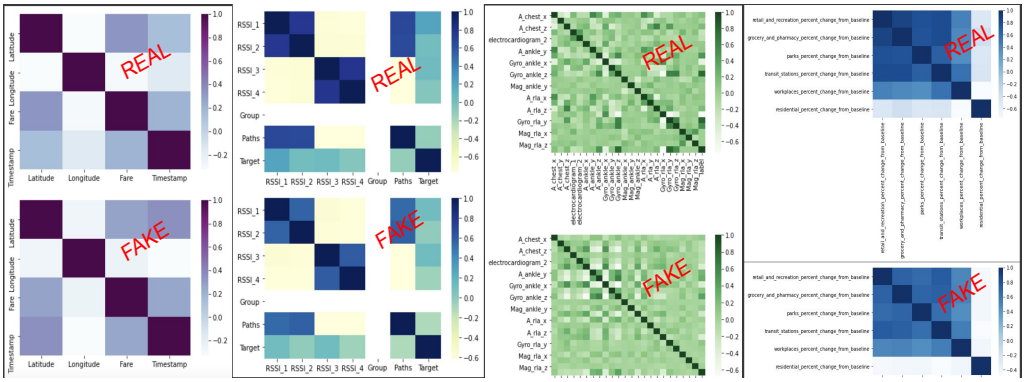

Figure 7.4: Pairwise correlations between variables for datasets under each of the uMA categories.

Correlation [78] is a bi-variate analysis that measures the strength of association between two variables and the direction of their relationship. The most widely used correlation metric is the Pearson coefficient for normally distributed data, which makes assumptions for the linearity (straight line relationship between two variables) and the homoscedasticity (data is equally distributed around the regression line). Pearson coefficient is also affected by outliers. Spearman and Kendall coefficients have most of the same assumptions as Pearson,

		Life		Loc		Health		Conn
	F1-R	F1-F	F1-R	F1-F	F1-R	F1-F	F1-R	F1-F
DT	0.87	0.55	0.25	0.35	0.60	0.33	1.00	0.91
LR	0.91	0.70	0.37	0.38	0.79	0.48	0.69	0.66
MLP	0.92	0.71	0.34	0.25	0.87	0.61	0.95	0.91
RF	0.63	0.24	0.46	0.89	1.00	0.49	0.99	0.99

Table 7.4: ML utility scores for Decision Tree (DT), Multi-layer Perceptron (MLP), Random Forest (RF) classification and Logistic regression (LR) models for the four uMA categories Lifestyle, Location, Health and Connectivity

except they work for data that doesn't always have a normal distribution. We have used Kendall's coefficient to accommodate for evaluation of unseen datasets with skewed values, where we may not know the distribution before the user runs the pipeline modules. Figure 7.4 shows pairwise Kendall correlation coefficients for datasets under the four uMA categories. From left to right on the Figure 7.4 we have uMA categories Location, Connectivity, Health and Lifestyle, with the top heat maps representing the real data and the bottom heat maps representing the fake data. As we can see from the images the synthetic datasets have managed to preserve the pairwise correlations of the original datasets across all uMA categories.

ML Utility tests Datasets across the four uMA categories location, connectivity, health and lifestyle are used to train well known ML models like Decision tree, MLP, random forest classifiers, and logistic regression model. Each model was trained with the original dataset and tested on samples from the corresponding generated trace and vice versa. We calculated F1-scores for each of the cases. F1-R are scores when the model was trained on the real dataset and F1-F is the score when the model is trained using the generated dataset. F1-score is a metric that combines precision and recall. In our use cases F1-scores closer to 1 represent the test datasets fitting better with the trained model. High F1-real scores suggest that the generated dataset has values that can be classified correctly using a model trained on the real dataset and vice versa for F1-fake scores. The ideal scenario is if the F1-R and F1-F scores are similar for a given dataset, which happens in the case of the location and the connectivity datasets. For the lifestyle and health datasets, the difference in the F1-F and F1-R scores, can be used to derive that the generated traces do not contain as many feature combinations as the real datasets. The high F1-R score for these two cases means the generated dataset can be classified accurately using models trained on the real dataset, but

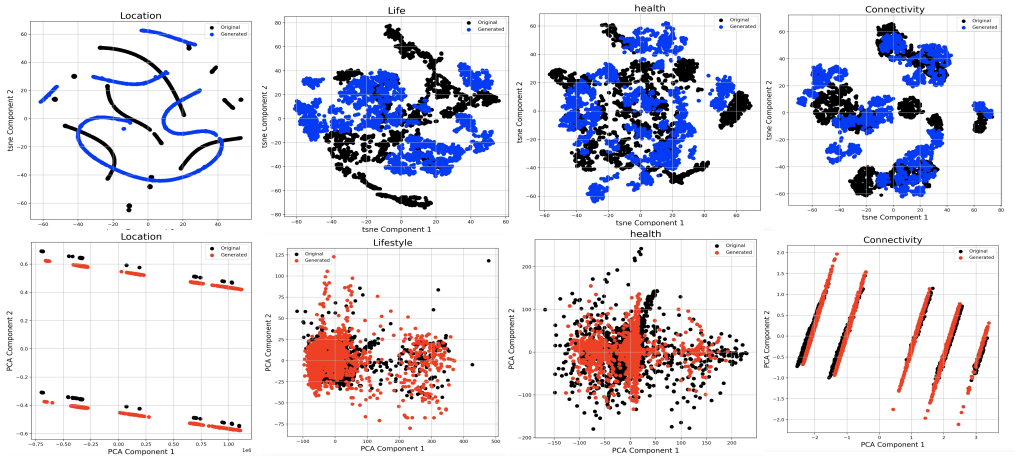

Figure 7.5: PCA and TSNE dimensionality reduction to compare real versus fake data.

the vice versa does not hold true.

Visualization: Principal Component Analysis (PCA) and t-distributed stochastic neighborhood embedding (TSNE) are both unsupervised dimensionality reduction techniques. Where PCA is linear while TSNE is non-linear. Between TSNE and PCA, the former is the more successful technique because of its nature of preserving cluster and local structure of a dataset and also its efficiency in dealing with outliers. TSNE is non-deterministic, while PCA is deterministic, which means that a TSNE representation will change for different minimas of KL divergence, whereas for PCA we constantly get the same output. TSNE is also known to not deal well with incomplete data. We have provided both types of charts in Figure 7.5, with the charts in blue showing TSNE components and red showing PCA components. All charts show that components derived from the synthetically generated data is very close to the components derived from the real data in the case of Both TSNE and PCA.

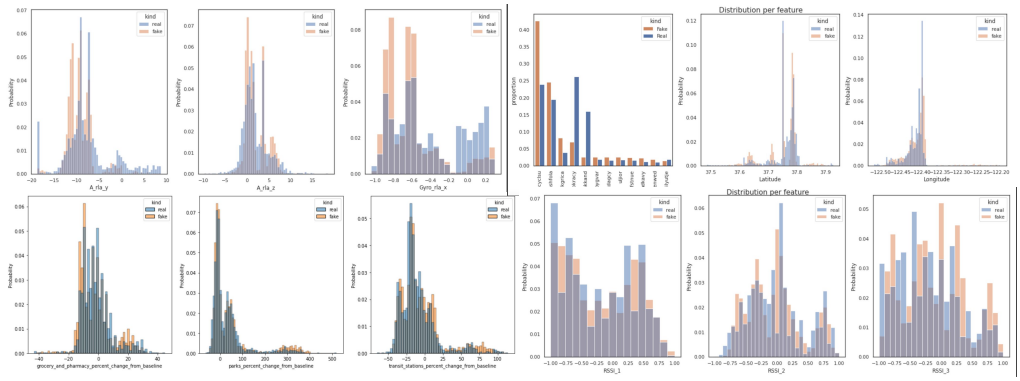

Figure 7.6: Comparison of features-wise class distribution across fake and real datasets

	Lifestyle	Location	Health	Connectivity
Average Accuracy	0.9465	0.8305	0.8714	0.9907
Similarity Score	0.9190	0.8368	0.7270	0.9981
1-MAPE estimator	0.1092	0.0158	0.1107	0.0581

Table 7.5: Similarity Scores

Feature Distribution: Figure 7.6 shows the probability of occurrence of each class datapoint within a distribution, for a given set of features across the real and synthetically generated data. The information in this figure helps us visualize whether the model is generating datapoints with similar value ranges as the real dataset.

Similarity Statistics: The similarity score is a combined score of overall accuracy, correlation between fake and real columns, 1- MAPE estimator results and the 1- MAPE 5 PCA components. The mean absolute percentage error (MAPE), or the mean absolute percentage deviation (MAPD) is used to measure accuracy of a prediction system. A MAPE score is usually between 1 to a 100 percent, where a lower score is better. In general a MAPE score less than 20 percent is considered a good score. Our MAPE scores across the uMA categories are approximately around the $1 - 10$ percent range. The average accuracy is an overall accuracy across the previous models, combining scores for models trained with both fake and real data.

7.5 Application Specific Analysis

This section outlines metrics for the different types of applications, trajectory based application traces like the SF taxi location traces, time series traces like the COVID lifestyle trace and $4D$ representations of RSSI signatures for the connectivity traces.

For the health dataset used in this use case, activity classification models are the best metric to be used. Since we have highlighted those in the application agnostic section, we will not review them again in this section.

Trajectory Analysis: As we mentioned before we use Python's scikit-mobility package [180] which was created to perform human mobility analysis. It allows us to extract mobility metrics from trajectory data, both at an individual and collective level and then evaluate the trajectories as well. Figure 7.7 highlights some individual mobility metrics of

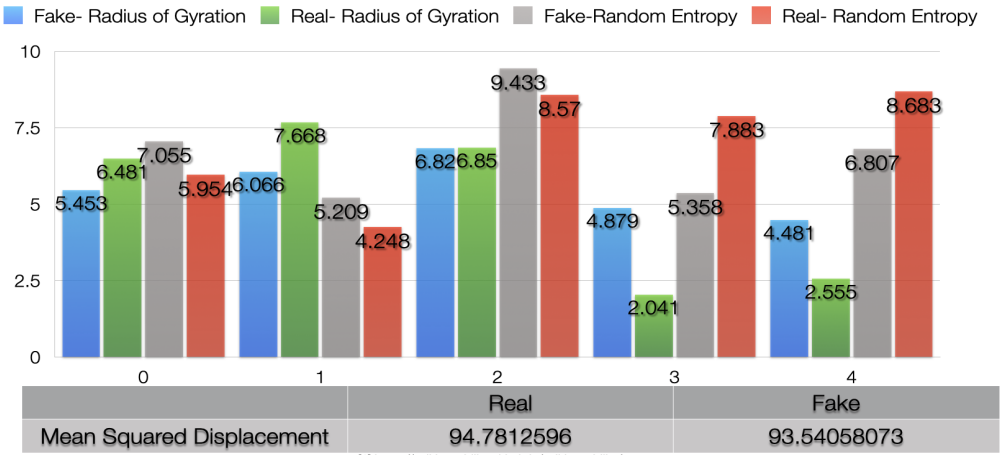

Figure 7.7: Individual and Collective metrics for trajectory analysis

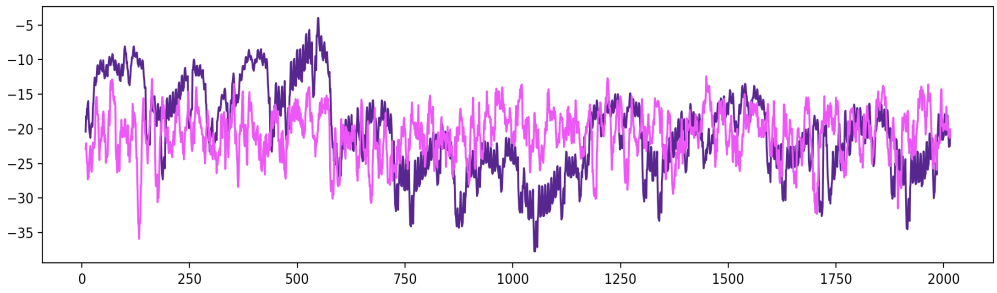

Figure 7.8: Real versus fake rolling statistics charts, the dark purple plot is for the real and the light purple is for the fake work feature distribution.

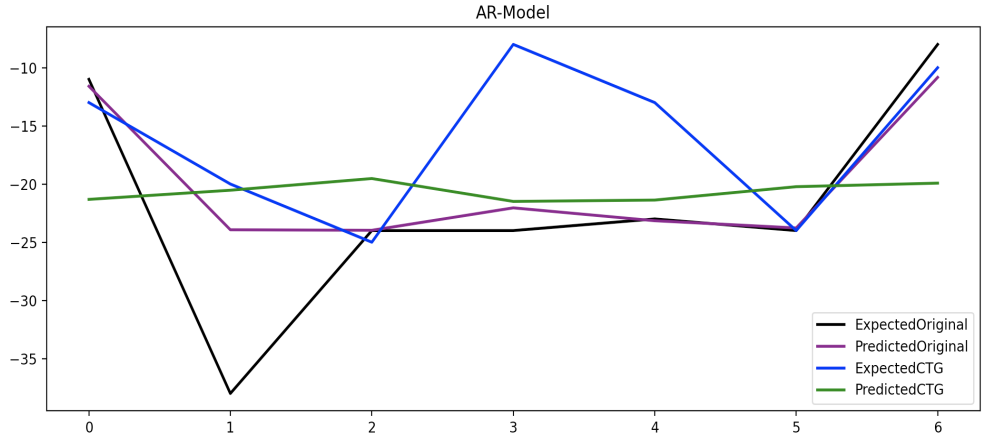

Figure 7.9: Time Series forecasting using Autoregressive model.

the real and fake datasets like the radius of gyration, which is the distance from the center of mass of a body at which the whole mass could be concentrated without changing its moment of rotational inertia about an axis through the center of mass; and the random entropy, which is the degree of predictability of locations visited by a user.

The figure also highlights collective metrics like mean squared displacement, which is a measure of the deviation of the position of an object with respect to a reference position over time. **Timeseries Analysis** Time-series data is a sequence of data points recorded across time intervals. The COVID dataset is sourced from the Google Mobility Report [39] and is an excellent example of a times series uMA dataset. Time Series analysis includes visualization of features with respect to time, statistical visualization, observing stationarity in the distribution, model building to test forecasting capacity of the distribution etc. Since time series analysis is used to study the effect of time of each feature, we have shortlisted the *percent change in work visits* during COVID as our representative feature. Figure 7.8 shows a plot of rolling mean value for the work feature against time, for both real and generated traces. We see that trace generated by CTAB-GAN plus accurately captured patterns of the real trace.

Stationarity of a series is an important time series metric, where the statistical properties like mean, variance, covariance do not vary with time. It is important for a series to be stationary as it makes for more precise statistical model predictions. Non-stationary series need to be converted into stationary series before working with them. Two statistical

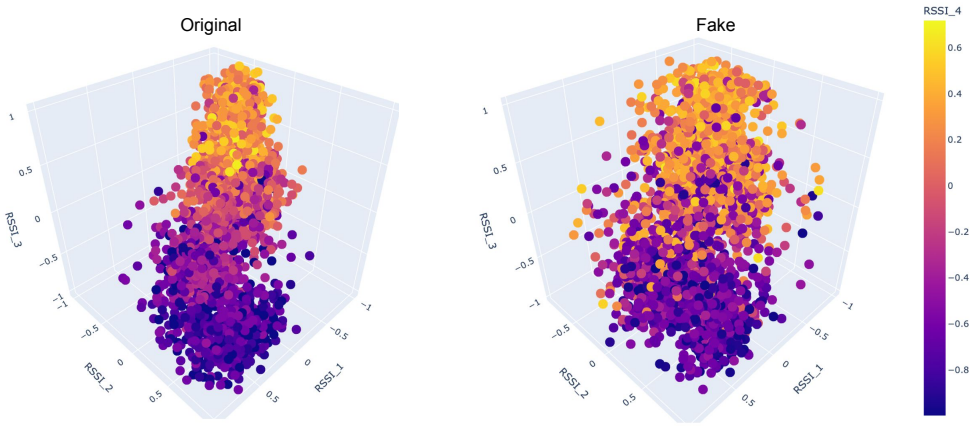

Figure 7.10: Real versus Fake RSSI Signatures on a $4-D$ plane .

tests Augmented Dickey-Fuller (ADF) Test and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test can be performed to check for stationarity.

- ADF: If the Test statistic is less than Critical Value and p-value is less than 0.05 time series is stationary.
- KPSS:If the Test statistic is less than Critical Value and p-value is less than 0.05 time series is non-stationary.

For our given original and generated series, ADF Critical value is approximately -2.567, ADF Test statistics is -3.31 and ADF P-value is approximately 0.01404. For our given original and generated series, KPSS Critical value is 0.347, KPSS Test statistics is 2.779149 for original and 0.16771 for generated and KPSS P-value is 0.10000. Proving that generated series are stationary like the original.

We have used an Auto-Regressive model that simply predicts future performance of the series based on its past performance. Figure 7.9 highlights forecasts made for the next 7 days for each of the series. As we can see forecasts for both the original and CTABGAN-plus generated data are accurate on days 1, 2,3 and 4.

Localization analysis RSSI-based localization analysis is an explored field with several analysis techniques for such information like floor localization [], room wise Wi-Fi strength analysis [], distance estimation from a point of reference []. More often than not RSSI signal strength is combined with other information like constant values of device transmission power, attenuation frequency, sensor attenuation gain; or information from

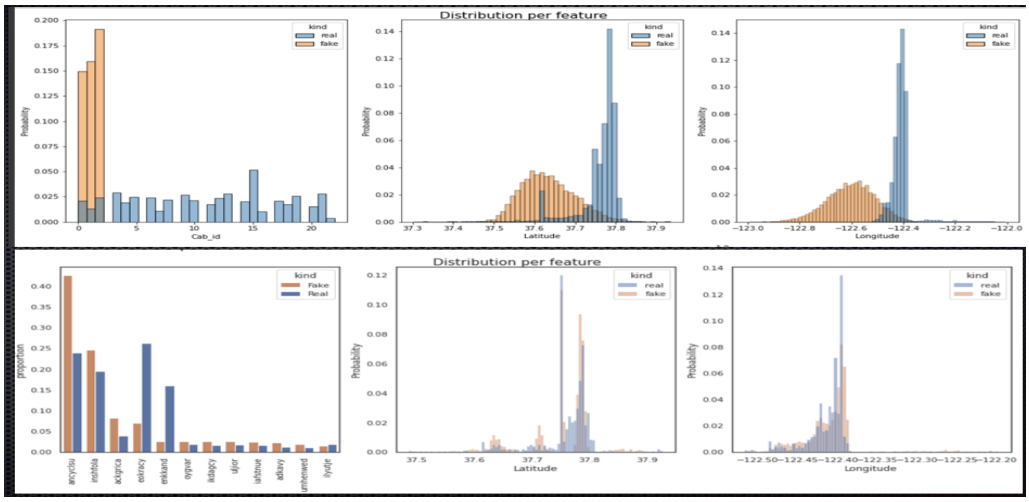

Figure 7.11: Comparison of label distribution in real versus generated data for both cramergan and ctab-gan plus. Chart on the top is for cramergan and bottom is for ctab-ganplus

bluetooth beacons. In our dataset, we do not have any of the other information needed to model a localization system.

So we use one of the quickest ways to compare real versus fake samples, that is visualizing RSSI signatures on a $4 - D$ plane with respect to different target features. Figure 7.10 shows the $4 - D$ representation of RSSI signatures, consisting of 5 sensor endpoints, from both the original and the fake dataset, where we see that the signature patterns are very similar in both datasets. As we see from the figure, the datapoints corresponding to the appropriate labels for both real and generated traces are very similar to each other.

7.6 Discussion

Our results are divided into two main sections: Hyperparameter tuning of the existing CRAMERGAN module to improve quality of generated traces and comparing CRAMERGAN to CTAB-GAN plus results to eventually prove that CTAB-GAN plus outperforms all the previous models for generation across the four uMA categories location, connectivity, health and lifestyle. In terms of resource utilization CTAB-GAN plus has an overall larger trained model size and requires more number of epochs for the GAN to converge. The size is a direct result of the modal collapse handling by the conditional models. Figure 7.11 clearly shows that with a slightly larger resource utilization, we are able to get a more accurate label distribution in the generated data. Cramergan fails to generate data for all

the labels and focuses only on the first 3 labels which is a result of the model focusing more on datapoints with a higher number of a specific label. The numbers throughout the paper are for 2000 randomly chosen data points in each trace. We chose a smaller number to have a cleaner visual representation of the results. For larger traces, the increase in training time is linear, which, when compared to other GAN models like TimeGAN (that have an exponential growth in training time with dataset size), is not as computation heavy. The actual training of the model with the large sizes of the raw uMA traces requires commercial grade servers. The trained model size of CRAMERGAN is around 8 MegaBytes and CTAB-GAN plus is around 12 MB, which can be easily stored in the framework and run by researchers on their local machines.

This pipeline has been wrapped into a framework will be made available after the thesis is published on GitHub. The different parts of the framework can be used by running simple python scripts with different command line arguments like: functionality, which would allow users to select if they want to analyze an existing trace, generate an existing trace or generate a new trace; model, where the user can use one of the four pre-trained models to generate either lifestyle, location, network or health traces, size of new dataset, format of generated traces (csv/json) etc. An output folder is created for every run of the python script, that will contains all the generated datasets and model/ data analysis reports.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

Motivated by a wide range of applications from urban planning, efficient communication, transit and transportation, public health and healthcare, commerce, critical network infrastructure planning and provisioning, to name a few, it has become increasingly important to better understand human mobility and activity. As a result, uMA characterization and modeling has been attracting considerable attention from researchers and practitioners. uMA records and traces have played a crucial role in enabling the exploration of how humans move in a variety of environments.

The main contributions of Chapter 4 include: (1) Proposing a novel taxonomy that classifies these traces based on a number of factors including their mobility mode, data source, data collection technology, information type and their current and potential future applications; (2) Categorizing several well-known public uMA datasets using the proposed taxonomy, along with providing their published source and data sharing policies; and (3) Using three uMA traces, each uniquely categorized using our taxonomy, to show real application of our taxonomy. Our study also discusses significant challenges associated with the publication and availability of real uMA traces, which goes on to motivate our ongoing work on developing realistic uMA trace generators.

Developing an all-in-one solution for collection, analysis and generation of open source uMA traces, while preserving the spatio-temporal diversity and versatility provided by

different trace categories is a non-trivial task. Chapter 5 discusses uMAD, a Machine learning based end to end analysis and generation framework, that achieves both trace generality and fidelity with reasonable computation resource consumption. Chapters 6 and 7 go on to show how CTABGAN Plus can be used as a generalizable GAN architecture to successfully create realistic synthetic uMA datasets across all uMA categories.

This project can be used by uMA researchers who deal with data privacy issues with respect to sharing results of algorithm developements on their private datasets. They can use their data to train our generator and consequently use the trained model to generate privacy preserved uMA data. These datasets are expected to produce similar results with their algorithms, which can then be shared freely.

8.2 Potential Extensions

The existing pipeline can be extended in the following major directions:

- Collection and Storage module: Addition of a repository where we can store the real and generated traces under the different categories of our taxonomy. As of now we only store trained models that can be used to generate a dataset at the users convenience. But in the future we can also have a repository or an online database like Crawdad, or Kaggle, where we can store the generated datasets, once we have verified their accuracy. Using this module the user will also be able to add newer traces into one of the four application categories of our taxonomy. Ingestion of these uMA traces will include analysis to classify them in the appropriate uMA category

- More types of generative models: Plans to maintain the framework by updating to newer ML generative models with greater fidelity as better models are developed. One of the current shortcomings of this tool is the lack of provision to provide prompts to the model before generation of traces. As described in [172] By Ryan O' Connor, diffusion models work by adding Gaussian noise successively to destroy training data, and then learning to recover the data by reversing this noising process. After training, diffusion models can be used to generate data by simply passing randomly sampled noise through the learned de-noising process. Along with the added benefits

of scalability and parallelizability, these models also replace the need for adversarial training. With respect to uMA data and our tool, replacing the GAN with a diffusion model can help by allowing us to leverage the Reinforcement Learning from Human Feedback (RLHF) property of these models.

- Generate innovative types of traces: Create new traces that haven't existed before by training the models on two existing traces.

Bibliography

[1] *Chapter 5. Training and common challenges: GANing for success.*

[2] *GANs: Common Problems.*

[3] *The power of data to enable the future of mobility*. https://nobrainerdata.com/2021/09/20/the-power-of-data-to-enable-the-future-of-mobility/, year = "2021".

[4] *Sklearns Mobility metrics.*

[5] *Urban Mobility data Platform.*

[6] *T-Drive*, 2008. https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/.

[7] *Brightkite*, 2009. https://snap.stanford.edu/data/loc-brightkite.html.

[8] *Cabspotting*, 2009. http://crawdad.org/ crawdad/epfl/mobility/20090224/cab/.

[9] *MobiClique application at SIGCOMM 2009.*, 2009. https://crawdad.org/thlab/sigcomm2009/20120715/.

[10] *Gowalla*, 2010. https://snap.stanford.edu/data/loc-gowalla.html.

[11] *LifeMap monitoring system*, 2012. https://crawdad.org/yonsei/lifemap/20120103/.

[12] *upb/hyccups*, 2012. https://crawdad.org/upb/hyccups/20161017/.

[13] *Geolife*, 2013. https://www.microsoft.com/en-us/download/details.aspx?id=52367.

[14] *Taxi cabs in Rome*, 2014. https://crawdad.org/roma/taxi/20140717/.

[15] *KTH data*, 2015. https://crawdad.org/kth/campus/20190701/.

[16] *Social Blue Conn Application*, 2015. https://crawdad.org/unical/socialblueconn/20150208/.

[17] *Fire Department of Asturias, Spain*, 2016. https://crawdad.org/oviedo/asturies-er/20160808/.

[18] *Global Transnational Mobility dataset.*, 2016. https://ec.europa.eu/knowledge4policy/dataset/ds00162$_en$.

[19] *uoi/haggle*, 2016. https://crawdad.org/uoi/haggle/20160828/one/.

[20] *2017 Downtown Pedestrian Counts*, 2017. https://data.world/louisville/6596a49b-77bc-47fe-a2b5-6aa7270a59ca.

[21] *copelabs/usense/NSense dataset*, 2017. https://crawdad.org/copelabs/usense/20170127/.

[22] *Mobility Wikipedia page*, 2017. https://www.kaggle.com/chicago/chicago-taxi-trips-bq/tasks?taskId=261.

[23] *The rise of mobility as a service*, 2017. https://www2.deloitte.com/content/dam/Deloitte/nl/Documents/consumer-business/deloitte-nl-cb-ths-rise-of-mobility-as-a-service.pdf.

[24] *NYC Citywide Mobility Survey*, 2018. https://data.cityofnewyork.us/Transportation/Citywide-Mobility-Survey-Main-Survey-2017/dd6w-hnq9.

[25] *US Internal Lifetime Mobility*, 2018. https://data.world/garyhoov/us-internal-lifetime-mobility.

[26] *Austin B-cycle Trips*, 2019. https://data.austintexas.gov/d/tyfh-5r8s.

[27] *BLEBeacon dataset*, 2019. https://crawdad.org/unm/blebeacon/20190312/.

[28] *Community RF Sensing via iPhones for Source Localization and Coverage Maps*, 2019. https://crawdad.org/tuc/mysignals/20191030/.

[29] *Covid-19 mobility trends in Apple Maps*, 2019. https://data.world/kgarrett/covid-19-mobility-trends.

[30] *FLEXRAN data*, 2019. https://crawdad.org/eurecom/elasticmon5G2019/20190828/.

[31] *COVID Mobility Trends*, 2020. https://www.apple.com/covid19/mobility.

[32] *Flow Monitoring in the Mediterranean and Western Balkans*, 2020. https://migration.iom.int/europe?type=arrivals.

[33] *GRID Bikeshare Data*, 2020. https://data.world/city-of-tempe/40b49ebc-ff11-43ed-b814-b5137fc53b4c.

[34] *Microsoft visualization of the COVID pandemic trends by Apple and Google*, 2020. https://community.powerbi.com/t5/COVID-19-Data-Stories-Gallery/COVID-19-Pandemic-Mobility-Trends/td-p/1163961.

[35] *Mobility changes in response to COVID-19*, 2020. https://github.com/descarteslabs/DL-COVID-19.

[36] *Synthesising Tabular Data using Wasserstein Conditional GANs with Gradient Penalty (WCGAN-GP)*, 2020. https://ceur-ws.org/Vol-2771/AICS2020$_paper_57.pdf$.

[37] *Travel Sensors*, 2020. https://data.world/cityofaustin/6yd9-yz29.

[38] *About MDS*, 2021. https://www.openmobilityfoundation.org/about-mds/.

[39] *COVID Mobility Trends by Google*, 2021. https://www.google.com/covid19/mobility/.

[40] *Crawdad Data License*, 2021. https://crawdad.org/data-license-agreement.html.

[41] *MDS and Data Privacy at the Open mobility foundation*, 2021. https://www.openmobilityfoundation.org/mds-data-privacy-at-the-omf/.

[42] *RF Data Factory*, 2021. https://www.rfdatafactory.com.

[43] *Synthea: Synthetic Patient Generation*, 2021. https://synthetichealth.github.io/synthea/.

[44] *Understanding the data in MDS*, 2021. https://github.com/openmobilityfoundation/mobility-data-specification/wiki/Understanding-the-Data-in-MDS.

[45] *Move: a blog by arity*, 2022. https://www.arity.com/move/.

[46] *Wikipedia: Jensen–Shannon divergence*, Multiple. https://en.wikipedia.org/wiki/JensenShannon_divergence.

[47] *Wikipedia: Kolmogorov–Smirnov test*, Multiple. https://en.wikipedia.org/wiki/KolmogorovSmirnov_test.

[48] *Wikipedia: Kullback–Leibler divergence*, Multiple. https://en.wikipedia.org/wiki/KullbackLeibler_divergence.

[49] *Wikipedia: Vector space model*, Multiple. https://en.wikipedia.org/wiki/Vector_space_model.

[50] Ouassim Adnane. *Common license types for datasets*, 2019. https://www.kaggle.com/general/116302.

[51] Daniel Adu-Gyamfi and Fengli Zhang. Mobility and trajectory-based technique for monitoring asymptomatic patients. *Journal of Information Technology Research (JITR)*, 15(1):1–18, 2022.

[52] A Ajinu and CP Maheswaran. A novel prediction model for mobility tracing of users with hybrid metaheuristic concept. *Wireless Networks*, 28(1):107–123, 2022.

[53] Nabeel Akhtar, Sinem Coleri Ergen, and Oznur Ozkasap. Vehicle mobility and communication channel models for realistic and efficient highway vanet simulation. *IEEE Transactions on Vehicular Technology*, 64(1):248–262, 2014.

[54] Adil Al Wahaibi, Amal Al Maani, Fatma Alyaquobi, Abdullah Al Manji, Khalid Al Harthy, Bader Al Rawahi, Abdullah Alqayoudhi, Sulien Al Khalili, Amina Al-Jardani, and Seif

Al-Abri. The impact of mobility restriction strategies in the control of the covid-19 pandemic: modelling the relation between covid-19 health and community mobility data. *International journal of environmental research and public health*, 18(19):10560, 2021.

[55] Basma Alharbi, Abdulhakim Ali Qahtan, and Xiangliang Zhang. Minimizing user involvement for learning human mobility patterns from location traces. In *AAAI*, pages 865–871, 2016.

[56] Alejandro Lara Allende and André Stephan. Life cycle embodied, operational and mobility-related energy and greenhouse gas emissions analysis of a green development in melbourne, australia. *Applied Energy*, 305:117886, 2022.

[57] Hamed Alqahtani, Manolya Kavakli-Thorne, and Gulshan Kumar. Applications of generative adversarial networks (gans): An updated review.

[58] Mohamed Ali Alzain, Collins Otieno Asweto, Suleman Atique, Najm Eldinn Elsser Elhassan, Ahmed Kassar, Sehar-un-Nisa Hassan, Mohammed Ismail Humaida, Rafeek Adeyemi Yusuf, and Adeniyi Abolaji Adeboye. Effectiveness of human mobility change in reducing the spread of covid-19: Ecological study of kingdom of saudi arabia. *Sustainability*, 14(6):3368, 2022.

[59] Roberto M Amadio. On modelling mobility. *Theoretical Computer Science*, 240(1):147–176, 2000.

[60] Roberto M Amadio and Sanjiva Prasad. Modelling ip mobility. *Formal Methods in System Design*, 17(1):61–99, 2000.

[61] Davide La Rosa Antonino Crivello, Filippo Palumbo and Paolo Barsocchi. A multisource and multivariate dataset for indoor localization methods based on wlan and geo-magnetic field fingerprinting. International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2016.

[62] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[63] Fereshteh Asgari, Vincent Gauthier, and Monique Becker. A survey on human mobility and its applications. *arXiv preprint arXiv:1307.0814*, 2013.

[64] Mudabber Ashfaq, Ali Tahir, Faisal Moeen Orakzai, Gavin McArdle, and Michela Bertolotto. Using t-drive and berlinmod in parallel secondo for performance evaluation of geospatial big data processing. In *Spatial data handling in big data era*, pages 3–19. Springer, 2017.

[65] Insaf Ashrapov. Tabular gans for uneven distribution. *arXiv preprint arXiv:2010.00638*, 2020.

[66] Insaf Ashrapov. Tabular GANs for uneven distribution. `https://pypi.org/project/tabgan/`, multiple.

[67] Davide Bacciu, Paolo Barsocchi, Stefano Chessa, Claudio Gallicchio, and Alessio Micheli. An experimental characterization of reservoir computing in ambient assisted living applications. *Neural Computing and Applications*, 24(6):1451–1464, 2014.

[68] Xuegang Ban and Marco Gruteser. Towards fine-grained urban traffic knowledge extraction using mobile sensing. In *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*, pages 111–117, 2012.

[69] Oresti Banos, Rafael Garcia, Juan A Holgado-Terriza, Miguel Damas, Hector Pomares, Ignacio Rojas, Alejandro Saez, and Claudia Villalonga. mhealthdroid: a novel framework for agile development of mobile health applications. In *International workshop on ambient assisted living*, pages 91–98. Springer, 2014.

[70] Hugo Barbosa, Marc Barthelemy, Gourab Ghoshal, Charlotte R James, Maxime Lenormand, Thomas Louail, Ronaldo Menezes, José J Ramasco, Filippo Simini, and Marcello Tomasini. Human mobility: Models and applications. *Physics Reports*, 734:1–74, 2018.

[71] Paul Barrett, John Hunter, J Todd Miller, J-C Hsu, and Perry Greenfield. matplotlib–a portable python plotting package.

[72] Paolo Barsocchi, Monica Bianchini, Antonino Crivello, Davide La Rosa, Filippo Palumbo, and Franco Scarselli. An unobtrusive sleep monitoring system for the human sleep behaviour

understanding. In *2016 7th IEEE international conference on cognitive infocommunications (CogInfoCom)*, pages 000091–000096. IEEE, 2016.

[73] Richard Becker, Ramón Cáceres, Karrie Hanson, Sibren Isaacman, Ji Meng Loh, Margaret Martonosi, James Rowland, Simon Urbanek, Alexander Varshavsky, and Chris Volinsky. Human mobility characterization from cellular network data. *Communications of the ACM*, 56(1):74–82, 2013.

[74] Rachna Behl and Indu Kashyap. A unified probabilistic factor model with social regularization for point of interest recommendation. *Materials Today: Proceedings*, 51:42–47, 2022.

[75] Loris Belcastro, Riccardo Cantini, and Fabrizio Marozzo. Knowledge discovery from large amounts of social media data. *Applied Sciences*, 12(3):1209, 2022.

[76] Marc G Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743*, 2017.

[77] Bauke Brenninkmeijer. table-evaluator. `https://github.com/Baukebrenninkmeijer/table-evaluator`, 2021.

[78] Complete Dissertation by Statistics Solutions. Correlation (Pearson, Kendall, Spearman). `https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/correlation-pearson-kendall-spearman/`, 2020.

[79] Francesco Calabrese, Giusy Di Lorenzo, and Carlo Ratti. Human mobility prediction based on individual and collective geographical preferences. In *13th international IEEE conference on intelligent transportation systems*, pages 312–317. IEEE, 2010.

[80] Yu Cao, Ang Li, Jinglei Lou, Mingkai Chen, Xuguang Zhang, and Bin Kang. An attention-based bidirectional gated recurrent unit network for location prediction. In *2021 13th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–5. IEEE, 2021.

[81] Márcio M.Lopes Leonardo L.Martins Edmundo Madeira Enzo Mingozzi Omer Rana Carlo Puliafito, Diogo M.Gonçalves and Luiz F.Bittencourt. Mobfogsim: Simulation of mobility and migration for fog computing. *Simulation Modelling Practice and Theory*, 101, May 2020.

[82] Meghna Chakraborty, Md Shakir Mahmud, Timothy J Gates, and Subhrajit Sinha. Analysis and prediction of human mobility in the united states during the early stages of the covid-19 pandemic using regularized linear models. *Transportation Research Record*, page 03611981211067794, 2022.

[83] Haiyang Chen. *Challenges and Corresponding Solutions of Generative Adversarial Networks (GANs): A Survey Study*.

[84] Kaiqi Chen, Min Deng, and Yan Shi. A temporal directed graph convolution network for traffic forecasting using taxi trajectory data. *ISPRS International Journal of Geo-Information*, 10(9):624, 2021.

[85] Sijia Chen, Jian Zhang, Fanwei Meng, and Dini Wang. A markov chain position prediction model based on multidimensional correction. *Complexity*, 2021, 2021.

[86] Andrew Chio, Daokun Jiang, Peeyush Gupta, Georgios Bouloukakis, Roberto Yus, Sharad Mehrotra, and Nalini Venkatasubramanian. Smartspec: Customizable smart space datasets via event-driven simulations. In *2022 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 152–162. IEEE, 2022.

[87] Yohan Chon, Hyojeong Shin, Elmurod Talipov, and Hojung Cha. Evaluating mobility models for temporal prediction with high-granularity mobility data. In *2012 IEEE International Conference on Pervasive Computing and Communications*, pages 206–212. IEEE, 2012.

[88] Carmela Comito, Deborah Falcone, and Domenico Talia. Mining human mobility patterns from social geo-tagged data. *Pervasive and Mobile Computing*, 33:91–107, 2016.

[89] Matteo Corain, Paolo Garza, and Abolfazl Asudeh. Dbscout: A density-based method for scalable outlier detection in very large datasets. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 37–48. IEEE, 2021.

[90] Hekmat Dabbas and Bernhard Friedrich. Benchmarking machine learning algorithms by inferring transportation modes from unlabeled gps data. *Transportation Research Procedia*, 62:383–392, 2022.

[91] Carlos Alberto V. Campos Katia Obraczka Danielle L. Ferreira, Bruno A.A Nunes. A deep learning approach for identifying user communities based on geographical preferences and its applications to urban and environmental planning. *ACM Transactions on Spatial Algorithms and Systems*, (17), April 2020.

[92] Tinh Cong Dao and Hai Thanh Nguyen. Human mobility prediction using k-latest check-ins. In *International Conference on Future Data and Security Engineering*, pages 36–49. Springer, 2021.

[93] Helen Cristina de Mattos Senefonte, Myriam Regattieri Delgado, Ricardo Lüders, and Thiago H Silva. Predictour: Predicting mobility patterns of tourists based on social media user's profiles. *IEEE Access*, 2022.

[94] Anne Marie Delaney, Eoin Brophy, and Tomas E Ward. Synthesis of realistic ecg using generative adversarial networks. *ArXiv*, abs/1909.09150, Sept. 2019.

[95] Tao Deng, Ghafour Ahani, Pingzhi Fan, and Di Yuan. Cost-optimal caching for d2d networks with user mobility: Modeling, analysis, and computational approaches. *IEEE Transactions on Wireless Communications*, 17(5):3082–3094, 2018.

[96] Nicholas N DePhillipo, Jorge Chahla, Michael Busler, and Robert F LaPrade. Mobile phone gps data and prevalence of covid-19 infections: Quantifying parameters of social distancing in the us. *Archives of Bone and Joint Surgery*, 9(2):217, 2021.

[97] Pandas development team. pandas-dev/pandas: Pandas. https://doi.org/10.5281/zenodo.3509134, February 2020.

[98] Trinh Minh Tri Do and Daniel Gatica-Perez. Contextual conditional models for smartphone-based human mobility prediction. In *Proceedings of the 2012 ACM conference on ubiquitous computing*, pages 163–172, 2012.

[99] Fredérić Docquier, Nicolas Golenvaux, Siegfried Nijssen, Pierre Schaus, and Felix Stips. Cross-border mobility responses to covid-19 in europe: new evidence from facebook data. *Globalization and Health*, 18(1):1–17, 2022.

[100] Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. *arXiv*, 1802.04208, Feb. 2018.

[101] Sijing Duan, Feng Lyu, Ju Ren, Yifeng Wang, Peng Yang, Desheng Zhang, and Yaoxue Zhang. Multitype highway mobility analytics for efficient learning model design: A case of station traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[102] Eman O Eldawy, Abdeltawab Hendawi, Mohammed Abdalla, and Hoda MO Mokhtar. Fraudmove: Fraud drivers discovery using real-time trajectory outlier detection. *ISPRS International Journal of Geo-Information*, 10(11):767, 2021.

[103] Sara Elhishi, Mervat Abu-Elkheir, and Ahmed Abou Elfetouh. Solomo cities: socio-spatial city formation detection and evolution tracking approach. *International Journal of Business Intelligence and Data Mining*, 18(1):109–126, 2021.

[104] Laetitia Gauvin Filippo Privitera Brennan Lake Ciro Cattuto Michele Tizzoni Emanuele Pepe, Paolo Bajardi. Covid-19 outbreak response, a dataset to assess mobility changes in italy following national lockdown. *Scientific Data*, 7(230), 2020.

[105] Yuki Endo, Hiroyuki Toda, Kyosuke Nishida, and Akihisa Kawanobe. Deep feature extraction from trajectories for transportation mode estimation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 54–66. Springer, 2016.

[106] Ericsson.

[107] Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional gans. *ArXiv*, abs/1706.02633, June 2018.

[108] Efthimios Bothos Babis Magoutas Luka Bradesko Johann Schrammel Gregoris Mentzas Evangelia Anagnostopoulou, Jasna Urbančič. From mobility patterns to behavioural change: leveraging travel behaviour and personality profiles to nudge for sustainable transportation. *Journal of Intelligent Information Systems*, pages 157–178, 2020.

[109] Farbod Faghihi and Petteri Nurmi. An empirical study on the regularity of route mobility. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, pages 1418–1425, 2016.

[110] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 world wide web conference*, pages 1459–1468, 2018.

[111] Jie Feng, Can Rong, Funing Sun, Diansheng Guo, and Yong Li. Pmf: A privacy-preserving human mobility prediction framework via federated learning. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1):1–21, 2020.

[112] Vincent Fortuin, Gunnar Rätsch, and Stephan Mandt. Multivariate time series imputation with variational autoencoders. *ArXiv*, abs/1907.04155, July 2019.

[113] Yucheng Fu and Yang Liu. Bubgan: Bubble generative adversarial networks for synthesizing realistic bubbly flow images. *Chemical Engineering Science*, 204:35–47, Aug. 2019.

[114] Huiji Gao and Huan Liu. Mining human mobility in location-based social networks. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 7(2):1–115, 2015.

[115] Qiang Gao, Fan Zhou, Goce Trajcevski, Kunpeng Zhang, Ting Zhong, and Fengli Zhang. Predicting human mobility via variational attention. In *The World Wide Web Conference*, pages 2750–2756, 2019.

[116] Salvador García-Ayllón and Phaedon Kyriakidis. Spatial analysis of environmental impacts linked to changes in urban mobility patterns during covid-19: Lessons learned from the cartagena case study. *Land*, 11(1):81, 2022.

[117] Xiaohu Ge, Junliang Ye, Yang Yang, and Qiang Li. User mobility evaluation for 5g small cell networks based on individual mobility model. *IEEE Journal on Selected Areas in Communications*, 34(3):528–541, 2016.

[118] Tuan Nguyen Gia, Mingzhe Jiang, Amir-Mohammad Rahmani, Tomi Westerlund, Pasi Liljeberg, and Hannu Tenhunen. Fog computing in healthcare internet of things: A case study on ecg feature extraction. In *2015 IEEE international conference on computer and*

information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive intelligence and computing, pages 356–363. IEEE, 2015.

[119] David Goetze. Evolution, mobility, and ethnic group formation. *Politics and the Life Sciences*, pages 59–71, 1998.

[120] Maria Gorodetski. Hyperparameter Tuning Methods - Grid, Random or Bayesian Search? https://towardsdatascience.com/bayesian-optimization-for-hyperparameter-tuning-how-and-why-655b0ee0b399, 2021.

[121] Günay Gültekin and Oğuz Bayat. A naïve bayes prediction model on location-based recommendation by integrating multi-dimensional contextual information. *Multimedia Tools and Applications*, pages 1–22, 2022.

[122] Andrea Hess, Karin Anna Hummel, Wilfried N Gansterer, and Günter Haring. Data-driven human mobility modeling: a survey and engineering guidance for mobile networking. *ACM Computing Surveys (CSUR)*, 48(3):1–39, 2015.

[123] Mohammad Asadul Hoque, Xiaoyan Hong, and Brandon Dixon. Analysis of mobility patterns for urban taxi cabs. In *2012 international conference on computing, networking and communications (ICNC)*, pages 756–760. IEEE, 2012.

[124] Sameh Hosny, Atilla Eryilmaz, and Hesham El Gamal. Impact of user mobility on d2d caching networks. In *2016 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2016.

[125] Tao Hu, Siqin Wang, Bing She, Mengxi Zhang, Xiao Huang, Yunhe Cui, Jacob Khuri, Yaxin Hu, Xiaokang Fu, Xiaoyue Wang, et al. Human mobility data in the covid-19 pandemic: Characteristics, applications, and challenges. *International Journal of Digital Earth*, 14(9):1126–1147, 2021.

[126] Mehdi Mirza Ian J. Goodfellow, Jean Pouget-Abadie and Yoshua Bengio. Generative adversarial networks.

[127] Takumi Ichimura and Shin Kamada. Adaptive learning method of recurrent temporal deep belief network to analyze time series data. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2346–2353. IEEE, May 2017.

[128] Sana Imtiaz, Muhammad Arsalan, Vladimir Vlassov, and Ramin Sadre. Synthetic and private smart health care data generation using gans. In *2021 International Conference on Computer Communications and Networks (ICCCN)*, pages 1–7. IEEE, 2021.

[129] Birgit Jaekel and Deepti Muley. Transport impacts in germany and state of qatar: An assessment during the first wave of covid-19. *Transportation research interdisciplinary perspectives*, 13:100540, 2022.

[130] Sapan H Mankad Sanjay Garg Jai Prakash V Verma. Geohash tag based mobility detection and prediction for traffic management. *SN Applied Sciences*, 2(1385), July 2020.

[131] Yunseok Jang, Gunhee Kim, and Yale Song. Video prediction with appearance and motion conditions. *arXiv*, abs/1807.02635, July 2018.

[132] Ali Imran Jehangiri, Tahir Maqsood, Arif Iqbal Umar, Junaid Shuja, Zulfiqar Ahmad, Imed Ben Dhaou, Mohammed F Alsharekh, et al. Limpo: lightweight mobility prediction and offloading framework using machine learning for mobile edge computing. *Cluster Computing*, pages 1–19, 2022.

[133] Peng Jiang, Cheng Chen, and Xiao Liu. Time series prediction for evolutions of complex systems: A deep learning approach. In *2016 IEEE International Conference on Control and Robotics Engineering (ICCRE)*, pages 1–6. IEEE, Apr. 2016.

[134] Renhe Jiang, Xuan Song, Zipei Fan, Tianqi Xia, Quanjun Chen, Satoshi Miyazawa, and Ryosuke Shibasaki. Deepurbanmomentum: An online deep-learning system for short-term urban mobility prediction. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[135] Shan Jiang, Joseph Ferreira, and Marta C Gonzalez. Activity-based human mobility patterns inferred from mobile phone data: A case study of singapore. *IEEE Transactions on Big Data*, 3(2):208–219, 2017.

[136] Yuqin Jiang, Xiao Huang, and Zhenlong Li. Spatiotemporal patterns of human mobility and its association with land use types during covid-19 in new york city. *ISPRS International Journal of Geo-Information*, 10(5):344, 2021.

[137] Deepak Pathak Trevor Darrell Alexei A Efros Oliver Wang Jun-Yan Zhu, Richard Zhang and Eli Shechtman. Toward multimodal image-to-image translation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 465–476. Curran Associates Inc., 2017.

[138] Raja Jurdak, Kun Zhao, Jiajun Liu, Maurice AbouJaoude, Mark Cameron, and David Newth. Understanding human mobility from twitter. *PloS one*, 10(7):e0131469, 2015.

[139] Dmytro Karamshuk, Chiara Boldrini, Marco Conti, and Andrea Passarella. Human mobility models for opportunistic networks. *IEEE Communications Magazine*, 49(12):157–165, 2011.

[140] Risto Katila, Tuan Nguyen Gia, and Tomi Westerlund. Analysis of mobility support approaches for edge-based iot systems using high data rate bluetooth low energy 5. *Computer Networks*, 209:108925, 2022.

[141] Shengren Ke, Meiyi Xie, Hong Zhu, and Zhongsheng Cao. Group-based recurrent neural network for human mobility prediction. *Neural Computing and Applications*, pages 1–21, 2022.

[142] Nauman Ali Khan, Wuyang Zhou, Mudassar Ali Khan, Ahmad Almogren, and Ikram Ud Din. Correlation between triadic closure and homophily formed over location-based social networks. *Scientific Programming*, 2021, 2021.

[143] Sinjoni Mukhopadhyay King, Faisal Nawab, and Katia Obraczka. A survey of open source user activity traces with applications to user mobility characterization and modeling.

[144] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*, 2017.

[145] Zann Koh, Yuren Zhou, Billy Pik Lik Lau, Ran Liu, Chau Yuen, and Keng Hua Chong. Clustering and analysis of gps trajectory data using distance-based features. *Available at SSRN 4012594*.

[146] Adriano Koshiyama, Nick Firoozye, and Philip Treleaven. Generative adversarial networks for financial trading strategies fine-tuning and combination. *ArXiv*, abs/1901.01751, Jan. 2019.

[147] Nikolay Laptev. Anogen: Deep anomaly generator. Technical report, Technical Report. Facebook. https://research.fb.com/wp-content/uploads, 2018.

[148] Minhyeok Lee and Junhee Seok. Controllable generative adversarial network. *IEEE Access*, 7:28158–28169, Feb. 2017.

[149] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

[150] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks.

[151] Hongtao Li, Yue Wang, Feng Guo, Jie Wang, Bo Wang, and Chuankun Wu. Differential privacy location protection method based on the markov model. *Wireless Communications and Mobile Computing*, 2021, 2021.

[152] Ji Li, Xin Pei, Xuejiao Wang, Danya Yao, Yi Zhang, and Yun Yue. Transportation mode identification with gps trajectory data and gis information. *Tsinghua Science and Technology*, 26(4):403–416, 2021.

[153] Xiaolong Li, Gang Pan, Zhaohui Wu, Guande Qi, Shijian Li, Daqing Zhang, Wangsheng Zhang, and Zonghui Wang. Prediction of urban human mobility using large-scale taxi traces and its applications. *Frontiers of Computer Science*, 6(1):111–121, 2012.

[154] Miao Lin and Wen-Jing Hsu. Mining gps data for mobility patterns: A survey. *Pervasive and mobile computing*, 12:1–16, 2014.

[155] Bin Liu, Qi Zhu, Weiqiang Tan, and Hongbo Zhu. Congestion-optimal wifi offloading with user mobility management in smart communications. *Wireless Communications and Mobile Computing*, 2018, 2018.

[156] Mingyang Liu, Junhao Han, Yushan Mei, and Yuguang Li. Dynamic balance between demand-and-supply of urban taxis over trajectories. *Mathematical Biosciences and Engineering*, 19(1):1041–1057, 2022.

[157] Xin Liu, Zhuo Li, Wenzhong Li, Sanglu Lu, Xiaoliang Wang, and Daoxu Chen. Exploring social properties in vehicular ad hoc networks. In *Proceedings of the Fourth Asia-Pacific Symposium on Internetware*, pages 1–7, 2012.

[158] Massimiliano Luca, Gianni Barlacchi, Bruno Lepri, and Luca Pappalardo. A survey on deep learning for human mobility. *ACM Computing Surveys (CSUR)*, 55(1):1–44, 2021.

[159] Massimiliano Luca, Luca Pappalardo, Bruno Lepri, and Gianni Barlacchi. Trajectory test-train overlap in next-location prediction datasets. *arXiv preprint arXiv:2203.03208*, 2022.

[160] Huiwen Luo, Haoming Zhang, Shigong Long, and Yi Lin. Enhancing frequent location privacy-preserving strategy based on geo-indistinguishability. *Multimedia Tools and Applications*, 80(14):21823–21841, 2021.

[161] Pablo Martí, Leticia Serrano-Estrada, and Almudena Nolasco-Cirugeda. Social media data: Challenges, opportunities and limitations in urban studies.

[162] Wes McKinney, Josef Perktold, and Skipper Seabold. Time series analysis in python with statsmodels. *Jarrodmillman. Com*.

[163] David H Metz. Mobility of older people and their quality of life. *Transport policy*, 7(2):149–152, 2000.

[164] MIT. Ydata-synthetic. `https://github.com/ydataai/ydata-synthetic#readme`, 2022.

[165] Anugerah Karta Monika. The utility of 'covid-19 mobility report'and 'google trend'for analysing economic activities. *Syntax Idea*, 3(6):1256–1268, 2021.

[166] Hugues Moreau, Andrea Vassilev, and Liming Chen. The devil is in the details: An efficient convolutional neural network for transport mode detection. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[167] Parisa Fard Moshiri, Hojjat Navidan, Reza Shahbazian, Seyed Ali Ghorashi, and David Windridge. Using gan to enhance the accuracy of indoor human activity recognition. *arXiv preprint arXiv:2004.11228*, 2020.

[168] Mamta Nain and Nitin Goyal. Energy efficient localization through node mobility and propagation delay prediction in underwater wireless sensor network. *Wireless Personal Communications*, 122(3):2667–2685, 2022.

[169] Mathias Niemann Tygesen, Francisco C Pereira, and Filipe Rodrigues. Unboxing the graph: Neural relational inference for mobility prediction. *arXiv e-prints*, pages arXiv–2201, 2022.

[170] Anastasios Noulas, Salvatore Scellato, Neal Lathia, and Cecilia Mascolo. Mining user mobility features for next place prediction in location-based services. In *2012 IEEE 12th international conference on data mining*, pages 1038–1043. IEEE, 2012.

[171] Karl Cedric U Obias and Elmer R Magsino. Extracting vehicular mobility dynamics from taxi fleet trajectories. In *2021 IEEE 9th Region 10 Humanitarian Technology Conference (R10-HTC)*, pages 1–6. IEEE.

[172] Ryan O'Connor. *The Math Behind GANs*, 2022. https://www.assemblyai.com/blog/diffusion-models-for-machine-learning-introduction/.

[173] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, pages 2642–2651. JMLR.org, 2017.

[174] Tamoghna Ojha, Theofanis P Raptis, Marco Conti, and Andrea Passarella. Balanced wireless crowd charging with mobility prediction and social awareness. *Computer Networks*, page 108989, 2022.

[175] Thomas Olutoyin Oshin, Stefan Poslad, and Zelun Zhang. Energy-efficient real-time human mobility state classification using smartphones. *IEEE Transactions on computers*, 64(6):1680–1693, 2014.

[176] Timothy Otim, Leandro Dörfer, Dina Bousdar Ahmed, and Estefania Munoz Diaz. Modeling the impact of weather and context data on transport mode choices: A case study of gps trajectories from beijing. *Sustainability*, 14(10):6042, 2022.

[177] Sujan Sarker Tamal Adhikary Md. Abdur Razzaque Palash Roy, Anika Tahsin and Mohammad Mehedi Hassan. User mobility and quality-of-experience aware placement of virtual network functions in 5g. *Elsevier Computer Communications*, 150:367–377, January 2020.

[178] John RB Palmer, Thomas J Espenshade, Frederic Bartumeus, Chang Y Chung, Necati Ercan Ozgencil, and Kathleen Li. New approaches to human mobility: Using mobile phones for demographic research. *Demography*, 50(3):1105–1128, 2013.

[179] Michela Papandrea, Karim Keramat Jahromi, Matteo Zignani, Sabrina Gaito, Silvia Giordano, and Gian Paolo Rossi. On the properties of human mobility. *Computer Communications*, 87:19–36, 2016.

[180] Luca Pappalardo, Filippo Simini, Gianni Barlacchi, and Roberto Pellungrini. scikit-mobility: A python library for the analysis, generation and risk assessment of mobility data. *arXiv preprint arXiv:1907.07062*, 2019.

[181] 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, June 2018.

[182] Vikram Patil, Shivam Parikh, Omkar N Kulkarni, Kalika Bhatia, and Pradeep K Atrey. Geosecure-c: A method for secure gps trajectory compression over cloud. In *2021 IEEE Conference on Communications and Network Security (CNS)*, pages 1–2. IEEE, 2021.

[183] Philip Pecher, Michael Hunter, and Richard Fujimoto. Data-driven vehicle trajectory prediction. In *Proceedings of the 2016 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, pages 13–22, 2016.

[184] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[185] Poria Pirozmand, Guowei Wu, Behrouz Jedari, and Feng Xia. Human mobility in opportunistic networks: Characteristics, models and prediction methods. *Journal of Network and Computer Applications*, 42:45–58, 2014.

[186] E Pisoni, P Christidis, and E Navajas Cawood. Active mobility versus motorized transport: User choices and benefits for the society. *Science of The Total Environment*, 806:150627, 2022.

[187] Flora Poecze, Claus Ebster, and Christine Strauss. Social media metrics and sentiment analysis to evaluate the effectiveness of social media posts. *Procedia computer science*, 130:660–666, 2018.

[188] Konstantinos Poularakis and Leandros Tassiulas. Exploiting user mobility for wireless content delivery. In *2013 IEEE International Symposium on Information Theory*, pages 1017–1021. IEEE, 2013.

[189] Bhaskar Prabhala, Jingjing Wang, Budhaditya Deb, Thomas La Porta, and Jiawei Han. Leveraging periodicity in human mobility for next place prediction. In *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2665–2670. IEEE, 2014.

[190] Python. Scikit-Optimize. `https://scikit-optimize.github.io/stable/`, multiple.

[191] Bozhao Qi, Lei Kang, and Suman Banerjee. A vehicle-based edge computing platform for transit and human mobility analytics. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, pages 1–14, 2017.

[192] Mengjun Qi, Zhongyuan Wang, Zheng He, and Tao Lu. Identifying users by asynchronous mobility trajectories. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 6811–6814. IEEE, 2019.

[193] Yuanyuan Qiao, Zhongwei Si, Yanting Zhang, Fehmi Ben Abdesslem, Xinyu Zhang, and Jie Yang. A hybrid markov-based model for human mobility prediction. *Neurocomputing*, 278:99–109, 2018.

[194] Rezwana Rafiq, Michael G McNally, Yusuf Sarwar Uddin, and Tanjeeb Ahmed. Impact of working from home on activity-travel behavior during the covid-19 pandemic: An aggregate structural analysis. *Transportation Research Part A: Policy and Practice*, 159:35–54, 2022.

[195] Jun Rao, Hao Feng, and Zhiyong Chen. Exploiting user mobility for d2d assisted wireless caching networks. In *2016 8th International Conference on Wireless Communications & Signal Processing (WCSP)*, pages 1–5. IEEE, 2016.

[196] Alicia Rodriguez-Carrion, Sajal K Das, Celeste Campo, and Carlos Garcia-Rubio. Impact of location history collection schemes on observed human mobility features. In *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*, pages 254–259. IEEE, 2014.

[197] Adam Sadilek and John Krumm. Far out: Predicting long-term human mobility. In *Twenty-sixth AAAI conference on artificial intelligence*, 2012.

[198] Muhammad Safdar, Arshad Jamal, Hassan M Al-Ahmadi, Muhammad Tauhidur Rahman, and Meshal Almoshaogeh. Analysis of the influential factors towards adoption of car-sharing: A case study of a megacity in a developing country. *Sustainability*, 14(5):2778, 2022.

[199] Soumalya Sarkar, Kin G Lore, Soumik Sarkar, Vikram Ramanan, Satyanarayanan R Chakravarthy, Shashi Phoha, and Asok Ray. Early detection of combustion instability from hi-speed flame images via deep learning and symbolic time series analysis. In *Annual Conf. of the Prognostics and Health Management*, volume 6, 2015.

[200] Konstantinos Amplianitis Sebastian Lutz and Aljosa Smolic. AlphaGAN: Generative adversarial networks for natural image matting. *ArXiv*, abs/1807.10088, July 2018.

[201] Wenhua Shao, Haiyong Luo, Fang Zhao, Yan Ma, Zhongliang Zhao, and Antonino Crivello. Indoor positioning based on fingerprint-image and deep learning. *IEEE Access*, 6:74699–74712, 2018.

[202] S Sharmila and BA Sabarish. Analysis of distance measures in spatial trajectory data clustering. In *IOP Conference Series: Materials Science and Engineering*, volume 1085, page 012021. IOP Publishing, 2021.

[203] Taylor Shelton, Ate Poorthuis, and Matthew Zook. Social media and the city: Rethinking urban socio-spatial inequality using user-generated geographic information. *Landscape and urban planning*, 142:198–211, 2015.

[204] Seungjae Shin, Hongseok Jeon, Chunglae Cho, Seunghyun Yoon, and Taeyeon Kim. User mobility synthesis based on generative adversarial networks: A survey.

[205] Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In *International Conference on Artificial Neural Networks*, pages 412–422, 2018.

[206] Gavin Smith, Romain Wieser, James Goulding, and Duncan Barrack. A refined limit on the predictability of human mobility. In *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 88–94. IEEE, 2014.

[207] Gürkan Solmaz and Damla Turgut. A survey of human mobility models. *IEEE Access*, 7:125711–125731, 2019.

[208] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.

[209] Xuan Song, Hiroshi Kanasugi, and Ryosuke Shibasaki. Deeptransport: Prediction and simulation of human mobility and transportation mode at a citywide level. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2618–2624, 2016.

[210] Xuan Song, Quanshi Zhang, Yoshihide Sekimoto, and Ryosuke Shibasaki. Prediction of human emergency behavior and their mobility following large-scale disaster. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 5–14, 2014.

[211] Open Source. CRAWDAD. `http://crawdad.org`, multiple.

[212] Open Source. Data.world website. `https://data.world/datasets/traces`, multiple.

[213] Open Source. GitHub. https://privamov.github.io/accio/docs/datasets.html, multiple.

[214] Open Source. Kaggle Mobility traces. https://www.kaggle.com/search?q=mobility, multiple.

[215] Sajad Sowlati, Rahim Ali Abbaspour, and Alireza Chehreghan. Identifying transportation modes from trajectory dataset using boosting and deep learning methods in intelligent transportation system. *Journal of Transportation Research*, 2022.

[216] Marco Aurélio Spohn and Matheus Henrique Trichez. An analysis of a real mobility trace based on standard mobility metrics. *Revista de Informática Teórica e Aplicada*, 26(1):26–35, 2019.

[217] Jake Tae. *The Math Behind GANs*, Multiple. https://jaketae.github.io/study/gan-math/.

[218] Shuntaro Takahashi, Yu Chen, and Kumiko Tanaka-Ishii. Modeling financial time-series with generative adversarial networks.

[219] Wei Ren Tan, Chee Seng Chan, Hernán E Aguirre, and Kiyoshi Tanaka. Artgan: Artwork synthesis with conditional categorical gans. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3760–3764. IEEE, Sept. 2017.

[220] Yuzuru Tanahashi, James R Rowland, Stephen North, and Kwan-Liu Ma. Inferring human mobility patterns from anonymized mobile communication usage. In *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia*, pages 151–160, 2012.

[221] Jinjun Tang, Fang Liu, Yinhai Wang, and Hua Wang. Uncovering urban human mobility from large scale taxi gps data. *Physica A: Statistical Mechanics and its Applications*, 438:140–153, 2015.

[222] Graham W Taylor, Geoffrey E Hinton, and Sam T Roweis. Two distributed-state models for generating high-dimensional time series.

[223] Fanny Thornton, Karen E McNamara, Carol Farbotko, Olivia Dun, Hedda Ransan-Cooper, Emilie Chevalier, and Purevdulam Lkhagvasuren. Human mobility and environmental change: a survey of perceptions and policy direction. *Population and Environment*, 40(3):239–256, 2019.

[224] Jan Tkačík and Pavel Kordík. Neural turing machine for sequential learning of human mobility patterns. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 2790–2797. IEEE, 2016.

[225] Eran Toch, Boaz Lerner, Eyal Ben-Zion, and Irad Ben-Gal. Analyzing large-scale human mobility data: a survey of machine learning methods and applications. *Knowledge and Information Systems*, 58(3):501–523, 2019.

[226] Duy Hoang Tran, Pieter Leyman, and Patrick De Causmaecker. Adaptive passenger-finding recommendation system for taxi drivers with load balancing problem. *Computers & Industrial Engineering*, 169:108187, 2022.

[227] Javier Turienzo, Pablo Cabanelas, and Jesús F Lampón. The mobility industry trends through the lens of the social analysis: A multi-level perspective approach. *SAGE Open*, 12(1):21582440211069145, 2022.

[228] Gábor Soós ; Dániel Ficzere ; Pál Varga. User group behavioural pattern in a cellular mobile network for 5g use-cases. *IEEE/IFIP Network Operations and Management Symposium*, April 2020.

[229] Andres I Vecino-Ortiz, Juliana Villanueva Congote, Silvana Zapata Bedoya, and Zulma M Cucunuba. Impact of contact tracing on covid-19 mortality: An impact evaluation using surveillance data from colombia. *Plos one*, 16(3):e0246987, 2021.

[230] Alina Velias, Sotiris Georganas, and Sotiris Vandoros. Covid-19: Early evening curfews and mobility. *Social Science & Medicine*, 292:114538, 2022.

[231] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python.

[232] Zia Wadud, Sheikh Mokhlesur Rahman, and Annesha Enam. Face mask mandates and risk compensation: an analysis of mobility data during the covid-19 pandemic in bangladesh. *BMJ global health*, 7(1):e006803, 2022.

[233] Dashun Wang, Dino Pedreschi, Chaoming Song, Fosca Giannotti, and Albert-Laszlo Barabasi. Human mobility, social ties, and link prediction. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1100–1108, 2011.

[234] Di Wang, Tomio Miwa, and Takayuki Morikawa. Comparative analysis of spatial–temporal distribution between traditional taxi service and emerging ride-hailing. *ISPRS International Journal of Geo-Information*, 10(10):690, 2021.

[235] Ge Wang, Fangmin Xu, Hengsheng Zhang, and Chenglin Zhao. Joint resource management for mobility supported federated learning in internet of vehicles. *Future Generation Computer Systems*, 129:199–211, 2022.

[236] Jinzhong Wang, Xiangjie Kong, Feng Xia, and Lijun Sun. Urban human mobility: Data-driven modeling and prediction. *ACM SIGKDD Explorations Newsletter*, 21(1):1–19, 2019.

[237] Rui Wang, Jun Zhang, SH Song, and Khaled B Letaief. Mobility-aware caching in d2d networks. *IEEE Transactions on Wireless Communications*, 16(8):5001–5015, 2017.

[238] Xinyi Wang, Xiang Yan, Xilei Zhao, and Zhuoxuan Cao. Identifying latent shared mobility preference segments in low-income communities: Ride-hailing, fixed-route bus, and mobility-on-demand transit. *Travel Behaviour and Society*, 26:134–142, 2022.

[239] Yan Wang. Government policies, national culture and social distancing during the first wave of the covid-19 pandemic: International evidence. *Safety Science*, 135:105138, 2021.

[240] Yingzi Wang, Nicholas Jing Yuan, Defu Lian, Linli Xu, Xing Xie, Enhong Chen, and Yong Rui. Regularity and conformity: Location prediction using heterogeneous mobility data. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1275–1284, 2015.

[241] Yongjie Wang and Maolin Li. Optimization algorithm design for the taxi-sharing problem and application. *Mathematical Problems in Engineering*, 2021, 2021.

[242] Le Wen, Mingyue Sheng, and Basil Sharp. The impact of covid-19 on changes in community mobility and variation in transport modes. *New Zealand Economic Papers*, 56(1):98–105, 2022.

[243] Wenchao Wu, Yixian Zheng, Nan Cao, Haipeng Zeng, Bing Ni, Huamin Qu, and Lionel M Ni. Mobiseg: Interactive region segmentation using heterogeneous mobility data. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*, pages 91–100. IEEE, 2017.

[244] Xiuchao Wu, Kenneth N Brown, and Cormac J Sreenan. Analysis of smartphone user mobility traces for opportunistic data collection in wireless sensor networks. *Pervasive and Mobile Computing*, 9(6):881–891, 2013.

[245] Xiu-Feng Xia, Miao Jiang, Xiang-Yu Liu, and Chuan-Yu Zong. Location-visiting characteristics based privacy protection of sensitive relationships. *Electronics*, 11(8):1214, 2022.

[246] Chunjing Xiao, Daojun Han, Yongsen Ma, and Zhiguang Qin. Csigan: Robust channel state information-based activity recognition with gans.

[247] Jian Xiong, Hengrui Hu, Peng Cheng, Can Yang, Zhiping Shi, and Lin Gui. Wireless resource scheduling for high mobility scenarios: A combined traffic and channel quality prediction approach. *IEEE Transactions on Broadcasting*, 2022.

[248] Tao Xu, Penghuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1316–1324, 2018.

[249] Xiping Yang, Zhiyuan Zhao, and Shiwei Lu. Exploring spatial-temporal patterns of urban human mobility hotspots. *Sustainability*, 8(7):674, 2016.

[250] Zheng Yang, Chenshu Wu, Zimu Zhou, Xinglin Zhang, Xu Wang, and Yunhao Liu. Mobility increases localizability: A survey on wireless indoor localization using inertial sensors. *ACM Computing Surveys (Csur)*, 47(3):1–34, 2015.

[251] Dezhong Yao, Chen Yu, Hai Jin, and Qiang Ding. Human mobility synthesis using matrix and tensor factorizations. *Information Fusion*, 23:25–32, 2015.

[252] Lin Yao, Xiaoying Xu, Jing Deng, Guowei Wu, and Zhaoyang Li. A cooperative caching scheme for vccn with mobility prediction and consistent hashing. *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[253] Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. Time-series generative adversarial networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[254] Haoxiang Yu. *A Comprehensive System for Dynamic and Distributed Taxi Ride-Sharing via Localized Communication*. PhD thesis, Miami University, 2021.

[255] Qingying Yu, Chuanming Chen, Liping Sun, and Xiaoyao Zheng. Urban hotspot area detection using nearest-neighborhood-related quality clustering on taxi trajectory data. *ISPRS International Journal of Geo-Information*, 10(7):473, 2021.

[256] Zhiwen Yu, Hui Wang, Bin Guo, Tao Gu, and Tao Mei. Supporting serendipitous social interaction using human mobility prediction. *IEEE Transactions on Human-Machine Systems*, 45(6):811–818, 2015.

[257] Jing Yuan, Yu Zheng, and Xing Xie. Discovering regions of different functions in a city using human mobility and pois. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 186–194, 2012.

[258] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. T-drive: driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems*, pages 99–108, 2010.

[259] Xu Yuting, Lim Zhu An, Sherie Loh Wei, and Phang Yong Xin. A geospatial analysis of tweets during post-circuit breaker in singapore. In *Geospatial Data Analytics and Urban Applications*, pages 79–100. Springer, 2022.

[260] Yuting Zhan, Alex Kyllo, Afra Mashhadi, and Hamed Haddadi. Privacy-aware human mobility prediction via adversarial networks. *arXiv preprint arXiv:2201.07519*, 2022.

[261] Chunkai Zhang and Yingyang Chen. Time series anomaly detection with variational autoencoders. *ArXiv*, abs/1907.01702, July 2019.

[262] Daqiang Zhang, Min Chen, Mohsen Guizani, Haoyi Xiong, and Daqing Zhang. Mobility prediction in telecom cloud using mobile calls. *IEEE Wireless Communications*, 21(1):26–32, 2014.

[263] Kai Zhang, Hui Zhang, Xin He, Li Zhou, Xiaohang Zhang, Chenghai He, and Longtao He. Regffm: A new graph model designed with spatial information for potential social connection mining. In *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 722–727. IEEE, 2022.

[264] Zheng ZHANG, Nan JIANG, Yibing CAO, Jiangshui ZHANG, and Zhenkai YANG. A method for friendship judgement based on improved gravity model with check-in data.

[265] Zilong Zhao. CTAB-GAN Plus. `https://github.com/Team-TUD/CTAB-GAN-Plus`, 2022.

[266] Zilong Zhao, Aditya Kunar, Robert Birke, and Lydia Y Chen. Ctab-gan+: Enhancing tabular data synthesis. *arXiv preprint arXiv:2204.00401*, 2022.

[267] Bin Zhou, Shenghua Liu, Bryan Hooi, Xueqi Cheng, and Jing Ye. Beatgan: Anomalous rhythm detection using adversarially generated time series.

[268] Zack Zhu, Ulf Blanke, and Gerhard Tröster. Inferring travel purpose from crowd-augmented human mobility data. In *Proceedings of the first international conference on IoT in urban space*, pages 44–49, 2014.

[269] Matteo Zignani and Sabrina Gaito. Extracting human mobility patterns from gps-based traces. In *2010 IFIP Wireless Days*, pages 1–5. IEEE, 2010.

[270] Matteo Zignani, Sabrina Gaito, and Gianpaolo Rossi. Extracting human mobility and social behavior from location-aware traces. *Wireless Communications and Mobile Computing*, 13(3):313–327, 2013.

[271] Matteo Zignani, Michela Papandrea, Sabrina Gaito, Silvia Giordano, and Gian Paolo Rossi. On the key features in human mobility: relevance, time and distance. In *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*, pages 260–265. IEEE, 2014.

[272] MM Zonoozi, P Dassanayake, and M Faulkner. Mobility modelling and channel holding time distribution in cellular mobile communication systems. In *Proceedings of GLOBECOM'95*, volume 1, pages 12–16. IEEE, 1995.