DICER: Data Intensive Computing Environment and Runtime for Evaluating Unprecedented Scale of Geospatial-Temporal Human Mobility Data

Debraj De†, Gautam Malviya Thakur, Jesse McGaha, Chance Brown, Xiuling Nie, Todd Thomas, James D. Gaboardi, Kevin Sparks, Annetta Burger, Liz McBride, Joon-Seok Kim, Licia Amichi, Chathika Gunaratne, Carter Christopher, Dan Zubko
Oak Ridge National Laboratory (ORNL), Oak Ridge, Tennessee, USA
{† Corresponding author }
Email: {†ded1, thakurg, mcgahajr, browncc, niex, thomastm, gaboardijd, sparkska, burgerag, mcbrideec, kimj1, amichil, gunaratnecs, christophesc, zubkodj}@ornl.gov

***Abstract*—With the significant increase in sources and volume of human mobility data through commercial data vendors as well as microsimulation of cities, the scale of geospatial-temporal data to analyze and assess for mobility characterization has grown to the level of Big Data. There are mobility related commercial organizations deploying scalable computing, but often the system architecture, workflow, and intermediate processing components are not fully disclosed in relevant scope. Current research literature has a notable lack of studies demonstrating architectures and workflows for human mobility analytics that are implemented on a TeraByte scale of geospatial-temporal data. In this context, this paper presents a hyperscale-level system solution named *DICER* (*D*ata *I*ntensive *C*omputing *E*nvironment and *R*untime) for processing and analytics of geospatial-temporal data at big data scale. Although the cluster computing architecture of *DICER* with Apache Spark job running on Kubernetes cluster is not new, there are innovations in the workflow, hierarchical processing logic, and a wide range of intermediate preprocessing and mobility metrics calculation. We have performed case studies to validate the effectiveness of *DICER* system solution by performing detailed analytics and assessment of human mobility microsimulation output at three different scopes and scale, including a usecase with 16.97 TeraByte and 259.2 Billion rows of data. In addition, we have presented another case study of utilizing DICER to perform the same mobility processing and comparative analytics on large-scale commercially available geospatial-temporal data. All these case studies validate the efficiency of *DICER* in computing population mobility characteristics from geospatial-temporal trajectory data at an unprecedented scale (not only just data volume, but also combination of: number of user entities, temporal frequency, spatial resolution, data duration).**

***Index Terms*—Geospatial temporal data, human mobility, mobility metrics, big data, big data analytics, cluster computing, cloud computing, testing and evaluation.**

I. Introduction

There is a long history of research literature analyzing, characterizing, and designing mobility metrics, in the field of human mobility research [6], [17]. But these works, most of the time, have used geospatial-temporal mobility data sizes that are less than the scale of "large-scale" and of "Big Data". On the other hand, there are relatively fewer and only recent works performing human mobility analytics and research at the data scale of what can be called large-scale or big data [9], [10], [13], [14]. These scientific studies predominantly concentrate on uncovering patterns in human mobility. However, it is worth noticing that commercial entities in the mobility sector often refrain from fully disclosing their system architecture, operational processes, and the various intermediate components involved in their workflows. There is a compelling need in the fields of human mobility and human dynamics for a system solution that describes all its parts (system architecture, operational workflow, intermediate processing components), and that shares case studies experience on performing at-scale mobility processing and analytics on geospatial-temporal data at big data scale. In this work we present our experience designing a scalable cloud-native architecture with cluster computing system resources, in order to analyze large-scale human mobility datasets in practically short time window. The system is named *DICER* (***D***ata ***I***ntensive ***C***omputing ***E***nvironment and ***R***untime) and consists of parallel workers in cluster computing environment deployed in commercial cloud. The purpose of *DICER* is to analyze multi-faceted human mobility behavior on very large-scale geospatial-temporal data. The core value is in detecting large range of human mobility metrics patterns on scalable mobility data within considerably short turnaround time.

For validation with relevant usecases, *DICER* is evaluated on microsimulated human mobility datasets generated at unprecedented scale: high-frequency, high-fidelity and scaled population of a city as AoI (area of interest). The dataset encompasses an extensive collection of mobility data, comprising over a billion records. Each record contains four parameters, collectively known as *ULLT* data: Unique identifier for each user entity or anonymous device, Latitude, Longitude, and a Timestamp. Three datasets generated from the microsimulation are evaluated with *DICER*: 10K user agents (simulating 7 days of patterns of life or PoL), 20K user agents (simulating 28 days of PoL), 50K user agents (simulating 60 days of PoL). Case studies are performed on these datasets where four key mobility data preprocessing and nine key mobility metrics are

computed with *DICER* system solution. With only 150-300 parallel workers (number of workers can be increased considerably), *DICER* completed all these computation on mobility datasets of 6.048 Billion, 48.384 Billion, 259.2 Billion rows (geospatial temporal datapoints) in about 4 hours, 11 hours and 23.5 hours respectively. These runtimes can considerably be decreased further with increasing the number of parallel works parameter in *DICER*.

The main contributions of this paper are as follows: (i) Besides presenting the system architecture dedicated for highly scalable computation of human mobility metrics on geospatial-temporal trajectory data, innovation in computation workflow, hierarchical processing logic, wide range of four intermediate preprocessing and nine mobility metrics calculation; (ii) Extensive case study of using the *DICER* system solution to do test and evaluation on microsimulation of human mobility and patterns of life (PoL) at an unprecedented scale and granularity (not only just big data volume); (iii) Another case study of using *DICER* on characterizing stopping/dwelling patterns of human mobility extracted from real-world commercial holdout data (from the same area of interest as the microsimulation).

Finally, this paper is organized as follows. Section II presents the current state-of-the-art in scalable computation of human mobility characteristics. Section III first introduces the unprecedented scale and granularity of the mobility datasets used in this paper, motivating the need of system solution *DICER*. Then Section IV presents our proposed solution *DICER* in details, including system architecture, computational workflow, hierarchical processing logic, wide range of intermediate preprocessing and mobility metrics calculation. Then Section V presents the extensive case studies of applying *DICER* solution. Finally Section VI concludes the paper.

II. Related Work

Computing human mobility characteristics on unprecedented scale and granularity of geospatial-temporal data may be achieved with high performance computing (HPC), distributed computing, or cluster computing frameworks. There are solutions available like GeoMesa [12] and Apache Sedona [20], which support large-scale geospatial querying and analytics on distributed or cluster computing system. These solutions are more broadly focused on any geospatial data analytics needs. Proposed *DICER* solution here is more specifically focused on the niche of geospatial-temporal human mobility patterns and metrics characterization. But to note that overall all these solutions have underlying Apache Spark (or equivalent) based distributed or cluster computing architecture.

The work in [18] includes surveying the literature on analyzing large-scale, high-dimensional human mobility data. Work in [11] analyzed 100,000 anonymized mobile phone traces where user locations are in terms of location of mobile towers, data temporal frequency is based on when phone calls happened, data duration is 6 months. With that dataset they analyzed metrics like time-independent characteristics of travel distance and probability of returning to a few highly frequented locations. Work in [7] analyzed 1,033,095 individual displacements data, and analyzed trajectories of 464,670 dollar bills obtained from the bill-tracking system (http://www.wheresgeorge.com). Due to the data source type, the spatial and temporal granularity is much sparse. The analysis focused on metrics like distribution of travelling distances and probability of remaining in a small, spatially confined region for a time. Another work [16] has analyzed mobile traces of 50,000 individuals, where user locations are in terms of nearby mobile tower locations, data temporal frequency is based on each user data having average call frequency of ≥ 0.5 hour^{-1}, data duration is 3 months. As for metric and patterns analysis, they have studied lack of variability in diverse movement patterns.

There is a pattern in all these works above in literature that the focus is on analyzing specific limited set of metrics or patterns, and very little details have been shared on the system architecture used (if any) on mobility data at-scale. In fact several works have their mobility dataset with large-number of users/devices, but actually sparse in several dimensions like temporal frequency, geospatial granularity, time duration of data. In these cases the mobility dataset actually do not need a big data processing architecture for analysis, and do not necessarily need cluster computing or distributed computing. In contrast in this work we have shared details of our $DICER$ system architecture, end-to-end workflow of computing several popular mobility metrics at-scale, and case studies of processing and analyzing very large-scale human mobility dataset (with high density in temporal frequency and spatial resolution).

In relevant commercial entities in this domain, there are several who perform analysis of large-scale geospatial temporal human (or other entities like vehicles) mobility data. Due to being commercial products and solutions, generally the big data processing and analysis architectures are not shared publicly at great details. Sometimes though, the overall big data architecture of such commercial organizations (e.g., Uber, SafeGraph) are shared publicly [1], [3]. But those generally covers high-level system architecture serving all types of their data and services, and do not share finer level and scoped details on system architecture and workflow of analyzing/assessing geospatial-temporal human mobility data at scale. In contrast, this work has shared details of design and usage of $DICER$ system architecture that is only focused on scalable processing and analysis of geospatial-temporal human mobility data. In the research literature there are some works that can be found sharing scalable system architecture specifically for analyzing geospatial-temporal mobility data at scale. The work in [19] introduces a system architecture (named δ architecture) for big data processing in the domain of mobility analytics. But for experimental evaluation they have only shared per-component and per-pipeline system evaluation with measures like throughput, latency. But there is no details on if any large-scale mobility dataset was used for validation and case study, and also no mobility metrics related analytics results are presented. Another work in [8] presented the system architecture, data collection, transfer, processing on large-

scale mobility data. They designed a platform called "Real Time Rome Platform" and used it to analyze [15] large-scale mobility data of the city of Rome, Italy. Among the data types, mobility data of human users was provided by Telecom Italia (TI) and consisted of aggregate location and trajectory data on callers. In that work [8] the data processing system architecture is described, but they either didn't use or otherwise described much of the parallel or distributed computing architecture for scaling. In contrast, in this work we have shared in details both: case studies of analyzing big data scale of human mobility data, and the parallel processing mechanism in $DICER$ along with system level performance reporting (performance of parallelism in the case studies).

III. Description of Human Mobility Datasets Generated

The four datasets used for case studies and validation are described here (before describing the *DICER* system) in order to highlight the following: (a) unprecedented combination of scale (data size), precision (spatial) and frequency (temporal) of the geospatial-temporal human mobility data present in the datasets; and (b) capability of *DICER* in being able to perform scalable computing on the datasets. Among the four datasets, *Data-1*, *Data-2*, *Data-3* (as they are named here) are output of human mobility and patterns of life (PoL) based microsimulation work from us. But more details of microsimulation modeling and methodologies are beyond the scope of this paper. Another *Data-4* (as it is named here) is the large-scale real-world human mobility dataset from the same AoI (area of interest) as that of the microsimulation. But unlike microsimulation output in *Data-1*, *Data-2* and *Data-3*, this *Data-4* is collected and prepared from commercially available data vendors. All the necessary details of these four datasets are available in Table I.

IV. *DICER* System Architecture and Processing of Geospatial-Temporal Mobility Data

In this section we describe the system architecture and operational workflow of *DICER*, followed by data processing steps in mobility characterization at-scale.

A. DICER System Architecture

The overall architecture for *DICER* system is shown in Figure 1. The step numbers (e.g., "1.1") and step descriptions (e.g., "Data Ingestion") are shown in the figure to illustrate the order of activities when performing the scalable processing and analysis of mobility datasets.

Starting from the top-left in the diagram, the input data to the system is geospatial-temporal mobility data of interest. Through the steps 1.1 and 1.2 of "Data Ingestion", it is ingested first to a storage bucket (currently using Google Cloud Storage Buckets) and then to a distributed database that can support scalable query and I/O (input/output) operations (currently using BigQuery of Google Cloud). Then in step 2 of "Conformance Testing" the data is checked for all the data sanity requirements, through SQL commands that then is run in fast and scalable manner through the distributed database (currently using BigQuery). Next the step 3.1 of "Submit Spark Jobs for computing Preprocerssing and Metrics in TraKit", Apache Spark computation jobs are submitted to the computing cluster which runs Apache Spark on Kubernetes (currently using GKE or Google Kubernetes Engine for Kubernetes service). The cluster consists of pods, single or multiple of which can run on the available computing nodes (currently nodes are the virtual machines in Google Cloud). The Kubernetes is set to auto-scale in between a minimum (say 10 parallel workers) and a maximum (say 300 parallel workers, but can be increased as needed to multiple of thousands parallel workers). To note that for the case studies presented in this paper, the Kubernetes auto-scaling was configured to use in most of the cases maximum 150-200 parallel workers.

As part of the workers in the pods doing the cluster computing job, each worker do the step 3.2 of "Querying Data by Workers in GKE Cluster". Through step 3.2 each worker gets an input data slice, that it's assigned at that time for computing the TraKit software on. TraKit is our software library for computing various human mobility metrics and performing analytics, more details of TraKit is out of scope of this paper. After computing the TraKit software modules, in the next step 4 of "TraKit Result Outputs to BigQuery", the scalable computing results are all written back to the distributed database (currently BigQuery). Followed by that there are step "5.1 Analysis of Preprocessing and Metrics Results" and step 5.2 "Save analytical results to BigQuery", where the statistical analysis and assessment of the computed preprocessing and metrics results in TraKit are done and saved in the distributed database. This analysis step though just needed a virtual machine with high enough memory and CPU capacity, and not the resource of doing cluster computing (unlike computing the preprocessing and metrics in TraKit at-scale in the cluster). After these, there are steps 6.1 for ETL (extract, transform and load) of all the results and step 6.2 of visualization. Also to note that there are other key components and activities involved in the *DICER* architecture: orchestration (currently using Apache Airflow); logging and monitoring of the cluster computing jobs as they run on the worker pods; CI/CD (continuous integration and continuous delivery/continuous deployment) pipeline management.

B. DICER Overall Workflow for Scalable Computing Jobs

As described in previous Section IV-A, *DICER* uses Cluster Computing, and runs Apache Spark application on a Kubernetes cluster. Currently *DICER* is running on commercial Google Cloud computing environment. However our core design of *DICER* do not specifically need any commercial cloud, and can run with on-premises/on-prem computing resources and open-source software. Apache Spark and Kubernetes used in *DICER* are all open-source themselves, this work just used the Kubernetes service of GKE (Google Kubernetes Engine). Now in the Cluster Computing environment, each computing worker (here a worker pod in Spark on Kubernetes cluster)

Dataset description	Number of users	Temporal frequency	Spatial Resolution	Data duration	Dataset size
Data-1: Microsimulation of human mobility and patterns of life (PoL) in AoI (area of interest) of Knoxville, Tennessee, U.S. Overall Knoxville MSA (metropolitan statistical area) is used as the AoI.	10,000 simulated human users as agents in microsimulation.	Every 1 second	At least 6 decimal digits precision in latitude and longitude (for geospatial location).	7 days	Around 6.048 Billion rows (columns being: user id, latitude, longitude, timestamp); 405.55 GigaBytes of logical bytes.
Data-2: -do-	20,000 simulated human users as agents in microsimulation.	-do-	-do-	28 days	Around 48.384 Billion rows (columns being: user id, latitude, longitude, timestamp); 3.17 TeraBytes of logical bytes.
Data-3: -do-	50,000 simulated human users as agents in microsimulation.	-do-	-do-	60 days	Around 259.2 Billion rows (columns being: user id, latitude, longitude, timestamp); 16.97 TeraBytes of logical bytes.
Data-4: Commercially available data obtained on users in MSA of Knoxville, U.S. The users represented in the data are based on data sources of the commercial data provider (possible data sources are various commercial mobile apps and check-ins of consenting users).	162,491 mobile devices represented in the commercial data.	Significantly more sparse than even every 1 hour, and also significantly non-uniform data in space and time. Sparseness and non-uniformity vary considerably across users in data.	Significant majority of latitude and longitude values have at least 4 decimal digits precision.	7 days	Around 98.334 Million rows (columns being: user id, latitude, longitude, timestamp); 11.91 GigaBytes of logical bytes.

TABLE I: Four different large-scale geospatial-temporal human mobility datasets, on which case studies have been done to use *DICER* system solution for scalable processing. Various properties and features of these datasets are described in this table.

in the computer cluster overall performs the same computing task but assigned with different input parameters (e.g., user id or geospatial hexagon polygon here) and thus different input data slice to perform computation on. The dynamics of distribution and assignment of input parameters to the workers are controlled and scheduled by software driver (here an Apache Spark driver pod).

Figure 2 illustrates the overall operational workflow in *DICER* for highly scalable cluster computing on geospatial-temporal mobility data. At the start a Spark job is submitted through *kubectl* command tool, which invokes a Spark Operator [4] YAML file consisting of major configuration parameters. One of the configuration parameters is the path to compiled Docker image bulit with a Docker engine. The Docker image is built using a Docker file and the Docker container that holds the entire *TraKit* software package designed by us. *TraKit* is an innovative framework from our research for comprehensively evaluating geospatial-temporal human mobility data. Detailed description of *TraKit* is out of scope of this work, the only relevant components of *TraKit* in this presented work context (preprocessing of mobility data and mobility related metrics) are described in next subsection IV-C. Another important configuration parameters in the YAML file is the main application python file, which consists of following steps implemented: (i) first querying the data slice from the complete database (here Google BigQuery) based on assigned input parameters (e.g., user id, geospatial polygon) and computing component (preprocessing or metrics modules of TraKit, individually or in batch); (ii) then calling the specific python function for the computing component of interest (preprocssing or metric) in the *TraKit* software package; (iii) finally directly (without sending to the Spark driver) writing the resulting data to the relevant output data table in the database (this avoids overloading the Spark driver). The other configuration parameters in the YAML file include: system configurations for cluster computing nodes like memory and CPU capacity assigned for each node; configurations for Apache Spark engine (like Spark version, memory assigned for Spark driver and the Spark executors, etc.); configurations for Kubernetes engine (like dynamic allocation of minimum and maximum executors for autoscaling). A few other configuration parameter and procedure used are: a persistent volume (disk storage) for storing all the logs from different Spark job sessions (since the logs on Spark driver get deleted after a Spark session ends); cluster status monitoring tool (we used K9s tool [2], while kubectl logs command is another option).

C. Processing Steps for Evaluating Geospatial-Temporal Human Mobility Data

From the large-scale of raw geospatial-temporal data table, first the relevant preprocessing modules are computed. Then from the preprocessing result tables, the different human mobility metrics are calculated. This is illustrated in Figure 3. Detailed descriptions of preprocessing and metrics modules in TraKit are out of scope of this work, but here are some brief descriptions. The preprocessing steps consist of following modules: Stop Points (SP), Unique Stop Points (USP), Set Unique Stop Points (SUSP), Timed Trips Linking Locations (TTLL) and Set Timed Trips Linking Locations (STTLL).

An individual's geospatial-temporal trajectory is considered as a combination of: displacements between locations and

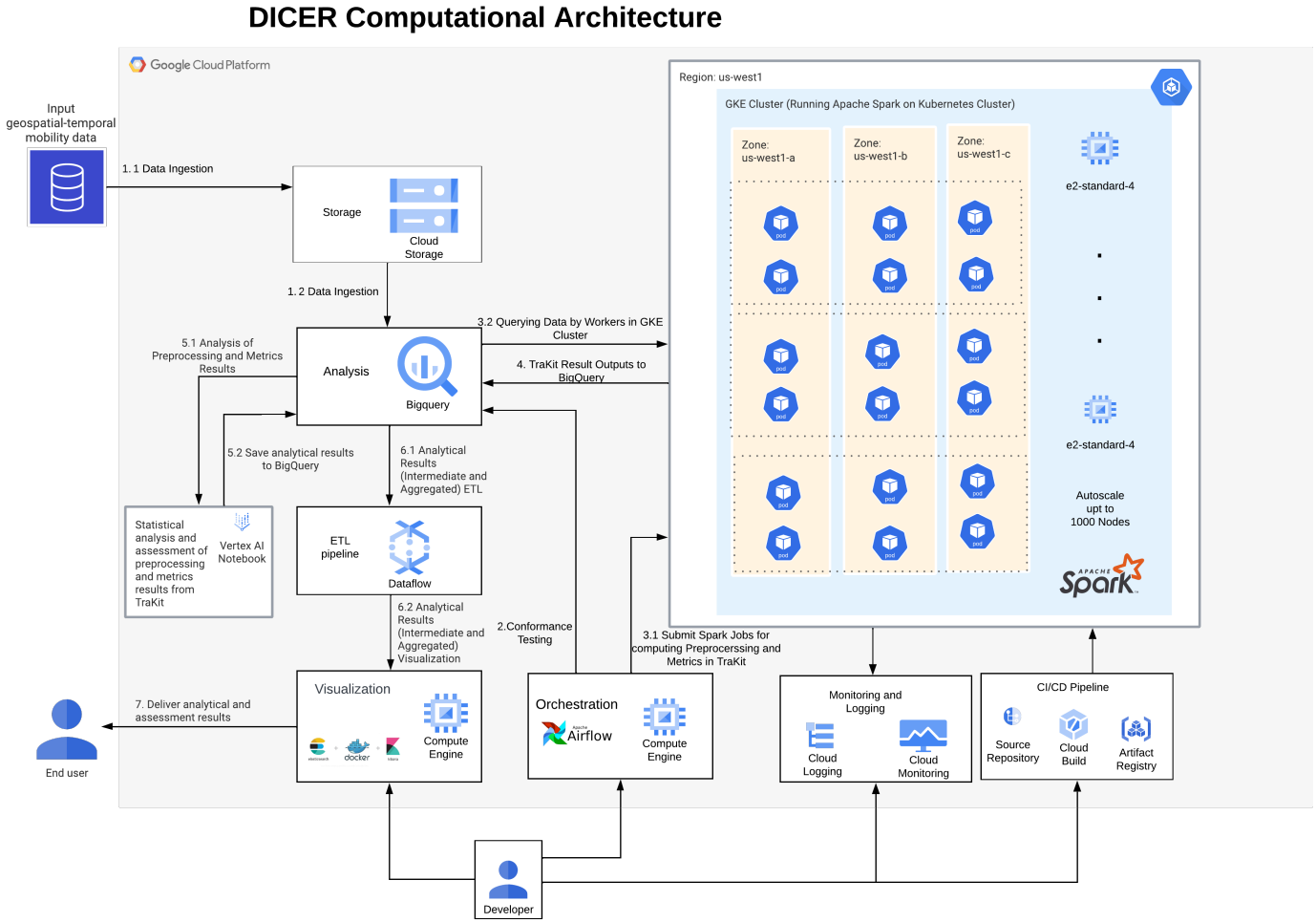


Fig. 1: System architecture of *DICER* for highly scalable cluster computing of geospatial-temporal mobility data.

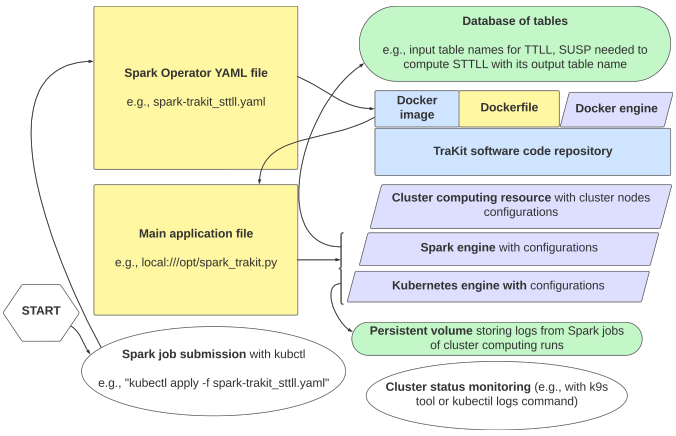


Fig. 2: Overall workflow in *DICER* for highly scalable cluster computing of geospatial-temporal mobility data.

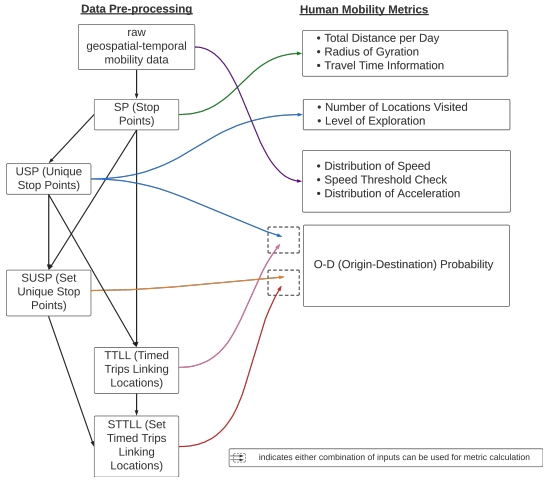


Fig. 3: Processing steps for evaluating geospatial-temporal mobility data using *DICER*.

pauses at locations. A stop location is defined by a pause an individual makes at a location. Now very briefly: SP is a pause an individual makes at a location in context to a movement trajectory; USP is re-occurring visitations at SPs by

an individual over a period of time; then SUSP is overlapping USPs from multiple individuals, aggregated together. Example of an SP is say the front location of home of user id say uid-1 where it stopped for sustained time on a certain day. Then USP is a centroid home location aggregated from SPs of that uid-1; Finally example of SUSP is a centroid home location for family members uid-1 and uid-2 residing in same home, aggregated from USPs of uid-1 and uid-2. Then TTLL and STTLL preprocessing results contain information on individual user's trips (across it's USPs) and all user population trips (across the SUSPs) respectively. Finally, following nine human mobility metrics are calculated (and then aggregated for analysis with statistical distributions) from TraKit for the case studies done (Section V to follow): Distance traveled per day; Radius of gyration; Number of locations visited; Level of exploration; Distribution of speed; Speed threshold check; Distribution of acceleration; Travel time information; Origin-destination probability. But to note that these are not the complete set of preprocessing modules and metrics modules available in our TraKit software.

V. Case Studies and Performance Benchmarking

First in Section V-A we present the case study of comparing several popular mobility metrics performance results, computed at-scale on the big data datasets *Data-1*, *Data-2*, *Data-3* (as was described in Section III and Table I). Then in Section V-B we present the case study of comparing at big data scale, the mobility microsimulations data with corresponding real-world commercial holdout data (*Data-4* in Table I).

A. Evaluating Mobility Microsimulations and their Evolution as they Scale-up

Previously Section III and Table I described the three large-scale microsmulations datasets generated by us: *Data-1*, *Data-2* and *Data-3*. The scopes of further describing these large-scale microsimulations of human mobility are though out of scope of this paper. Now using *DICER* system solution we have computed several popular and relevant human mobility metrics to do at-scale Test and Evaluation (T&E) of *Data-1*, *Data-2* and *Data-3*.

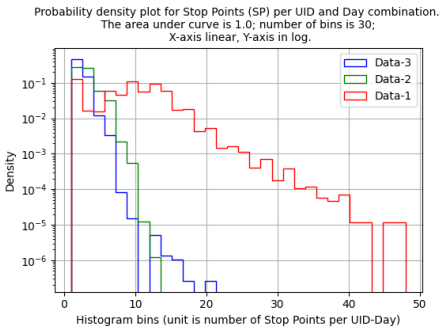
As was previously described in Section IV-C and Figure 3, we computed following nine mobility metrics at-scale: Total distance per day; Radius of gyration; Travel time information; Number of locations visited; Level of exploration; Distribution of speed; Speed threshold check; Distribution of acceleration; Origin-destination (O–D) probability. As per Figure 3 these metrics are computed in our mobility research software *TraKit* from the pre-processing results of SP (Stop Point), USP (Unique Stop Point), SUSP (Set Unique Stop Point), TTLL and STTLL (Set Timed Trips Linking Locations). Further details of *TraKit* is out of scope of this paper. Now we share the system performance and mobility characteristic performance for each of preprocessing or metrics components, and compared across *Data-1*, *Data-2* and *Data-3*. The computation times by *DICER* overall are reported in Table II.

First we compare the performance of relevant preprocessing components of SP, USP, SUSP as shown in Figure 4. In Figure 4(a) it can be seen that the range of number of SPs per UID (microsimulation user agent) per day was much higher in *Data-1*, compared to the same in *Data-2* and *Data-3* (although the high-level trend or shape of distribution was not significantly different). To observe that as significantly more user agents and days of mobility were simulated in *Data-2* and *Data-3*, the quality of microsimulation progressed and aggeragted pattern of SPs (temporal stop/stay behaviors) settled better than in *Data-1*. There was different observations from distributions of USP and SUSP as can be seen in Figure 4(b) and Figure 4(c). The distribution of USP (distribution of number of USPs per UID) and also the distribution SUSP (distribution of number of UIDs visiting each of the SUSP) did not change drastically for *Data-1*, *Data-2*, *Data-3*. The range on the X-axis extended/increased (in distribution of USP and SUSP) from *Data-1* to *Data-2* and then to *Data-3*. This is explainable because as more number of days as well as more number of UIDs are simulated, there is gradual diversity and thus increase number of unique locations being visited. Noted than USP and SUSP are not function of time itself, but SP is.

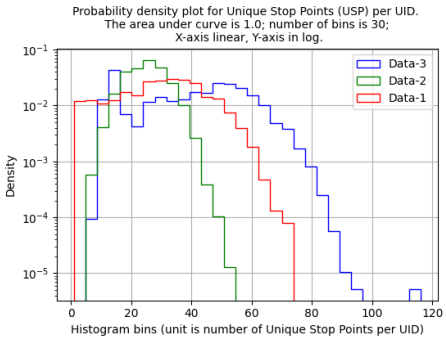
Among the other preprocessing result module, STTLL (Set Timed Trips Linking Locations) consists of a network graph where the nodes are the SUSPs and edges contain all details of individual trips (with details like which UID took a trip from trip start and end timestamps) between two SUSPs. As calculated by *DICER*, *Data-1* STTLL network graph had 16,845 nodes, 295,748 edges, and total detected 475,210 individual trips. *Data-2* STTLL network graph had 26,690 nodes, 638,480 edges, and total detected 1,524,155 individual trips. Then *Data-3* STTLL network graph had 26,105 nodes, 913,045 edges, and total detected 2,080,086 individual trips.

Now we describe the nine mobility metrics performance. The metric "total distance per day" is total distance traveled (travel or trips happenning in between stop/stay locations) per-day per-UID. Figure 5(a) shows the distribution of distance traveled per day per UID. It can be observed that *Data-1* for some reason had significantly more distance being traveled per day by user agents, compared to in *Data-2* and *Data-3*. Next metric is "radius of gyration" which quantifies the spatial extent of an individual user's characteristic distance traveled. Figure 5(b) shows the distribution of radius of gyration. It can be seen that overall distribution pattern and range for radius of gyration did not change much across *Data-1*, *Data-2*, *Data-3*. Next metric is "travel time information", which is travel time (in seconds) for travel/trips happening between two time consecutive stop/stay locations for each UID. Figure 5(c) illustrates the distribution of travel time information. It can be observed that the travel time had considerably higher values (on X-axis) in *Data-3* compared to in *Data-1* and *Data-2*.

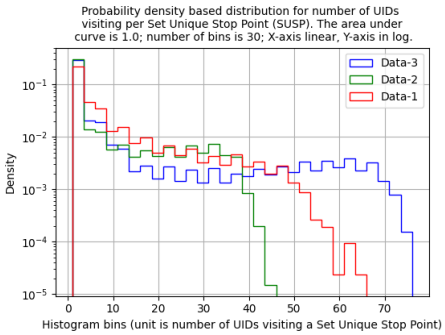
The next metric is "number of locations visited", which represents the number of unique locations visited per day by each UID. Figure 6(a) shows the distribution of number of USPs visited per-UID per-day. It can be observed that there was a little bit more number of unique locations being visited



((a)) Probability density based distribution for number of Stop Points (SP) for each UID-Day combination.

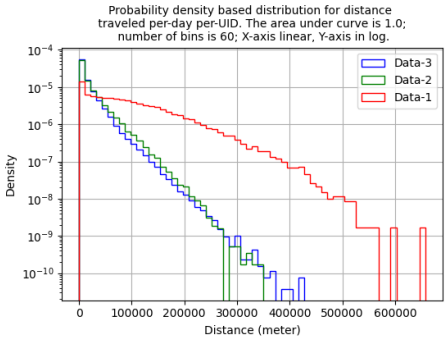


((b)) Probability density based distribution for number of Unique Stop Points (USP) per UID.

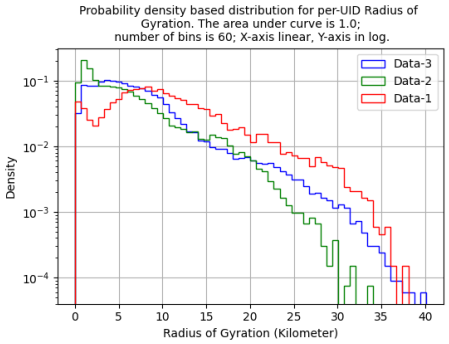


((c)) Probability density based distribution for number of UIDs visiting each of the Set Unique Stop Points (SUSP).

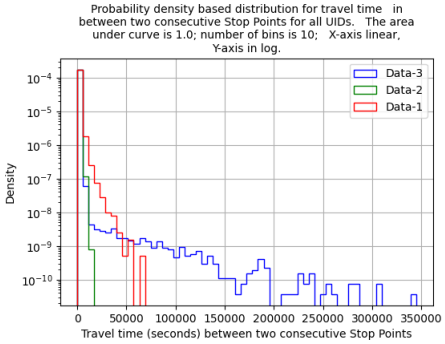
Fig. 4: Statistical distribution based analysis for SP, USP, SUSP preprocessing modules of TraKit. SP, USP, SUSP results are computed at-scale using our *DICER* system solution.



((a)) Distribution of metric total distance per day.

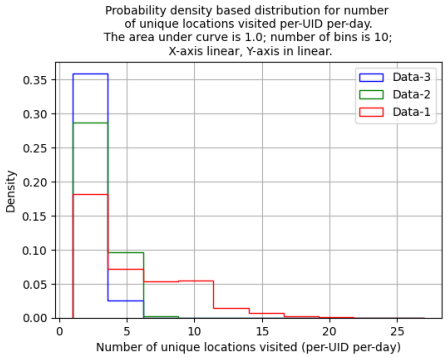


((b)) Distribution of metric radius of gyration.

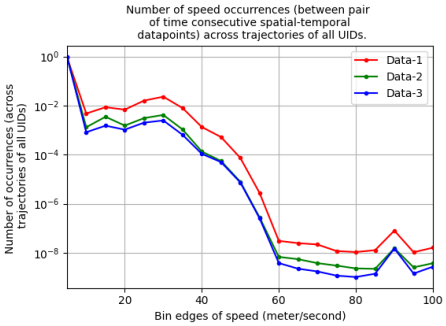


((c)) Distribution of metric travel time information.

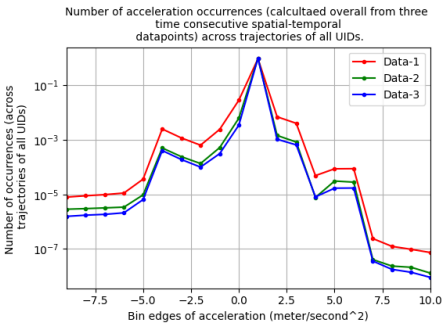
Fig. 5: Statistical distribution based analysis for following mobility metrics as computed on each of *Data-1*, *Data-2* and *Data-3*: total distance per day, radius of gyration, travel time information.



((a)) Characteristics of metric number of locations visited



((b)) Characteristics of metric distribution of speed.



((c)) Characteristics of metric distribution of acceleration.

Fig. 6: Statistical distribution based analysis for following mobility metrics as computed on each of *Data-1*, *Data-2* and *Data-3*: number of locations visited, distribution of speed, distribution of acceleration.

per-day overall by the UIDs in *Data-1* compared to in *Data-2* and *Data-3*, but it was not significantly higher. Next metric is "level of exploration", which is defined and calculated as visitation frequency to the top-K (K=10 used here) visited unique locations by each UID. As can be observed in Figure 8(a) the visitation frequency(%) vs. ranked top-10 visited unique locations plot didn't change much and fundamentally remained similar across the *Data-1*, *Data-2* and *Data-3*.

The next two metrics compared are the "distribution of speed" and "distribution of acceleration". Unlike several other metrics of computing from preprocessed results (like from SP, USP, SUSP), these two metrics are computed from the very raw datapoints in the geospatial-temporal input data. Each speed occurrence is calculated from pair of time consecutive datapoints of an UID and unit being meters/second. Then each acceleration occurrence is calculated from set of three time consecutive datapoints of an UID (essentially computing the change of speed in two time consecutive speed occurrence values), and unit being meters/second^2. Figure 6(b) and Figure 6(c) show that the distribution of speed and distribution of acceleration, both overall did not change much across *Data-1*, *Data-2*, *Data-3*. One point to note that in representing the histogram here as line plot (in both Figure 6(b) and Figure 6(c)), while each lineplot datapoint consist of normalized frequency value or height in histogram and bin in histogram (speed bin or acceleration bin), the right edge value of the bin was chosen (instead of choosing the left edge of bin, or just using the average of right bin edge and left bin edge). This affects the visualization of the plots in Figure 6(b) and 6(c).

Another metric calculated was "speed threshold check", which detects and counts the number of speed occurrences where speed exceeds 100 meters/second. This speed threshold value is chosen based on microsimulation modeling scopes (that were used behind generating *Data-1*, *Data-2*, *Data-3*), which is mode of transport as personally owned vehicles but not modes like train, bus, airplane. In this metric performance: *Data-1* had 525 UIDs (out of 10K UIDs) having total 592 speed violations (out of about 6,048,000,000-10,000 speed occurrences); *Data-2* had 1,844 UIDs (out of 20K UIDs) having total 3,344 speed violations (out of about 48,384,000,000-20,000 speed occurrences); and *Data-3* had 1,713 UIDs (out of 50K UIDs) having total 2,371 speed violations (out of about 259,200,000,000-50,000 speed occurrences).

Lastly, Figure 7 illustrate normalized version of Origin-Destination (O–D) matrix calculated and compared across *Data-1*, *Data-2*, *Data-3*. The O–D matrix before normalization represents the total number of individual trips that occur between each pair of Uber H3 hexagons in the area of interest. The H3 hexagons with spatial resolution Res-5 [5] were extracted that are geospatially present (completely or partially) inside the single bounded polygon of the AOI (area of interest) in each of *Data-1*, *Data-2*, *Data-3*. Then from the pre-normalized O–D matrix, it is normalized in following manner. Population trip counts from the H3 hexagons "to" the H3 hexagons are normalized so that for *each from-hexagon*, the sum of trip fractions to *all the to-hexagons* is 1.0. Now comparing this normalized (with the "to" aspect) O–D matrix across *Data-1*, *Data-2*, *Data-3* in Figure 7, it was observed that the O–D trips patterns improved (more geospatial distribution of trips occurring across the H3 hexagons) in *Data-2* and *Data-3* as microsimulation evolved and improved, compared to *Data-1*. Taking the set of unique H3 hexagons in *Data-1*, *Data-2*, *Data-3*, there were total 50 unique H3 hexagons (Uber H3 spatial resolution Res-5 [5]) present (and plotted in X and Y axis of the heatmap in Figure 7).

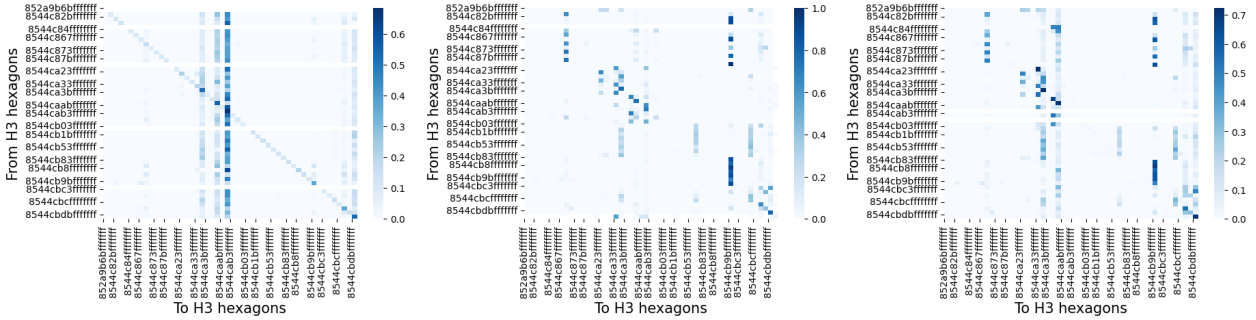
Overall Section V-B has validated the capability of *DICER* system solution in computing large range of preprocessing and mobility metrics rapidly and in highly scalable manner, and the results then enabling deep and multi-faceted analytics and comparsion of mobility metrics performance. Note again that computation times by *DICER* overall are reported in Table II.

B. Comparing at Large-scale, the Mobility Microsimulations Data with Real-world Commercial Holdout Data

In this Section we have validated how using *DICER* system, commercial holdout human mobility data can be processed at-scale, for then comparing with the microsimulation output for building reference and benchmarks to real-world human mobility. The large-scale commercially purchased (from relevant data vendors in the market) off-the-shelf dataset is described as *Data-4* previously in Section III and Table I. We have computed the pre-processing modules in TraKit (SP, USP, SUSP, STTLL) at-scale on *Data-4*, and then compared distribution with the same in *Data-1*, *Data-2*, *Data-3*.

To note that the commercially available holdout data in *Data-4* is just 7 days of data from the overall same AOI (area of interest) of Knoxville, Tennessee, U.S. This time duration of 7 days matches with that of *Data-1*, but not with time duration of *Data-2* or *Data-3*. More data acquiring from the commercial data vendors is an ongoing work by us. Besides this, another challenge in *Data-4* and also widely in any commercially available geospatial-temporal human mobility data is that the datapoints are: sparse both temporally and geospatially; non-uniform across space and time and across user ids; also have noise added both for privacy and due to instrumentation/data source used. These challenges or deficiencies in data are not present in microsimulation driven human mobility data generation. Therefore fair comparison of at least *Data-2* and *Data-3* with commercial holdout data is a work-in-progress by us.

Figure 8(b) and Figure 8(c) present the computed distribution pattern for *Data-4* (compared with *Data-1*, *Data-2*, *Data-3*) for the preprocessing component of SUSP (set unique stop point). The extended range of X-axis for *Data-4* in this distribution of SUSPs is potentially due to higher number of users present in the dataset. To note that due to space limitations, the distributions for SP and USP could not be shown here. As for computation time with *DICER*: SP computation time was about 2 hours (max 150 parallel workers in cluster); USP computation time was about 1 hour



((a)) Normalized O–D for *Data-1*. ((b)) Normalized O–D for *Data-2*. ((c)) Normalized O–D for *Data-3*.

Fig. 7: Origin-Destination (O–D) probability matrix calculated for *Data-1*, *Data-2*, *Data-3*. The population trip counts from the H3 hexagons to the H3 hexagons are normalized so that *for each from-hexagon*, the sum of trip fractions *to all the to-hexagons* is 1.0.

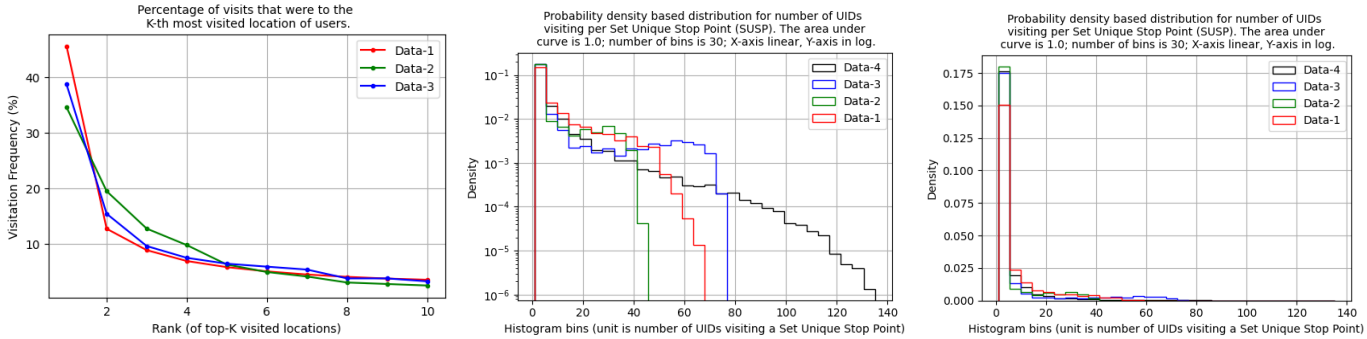
Computation module	**Computation time on Data-1 dataset**	**Computation time on Data-2 dataset**	**Computation time on Data-3 dataset**
Pre-processing: Stop points (SP)	1 hour 10 minutes, max 150 parallel workers	6 hour 48 minutes, max 150 parallel workers (more workers can make it faster)	about 12 hours, max 200 parallel workers (more workers can make it faster)
Pre-processing: Unique stop points (USP)	7 minutes, max 150 parallel workers	8 minutes, max 150 parallel workers	20 minutes, max 200 parallel workers
Pre-processing: Set unique stop points (SUSP)	2 hours 29 minutes, max 150 parallel workers (compute time might have been affected by our other computing jobs simultaneously accessing the Google Cloud BigQuery database I/O bandwidth)	1 hour 35 minutes, max 150 parallel workers	1 hour 35 minutes, max 300 parallel workers
Pre-processing: Set Timed Trips Linking Locations (STTLL)	2 minutes, max 150 parallel workers	3 minutes, maximum 150 parallel workers	15 minutes, max 150 parallel workers
Three metrics computed in batch: (i) Distance traveled per day; (ii) Radius of gyration; and (iii) Number of locations visited	8 minutes, max 150 parallel workers	45 minutes, max 150 parallel workers	about 5 hours, max 150 parallel workers (more workers can make it faster)
Metric: Level of exploration	6 minutes, max 150 parallel workers	30 minutes, max 150 parallel workers	about 2 hours, max 150 parallel workers
Multiple metrics: (i) Distribution of speed; (ii) Speed threshold check; (iii) Distribution of acceleration	about 30 minutes, used Google Cloud BigQuery SQL (instead of using *DICER* directly)	about 45 minutes, used Google Cloud BigQuery SQL (instead of using *DICER* directly)	about 1 hour, used Google Cloud BigQuery SQL (instead of using *DICER* directly)
Metric: Travel time information	3 minutes, maximum 150 parallel workers	10 minutes, max 150 parallel workers	1 hour 10 minutes, max 150 parallel workers (runtime may have been affected by our other computation jobs accessing database I/O bandwidth)
Metric: Origin-destination probability	2 seconds, used Google Cloud BigQuery SQL (from inputs H3 hexagon data, and STTLL from *DICER*)	3 seconds, used Google Cloud BigQuery SQL (from inputs H3 hexagon data, and STTLL from *DICER*)	3 seconds, used Google Cloud BigQuery SQL (from inputs H3 hexagon data, and STTLL from *DICER*)

TABLE II: Computation time by *DICER* system for calculating different pre-processing and metrics in human mobility. Computation times are reported on different mobility datasets *Data-1*, *Data-2*, *Data-3* along with notes in some cases.

30 minutes (max 150 parallel workers); SUSP computation time was 30 minutes (max 150 parallel workers). To note that computing the metrics on *Data-4* and comparing with *Data-1*, *Data-2*, *Data-3* are still work-in-progress for us.

VI. Conclusion and Discussion

There are compelling needs in the research of large-scale or big data scale analysis of geospatial-temporal human mobility data, in terms of more sharing of: system architectures very specific to this application; process workflow of scalable computation; detailed case studies with data at big data scale. In an effort to address these needs, in this paper we have presented *DICER* (Data Intensive Computing Environment and Runtime) system solution for processing and analytics of geospatial-temporal data at big data scale. Detailed description of *DICER* is provided: application specific system architecture, process workflow for scalable computing jobs, hierarchical processing steps in computing varied mobility metrics. Then extensive case studies are presented with detailed system performance and mobility metrics characterization regarding scalable computing with *DICER* on large-scale mobility data



((a)) Characteristics of metric level of exploration. The plot is visitation frequency(%) vs. ranked top-10 visited unique locations, aggregated across all UIDs.

((b)) Probability density based distribution for number of UIDs visiting each of the Set Unique Stop Points (SUSP). Y-axis is in *log scale*.

((c)) Probability density based distribution for number of UIDs visiting each of the Set Unique Stop Points (SUSP). Y-axis is in *linear scale*.

Fig. 8: Figure (a): Metric metric level of exploration computed across *Data-1*, *Data-2*, *Data-3*. Figure (b) and (c): Distribution of SUSP (Set Unique Stop Points) computed on *Data-4*, and compared with same on *Data-1*, *Data-2*, *Data-3*.

from microsimulations and real-world commercial data. Future works include increasing the collection of commercial holdout data, and then computing and characterizing the metrics on the holdout real-world data and comparative analytics with the microsimulation outputs.

Acknowledgments:

This work is supported by the Intelligence Advanced Research Projects Activity (IARPA). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOE, or the U.S. Government.

This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (https://energy.gov/downloads/doe-public-access-plan).

References

[1] Cost-efficient open source big data platform at uber. https://www.uber.com/blog/cost-efficient-big-data-platform/.

[2] k9s. https://k9scli.io/.

[3] Safegraph architecture. https://www.safegraph.com/blog/datachronicle-architecture-achieving-traceability-for-large-scale-data-processing-in-safegraph.

[4] Spark operator. https://github.com/GoogleCloudPlatform/spark-on-k8s-operator.

[5] Uber h3 resolutions. https://h3geo.org/docs/core-library/restable/.

[6] F. Bai, Narayanan Sadagopan, and A. Helmy. Important: a framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 825–835 vol.2, 2003.

[7] Dirk Brockmann, Lars Hufnagel, and Theo Geisel. The scaling laws of human travel. *Nature*, 439(7075):462–465, 2006.

[8] Francesco Calabrese, Massimo Colonna, Piero Lovisolo, Dario Parata, and Carlo Ratti. Real-time urban monitoring using cell phones: A case study in rome. *IEEE Transactions on Intelligent Transportation Systems*, 12(1):141–151, 2011.

[9] Christos Doulkeridis, Akrivi Vlachou, Nikos Pelekis, and Yannis Theodoridis. A survey on big data processing frameworks for mobility analytics. *SIGMOD Rec.*, 50(2):18–29, aug 2021.

[10] Fosca Giannotti, Mirco Nanni, Dino Pedreschi, Fabio Pinelli, Chiara Renso, Salvatore Rinzivillo, and Roberto Trasarti. Unveiling the complexity of human mobility by querying and mining massive trajectory data. *The VLDB Journal*, 20:695–719, 2011.

[11] Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns. *nature*, 453(7196):779–782, 2008.

[12] James N. Hughes, Andrew Annex, Christopher N. Eichelberger, Anthony Fox, Andrew Hulbert, and Michael Ronquest. GeoMesa: a distributed architecture for spatio-temporal fusion. In *Geospatial Informatics, Fusion, and Motion Video Analytics V*, volume 9473, page 94730F. International Society for Optics and Photonics, SPIE, 2015.

[13] Pan Hui, Richard Mortier, Michal Piórkowski, Tristan Henderson, and Jon Crowcroft. Planet-scale human mobility measurement. In *Proceedings of the 2nd ACM International Workshop on Hot Topics in Planet-Scale Measurement*, HotPlanet '10, 2010.

[14] Yuanyuan Qiao, Yihang Cheng, Jie Yang, Jiajia Liu, and Nei Kato. A mobility analytical framework for big mobile data in densely populated area. *IEEE Transactions on Vehicular Technology*, 66(2):1443–1455, 2017.

[15] Jonathan Reades, Francesco Calabrese, Andres Sevtsuk, and Carlo Ratti. Cellular census: Explorations in urban data collection. *IEEE Pervasive Computing*, 6(3):30–38, 2007.

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

[17] Gautam S. Thakur and Ahmed Helmy. Cobra: A framework for the analysis of realistic mobility models. In *2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 145–150, 2013.

[18] 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*, 58:501–523, 2019.

[19] George A. Vouros, Apostolis Glenis, and Christos Doulkeridis. *The δ Big Data Architecture for Mobility Analytics*, pages 315–339. Springer International Publishing, Cham, 2020.

[20] Jia Yu, Jinxuan Wu, and Mohamed Sarwat. Geospark: a cluster computing framework for processing large-scale spatial data. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '15, 2015.