Spatially Representative Online Big Data Sampling for Smart Cities

Isam Mashhour Al Jawarneh, Paolo Bellavista , Antonio Corradi , Luca Foschini , Rebecca Montanari

Dipartimento di Informatica – Scienza e Ingegneria, University of Bologna
Viale Risorgimento 2, 40136 Bologna, Italy
{isam.aljawarneh3, paolo.bellavista, antonio.corradi, luca.foschini, rebecca.montanari}@unibo.it

***Abstract*— The diversity of sensing options that IoT offers imposed requirements to evolve stream processing engines so to cope with highly heterogeneous and fast-pace data streams challenging their computing capacities. Location intelligence applications aim at exploiting those geo-referenced data in generating visualizations and dashboards that provide deep insights for assisting decision making in smart cities and urban planning. As data arriving are mostly geo-referenced and the rate is fluctuating in pace and skewness, computations upon streams should depend on approximation by applying methods such as sampling. Representativeness in sampling designs remains the pivotal concern in the literature. In spatial data streams contexts, it loosely means selecting proportional counts of spatial tuples from each group of tuples that belong to the same real geometry (i.e., geographically residing in the same proximity) within each streaming time window. This is challenging in streaming settings because spatial data is parametrized, loosing hence it is real geometries, which requires costly geometric operations to project them back to maps. To close this void, we have designed SpatialSPE in a previous work and incorporated an efficient fine-grained spatial online sampling method (SAOS) transparently within its layers. In this paper, we extend SAOS (the novel method is termed ex-SAOS) by new features that allow efficient online spatial sampling on a coarser level, which is a requirement in smart city scenarios. Our results show that ex-SAOS is efficient and effectively extends SAOS for more general smart city and urban computing scenarios.**

***Keywords*— spatial sampling, spark streaming, smart city, stratified sampling, geohash.**

I. INTRODUCTION

The widespread adoption of IoT devices have caused avalanches of geo-referenced data streams to flow endlessly and feed DSMSs, and specifically Stream Processing Engines (hereafter SPE for short) [1]. The timely exploration of those streams offers deep insightful analytics that assist strategic planning in all aspects of our lives, including city planning, urban computing, and health care [2]. *Low-latency* and *high-estimation-quality* (lowering the error-bound tied to such an approximation) are the two greatly antithetical QoS goals that need to be trade off in a plausible way. Latency is the total time required for processing all streaming tuples in an end-to-end way. Error-bound tied to such an approximation determines the estimation quality. Those QoS goals are colliding in such a way that lowering the latency may force lower sampling fractions, which in turns, lead to an undesirable lower estimation-quality. A well-performing solution would search for suitable parameters that optimize both QoS goals. Deterministic solutions, where exactness is required, cannot normally strike a plausible balance between those contradicting QoS goals. Thus, Approximate Query Processing (AQP) lends itself as an alternative probabilistic path that has shown promising in striking a balance between QoS goals. The fact that, more than often, users are willing to abandon tiny error-bounded estimation quality by accepting a small reduction in the gain profit margin for the benefit of even a small latency gain. In other terms, it is important to comprise an acceptable degree of exactness but on the price of avoiding the slowness induced by an exhaustive search, thus striking a balance between conflicting QoS. AQP depends on many data size reduction techniques, from which sampling presents itself as a leading solution. Sampling means selecting a portion of the total data (i.e., population) and compute an error-bounded statistic based on that portion. A great challenge relates to designing a sampling scheme that can select representative samples that yield estimations with rigorous error-bounds [3]. Most online sampling methods embrace randomness, by depending on sampling schemes that are based on random sampling. However, most interesting data are highly skewed (as opposed to the normal distribution). Designs that are based on randomness proved inefficient for non-uniformly distributed data such as geospatial data. In real scenarios, data streams are geo-referenced and being attuned to this characteristic in every aspect of the SPE is essential for location intelligence to success, including the online sampling scheme. Aiming at closing those gaps, in a previous study [4], we have designed and implemented SpatialSPE (short for Spatial Stream Processing Engine), together with a specialized online spatial sampling method SAOS, which is a fast in-

memory first-in-class online spatial sampling scheme and incorporated it with an emerging SQL-like based micro-batch SPE, Specifically Spark Structured Streaming [5], (SpSS as a shorthand).

In this paper, we introduce an extended version of SAOS (termed ex-SAOS) that allows coalescing spatial data on a coarser level before applying the spatial sampling method. More in details, the plain method SAOS selects proportional data from each group of spatial points falling under the same fine-grained group (i.e., geohash, which is a regularly-shaped region in the space). On the contrary, ex-SAOS (short for extended SAOS) extends SAOS by allowing coarser levels of granularity (i.e., withdrawing points from irregularly-shaped polygons, known as neighbourhoods, districts, boroughs etc., in city management terms). By introducing ex-SAOS, we make the following contributions in this paper. First, we enrich the applicability of SpatialSPE so that we streamline its adoption for smart city scenarios. This is because we enable elasticity in the level of sampling granularity. For example, a coarser level based on neighbourhoods of a city. The second contribution is that we have built a standard-compliant prototype on top of an emerging stream processing engine (Spark Structured Steaming [5] , SpSS hereafter for short) that is the first-in-class providing a declarative SQL-like API for stream processing, following the trending layered-up software stack. We have further extended our previous retrofitting of the SpSS query incrementalizer so that it becomes aware of the spatial approximate queries on a coarser level (arbitrarily-shaped polygons instead of regularly-shaped geohashes in our previous work). Incrementalization means that results accuracy will be improving stepwise. Queries include single spatial queries, such as approximating a study target variable (e.g., the '*average'* or '*total'* of a variable). We also support spatial online aggregations, such as Top-N rank geo-statistics. To the best of our knowledge, we are not aware of any system from the relevant literature that achieves these goals.

II. Related Works

From the relevant literature, few works apply dimensionality reduction-based approaches such as the works by [6-8]. Nevertheless, those are compute-intensive and thus are considered inapplicable in distributed online computing deployments. As a clear example on dimensionality reduction, [9] have designed a method for finite populations, which they term as generalized random-tessellation stratified (GRTS) , which is based on transforming the two-dimensional into a lower-dimensional survey space. Thereafter, arbitrarily ordered spatial addresses are generated and a systematic sampling is applied to withdraw a well-balanced random representative sample. The idea resorts to the fact that geometrically-nearby objects which are proximate in the two-dimensional planar geometry end up in nearby locations when mapped to a one-dimensional counterpart. However, we argue that well-spread sample does not directly imply well-representativeness, where the systematic module can unfairly ignore some regions while withdrawing the samples. Along the same lines, [7] introduces a sampling method that depends also on a dimensionality reduction approach that is based on space-filling curves. They depend on the ordering that is offered by space-filling curves so that contiguously numbered points are representing a well-balanced spatial sample. Those works from the literature are inapplicable for distributed computing environments. They are not able to achieve incremental geo-statistical computation approximation results that improve stepwise as time ticks forward. To close that void, in a previous work, we have designed SpatialSPE, a spatial processing engine with a specialized spatial-aware sampling method (SAOS) [4] . SpatialSPE has introduced incrementalization over geo-referenced data streams using a declarative API, a target that that was completely novel at the time. By the time, we have relied on dimensionality reduction approach (specifically geohash) for selecting proportionate spatially-representative samples. In the plain implementation of SAOS, we have selected the same percentage of points for each geohash during each time interval (known as batch interval in online stream processing parlance). Geohashes can be heuristically thought of as grid squares resulting from the division of flattened planar geometry (the survey area). SAOS simply works by first calculating the covering geohashes for each administrative part of a city (normally known as neighbourhoods), thereafter selecting the same percentage from each group points having the same geohash arriving during a batch interval means that we approximately choose fair amount of points from each neighbourhood. However, geohash encoding is an approximation and few points may have the same geohash despite belonging to different neighbourhoods (a problem known as 'false positives' or 'edge cases'). This is caused by the approximation that depends on Minimum Bounding Rectangles (MBR), which intersect causing the accumulation of 'false positives' on the intersections. A refinement step is needed then in case we are interested in specifying to which space region a point belongs in real geometries.

To close this gap, in this paper, we further extend SAOS (we term the new version as ex-SAOS) so that we discard the 'false positives' before performing the sampling. In this way, we guarantee that we exactly are selecting the same proportion of points from each polygon (city official administrative divisions such as neighbourhood, boroughs, districts, etc.,). The new design we envisaged by ex-SAOS is targeting smart city scenarios, where cities are normally divided by municipalities into administrative parts (known as neighbourhoods, boroughs, districts, etc.). Our ex-SAOS method is general and can be applied to city administrative divisions of any kind and shape.

III. Theoretical foundations

A. Short Primer on sampling

Sampling loosely means selecting a miniature of a population aiming at estimating a population quantity (e.g., 'average' or 'count' of a target variable). Population is all units present in a survey region. For instance, 'all trees in a forest', where sampling is normally applied for estimating the 'average tree basin size'. Estimators are tied to variance statistics that measure their accuracy [10] . The sampling design specifies the way we select sampling sites and samples from a population. Good designs are those that can select representative samples, in which case samples are considered microcosms versions of populations they are representing. Those samples are expected to be used for yielding estimates with a known degree of accuracy or confidence, thus avoiding

sampling biasedness by not overlooking some groups of the population [10]. The degree of accuracy is measured normally by applying Standard Errors (SE) which appear because of depending on a sample instead of the population for calculating target variable's estimates.

The two most widely adopted sampling design in the literature are simple random sampling (SRS) and Simple Stratified Sampling (SSS). SRS assigns equal selection probabilities to all units of a population. It then assigns labels to every unit and selects labels arbitrarily until reaching the sample size threshold. On the other side, SSS selects proportional units from each group set in a stratified population. Sampling students from schools, we take 50% boys and 50% girls, where boys and girls are stratum in this case. In short, the distinction between the two designs is that SSS applies SRS within each group (stratum) in the population [11].

B. Spatial Online Sampling Designs

Applying purpose-built sampling designs to geo-referenced data is known as *spatial sampling*, and it is widely adopted in a variety of real-life domains such as environmental monitoring [12] . It can be loosely defined with a ternary $(\psi, \Im, \Re)$, where $\Re$ is the embedding geometrical space from which we withdraw samples, $\Im$ is the sampling frame or design (e.g., SRS, SSS) overlaying the survey area (i.e., the embedding geometrical space), ψ is a statistic applied for estimating a target variable (e.g., 'total' and 'mean' of a target variable). The proper choices of $\Im$ and ψ heavily affect appropriateness of a spatial sampling design [12]. Reducing the sampling variance in spatial settings means selecting spatially representative samples, which take the spatial characteristics into considerations, such as the spatial distribution of the objects in the survey area [13]. This is normally achieved by preserving the so-called spatial co-locality [14] . Thus, respecting Tobler's first law of geography which lays down the foundations of spatial co-existence of objects in real geometries, stressing the fact that nearby spatial entities are more autocorrelated than those far apart [15]. A heuristically valid solution for preserving such co-location trait is by imagining the earth flattened out as a two-dimensional planar irregular grid-like representation, thereafter sampling proportional quantities from each group in each subregion of that embedding space (representing cell or polygon in a grid-based representation overlaying the study area), which normally leads to accountable statistical estimations with minimized errors [12, 15]. Estimation quality is a highly enforced QoS goals in SLAs for all smart city applications. Thus, requiring solutions to adapt to a predefined set of QoS goals. In the case of estimations that are based on sampling for example, this could be achieved by lowering the estimations variances, which, in turns, leads to lower SEs. A contradicting accompanied QoS target is normally lowering the latency. Those stretched contradicting QoS goals are hardly achieved by SRS-based designs applied in smart cities. Even if it, by chance (based on a specific data distribution), performs well at times, it fails at most other times. This is simply because SRS-based designs overlook study regions by selecting disproportionally unfair number of entities from each group of naturally stratified study area such as spatially-rich survey areas.

The overarching traits offered by stratification-based sampling designs have encouraged us to adopt a stratified-based design for selecting well-representative samples from environments that are rich with patchy spatial distributions, where spatial objects are normally clumped into few patches. This is based on the observation that spatially co-located objects share the same characteristics normally [16, 17] . Thus, applying a stratified-like design means selecting proportionally fair number of samples from each group in the spatial population. Having said that, selecting well-geographically spread-out samples is known to yield better estimation for study target variables. We term samples satisfying those properties as *spatially representative samples*.

Online sampling imposes harsh constraints that do not normally affect designs operating on data-at-rest. We term designs that are operating in non-stationary (data-in-motion) anisotropy data streaming settings as *spatial online sampling* methods. SPEs process infinite streams of big data by either applying a record-at-a-time stream processing or micro-batch processing models. In this paper, we focus on the latter, where streaming data is gathered into temporary in-memory parsimonious storage every small-time interval (known as 'batch interval' or 'trigger interval'), thereafter, the processing pipeline is applied to each micro-batch independently. The first requirement is then being able to adapt traditional sampling designs so that they can efficiently operate on often temporally and pace-wise fluctuating spatial data streaming flows. One of the demanding requirements, for example, is being able to *incrementalize* results obtained online by being able to build them up gradually as new data arrives taken into consideration that the retention policy may not allow to store historical streaming data for future re-computation. For example, in time-based micro-batching window semantics, an 'average' of a target variable should be updated after each batch interval, thus incrementalizing it.

IV. Efficient Approximate Processing for Smart Cities

A. Usage Model and Baseline System

Strategic planning managers in municipalities of metropolitan cities need smart software designs that enable them to easily comprehend the big picture so that they can plan for more sustainable city resources. This is normally achieved by serving high-level views of aggregations of the facts inherited from huge amounts of data on the form of dashboards and heatmaps. Those target outcomes pass through complex pipeline software engines, for example an application of complex clustering algorithms [2]. Map rendering with a full population overlayed on parsimonious space-constrained devices could easily turn the view unclear, thus hindering the decision-making process. This is normally caused by the fact that spatial entities gather mostly in specific places at same times affected by autocorrelation properties.

Consider a toy example of an interactive geospatial query that requests to interactively generate heatmaps of "human and vehicles in-motion grouped by district in the city of Rome in Italy". In a rush hour, were objects are normally clumped into few districts (such as city centre) may easily results in a clutter. In this simple case, a natural solution is spatial online sampling, where we select spatially representative samples from each neighbourhood (boroughs, districts, and any other irregularly shaped regions of the administrative divisions of a city)

Algorithm 1: Extended- Spatial-Aware Online Sampling (ex-SAOS)

ex-SAOS (tuples$_i$, samplingMap, coverGeo, sampFraction, seed)
r = rand(seed), sample ← {}
//perform inner join on geohash
joinResult = tuples$_i$.join(coverGeo)
Foreach tuple t **in** joinResult **do**
//return the polygon to which this tuple belongs
  polygon ← getPolygon (t)
  //get sampling fraction for this polygon *key* = fraction$_i$, *or zero*
  fraction$_i$ ← samplingMap.getOrElse(polygon,0.0)
  //toss a coin selecting items from each polygon *in current batch*
  If (P (r < fractioni)) S.put(tuple)
return S

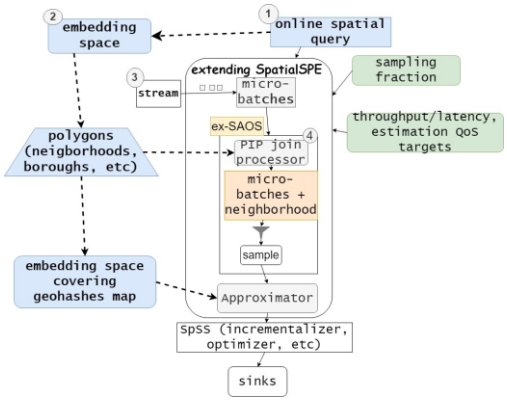

Fig. 1 SpatialSPE transparently enriched with ex-SAOS

independently. Our baseline system is a spatial online sampling design that selects proportionally fare counts of spatial entities from equally-sized and equally-shaped spatial regions. Our baseline is a sampling method that we have designed in previous work (termed as SAOS [4]). Imagining Earth flattened out, SAOS divides the survey geometric area into equally-sized and regularly-shaped (squares) regions of specific length. It does so, by overlaying the study area with a regular grid, thereafter, imposing a z-order curves-based indexing known as geohashes. Geohash [1] is simply an approximation dimensionality reduction approach that transforms GPS coordinates (latitudes/longitudes) into a single string representing regularly-sized and shaped squares in a grid, where geometrically proximate points have the same geohash, thus residing in the same geohash-represented square in real geometries [18]. This is normally susceptible to a degree of inaccuracy where 'false positives' result from two points falling far apart in real geometries while having the same geohash. This is caused by the fact that grid-based division of flat geometry (that is applied by SAOS) assumes a perfectly flat surface, thus causing distortions near the earth poles [19]. Geohash was selected in our previous work as it is the computationally cheapest dimensionality reduction approach among others (including googles S2[2] and Uber's Hexagonal Hierarchical Geospatial Indexing, H3[3]) To close this gap, in this paper we design an extended version of SAOS (we term the novel method as ex-SAOS) and incorporate it within the layers of SpatialSPE so that SAOS and ex-SAOS reinforce and enrich each other without their limitations. In addition to this baseline, ex-SAOS is compared to an SRS analogous design that samples randomly with equal inclusion values for all points in the sampling area, thus unduly overlooking regions, resulting in maps that are not well-representing distributions in real geometries, which negatively affects the decision making in smart cities. This usage model communicates the necessity for an online spatial sampling design that considers arbitrarily-shaped regions in a city.

B. Incorporating ex-SAOS into SpatialSPE

To efficiently be able to draw spatially representative samples from arbitrarily sized regions in a study area, we have designed and extended version of SAOS (the new version is termed as ex-SAOS, for extended-SAOS) and incorporated it within the layers of SpatialSPE as illustrated in the context diagram of figure 1.SpatialSPE receives the online spatial query in addition to QoS goals (expressed as estimation quality, latency, and throughput targets). It also receives a sampling rate (e.g., calculated through an external controller). It worth noticing that sampling rates are served to the system as an external input, we are not providing any cost model that feeds a controller for mapping QoS (such as lowering latency and maximizing resource utilization) goals into an adaptive sampling rate. The incorporation works as follows. First, a file is served to the system representing the embedding space from which points will be withdrawn. The file contains polygons representing the administrative divisions of a city (known as neighbourhoods, boroughs, districts, etc.,). This file comes in many formats (including GeoJSON and shapefiles representations). We explode this file in a way that we generate a list of covering geohashes for every polygon. From the other side of the system, raw geo-referenced spatial points arrive (including GPS coordinates in the form of longitudes and latitudes). Every batch interval, a micro-batch is formed from a group of arriving tuples, those tuples are fed to ex-SAOS, which then proceeds as follows. It first applies a geohash transformation method (a cheap version having linear complexity) to all tuples of the micro-batch. A super quick spatial join algorithm from Spark's Magellan [4] is then adapted so that we join tuples in the micro-batch with the exploded polygons file (city neighbourhoods, boroughs, districts, etc.,). The result is a new micro-batch of tuples with a field that specifies to which polygon each tuple belongs. Hitting this point, spatial objects are readily stratified. Magellan stock version works with static-static join (i.e., with no streaming source), hence, with a tiny patch code we have retrofitted it so that it works with stream-static join (with one side of the join being a stream). ex-SAOS then selects fairly proportional amounts from all polygons and serves the resulting sample to an approximator that operates on top of Spark Structure Streaming (hereafter SpSS for short) , taking full advantage of the incrementalizer and optimizers of the underlying system in generating incremental query results(e.g., every time window) . Fair sample selection is achieved by simply applying SRS within each polygon independently with equal inclusion probabilities (thus resembling SRS without replacement within each polygon). Overall, ex-SAOS resorts to a stratified sampling design. The workflow of ex-SAOS is codified in algorithm 1. It is similar to the baseline (SAOS)

[1] http://geohash.org/
[2] https://s2geometry.io/
[3] https://github.com/uber/h3
[4] https://github.com/harsha2010/magellan

except for the fact that we sample proportional counts of tuples from each polygon (city neighbourhoods, boroughs, districts, etc.,) independently. As a way of contrast, SAOS samples on a granular level (geohash level). Enabling a coarser level through ex-SAOS is very efficient for smart city dynamic application scenarios.

To take a utilitarian perspective, ex-SAOS works as the following heuristic overview. Imagining the earth flattened out, ex-SAOS overlays the survey area with an arbitrarily-sized grid. The grid is constructed based on the polygons file that is served as an input. The method then continues by selecting arbitrarily a spatially- fair count of tuples from each polygon (retrieving the sampling fraction for each polygon from map served externally by the user). Having done that, our method resorts to stratified sampling, which is more plausible comparing with other sampling designs because it is known to yield better geo-statistical estimates of target variables in spatial patchy environments. The main essence of the method is the reliance on dimensionality reduction where we reduce parametrized two-dimensional space representations into one-dimensional counterparts while at the same time preserving spatial shape and locality.

C. Geospatial Queries Supported

We support the same set of geospatial queries that we have supported in our previous work [3], but this time with a coarser level for the stateful aggregation queries. We support linear queries that estimate summary statistics for target variables depending on the sample instead of the population. For example, an 'average' of a target variable. Since our ex-SAOS sampling design resorts to a stratified-like sampling, the theory of stratified sampling applies [10]. Suppose we have a total of K polygons (each polygon is a stratum), y_{kj} indicates a value of the j_{th} tuple in polygon k. t (that pronounced tau) is then the population total of stratum k (polygon in this case is a stratum). Then population total of a target variable y is estimated using $\hat{t}_{\text{exSAOS}} = \sum_{k=1}^{K} t_k = \sum_{k=1}^{K} N_k \bar{y}_k$. Thereafter the average is estimated using $\bar{Y}_{ex\text{-}SAOS} = \hat{t}_{ex\text{-}SAOS}/N = \sum_{i=1}^{I} (N_i/N)\bar{y}_i$. Because SpSS does not provide over-the-counter plugins for those estimators (SpSS is not spatial-aware), we have transparently incorporated a patch for achieving that. The other type of spatial queries we support is the stateful aggregations (specifically Top-N queries). For example, "top-3 boroughs in Bologna city in Italy where people tend to check out shared bikes". Online aggregations differ from static batch counterpart in that the former requires managing state between batch intervals, thus achieving a consistency.

To quantify the uncertainty for linear queries, we calculate a relative error depending on $RE = z_{\alpha/2}(SE(\bar{Y}_{exSAOS})/\bar{Y}_{exSAOS})$, where SE is the standard error, $z_{\alpha/2}$ is the upper $\alpha/2$ point of the normal distribution. Readers are referred to our previous work on explanation of the derivation of the equation [3]. For the same group of queries, we also calculate the accuracy loss using the following formula: $loss = |estimatedAverage - trueAverage| \,/\, trueAverage$. For stateful aggregations, we apply Spearman's rank correlation coefficient [20] (Spearman's rho hereafter for short), which measures statistical dependency between rankings of two variables. Specifically, we apply $\rho_{rg} = \frac{cov(rank_{nos}, rank_{samp})}{\sigma_{rank_{nos}} \sigma_{rank_{samp}}}$.

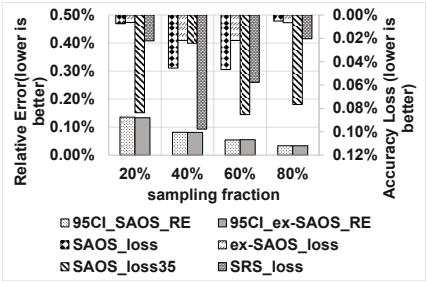

Fig. 2 Estimation accuracy of ex-SAOS vs. SAOS and an SpSS-based SRS, for linear queries. Primary access on the left shows the relative error (RE), whereas secondary access on the right shows the accuracy loss. CI is the confidence interval. 35 is the gohash precision.

V. PERFORMANCE EVALUATION AND RESULTS

A. Deployment Settings and Benchmarking

Dataset. We use the NY City taxicab trips datasets [5] benchmark. We select a cohort of six months dataset (circa nine million tuples) representing data taxi rides for the first six months of 2016. We select the green taxi trip records, which include fields such as GPS locations and itinerary distances.

Deployment and experimental settings. We run our tests over Microsoft Azure HDInsight Cluster hosting Apache Spark version 2.2.1. It consists of 6 NODES (2 Head + 4 Worker) with 24 cores. Head (2 x D12 v2) nodes , and Worker (4 x D13 v2) nodes. Each head node operates on 4 cores with 28 GB RAM and 200 GB Local SSD memory, and quantities are double those figures for worker nodes.

B. Results and discussion

We have tested the extended SAOS method (ex-SAOS) incorporated within SpatialSPE against the previous plain version (SAOS). The only parameter we vary in all the tests is the sampling fraction. We apply the following linear query: "find the average trip distance of a NYC taxicab itinerary trip during the first six months of the year 2016" as a linear query. As a Top-N, we test using the following query: "what are the top-10 neighbourhoods in NY city in USA with highest taxi orders". Figure 2 shows a comparison for the linear summary statistics queries. It is evident that for parsimonious streaming settings which necessitate small sampling fractions (such as 20% as shown in the figure), ex-SAOS outperforms SAOS. However, this benefit vanishes as we increase the bound of the sampling fraction. This is totally healthy, as SAOS applies proportional sampling fractions (equal for all groups or geohashes in this case) to each geohash. Geohashes in the end covers the polygons. As the sampling fractions increases, sampling on a granular level (geohash by SAOS) starts to perform similarly to that of a coarser level (polygon-level by ex-SAOS). The advantage by ex-SAOS is that it allows sampling on arbitrarily-sized and shaped polygons as opposed to SAOS. This is very beneficial in smart city scenarios and complex applications such as those required by environmentalists and in agrobiodiversity. It may be required for example to sample different fractions from each polygon.

[5] https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

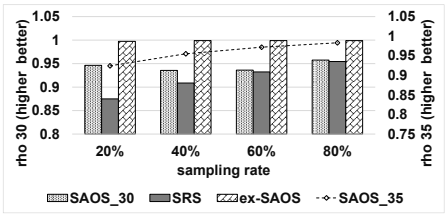

Fig. 3 spearman's rho geohash 30 and 35: ex-SAOS Vs. SAOS & SpSS-based SRS

This can easily be achieved by ex-SAOS. On the contrary, it requires additional work from the programmer side to be able to apply it to SAOS. It is also clear from figure 2 that the accuracy loss induced by the approximation in ex-SAOS is the lowest compared to the other competing counterparts (SAOS with various geohashes, 30 and 35 in this case, in addition to the SRS-based counterpart). This complies with the results obtained for the relative error. On average, we obtain a reduction in the loss that reaches 23% when using ex-SAOS instead of SAOS (with geohash 30) and roughly 67% using ex-SAOS instead of SAOS (with geohash 35), in addition to around 70% using ex-SAOS instead of SRS-based sampling. Figure 3 shows that ex-SAOS outperforms SAOS (with all geohash settings, being 30 or 35) and the SpSS-SRS random sampling in term of the accuracy for the stateful aggregation queries (Top-N ranking geo-statistics). On average, we obtain 4% and 5.8% gains in Top-N accuracy (rho) by using ex-SAOS for top ranking instead of SAOS, with geohashes 35 and 30, respectively. We even perform better when comparing ex-SAOS to the SRS-based solution as we obtain 9% rho accuracy gain when using ex-SAOS instead of SRS-based sampling, which is better than the roughly 3% that we may obtain by applying the plain SAOS against SRS-based design.

It worth mentioning that we have data that is highly skewed (which is the worst-case scenario). This means that our principles are applicable to less skewed data. Despite being highly skewed. The ‘average’ estimator has an approximately normal (bell-shaped curve) distribution sampling distribution. That is the reason that allowed us to depend on the Central Limit Theorem (CLT) [10] , where principles from traditional statistical sampling applies, specifically stratified and simple probability sampling theories. The same fact applies to any kind of less skewed data (having, by itself, a normal distribution or eventually resorts on its ‘average’ estimator to a normal distribution). This proves that similar trends in the results are assured for any kind of spatial datasets.

VI. Summary and Future Research Frontiers

The idea that spatially-balanced sampled datasets yield better estimations than simple probability sampling methods is well established in the relevant literature. In accordance with that, there are some frameworks for incorporating spatial awareness into statistical sampling. Some methods are based on splitting the study area into cells (traditionally known as tessellation, which implies dividing the study area into polygons, either equally- or arbitrarily-sized) and treating each cell as a stratum, thus simplifying the application of stratified-alike sampling designs, which is plausible in geo-statistics. However, those methods are not ready for distributed computing settings. Furthermore, they incorporate computationally expensive structures, such as tree-based hierarchal representation structures that renders them, despite being efficient theoretically, unsuitable for extension to the distributed computing world. On the other side, distributed big data processing systems are evolving fast in an unprecedented way, reflecting the need for systems that adapt to the fluctuating and oscillating pace of big datasets that show temporal skewness.

In this paper, we have extended our support for our robust SpatialSPE spatial stream processing engine. We have added a new spatial online sampling method that operates on coarser-granularity by allowing the strata to have an arbitrary shape and size. Stated another way, sampling from polygons instead of regularly and equally-sized geohashes. The novel method ex-SAOS complements efficiently the SpatialSPE design, specifically for more complicated smart city and urban planning scenarios, which require flexibility, elasticity, representativeness and QoS guarantees in the spatial sampling designs.

Acknowledgment

This research was supported by the IDEHA project funded by PON “RICERCA E INNOVAZIONE” 2014-2020 (no. J46C18000440008).

[1] M. Zaharia, T. Das, H. Li, T. Hunter, S. Shenker and I. Stoica, "Discretized streams: Fault-tolerant streaming computation at scale," in Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles, 2013, pp. 423-438.

[2] I. M. Aljawarneh, P. Bellavista, C. R. De Rolt and L. Foschini, "Dynamic identification of participatory mobile health communities," in Cloud Infrastructures, Services, and IoT Systems for Smart CitiesAnonymous Springer, 2017, pp. 208-217.

[3] K. Li and G. Li, "Approximate query processing: what is new and where to go?" *Data Science and Engineering,* vol. 3, *(4),* pp. 379-397, 2018.

[4] I. M. Al Jawarneh, P. Bellavista, L. Foschini and R. Montanari, "Spatial-aware approximate big data stream processing," in *2019 IEEE Global Communications Conference (GLOBECOM),* 2019, pp. 1-6.

[5] M. Armbrust, T. Das, J. Torres, B. Yavuz, S. Zhu, R. Xin, A. Ghodsi, I. Stoica and M. Zaharia, "Structured streaming: A declarative API for real-time applications in apache spark," in *Proceedings of the 2018 International Conference on Management of Data,* 2018, pp. 601-613.

[6] A. J. Lister and C. T. Scott, "Use of space-filling curves to select sample locations in natural resource monitoring studies," *Environ. Monit. Assess.,* vol. 149, *(1-4),* pp. 71-80, 2009.

[7] D. L. Stevens Jr and A. R. Olsen, "Spatially balanced sampling of natural resources," *Journal of the American Statistical Association,* vol. 99, *(465),* pp. 262-278, 2004.

[8] A. Grafström, N. L. Lundström and L. Schelin, "Spatially balanced sampling through the pivotal method," *Biometrics,* vol. 68, *(2),* pp. 514-520, 2012.

[9] A. R. Olsen, "Generalized Random Tessellation Stratified (GRTS) Spatially-balanced Survey Designs for Aquatic Resources." *US Environmental Protection Agency, National Health and Environmental Effects Research Laboratory,* 2005.

[10] S. L. Lohr, *Sampling: Design and Analysis.* Nelson Education, 2009.

[11] S. K. Thompson, *Sampling.* Wiley, 2012.

[12] J. Wang, R. Haining and Z. Cao, "Sample surveying to estimate the mean of a heterogeneous surface: reducing the error variance through zoning," *Int. J. Geogr. Inf. Sci.,* vol. 24, *(4),* pp. 523-543, 2010.

[13] J. Wang, G. Christakos and M. Hu, "Modeling spatial means of surfaces with stratified nonhomogeneity," *IEEE Trans. Geosci. Remote Sens.,* vol. 47, *(12),* pp. 4167-4174, 2009.

[14] P. Lorkowski and T. Brinkhoff, "Towards real-time processing of massive spatio-temporally distributed sensor data: A sequential strategy based on kriging," in *Agile 2015*Anonymous Springer, 2015, pp. 145-163.

[15] J. Wang, A. Stein, B. Gao and Y. Ge, "A review of spatial sampling," *Spatial Statistics,* vol. 2, pp. 1-14, 2012.

[16] I. M. Al Jawarneh, P. Bellavista, A. Corradi, L. Foschini, R. Montanari and A. Zanotti, "In-memory spatial-aware framework for processing

proximity-alike queries in big spatial data," in *2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD),* 2018, pp. 1-6.

[17] I. M. Aljawarneh, P. Bellavista, A. Corradi, R. Montanari, L. Foschini and A. Zanotti, "Efficient spark-based framework for big geospatial data query processing and analysis," in *2017 IEEE Symposium on Computers and Communications (ISCC),* 2017, pp. 851-856.

[18] I. M. Al Jawarneh, P. Bellavista, F. Casimiro, A. Corradi and L. Foschini, "Cost-effective strategies for provisioning NoSQL storage services in support for industry 4.0," in *2018 IEEE Symposium on Computers and Communications (ISCC),* 2018, pp. 1227.

[19] I. M. Al Jawarneh, "Quality of Service Aware Data Stream Processing for Highly Dynamic and Scalable Applications," 2020. Alma, University of Bologna.

[20] A. Lehman, N. O'Rourke, L. Hatcher and E. Stepanski, *JMP for Basic Univariate and Multivariate Statistics: Methods for Researchers and Social Scientists.* Sas Institute, 2013.