DeepSampling: Selectivity Estimation with Predicted Error and Response Time

Tin Vu
Computer Science and Engineering
University of California, Riverside
tin.vu@email.ucr.edu

Ahmed Eldawy
Computer Science and Engineering
University of California, Riverside
eldawy@ucr.edu

ABSTRACT

The rapid growth of spatial data urges the research community to find efficient processing techniques for interactive queries on large volumes of data. Approximate Query Processing (AQP) is the most prominent technique that can provide real-time answer for ad-hoc queries based on a random sample. Unfortunately, existing AQP methods provide an answer without providing any accuracy metrics due to the complex relationship between the sample size, the query parameters, the data distribution, and the result accuracy. This paper proposes DeepSampling, a deep-learning-based model that predicts the accuracy of a sample-based AQP algorithm, specially selectivity estimation, given the sample size, the input distribution, and query parameters. The model can also be reversed to measure the sample size that would produce a desired accuracy. DeepSampling is the first system that provides a reliable tool for existing spatial databases to control the accuracy of AQP.

CCS CONCEPTS

• **Information systems** → *Data management systems*; • **Computing methodologies** → **Machine learning approaches**.

KEYWORDS

deep learning, spatial sampling, spatial computing

ACM Reference Format:
Tin Vu and Ahmed Eldawy. 2020. DeepSampling: Selectivity Estimation with Predicted Error and Response Time. In *DeepSpatial 2020: ACM SIGKDD Workshop on Deep Learning for Spatiotemporal Data, Applications, and Systems, August 24, 2020, San Diego, CA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Recently, there has been a notable increase in the amounts of spatial data collected by satellites, social networks, and autonomous vehicles. The main method that data scientists use to process this data is through *interactive exploratory queries*; i.e., an ad-hoc query that should be answered in a fraction of a second. Existing studies show that a response time of more than a few seconds to these queries would negatively impact the productivity of the users [15].

DeepSpatial 2020, August 24, 2020, San Diego, CA

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

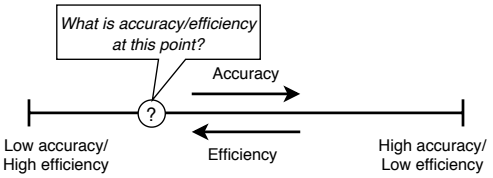
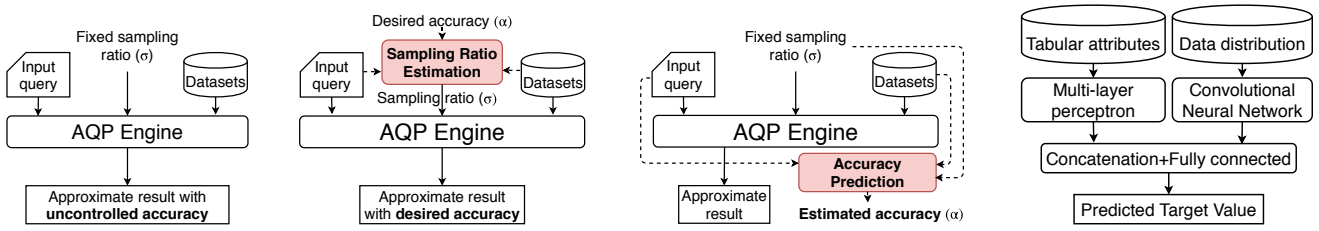


Figure 1: Trade-off between accuracy and efficiency in AQP

Unfortunately, existing big-spatial data systems [4, 9, 24, 29, 30], require way more than that to run even the simplest queries, hence, they cannot answer interactive exploratory queries.

The most viable solution to the interactive exploration problem is approximate query processing (AQP) which uses a small data synopsis, e.g., a sample, to provide an approximate answer within a fraction of a second. This technique provides up-to three orders of magnitude speedup with a very high accuracy for several fundamental problems, including selectivity estimation, clustering, and spatial partitioning [23]. Figure 1 depicts the trade-off between the *accuracy* of the approximate answer and the *efficiency*, i.e., running time, which is highly correlated with the sample size. Unfortunately, this accuracy/efficiency trade-off is very hard to calculate which discourages many users from using AQP systems. Existing solutions either provide answers without any performance guarantee or make unrealistic assumptions such as uniform distribution or independence between dimensions [1, 3, 6, 12–14, 17, 18, 21–23]. This problem is particularly challenging due to the intertwined relationship between the sample size, query parameters, algorithm logic, data distribution, and result accuracy.

This paper proposes DeepSampling, a novel deep learning based model to predict the relationship between accuracy and relative sample size for AQP. The main challenge is how to build a model that works well for any spatial data distribution and query parameters. To solve this problem, we build a deep neural network that takes as input the query parameters and a histogram that represents the data distribution. This idea can work in two modes: *1) given a sample size, it estimates the expected accuracy, or 2) given a desired accuracy, it calculates the required sample size.* The idea is generic and can work with any approximate algorithm by building a separate model for each one. DeepSampling can be integrated into any existing spatial data system that supports AQP. To the best of our knowledge, DeepSampling is the first system that supports predictable error AQP for spatial data analysis problems. We run an experimental evaluation on both synthetic and real data on the selectivity estimation problem and the results show that the proposed method can accurately model the delicate relationship between accuracy and sample size and is portable to many distributions.



(a) Existing AQP engine (b) Sample Ratio Estimation (c) Accuracy Prediction (d) DeepSampling architecture

Figure 2: DeepSampling addresses critical problems on existing AQP systems

In summary, this paper makes following contributions. (1) Design a deep neural network model for predictable error and response time for approximate query processing in spatial data analysis. (2) Apply this model to the **selectivity estimation** algorithm to solve two problems, sample size estimation and accuracy prediction. (3) Validate the model through experiments and publish the pre-trained model for wide use.

2 RELATED WORK

Approximate query processing: AQP is a common method in many spatial data management systems. In AQP, the answer is estimated by executing the query on a small sample of the dataset, instead of scanning entire dataset. AQP is applied on several problems such as selectivity estimation, clustering, and spatial partitioning [23]. For example, SpatialHadoop [9], ScalaGiST [16], Simba [29], SATO [25] use a sample of the input dataset to compute the minimum bounding rectangles (MBRs) for their spatial partitioning operation. Sampling is also used to cluster very large datasets [5, 31]. Specially, sampling is the fundamental method for many selectivity estimation algorithms for spatial data [2]. The main idea of AQP is the trade-offs between query response time and accuracy as shown in Figure 1. The common drawback of existing systems is the lack of a mechanism to choose a suitable sampling ratio to achieve a desired accuracy. For instance, SpatialHadoop just chooses a fixed 1% sample of dataset to compute partition MBRs, which is not always the best choice. DeepSampling addresses this challenge by suggesting the minimum sampling ratio such that the desired accuracy could be achieved. For non-spatial data, BlinkDB [3] provides a bounded errors for standard relational queries. However, BlinkDB assumes the independence of data dimensions, which is not applicable for spatial data.

Deep learning and spatial data: In recent years, the research community has witnessed the rapid growth of research projects in the intersection of big spatial data and machine learning [19]. One of the important research directions is scalable statistical inference systems for big spatial data analysis. For instance, TurboReg [20] is a scalable framework for building spatial logistic regression models. TurboReg is built on top of Markov Logic Network, which is able to predict the presence and absence of spatial phenomena in a geographical area with reasonable accuracy. DeepSPACE [26] is a deep learning-based approximate geospatial query processing engine. DeepSPACE utilize the learned data distribution to provide a quick response for spatial queries with reasonable accuracy. Both TurboReg and DeepSPACE hold the common drawback that they cannot guarantee a required precision of their answers. DeepSampling aims to overcome this issue by providing a prediction model such that the required precision is always met with a reasonable of sampling ratio budget.

3 SELECTIVITY ESTIMATION WITH PREDICTED ERROR AND RESPONSE TIME

3.1 Problem definition

This paper focuses on the prediction model for the selectivity estimation problem but the proposed approach can be easily generalized to other problems such as K-means clustering or spatial partitioning. The goal is to find the relationship between accuracy and sample size and toward this goal we define two problems, *accuracy prediction* and *sample size estimation* which are both defined in this section. First, we will define the accuracy of an approximate answer in the selectivity estimation (SE) problem.

Definition 3.1 (query accuracy). In the SE problem, given an approximate answer π and a ground truth Π for query range Q, the accuracy of the approximate answer π is

$$acc(\pi, \Pi) = max(0, 1 - |\Pi - \pi|/\Pi) \quad (1)$$

Based on this definition, we define the following two problems:

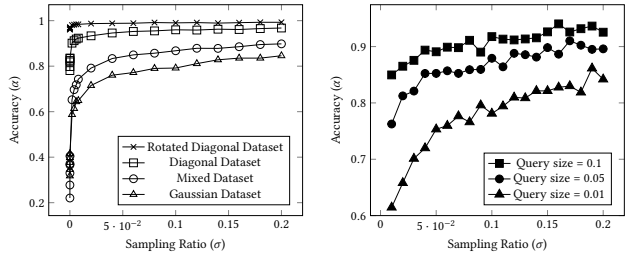
Problem 1 (Sampling Ratio Estimation): *Given a dataset D, a query range Q, and a desired accuracy α, predict the minimum value of sampling ratio σ such that $acc(\pi, \Pi) \geq \alpha$.*

Problem 2 (Accuracy Prediction): *Given a dataset D, a query range Q, and a sampling ratio σ, predict the accuracy α such that $|acc(\pi, \Pi) - \alpha|$ is minimized.*

Both problems are very important in approximate geospatial query processing. If we could address these problems, the existing spatial database systems could minimize the computation effort for sampling process while still achieving a desired accuracy for their answers. Figure 2 shows how DeepSampling enhances performance of existing approximate query processing systems. Instead of fixing a sampling ratio as Figure 2(a), an AQP engine can use Problem 1 to calculate a suggested minimal sampling size to achieve the used-desired accuracy as shown in Figure 2(b). Conversely, if the system has a fixed sampling ratio, it can apply Problem 2 to estimate the result accuracy as shown in Figure 2(c).

3.2 Prediction with mixed data sources

In general, we know that the accuracy (α) of an approximate answer increases with the sampling ratio (σ). However, we show in this part



(a) Different distributions (b) Different query sizes

Figure 3: How sampling ratio (σ) relates to accuracy (α)

that this relationship is more complex than that. Figure 3 shows examples of how these two quantities are related to each other. First, Figure 3(a) shows that this relationship highly depends on the dataset distribution. While for all distributions the sampling ratio and accuracy are highly correlated, the relationship is different for each dataset. For example, for the rotated diagonal dataset, the accuracy ranges from 96% to 99% for all sampling ratios while for the mixed distribution dataset, the accuracy ranges from 22% to 90%. Second, Figure 3(b) shows the relationship for different query sizes. This time, we see that the relationship highly depends on the query size as well.

These observations show how challenging the problem is. To build an accurate model, we need to take into account the input data distribution and the query size. For other problems, the query size could be replaced with other query parameters, e.g., the number of clusters for the K-means clustering problem, or the number of partitions for the spatial partitioning problem.

3.3 DeepSampling architecture

Figure 2(d) shows an overview of the proposed architecture of the DeepSampling model. This architecture is used to solve both problems described earlier, *sampling ratio estimation* and *accuracy prediction*. To avoid repetition, we write between (parentheses) the changes that need to be made for the *accuracy prediction* problem.

To build an accurate and portable model that accounts for the query size and the data distribution, the proposed model takes two sets of inputs, *tabular data* and *data distribution*.

The **tabular input layer** consists of data taken from the processing logs which includes the query size (q), the sampling ratio (σ), and the resulting accuracy (α). If we need to apply this architecture for other problems, then the query size will be replaced with other query parameters, e.g., number of clusters. Also, the accuracy will be calculated differently. This data is passed to a multi-layer perceptron (MLP) model. MLP is a feedforward neural network with at least three layers of nodes: an input layer, a hidden layer and an output layer. We chose MLP for tabular input since it can be used to learn complex mathematical models by regression analysis [8].

The **data distribution input layer** catches the distribution of the input dataset. In this paper, we use a uniform histogram which is expected to accurately catch the dataset distribution if computed at a reasonable resolution. The histogram resolution is a system parameter that we study in the experiments section. Since this histogram is a 2D matrix with spatial relationship between histogram bins, it is fed to a convolutional neural network (CNN) layer.

The **concatenation layer** combines the output of the MLP and CNN layers together and feed them to a fully connected (FC) layer. The final layer of FC is a single node with linear activation so that the model output is the predicted sampling ratio (or accuracy). The **loss function** of the final node provides a feedback on how accurate the predicted value is. Based on the problem definition in Section 3.1, we use mean absolute percentage error (MAPE) as the loss function which is the average absolute percentage error of actual value A_t and forecast value F_t for all training points $t \in [1, n]$ as shown in Equation 2. MAPE is commonly used in regression models since it is very intuitive interpretation for relative errors.

$$MAPE = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right| \quad (2)$$

Table 1: Parameters for the selectivity estimation (SE) query

Parameter	Values (Default)
Dataset distribution	Uniform, Gaussian, Diagonal, Sierpinski, Bit, Parcel, Mixed
Sampling ratio (σ)	0.001, 0.0015,...,0.2
Query size (q)	0.01,0.02,...,0.1.
Histogram size (h)	$1 \times 1 \ldots (16 \times 16) \ldots 64 \times 64$

4 PRELIMINARY RESULTS

This section gives some preliminary results when applying the proposed approach to the selectivity estimation problem. In particular, we wanted to answer the following questions:

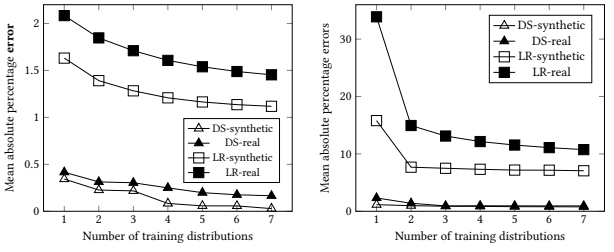
(1) How accurately does the model account for the data distribution and query size?
(2) Can the model solve both problems efficiently?
(3) Is the model portable enough so that we can test it on a new data distribution that was not in the training set?

4.1 Experimental setup

We implement the proposed model in Figure 2(d) using Keras [7]. The source code, training data and models are available at [27].

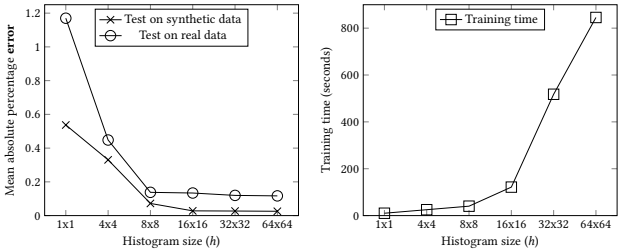
Datasets: We use both synthetic and real datasets in our experiments. We generated a total of 144 synthetic datasets using the open-source spatial data generator [28]. The dataset distributions are listed in Table 1 and the detailed distribution parameters are included in the source code [27]. We also used two real datasets: `OSM-Nodes` [10] and `OSM-Lakes` [11]. The real datasets are only used for testing but never for training the model.

Parameters: In addition to the dataset distribution, we also vary the sampling ratio (σ), the query size (q), and the histogram size (h). The query size is the ratio between the area of the query rectangle and the area of the input minimum bounding rectangle (MBR). Our query workload consists of square queries centered at random locations in the input space. Table 1 summarizes all the parameters that we vary in our experiments. In total, our generated dataset contains 54, 720 data points.



(a) Accuracy Prediction problem (b) Sampling Ratio Estimation problem

Figure 4: Accuracy of DeepSampling and linear regression



(a) Effect on accuracy (b) Effect on training time

Figure 5: Effect of histogram resolution

Metrics: We use mean absolute percentage error (MAPE) to evaluate the accuracy of a prediction model. The lower the value of MAPE, the better the model is.

Baseline method: We compare the proposed model to a linear regression(LR) model which takes the tabular input and predict a numeric output. The reason behind this choice is that we want to see how the dataset distribution input makes a difference to the baseline which only takes query attributes into account.

4.2 Accuracy Prediction

In the first experiment, we build a model to predict the average query accuracy, given the sampling ratio, query size and dataset histogram of size 16×16. In particular, we use the synthetic datasets with 54, 720 data points described in Section 4.1 to train and test our proposed model. To observe how training data distribution affects the test accuracy, we organized the training data into different combinations of 1 to 7 distributions in Table 1. For each combination, we take a split of 75% data points for training process. We test all the trained models with 25% of the synthetic data points. We also test on 2800 data points that we collected from SE queries on real `OSM-Nodes` and `OSM-Lakes` dataset.

Figure 4(a) shows an interesting observation that the more data distributions we used for training process, the more accurate it is. This is expected since some simple distributions might not be able to capture important insights of test datasets. DeepSampling model is doing very well when we tested on both synthetic data and real data (MAPE is around 3% and 16%). This shows the portability of the model. Even though the model was trained only on synthetic data, it still provided good results for the real dataset. In the future, we plan to add more synthetic data to make the model even more accurate with real data. On the other hand, the linear regression baseline, due to its simplicity and the lack of data distribution, did not achieve a good accuracy. For the test on real dataset, its prediction is even more than 100% beyond the actual mean accuracy value.

4.3 Sampling Ratio Estimation

In this experiment, we build a model based on DeepSampling to predict sampling ratio, given a desired query accuracy, query size and dataset histogram of size 16×16. We use the same set of training and testing split as mentioned in Section 4.2.

Table 4(b) shows that DeepSampling is still doing better than the baseline when applied on both synthetic and real data. The errors are relatively higher than the accuracy prediction problem in Section 4.2. The reason is that the range of the accuracy in the training set is narrow as compared to the range of sampling ratio. For example, in Figure 3, the accuracy in some cases stays above 95% while the sampling ratio ranges from 0.1% to 20%. Nonetheless, the DeepSampling approach is consistently more accurate than the linear regression baseline. These results are consistent with existing work that found that the sampling ratio estimation problem is more difficult. For example, in BlinkDB [3] this problem is solved by simply choosing from a predefined set of points, sampling ratio and accuracy, and interpolating between them if needed.

4.4 Effect of histogram resolution

To choose a good histogram size, this experiment studies the trade-off between the model accuracy and training time as we vary the histogram size as depicted in Figure 5. In this experiment, we vary the histogram resolution from 1×1 (effectively no histogram) to 64×64. Figure 5(a) shows the accuracy of the model when tested on both synthetic and real data as the histogram size increases. It is clear from this experiment that the histograms with higher resolutions carry more information that makes the model more accurate. However, the model stabilizes at 16×16 where the histogram is accurate enough to catch the distributions in the training set.

Figure 5(b) shows the total time of the training phase, i.e., the time until the model stabilizes. As expected, the model takes more time to train as the histogram resolution increases due to the large input that goes through the CNN model. From this experiment, we choose to set the histogram size to 16×16 which gives a good accuracy in a reasonable time.

5 SUMMARY AND FUTURE WORK

In this paper, we introduced DeepSampling, a deep-learning-based system that provides predicted errors for approximate geospatial query processing. The proposed model combines the sampling ratio, the result accuracy, the query parameters, and the input data distribution. We carry some preliminary results when we apply DeepSampling to improve performance of selectivity estimation query. The results show that the proposed model can accurately compute the sampling ratio and accuracy for many synthetic and real distributions. In the future, we will apply the same model on other important approximate spatial problems such as K-means clustering and spatial partitioning.

REFERENCES

[1] Ashraf Aboulnaga and Jeffrey F. Naughton. 2000. Accurate Estimation of the Cost of Spatial Selections. In *ICDE*. IEEE, San Diego, CA, 123–134.

[2] Swarup Acharya, Viswanath Poosala, and Sridhar Ramaswamy. 1999. Selectivity estimation in spatial databases. In *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*. ACM, "", 13–24.

[3] Sameer Agarwal, Barzan Mozafari, Aurojit Panda, Henry Milner, Samuel Madden, and Ion Stoica. 2013. BlinkDB: queries with bounded errors and bounded response times on very large data. In *Proceedings of the 8th ACM European Conference on Computer Systems*. ACM, "", 29–42.

[4] Furqan Baig et al. 2017. SparkGIS: Resource Aware Efficient In-Memory Spatial Query Processing. In *SIGSPATIAL*. ACM, Redondo Beach, CA, 28:1–28:10.

[5] Jeremy Bejarano, Koushiki Bose, Tyler Brannan, Anita Thomas, Kofi Adragni, Nagaraj K Neerchal, and George Ostrouchov. 2011. Sampling within k-means algorithm to cluster large datasets. *UMBC Student Collection* 1, 1 (2011), 1–1.

[6] Harry Chasparis and Ahmed Eldawy. 2017. Experimental evaluation of selectivity estimation on big spatial data. In *Proceedings of the Fourth International ACM Workshop on Managing and Mining Enriched Geo-Spatial Data, Chicago, IL, USA, May 14, 2017*. Acm, "Chicago, IL, USA", 8:1–8:6.

[7] François Chollet et al. 2015. Keras. https://keras.io.

[8] George Cybenko. 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems* 2, 4 (1989), 303–314.

[9] Ahmed Eldawy et al. 2015. Spatial partitioning techniques in SpatialHadoop. *Proceedings of the VLDB Endowment* 8, 12 (2015), 1602–1605.

[10] Ahmed Eldawy and Mohamed F. Mokbel. 2019. All points on the map as extracted from OpenStreetMap. https://doi.org/10.6086/N100004J#mbr=9qh2s0vt,9qhf0614 Retrieved from UCR-STAR https://star.cs.ucr.edu/?OSM2015/all_nodes&d.

[11] Ahmed Eldawy and Mohamed F. Mokbel. 2019. All water areas in the world from OpenStreetMap. https://doi.org/10.6086/N1668B70 Retrieved from UCR-STAR https://star.cs.ucr.edu/?OSM2015/lakes&d.

[12] Yannis E. Ioannidis. 1993. Universality of Serial Histograms. In *VLDB*. VLDB, Dublin, Ireland, 256–267.

[13] Ji Jin, Ning An, and Anand Sivasubramaniam. 2000. Analyzing Range Queries on Spatial Data. In *ICDE*. IEEE, San Diego, CA, 525–534.

[14] Richard J. Lipton, Jeffrey F. Naughton, and Donovan A. Schneider. 1990. Practical Selectivity Estimation through Adaptive Sampling. In *SIGMOD*. ACM, Atlantic City, NJ, 1–11.

[15] Zhicheng Liu and Jeffrey Heer. 2014. The Effects of Interactive Latency on Exploratory Visual Analysis. *Proceedings of the IEEE Transactions on Visualization and Computer Graphics, TVCG* 20, 12 (2014), 2122–2131.

[16] Peng Lu et al. 2014. ScalaGiST: Scalable Generalized Search Trees for MapReduce Systems. *PVLDB* 7, 14 (2014), 1797–1808.

[17] Frank Olken and Doron Rotem. 1993. Sampling from Spatial Databases. In *ICDE*. IEEE, Vienna, Austria, 199–208.

[18] Viswanath Poosala, Yannis E. Ioannidis, Peter J. Haas, and Eugene J. Shekita. 1996. Improved Histograms for Selectivity Estimation of Range Predicates. In *SIGMOD*. ACM, Montreal, Quebec, Canada, 294–305.

[19] Ibrahim Sabek and Mohamed F Mokbel. 2019. Machine learning meets big spatial data. *Proceedings of the VLDB Endowment* 12, 12 (2019), 1982–1985.

[20] Ibrahim Sabek, Mashaal Musleh, and Mohamed F Mokbel. 2018. TurboReg: A framework for scaling up spatial logistic regression models. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, "", 129–138.

[21] AB Siddique and Ahmed Eldawy. 2018. Experimental Evaluation of Sketching Techniques for Big Spatial Data.. In *SoCC*. ACM, "", 522.

[22] AB Siddique, Ahmed Eldawy, and Vagelis Hristidis. 2019. Euler++: An Improved Selectivity Estimation for Rectangular Spatial Records. In *IEEE Big Spatial Data Workshop*. IEEE, "", 1.

[23] Abu Bakar Siddique, Ahmed Eldawy, and Vagelis Hristidis. 2019. Comparing synopsis techniques for approximate spatial data analysis. *Proceedings of the VLDB Endowment* 12, 11 (2019), 1583–1596.

[24] MingJie Tang, Yongyang Yu, Qutaibah M. Malluhi, Mourad Ouzzani, and Walid G. Aref. 2016. LocationSpark: A Distributed In-Memory Data Management System for Big Spatial Data. *PVLDB* 9, 13 (2016), 1565–1568.

[25] Hoang Vo, Ablimit Aji, and Fusheng Wang. 2014. SATO: a spatial data partitioning framework for scalable query processing. In *SIGSPATIAL*. ACM, Dallas/Fort Worth, TX, 545–548.

[26] Dimitri Vorona, Andreas Kipf, Thomas Neumann, and Alfons Kemper. 2019. DeepSPACE: Approximate Geospatial Query Processing with Deep Learning. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, "", 500–503.

[27] Tin Vu and Ahmed Eldawy. 2020. Deep Sampling. https://github.com/tinvukhac/deep-sampling.

[28] Tin Vu, Sara Migliorini, Ahmed Eldawy, and Alberto Belussi. 2019. Spatial Data Generators. In *1st ACM SIGSPATIAL International Workshop on Spatial Gems (SpatialGems 2019)*. ACM, "", 7.

[29] Dong Xie, Feifei Li, Bin Yao, Gefei Li, Liang Zhou, and Minyi Guo. 2016. Simba: Efficient In-Memory Spatial Analytics. In *SIGMOD*. ACM, San Francisco, CA, 1071–1085.

[30] Jia Yu, Jinxuan Wu, and Mohamed Sarwat. 2015. 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*. ACM, ACM, "", 70.

[31] Jian Yu, Miin-Shen Yang, and E Stanley Lee. 2011. Sample-weighted clustering methods. *Computers & mathematics with applications* 62, 5 (2011), 2200–2208.