Knowledge-centric Analytics Queries Allocation in Edge Computing Environments

Stefanos Sagkriotis, Kostas Kolomvatsos, Christos Anagnostopoulos, Dimitrios P. Pezaros, Stathes Hadjiefthymiades*
School of Computing Science, University of Glasgow
e-mail: s.sagkriotis.1@research.gla.ac.uk; {kostas.kolomvatsos, christos.anagnostopoulos, dimitrios.pezaros}@glasgow.ac.u.k

*Dept Informatics & Telecommunications, University of Athens
e-mail: shadj@di.uoa.gr

***Abstract*—The Internet of Things (IoT) involves a huge number of devices that collect data and deliver them to the Cloud. The processing of data at the Cloud is characterized by increased latency in providing responses to analytics queries defined by analysts or applications. Hence, Edge Computing (EC) comes into the scene to provide data processing close to the source. The collected data can be stored in edge devices and queries can be executed there to reduce latency. In this paper, we envision a case where entities located in the Cloud undertake the responsibility of receiving analytics queries and decide on the most appropriate edge nodes for queries execution. The decision is based on statistical signatures of the datasets of nodes and the statistical matching between statistics and analytics queries. Edge nodes regularly update their statistical signatures to support such decision process. Our performance evaluation shows the advantages and the shortcomings of our proposed schema in edge computing environments.**

I. Introduction

Edge Computing (EC) facilitates the efficient management of data collected by numerous nodes. The vast infrastructure of the IoT involves heterogeneous devices capable of interacting with their environment and collecting data. Data could be gathered into a number of nodes at the edge of the network, thus, performing local processing to save time and resources for the provision of analytics. Analytics are derived over the collected data and corresponding to analytics queries issued by analysts, users, or applications. We envisage a set of Edge Nodes (ENs) and streams of analytics queries allocated to such nodes for execution. The ENs are regarded as distributed data repositories where queries can be executed through their corresponding Query Processors (QPs). The efficient management and allocation of incoming analytics queries as well as the provided results characterize the success of the supported applications. Usually, applications demand a response in the minimum time to provide high quality services to end users. Hence, the ENs/QPs should adopt query allocation and execution plans that limit the time required for obtaining the final analytic result.

Challenges: Should we desire to significantly reduce the time required for delivering final results of incoming analytics queries, the selection of the appropriate query execution plans is the first step of the process. Then, one should involve the selection of an efficient queries allocation plan. This means that one should select the most appropriate subset of nodes that will deliver the appropriate results according to queries semantics and in the minimum time. Then, an efficient aggregation mechanism on the *partial* results should be invoked. In this paper, we focus on the allocation process of analytics queries arriving at a Query Controller (QC) in the Cloud taking into consideration the sufficient statistics (*statistical signatures*) of the data present in each EN. We consider that the underlying nodes are logically clustered based e.g., on geospatial criteria, imposed by the analytics applications. We introduce a decision making mechanism that exploits the statistical signatures of nodes' datasets and concludes on most appropriate subset of nodes for allocation and execution. Based on this approach we reduce the time for getting responses and increase their quality as nodes with irrelevant data (compared to queries semantics) are excluded from the query allocation and execution process. The ENs regularly send their statistical signatures to the QC (at pre-defined intervals) involving the elimination of outliers and the reduction of the data dimensionality. This way, ENs send reliable statistics for their dataset that consequently support the QC's decision making.

The paper is organized as follows. Section II reports on related work, Section III formulates the problem and Section IV introduces our scheme. In Section V, we provide experimental evaluation of our mechanism, while Section VI concludes the paper reporting future research agenda.

II. Related Work

In edge computing environments many efforts study opportunities for management of distributed data. The Dragon scheme in [17] identifies nodes that can reply to users' requests based on criteria describing nodes themselves and their data. In [15] the authors propose a distributed data service providing functionality for data collection and processing. The objective is to enable multiple IoT middleware systems to share common data services covering interoperability issues. The parallel execution of queries increases the performance of the applications and at the same time, analytics are retrieved by different nodes aggregating them to deliver the final outcome. It is of high importance to obtain a view on the data statistics on each node since we can estimate the relevance between the data and analytics queries as studied in [18], [19], [20]. Multiple efforts handle the problem of allocating data to specific nodes. In [4] data streams are partitioned on-the-fly taking into consideration the query semantics. A multi-route optimizer is proposed in [5] exploting intra- and inter-stream correlations to produce effective partitions. The schemes in [11] and [26] propose the separation of streams into sets of sub-streams over which queries are executed in parallel.

Query engines for the IoT domain [24] provide results in real time and data are processed at the devices at network edge. Such edge-centric processing is important in real-time, mission-critical applications such as self-driving vehicles [7]. Significant examples of edge-centric processing are: [1], [8], [21], [23], [27], [14]. Some efforts deal with the automated separation of queries graphs into two sets [10]: queries processed at the edge devices and queries processed in the cloud. In [22], the authors adopt statistical learning to recommend a previously generated query plan to the optimizer for a given query. The objective is to predict the query execution time for workload management and capacity planning. The delivery of edge analytics involves communication efficient predictive modeling within the edge network [13]. Analytics are derived by models dealing with dynamic optimal decisions for data deliver in light of communication efficiency [27], [6]. Several schemes exploit the computational capability of edge nodes to launch algorithms directly at the data sources [2], [9], [16].

Contribution: In our past research we dealt with the allocation of queries to a set of nodes; in [18], we proposed a time-optimized scheme for selecting the appropriate nodes adopting the *odds algorithm* while in [19], we presented a reinforcement learning model for query allocation. Our extended work in [20] incorporates statistical learning processes for query load balancing. The proposed scheme in this paper significantly supports our previous schemes by intelligently providing a statistics-based efficient mechanism responsible to deliver the minimum sufficient information of the data for query allocation in edge computing environments. Such mechanism is *exposed* to the Cloud infrastructure being part of the schemes proposed in [18], [19], [20].

III. Definitions & Problem Formulation

We consider a set of N ENs $\mathcal{E} = \{n_1, n_2, \dots, n_N\}$ placed at various locations, e.g., in a city. IoT nodes like smartphones and sensors are connected with ENs to deliver their contextual data. At the upper layer, e.g., in the Cloud, there is a set of Query Controllers (QCs) responsible to receive and execute analytics queries $q_1, q_2, \dots$ defined by analytics applications and/or end users (data analysts). Such queries are then allocated to the appropriate QPs for execution functioning in the available ENs. Consider the set of N QPs $\mathcal{P} = \{p_1, p_2, \dots, p_N\}$, each one corresponding to an EN. A QC after receiving an analytics query, it invokes the most appropriate subset $\mathcal{P}' \subset \mathcal{P}$ of QPs to get their query results and return the final *aggregated* result to the requesting applications depending on how *relevant* is the query to the underlying data of each EN. The determination of the subset $\mathcal{P}'$ is achieved from certain statistics of data that each EN delivers to the back-end infrastructure, i.e., QC. Such statistics support the QC with the necessary view on what data are present in each EN used for the statistical matching with each incoming analytics query. Figure 1 illustrates the considered architecture.

Definition 1: A dataset $D_i = \{\mathbf{x}_j\}_{j=1}^{m_i}$ of the EN i is a set of m_i row data vectors $\mathbf{x} = [x_1, \dots, x_d] \in \mathbb{R}^d$ with real attributes $x_k \in \mathbb{R}$.

Analytics queries are issued over a d-dimensional data space and bear two key characteristics: First, they define a subspace of interest, using various predicates on attribute (dimension) values. Second, they perform aggregate functions over said data subspaces (to derive key statistics over the subspace of interest). We adopt a general vectorial representation for modeling a query over any type of data storage/processing system. Predicates over attributes define a data subspace over a dataset D formed by a sequence of logical conjunctions using (in)equality constraints ($\leq, \geq, =$). A *range-predicate* restricts an attribute x_k to be within range $[l_k, u_k]$: $x_k \geq l_k \wedge x_k \leq u_k$, $k = 1, \dots, d$. We model a range query over a dataset D through conjunctions of predicates, i.e., $\bigwedge_{k=1}^{d}(l_k \leq x_k \leq u_k)$ represented as a vector in $\mathbb{R}^{2d}$.

Definition 2: A (range) row analytics query vector is defined as $\mathbf{q} = [l_1, u_1, \dots, l_d, u_d] \in \mathbb{R}^{2d}$ corresponding to the range query $\bigwedge_{k=1}^{d}(l_k \leq x_k \leq u_k)$.
For instance, consider an analytics range query asking for extracting the correlation between temperature x_1 and humidity x_2 in the 2-dim subspace $[5, 10] \times [80, 100] \subset \mathbb{R}^2$. If such a query is executed over a dataset where the pairs (temperature, humidity) are outwith the above-mentioned 2-dim subspace, then the corresponding node will waste computational resources for executing this range query. In addition, such results will affect the final response due to aggregation.

Every EN i at pre-defined intervals calculates certain statistics of its dataset D_i forming the statistical signature: $\mathcal{S}_i$. $\mathcal{S}_i$ contains sufficient statistics of the underlying data vectors in D_i. In this paper, we adopt the following of statistics for the signature: the mean row vector $\boldsymbol{\mu} = [\mu_1, \dots, \mu_d]$, the variance row vector $\boldsymbol{\sigma} = [\sigma_1, \dots, \sigma_d]$ and the eigenbase $\mathbf{W}_{d \times K}$ of the first $K \leq d$ column eigenvectors $\mathbf{w}_k \in \mathbb{R}^d$ (Principal Components) produced by the Principal Component Analysis (PCA) [12] over the data in D_i (or a sample); see Section IV-B. The signature of EN i is:

$$\mathcal{S}_i = \{\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i, \mathbf{W}_{i,d \times K}\}. \quad (1)$$

Note that, the signature should be constructed and incrementally updated in limited time, thus, the performance of nodes is not affected. The extraction of the signature $\mathcal{S}_i$ is based on a multidimensional outliers elimination model. The outliers elimination model is based on an aggregation scheme over two known statistical measures of χ^2 and the Grubb's test in order to decide about the presence of outliers in the D_i. The outcomes of the two outliers techniques are combined and then the signature is constructed over *outliers-free* data.

Each dataset D_i is updated over the time as data streams are produced by IoT devices at high rates. In our context, the QC does not have any view on the data present in every dataset since data are not delivered to the Qc; *only* their corresponding statistical signatures are delivered and updated regularly.

The QC receiving a query $\mathbf{q}$ should conclude on a matching degree between the query and the available signatures $\{\mathcal{S}_i\}_{i=1}^N$. Based on this matching, the QC should decide on the most appropriate subset $\mathcal{P}' \subset \mathcal{P}$ of QPs referring to those EDs that will selectively execute the query q. Based on this partial engagement of the QPs, irrelevant ENs are excluded from the query execution thus avoiding providing results that do not match the query predicates. Hence, irrelevant nodes are not involved in the execution of queries whose data are not matched with the queries semantics (represented by predicates). We then formalize our problem:

Problem 1: Given an analytics query $\mathbf{q}$ to the QC and a set of N statistical signatures $\{\mathcal{S}_i\}_{i=1}^N$ derived from N ENs, seek the most appropriate subset of QPs $\mathcal{P}' \subseteq \mathcal{P}$ which will be engaged for executing the query $\mathbf{q}$.

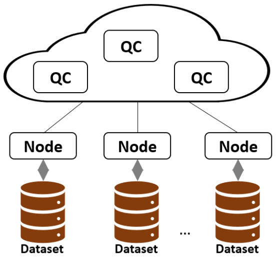

Fig. 1. The considered architecture with QCs and edge nodes.

IV. Scaling-out Queries Assignment

In this section, we elaborate on our methodology for finding the relevant subset $\mathcal{P}'$ of the QPs given a random range analytics query $\mathbf{q}$ in the QC. We first introduce an aggregation mechanism for removing outlies from the ENs' datasets before constructing the statisitcal signatures. Then, we provide the construction of the signatures and elaborating on the methodology of selecting the most relevant ENs for query execution based on the query semantics and the signatures.

A. Aggregation-based Outliers Elimination

For detecting multivariate outliers in a dataset D_i, we can rely on widely adopted techniques, which could be categorized to: (i) statistical-based (parametric or non-parametric approaches), (ii) nearest neighbor-based, (iii) clustering-based, (iv) classification-based (Bayesian network-based and support vector machine-based approaches), and (v) spectral decomposition-based approaches. In this work we focus on the statistical methods that require less computational resources. We introduce an aggregation scheme of the χ^2 metric and the Grubb's test [12] for the final outliers outcome. A data vector $\mathbf{x} \in D_i$ is considered as an outlier if the χ^2-statistic exceeds a specific threshold, which is defined as:

$$\chi^2(\mathbf{x}) = \sum_{k=1}^{d} \frac{(x_k - \mu_k)^2}{\mu_k} \tag{2}$$

where μ_k is the k-th mean value dimension of the mean vector $\boldsymbol{\mu}$ of the dataset D_i. According to the central limit theorem when d is large ($\gg 30$), the χ^2 has approximately a Gaussian distribution [25]. The Grubb's test is also adopted for outlier elimination providing the so-called z-score for data vector $\mathbf{x} \in D_i$ defined as: $z(\mathbf{x}) = \frac{\max(\|\mathbf{x}-\boldsymbol{\mu}\|)}{\max_{k=1,\ldots,d}(\sigma_k)}$. The vector $\mathbf{x} \in D_i$ is considered as an outlier when

$$z(\mathbf{x}) \geq \frac{|D_i| - 1}{\sqrt{|D_i|}} \sqrt{\frac{t^2_{\alpha/(2|D_i|),|D_i|-2}}{|D_i| - 2 + t^2_{\alpha/(2|D_i|),|D_i|-2}}} \tag{3}$$

where $t^2_{\alpha/(2|D_i|),|D_i|-2}$ is retrieved by the t-distribution at the significance level of $\alpha/(2|D_i|)$.

Our outlier aggregation mechanism is based on a conjunction of the outlier result determined by $\chi^2(\mathbf{x})$ and Grubb $z(\mathbf{x})$. Specifically, let the outlier indicator functions be $I_\chi(\mathbf{x}) = 1$ and $I_G(\mathbf{x}) = 1$, respectively denoting that $\mathbf{x}$ is an outlier based on the above-mentioned statistics. Then, the vector $\mathbf{x}$ is considered as an outlier in the dataset D_i if $I_\chi(\mathbf{x}) \wedge I_G(\mathbf{x}) = 1$; otherwise the data vector is not an outlier. This means that we require both methods should agree on the result. If there is a disagreement, $\mathbf{x}$ is not an outlier and is included in the construction of the statistical signature $\mathcal{S}_i$.

B. Statistical Signature

The statistical signature $\mathcal{S}_i$ of EN i is based on the data vectors in D_i, which are not considered outliers based on the above-mentioned methodology, i.e., $\tilde{D}_i = \{\mathbf{x} \in D_i : I_\chi(\mathbf{x}) \wedge I_G(\mathbf{x}) = 0\}$. The basic statistics of the mean vector $\boldsymbol{\mu}$ and the variances vector $\boldsymbol{\sigma}$ are directly determined and efficiently incrementally updated from the outliers-free dataset $\tilde{D}_i$. They are both used for matching with the query predicates, as will be discussed later. Now, for establishing the minimum sufficient statistics that can reflect the basis of the underling data, we use the first K_d principal components of the data vectors in $\tilde{D}_i$ that explain the α percentage of the inherent variance (normally $\alpha = 0.9$). Specifically, we seek the eigenbase of the outlier-free data $\tilde{D}_i$ such that given an random data vector $\mathbf{y} \in \mathbb{R}^d$ we can efficiently determine if this vector can be reconstructed (derived from) from the eigenvectors of those data $\mathbf{x} \in \tilde{D}_i$. This is the rationale behind the concept of the signature where we extract the sufficient synopsis of the data deriving the most representative eigenvectors of $\tilde{D}_i$. If the vector $\mathbf{y}$ can be explained by the eigenbase of $\tilde{D}_i$ then we draw the conclusion that $\mathbf{y}$ belongs (can be projected onto) to the subspace of $\tilde{D}_i$. Otherwise, $\mathbf{y}$ is considered statistically irrelevant to $\tilde{D}_i$. In order to come up with this reasoning, we need first to derive the K PCs of $\tilde{D}_i$ by adopting (incremental) PCA over the dataset $\tilde{D}_i$.

In PCA over the $\tilde{D}_i$, we seek the $d \times K$ matrix $\mathbf{W}_i$ of K column (eigen)vectors $\{\mathbf{w}_k\}_{k=1}^K$ that minimizes the objective:

$$\min_{\mathbf{W}_i \in \mathbb{R}^{d \times K} : \mathbf{W}_i^\top \mathbf{W} = \mathbf{I}} \sum_{\mathbf{x}_j \in \tilde{D}_i} \|\mathbf{x}_j - \mathbf{W}_i \mathbf{W}_i^\top \mathbf{x}_j\|^2, \tag{4}$$

where $\|\mathbf{x}\|$ is the Euclidean norm of the vector $\mathbf{x}$. That is, we try to find those K PCs in the eigenbase $\mathbf{W}_i$ such that when we project a d-dim vector onto the subspace defined by those PCs, the error of the projection vector $\tilde{\mathbf{x}} = \mathbf{W}_i\mathbf{W}_i^\top\mathbf{x}$ and the actual vector $\mathbf{x}$ is minimized. Hence, we argue that if $\mathbf{x}$ belongs to this subspace then the projection error is the minimum w.r.t. the K PCs. We select the first K PCs, which are ordered by their eigenvalues $\lambda_k, k = 1, \ldots, K$, such that the explain $\alpha(\%)$ of the variances in $\tilde{D}_i$. This is achieved by selecting the first K PCs such that: $K = \min\{K' : \frac{\sum_{k=1}^{K'} \lambda_k}{\sum_{k=1}^{d} \lambda_k} \geq \alpha\}$. Given a *projection error tolerance* $\epsilon > 0$ and the eigenbase $\mathbf{W}_i$ reflecting the sufficient eigenvectors from $\tilde{D}_i$ we infer that a random vector $\mathbf{y} \in \mathbb{R}^d$ belongs to the subspace defined by the PCs of $\tilde{D}_i$ iff its projection error $\|\mathbf{y} - \mathbf{W}_i\mathbf{W}_i^\top\mathbf{y}\| \leq \epsilon$; otherwise, the vector $\mathbf{y}$ is considered statistical irrelevant (cannot be explained from) to $\tilde{D}_i$, or in other words it is highly unlikely to be observed in EN i. The statisitcal signature $\mathcal{S}_i = \{\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i, \mathbf{W}_i\}$ is then delivered to the QC. Note, the eigenbase $\mathbf{W}_i$ can be incrementally updated with trivial computational complexity adopting well-known incremental PCA methods. The EN i regularly updates the QC with an updated signature $\mathcal{S}'_i = \{\boldsymbol{\mu}'_i, \boldsymbol{\sigma}'_i, \mathbf{W}'_i\}$ iff there is a significant difference in $\|\boldsymbol{\mu}_i - \boldsymbol{\mu}'_i\|$, $\|\boldsymbol{\sigma}_i - \boldsymbol{\sigma}'_i\|$ and $\|\mathbf{W}_i - \mathbf{W}'_i\|$; otherwise there is no meaning for an update.

C. Signature-based Query Assignment

The QC receives the signatures $\{\mathcal{S}_i\}_{i=1}^N$ from all N ENs in order to reason about the most relevant subset of QPs to engage for each analytics query. This means that, for each query $\mathbf{q}$ a different subset $\mathcal{P}'$ of QPs is determined engaging the corresponding ENs. Given an analytics query $\mathbf{q} = [l_k, u_k]_{k=1}^d \in \mathbb{R}^{2d}$, the QC derives its d-dim center query vector $\mathbf{y} = [\frac{l_k+u_k}{2}]_{k=1}^d = [y_k]_{k=1}^d \in \mathbb{R}^d$, where each k-th component y_k refers to the center of k-th range predicate, i.e., $y_k = \frac{l_k+u_k}{2}$. The center query $\mathbf{y}$ is then projected onto each eigenbase $\mathbf{W}_i$ of EN i in order to judge whether the center predicate semantics are projected over the data subspace define by each $\tilde{D}_i$. If the vector $\mathbf{y}$ is approximately considered to belong to the eigenbase $\mathbf{W}_i$ based on the error tolerance $\|\mathbf{y} - \mathbf{W}_i\mathbf{W}_i^\top\mathbf{y}\| \leq \epsilon$ then the associated QP i is a candidate to be engaged for the execution of this query. Otherwise, the dataset D_i is not relevant for providing analytics results for the query $\mathbf{q}$. In the case where the QP i is candidate for query $\mathbf{q}$, we further examine if the underlying data are statistically sufficient (in number) to support the query. This is examined by using the ratio of data dimensions with $y_k \in [\mu_k - \sigma_k, \mu_k + \sigma_k]$, $k = 1, \ldots, d$. If we let $I(y_k) = 1$ if $|y_k - \mu_k| \leq \sigma_k$; otherwise 0, we can then introduce the *degree of engagement* $I(\mathbf{q}, \mathcal{S}_i)$ of an analytics query to a QP i over the dataset D_i of EN i as:

$$I(\mathbf{q}, \mathcal{S}_i) = \begin{cases} 0 & \text{if } \|\mathbf{y} - \mathbf{W}_i\mathbf{W}_i^\top\mathbf{y}\| > \epsilon, \\ 1 + \frac{1}{d}\sum_{k=1}^d I(y_k) & \text{otherwise} \end{cases} \tag{5}$$

Hence, given a query $\mathbf{q}$ at the QC, we first derive its center query vector $\mathbf{y}$ and then for each QP i, we check its corresponding degree of engagement $I(\mathbf{q}, \mathcal{S}_i)$ based on the signature $\mathcal{S}_i$. Then, the QC determines the subset $\mathcal{P}'(\mathbf{q}) \subseteq \mathcal{P}$ of those QPs with $I(\mathbf{q}, \mathcal{S}_i) > 0$, i.e.,

$$\mathcal{P}'(\mathbf{q}) = \{p_i \in \mathcal{P} : I(\mathbf{q}, \mathcal{S}_i) > 0\}. \tag{6}$$

The QC assigns the analytics query $\mathbf{q}$ to the QPs belonging in $\mathcal{P}'(\mathbf{q})$ to execute that query over their corresponding ENs.

V. Experimental Evaluation

This section describes the experimental setup that has been employed to evaluate the proposed mechanism presented in Section IV. Furthermore, the performance metrics that have been used are presented alongside the comparison baseline, which has been established to examine the performance of our mechanism. The outcome of the performance assessment is discussed at the end of this section.

A. Setup

To evaluate our approach, we simulated a deployment scenario that involves four ENs that gather two-dimensional data over time. To make the experiments reproducible, a publicly available dataset has been chosen to simulate the data repositories of the ENs [28]. The dataset is composed of sensor readings that originate from four Raspberry Pis, each one attached to an Unmanned Surface Vehicle (USV)[1]. The sensors gather temperature and humidity readings over time at the sea surface, hence, the two dimensional data vectors of our scenario. However, different nodes stop gathering data at different times. To compensate for that inconsistency, we chose the time at which the first USV node stops gathering data to indicate the end of the dataset. In this way, we maintain consistency through all nodes for data gathered before that time.

When the simulation commences, no data is available to the (USV) nodes. Instead, data is parsed gradually according to typical node behaviour. Once a predefined amount of data is available in an EN/USV, the method that builds the statistical signature for this last amount of data is triggered and builds the relevant statistical signature $\mathcal{S}_i$ as defined in (1). All USV/nodes transmit their latest statistical signatures to the QC. In turn, the QC receives a four-dimensional analytics query randomly generated according to a uniform distribution which ranges between $[0, 40]$ for humidity and $[0, 60]$ for temperature attributed. The QC, based on the initial query $\mathbf{q}_k$, calculates the two-dimensional center query vector $\mathbf{y}_k$. Decisions on which nodes will accommodate the query are based on the mechanism described in Section IV-C. The error tolerance ϵ defined in (5) is pre-determined to a certain value for each run of the experiment.

B. Performance Metrics

After queries are allocated to QPs, the responses are based on the available contextual data on each node. Prior to using the proposed method, the query would be allocated to *every* QP in the network, i.e., the baseline solution engaging *all*

[1] The contextual data were derived from the experimentation EU-funded project GNFUV: https://sites.google.com/view/gnfuv/home

USVs. By using our proposed query allocation and execution mechanism, the amount of nodes that receive the query is expected to decrease because of the statistical irrelevance of their data. As a result, the data that will formulate the response to the query will be fewer and more *relevant* to the query, which is our desideratum. In order to evaluate this rationale, we calculate the total variance of the data existing in nodes after a query allocation: $V'_{total} = \sum_{i=1}^{N'} \sigma_i$, for $N' = |\mathcal{P}'(\mathbf{q})|$ and compare it against the total variance of data across *all* nodes (prior to applying the proposed mechanism) $V_{total} = \sum_{i=1}^{N} \sigma_i$, for $N = |\mathcal{P}|$, where $|\mathcal{P}|$ is the cardinality of the set $\mathcal{P}$.

After the query allocation phase is complete, the nodes that have been found to hold *relevant data for the issued queries* are marked in order to examine the total node-involvement ratio per query. The *involvement ratio* is defined as the number of nodes that were identified to hold relevant data over the total number of nodes $r = \frac{N'}{N}$, with $0 \leq N' \leq N$. Such involvement ratio provides a high-level metric of the resources that are utilized per query execution. By utilizing a portion of the nodes and thus having a small involvement ratio, the message exchanges are reduced and fewer query processing transactions are required. Maintaining a small involvement ratio r while accommodating the needs of queries translates to more efficient use of the available resources: this is an important aspect when considering scenarios involving resource-constrained edge nodes, like the UxV environments.

C. Performance Assessment

We set a comparison baseline by examining the case of not using the query allocation mechanism. In that case, the queries are allocated to every QP available in the edge network. After conducting experiments for this use case and obtain the aforementioned metrics, we repeat the experiments with the query allocation mechanism enabled. To ensure convergence of the obtained results, we conducted repeated experiments and used the average of the outcome. The query allocation mechanism was executed 1000 times over the previously described setup. The error tolerance was fixed, $\epsilon = 40$ for the execution of the experiments. Since the data are gathered in batches before a statistical signature is built, to ensure synchronization between the queries we fixed the size of the batches to 40 rows, resulting in statistical signatures for data of 40 rows and 2 columns.

The results in Fig. 2 shows that the V'_{total} values obtained through the use of the query allocation mechanism are significantly lower when compared with the values occurring when the mechanism is not in use. The observed variance decrease fluctuates between $30\% - 60\%$, depending on the query. The reduced variance indicates a *higher quality of the query response* as it will be based on data with smaller variance not including irrelevant contextual information.

In order to examine the relation of the reduced variance and the involvement ratio, we compared the two metrics in Fig. 3. The plot shows that variance decrease is, for the most of the time, inversely proportional to the involvement ratio. What this reveals that data from nodes with irrelevant information are *not* included in the query execution process. This results from the selectivity of the statistical signatures, which are exploited to represent the current eigenbase of each contextual data space of each EN. Hence, incoming queries are directed to those ENs where the underlying contextual data space is relevant to the analytics query. The statistical relevance results to more accurate predictive analytics results and also to engage only those ENs holding relevance contextual data per query. Our mechanism succeeded in avoiding the use of unnecessary resources for responding to a query, by examining the statistical relevance of the data stored on each node, *before* assigning a query.

In Table I, we included a summary of measurements of the average involvement ratio for different error tolerance values and batch sizes. Our aim is to determine the impact that the error tolerance ϵ has on the involvement ratio. We found that regardless of the batch size, increasing values of the error tolerance lead to higher average involvement ratio.

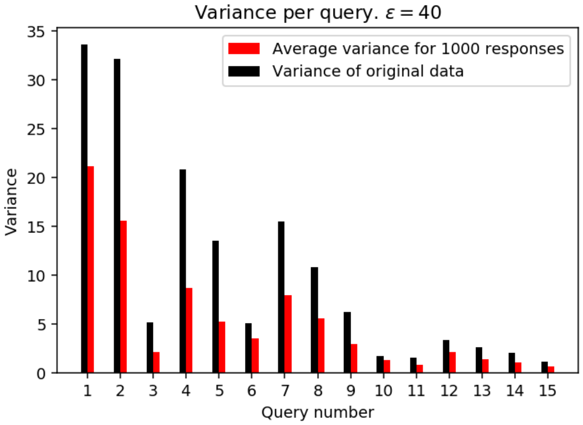

Fig. 2. Comparison between the variance that a query is exposed to w.r.t. the baseline solution and our proposed mechanism.

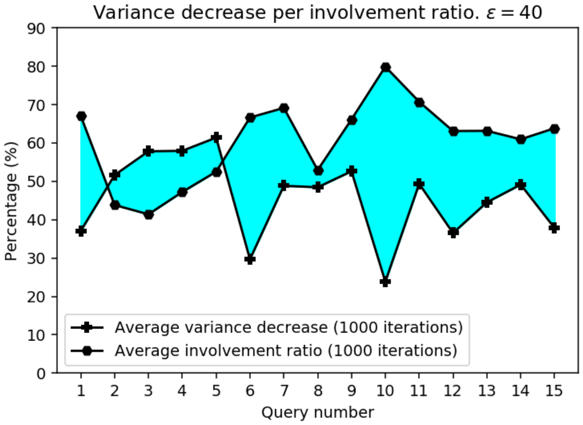

Fig. 3. The involvement ratio r is compared to variance decrease that occurs from our query allocation mechanism.

TABLE I
IMPACT OF ERROR TOLERANCE ϵ ON EXPECTED INVOLVEMENT RATIO r.

Average involvement ratio r			
	Batch size		
tolerance ϵ	30	40	50
20	0.225	0.21667	0.20834
30	0.425	0.38334	0.4375
40	0.6625	0.58334	0.47916
50	0.7625	0.88334	0.85416
60	0.9875	1	0.875

VI. CONCLUSIONS

In this work we proposed an edge-centric query allocation and execution mechanism for analytics queries based on the *statistical relevance* of the edge nodes. The collected contextual data stored in edge computing and sensing devices are exploited to *attract* only those analytics queries that are relevant for the underlying data sub-spaces and then being executed locally to significantly reduce latency. We propose a methodology where entities located in the Cloud undertake the responsibility of receiving analytics queries and decide on the most appropriate edge nodes for queries execution thus minimizing the redundant engagement of edge nodes holding irrelevant contextual data. The decision is based on statistical signatures (derived from the principal eigenvectors) of the datasets of nodes and the direct statistical matching between statistics and analytics queries. Our performance evaluation shows the advantages of our proposed schema in edge computing environments engaging only a subset of nodes relevant for incoming queries.

ACKNOWLEDGMENT

This research has been supported in part by the UK Engineering and Physical Sciences Research Council (EPSRC) projects EP/N033957/1, and EP/P004024/1; by the European Cooperation in Science and Technology (COST) Action CA 15127: RECODIS – Resilient communication and services; and by the EU H2020 GNFUV Project RAWFIE-OC2-EXP-SCI (Grant No. 645220) under the EC FIRE+ initiative.

REFERENCES

[1] Al-Hoqani, N., 'In-Network On-Demand Query-Based Sensing System for Wireless Sensor Networks', Wireless Comm.and Networking Conf. 2017, 1–6.

[2] Anagnostopoulos, C., Quality-optimized predictive analytics, Applied Intelligence, 45(4), 2016, pp. 1034–1046.

[3] Anagnostopoulos, C., Time-optimized contextual information forwarding in mobile sensor networks, J. Parallel and Distributed Computing, 74(5), 2014, pp. 2317–2332.

[4] Balkensen, C., Tatbul, N., 'Scalable Data Partitioning Techniques for Parallel Sliding Window Processing over Data Streams', 8th Int. Workshop Data Management for Sensor Networks, 2011.

[5] Cao, L., Rundensteiner, E., 'High Performance Stream Query Processing with Correlation-Aware Partitioning', VLDB J., 7(4):265–276, 2013.

[6] Cheng, N., et al., Vehicle-assisted device-to-device data delivery for smart grid, IEEE Trans. on Vehicular Technology, 65(4): 2325–2340, 2016.

[7] Chowdhery, A. et al., 'Urban IoT Edge Analytics', Fog Computing in the Internet of Things, Springer, 2018, 101–120.

[8] Diallo, O. et al., 'Real-time data management on wireless sensor networks: A survey', J. Netw. Computer Applications, 1013–1021, 2012.

[9] Gabel, M., et al., Monitoring least squares models of distributed streams, 21th ACM SIGKDD, 319–328, 2015.

[10] Govindarajan, N. et al., Event processing across edge and the cloud for internet of things applications, IEEE MDM 2014, 101–104.

[11] Gedik, B., 'Partitioning Functions for Stateful Data Parallelism in Stream Processing', VLDB J. 23(4):517–539, 2014.

[12] Han, J., Kamber, M., Pei, J., 'Data Mining, Concepts and Techniques', Morgan Kaufmann Publ. 2012.

[13] Harth, N., and Anagnostopoulos, C., Quality-aware Aggregation & Predictive Analytics at the Edge, IEEE Big Data, 17–26, 2017.

[14] Hu, L. et al., 'A Stream processing system for multisource heterogeneous sensor data', Sensors, 2016.

[15] Huacarpuma, R. C., at el., 'Distributed Data Service for Data Management in Internet of Things Middleware', Sensors, 17, 2017.

[16] Kamath, G., et al., Pushing analytics to the edge, IEEE GLOBECOM, 1–6, 2016.

[17] Kolcun, R., McCann, J. A., 'Dragon: Data Discovery and Collection Architecture for Distributed IoT', Int. Conf. on IoT, 2014.

[18] Kolomvatsos, K., 'An Intelligent Scheme for Assigning Queries', Applied Intelligence, 2018.

[19] Kolomvatsos, K., Hadjiefthymiades, S., 'Learning the Engagement of Query Processors for Intelligent Analytics', Applied Intelligence, 46(1):96–112, 2017.

[20] Kolomvatsos, K., Anagnostopoulos, C., 'Reinforcement Machine Learning for Predictive Analytics in Smart Cities', Informatics, 4(16), 2017.

[21] Mo, S. et al., 'TinyQP: A query processing system in wireless sensor networks', Intl Conf Web-Age Info Management, 788–791, 2013.

[22] Quoc, H. N. M., et al., 'A learning approach for query planning on spatio-temporal IoT data', 8th Intl Conf Internet of Things, 2018.

[23] Rahmani, A. M., 'Exploiting smart e-health gateways at the edge of healthcare internet-of-things: A fog computing approach', Future Generation Computer Systems, 78:641–658, 2017.

[24] Widya, P. W., wet al., 'A oneM2M-Based Query Engine for Internet of Things (IoT) Data Streams', Sensors, 18, 2018.

[25] Ye, N., Chen, Q., 'An Anomaly Detection Technique Based on a Chi-Square Statistic for Detecting Intrusions into Information Systems', QRE International, 17:105–112, 2001.

[26] Zeitler, E., Risch, T., 'Scalable Splitting of Massive Data Streams', DASFAA, vol 5982, Springer, 2010.

[27] Harth, N., Anagnostopoulos, C. 'Edge-centric Efficient Regression Analytics', IEEE EDGE 2018, USA.

[28] GNFUV Unmanned Surface Vehicles Sensor Data. https://archive.ics.uci.edu/ml/datasets/GNFUV+Unmanned+Surface+Vehicles+Sensor+Data+Set+2