Proactive & Time-Optimized Data Synopsis Management at the Edge

Kostas Kolomvatsos*, Christos Anagnostopoulos†, Maria Koziri*, Thanasis Loukopoulos‡
* Department of Informatics and Telecommunications, University of Thessaly
†School of Computing Science, University of Glasgow
‡Department of Computer Science and Bioinformatics, University of Thessaly

***Abstract*—Internet of Things offers the infrastructure for smooth functioning of autonomous context-aware devices being connected towards the Cloud. Edge Computing (EC) relies between the IoT and Cloud providing significant advantages. One advantage is to perform local data processing (limited latency, bandwidth preservation) with real time communication among IoT devices, while multiple nodes become hosts of the collected data (reported by IoT devices). In this work, we provide a mechanism for the exchange of data synopses (summaries of extracted knowledge) among EC nodes that are necessary to give the knowledge on the data present in EC environments. The overarching aim is to intelligently decide on when nodes should exchange data synopses in light of efficient execution of tasks. We enhance such a decision with a stochastic optimization model based on the Theory of Optimal Stopping. We provide the fundamentals of our model and the relevant formulations on the optimal time to disseminate data synopses to network edge nodes. We report a comprehensive experimental evaluation and comparative assessment related to the optimality achieved by our model and the positive effects on EC.**

***Index Terms*—Edge Computing, Data Synopsis, Optimal Stopping Theory, Network Monitoring, Context-awareness.**

I. Introduction

The new form of the internet leads to a vast Internet of Things (IoT) infrastructure [37], where numerous autonomous devices are interconnected, interact and exchange data and knowledge. End devices exhibit data sensing and processing capabilities and are connected with the upper layers of the infrastructure to deliver data and knowledge. Knowledge diffusion involves either local (in-network/on-board) data processing or data transfer through the network towards the Cloud for further processing. The increased latency for getting responses when relying to Cloud triggers the need of novel efficient methods for local storage and management. The advent of Edge Computing (EC) [37] deals with data storage and processing close to locations where data are originally observed. There, we evidence numerous EC Nodes (ENs) that process data limiting the latency in the provision of knowledge. Knowledge can be represented in the form of predictive analytics queries, tasks, or explanatory-driven models and statistics, e.g., regression coefficients, histograms, digests. ENs tend to improve their computational resources as the processors and memory become even more powerful, while their connectivity is not guaranteed [21]. The local storage of data can guarantee the efficient support of analytics queries issued in various locations in the network, especially when there is limited connectivity. Certain research efforts study the sophisticated selection of data hosts based on their available memory and energy levels. However, such approaches do not incorporate data consistency mechanisms. Additionally, by having a minimum sufficient statistics of the collected data efficiently assists query assignments tasks as well as execution plans, i.e., one can assign predictive analytics queries to specific EC/IoT nodes instead of migrating data or queries. For instance, if an IoT application requests temperature values over pre-defined threshold, one can trivially exclude nodes that their current datasets contain temperatures below this threshold. Moreover, the distributed data storage assists in the elimination of the probability of data loss and cope with nodes failure.

The intention of our work is to support the context-aware autonomous nature of ENs to efficiently process the data and deliver responses to incoming predictive analytics queries, or simply *queries*. When a new query arrives, ENs should decide *where* the query will be executed based on the requests for data. Decisions should be taken in a distributed manner over the minimum sufficient statistics of data present in the entire ecosystem. That is because, the EN where queries are initially issued might not always be the owner of the data to get an appropriate response. For any decision taken, ENs should have a holistic view on the data of their networked neighboring peers to avoid relying on data migration for query local processing. Instead of migrating the relevant data (obviously, data migration negatively affects the network performance), ENs need to decide to 'migrate' the query and anticipate for the final response. To obtain a holistic view on the data owned by their peers, ENs rely on statistical synopses that are lightweight and can be efficiently transferred within the network. However, such synopses should be valid and effectively depict the statistics of the corresponding dataset(s). In this context, we advance the autonomous behavior of each EN with a context-aware mechanism that: (i) *monitors* the updates on synopses and the network performance; and (ii) decides *when* it is the appropriate time to disseminate these synopses to peer ENs. The appropriate time relies on the dynamic changes in synopses, current status of the network, and future estimations of the network's performance.

Our mechanism adopts on-line statistical learning to support the aforementioned decision making mechanism relying on the awareness of the network performance context and the forecast

Manuscript received April, 2020. Corresponding author: K. Kolomvatsos (email: kostasks@cs.uth.gr).

of the network status. Statistical information is 'smoothly' combined with the changes in the synopses as new data arrive in ENs by developing an ensemble of past observations and future estimates. The current and future network status and the updates of synopses become the basis of our proactive, time-optimized model for deciding *when* it is the right time to disseminate synopses to ENs. The final decision is optimally achieved based on the principles of Optimal Stopping Theory (OST) [38]. To the best of our knowledge, in principle, our mechanism is the first to achieve context-aware and pro-active optimal decision making using OST by taking into account significant differences (in magnitude) in synopses previously disseminated to ENs and nowcasting the current and forecasting the future network status. This yields communication-efficient distribution of synopses. The salient contributions are:

- a time-optimized monitoring mechanism over magnitude of synopsis changes while being aware on the current and future network context;
- statistical learning models trained over historical performance used for forecasting future network performance;
- mathematical formulation and analysis of a proactive OST stochastic optimization model supporting the behavior of autonomous ENs on deciding at the best time to distribute updated synopses to peers;
- extensive experimental evaluation and comparative assessment with other models found in the literature showcasing the strengths of our approach.

The paper is organized as follows: Section II reviews related work & Section III provides an overview of our problem and mechanism. Section IV introduces mechanisms for synopses and network performance monitoring, while Section V introduces statistical learning methods for network performance context. Section VI provides an analysis and stochastic optimization solution of our mechanism. Section VII reports on the experimental evaluation and comparative assessment, while Section VIII concludes the paper with future research agenda.

II. Related Work

The vast amount of data reported through data streams poses unique space and time constraints on their computation [3]. Due to the huge amount of data, data synopses are undeniably useful for a variety of applications. Data synopses depict a 'high' level description of data and represent their statistics [3]. In this work, we focus on synopses used to (i) approximate query [14]: estimate query results in an online manner; (ii) approximate join estimation [24]: estimate the size of a join operation; (iii) calculate aggregate statistics over data [19], [33]; (iv) data mining [1], [2]: provision of synopses instead of individual data points, e.g., clustering, classification rules. Synopses act as 'representatives' of data and usually involve summarizations or selection of specific subsets [31]. Such statistical and sufficient representations become useful when they need to be transferred in the network or be processed in real time for decision making. Indicatively, synopses generation techniques involve sampling [31], load shedding [45], sketching [9] and micro cluster based summarization [1]. Other statistical techniques, like histograms and wavelets are also adopted for synopses [3]. The selection of the specific technique is a decision that depends on the problem context and the system resources. Research efforts in the literature deal with partitioning large volumes of data. The partitioning assists in the parallel management of data and specific techniques are adopted to split the data in a manner where after the end of the process, we obtain a view on the contents of each partition (refer to [40] for a survey). Should we efficiently combine data partitioning with synopsis, we could limit the time required for providing the final responses to issued queries. We expect to significantly reduce the data processing time by obtaining a view on their statistics *in advance*.

Schism in [20] is a workload-aware approach for database partitioning improving the scalability of shared-nothing distributed databases. ElaTras in [23] addresses scalability and leverages elasticity of Cloud data stores by providing scalable transactional access. The data look up process in [27] inspired by mobile computing roaming protocols aims to efficiently handle data migration without affecting scalability. The transaction-aware mechanism for data partitioning in [39] performs efficiently for online transactional applications in the Cloud. The model in [28] guarantees quality of data by applying outliers detection mechanisms to correct data partitions in EC. The above-mentioned approaches are not aware of the edge network status/context (principally: bandwidth, throughput, latency), while not being pro-active of future network performance to decide on knowledge diffusion and/or data share within the network. And this is our primary desideratum of our approach: be network-context aware and proactive by *nowcasting* the present and *forecasting* future network performance, thus, (i) becoming communication efficient while (ii) retaining quality of analytics by optimally deciding *when* to disseminate altered synopses.

Resource management at the network edge is emerging. When tasks are offloaded at the edge, there is a need for efficient resource management that facilitates task/data processing and the timely provision of results. ENs adopt three fundamental policies for workloads execution [46]: (i) through *aggregation models* where data, coming from multiple devices, are collected and pre-processed in an EN [47]. Data are locally processed before transferred to Cloud, thus, limiting the time for delivering the results; (ii) through *collaborative models* where ENs interact with IoT devices offloading subsets of tasks [50]; (iii) through *centralized approach* where ENs act as execution points for the tasks offloaded by IoT devices [41]. ENs exhibiting more computational capabilities than IoT devices undertake the responsibility of performing intensive tasks, however, at the expense of being overloaded. As discussed in [46], research in offloading workloads towards ENs focus on caching [12], context-aware web browsing, and video pre-processing [42], while, recently, approaches cope with the resource management problem [16], [46], [34]. They address the challenges of deciding on tasks/data offloading to ENs taking into consideration communication as in [13] and [36]. However, the decision making of the above-mentioned approaches is unaware of the past, current, and future network performance, which is a crucial factor especially in dynamic environments. More importantly, obtaining local knowledge *in* *advance* for tasks/data offloading (achieved by synopses over local data) requires efficient mechanisms for timely, up-to-date synopsis diffusion reflecting actual changes of the data. This significantly drives the quality of results obtained once ENs are *holistically* aware of their data and neighbors minimum sufficient statics thus guaranteeing data/query-task execution only to *appropriate* ENs. Our mechanism supports these desiderata by not only drastically departing from the current approaches being network-context and data-summaries aware, but also leveraging the decision making by being pro-active due to (i) learning the past and forecasting future network context and (ii) radically relying on the evaluation of inherent changes on synopses due to the changing nature of the underlying data; which is a dominant factor in EC environments. And, the fundamental differentiation of our approach with the literature is that the decision making is achieved by fulfilling an incremental stochastic optimization objective, departing from arbitrary hard/threshold-based decision makings found in certain approaches, e.g., [13], [36]. To the best of our knowledge our mechanism is the first network context and synopsis-aware stochastic objective-driven decision making, whose performance and adaptability is thoroughly evaluated and presented in our experimental section.

III. Preliminaries and Overview

A. Edge Computing Infrastructure

For a nomenclature, refer to Table I (r.v. stands for random variable). Consider a set of N ENs $\mathscr{N} = \{n_1, n_2, \dots, n_N\}$, with datasets $\mathscr{D} = \{D_1, D_2, \dots, D_N\}$. Each EN n_i communicates directly with its networked neighbors in $\mathscr{N}_i \subseteq \mathscr{N}$ (see Fig. 3(left)). Every dataset $D_i = \{\mathbf{x}_j\}_{j=1}$ contains real-valued d-dimensional data vectors $\mathbf{x} = [x_1, x_2, \dots, x_d]^\top \in \mathbb{R}^d$. Each dimension refers to contextual feature, e.g, temperature, humidity, gas concentration. Contextual vectors can be either captured by ENs themselves or delivered by sensors directly connected to ENs. EN n_i incrementally extracts *knowledge* from its D_i by adopting any arbitrary methodology, e.g., on-line regression, classification, estimation of uni-variate histogram per attribute, non-linear statistical dependencies among attributes, and clustering. Without loss of generality, we abstract each on-line knowledge derivation methodology as a *statistical synopsis* $\mathscr{S}$. Such synopsis is represented in a vectorial space via a l-dimensional synopsis vectors $\mathbf{s} = [s_1, s_2, \dots, s_l]^\top \subset \mathbb{R}^l$.

Definition III.1. A synopsis $\mathscr{S}_i$ is a summarization of the dataset D_i of the EN n_i depicting its statistical characteristics.

Synopsis Examples: A synopsis $\mathscr{S} \equiv \{\mathbf{s}\}$ represents the linear multivariate regression coefficients $\mathbf{s} \in \mathbb{R}^d$ between the input vectors $\mathbf{x}_{IN}^\top \in \mathbb{R}^{d-1}$ and the output $x_O \in \mathbb{R}$ from data $\mathbf{x} = [\mathbf{x}_{IN}, x_O]$ such that $x_O = \mathbf{x}_{IN}^\top \mathbf{s}$. Alternatively, in non-linear regression, $\mathbf{s} \in \mathbb{R}^l$ represents the non-linear coefficients of the function estimation $x_O \sim f(\mathbf{x}_{IN}; \mathscr{S})$ as e.g., $\mathscr{S} \equiv \{\mathbf{s}_j, \alpha_j\}_{j=1}^k$ being the set $k < |D|$ of the Support Vectors (SV) $\mathbf{s}_j$ of the data set D and the associated SV coefficients α_j. In clustering, synopsis $\mathscr{S}$ represents the k-centroids, i.e., the k-means $\mathscr{S} \equiv \{\mathbf{s}_1, \dots, \mathbf{s}_k\} \subset \mathbb{R}^d$ of the data space defined by D with $k < |D|$. In linear data compression, synopsis $\mathscr{S} \equiv \{\mathbf{s}_1, \dots, \mathbf{s}_k\} \subset \mathbb{R}^d$ represents the first $k < d$ Principal Components (PCs) of the eigenbase corresponding to the dataset D. These examples abstract synopsis as parameters in statistical learning: supervised (regression) and unsupervised (clustering; compression). Obviously the list is not exhaustive; however, we can fairly represent the derived knowledge from each EN's dataset via synopsis.

Application Scenarios: In the network edge, we obtain N local synopses $\mathscr{S}_1, \dots, \mathscr{S}_N$ over contextual data for building e.g., ML-driven tasks (clustering, regression, dim. reduction); refer to Unmanned Surface Vehicles (USVs) floating over sea surface for environmental monitoring in GNFUV project[1] in Fig. 1. Every EN/USV is responsible for maintaining synopsis up-to-date due to underlying data change (concept drift) and monitoring network context, e.g., throughput. ENs share synopses regularly obtaining a 'neighboring view' on the statistical properties of peers' data while executing tasks/queries (e.g., inference) allocated by a central base (Fig. 1). This supports applications requiring pro-active decision making for: (i) allocating tasks/analytics queries *only* to relevant ENs based on their synopses studied in [30] and [49], (ii) share and update ML models in Federated Learning [5] for environmental monitoring based on synopses and network load, (iii) transfer tasks for ML models training & inference ENs based on network congestion [36], and (iv) separating context data by distributively gathering *similar* data to *same* datasets studied in [28]. As exemplified in Fig. 1, context data are stored to ENs, thus, one can effectively train local ML models, securing quality of predictive analytics tasks. This evidently facilitates the allocation of analytics queries to ENs accessing only the most 'relevant' data for analysis.

ENs exchange synopses *and not data*, however, potential problems are expected related to increased network congestion, when data change and synopsis share rates are high, and synopsis inconsistencies, when synopses share rate is kept low. We trade off between communication overhead, 'freshness' and consistency of synopses: ENs should share up-to-date synopses letting peers obtain consistent synopses to reflect data statistical properties, at the expense of deluging the edge. We can potentially reduce sharing rate achieving less expected network deluge in light of 'obsolete' synopses and inconsistency among local and delivered synopses, thus degrading quality of analytics tasks due to discrepancy in data summaries across peers. By inefficiently flooding synopses updates being unaware of network context dramatically increases network congestion & latency violating the EC's principle. A sophisticated decision making is then deemed necessary to optimally balance this trade-off taking into consideration the dynamic nature of network and data contexts in EC.

B. Overview of Our Approach

We focus on the above-mentioned trade off by contributing with a distributed decision making mechanism for synopsis dissemination; the overarching aim is to let ENs know about the inferred knowledge on the distributed datasets to decide on

[1] https://sites.google.com/view/gnfuv/home

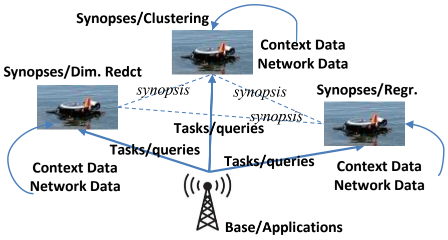

Fig. 1: USVs (GNFUV[1]) share synopses while monitoring network context and environmental data, and executing tasks.

TABLE I: Nomenclature

Notation	Description
$\mathscr{N}$, $\mathscr{N}_i$	Set of ENs, neighborhood of EN n_i
n_i	EC node (EN) i
D_i	ith dataset at the ith EN
$\mathbf{x}$	d-dimensional data vector in $\mathbb{R}^d$
$\mathscr{S}_i$	Synopsis in EN n_i
$\mathbf{s}$	l-dimensional synopsis vector in $\mathbb{R}^l$
CF	Cluster Feature in tree-based synopsis creation
α	Least number of points in synopsis dominant clusters
t	Time instance for synopsis update
$\mathbf{r}$	Synopsis update quantum
T	Pre-defined interval for sending a synopsis
R	Magnitude (L1 norm) of update quantum (r.v.)
Z	Reward at every round of the update epoch (r.v.)
V; I	Network performance and agreement indicators (r.v.)
v; θ	Network performance parameter; threshold
β, $\hat{\beta}$, $\tilde{\beta}$	Actual, estimated and forecast probabilities $P(v \leq \theta)$
$\mathscr{W}$	Sliding window size T of network performance values
γ	Odds factor in optimal stopping criterion (16)
ε	SMAPE forecast error
$f_X(x)$; $F_X(x)$	probability density; cumulative distribution function of X

certain analytics tasks. Specifically, we introduce a sequential decision making mechanism on each EN that attempts to find out *when* it is the *right* time to disseminate local synopses to peer ENs for enhancing the knowledge awareness. We avoid enforcing all the ENs disseminating their updated synopses just 'after' the expiration of a pre-defined interval. That is simply not the case, since not all synopses updates are synchronized as different ENs sense different data with different patterns/trends, thus, we cannot assume that all the derived synopsis are subject to change. Moreover, on a strict interval-based protocol, ENs should disseminate their synopses no matter if they are the same as in the previous reporting epoch. If ENs were 'synchronized' for synopses dissemination, the network would be expected to be deluged, which does not scale especially when the number of ENs is relatively high.

We enhance ENs' behavior that decides the delivery of synopses to peers just after it is expected that synopses will have been significantly altered compared to previous epoch(s). The target is to optimally limit the communication while ensuring high quality of disseminated synopses. A synopsis decision dissemination is based on the qualitative perspective of significant changes of synopses while being aware of the network performance. The EN monitors: (i) the significant changes in local synopsis as new data are received and (ii) the network performance over fundamental metrics like latency, bandwidth, loss of connectivity. We then implicitly incorporatee into the network the necessary 'randomness' for individual decision making, thus, potentially avoiding network flooding. Such 'randomness' is achieved due to different data arriving to different ENs and their stochastic decision making. The 'randomness' assists in reducing the possibility of synopses dissemination at the same time limiting network deluge.

Assume that at discrete time $t = \{1, 2, \dots\}$ a new data vector $\mathbf{x}_t$ arrives at EN n_i. The synopsis $\mathbf{s}_i$ is then updated to a new $\mathbf{s}_i^t$. Let $\mathbf{r}_t = \mathbf{s}_i - \mathbf{s}_i^t$ be the difference (or discrepancy) over the current (last sent) $\mathbf{s}_i$ and the updated synopsis $\mathbf{s}_i^t$. We coin this discrepancy as *update quantum*.

Definition III.2. Update quantum $\mathbf{r}_t$ is the discrepancy between $\mathbf{s}_i$ and $\mathbf{s}_i^t$ synopsis vectors.

EN n_i calculates synopsis discrepancy $\mathbf{r}_t$ at time t as new data vectors are received. ENs should delay the delivery of altered $\mathbf{s}_i^t$ until they experience significant difference, i.e., a high *magnitude* of $\mathbf{r}_t$. Then, neighboring ENs should be efficiently informed on the altered synopsis.

Definition III.3. An *update epoch* is the amount of time between sending two consecutive updates of the current synopsis.

The update epoch is realized within pre-defined intervals, $T, 2T, 3T, \dots$ ($T > 0$). Without loss of generality, we focus on an individual interval, e.g., $[1, \dots, T]$. At each t, ENs check if $\mathbf{r}_t$ excuses to send the updated synopsis before the update epoch expires taking into account the network context. ENs observe the time series of update quanta $\{\mathbf{r}_t\}$. At the same time, ENs nowcast and forecast the network performance to gain a holistic view on the best time for synopsis dissemination. Our idea is to accumulate the magnitude of $\mathbf{r}_t$ over time and identify when the accumulated quanta depict significant discrepancy in the synopsis. When such discrepancy is high and cannot be tolerated, ENs should *stop* the monitoring process, *send* the altered synopsis and *start off* a new epoch.

IV. Synopses & Network Performance Monitoring

A. Synopses Construction & Update

The synopsis is update is due to reception of new data. There are many synopsis construction methods [18], [3], e.g., sampling, histograms, wavelets, sketches, and micro-cluster summarization. Our mechanism is not affected by the synopsis construction methods; however, fast techniques are adopted for ENs to reduce computational load. Synopsis update followed by decision making should be performed in real-time. Therefore, we adopt the online micro-clustering [7] for synopsis creation, since it supports hierarchical grouping of the received data depicting their statistics at the same time. Micro-clustering defines a tree-based structure based on the *Cluster Feature* $CF = \langle L, LS, SS \rangle$ maintained for each cluster such that: L is the number of data points; LS is the linear sum of data points, and SS is the square sum of data points SS. The most significant feature of this approach is that we add CFs to aggregate two clusters. Our mechanism calculates the CFs at pre-defined intervals. In intermediate tree-nodes, we add entries in the form $[CF_k, Child_k]$, where pointers to underlying children are kept to maintain the hierarchy of clusters. The number of leaf nodes is threshold-controlled which affects the tree height. When new data arrive in an EN, the closest cluster is found and the internal and leaf nodes are updated accordingly, i.e., alterations in CFs. Once multiple clusters have been constructed, the synopsis refers to the internal node(s) that represent clusters with at least α data vectors. When the synopsis is to be sent to peer ENs, we scan the CF-tree and retrieve those clusters containing at least α data points, i.e., synopsis is the set of the α-dominant clusters. Let median $M = median_k(L_k)$ and median absolute deviation (MAD) of number of points L_k per cluster be $MAD = median_k(|L_k - M|)$, we select those clusters with at least $\alpha = M - 3MAD$[2]; any other cluster is considered as 'outlier' and excluded from the synopsis. Therefore, synopsis represents the data (sub)regions that dominate the dataset and excludes data that are dissimilar to the majority.

B. Network Performance Monitoring

Each EN apart from maintaining synopsis, monitors, nowcasts and forecasts network performance metrics to assess the optimal synopsis delivery time. Network performance metrics are categorized into [25]: (i) *communication availability*; (ii) *packet loss and error*; (iii) *end-to-end latency*; (iv) *throughput* (volume of data transferred per time unit). There are many efforts for monitoring packet error loss, latency and throughput while operating systems offer access to such parameters (e.g., via files -/proc/net/dev or commands -netperf in Linux). The approach in [43] predicts bandwidth congestion using weighted random early detection, while LinkWidth [15] observes the induced traffic fluctuation and accurately characterizes the available bandwidth of individual links and entire network paths. Concerning packet error loss, time series prediction and association analysis between traffic delays and queue sizes at network gateways are studied in [8], latency prediction in [32], and estimation of throughput in [35]. In this work, we rely our nowcasting and forecasting on the throughput (Mbit/s) metric whose statistical properties are incorporated to the optimal decision making.

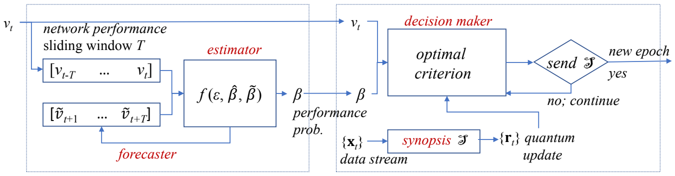

Fig. 2: Components of EN n_i: Network performance estimator and forecaster, synopsis maintenance and OST decision maker.

V. Statistical Learning of Network Performance

EN n_i when receiving new data $\mathbf{x}$ calculates the update quanta between its current synopsis and the one sent to peer ENs $\mathcal{N}_i$. Let $\mathbf{s}$ and $\mathbf{s}^t$ be the current and updated synopsis at t,

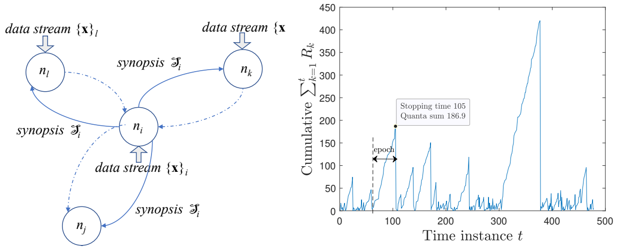

Fig. 3: (Left); EN n_i and its neighborhood $\mathcal{N}_i$ exchanging synopses; (right) cumulative sum of synopsis discrepancies and epoch definition against time t.

respectively, with discrepancy $\mathbf{r}_t = \mathbf{s} - \mathbf{s}^t = [s_1 - s_{1,t}, \dots, s_d - s_{d,t}]$. Let the L_1 norm (magnitude) of discrepancy

$$R_t = \|\mathbf{r}_t\|_1 = \|\mathbf{r}_t\|_1 = \sum_{k=1}^{d} |s_k - s_{k,t}|, \tag{1}$$

and let V_t be the random variable (r.v.) indicating whether a network performance parameter value v_t (e.g., throughput) is below threshold θ at t. Hence, $V_t = 1$, if $v_t \leq \theta$ denoting (without loss of generality) that network performance is of bad quality; otherwise $V_t = 0$. We define the probability $\beta = P(v_t \leq \theta) = P(V_t = 1)$, thus, $P(V_t = 0) = 1 - \beta > 0$. V_t is affected by the type of the performance parameter and therefore $\beta = F_V(\theta)$ is the value of the cumulative distribution function (cdf) F_V of network parameter V. Performance parameters can be proportional or non-proportional. For having a high quality network performance, we should have proportional values exceeding threshold and non-proportional values below threshold, e.g., good network is indicated by increased throughput with low packet loss. Two components in Fig. 2, probability β estimator and forecaster, are introduced to nowcast and forecast network context.

A. Network Performance Probability Estimator

EN's estimator shown in Fig. 2 nowcasts a proportional network parameter v over time storing the recent T values $v_{t-T+1}, v_{t-T+2}, \dots, v_t$ in a sliding window $\mathcal{W}$. EN estimates $\beta = F_V(\theta)$ via estimating the probability density function (pdf) $f_V(v) = P(V = v)$ of v adopting non-parametric Kernel Density Estimation (KDE) [11]. Having T recent samples $\{v_{t-T+k}\}_{k=1}^{k=T}$, $f_V(v)$ is estimated as:

$$\hat{f}_V(v;T) = \frac{1}{T \cdot h} \sum_{k=1}^{T} K\left(\frac{|v - v_{t-T+k}|}{h}\right), \tag{2}$$

where $h > 0$ is the bandwidth of the symmetric kernel $K(u)$ (integrating to unity). One of the most frequent adopted kernel function is the Gaussian, i.e., $K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$. We rely on an incremental estimation of $\hat{f}_V(v)$, $\hat{f}_V(v;k)$ for $k = 1, \dots, T$, by previous estimate $\hat{f}_V(v;k-1)$ and the current value v_k. We recursively obtain for $k = 1, \dots, T$:

$$\hat{f}_V(v;k) = \frac{k-1}{kh} \hat{f}_V(v;k-1) + \frac{1}{kh} K\left(\frac{|v - v_{t-T+k}|}{h}\right) \tag{3}$$

Upon capturing v_k at k, $\hat{f}_V(v;k)$ is incrementally estimated by $\hat{f}_V(v;k-1)$, thus, there is no need to store all the previous

[2] Miller, J. (1991). Reaction time analysis with outlier exclusion: Bias varies with sample size. J Exper. Psych. 43A(4):907–912

$k-1$ values for estimating $\hat{f}_V(v;k)$. If we apply the Gaussian function on the KDE, we obtain an estimation of the cdf $\hat{F}_V(v;T) = \int_{v_{\min}}^{v} \hat{f}_V(u;T)du$ using the T values $\{v_{t-T+k}\}_{k=1}^{k=T}$:

$$\hat{F}_V(v;T) = \frac{1}{T}\sum_{k=1}^{T}\frac{1}{2}\left(1+\operatorname{erf}\left(\frac{v-v_{t-T+k}}{\sqrt{2}}\right)\right) \tag{4}$$

where $\operatorname{erf}(\cdot)$ is the error function. Hence, at time t, we obtain the estimation of $\beta = P(V_t = 1) \approx \hat{\beta} = \hat{F}_V(\theta;T)$.

B. Network Performance Probability Forecaster

At t, EN estimates β based on T historical values, i.e., $\hat{\beta}$. EN uses these values (see Fig. 2) for forecasting the probability $P(V_{t+\tau} = 1)$ in horizon T future time instances $\tau = 1,\ldots,T$. That is, EN forecasts $v_{t+\tau}$ where $\tau = 1,\ldots,T$ and then predicts the *future* β value, notated by $\tilde{\beta}$, in the upcoming horizon T. Our decision is based on the historical as well as on the future trends of the network parameter. When $\tau = 1$, we obtain one-step ahead forecasting at time $t+1$.

For performing long-term forecasting, strategies can be recursive; direct; DirRec; MIMO; and DIRMO [44]. Our desideratum is to forecast $v_{t+\tau}$ within future horizon T based on the past T observed values up to t and the forecasts from $t+1$ up to $t+\tau-1$, $\tau > 1$. Hence, we seek a method that cannot be prune in error collection in successive rounds. We then adopt the direct strategy to forecast the upcoming values $v_{t+\tau}$, $\tau = 1,\ldots,T$. Such strategy does not use any approximated values to calculate future estimations being then immune to accumulation of errors. We use an ensemble of forecasters and randomly select one for each of the τ steps as studied in [29]. Based on the T forecasts, i.e., $\tilde{v}_{t+1},\ldots,\tilde{v}_{t+T}$, we estimate the $P(\tilde{V} = v)$ of the forecast $\tilde{V}$ parameter using KDE *over* the T forecast values:

$$\tilde{f}_{\tilde{V}}(v;T) = \frac{1}{T\cdot h}\sum_{\tau=1}^{T} K\left(\frac{|v-\tilde{v}_{t+\tau}|}{h}\right). \tag{5}$$

Similarly, using the recursion for the incremental approach as described above, we estimate the cdf $\tilde{F}_{\tilde{V}}(v;T)$, thus, we obtain an estimate of the *future* forecast β within the next horizon T such that: $\tilde{\beta} = \tilde{F}_{\tilde{V}}(\theta;T)$. Note that, $\tilde{\beta}$ estimates the probability that a forecast $\tilde{V}$ value is less that θ, i.e., $\tilde{\beta} = P(\tilde{v} \leq \theta)$.

The estimated cdf of past values indicates the realization of $\hat{\beta}$ based on the actual observations of the monitored network performance. We introduce the concept of *reliability* on the performance metric when the actual performance metric can be successfully predicted given any arbitrary forecast error $|v_t - \tilde{v}_t|$. Given this concept, the estimated cdf of future values indicates the realization of $\tilde{\beta}$ based on the future forecasts. The upcoming epoch is based on a reliable prediction of the performance metric given the recent T historical values and the expected forecast error. Both, $\hat{\beta}$ and $\tilde{\beta}$ are aggregated through an ensemble average to obtain a reliable β for the upcoming epoch. Specifically, the predictability of the forecaster is taken into consideration for aggregating $\hat{\beta}$ and $\tilde{\beta}$. We use the symmetric mean absolute percentage forecast error SMAPE:

$$\varepsilon = \frac{1}{T}\sum_{i=1}^{T}\frac{|v_i-\tilde{v}_i|}{|v_i|+|\tilde{v}_i|} \in [0,1], \tag{6}$$

derived from the previous epoch after observing the actual performance values. Then, the EN starts off an epoch with:

$$\beta = \tilde{\beta} + \max(0.5,\varepsilon)(\hat{\beta} - \tilde{\beta}). \tag{7}$$

With $\varepsilon \geq 0.5$ (less reliability) we pay attention on historical observations; while $\varepsilon < 0.5$ makes the model to enter the decision making epoch with high reliability.

VI. Time-Optimized Synopses Management

Consider the random indicator I_t over v_t w.r.t. θ such that:

$$I_t = \begin{cases} 1 & : v_t > \theta \bigwedge \tilde{v}_t > \theta, \text{ (agreement)} \\ 0 & : v_t \leq \theta \bigwedge \tilde{v}_t \leq \theta, \text{ (agreement)} \\ 1-\varepsilon & : \text{in any other case (disagreement).} \end{cases} \tag{8}$$

I_t indicates when both the actual v_t and the forecast $\tilde{v}_t$ are in agreement w.r.t. θ or in disagreement. Specifically, when $v_t > \theta \bigwedge \tilde{v}_t > \theta$ or $v_t \leq \theta \bigwedge \tilde{v}_t \leq \theta$, the current observation of the network performance and the forecast are in agreement concerning the quality of the network. In this case, $I_t = 1$ depicting a good quality and $I_t = 0$ indicating a bad quality, respectively, as the forecast aligns with the actual observation compared with θ. In any other case, there is a disagreement between the actual and the forecast values. In this case v_t and $\tilde{v}_t$ are one on each side of θ, i.e., either $v_t > \theta$ and $\tilde{v}_t < \theta$ or $v_t < \theta$ and $\tilde{v}_t > \theta$. The degree of such disagreement is modelled as a function of the forecast SMAPE ε: the lower the error, i.e., v_t and/or $\tilde{v}_t$ approach θ from opposite directions, the lower the disagreement and the indicator of a good performance tends to unity. If the error is relatively high ($\varepsilon \to 1$), then, there is high unreliability on a good performance of the network, thus, the indicator tends to zero, in a pessimistic mode. Based on this formulation, the EN observes the time series $\{I_k\}_{k=1}^{k=t}$ of the indicator ahead of time up to time instance t to judge on a good or bad behavior of the network performance based on the inherent agreement / disagreement of the observed and forecast performance values.

Given a significant cumulative change on the synopsis since the last time $t = 1$ it sent the updated synopsis, the EN desires to disseminate the up-to-date synopsis if $\{I_k\}_{k=1}^{k=t}$ up to t are expected to be at least non-zeros, i.e., $I_k \in \{1, 1-\varepsilon\}$ for $k = 1,\ldots,t$ with high probability. This signals a potential good network in terms of the performance metric for the EN to disseminate the up-to-date synopsis. Moreover, the cumulative sum of the synopsis discrepancies (quanta) triggers the EN for synopsis dissemination as well. The decision making for synopsis dissemination is based on two stochastic quantities:

- continuous good network performance represented by the product of random indicators $\prod_{k=1}^{t} I_k$, desired to be positive ($I_k > 0$) for a long period;
- cumulative sum of discrepancies $\sum_{k=1}^{t} R_k$, desired to be high enough to postpone synopsis dissemination, thus, avoiding network deluge with many updates.

However, we cannot leave such cumulative sum to be arbitrarily high, since the local synopsis will be far discrepant with the most recently sent synopsis, degrading the quality of information. Moreover, we cannot procrastinating the synopsis dissemination in light of communication efficiency since the network performance is not expected to be of high quality for a high period of time since the last time we disseminated the synopsis. Such procrastination is based on the estimated and forecast probabilities $\hat{\beta}$ and $\tilde{\beta}$, respectively. The core idea is that: *the EN should decide to disseminate the updated synopsis dealing with the trade-off between quality of information processing by minimizing the synopsis discrepancies within its peers, with delaying the synopsis dissemination in light of reducing communication overhead.* Nonetheless, when the EN decides on disseminating the updated synopsis, the network at that time instance should be strictly in a good performance, i.e., $v_t > \theta$. Otherwise, the EN should re-schedule the synopsis dissemination in a later stage. Then, given that $P(I_t = 1) = (1-\hat{\beta})(1-\tilde{\beta})$, $P(I_t = 0) = \hat{\beta}\tilde{\beta}$ and $P(I_t = 1-\varepsilon) = 1 - P(I_t = 1) - P(I_t = 0)$, at t the expected network performance is given by:

$$\mathbb{E}[I_t] = \hat{\beta}(1-\tilde{\beta})(1-\varepsilon) + (1-\hat{\beta})(1-\varepsilon\tilde{\beta}). \tag{9}$$

Based on this reasoning, we define a decision function that encapsulates the delay of synopsis dissemination in light of communication overhead given that the network demonstrates a good behavior, thus, triggering the EN to send its updated synopsis in its neighborhood. As long as the EN delays sending the synopsis, then the cumulative discrepancy sum $\sum_{k=1}^{t} R_k$ monotonically increases while the risk of having a positive product $\prod_{k=1}^{t} I_k$ increases to due to the stochastic nature of the network performance. If, at t, the EN decides to send the updated synopsis, we enforce the network to be in a good performance to avoid re-scheduling the synopsis dissemination, as shown in Fig. 3(right). We then define the stochastic function:

$$Z_t \quad = \quad \prod_{k=1}^{t} I_k \cdot \sum_{k=1}^{t} R_k, \tag{10}$$

which reflects the payoff of postponing the synopsis dissemination in light of minimizing communication overhead. Hence, the decision function to be maximized is:

$$Y_t = \begin{cases} Z_t : \text{if } v_t > \theta, \\ 0 \ : \text{otherwise}, \end{cases} \tag{11}$$

Y_t reflects the gain (payoff) of deciding to disseminate an updated synopsis at that time where the network is in a good performance. Note that it might be the case to have decided at $k < t$ to postpone the synopsis dissemination for time t, but at some time instances in between $k+1$ and $t-1$ the network behaves badly, thus, $\prod_{l=k+1}^{t-1} I_l = 0$. In this case, our decision receives zero reward. Or, even if the network behaves good in our decided postponement period, i.e., $\prod_{l=k+1}^{t-1} I_l = 1$, the decision on sending the updated synopsis at t receives zero reward since at that time $v_t \leq \theta$. On the other hand, based on the two above-mentioned sequential risks, we desire to postpone the synopsis update to avoid network overload. Given that I_t, R_t, v_t are stochastic, thus, the decision Y_t to be maximized is also random, we need to find the (optimal) time t^* to attain the supremum of the expectation of Y_t. Our time-optimized problem is to maximize the expectation of decision under the sequential risks expressed as:

$$t^* = \arg ess \sup_{t \geq 0} \mathbb{E}[Y_t], \tag{12}$$

denoting the estimation of the optimal stopping time $t^* \geq 0$ that maximizes the essential supremum of the expected Y_t.

We propose a methodology to find a rule for estimating t^* in (12) to maximize the expectation. First, we need to prove that given our function in (11) with the involved risks, t^* exists and attains the essential supremum of this stochastic expectation in a real-valued horizon, i.e., $t^* < +\infty$. After proving its existence, we require to establish a sequential-based decision making rule which relies on the statistics $\hat{\beta}$ and $\tilde{\beta}$ (observed discrepancies and network perofmance context) that obtains optimal value for t^*. Given this stopping time rule, we require to guarantee its uniqueness in our problem. Any other sequential decision making rule based on the above statistics does not maximize the expectation.

1) Optimal Stopping Theory: The OST deals with the problem of choosing the best time instance to take the decision of performing a certain action. This decision is based on sequentially observed random variables to maximize a well-defined expected reward [17]. For given random variables $X_1, X_2, \ldots$ and measurable functions $Y_t = \mu_t(X_1, X_2, \ldots, X_t)$, $t = 1, 2, \ldots$ and $Y_\infty = \mu_\infty(X_1, X_2, \ldots)$, the problem is to find a stopping time t^* to maximize $\mathbb{E}[Y_{t^*}]$. t^* is a random variable with values in $\{1, 2, \ldots\}$ such that the event $\{t^* = t\}$ is in the Borel field (filtration) $\mathbb{F}_t$ generated by $X_1, \ldots, X_t$, i.e., the only available information we have obtained up to t: $\mathbb{F}_t = \mathbb{B}(X_1, \ldots, X_t)$. Hence, the decision to stop at t is a function of $X_1, \ldots, X_t$ and does not depend on future observations.

Theorem VI.1 (Existence of Optimal Stopping Time)**.** *If* $\mathbb{E}[\sup_t Y_t] < \infty$ *and* $\lim_{t \to \infty} \sup_t Y_t \leq Y_\infty$ *almost surely (a.s.), then the stopping time* $t' = \inf\{t \geq 1 | Y_t = ess \sup_{t' \geq t} \mathbb{E}[Y_{t'} | \mathbb{F}_t]\}$ *is optimal.*

Proof. See [17]. ■

The (essential) supremum $ess \sup_{t' \geq t} \mathbb{E}[Y_{t'} | \mathbb{F}_t]$ is taken over all stopping times t' such that $t' \geq t$ a.s. The optimal stopping time t^* is obtained through the 'principle of optimality' [4].

2) An Optimal Stopping Time for Synopses Dissemination: The proposed time-optimized synopses dissemination model concludes on a decision based on sequential observations of pairs (I_t, R_t) at t. Without loss of generality, I_t and R_t are statistically independent: the former expresses the network performance status, while the latter the discrepancy of the current synopsis due to possible changes of the underlying data. Persisting non-zero values of I_t make ourselves confident on disseminating the new synopsis given a significant cumulative discrepancy. On the other hand, as we delay our decision on a dissemination decision, the observation process progresses further, thus yielding possible higher synopsis discrepancy given a bad network performance. A decision taken at t is:

- either to assert that the network quality is high along with significantly high cumulative discrepancy, then, disseminate the updated synopsis to the neighbouring ENs,
- or continue the synopsis observation and network monitoring process at time $t+1$, then, proceed with a decision.

We will show that our mechanism, illustrated in Fig. 2, based on the cumulative discrepancy sums $\sum_{\tau=1}^{t} R_\tau$ and network performance indicator products $\prod_{\tau=1}^{t} I_\tau$ determines an optimal stopping time that maximizes (12).

Theorem VI.2. *An optimal stopping time for the synopsis dissemination decision problem in (12) exists.*

Proof. Based on Theorem VI.1 We have to prove that the optimal stopping time t^* exists and is derived from the principle of optimality [4]: prove that $\lim_{t\to\infty} \sup Y_t \leq Y_\infty$ a.s. and $\mathbb{E}[\sup_t Y_t] < \infty$. I_t's are non-negative and from the strong law of numbers $\frac{1}{t}\sum_{\tau=1}^{t} I_\tau \to \mathbb{E}[I]$ a.s. such that:

$$Y_t \leq t\gamma^t (1/t) \sum_{\tau=1}^{t} I_\tau \approx t\gamma^t \mathbb{E}[I] \overset{a.s.}{\to} 0, \quad (13)$$

with $\lim_{t\to\infty} \sup_t Y_t = Y_\infty = 0$ for $\gamma \in (0,1)$. In addition,

$$\sup_t Y_t \leq \sup_t \gamma^t \sum_{\tau=1}^{t} I_\tau \leq \sup_t \sum_{\tau=1}^{t} \gamma^\tau I_\tau \leq \sum_{\tau=1}^{\infty} \gamma^\tau I_\tau. \quad (14)$$

Hence, $\mathbb{E}[\sup_t Y_t] \leq \sum_{\tau=1}^{\infty} \gamma^\tau \mathbb{E}[I] = \mathbb{E}[I]\frac{\gamma}{1-\gamma} < \infty$. ■

The network indicators $\{I_t\}$ are non-negative, thus, our problem is classified as *monotone* [17]. In such a case, t^*, since it exists by Theorem VI.2, is obtained by the *1-stage look-ahead optimal rule* (1–sla) [17]. In this particular case of our problem, we should decide on a synopsis dissemination at the *first* stopping time t at which $Y_t \geq \mathbb{E}[Y_{t+1}|\mathbb{F}_t]$ given the filtration $\mathbb{F}_t = \{(I_1, R_1,)\}$ of the observed indicator values and synopsis discrepancy values up to t, i.e.,

$$t^* = \inf\{t \geq 1 | Y_t \geq \mathbb{E}[Y_{t+1}|\mathbb{F}_t]\} \quad (15)$$

In our monotone optimal stopping problem with observed pairs $(I_1, R_1), (I_2, R_2), \ldots$ and rewards $Y_1, Y_2, \ldots, Y_\infty$, the 1-state look-ahead optimal since $\sup_t Y_t$ has finite expectation with $\lim_{t\to\infty} \sup_t Y_t = Y_\infty = 0$ a.s.; refer to proof of Theorem VI.2.

Theorem VI.3. *Given the L_p-norm expectation of the synopsis discrepancy $\mathbb{E}[\|\mathbf{r}\|_p]$ and the approximated probabilities $\hat{\beta} > 0$ and $\tilde{\beta} > 0$ of the observed and the predicted network performance metric, the optimal stopping time t^* for the synopsis dissemination decision problem in (12) is:*

$$t^* = \inf\{t \geq 1 | \sum_{k=1}^{t} R_k \geq \frac{\gamma}{1-\gamma}\mathbb{E}[\|\mathbf{r}\|_p]\}, \quad (16)$$

with $\gamma = (1-\hat{\beta})(1+(1-\varepsilon)\hat{\beta}(1-\tilde{\beta})-\varepsilon\tilde{\beta})$.

Proof. The expectation of $\mathbb{E}[Y_{t+1}|\mathbb{F}_t]$ depends on the filtration $\mathbb{F}_t$. Assume that up to t, the risk $\prod_{k=1}^{t} I_k > 0$ and we have not stopped. That is, we take one more observation (I_{t+1}, R_{t+1}) and then decide on $t+1$. We obtain that:

$$\mathbb{E}[Y_{t+1}|\mathbb{F}_t] = \mathbb{E}[Z_{t+1}|\mathbb{F}_t] \cdot P(v_{t+1} > \theta|\mathbb{F}_t). \quad (17)$$

Since $Z_{t+1} = \prod_{k=1}^{t} I_k \cdot I_{t+1} \cdot (\sum_{k=1}^{t} R_k + R_{t+1})$, given the filtration $\mathbb{F}_t$, we obtain that:

$$\mathbb{E}[Z_{t+1}|\mathbb{F}_t] = \prod_{k=1}^{t} I_k \cdot \mathbb{E}[I_{t+1}] \cdot (\sum_{k=1}^{t} R_k + \mathbb{E}[R_{t+1}]), \quad (18)$$

with probability $P(v_{t+1} > \theta|\mathbb{F}_t) = 1-\hat{\beta}$. Hence, the expected reward at $t+1$ given that we do no stop at t is then:

$$\begin{aligned}\mathbb{E}[Y_{t+1}|\mathbb{F}_t] &= (1-\hat{\beta})(1+(1-\varepsilon)\hat{\beta}(1-\tilde{\beta})-\varepsilon\tilde{\beta}) \\ &\cdot \prod_{k=1}^{t} I_k \cdot (\sum_{k=1}^{t} R_k + \mathbb{E}[\|\mathbf{r}\|_p]),\end{aligned}$$

replacing the expectation $\mathbb{E}[I]$ from (9) and the expectation of the synopsis discrepancy $\mathbb{E}[R] = \mathbb{E}[\|\mathbf{r}\|_p]$. Based on the 1-step look ahead optimal stopping rule, we stop and disseminate the synopsis at the first time instance where the current reward at t is greater than the expected reward should we continue at $t+1$ and then stop. Hence, we obtain the inequality from (15), that is, we stop when $Y_t = \prod_{k=1}^{t} I_k \cdot \sum_{k=1}^{t} R_k \geq \mathbb{E}[Y_{t+1}|\mathbb{F}_t]$ or at time t^* such that: $t^* = \inf\{t \geq 1 | \sum_{k=1}^{t} R_k \geq \frac{\gamma}{1-\gamma}\mathbb{E}[\|\mathbf{r}\|_p]\}$, with $\gamma = (1-\hat{\beta})(1+(1-\varepsilon)\hat{\beta}(1-\tilde{\beta})-\varepsilon\tilde{\beta})$. ■

Since our optimal stopping time problem exists and is monotone according to Theorems VI.2 and VI.1, the provided optimal stopping time in Theorem VI.3 is unique.

Corollary 1. *The solution of the optimal stopping time t^* provided in Theorem VI.3 is unique.*

Proof. See [22]. ■

Corollary 2. *The EN starts off the synopsis dissemination decision process iff: $\hat{\beta} < \left(\frac{\tilde{\beta}}{1-\tilde{\beta}}\right)\left(\frac{\varepsilon}{1-\varepsilon}\right)$.*

Proof. The odds in Eq(16) $\frac{\gamma}{1-\gamma}$ should be positive. Since $\gamma > 0$, we obtain the upper bound of $\hat{\beta}$ by re-arranging the factors. ■

EN locally executes Algorithm 1: EN monitors past and predicts future network performance (lines 1-2) to estimate β for the OST process, while accumulates quanta (lines 5-7) to decide on synopsis dissemination to neighbors (lines 8; 13-16). This requires $O(T)$ time for estimating & predicting β, $O(dT)$ space for storing data in $\mathscr{W}$, $O(1)$ time for assessing criterion in (16). Space and time complexity is $O(dT)$ for maintaining the CF-based synopsis. Evidently, the overall mechanism is deemed appropriated w.r.t. computational load to be adopted by ENs in resource-constrained EC environments due to its lightweight complexity and low-memory-footprint.

VII. Performance Evaluation

We provide a performance evaluation of our model, coined **Optimal Stopping Decision Model (OSDM)** and a comparative assessment with the approaches in [13], [34], and [36].

A. Experimental Setup

1) Performance Metrics: We first calculate the *density of optimal stopping times t^** to investigate the statistical properties of the time required for deciding on synopsis dissemination to peers. In addition, given a time horizon T, we define the *expected percentage of conclusion decision time* ϕ where OSDM requires to reach a decision over n runs, i.e., $\phi = \frac{1}{n}\sum_{k=1}^{n} \frac{t_k^*}{T}$. A value of $\phi \to 1$ denotes that OSDM spends the whole horizon T to reach an optimal decision; while

Algorithm 1: Time-optimized algorithm for EN n_i.

```
1  monitor net performance $v_t$; add and slide window $\mathscr{W}$;
2  estimate $\hat{\beta}$ and forecast $\tilde{\beta}$;
3  $S \leftarrow 0$; STOP $\leftarrow$ FALSE;
4  while $\neg$STOP do
5      receive data $\mathbf{x}_t$ and update synopsis $\mathscr{S}_i(\mathbf{x}_t)$;
6      calculate quantum $\mathbf{r}_t = \mathbf{s}_i - \mathbf{s}_i^t$ and discrepancy $R_t$;
7      update $S \leftarrow S + R_t$;
8      if criterion (16 ) holds true then
9          STOP $\leftarrow$ TRUE /*new epoch*/ ;
10         send $\mathscr{S}_i$, $\forall j \in \mathscr{N}_i$ /*send synopsis*/ ;
11         calculate $\beta$ in (7) w.r.t. $T, \hat{\beta}, \tilde{\beta}$; go to line 1;
12     else
13         monitor $v_t$; add and slide $\mathscr{W}$ /*continue*/;
14         estimate $\hat{\beta}$; forecast $\tilde{\beta}$; go to line 4;
15     end
16 end
```

when $\phi \rightarrow 0$, OSDM achieves optimal synopsis dissemination at early stages of the update epochs. We also introduce the metric δ indicating the *expected magnitude of the synopsis quantum* (average L1 norm discrepancy $\|\mathbf{r}\|_1$) over n runs at optimal times t^*, i.e., $\delta = \frac{1}{n}\sum\|\mathbf{s}^{t_k^*} - \mathbf{s}\|_{k=1}^{n}$. This metric is evaluated at optimal times t^* (at the time updated synopses are disseminated) and denotes the ability of OSDM to 'react' even in limited changes in the updated synopses over the time until before their delivery. To investigate the capability of OSDM to avoid network overloading of the network by minimizing the number of synopses updates (messages), we calculate the expected frequency where OSDM delivers the synopsis at $t^* \in [1,T]$, i.e., $\psi = \frac{1}{n}\sum_{k=1}^{n}\frac{T}{t_k^*}$, with $\psi \in [0,T]$. When $\psi \rightarrow T$, OSDM stops frequently, thus, many messages are transferred in the network increasing communication; with $\psi \rightarrow 0$, synopses are mostly delivered after the expiration T.

2) Real Datasets: We experiment with two real datasets: Dataset D1 [6] contains 3-dim. ($d = 3$) vectors with TCP network performance metrics[3] $\mathbf{x} = [x_1, x_2, x_3]$, with $x_1 =$ throughout (MBit/s), $x_2 =$ size of TCP congestion window (Kbytes), and $x_3 =$ link cost (w.r.t. Babel[4] routing protocol), recorded every 10s by a swarm of ten Unmanned Surface Vehicles (USVs). Each USV, equipped with a Raspberry Pi for local computation, floats on sea surface in the coastal area of Skaramagas, Greece, communicating using TCP. Fig. 4 shows the pdf of throughput and congestion metrics with Coefficient of Variation (CoV) 0.80 and 1.15, resp., indicating highly variance variables (CoV is std deviation to mean ratio $\frac{\sigma}{\mu}$). Congestion is strongly, positively correlated with throughput (Pearson coefficient 0.79), while the link cost has negative moderate correlation with congestion and throughput (-0.56 and -0.43, resp.). Dataset D2 [26] contains 2-dim. ($d = 2$) USVs sensor readings[5] $\mathbf{x} = [x_1, x_2]$ ($x_1 =$ humidity, $x_2 =$ temperature; see Fig. 4) recorded every 10s from a swarm of four USVs floating in Skaramagas. The CoV for temperature and humidity is 0.11 and 0.09, resp. with Pearson coefficient -0.41, indicating negative moderate correlated variables with relatively low variance. The throughput is used as the network performance metric v whose pdf is shown in Fig. 4(upper-left) for estimating $\hat{\beta}$ and forecasting $\tilde{\beta}$. Finally, $\alpha = (45, 34)$ in D1 and D2, resp., corresponding to 79% and 85% of datasets size.

3) Models Under Comparison: We provide a comparative assessment of OSDM with the following approaches: (i) **Random Decision Model (RDM)**, which takes a decision when the accumulated synopses quanta are over a pre-defined threshold and is unaware of network performance. RDM is adopted from the random scheduling model in [13]; (ii) **Prediction-based Model (PM)** [34], which proceeds with a decision when the forecast of future network performance metric violates threshold θ. We use our estimate $\tilde{\beta}$ for a fair comparison with PM w.r.t. threshold θ; (iii) **SPY Model** [36], which takes decisions based on the optimal stopping time as a percentage of the average stopping times over the past.

We take average values of the metrics for all models over $n = 1000$ runs, $T = 1000$ horizon with SMAPE ranged $\varepsilon \in [0.2, 0.8]$. At each run, we construct synopses, calculate update quanta and monitor the network throughput.

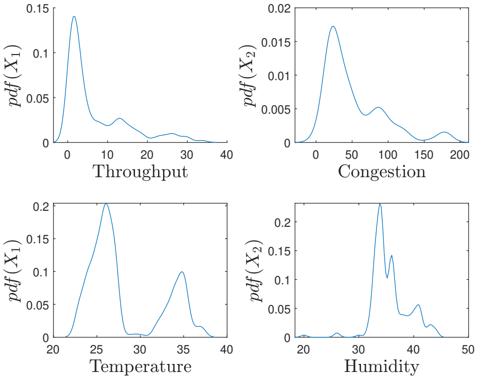

Fig. 4: Probability densities for (upper) throughput and congestion (D1); (lower) temperature and humidity (D2).

B. Performance Assessment

Fig. 5 shows how odds factor γ depends on the estimated and forecast probabilities $\hat{\beta}$ and $\tilde{\beta}$ resp., conditioned on the lowest and highest error $\varepsilon \in \{0.2, 0.8\}$. Factor γ turns higher with low ε yielding OSDM to postpone synopsis dissemination in light of minimizing communication. Also, γ is significantly affected by $\tilde{\beta}$ instead of $\hat{\beta}$ reflecting the impact of forecasting to decision making. High $\tilde{\beta}$ leads to a low γ no matter $\hat{\beta}$ & ε; while the higher γ becomes, the higher the expected synopsis quantum affects the decision. With high γ, OSDM accumulates more quanta before disseminating the synopsis. We conclude that increased estimated probability to violate network performance metric 'alleviates' OSDM to take advanced decisions. When forecasting $\tilde{\beta}$ is unreliable evidenced by high $\varepsilon = 0.8$ (Fig. 5(right)) especially due to unpredictable network performance, OSDM avoids postponing synopsis delivery (reflected

[3] https://mm.aueb.gr/unsurpassed/
[4] https://tools.ietf.org/html/rfc6126
[5] https://sites.google.com/view/gnfuv

by decreasing odds factor γ; evidenced in Fig. 5)) as the network performance deteriorates (small $\hat{\beta}$). There is no reason OSDM to wait for better network status in the future and rely on this decision, since any prediction on the foreseeable network status is not reliable. Thus, in light of avoiding the risk of increasing arbitrarily the discrepancy between synopses, OSDM decides to disseminate synopses at an early epoch stage, at the expense of increased communication.

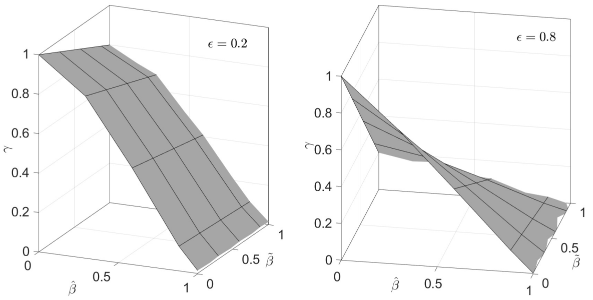

Fig. 5: Impact of estimated and forecast probabilities on decision factor γ conditioned to min & max SMAPE ε.

Fig. 6 (left) shows the feasible space for estimated $\hat{\beta}$ against forecast $\tilde{\beta}$ and error ε interpolating pairs (see Corollary 2). OSDM starts off the process when $\hat{\beta}$ is high achieved by high $\tilde{\beta}$ & ε. This is the 'normal' process as OSDM concludes that there is an increased risk for a 'bad' network, thus, adapting by postponing the process (i.e., high $\hat{\beta}$). OSDM decides to accumulate more quanta before the final decision takes place. The accumulated quanta are proportionally affected by the expected synopsis difference $\mathbb{E}[\|\mathbf{r}\|_p]$ and odds factor γ as evidenced in Fig. 6 (right); here we take the L1 norm ($p = 1$). OSDM is heavily affected by the probability of dealing with a limited network performance: OSDM is adjusted by decreasing the synopsis dissemination frequency while waiting to accumulate more quanta. Instead, when expected synopsis difference and γ are low, OSDM increases the frequency accordingly to keep peers up-to-date.

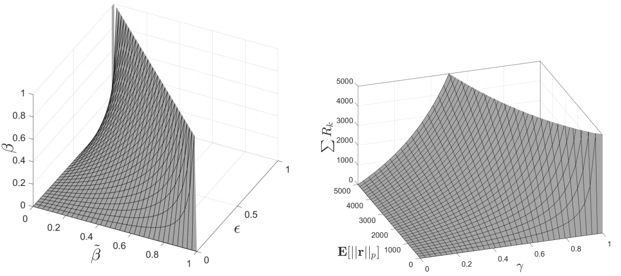

Fig. 6: (Left) Maximum $\hat{\beta}$ value for starting off a synopsis dissemination process; (right) accumulated sum of quanta triggering a dissemination decision.

C. Comparative Assessment

Fig. 7 shows the densities $pdf(t^*)$ of synopsis update times t^* for all models: OSDM, RDM, PM and SPY, having low mean estimated and forecast probabilities $\mathbb{E}[\hat{\beta}] \approx \mathbb{E}[\tilde{\beta}] \approx 0.20$. This reflects a 'good' performance network. All models assume similar dissemination frequency except for PM achieving mostly low t^* times (high frequency); PM is expected to flood the network. Fig. 8 shows $pdf(t^*)$ densities having a 'bad' performance network with $\mathbb{E}[\hat{\beta}] \approx \mathbb{E}[\tilde{\beta}] \approx 0.70$. Here, OSDM's update times range in (1,10) and (1,25) in D1 and D2, resp. This confirms our observations related to the intention of OSDM to accumulate update quanta while postponing the decision due to bad network performance.

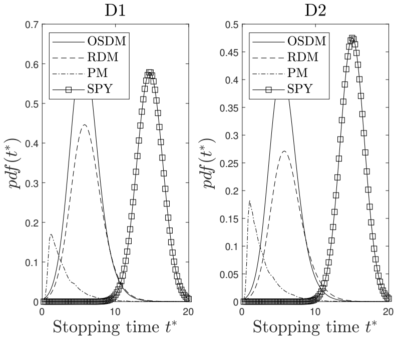

Fig. 7: Probability densities of t^* for OSDM, RDM, PM, SPY with $\mathbb{E}[\hat{\beta}] \approx \mathbb{E}[\tilde{\beta}] \approx 0.20$ ('good' network performance).

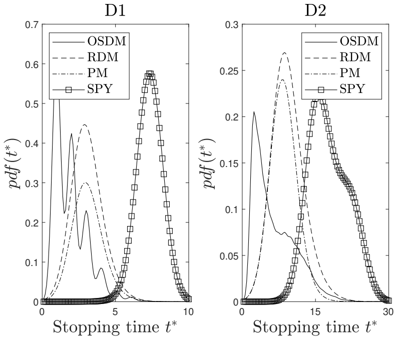

Fig. 8: Probability densities of t^* for OSDM, RDM, PM, SPY with $\mathbb{E}[\hat{\beta}] \approx \mathbb{E}[\tilde{\beta}] \approx 0.70$ ('bad' network performance).

Figures 9 & 10 show the expected percentage of delay in decision ϕ over good and bad network performance. The time percentage spent for the decision process varies across models. In a bad performance network ($\mathbb{E}[\tilde{\beta}] = 0.7$, OSDM postpones the final decision in high hopes of successfully disseminating synopsis updates. While, in a good performance network, OSDM synopsis delivery is achieved after the accumulation of a low difference to keep peer ENs up-to-date. Notably, D2 results in higher ϕ compared to D1 mainly due to the lower data variability (recall the CoV data values in both D1 and D2). Hence, OSDM delays more to accumulate a significant synopsis difference and then distributes it to the network.

Figures 11 & 12 show the impact of estimated and forecast probability on the expected synopsis magnitude δ (anticipated synopsis discrepancy at the time of stopping). One can observe that δ is below 50 for all models and in the majority of $\hat{\beta}$. While, this observation does not stand true for OSDM when $\hat{\beta} \to 1$. In that case, OSDM 'foresees' a bad quality of the

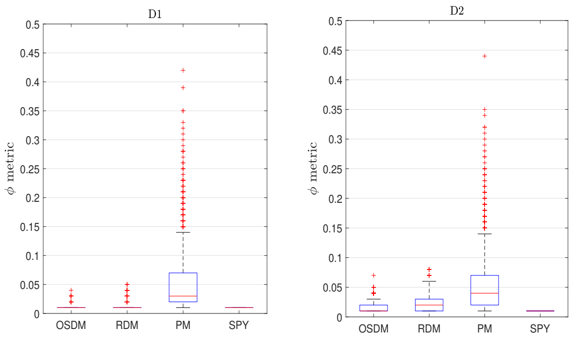

Fig. 9: Expected percentage of optimal decision time ϕ over 'good' network performance ($\mathbb{E}[\hat{\beta}] \approx \mathbb{E}[\tilde{\beta}] \approx 0.2$); for D1, D2.

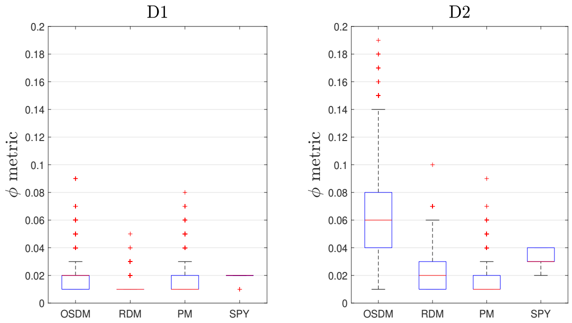

Fig. 10: Expected percentage of optimal decision time ϕ over 'good' network performance ($\mathbb{E}[\hat{\beta}] \approx \mathbb{E}[\tilde{\beta}] \approx 0.7$); for D1, D2.

network and decides to accumulate more quanta postponing the stopping decision to avoid the risk of unsuccessful delivery of synopses. This holds true for both D1 and D2, even if with D2 OSDM obtains lower δ than D1; we obtain similar results in Fig. 12. OSDM scalably manages to achieve average magnitude below 20 for both D1 and D2 no matter $\tilde{\beta}$, while does not lead to very high δ even when $\tilde{\beta} \rightarrow 1$. This indicates that OSDM heavily relies on past observations of network performance instead of future estimations.

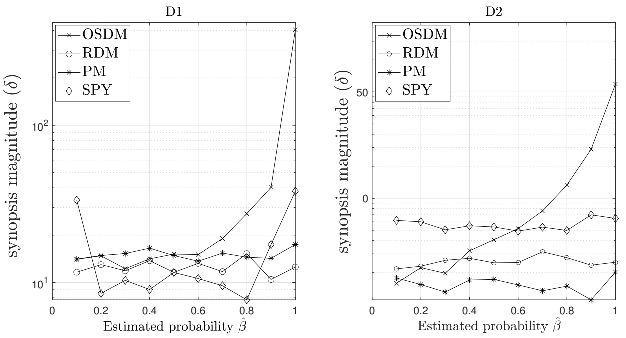

Fig. 11: Expected synopsis magnitude δ (log-scaled) vs. estimated $\hat{\beta}$ for all models in D1 and D2.

Finally, we experiment with the expected communication coping with the trade-off quality of disseminated synopsis and communication under monitoring the network performance. Fig. 13 shows how the communication (# synopsis update messages) is affected by the stopping time which, in turn, affected by the accumulated quanta and network performance for all models. As OSDM's forecast of bad performance network increases, the expected communication decreases. OSDM is fully adapted to the network context by reducing the messages to avoid potential bottlenecks that will jeopardize the delivery of synopses. In case where the possibility of having a bad network performance is low, OSDM increases message rate being aligned with the magnitude of consecutive synopses discrepancy. OSDM assumes slightly higher communication in D1 than in D2 as the underlying data exhibit higher variability. Hence, the discrepancy between subsequent synopses are reduced leading to an increased rate of 'firing' a stopping decision. Evidently, the other models do not adjust their delivery rate since they fail to monitor the network performance and take the network context into consideration. This yields OSDM appropriate for EC environments monitoring and forecasting network performance in optimally deciding on disseminating pieces of knowledge among ENs thus supporting distributed predictive analytics and data management tasks.

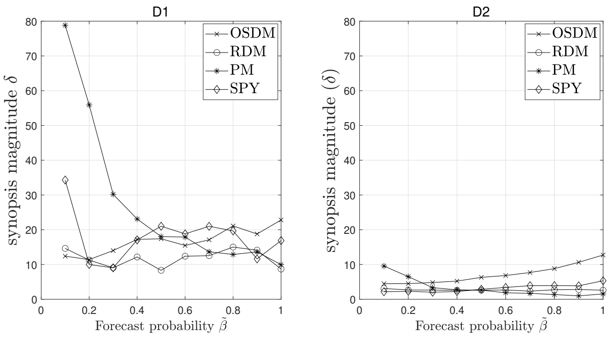

Fig. 12: Expected synopsis magnitude δ vs. forecast $\tilde{\beta}$ for all models over D1 and D2.

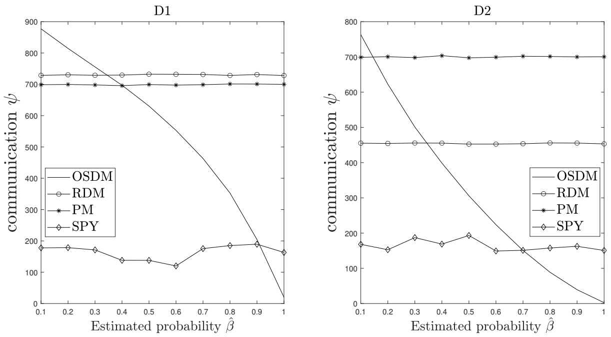

Fig. 13: Communication load (#messages of synopsis updates) vs. estimated $\hat{\beta}$ for D1, D2.

VIII. Conclusions & Future Work

The fundamental data management decision ENs should deal with is successfully dissemination of data synopses to facilitate their peers to decide where to allocate tasks (e.g., predictive analytics) and processes (e.g., data analysis) in EC environments. We focus on in-network context-aware delivery of data synopses to peers under two goals: (i) deliver data summaries when significant updates are experienced and (ii) avoid unnecessary network communication. We introduce a time-optimized context-aware mechanism adopting the principles of the Optimal Stopping Theory and present the relevant formulations and theorems. Nodes monitor synopses changes and network context and optimally decide when to diffuse knowledge. Our experiments showcase that the proposed mechanism efficiently fulfils the objectives evidenced by comprehensive performance and comparative assessment with other models.

Our research agenda includes an uncertainty management mechanism coping with inherent uncertainty of network performance prediction e.g., forecasting network latency. We also plan to develop lightweight multivariate decision making capturing not only individual node's context, but also the context of its interconnected neighbors.

Acknowledge

Funders: UK/EPSRC #EP/R018634/1, EU/H2020 #645220.

References

[1] Aggarwal, C., Han, J., Wang J., Yu, P., 'A Framework for Clustering Evolving Data Streams', VLDB, 2003.

[2] Aggarwal, C., Han, J., Wang, J., Yu, P., 'On-Demand Classification of Data Streams', ACM KDD Conference, 2004.

[3] Aggarwal, C., Yu, P., 'A Survey of Synopsis Construction in Data Streams', ch. in 'Data Streams, Models and Algorithms', ed. Aggarwal, C., Springer Science & Business Media, 2007.

[4] Dimitri P. Bertsekas, 'Dynamic Programming and Optimal Control', Athena Scientific, 2005.

[5] Anagnostopoulos, C. 'Edge-centric inferential modeling & analytics, J. Network and Computer Applications, Elsevier, Aug 2020, 164: 102696.

[6] Aliaj, E., et al. 'A platform for wireless maritime networking experimentation', GIIS, 2018.

[7] Amini, A., Wah, T. Y., 'Density Micro-Clustering Algorithms on Data Streams: A Review', Intl. Conf. Engineers & Computer Scientists, 2011.

[8] Atiya, A. et al. 'Packet Loss Rate Prediction Using the Sparse Basis Prediction Model', IEEE Trans. Neural Networks, 18(3):950–954, 2007.

[9] Babcock, B., Babu, S., Datar, M., Motwani, R., Widom, J., 'Models and issues in data stream systems', PODS, 2002.

[10] Backhouse, A., Gu, I. Y. H., 'A Bayesian Framework-based and-to-end Packet Loss Prediction in IP Networks', IEEE ISMSE, 2004.

[11] Baszczyńska, A., 'Kernel Estimation of Cumulative Distribution Function of a Random Variable with Bounded Support', Stats Trans. 17(3):541–556, 2016.

[12] Bhardwaj, K., Agrawal, P., Gavrilovska, A., Schwan, K., 'AppSachet: Distributed App Delivery from the Edge Cloud', 7th Intl. Conf. Mobile Computing, Applications, and Services, 2015, pp. 89–106.

[13] Breitbach, M., Schafer, D., Edinger, J., Becker, C., 'Context-Aware Data and Task Placement in Edge Computing Environments', IEEE PerCom, 2019.

[14] Chakrabarti K., Garofalakis M., Rastogi R., Shim, K., 'Approximate Query Processing with Wavelets', VLDB J, 10(2-3):199–223, 2001.

[15] Chakravarty, S., Stavrou, A., and Keromytis, A. D., 'LinkWidth: A Method to Measure Link Capacity and Available Bandwidth using Single-End Probes', TR-CUCS-002-08, Columbia University 2008.

[16] Cherrueau, R. A., Lebre, A., Pertin, D., Wuhib, F., Soares, J., 'Edge Computing Resource Management System: a Critical Building Block! Initiating the debate via OpenStack', USENIX 2018, pp. 1–6.

[17] Chow, Y. S., Robbins, H., Siegmund, D., 'Great Expectations: The theory of optimal stopping', Houghton Mifflin, Boston, 1971.

[18] Cormode, G., 'Summary Data Structures for Massive Data', In: Bonizzoni P., Brattka V., Löwe B. (eds) The Nature of Computation, Logic, Algorithms, Applications, vol 7921, Springer, 2013.

[19] Cormode G., Muthukrishnan, S., 'What's hot and what's not: Tracking most frequent items dynamically', ACM PODS, 2003.

[20] Curino, C., Jones, E., Zhang, Y., Madden, S., 'Schism: A workload-driven approach to database replication and partitioning', VLDB Endowment, 3(1-2):48–57, 2010.

[21] D' Adria, F. et al., 'Data Movement in the Internet of Things Domain', Service Oriented and Cloud Computing, Springer, 2015.

[22] Darling, D. A., Liggett, T., Taylor, H. M., 'Optimal stopping for partial sums', The Annals of Mathematical Statistics 43, 4 (1972), 1363–1368.

[23] Das, S., Agrawal, D., Abbadi, A. E., 'Elastras: An elastic transactional data store in the cloud', USENIX, 2, 2009.

[24] Dobra A., Garofalakis M. N., Gehrke J., Rastogi, R., 'Sketch-Based Multi-query Processing over Data Streams', EDBT 2004.

[25] Hanemann, A., Liakopoulos, A., Molina, M., Swany, D. M., 'A study on network performance metrics and their composition', Campus-Wide Information Systems, 23(4):268–282, 2006.

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

[27] Kamal, J., Murshed, M., Buyya, R., 'Workload- aware incremental repartitioning of shared-nothing distributed databases for scalable OLTP applications', Future Generation Computer Systems, 2016, pp. 421–435.

[28] Kolomvatsos, K., 'A Distributed, Proactive Intelligent Scheme for Securing Quality in Large Scale Data Processing', Computing, 2019.

[29] Kolomvatsos, K., 'An Intelligent, Uncertainty Driven Management Scheme for Software Updates in Pervasive IoT Applications', Future Generation Computer Systems, 83:116–131, 2018.

[30] Kolomvatsos, K., Anagnostopoulos, A., 'Multi-criteria Optimal Task Allocation at the Edge', Future Generation Computer Systems, 93:358–372, 2019.

[31] Lakshmi, K. P., Reddy, C. R. K., 'A Survey on Different Trends in Data Streams', IEEE Intl Conf Netw. & Inf. Techn., 2010.

[32] Madhyastha, H. V., Anderson, T., Krishnamurthy, A., Spring, N., Benkataramani, A., 'A Structural Approach to Latency Prediction', IMC, 2006.

[33] Manku, G., Motwani, R., 'Approximate Frequency Counts over Data Streams', VLDB, 2002.

[34] Martin, R., Vahdat, A., Culler, D., Anderson, T., 'Effects of Communication Latency, Overhead, and Bandwidth in a Cluster Architecture', 4th Annual International Symposium on Computer Architecture, 1997.

[35] Mirza, M., Springborn, K., Banerjee, S., Barford, P., Blodgett, M., Zhu, X., 'On the Accuracy of TCP Throughput Prediction for Opportunistic Wireless Networks', 6th Annual Conference on Sensors, Mesh and Ad Hoc Communications and Networks, 2009.

[36] Harth, N., Anagnostopoulos, C. 'Quality-aware aggregation & predictive analytics at the edge', 2017 IEEE Big Data 2017, pp. 17–26.

[37] Najam, S., et al, 'The Role of Edge Computing in Internet of Things', IEEE Communications Magazine, 2018.

[38] Peskir, G., Shiryaev, A., 'Optimal Stopping and Free Boundary Problems', ETH Zuerich, Birkhauser, 2006.

[39] Phansalkar, S., Dani, A., 'Transaction Aware Vertical Partitioning Of Database (TAVPD) for Responsive OLTP Applications in Cloud Data Stores', Theoretical & Applied Information Techn., 59(1):73–81, 2014.

[40] Phansalkar, S., Ahirrao, S., 'Survey of Data Partitioning Algorithms for Big Data Stores', 4th International Conference on Parallel, Distributed and Grid Computing, 2016.

[41] Sardellitti, S., Scutari, G., Barbarossa, S., 'Joint Optimisation of Radio and Computational Resources for Multicell Mobile-Edge Computing', IEEE Trans. Sign. Inf. Proc. over Networks, 1(2):89-–103, 2015.

[42] Simoens, P., Xiao, Y., Pillai, P., Chen, Z., Ha, K., Satyanarayanan, 'Scalable crowd-sourcing of video from mobile devices', 11th Intl Conf. Mobile systems, applications, and services, 2013, 139–152.

[43] Sivakumar, K., Vivekanandan, K., Devi, T., Ramasubramanian, T., 'Switch Bandwidth Congestion Prediction in Cloud Environment', Procedia Computer Science, 2015, 235–243.

[44] Taieb, S. B., Bontempi, G., Atiya, A. F., Sorjamaa, A., 'A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition', Expert Systems with Applications, 39:7067–7083, 2012.

[45] Tatbul, N., Zdonik, S., 'A subset-based load shedding approach for aggregation queries over data streams', VLDB 2006.

[46] Wang, N., et al, 'ENORM: A Framework for Edge Node Resource Management', IEEE Trans. Service Computing, 2017.

[47] Yao, Y., Cao, Q., Vasilakos, A. V., 'EDAL: An Energy-Efficient,Delay-Aware, and Lifetime-Balancing Data Collection Protocol for Wireless Sensor Networks', IEEE MASS 2013, 182–190.

[48] Yao, J., Kanhere, S., Hassan, M., 'An Empirical Study of Bandwidth Predictability in Mobile Computing', WinTech, 2008.

[49] Alghamdi, I., Anagnostopoulos, C. Pezaros, P. 'On the Optimality of Task Offloading in Mobile Edge Computing Environments', IEEE GLOBCOM 2019, pp:1–6.

[50] Zhou, A. et al. 'Optimal Mobile Device Selection for Mobile Cloud Service Providing', J. Supercomputing, 72(8):3222–3235, 2016.