COSINE: Collaborator Selector for Cooperative Multi-Device Sensing and Computing

Huber Flores
University of Tartu
huber.flores@ut.ee

Agustin Zuniga, Farbod Faghihi,
Xin Li, Samuli Hemminki,
Sasu Tarkoma
University of Helsinki
fistname.lastname@helsinki.fi

Pan Hui
HKUST
University of Helsinki
panhui@cse.ust.hk

Petteri Nurmi
University of Helsinki
petteri.nurmi@helsinki.fi

Abstract—Pervasive availability of programmable smart devices is giving rise to sensing and computing scenarios that involve collaboration between multiple devices. Maximizing the benefits of collaboration requires careful selection of devices with whom to collaborate as otherwise collaboration may be interrupted prematurely or be sub-optimal for the characteristics of the task at hand. Existing research on collaborative scenarios has mostly focused on providing mechanisms that can establish and harness collaboration, without considering how to maximally benefit from it. In this paper, we contribute by developing COSINE as a novel approach for selecting collaborators in multi-device computing scenarios. COSINE identifies and recommends collaborators based on a novel information theoretic measure based on Markov trajectory entropy. Rigorous experimental benchmarks carried out using a large-scale dataset of device-to-device encounters demonstrate that COSINE can significantly improve collaboration benefits compared to current state-of-the-art solutions, increasing expected duration of collaboration and reducing variability of collaborations.

Index Terms—collaborative sensing, collaborative computing, mobile computing, pervasive computing, multi-device systems, device-to-device, opportunistic collaborations

I. INTRODUCTION

Pervasive availability of programmable smart devices is giving rise to computing scenarios that involve *collaboration* between multiple users or devices. For example, collaborative sensing scenarios have examined how energy and other resource costs can be saved by sharing costs among neighboring devices [1]–[3] or how the capabilities of individual devices can be augmented through collaboration [4]–[6]. As another example, opportunistic computing scenarios have explored the potential of using collaboration between proximal devices to support computing tasks [7]–[9]. Finding suitable collaborators is critical for the success of these scenarios as the costs of finding and managing collaboration may otherwise offset the benefits it brings [1]. This is especially the case when collaboration has a short duration or is fragmented in nature.

Finding optimal collaborators can be highly difficult due to the fact that the devices that are available for collaboration at any given time are influenced by human mobility – regardless of whether the devices are part of the environment or carried by other people [1], [10], [11]. Another complicating factor is that the best collaborators depend on the context where collaboration takes place and characteristics of the task at hand. For example, at work or at home the best collaborators may be those that have close social relationships with us, but during our commute the best collaborators may be people that are strangers that we encounter in that particular setting (so-called familiar strangers [12]). Existing research on collaborative multi-device sensing and computing has largely focused on supporting collaboration, without looking at how to find the *best possible collaborators* for the current context. For example, several works have explored how to minimize energy and resource overhead of *finding candidates* without looking at how to select the best ones amongst them [13]–[17]. Other works have explored how to provide middleware support for specific kinds of collaboration scenarios without addressing how the collaborators are chosen for these tasks [1], [2], [4], [18]. To maximize benefits of collaboration, the characteristics of collaboration and the context where it takes place need to be considered. Indeed, as we show in Section II, the selection of collaborators can be highly sub-optimal unless characteristics of collaboration are taken into account.

In this paper, we contribute by developing COSINE as a *collaborator selector service* for cooperative multi-device sensing and computing scenarios. COSINE overcomes the main limitations of current solutions through a novel information theoretic approach, where potential collaborators are ranked according to their *Markov trajectory entropy* [19]. The use of Markov trajectory entropy enables our approach to adapt collaborator selection to different types of collaborations or task characteristics, and to account for individual variations in mobility. We demonstrate the benefits of COSINE through rigorous benchmark experiments conducted on a large-scale dataset of device encounters that contains different mobility patterns, and collaboration contexts (locations). Our results demonstrate that COSINE *consistently identifies* collaboration opportunities with a high duration. The median duration of collaboration opportunities for COSINE is 22 minutes. Equally importantly, COSINE produces the most consistent collaborations among all baselines and has least variation to the conditions of collaboration. Accordingly, COSINE not only identifies high quality collaborators, but recommends collaboration opportunities that have the most predictable duration, facilitating scheduling and allocation of tasks across the collaborating devices.

Summary of Contributions:

- **New method** We develop COSINE, a novel regularity-based collaborator selector for cooperative multi-device sensing and computing tasks.
- **New insights** We demonstrate that current collaborator selection algorithms are sensitive to the location where collaboration takes place and characteristics of the task where collaboration is required, motivating the need for solutions that can work consistently and robustly across wide range of collaboration contexts.
- **Improved Performance** We perform rigorous benchmarks to demonstrate that COSINE significantly improves benefits of collaboration. Compared to state-of-the-art, our approach recommends collaborators that are available for longer duration. COSINE is also more robust in identifying collaboration opportunities, providing consistent and predictable recommendations across different locations and users.

II. MOTIVATION

We begin by demonstrating the need for improved collaborator selection by showing how current state-of-the-art solutions are sensitive to characteristics of user mobility and the context where collaboration takes place. As representative example of state-of-the-art approaches we consider the familiarity-based collaborator selector algorithm integrated into the CoMon collaborative context monitoring platform [1]. As part of our main experiments, in Section V, we consider a wider range of baselines and demonstrate that the same findings hold also for them. The dataset we consider is described in Section IV-A and the familiarity-based collaborator selection algorithm is described in Section IV-B.

Figure 1 shows the performance of the familiarity-based collaborator selection algorithm for four different locations in our primary dataset. From the figure, we can observe that characteristics of encounters contain significant variation across the locations. For example, at railway station encounters are generally short, whereas in entertainment areas encounters last longer on average but also contain more variation. We can observe the performance of the familiarity-based collaborator selector to be highly sensitive to characteristics of the location and tending to result in the algorithm identifying short collaboration opportunities. Also, the performance is sensitive to the number of encounters that devices need to have in common. While higher threshold tends to result in longer collaboration opportunities, finding suitable people for a given threshold is not always possible, e.g., in the park or railway it is difficult to find devices that have encounter frequency of 75.

The performance analysis highlights how state-of-the-art collaborator selection algorithms are sensitive to the characteristics and context of collaboration. This makes it difficult to find parameter values for these algorithms so that the performance is robust in different locations and collaboration contexts. Our analysis also highlights how the characteristics of collaboration are likely to vary considerably during the day according to human mobility, hence solutions that can work robustly across different contexts are required.

III. COSINE COLLABORATOR SELECTOR

COSINE is a collaborator selector service that uses a novel information theoretic approach to identify and recommend the best possible collaborators based on their regularity characteristics. As measure of regularity, we use Markov trajectory entropy [19], which allows COSINE to overcome the main limitations of existing collaboration selection approaches: (i) incapability to adapt to varying mobility patterns; and (ii) incapability to account for different task and collaboration characteristics. Indeed, as shown in the previous section, existing approaches to collaboration selection are highly sensitive to the characteristics of collaboration and mobility patterns, working reasonably in some situations but failing completely in others. Figure 2 presents an overview of COSINE, which consists of three phases: (i) signal quantization, (ii) regularity extraction and (iii) selection of collaborators.

A. Measurements and Signal Quantization

COSINE operates on measurements consisting of timestamped device encounters. In the first phase, measurements are aggregated continuously into a signal, such that its regularity can be analyzed. Formally, let $\ddot{d} = \{d_{x,y} | 0 < d_{x,y}; x, y \in D\}$ denote the set of possible duration values for encounters where $d_{x,y}$ is an encounter between device x and y, and D denotes the possible devices that can be encountered. We quantize the duration values into one of finite number of states $S = \{s_m \in S : 1 \le m \le M\}$. Modeling the signal using a fixed number of states S allows us to consider it as a discrete trajectory and to quantify regularity through transitions between states of the trajectory; see Figure 2a. Naturally, the transformation of $\ddot{d}$ into its quantized counterpart $\ddot{d}'$ results in some information loss as the signal is represented with fewer number of states. However, as long as the relative properties of the discrete signal are preserved when compared to the original, this loss of information does not have any significant impact for the extraction of regularity. COSINE also supports augmenting the measurements with additional information that can be used to condition the duration values. For example, in our experiments, we consider measurements of the form $a = (x, y, t, l, d_{x,y})$ where t and l correspond to the time and location of the encounter. These variables can be used to condition the duration values and to learn models that are specific to the time-of-day or spatial region where the encounter takes place. In our experiments we perform quantization using k-means clustering where the number of clusters k equals the desired number of quantization levels.

B. Regularity Extraction

In the second phase, depicted in Figure 2b, we quantify regularity of encounters. Intuitively, our formulation captures the consistency of encounters and helps to identify device pairs that have predictable durations. We perform the quantification by considering the measurements in $\ddot{d}'$ as a realization of a Markov chain. This can be understood as a model where the encounter frequency values are interpreted as a random walk in a finite graph. The vertices of the graph correspond to the
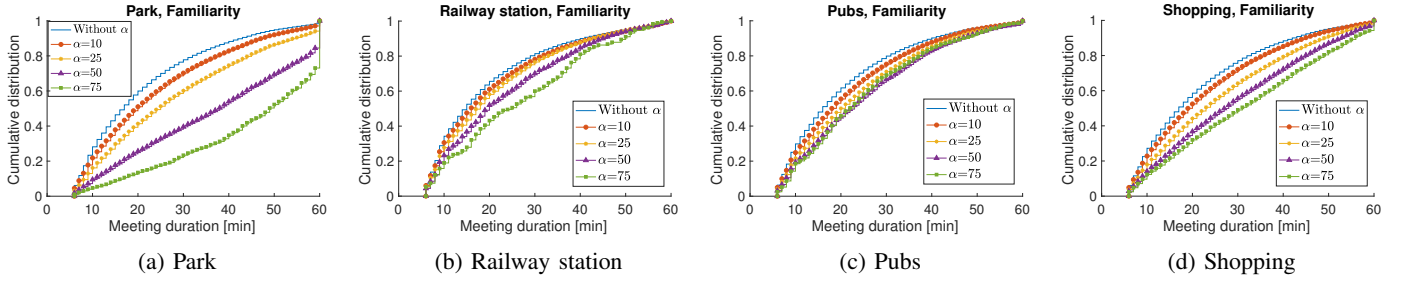
(a) Park   (b) Railway station   (c) Pubs   (d) Shopping

Fig. 1: Selection of collaborators with CoMon in different contexts (a) – (d).

(a) Quantization of measurements   (b) Extraction of regularity   (c) Selection of collaborators
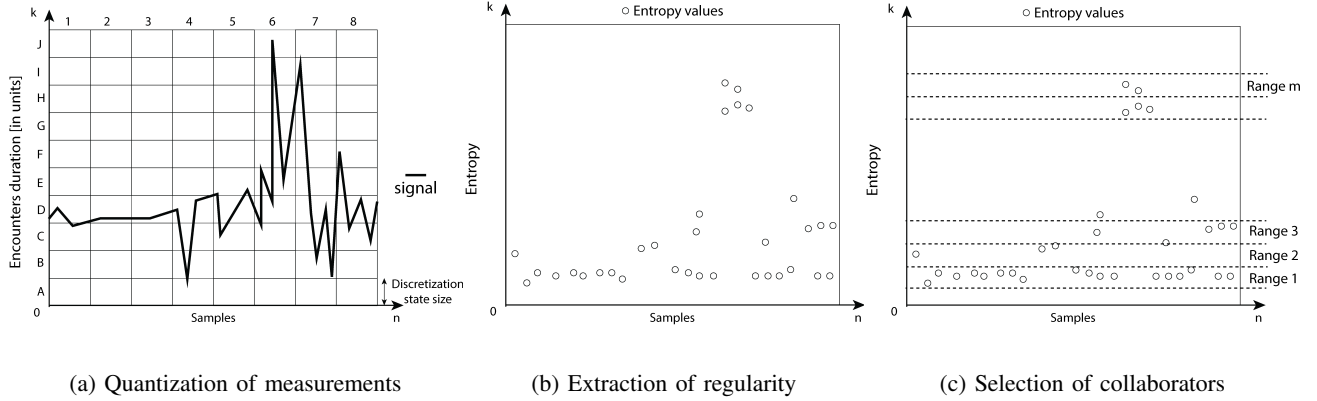
Fig. 2: Overview of the COSINE approach illustrating the three main phases (a) – (c) of finding suitable collaborator devices.

different devices, and the edges to encounter durations. With this formulation, the regularity of each encounter pair can be quantified by constructing a Markov trajectory entropy matrix H where each element $h_{x,y}$ corresponds to the entropy of all encounter durations between devices x and y; see Figure 2b.

To calculate the Markov trajectory entropy matrix H, we first form a transition probability matrix P. Each element $p_{i,j} \in P$ corresponds to the probability of observing a transition from state i to state j, where the states correspond to quantized duration values (i.e., $\ddot{d}'_{x,y}$). To estimate P, we simply calculate the ratio of transitions between cells i and j compared to all transitions from state i. To ensure the resulting chain is irreducible, we also smooth the probabilities by a small constant ϵ. Smoothing also provides robustness against missing data as it allows the chain to reach cells that have no observations. As we interpolate the measurements to have consistent spatial and temporal resolution, we need to apply the smoothing only to grid cells that are adjacent. Once the transition matrix has been formed, the matrix H of Markov entropies can be calculated using the following formula [19]:

$$H = K - \tilde{K} + H_\Delta. \tag{1}$$

Here K is the matrix:

$$K = (I - P + A)^{-1}(H^* - H_\Delta) \tag{2}$$

where I is the identity matrix, P is the transition probability matrix, and A is a matrix containing the stationary distribution μ of the Markov chain, i.e., $a_{i,j}1 = \mu_j$. The matrix H^* corresponds to the single-step state entropies of the measurement sequences originating from state i and is given by $H^*_{i,j} = H(P_{i.}) = -\Sigma_k P_{i,k} log P_{i,k}$. The matrix H_Δ is a diagonal matrix where each non-zero element corresponds to the entropy rate of state i:

$$(H_\Delta)_{i,i} = \frac{H(\chi)}{\mu_{i.}} = \frac{-\Sigma_{i,j}\mu_i P_{i,j} log P_{i,j}}{\mu_{i.}}, \tag{3}$$

i.e., the matrix H_Δ captures the average contribution of each state to the uncertainty associated with sequences through i. Finally, matrix $\tilde{K}$ corresponds to a matrix where the ijth element is equal to the element $k_{j,j}$ in matrix K.

C. Selection of Collaborators

In the third and final phase, we derive entropy ranges with upper and lower bounds that depict grouping of entropy values; see Figure 2c for an illustration. We rank all entropy ranges based on cardinality, and select the most suitable candidates according to the frequency of their entropy range, i.e., the ones with most regular encounters. From these available candidates, we select the one that has the *highest expected proximity duration*. By using the expected duration, our approach can determine collaborators that are *consistently* available for a long duration instead of recommending users that have high variation in their availability times. A collaborator is selected within the entropy range by analyzing its proximity around the mean of the range.

D. Implementation

We have implemented two prototypes of COSINE, one performing the necessary computations on a server using a combination of Python and R scripts, and one running on smartphones (tested with Samsung S6 smartphones). In the former, the regularity model that is used to assess suitability of collaborators is calculated on a server. On the server we use R. Quantization is performed using the package *arules* and the Markov transition matrix is formed using *markovchain*. Regularity analysis is implemented in Python using *Numpy* and called through the main interface of our program using *rPython*. The overall approach contains ≈350 LoC. In the mobile version, we have reduced external dependencies and implemented simplified versions of the necessary routines. Our mobile prototype runs as a background service that provides a query interface to applications to assess the candidates for collaboration in a particular location. Collaborators data is stored in a SQLite database on the device. We use a default number of quantization levels and incrementally learn a Markov transition matrix from encounters for the default level. Once sufficient amounts of encounters have been collected, we run k-means on the phone to estimate the optimal quantization level. If this differs by more than 2 levels, we fetch all the data stored in the database and perform a one-off operation where the transition matrix is recalculated for the given quantization level. Any new encounters are used to update the matrix incrementally, similarly to the default model. We have released our code as open source in GitHub https://github.com/mobile-cloud-computing/Regularity.

IV. EXPERIMENTAL SETUP

We next describe the primary dataset and baselines that we consider in our experiments. The dataset and baselines are considered both in the feasibility evaluation described in Section II and in our main experimental evaluation presented in Section V. In all experiments, we evaluate collaborator selector performance using the duration of discovered collaborations.

A. Datasets

DataSet: As our primary dataset we consider a crowdsensed dataset that contains real world mobility traces and app usage patterns. The dataset is collected from a cellular operator in Shanghai over a one-week period. The use of a large-scale dataset collected through a cellular operator ensures our data is representative of real human mobility, and ensures the collaboration opportunities reflect those that we would expect to encounter in a real-world deployment.

The dataset we use provides coarse-grained information about base stations in the metropolitan area, and session information from users connecting to base stations. Each sample in the dataset describes the connectivity process of a device to the network and includes device identifier, the start and end time of a session, the amount of data transferred during a session (in bytes), the id of the base station that handles the connection, and the GPS coordinates of the base station. Our dataset contains data from $137\,495$ devices and $10\,363$ base stations. For privacy reasons, the identifiers of devices and base stations are all anonymized.

Pre-Processing: Before our analysis, we filter the dataset to consider only measurements where devices are in a proximal communication range through a base station. We refer to this as co-location of devices. Co-located devices are available for collaboration whenever they are connected to the same base station at the same time. By analyzing the duration of sessions of users connected to the same base station, we identify pairs of devices which are connected to the same base station. We then estimate the encounter duration of these devices. We accomplish this, by first calculating the session duration of each device in the dataset and subtracting the starting time from an ending time of a connection sample. We then model the duration distribution using a percentile scale and discard sessions below 10th percentile or above 90th percentile. This is performed to reduce impact of sessions that have a very small duration or that happen very infrequently. As our approach focuses on selecting long stable collaborators, we analyze collaborations that can be harnessed during a time period $\geq$ 5 min, which is the minimum time for a sensing task to beneficial in a collaboration between two devices [1]. We also discard base stations and users with a total amount of measurements below the 10th percentile. We then perform temporal alignment of all sessions per device into hourly bin intervals, and match sessions of devices that share the same interval. We focus our analysis on the 20 km^2 area with higher density of users and the $1,000$ users with higher amount of samples. After this step, the resulting dataset for the analysis contains $1,000,000$ D2D encounters.

Data Representativeness: Previous works on collaboration have relied on encounters detected directly on the mobile devices, e.g., by detecting devices that are within the range of BLE or other short-range networking technology. In our experiments we consider instead an elastic edge provisioning application where devices are available for collaboration whenever they are connected to the same network base station. Accordingly, in our case devices are not necessarily physically co-located in proximal range. To demonstrate that the characteristics of our dataset are similar to previous works, Figure 3 compares the proximity characteristics of our dataset with those in the dataset used by Lee et al. [1]. The distribution of encounters has the same shape in both datasets, but our dataset increases more sharply due to higher availability of short collaborations. This can be expected as our dataset is more recent and reflects the wider availability of smart devices.

B. Baselines

We compare COSINE against three baselines. For all approaches, a parameter α specifies the condition that is used for selecting collaborators. We evaluate different values of α to find the most suitable candidates for collaboration.

Familiarity: As the primary baseline, we consider the familiarity-based collaborator selection algorithm of the CoMon collaborative monitoring platform [1]. Familiarity refers to the frequency of encounters between devices, i.e.,
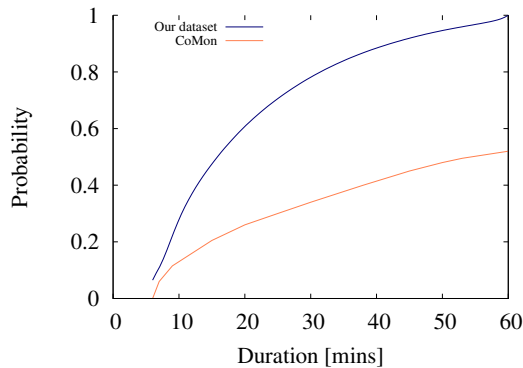
Fig. 3: Data representativity of our dataset when compared with state-of-the-art findings.

α depicts the number of encounters between devices. For example, two devices that encounter each other several times are very familiar whereas two devices that encounter a few times are less familiar.

Permanency: As our second baseline, we develop a collaborator selector that relies on permanency. Permanency refers to the amount of time in which devices remain in proximal range. Here α depicts the collaboration time required for a candidate device. Permanency is important as a successful collaboration must finish before devices are out of range. We use α as threshold to analyze different values of permanency to identify collaborators that can be selected.

Magnitude: Since the benefit of collaborations is to distribute the cost of a task among multiple devices, we develop a collaborator selector that defines the magnitude of the collaboration. In this case, α depicts the number of devices required to distribute a task. We analyze different magnitude values of α to select a group of devices of different sizes.

V. Experimental Evaluation

We evaluate COSINE through rigorous benchmark using the dataset described in Section IV-A. In the evaluation we focus on demonstrating that device encounters have sufficient regularity for being useful source of identifying collaborations, and demonstrating that COSINE can find collaborations that have long duration and that are predictable. Predictability is essentially a measure of robustness and facilitates task allocation and system optimization. Indeed, if collaboration duration fluctuates considerably, scheduling tasks for collaboration becomes difficult. We also compare COSINE against the baselines introduced in Section IV-B.

A. Regularity extraction from candidates

Since the potential of our proposed approach depends on assessing regularity of candidate devices, we begin our evaluation by demonstrating that enough regularity can be extracted from mobility traces depicting encounters with other devices. Regularity is measured by entropy values, with low entropy implying consistent encounter duration over time and high entropy implying highly fluctuating encounter duration. For our analysis, we select three different representative traces from users in our dataset that show high, medium, and low encounter frequency for devices. Through our analysis, we demonstrate that regularity can be used to model the whole spectrum of possible collaborations.

Figure 4 shows the results. From the figure, we can identify different encounters experienced by a user during one week. We can also observe the average duration of each encounter per hour. In addition, the figure also includes the entropy values that can be extracted from the measurements. We can observe that, for all the cases, regularity as entropy values can be extracted to characterize different types of candidates and that the duration of the encounter is characterized into different entropy ranges. An entropy range consists of an upper bound and a lower bound entropy values. For instance, from Figure 4a, we can observe a clear range between $8 - 11$ minutes of entropy characterization. Similarly, for Figure 4c, we can observe a range between $2 - 5$ minutes. This suggests that different duration periods of encounters can be clearly characterized with regularity. We next demonstrate how these periods can be used to optimize collaboration between devices.

B. Selection of collaborators

The previous subsection demonstrated that sufficient regularity can be extracted from mobility traces. We next demonstrate that COSINE is capable of identifying the collaborators with the longest and stablest periods of proximity.

Figure 5 shows the results of the analysis over our dataset. As shown in Figure 5a, we can observe that candidates for collaboration can be characterized and ranked using different entropy ranges. From the figure, it is possible to see up to 9 ranges of characterization and their respective frequencies. While some ranges are more frequent than others, each range depicts the devices that share similar periods of co-located duration. We then use each entropy range to evaluate the performance of selecting collaborators. We refer to each particular entropy range as an α. Figure 5 plots the cumulative distribution function (CDF) for different α values. We can observe that by using regularity it is possible to select candidates for collaboration close to the average duration with low error. Moreover, we can observe that $\alpha = 3$ provides the optimal results for selecting collaborators with the largest duration of co-location. This also can be observe in Figure 5c that quantifies the average duration and number of devices available when selecting collaborators. We can observe that with the optimal selection of devices, we can obtain devices up to 22 min for collaborations. In contrast, when using other α values, we can observe that while the number of devices remains almost the same, the duration of collaborator reduces significantly between 17 and 19 min.

C. Comparison with other approaches

We next compare our regularity-based approach with the baselines described in Section IV-B. Figure 6 plots the CDF of collaboration opportunities. The distribution closely mirrors
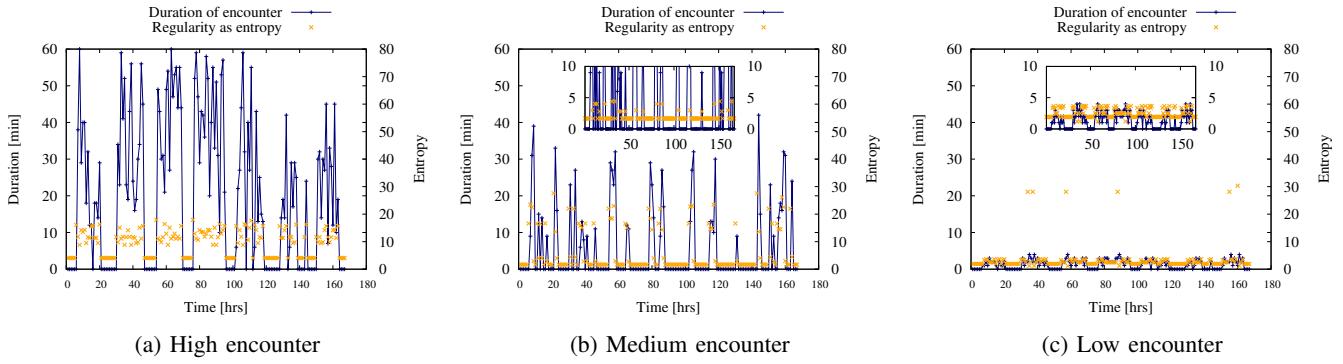
(a) High encounter

(b) Medium encounter

(c) Low encounter

Fig. 4: Regularity extraction from three traces with various frequency of device encounters (a) – (c).

(a) Characterization of candidates
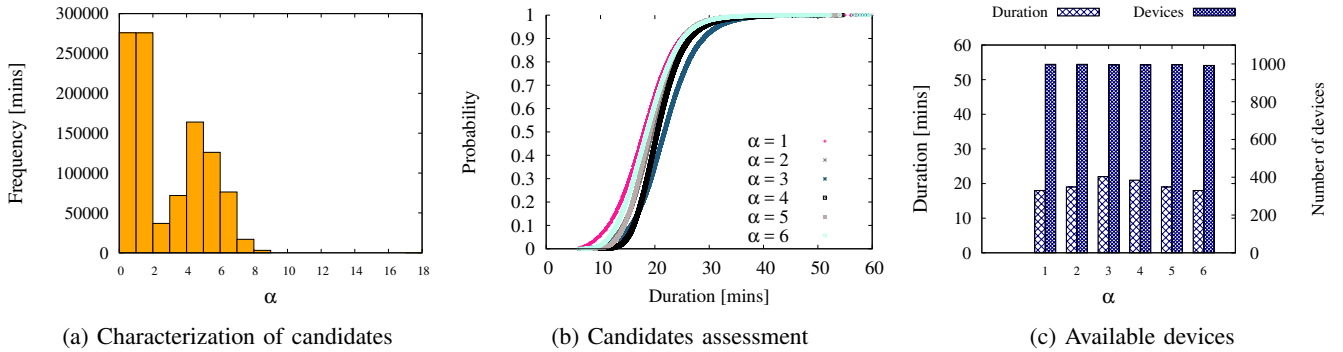
(b) Candidates assessment

(c) Available devices

Fig. 5: Collaborators selection with COSINE (a) – (c).

the results of Lee et al. [1], suggesting that the fundamental principles of collaborations largely have remained the same over time – which is to be expected due to characteristics of human mobility. From Figure 6a, we can observe that the higher the familiarity of devices, the longer the duration of collaboration opportunities usually is. The likelihood of finding devices to collaborate is high when familiarity is low ($\alpha = 0$) but decreases sharply as familiarity requirement is increased (from $\alpha = 25$ to $\alpha = 450$). Figure 6d further characterizes collaboration opportunities with familiarity by looking at how the number of available devices and the duration of collaboration varies when familiarity between devices changes. From the figure we can observe that the characteristics of collaboration opportunities vary significantly depending on requirements placed on collaboration. For example, when $\alpha = 25$, it is possible to observe that the likelihood of the CDF drops to 0.4 (Figure 6a). This suggests from a quantifiable point of view (Figure 6d) that up to 1000 devices can be harnessed during ≈ 20 minutes. However, when familiarity requirements are high ($\alpha = 450$), only 2 devices can be harnessed during 35 minutes with the same probability. Similarly, it is possible to observe same trends with Permanency and Magnitude. For both approaches, we can observe that as α increases, the number of available collaborators decreases From all the baselines, we can observe that Permanency selects the collaborators with the longest periods that are up to 60 min, but the pool of available

collaborators decreases significantly. More importantly, we can observe that for all the baselines, the results are oscillating and with high variance when using different α thresholds. This suggests that the selection of collaborations is unstable and unreliable. This can be also observed from the CDFs, which show that data is very spread out from the average.

Figure 7 compares COSINE against the baselines. For each approach, we select the α that provides the largest number of available devices with the longest duration. For having a fair comparison, we also ensure that the number of available devices does not differ significantly between α values. Thus, we select $alpha$ values that provides estimates for around 1000 devices. Results of the comparison are shown in Table I. From the table, we can observe that regularity selects candidates with up to 22 min, which outperforms the state-of-the-art familiarity approach. While permanency results in higher expected collaboration duration, we can observe that its results fluctuate considerably whereas regularity provides a stable selection of collaborators. To provide more insights into this behaviour, the Mean Absolute Deviation (MAD) for regularity is consistent, reaching 5.66 across all durations. Familiarity and Permanency provide good estimations for collaborations up to 20 and 28, min, however, we can observe higher variance to select collaborators. From the table, we can observe that the MAD for both, Familiarity and Permanency increases by a factor of 2.45X, which makes them unreliable for supporting and identifying long term collaborations.
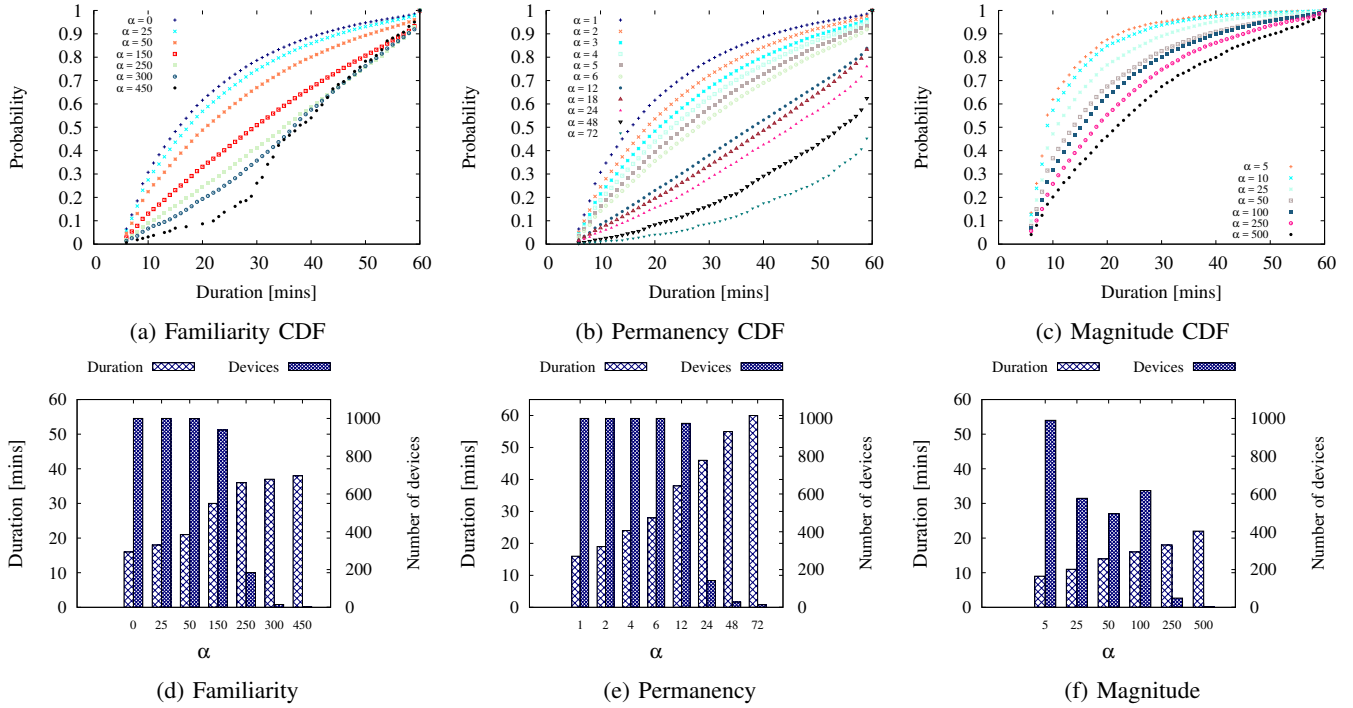
(a) Familiarity CDF

(b) Permanency CDF

(c) Magnitude CDF

(d) Familiarity

(e) Permanency

(f) Magnitude

Fig. 6: Selection of collaborator for the considered baselines for various α-values, along with cumulative distribution functions (CDF) over duration of the collaboration.

Selector mechanism	Average duration (min)	MAD
Regularity	22	5.66
Familiarity	20	13.74
Permanency	28	13.74
Magnitude	9	13.34

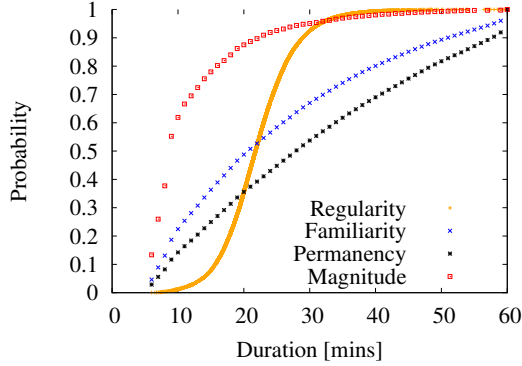TABLE I: Comparison of different approaches.

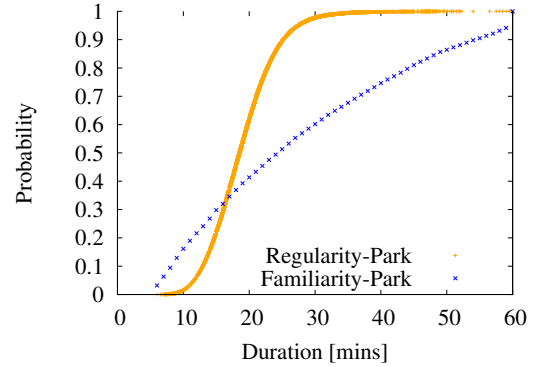Fig. 7: Comparison of approaches to select collaborators

Fig. 8: Selection of collaborators using different mechanisms.

Figure 8 shows the results of the comparison. We can observe that the results of COSINE are consistent even when considering specific locations. From our results, we estimate a MAD of 5.59 for regularity and 13.74 for familiarity. Thus, our approach provides more stable selection around the average duration that is found in a location.

VI. COLLABORATIVE USE CASE

As we demonstrate in previous section, COSINE selects better candidates for long collaborations. In this section, we quantify the amount of additional energy gains that can be obtained with COSINE. As our case study, we rely on a multi-device computing use case based on cross-device app

D. Selection performance in different contexts

We next demonstrate that COSINE can adapt its selection mechanism to different contexts. We also include the results of our familiarity baseline for comparison. We select a park as the location context as it provides the most oscillating mobility patterns from all our selected locations as shown in Section II.

offloading; similar to the one presented in [20]. In app offloading, devices that share proximal range and execute the same app, do not distribute the load of app execution, but instead, they schedule the execution of the app in individual devices, such that a device executes the application at the time while both devices benefit from the collaborative execution. A device saves energy based on the duration time of the collaboration.

A. Experimental Setup

Devices and applications: We consider S5 (i9505) and Nexus (i9250) smartphones in our experiments. We used Monsoon power meter[1] to obtain ground truth energy measurements of the applications. We consider three apps from Google Play store. To generalize our results, apps from different categories, and compatible with our devices were chosen.

Augment[2]: is an augmented reality application where one can manipulate and view virtual objects in the physical environment. We used the app to display a virtual chair for 5 minutes, and repeated the experiment 5 times. For S5, the average time and power were stable, $t = 303.24\ s$ and $p = 3138.92\ mW$, respectively. For Nexus, the average time and power were stable, $t = 306.52\ s$ and $p = 2578.55\ mW$.

Chess[3]: is a game that implements a minimax algorithm to challenge users. We ran the application with a new game each time, keeping the game running for 5 minutes at a time. For S5, the average time and power were stable, $t = 304.11\ s$ and $p = 2067.49\ mW$, respectively. For Nexus, the average time and power were stable, $t = 306.56\ s$ and $p = 2524.39\ mW$.

Face recognition[4]: is an application that allows the device to identify a registered person based on their facial features. We ran the face detection feature of the application for roughly 5 minutes and repeated the experiment 5 times. For S5, the average time and power were stable, $t = 297.55\ s$ and $p = 2509.73\ mW$, respectively. For Nexus, the average time and power were stable, $t = 297.74\ s$ and $p = 2444.11\ mW$.

Results: We estimate the amount of energy that can be gained in different collaborations between the two devices in our setup. We used the 22 min of collaboration estimated from our regularity approach. As our dataset is collected from real world interactions, 22 min is a representative duration that can be found in the wild. Results are shown in Table II (*Regularity*). We also include the energy gains when using familiarity for comparison (*Familiarity*). We can observe that the gains obtained by COSINE account for an additional of 10% of energy improvement across the devices (+*Benefit*). Moreover, if we consider that our approach selects the same average duration regularly for all the selected collaborators, we can account for further aggregated gains.

VII. DISCUSSION

Naturally there is room for further work and improvements. We discuss a few points here.

[1] https://www.msoon.com/

[2] https://play.google.com/store/apps/details?id=com.ar.augment

[3] https://github.com/huberflores/CodeOffloadingChess

[4] https://play.google.com/store/apps/details?id=ch.zhaw.facerecognition

Collaboration	Familiarity (mW)	Regularity (mW)	+Benefit (mW)
S5 executes Nexus saves			
Augment	12421.52	13663.68	1242.16
Chess	8158.19	8974.01	815.82
Face recognition	10121.57	11133.73	1021.16
Nexus executes S5 saves			
Augment	10094.80	11104.28	1009.48
Chess	9881.48	10869.63	988.15
Face recognition	9850.64	10835.71	985.07

TABLE II: Comparison of energy gains obtained when using different approaches to select collaborators.

Implications for collaborations: While other approaches to select collaborators are more suitable to spot short term collaborations that are eligible for networking use cases, in our work, we demonstrate that it is possible to engage into long and stable collaborations between multiple devices. This suggests that it is possible to envision more sophisticated use cases for collaborations between devices that can involve the usage of other resources from devices, e.g., storage, and other resource intensive sensors like thermal cameras. Our approach can also be used to design new types of duty cycling schemes that account for collaboration. For example, thermal cameras heat up when operated, which can reduce their accuracy and performance [21]. By sharing sampling cycles across different devices, it becomes possible to allow devices to cool down between samples, thus mitigating errors.

Room for improvement and limitations: We focused on finding collaborators with the longest period of duration in a particular context. While these findings can reduce the intervals of discovery and improve communication negotiations between devices, our main contribution focuses on enabling the sharing of resources between devices, e.g., CPU and sensors. In parallel to this, while we considered different locations and temporal contexts, we did not track specific devices during our study. As a result, our results do not account for influence of transportation or other mobility variations. Group mobility is very important as scenarios in which a group of devices traveling together can be stable enough to execute collaborative tasks on the move. In our previous work, we have shown that mobility patterns exhibit significant regularity [22], suggesting that our method would be beneficial also in mobility scenarios.

Static devices and devices on the move: mobile devices often employ MAC randomization to mask their actual address in an attempt to hide the identity of individuals and protect their privacy. While this introduces an extra layer of complexity to select collaborators, there is still several ways to track specific users [23], [24]. Moreover, mobile platforms, e.g., Android, implement already transparent mechanisms of interaction, which allow users to hide the MAC address while preserving their unchanged identify available to others. Since our approach characterizes places for optimal selection of collaborators, a potential candidate can be then selected based

on its relation with a location rather than through its identity.

Other applications: Collaboration between multiple devices that are stable makes it possible to exploit collaboration in new types of scenarios. For example, devices can perform asynchronous tasks, such as training a machine learning model by transferring model training between multiple devices until it has been fully learned. As each device only performs small portion of the learning, devices' resources can be better preserved. Another example is edge intelligence where the level of decision support can be adjusted depending on stability of devices with more powerful models preserved for situations where high amount of collaborators are the most stable. Similarly, small scale data centres powered by smartphones can benefit from selection of stable collaborators [7]. In such cases, cloud-like computing tasks can be push to massive group of devices that are in proximal range for long time.

Data collection in proximity networks: Besides modeling the frequency of encounters between devices in the wild, the characteristics of our data collection setup also emulate emerging 5G deployments where edge servers or base stations directly offer short range local connectivity. Our results provide new insights for multi-device systems and applications that are enabled though low latency communications. Another interesting opportunity is to combine regularity with opportunistic crowdsensing [25] and examine how stability characteristics change when data covers only a subset of the users.

VIII. RELATED WORK

Collaborator selection: Device-to-device communications enable opportunistic collaboration where devices' resources are interconnected to operate as a whole [4]. Previous work on collaborator selection has predominantly focused on networking scenarios where the devices are acting toward a common goal, the characteristics of the tasks do not change over time, and the topology of the network changes only slowly [26], [27]. Examples include wireless sensor networks [28], mobile ad-hoc networks [29], peer-to-peer communications [3], [30], delay-tolerant network and decentralized social systems [31]. In all of these scenarios, collaboration selection has focused on finding the device(s) that provide best resource support for the given task. However, not all devices are suitable as collaborators as they may be available only for a short time or not willing to share sufficient resources for collaboration [32]. Unlike others, COSINE addresses how to select the best and most stable collaborator for any task.

Multi-device sensing: Collaborative sensing approaches share the cost of sampling a sensor and includes approaches to schedule and coordinate the sampling of a sensor between multiple devices. As an example, Rio [4] is an I/O sharing framework that supports unmodified applications and exposes I/O resources for sharing with others. Other collaborative frameworks for sensing includes CoMon [1], CoSense [2], Remora [18], and CeeSR [5]. Collaborative sensing can be also supported by offloading techniques. Sensing tasks are offloaded to external servers (in a fixed location) to reduce energy consumption and improve quality of sensed data [33], [34]. While offloading can indeed reduce energy drain, making optimal offloading decisions is complex [35]. Moreover, benefiting from offloading sensing tasks requires a long stream of sensor data to be exchanged, which increases energy cost [13], [14], [16], [36]. Thus, selecting collaborators that share the same context for the longest period of time is critical for these systems. Our work addresses this key issue.

Multi-device computing: since the computational capabilities of devices have increased dramatically, collaborations between devices have been envisioned as distributed solutions that can contribute computational resources to edge or fog infrastructure [37], [38]. Several frameworks to distribute computational tasks among multiple devices have been proposed over the years [20], [39], [40]. Smartphones can be harnessed to create an aggregated computational infrastructure among multiple devices, such as FemtoClouds [8] and Micro-clouds [41]. Smart and IoT (Internet of Things) devices can also be merged into a cloudlet-like infrastructure to execute collaborative machine learning tasks [7], [9]. Cloudlets that rely on multiple devices have also been proposed [42], [43]. Battery life also has been quantified to foster the adoption of smartphones as edge infrastructure [44]. Since the execution of computational tasks can vary significantly based on multiple contextual factors, such as type of device, number of devices, quality of the CPU resource, intensity of the task, among others, the selection of a collaborator is fundamental. In our work, we focus on determining the collaborators that are most suitable depending on the context of the device that is seeking collaborations.

IX. SUMMARY AND CONCLUSIONS

In this paper, we contribute by developing a regularity-based approach to select optimal collaborators for multi-device systems. We find that our approach can be used to spot optimal collaborators with the longest and more stable duration periods. Moreover, our approach outperforms other state-of-the-art approaches to select collaborators in different contexts, providing most stable performance across different locations. This opens up a new plethora of use cases for collaborations between devices that can exploit different resources of devices, such as CPU or sensors. In addition, we also demonstrate the potential of our approach in a collaborative offloading use case. We show that COSINE improves energy saving in collaborative applications, and enhances the performance of selecting stable collaborators 2.45 times.

REFERENCES

[1] Y. Lee, Y. Ju, C. Min, S. Kang, I. Hwang, and J. Song, "Comon: Cooperative ambience monitoring platform with continuity and benefit awareness," in *Proc. of the ACM International Conference on Mobile Systems, Applications, and Services (MobiSys 2012)*, 2012.

[2] S. Hemminki, K. Zhao, A. Y. Ding, M. Rannanjärvi, S. Tarkoma, and P. Nurmi, "Cosense: A collaborative sensing platform for mobile devices," in *Proc. of the 11th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2013, p. 34.

[3] J. Scott, J. Crowcroft, P. Hui, and C. Diot, "Haggle: A networking architecture designed around mobile users," in *Proc. WONS '06*, 2006, pp. 78–86.

[4] A. Amiri Sani, K. Boos, M. H. Yun, and L. Zhong, "Rio: a system solution for sharing i/o between mobile systems," in *Proc. of the 12th annual international conference on Mobile systems, applications, and services*. ACM, 2014, pp. 259–272.

[5] J. Leppänen, M. Pelkonen, H. Guo, S. Hemminki, P. Nurmi, and S. Tarkoma, "Collaborative and energy-efficient speech monitoring on smart devices," *Computer*, vol. 49, no. 12, pp. 22–30, Dec 2016.

[6] K. K. Rachuri, C. Efstratiou, I. Leontiadis, C. Mascolo, and P. J. Rentfrow, "Metis: Exploring mobile phone sensing offloading for efficiently supporting social sensing applications," in *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2013, pp. 85–93.

[7] E. Lagerspetz, X. Li, J. Hamberg, H. Flores, P. Nurmi, N. Davis, and S. Helal, "Pervasive data science on the edge," *IEEE Pervasive Computing*, 2019.

[8] K. Habak, M. Ammar, K. A. Harras, and E. Zegura, "Femto clouds: Leveraging mobile devices to provide cloud service at the edge," in *Proc. CLOUD '15*. IEEE, 2015, pp. 9–16.

[9] H. Li, K. Ota, and M. Dong, "Learning iot in edge: Deep learning for the internet of things with edge computing," *IEEE Network*, vol. 32, no. 1, pp. 96–101, 2018.

[10] M. De Domenico, A. Lima, and M. Musolesi, "Interdependence and predictability of human mobility and social interactions," *Pervasive and Mobile Computing*, vol. 9, no. 6, pp. 798–807, 2013.

[11] D. Belli, S. Chessa, L. Foschini, and M. Girolami, "A probabilistic model for the deployment of human-enabled edge computing in massive sensing scenarios," *IEEE Internet of Things Journal*, 2019.

[12] E. Paulos and E. Goodman, "The familiar stranger: anxiety, comfort, and play in public places," in *Proc. of the SIGCHI conference on Human factors in computing systems*. ACM, 2004, pp. 223–230.

[13] C. Julien, C. Liu, A. L. Murphy, and G. P. Picco, "Blend: Practical continuous neighbor discovery for bluetooth low energy," in *Proc. IPSN '16*. ACM, 2017, pp. 105–116.

[14] C. Liu, J. Hua, and C. Julien, "Scents: Collaborative sensing in proximity iot networks," in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 2019, pp. 189–195.

[15] C. Liu, J. Hua, C. Hu, and C. Julien, "Stacon: Self-stabilizing context neighborhood for mobile iot devices," in *Proc. PerCom Workshops '19)*. IEEE, 2019, pp. 361–363.

[16] P. H. Kindt, D. Yunge, G. Reinerth, and S. Chakraborty, "Griassdi: Mutually assisted slotless neighbor discovery," in *2017 16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2017, pp. 93–104.

[17] L. Wei *et al.*, "Lightning: a high-efficient neighbor discovery protocol for low duty cycle wsns," *IEEE Communications Letters*, vol. 20, no. 5, pp. 966–969, 2016.

[18] M. Keally, G. Zhou, G. Xing, and J. Wu, "Remora: Sensing resource sharing among smartphone-based body sensor networks," in *2013 IEEE/ACM 21st International Symposium on Quality of Service (IWQoS)*. IEEE, 2013, pp. 1–10.

[19] L. Ekroot and T. M. Cover, "The entropy of markov trajectories," *IEEE Trans. On Information Theory*, vol. 39, no. 4, pp. 1418–1421, 1993.

[20] P. Guo, B. Hu, R. Li, and W. Hu, "Foggycache: Cross-device approximate computation reuse," in *Proc. of the 24th Annual International Conference on Mobile Computing and Networking*. ACM, 2018, pp. 19–34.

[21] T. Malmivirta, J. Hamberg, E. Lagerspetz, X. Li, E. Peltonen, H. Flores, and P. Nurmi, "Hot or not? robust and accurate continuous thermal imaging on flir cameras," in *Proc. of the 2019 IEEE International Conference on Pervasive Computing and Communications (PerCom 2019)*. IEEE.

[22] F. Faghihi and P. Nurmi, "An empirical study on the regularity of route mobility," in *Proc. of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, 2016, pp. 1418–1425.

[23] M. Vanhoef *et al.*, "Why mac address randomization is not enough: An analysis of wi-fi network discovery mechanisms," in *Proc. of the 11th ACM on Asia Conference on Computer and Communications Security*. ACM, 2016, pp. 413–424.

[24] C. Matte, M. Cunche, F. Rousseau, and M. Vanhoef, "Defeating mac address randomization through timing attacks," in *Proc. of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM, 2016, pp. 15–20.

[25] A. Zuniga, H. Flores, E. Lagerspetz, S. Tarkoma, J. Manner, P. Hui, and P. Nurmi, "Tortoise or hare? quantifying the effects of performance on mobile app retention," in *International World Wide Web Conference on World Wide Web (WWW)*. International World Wide Web Conferences Steering Committee, 2019.

[26] Y. Qiu, S. Li, X. Xu, and Z. Li, "Talk more listen less: Energy-efficient neighbor discovery in wireless sensor networks," in *IEEE INFOCOM 2016*. IEEE, 2016, pp. 1–9.

[27] K. Wang, X. Mao, and Y. Liu, "Blinddate: A neighbor discovery protocol," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 949–959, 2014.

[28] M. C. Vuran, Ö. B. Akan, and I. F. Akyildiz, "Spatio-temporal correlation: Theory and applications for wireless sensor networks," *Computer Networks*, vol. 45, no. 3, pp. 245–259, 2004.

[29] W. Sun, Z. Yang, X. Zhang, and Y. Liu, "Energy-efficient neighbor discovery in mobile ad hoc and wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1448–1459, 2014.

[30] S. Kosta, V. C. Perta, J. Stefa, P. Hui, and A. Mei, "Clone2clone (c2c): Peer-to-peer networking of smartphones on the cloud," in *USENIX Workshop '13*, 2013.

[31] E. Lagerspetz, H. Flores, N. Mäkitalo, P. Hui, P. Nurmi, S. Tarkoma, A. Passarella, J. Ott, P. Reichl, M. Conti *et al.*, "Pervasive communities in the internet of people," in *2018 IEEE Int. Conf. on Pervasive Computing and Communications Workshops*. IEEE, 2018, pp. 40–45.

[32] Y. Li, T. Wu, P. Hui, D. Jin, and S. Chen, "Social-aware d2d communications: Qualitative insights and quantitative analysis," *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 150–158, 2014.

[33] K. K. Rachuri *et al.*, "Smartphone sensing offloading for efficiently supporting social sensing applications," *Pervasive and Mobile Computing*, vol. 10, pp. 3–21, 2014.

[34] H. Debnath, N. Gehani, X. Ding, R. Curtmola, and C. Borcea, "Sentio: Distributed sensor virtualization for mobile apps," in *IEEE International Conference on Pervasive Computing and Communications (PerCom 2018)*. IEEE, 2018, pp. 1–9.

[35] H. Flores, P. Hui, P. Nurmi, E. Lagerspetz, S. Tarkoma, J. Manner, V. Kostakos, Y. Li, and X. Su, "Evidence-aware mobile computational offloading," *IEEE Trans. Mobile Computing*, vol. 17, no. 8, pp. 1834–1850, 2017.

[36] C. Boldrini, M. Conti, and A. Passarella, "The stability region of the delay in pareto opportunistic networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 1, pp. 180–193, 2014.

[37] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

[38] X. Su, P. Li, J. Riekki, X. Liu, J. Kiljander, J.-P. Soininen, C. Prehofer, H. Flores, and Y. Li, "Distribution of semantic reasoning on the edge of internet of things," in *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2018, pp. 1–9.

[39] A. Dou, V. Kalogeraki, D. Gunopulos, T. Mielikainen, and V. H. Tuulos, "Misco: a mapreduce framework for mobile systems," in *Proc. of the 3rd international conference on pervasive technologies related to assistive environments*. ACM, 2010, p. 32.

[40] F. Marozzo, D. Talia, and P. Trunfio, "P2p-mapreduce: Parallel data processing in dynamic cloud environments," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1382–1402, 2012.

[41] A. Sathiaseelan *et al.*, "Cloudrone: Micro clouds in the sky," in *Proc. of the 2nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*. ACM, 2016.

[42] E. Koukoumidis, D. Lymberopoulos, K. Strauss, J. Liu, and D. Burger, "Pocket cloudlets," in *ACM SIGARCH Computer Architecture News*, vol. 39, no. 1. ACM, 2011, pp. 171–184.

[43] T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt, "Cloudlets: Bringing the cloud to the mobile user," in *Proc. of the 3rd ACM workshop on Mobile cloud computing and services*. ACM, 2012.

[44] H. Flores, R. Sharma, D. Ferreira, V. Kostakos, J. Manner, S. Tarkoma, P. Hui, and Y. Li, "Social-aware hybrid mobile offloading," *Pervasive and Mobile Computing*, vol. 36, pp. 25–43, 2017.