ShareLikesCrowd: Mobile Analytics for Participatory Sensing and Crowd-sourcing Applications

Arkady Zaslavsky[#1], Prem Prakash Jayaraman[#2], Shonali Krishnaswamy[*3]

[#]*ICT Centre, CSIRO, Canberra, Australia*

[1]arkady.zaslavsky@csiro.au

[2]prem.jayaraman@csiro.au

[*]*I2R, Singapore*

[3]spkrishna@i2r.a-star.edu.sg

Abstract— **Data and continuous data streams from mobile users/devices are becoming increasingly important for numerous applications including urban modelling, transportation, and more recently for mobile crowd-sensing to support citizen journalism and participatory sensing where sensor informatics and social networking meet. While significant efforts have focused towards the analysis of mobile user data, a key challenge that needs to be addressed in order to realize the full-potential is to address the scalability issues of real-time data collection and processing at run time. By scalability, we refer to both the challenges of data capture from a large number of users, as well as the issues of energy consumed on individual devices as a result of that capture. In this paper, we present mobile/on-board data stream mining as an effective approach to address the scalability issues of mobile data collection and run-time processing and as a significant component of mobile run-time analytics. We present experimental evaluation using the Nokia mobile data challenge open track dataset to show the significant energy and bandwidth savings that mobile data stream mining can achieve with no significant loss of useful information in this process.**

I. Introduction

The past decade has seen the emergence of smart phones equipped with powerful but limited processing capabilities, a range of sensors and internet connectivity capabilities through wireless communication networks. The advent of technology has transformed mobile devices from a mere telephony device to a rich source of data. The data from these ubiquitous devices offers exciting opportunities for the development of novel Internet of Things (IoT) applications. Mobile devices equipped with sensors and communication capabilities can act as a bridge to embedded IoT objects (e.g. coffee machines), or generate information about the environment themselves [1]. In most cases, the data generated from the mobile devices are processed and transmitted to a remote server/cloud for further processing. The increasing amount of data being generated by mobile devices has resulted in significant research interest in mobile data analytics in recent years. Mobile data analytics comprise a set of tools that process, analyse and visualise data originating from mobile devices and other source (e.g. social media, wireless sensor networks).

The applications that can take advantage of mobile analytics can be classified as mobile crowd-sourcing and crowd-sensing. Mobile crowd-sensing and crowd-sourcing applications [2, 3] focus on monitoring large-scale phenomena that cannot be measured using information from a single individual. Mobile crowd-sensing and crowd-sourcing applications leverage on the plethora of sensors in smart phones to collect information from a large group of mobile users and analyse/use that information for the benefit of the group. Mobile crowd-sensing/sourcing applications need to integrate seamlessly with the user's working context by giving priority to users' operating requirements. A key function of crowd-sensing/sourcing applications is to continuously sense, process and upload sensed/processed data to the cloud/remote servers. These functions are energy expensive and in some cases can result in battery depletion. Mobile analytics can play a vital role to the success of crowd-sensing/sourcing applications by significantly reducing the amount of energy consumed by the mobile device.

Analysis on data can be performed both locally (mobile device) and remotely (server). The most prominent local analysis is conversion of raw sensed values into application usable values (E.g. Analog-to-Digital Convertor). This analysis is an inherent function of the mobile device's operating system software (e.g. iPhone IOS, Android, Blackberry). The discussion on data analytics go beyond processing offered by the mobile device operating system. We focus on data analytics algorithms that can aid crowd-sensing and sourcing frameworks/applications.

A key challenge for mobile analytics is where to perform analysis on data; locally or in the cloud. Various crowd-sensing/sourcing reference architectures employ different combination i.e. local, cloud or combination of both. Another key challenge is determining the heuristics and algorithms that achieve the desired application functionality [2]. These heuristics can range from data mediation techniques such as filtering, noise elimination, etc, context inference such as user activity, noise levels [3] etc and data mining approaches such as clustering [4], decision trees [5] etc.

II. Scenario

Consider the scenario fire in a lecture theatre illustrated in Fig 1. In such situations, it is key to determine the source of

fire and divert people to safety. Assuming a hypothetical situation where smart phones (depicted as black boxes in Fig 1) in the lecture theatre are equipped with sensors to detect temperature and humidity and have internet connectivity, a mobile crowd-sensing/sourcing application could use the mobile devices as sensor data sources to determine location of fire. Further, it can also be used to determine the intensity and direction in which the fire is spreading therein guiding people in the lecture theatre towards safety. In this scenario, mobile analytics equips the application running in the smart phone with the intelligence to processes data obtained from onboard sensors in real-time and help the user with the decision making process.

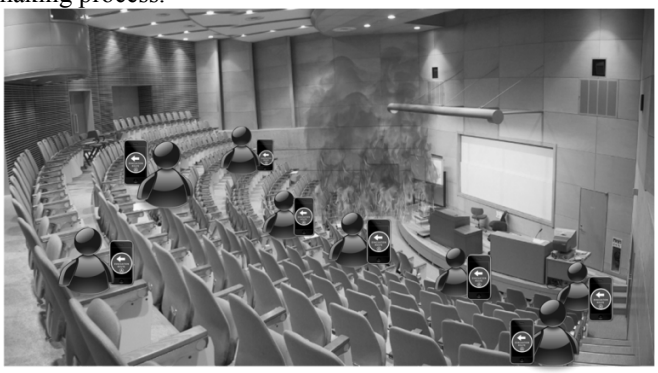


Fig. 1 Scenario – Fire in lecture theatre

III. Related Work

Mobile crowd-sensing also popularly called opportunistic sensing [2, 6] is an autonomous sensing approach that requires minimal user involvement (e.g. continuous processing of noise level around user's location). Numerous real and successful mobile crowd-sensing applications have started to emerge such as WAYZ [1] for real-time traffic/navigation information and Wazer2[2] for real-time, location-based citizen journalism among others. Mobile crowd-sensing applications thrive on the data obtained from diverse sets of smart phones owned and operated by humans. A number of mobile crowd-sensing applications use a combination of personal sensing and social networks to sense user's context.

Crowd-sourcing application on the other hand focus on outsourcing jobs to a online, distributed community of people by dividing the job into multiple subtasks [7]. Mobile crowd-sourcing is also popularly called participatory sensing as it requires active user involvement (e.g. taking images, updating traffic details etc). Numerous research prototypes have emerged in recent years such as crowd-sourcing information from mobile devices for enabling smart grids [8], Crowd-sourcing data analytics systems (CDAS) framework [9] and commercial applications that necessarily does not use mobile device like Amazon Mechanical Turk (https://www.mturk.com/mturk/welcome) among others.

Irrespective of the type of application i.e. mobile crowd-sensing or sourcing, mobile analytics play an important role in delivering key application functionalities. As mentioned earlier, mobile data analysis can be performed locally or at a remote server/cloud infrastructure. Fig 2 presents a broad classification of mobile analytic applications that in one way or another depend on sensor data originating from mobile devices. The sense component in Fig 2 could include data originating from mobile device's onboard sensors, user response to queries in crowd-sourced applications and social networking data originating from mobile users.

We classify applications that employ mobile analytics into the following:

- Push-based independent systems with local sensing and processing
- Push/Pull-based independent/collaborated systems with local sensing and cloud processing/storage
- Push/Pull-based collaborated system with distributed processing and load balancing between cloud and mobile device

1) Push-based independent systems with local sensing and processing

These systems are part of the distributed sensing infrastructure but do not collaborate with peer devices to achieve a common goal. MetroSense [10] project at Dartmouth is an example of such a system. The project aims in developing crowd-sensed/sourced applications, classification techniques, privacy approaches and sensing paradigms for mobile phones. The CenceMe [11] project under the MetroSense umbrella is a personal sensing system that enable members of social networks to share their presence.

The CenceMe application incorporates mobile analytics by capturing user activity (e.g., sitting, walking, meeting friends), disposition (e.g., happy, sad, doing OK), habits (e.g., at the gym, coffee shop today, at work) and surroundings (e.g., noisy, hot, bright, high ozone) to determine presence. The CenceMe system comprises two parts, the phone software and back-end software. The phone software is implemented on a Nokia N95 running Symbian operating system [11]. The phone software is developed in Java Micro Edition (JME) which interfaces with Symbian C++ modules controlling the hardware. Further, the mobile phone software also incorporates phone classifiers that perform analytics on data before transmitting it to the back-end server. The mobile analytics implemented on the phone include activity recognition and audio classification. The audio classifier uses the mobile phone's microphone to determine presence. A more recent version of CenceMe has also been developed for Apple iPhone. SoundSense [12] is another project under the MetroSense umbrella that exploits microphone sensor on mobile phones to infer human activity, location and social events. The system has been implemented on Apple iPhone. This system works entirely on the mobile device performing complex mobile data analysis including feature extraction, decision tree classifications and markov model recognition to analyse voice and music. It also incorporates an unsupervised adaptive classification engine.

MineFleet [13] is a distributed vehicle performance data mining system designed for commercial fleets. In MineFleet [13], dedicated patented custom built hardware devices are

[1] http://www.wayz.com/)
[2] https://www.wazer2.co.il/)

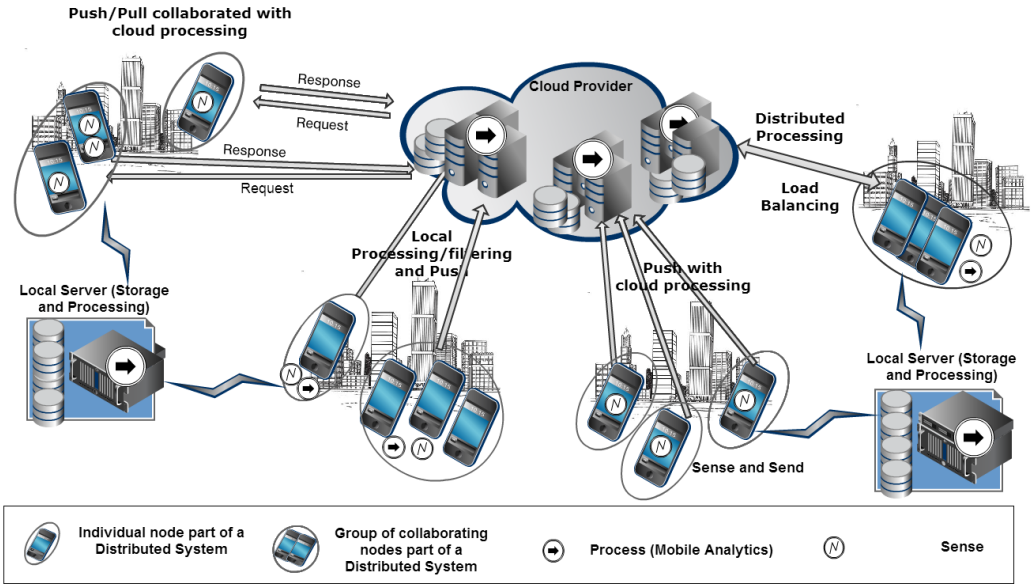


Fig. 2 Broad Classification of Mobile Analytics Enabled Systems

used on fleet trucks to continuously process data generated by the truck. MineFleet system comprises an onboard data stream mining module that performs extensive processing of data using various statistical and data stream mining algorithms. This data stored locally is transmitted to an external MineFleet Server for further processing when network connectivity is available. The MineFleet onboard data stream mining module can run on many different types of embedded devices in-vehicle-tablet-PCs, laptops and mobile phones. We note, MineFleet which is a commercial product from experimental MobiMine application [14] is developed with a specific application goal namely fleet management. Other application that employs mobile analytics to some extent on the mobile device include Serendipity [15], WhozThat [16] and SocialFusion[17].

2) Push/pull-based independent/collaborated systems with local sensing and cloud processing/storage

The push/pull-based independent/collaborating systems with local sensing and cloud/remote server processing are a class of application architectures that apply analytics on mobile sensor data on powerful servers based at remote locations on in the cloud. These applications can work independently and in collaborative mode. For example, users of smart phones report crime incidents in a given location to form a holistic view of crime level in that location [18]. Such applications transmit local sensed data to a remote server/cloud infrastructure for further processing and analysis. Sickweather [19], is an example of a crowd-sourced online social health network for sickness (e.g. fever) forecasting and mapping. Sickweather require users to report their health status to an online web system that uses a patent pending algorithm to track and map the data. The system also integrates with social networking sites pulling data published by users. The analysed data with location coordinates is presented visually in a map allowing other users to track sickness levels in different locations.

CROWDSAFE [18] is another application that leverages on crowd-sourced data to identify crimes using active user participation. The system depends on user data about crime incidents to compute safety route for other crowd-sourced users using the application. The system has two components namely application server and the mobile device interface [18]. The mobile device interface is the client application with functions to search, report and propose safety routes to users. The client is developed as a native application on Google's Android platform. The analytics employed at the server include safe route computation taking into consideration distance and safety levels and crime analytics to identify crime hotspots using K-Means clustering.

MobiMine [14] is an experimental mobile data mining system that allows intelligent monitoring of time-critical financial data. The system is implemented on hand-held PDA's. The system comprises two components namely the MobiMine Server and the MobiMine client. The MobiMine server employs complex data mining algorithms including clustering, decision trees etc to analyse stream data. The Mobile Client manages portfolio and watch list management. The portfolio module manage user preferences on stocks, risks and the watch list module uses a quadratic programming-based optimization algorithm to select a stock watch list using data obtained from the server and user's portfolio.

Crowd-sourcing data analytics system (CDAS) [9] is another example of a mobile analytics framework. In CDAS, the participants are part of a distributed crowd-sourced system. The CDAS system enables deployment of various crowd-sourcing applications that require human involvement for simple verification tasks delivering high accuracy. The system follows a two-stage approach. In the first stage, the given job is performed by a high-performance computer. The result of the job is then broken into subpart and sent to human workers for verification using Amazon Mechanical Turk (AMT). The results from human workers are combined to compute the final result. The CDAS system incorporates complex analytics

that enables it to disseminate jobs, obtain results from workers and compare answers from different workers to determine the correct one. Mobile edge capture and analysis middleware for social sensing applications (MECA) [20] is another middleware for efficient data collection from mobile devices in a efficient, flexible and scalable manner. MECA provides a platform by which different applications can use data generated from diverse mobile data sources (sensors). The proposed MECA architecture has three layers comprising data layer (mobile data sources – mobile phones), edge layer (base stations that select and instruct a device or group of devices to collect data and process data), phenomena/application layer (the backend that determines the edge nodes to process application request). The mobile analytics performed on the data in CDAS and MECA are at the cloud/remote server layer. The mobile phones act as mere data sensing and transmitting points. Karnouskos [8] proposes a crowd-sourcing application that use data from smart phones as an enabler for smart grid applications. In this case, the analytics is performed at the server end in the internet. The server provides analytics on user profiles, functions to process historic and current data and provides insight into energy consumption. The client prototype is implemented on android 2.3 platform. The server has been implemented as a REST web service developed in JAVA using Jersey framework. The data is stored on a mySQL database.

Chen and Hsu [21] propose data stream analytic as a cloud service. The work focuses on continuous processing of data streams in the cloud. The data streams originate from mobile devices. They propose an extension to the SQL query engine enabling the system to process unbounded streams of data cycle by cycle, making the analysis result visible to the user. Viswanathan et al. [22] propose the use of mobile analytics for oil production on data streaming from sensors. The architecture of the proposed system employs a remote server for data analysis. Though this application is not based on analytics on data from mobile devices, we note, it has many similarities to other crowd-sourced/sensed application discussed earlier. Viswanathan et al. [22] present a prototype implementation of a Live Operation Intelligence framework that supports analytical operators including multivariate time series analysis, patter matching and discovery, anomaly detection, correlation of events, prediction over time series data and event streams, Bayesian casual models for diagnostics and root cause analysis over event streams and visualisation.

Opportunistic sensing also called crowd-sensing approaches discussed in the literature [6, 23] mostly focus on wireless sensor networks. It is interesting to note that most of these architectures for opportunistic sensing perform mobile analytics in the cloud or remote-servers while the sensors act as data sources. Le et al. [6] propose a opportunistic data dissemination approach for mobile phone sensor networks. This approach focuses on collecting data from mobile devices using intermediately fixed infrastructure that handles mobile data analytics. The mobile data analytics proposed in the paper focus on efficient routing of data from source to destination.

3) Push/Pull-based collaborated system with distributed processing and load balancing between cloud and mobile device

The push/pull based collaborated systems distribute processing across the mobile platform and the cloud infrastructure. Mobile analytics is implemented both at the powerful cloud platform and resource-constrained mobile device platform. The system also has the ability to balance load between the mobile device and the cloud/server resources making it a hybrid approach. The local and cloud-based systems presented earlier in the literature do not apply load balancing. Further, the previous two classes of mobile analytic applications process data either at the mobile device end or the more powerful cloud end. The more generic frameworks for crowd-sourcing focus mostly on mobile analytics on the cloud while crowd-sensing methods are developed for a specific niche of applications e.g. MineFleet [13]. In [3], we have

	Real Time	Mobile Stream Processing	Sensor Data (Mobile Device)	Social Network	Context Aware	Push/ Pull	Local Storage /Query Processing	Local Processing (Mobile Analytics)	User Interaction
Sickweather [11]	✓	✗	✗	✓	✗	Push	✗	✗	High
CenceMe [13]	✓	✗	✓	✓	✗	Push	✗	✓	Low
SoundSense [14]	✓	✗	✓	✗	✗	Push	Local Storage	✓	Low
CROWDSAFE [10]	✗	✗	✗	✗	✗	Pull	Local Storage	✗	High
MobiMine [15]	✓	✓	✗	✗	✗	Push/ Pull	Local Storage	✓	LOW
MineFleet [16]	✓	✓	✓	✗	✗	Push	Local Storage	✓	LOW
CDAS [9]	✓	✗	✗	✗	✗	Pull	Local Storage	✗	HIGH
MECA [20]	✓	✗	✓	✗	✗	Pull	Local Storage	✗	MEDIUM
Smart Grid Enabler (Karnouskos [8]	✗	✗	✓	✗	✗	Pull	Local Storage	✗	LOW
Chen and Hsu [21]	✗	✗	✓	✗	✗	Push	✗	✗	LOW
CrowdSearch [23]	✓	✗	✗	✗	✗	Pull	Local Storage	✗	HIGH
CAROMM with GSNLite	✓	✓	✓	✓	✓	Push/ Pull	✓	✓	LOW

Table 1: Comparison of existing Crowd-sourcing/sensing approaches incorporating mobile analytic

developed the Context-Aware Real-time Open Mobile Miner (CAROMM) engine, a highly scalable and efficient data collection architecture for mobile crowd-sensing application leveraging on-board mobile data analytics to distribute processing with server implemented in the cloud. CAROMM employs data stream mining algorithms to process raw sensor data resulting in reduced data transmission and energy consumption. The cloud employs complex context-aware mobile analytics to process data collected from multiple mobile devices delivering real-time and situation-aware location information to mobile users. The CAROMM framework and the counterpart cloud implementation is a generic crowd-sensing framework that can be used to develop novel crowd-sensing application that could take advantage of mobile analytics distributed across the powerful crowd platform and the resource-constrained mobile device platform. We present a summary of the survey approaches in Table 1. One commonality among most of the approaches is use of cloud for further processing and analysis.

IV. CAROMM: Architecture for Context-Aware Mobile Crowd-Sensing Application

In this section, we present and demonstrate CAROMM, a novel innovative mobile crowd-sensing framework that uses a scalable data collection engine to deliver real-time and situation-aware location information for mobile users. The term mobile analytics in the context of our proposed CAROMM architecture is a set of pluggable tools that can process and analyse data both locally (on the mobile device) and the cloud (remote server) taking into consideration resource-constrained operation of the mobile device and load balancing requirements.

Fig 3 presents an extended CAROMM system architecture incorporating GSNLite. The CAROMM architecture facilitates deployment of crowd-sensing applications. The framework depicted in Fig 3 is developed using a plug-n-play architecture i.e. ability to add/remove components on-the-fly (e.g. mobile analytics tools for financial application or time series analysis tools for smart grid applications). The Mobile Data Collection & Analysis Module residing on the mobile device captures sensory data, performs local continuous real-time stream mining on the data and uploads analysed information to the cloud where further analysis, management and fusion of the incoming multiple streams need to be performed. To intelligently send only analysed information, we use our resource-aware clustering technique to identify significant changes in the situation. The proposed data collection module enables, as demonstrated later, cost-efficient collection and processing of mobile device data using on-the-move data stream mining. The data collection component uses a modular plug-in based architecture. It has the following components namely, User Interface Controller, Open Mobile Miner (includes Data Analysis-Cluster engine), Data Collection Manager and Server Upload Manager and GSNLite. The GSNLite component is responsible to interface with sensors on-board the mobile device and any external sensors. The GSNLite is a mobile version of the widely used Global Sensor Networks (GSN - http://sourceforge.net/apps/trac/gsn), a middleware for sensor data management. We note, the proposed GSNLite component presented in the architecture (Fig 3) is a work under progress. The current implementation of CAROMM framework employs a data collection module to interface with the sensors on the mobile device.

The integration of GSNLite with CAROMM framework has the following advantages. GSNLite provides CAROMM with the ability to communicate with external sensors using the abstraction of virtual sensors without bothering about interface implementation (e.g. using existing wrappers). Further, GSNLite will integrate a local storage and query processing engine with CAROMM enabling the system to continuously processes and store data locally which can be retrieved by applications when needed. For example, mobile devices with a CAROMM-GSNLite framework can respond to user queries from stored data enabling a true distributed mobile data source. Further, application can create data collection schedules on mobile devices with GSNLite and CAROMM features for continuous data stream processing, storing and querying. We note, our discussion on GSNLite is only at an architectural level. We are currently working on porting GSN to a mobile device platform and integrating its functions with CAROMM.

In this paper, the key component that is of interest is the Data Analysis-Cluster Engine (Mobile Analytics) – and in the

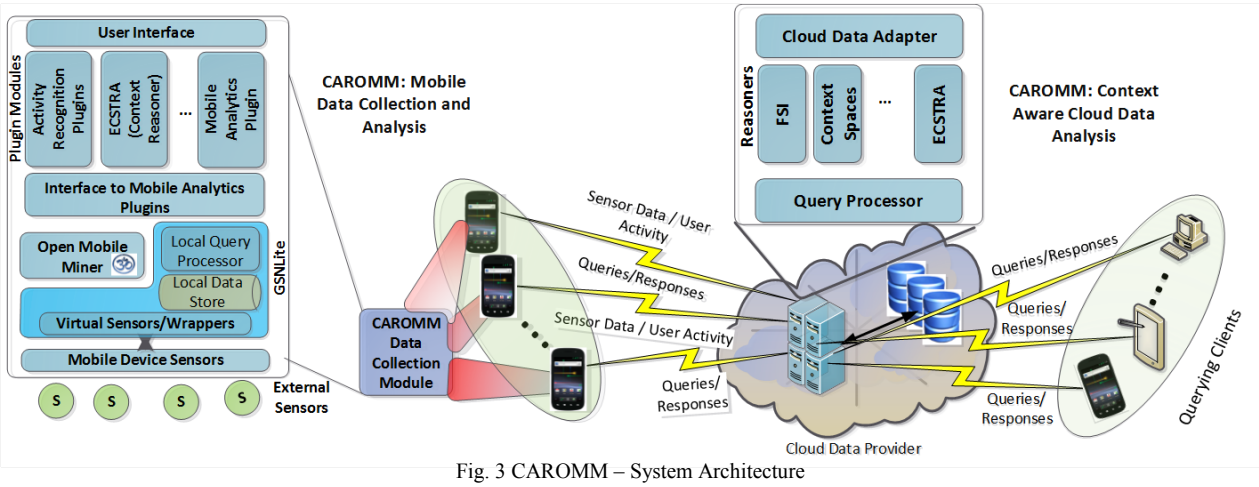


Fig. 3 CAROMM – System Architecture

interests of space, we restrict our discussion to the clustering component and its operation. The data analysis-cluster engine is the core component of the proposed data collection module. It handles all processing and analysis of data. The analysis engine performs continual mining over sensed data. For continual data mining, we use the generic toolkit for mobile data mining namely Open Mobile Miner (OMM) [24]. We have used the Light Weight Cluster (LWC) algorithm implemented in OMM toolkit to perform clustering over sensed data. OMM is a powerful resource aware mobile data miner. OMM adapts its functioning depending on resource availability on the mobile device. The LWC algorithm uses data adaptation techniques to match high-speed data streams and achieves optimum accuracy based on available resources [25]. The LWC algorithm is an outcome of our previous works in the area of mobile data stream mining. The data analysis engine uploads periodic updates of clustered data to a remote server (or the cloud using the Cloud Manager). Moreover, the data analysis engine incorporates change detection, i.e. it has the ability to determine significant change in sensed data. Any significant change in the sensed data results in the data being uploaded to the server. Further, we have also implemented a timeout procedure that will upload clustered data to the server if no change was detected over certain periods of time. The analysis engine is capable of performing data mining on multiple attributes. Hence, the change detection works across a multitude of sensed data. The use of mobile stream mining helps the data collection module upload data to the cloud only when a change in the environmental context is detected. A key objective of a good mobile data collection architecture is to significantly reduce battery and network bandwidth usage by taking advantage of on-mobile device capabilities and at the same time attaining high level of data accuracy. We show in our experiments that this can result in significant savings in energy and bandwidth usage while still retaining high level of data accuracy. Our proposed data collection strategy achieves the above objectives, as we show in the experimental evaluation.

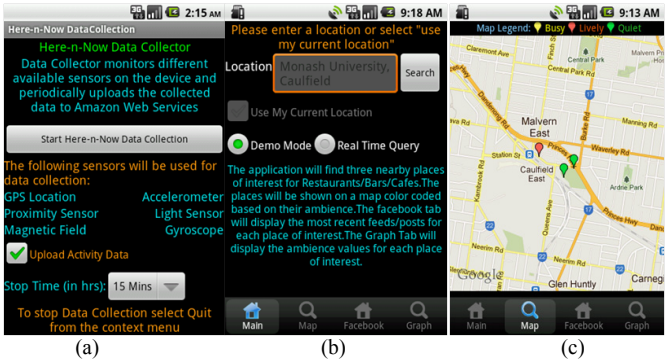
The Context-Aware Data Processing Module residing in the cloud provides the function to fuse data from multiple data streams and perform complex mobile data analysis e.g. situation inference based on multiple mobile data streams. As shown in Fig. 3, Fussy Situation Inference (FSI) [26], Context Spaces [27], ECSTRA [28] etc are examples of inference engines that can be integrated into the CAROMM framework. The innovation of CAROMM is that it allows development of any number of mobile crowd-sensing applications given its modular engineering and highly scalable and customizable architecture. The current implementation of the architecture includes situation-based reasoning engines namely FSI and Context Spaces and a REST-based web-service that the user can use to query analysed data from the cloud. Since CAROMM is a modular architecture, the existing mobile analytic tools can be easily extended to incorporate more complex processing depending on application needs.

V. HERE-N-NOW: A CONTEXT-AWARE CROWD-SENSING APPLICATION DEVELOPED USING CAROMM

In this section, we demonstrate a crowd-sensing application namely Here-n-Now built using CAROMM framework. The Here-n-Now application leverages on CAROMM's modular architecture by implementing multiple mobile analytic tools on the mobile device and the cloud.

Here-n-Now is a location information service that uses mobile crowd-sensing to collect sensory data and activity data from a large number of mobile users. Users interested in using the location information service are requested to allow participatory collection of sensory data and activity recognition based on the sensed data. Using the sensed data and user activity levels at a given location Here-n-Now is able to provide real-time reasoning about different situations/ambience of the locations. In the current implementation of Here-n-Now, we have used a Neural Network-based activity recognition model for recognising four basic activities - walking, running, sitting, and driving using the accelerometer data from the mobile device. The Here-n-Now application has the following innovative features: (i) its built on CAROMM, a highly energy-efficient and scalable data collection approach for mobile crowd-sensing applications, (ii) it captures in addition to sensory data, the activity of the user (e.g. walking, running, sitting, etc.), (iii) utilises both sensory data and user activity data as context for inferring the current situation at places of interest, (iv) includes a cloud based context-reasoning engine for fusing and processing real-time continuous data received from mobile devices, and (v) correlates mobile sensory information to information from social media in real-time.

To demonstrate a proof-of-concept implementation of Here-n-Now, we consider an application for selection of a venue for an evening out. We characterise different venues as lively, busy and quiet based on physical phenomena such as light levels, noise levels, estimates of crowd intensity, and user activity levels. A user can post a query stating a place of interest (say, bar or restaurant) in a particular location (suburb, city). Our application will then identify relevant places of interest in/near that location and collect all uploaded data related to the places of interest. The sensory data and activity data are then used for situation inferencing in each of the places of interest. The places of interest along with their situations/ambience, and any current social media updates regarding that location are then returned to the user. Fig 4 (a-



(a) (b) (c)

Fig. 4 Screenshots of Here-n-Now Application
(a): Here-n-Now Data Collection Application (b): Here-n-Now Query Application (c): Outcomes of User query for a given location

c) shows screen shots of the application showing data collection, query and results.

We implemented two mobile clients both developed using Android for the Here-n-Now proof-of-concept implementation. The first mobile client is the Here-n-Now Data Collection and Analysis application (Fig 4a) that runs on user's mobile device incorporating the mobile analytics discussed earlier. The second client is the query client (Fig 4b-4c) that user can use to query a given location's ambience. The Context-Aware Data Processing Module (Server) is deployed in Amazon Elastic Compute Cloud (EC2 -http://aws.amazon.com/ec2/). Google places API is used to return places of interest nearby a location. The current implementation use restaurants/bars/cafes as categories for places of interest.

VI. Experimental Evaluation of CAROMM's Mobile Data Collection and Analysis Module

To experimentally evaluate the savings gained by using mobile analytics on the mobile device, we present a detailed evaluation of CAROMM's mobile data collection and analysis module (Data Analysis-Cluster Engine). The experiments were conducted using Nokia Mobile Dataset [29]. The data set is consists of data collected from over 40 users over a period of 1 year. We have obtained permission from Nokia to use the dataset for research purposes.

In a real/live setting the mobile device will generate the data either through its sensors, through maintaining records (e.g. call logs, contacts and so on), or through system management (e.g. WiFi connections and so on). In our experimental evaluation, since the data has already been collected and given the large amount of data generated per user, we decided to stream the data to the phone instead of storing the data in the phone. While, in this experimental evaluation, we focus on the GPS data for the users, we do note that the mobile analytics we have developed/presented are generic and can easily be applied to other sensor data. The experiments were run in two modes–raw transmission and clustered (mobile analytics) mode. In raw transmission mode, the data was directly uploaded to the server without any processing. A similar experiment was conducted in clustered mode where data was processed using our mobile data stream clustering engine on the mobile device before uploading. This was done to determine the number of data packets successfully uploaded, total battery drain on the device and uploaded data size.

The user set for experiment was chosen randomly from the given 40 sets of users. Since the data is very large, we restricted our experiments to a few hours for each user simulating real world data collection scenarios. Our results show that significant savings can be obtained by using the mobile data stream mining technique presented in this paper.

The aim of the experiments is to comparatively evaluate our mobile stream clustering approach vis-à-vis the traditional data collection/transfer approach for real-time data collection scenarios. The evaluation parameters are:

- The amount of data sent in terms of both data size and number of packets;
- The energy drain on the phone.

We now present the results showing the amount of data sent in terms of number of packets in Fig 5. As can be seen in the comparison between the raw data collection and the clustered approach for the sampled users, the traditional approach has not only higher variability between users (in terms of packets sent), but is also significantly higher in terms of the actual numbers. It is also noteworthy that this result is only for a specific time interval of 30 min or so. Thus, the significant reduction of data packets that mobile data stream clustering is able to achieve is quite evident.

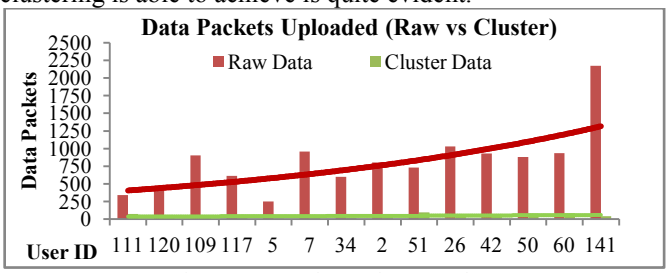


Fig. 5 Raw Vs Clustered Data Packets

The total amount of data uploaded for the time period of evaluation per user is presented in Fig 6. As with the data packets, the significant reduction in the actual amount of data sent when the clustered approach is used is remarkable. Furthermore, given that we perform change detection over the clustering, there is considerably reduced impact of variability between users. However, we do know that if a user's data is changing considerably, then it is quite likely that the clustered approach with change detection will also result in higher data transmission in order to capture the real changes in the data.

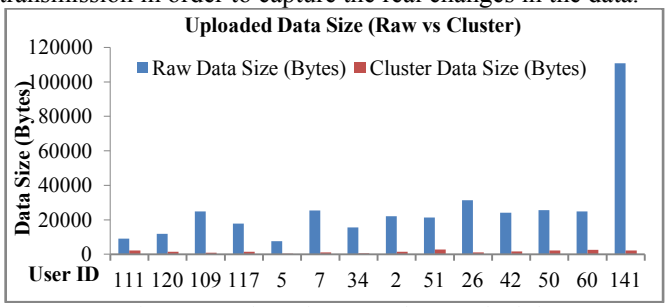


Fig. 6 Comparison of Amount of Data Uploaded

Fig 7 shows the aggregated energy savings comparison across the sampled users for the clustered approach and the raw approach. Here again we see that the clustered approach results in a lower battery drain. What is important to consider is that this drain is an average for 14 users within a 30 minute time interval. Thus, any longer term real-time data collection is definitely a case which necessitates intelligent strategies such as mobile data analytics.

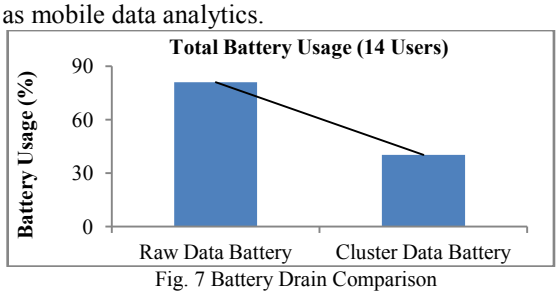


Fig. 7 Battery Drain Comparison

VII. CONCLUSION AND FUTURE WORK

In this paper we presented CAROMM, an approach to collect and process data streams from a large number of mobile devices using mobile analytics (on-the-move mining). We demonstrated the scalability and energy efficiency of CAROMM. We presented experimental evaluation using the Nokia Mobile Data Challenge open track dataset to show the significant energy and bandwidth savings that CAROMM using mobile data stream mining (analytics) can achieve.

Future work will include adding more functionality and components to CAROMM mobile analytics solutions.

ACKNOWLEDGEMENTS

The author(s) acknowledge Wanita Sherchan, Pari Delir Haghighi, Seng Loke and Abhijat Sinha for their contributions to this work. Part of this work has been carried out in the scope of the ICT OpenIoT Project which is co-funded by the European Commission under seventh framework program, contract number FP7-ICT-2011-7-287305-OpenIoT. The author(s) acknowledge help and contributions from all partners of the OpenIoT project and support from CSIRO Sensors and Sensor Networks Transformational Capability Platform (SSN TCP).

REFERENCES

[1] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *Comm. Mag.,* vol. 48, pp. 140-150, 2010.

[2] R. K. Ganti, Y. Fan, and L. Hui, "Mobile crowdsensing: current state and future challenges," *Communications Magazine, IEEE,* vol. 49, pp. 32-39, 2011.

[3] W. Sherchan, P. P. Jayaraman, S. Krishnaswamy, A. Zaslavsky, S. Loke, and A. Sinha, "Using On-the-Move Mining for Mobile Crowdsensing," in *Mobile Data Management (MDM), 2012 IEEE 13th International Conference on*, 2012, pp. 115-124.

[4] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*: Prentice-Hall, Inc., 1988.

[5] J. R. Quinlan, "Induction of Decision Trees," *Mach. Learn.,* vol. 1, pp. 81-106, 1986.

[6] V.-D. Le, H. Scholten, and P. Havinga, "Towards Opportunistic Data Dissemination in Mobile Phone Sensor Networks," presented at the Eleventh International Conference on Networks, ICN 2012, Saint Gilles, Reunion, 2012.

[7] E. H. Chi and M. S. Bernstein, "Leveraging Online Populations for Crowdsourcing," *Internet Computing, IEEE,* vol. 16, pp. 10-12, 2012.

[8] S. Karnouskos, "Crowdsourcing information via mobile devices as a migration enabler towards the SmartGrid," in *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, 2011, pp. 67-72.

[9] X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang, "CDAS: a crowdsourcing data analytics system," *Proc. VLDB Endow.,* vol. 5, pp. 1040-1051, 2012.

[10] *MetroSense*. Available: http://metrosense.cs.dartmouth.edu/

[11] A. T. Campbell, S. B. Eisenman, K. Fodor, N. D. Lane, H. Lu, E. Miluzzo*, et al.*, "CenceMe: Injecting Sensing Presence into Social Network Applications using Mobile Phones (Demo Abstract)," in *Proc. of Ninth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, Hong Kong, , May 2008.

[12] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell, "SoundSense: Scalable Sound Sensing for People-Centric Sensing Applications on Mobile Phones," in *Proc. of 7th ACM Conference on Mobile Systems, Applications, and Services (MobiSys '09)*, Krakov, Poland, 2009.

[13] H. Kargupta, K. Sarkar, and M. Gilligan, "MineFleet: an overview of a widely adopted distributed vehicle performance data mining system," presented at the Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, Washington, DC, USA, 2010.

[14] H. Kargupta, B.-H. Park, S. Pittie, L. Liu, D. Kushraj, and K. Sarkar, "MobiMine: monitoring the stock market from a PDA," *SIGKDD Explor. Newsl.,* vol. 3, pp. 37-46, 2002.

[15] N. Eagle and A. Pentland, "Social Serendipity: Mobilizing Social Software," *IEEE Pervasive Computing,* vol. 4, pp. 28-34, 2005.

[16] A. Beach, M. Gartrell, S. Akkala, J. Elston, J. Kelley, K. Nishimoto*, et al.*, "WhozThat? evolving an ecosystem for context-aware mobile social networks," *Network, IEEE,* vol. 22, pp. 50-55, 2008.

[17] A. Beach, M. Gartrell, X. Xing, R. Han, Q. Lv, S. Mishra*, et al.*, "Fusing mobile, sensor, and social data to fully enable context-aware computing," presented at the Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications, Annapolis, Maryland, 2010.

[18] S. Shah, F. Bao, C.-T. Lu, and I.-R. Chen, "CROWDSAFE : Crowd Sourcing of Crime Incidents and Safe Routing on Mobile Devices," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS 2011)*, Chicago, Illinois, 2011, pp. 521-524.

[19] *Sickweather*. Available: http://www.sickweather.com/

[20] F. Ye, R. Ganti, R. Dimaghani, K. Grueneberg, and S. Calo, "MECA: mobile edge capture and analysis middleware for social sensing applications," presented at the Proceedings of the 21st international conference companion on World Wide Web, Lyon, France, 2012.

[21] Q. Chen and M. Hsu, "Data stream analytics as cloud service for mobile applications," presented at the Proceedings of the 2010 international conference on On the move to meaningful internet systems: Part II, Hersonissos, Crete, Greece, 2010.

[22] K. Viswanathan, C. Gupta, C. Lakshminarayan, M. C. Hao, U. Dayal, R. Vennelakanti*, et al.*, "An Application of Sensor and Streaming Analytics to Oil Production," presented at the COMAD, 2011.

[23] L. Qilian, C. Xiuzhen, and C. Dechang, "Opportunistic Sensing in Wireless Sensor Networks: Theory and Application," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, 2011, pp. 1-5.

[24] S. Krishnaswamy, M. Gaber, M, M. Harbach, C. Hugues, A. Sinha, B. Gillick*, et al.*, "Open Mobile Miner: A Toolkit for Mobile Data Stream Mining," presented at the ACM Knowledge Discovery in Databases (ACM KDD 2009) Demo and Short Paper, Paris, June-July 2009.

[25] M. M. Gaber, S. Krishnaswamy, and A. Zaslavsky, "Cost-efficient mining techniques for data streams," presented at the Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation - Volume 32, Dunedin, New Zealand, 2004.

[26] P. Delir Haghighi, S. Krishnaswamy, A. Zaslavsky, and M. Gaber, "Reasoning about Context in Uncertain Pervasive Computing Environments," in *Smart Sensing and Context*. vol. 5279, D. Roggen, C. Lombriser, G. Tröster, G. Kortuem, and P. Havinga, Eds., ed: Springer Berlin Heidelberg, 2008, pp. 112-125.

[27] A. Padovitz, S. W. Loke, and A. Zaslavsky, "Towards a theory of context spaces," in *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, 2004, pp. 38-42.

[28] A. Boytsov and A. Zaslavsky, "ECSTRA: distributed context reasoning framework for pervasive computing systems," presented at the Proceedings of the 11th international conference and 4th international conference on Smart spaces and next generation wired/wireless networking, St. Petersburg, Russia, 2011.

[29] *Nokia Mobile Data Challange*. Available: http://research.nokia.com/page/12000