A Blockchain based Architecture for the Detection of Fake Sensing in Mobile Crowdsensing

Conference Paper · June 2019

DOI: 10.23919/SpliTech.2019.8783092

4 authors:

Mohamad Arafeh
3 PUBLICATIONS **1** CITATION
SEE PROFILE

May El Barachi
University of Wollongong in Dubai
80 PUBLICATIONS **329** CITATIONS
SEE PROFILE

Azzam Mourad
Lebanese American University
105 PUBLICATIONS **884** CITATIONS
SEE PROFILE

Fatna Belqasmi
Zayed University
72 PUBLICATIONS **557** CITATIONS
SEE PROFILE

A Blockchain based Architecture for the Detection of Fake Sensing in Mobile Crowdsensing

Mohamad Arafeh[1], May El Barachi[2]*, Azzam Mourad[1], and Fatna Belqasmi[3]

[1] *Computer Science and Mathematics Department, Lebanese American University,* Beirut, Lebanon
[2] *Faculty of Engineering & Information Sciences, University of Wollongong in Dubai,* Dubai, United Arab Emirates
[3] *College of Technological Innovation, Zayed University,* Abu Dhabi, United Arab Emirates

Abstract— **With the emergence of mobile crowdsensing (MCS) we now have the possibility of leveraging the sensing capabilities of mobile devices to collect information and intelligence about cities and events. Despite the promise that MCS brings, this new concept opens the door to a multitude to security and privacy threats and attacks. Indeed, the human involvement in the crowdsensing process and the openness of this process to any participant, render the task of securing MCS environments a very challenging task. In this work, we propose a Blockchain based hybrid architecture for the detection and prevention of fake sensing activities in MCS. Our architecture leverages the capabilities of the Blockchain network and introduces a new role to the MCS architecture to ensure the validation of the collected information. Combining both data quality metrics along with behavioral analysis based participants' reliability scoring, our solution is able to detect variations in behavior and quality of contributions. The proposed solution was tested with real life data collected from 200 mobile users, over the span of 2 years, and the results obtained are very promising.**

Keywords—Crowdsensing, fake sensing, Blockchain, smart cities, citizens' behavior monitoring.

I. Introduction

For many years, contextual information acquisition and sensing activities were conducted using traditional Wireless Sensor Networks [1]. However, recently, with the widespread use of smartphones and the continuous increase in their capabilities, a new sensing paradigm has emerged: Mobile crowdsensing (MCS) [2]. The concept of crowdsensing leverages the power of the crowd to collect data about a phenomena of interest (e.g. noise level in the city, or citizens' density in certain areas). Typically, a MCS system encompasses a set of data consumers (interested in crowd sensed data), data collectors (i.e. participants collecting the requested information using their phones' sensors), and a crowdsensing platform/sever acting as intermediary between data collectors and data consumers. The broker is typically responsible of receiving sensing requests from consumers, selecting the best participants for the task (based on different selection criteria), the collection of the data from the collectors who accepted the request, the data validation and its aggregation to form the final reports to be sent to the consumers, and the distribution of payments to the participants.

Due to the unique characteristics of MCS, such environments pose interesting research challenges, and are subject to many security threats and attacks. Such attacks can range from individual pollution attacks (intentionally manipulating reports to give wrong information), to malicious denial of service attacks (accepting sensing request and not returning results to prevent other honest users from participating in the sensing activities), to honest but selfish denial of service attacks (accepting all sensing requests, but completing them over an extended period of time to save resources), to orchestrated pollution attacks (group of malicious users agreeing to give conflicting reports that are far from the truth value, to impact the reliability of the output). Indeed, in MCS environments, any malicious user can act as participant. Moreover, some of the techniques used for privacy preservation in MCS (e.g. location cloaking and data anonymization) make it difficult to detect attacks and identify their sources. Most importantly, due to the human involvement in the crowdsensing process and the diversity and complexity of users' behaviors, it is very challenging to detect and prevent sophisticated security threats and attacks in a crowdsensing environment.

Some of the existing solutions attempted to address those challenges. In [3], the authors proposed a privacy-aware reputation system to address the problem of fake sensing. In this approach, the crowdsensing server associates a reputation score with each contributing device – a score that reflects the level of trust perceived by the application server about the data uploaded by that device over a period of time. A high reputation score is an indication that a particular device has been reporting reliable measurements in the past. Hence, it warrants that the server places a higher level of trust in the sensor readings from that device in the future. In [4], the authors proposed a user registration process to address Sybil attacks. In this approach, mobile crowdsensing participants are required to contribute some of their device resources in order to get registered; thus deterring people from freely register for an unlimited number of accounts. Anonymization, Spatio-Temporal Cloaking, and MIX Networks have also proved to be effective against Task tracing attacks [5,6,7] while protecting the crowd from information leakage. Furthermore, incentive and punishment mechanisms were found to increase the competency and the collaboration of the crowds while defending against MCS denial of service attacks [8]. Moreover, policy-based privacy preferences were used to increase the trust of the crowd in the platform by giving them a full control over their shared data [9,10]. Such policies included settings to ignore location-based tasks when the participant is within a specified range of a sensitive location (e.g. home or work), ignore narrow tasks, limit the number of tasks per time periods, or avoid sharing information that could be linked to previously disclosed data.

In this work, we propose a Blockchain based architecture for the detection and prevention of fake sensing attacks in MCS. Relying on a hybrid quality and reliability based approach and Blockchain based concepts, our proposed architecture combines historical data quality scoring with behavioral analysis based participants' reliability metrics to detect fake sensing activities in MCS environments.

Furthermore, it introduces watchdog miners as new role in the MCS architecture, as mean to validate the contributions of participants and dynamically adapt their payments based on their behavior and performance. Our proposed solution was implemented and tested using the data of 200 mobile users that reported their locations and activities over the course of 2 years.

The rest of the paper is organized as follows: In Section 2, we give and overview of the security and privacy threats/attacks that exist in MCS environments. Section 3 presented our targeted use case. Section 4 details our proposed architecture, including its architectural components' design and operation, as well as the data quality and participants' reliability determination approaches used. In section 6, we present our experimental results, and end the paper with our conclusions in section 5.

II. Overview of Security and Privacy Attacks in Mobile Crowdsensing

In mobile crowdsensing, users can participate in any sensing task, as long as they satisfy the sensing criteria (e.g. their presence in the area of interest, the sensors supported by their devices, their remaining battery level) [11]. Moreover, to preserve the privacy of participants, some mobile crowdsensing rely on data anonymization and location cloaking techniques. Those unique characteristics of mobile crowdsensing environment open the door to a variety of security threats and challenges.

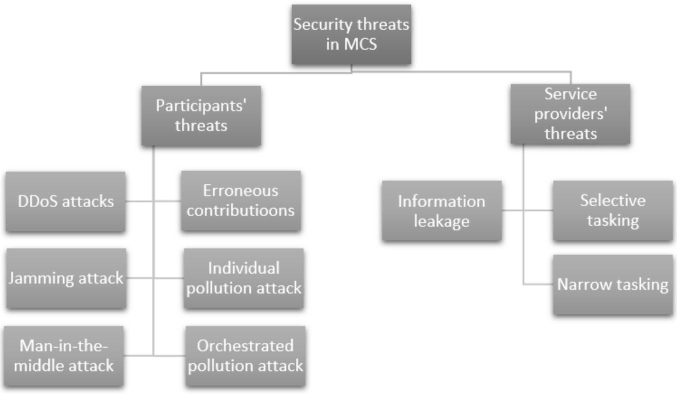


Fig. 1. Overview of Security Threats in Mobile Crowdsensing

As shown in figure 1, mobile crowdsensing security threats can be classified in two main categories: 1) Threats posed by participants; and 2) Threats posed by MCS service providers acting as brokers between data collectors and data consumers.

The threats posed by participants include:

- ***Erroneous contributions:*** Providing the wrong data due to a misunderstanding of the task requirements, possessing a low-quality or defective device, or due to the presence of malware on the device.
- ***Individual pollution attack (or fake sensing):*** Malicious participants purposefully changing sensors' readings to pollute the final report with fake data. An illustration of a pollution attack consists in putting the mobile phone in a pocket or a purse to change/impact the sensor reading, or the exposure of the phone to events for a too short period of time.
- ***Orchestrated pollution attack:*** A group of malicious users agreeing to give conflicting reports that are far from the truth value, to impact the reliability of the output.
- ***Distributed Denial of Service (DDoS) attack:*** Typically, a DoS attack seeks to prevent access to the service by crashing the server hosting the main application. In the context of mobile crowdsensing, there are two variants of the DDoS attack. The first is the DDoS attack by malicious participants. In this attack, malicious users accept sensing requests and do not return a response, thus preventing other honest users from participating and being selected. The second variant is the DDoS attack by honest but selfish users, who accept all sensing requests and try to complete them over a longer period of to receive more rewards.
- ***Jamming Attack:*** By injecting fake signals with the same frequencies as those used by legitimate participants, a jammer can interrupt their ongoing communications with the server.
- ***Man-in-the-middle Attack:*** Malicious users placing themselves in the middle of the communication between legitimate users and the server, with the aim of stealing or modifying the exchanged data.

The threats posed by service providers are mostly privacy related threats in which the tasks' distributor takes advantage of the collected data to reveal private information about the users. Among those threats, we note:

- ***Selective Tasking:*** Such attack occurs during the process of tasks' distribution by assigning tasks to a limited set of participants, in order to discover their attributes or trace them (assigning tasks to only one participant).
- ***Narrow Tasking:*** Such attack occurs during the process of task definition, by creating tasks that impose strict constraints on participants' attributes or the devices they carry (e.g., requiring a special lifestyle or a rare sensor type to qualify for the task). This attack results in the disclosure of identity or other sensitive attributes of the participant who accepts such a strict task.
- ***Information Leakage:*** In mobile crowdsensing, all participants' data and sensing records are typically stored in a centralized server. This places poses a high risk of information leakage due to some internal bugs or external adversaries targeting the system for information theft or modification.

In this work, we focus on the detection and prevention of participants' related attacks, specifically those impacting the data trustworthiness, as well as the data reliability and availability – including erroneous contributions, pollution attacks, and DDoS attacks. We start by presenting the case study we are targeting for this work, then detail the solution proposed.

III. Citizens' Activity Monitoring Use Case

The concept of smart cities stems from the need to tackle the challenges related to the rapid urban population growth combined with resources' scarcity. This concept focuses on the integration of design and technology in the urban fabric to achieve a better quality of life, and enable people to live in the smartest, most efficient, and most sustainable way possible. A key function of any Smart City initiative is to be able to continuously monitor and track the city's assets, people, and objects, and use their related data as intelligence for the streamlining of the city's operation and improvement of its performance.

Our use case is related to the idea of monitoring the activities of citizens, e.g.: their locations, to which places they go, what they do throughout the day, and with whom they interact. These pieces of information can be used to infer citizens' activities at different locations in the cities, as individuals as well as groups. A citizen can report individual information about himself/herself, or about people around them (crowd related information). Such intelligence can be used to plan and provision services within the city, to meet the citizens' needs. For instance:

- Based on the number of citizens using public transportation during different times of the day, public transportation forecasting and capacity planning can be done, and could lead to changing the number of metros/buses/trams available at different times, or their frequency of operation.
- Based on the number of citizens eating breakfast, lunch, dinner every day in certain areas, a certain number of restaurants or coffee shops need to be provided to meet the needs.
- Based on mobility and traffic patterns of drivers, traffic flow management could be optimized.
- Based on the need for different entertainment venues (E.g. cinemas, theatres, clubs…etc), a suitable number of entertainment facilities can be planned and provided accordingly.

Different types of malicious activities and threats can be observed in such a dynamic scenario, including: 1) *Erroneous contributions* **-** Collecting activity information at wrong location by mistake, or misunderstanding instructions; 2) *Malicious contributions* **-** Manipulating report to give wrong type of activity about self or others, intentionally; 3) *Individual pollution attack* - Intentionally closing GPS sensor and reporting the wrong place, to provide incomplete information that cannot be correlated against GPS data; 4) *Orchestrated pollution attack* **-** A group of malicious users agree to give giving conflicting reports about an activity/event occurring in the city; 5) *Malicious users' DDoS attacks* - Users accepting sensing request, and not returning the activity recording, thus preventing other honest users from participating and being selected; and 6) *DDoS attack by honest but selfish users* - Users accepting all sensing requests, but completing them over a long period of time, thus preventing others from participating in the process.

IV. A Blockchain based Architecture for Detecting and Preventing Fake Sensing in MCS

In this section, we present the architecture we are proposing for the detection and prevention of data reliability and availability threats in mobile crowdsensing, such as those presented in our use case. We start by discussing the system's architecture, then detail the data quality calculation and the participants' reliability determination approaches leveraged in our architecture.

A. High Level System Architecture

The architecture we are proposing relies on both data quality and participants' reliability as well the Blockchain concepts and capabilities, for the detection of pollution and DDoS attacks in mobile crowdsensing. Figure 2 depicts a high level view of our architecture, which encompasses three main roles that are connected through a Blockchain business network and share an immutable ledger of crowdsensing transactions.

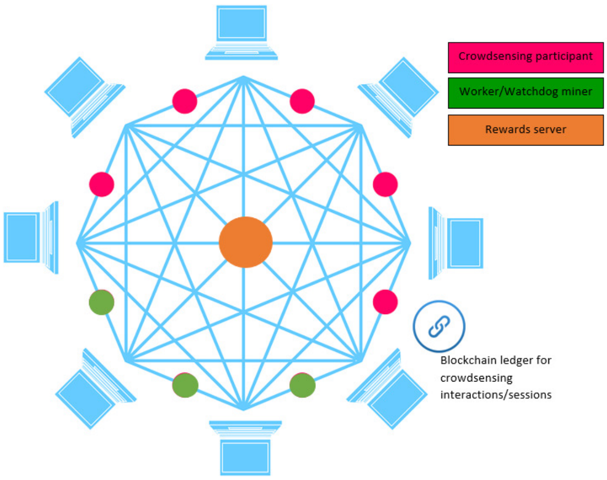


Fig. 2. High Level System Archicture

The roles encompassed in our architecture are:

1. ***Crowdsensing participants***, acting as either data collectors responsible of collecting the data required for crowdsensing tasks or consumers requesting crowdsensing reports.
2. ***Workers*** playing the role of *Watchdog miners*: Those watchdog nodes are responsible of authenticating participants; collecting the data from the collectors; evaluating the quality of the data records provided by data collectors; computing the participants' reliability scores based on a behavioural analysis technique; as well as the validation of the final reports using a consensus mechanism.
3. ***Rewards' Server***: The server is responsible of selecting participants for sensing tasks, as well as implementing a smart contracts based incentive mechanism. This mechanism assigns rewards or penalties to participants based on the quality of the data they provide and their reliability scores (inferred from their behaviour). In this scheme, honest and reliable participants that provide high quality data will be allocated high incentives, while malicious and non-reliable participants or those providing low quality data will be given penalties.

B. Architectural Components and System's Operation

Figure 3 details the software architecture of our architectural components. As shown in the figure, the ***Server*** consists of three components: 1) An authentication manager responsible of participants' profiles management and their authentication; 2) A task manager that receives sensing tasks from customers and selects the most suitable participants for each task; and 3) A payment manager responsible of allocating rewards (and penalties) to participants based on the allocated budget and the participants' performance for the task.

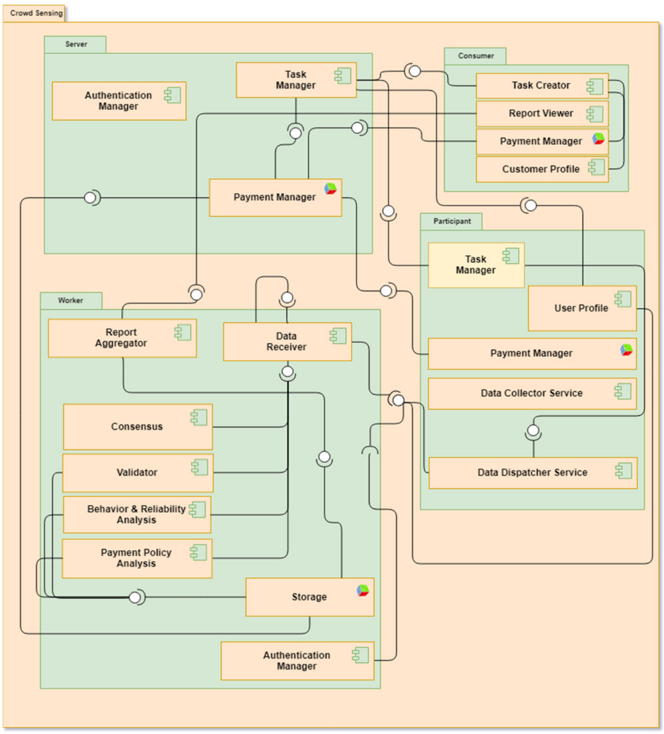


Fig. 3. Architectural Components' Software Architecture

The ***Consumer node*** is mainly used to issue crowdsensing requests and pay for the obtained reports. This node consists of 4 modules: 1) A consumer profile maintaining the user's profile, active sessions, and authentication credentials; 2) A task creation module for the creation of new crowdsensing request by specifying the event of interest, its location, the minimum quality level required, the expiry date and time for the task, and the budget allocated.; 3) A report viewer allowing consumers to view the obtained crowdsensing reports; and 4) A payment manager responsible of the execution of the smart contract and the transfer of the fees per task from the consumer's digital wallet to the server's wallet (upon the authorization of the consumer).

The ***Participant node*** is used for participation in sensing requests and the collection of the needed information. Similar to the consumer node, the participant node maintains the user's profile including the user's device information and sensing capabilities, the device's remaining battery level, and the authentication credentials. It should be noted that users' profiles impact their eligibility for being selected for sensing requests. Once a participant is selected for a task by the server's task manager module, the participant's data collector service receives the crowdsensing request from the server and initiates the data collection process. Once the data collected, it is uploaded by the Data dispatcher server to a nearby worker for processing. Finally, payments made to the participant's wallet are processed by the payment manager.

At the heart of our architecture lies the ***Worker node*** that acts as watchdog and miner node. This node receives crowdsensing data from participants, then passes it to a three steps process: 1) The Validator module responsible of evaluating the data quality and assigning it a quality score; 2) The Behavioral and Reliability Analysis module responsible of assigning a reliability score to participants based on their behavior in recent tasks; and finally 3) A Payment Policy Analysis module to determine the payment for the participant based on the quality and reliability scores previously calculated. Other modules in the worker node include the report aggregator responsible of aggregating and forming the final report from the (valid) data records collected; the consensus module responsible of the validation of the final reports based on a consensus mechanism between workers; and the authentication manager responsible of participants' authentication for providing sensing tasks related data.

Figure 4 illustrates the operation of the system, focusing on the interactions between a participant and a worker node.

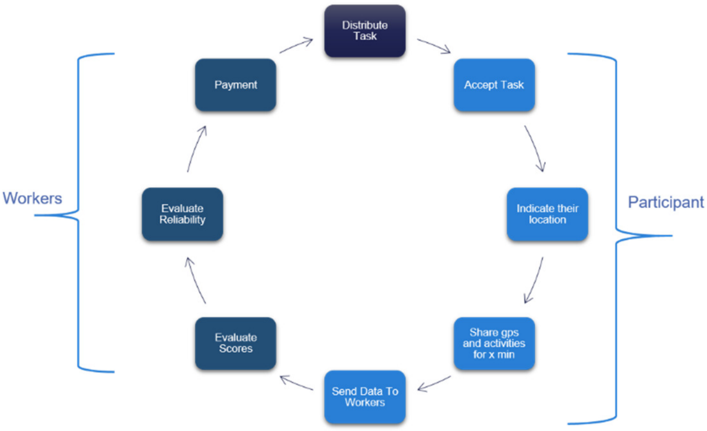


Fig. 4. System's Operation

As shown in the figure, the process starts when a participant gets selected by the server for a sensing task (activity reporting task in our use case), and receives the task's request. Upon the acceptance of the request, the participant must indicate their location from a drop down menu (e.g. home, work, shopping, restaurant…etc), while the GPS coordinates and the accelerometer data (used to infer activity) are automatically determined using the phone sensor. All this information is send to a nearby worker node for processing, and the process is repeated according to the task's data collection frequency, until its expiry. Once the task expires and all participants' data is collected, worker nodes start the evaluation process by assigning data quality scores to the different data records, as well as a historical data quality score based on the history of data records provided by each participant. Afterwards, each participant is assigned a reliability score based on behavioral analysis. In the coming sections, we detail the data quality calculation approach and the participants' reliability determination approach we used in our architecture.

C. Data Quality Calculation Approach

In this section, we discuss the process of generating the data quality scores for each of our participants. Let us suppose that each participant P receives task T, and should reply with a specified number of observations. In each task, a participant must indicate a label l for his location, while the data collector will automatically collect the GPS and accelerometer data according to the data collection frequency required, and the task expires. In that case, the contribution of each participant for each task will consist of a tuple of a scalar activity value, and an accuracy value. Nowadays, Mobiles phones are equipped with built-in accelerometer and most common mobile operating system come with a powerful activity recognitions technique that can transform on the spot the accelerometer data to user activities. $P\ x\ T$ is a matrix represented by X, which encompasses the historical data set of all the participants in all the tasks. Furthermore, each participant's data in each task is denoted by x_{ij} - a vector of participants' observations $(d_{a1}, d_{a2}, \dots\ d_{an})$ where i and j present the task and the user respectively. d is the count of activity a reported by each participant.

To calculate the data quality score, we employed a multi-features Expectation Maximization (EM) algorithm with Gaussian Mixture Model. The EM algorithm is a method that could solve the problem of finding unobserved latent variables using an iterative technique. It consists of two main steps: Expectation and maximization.

In the Expectation Step: We compute the expectation e_j for each data point in a task T with the probability of this point determined using the Gaussian Mixture Model. his probability is presented by (1)

$$e_{ij} = \frac{N(x_{ij}\ ;\ \mu_j, \sigma_j)}{\Sigma_{k=1}^{k} N(x_{k-j}\ ;\ \mu_j, \sigma_j)} \qquad (1)$$

$N(x_{ij}\ ;\ \mu_j, \sigma_j)$ denotes a Multivariate Gaussian described by the probability density function where:

$$N(x_{ij}; \mu_j, \sigma_j) = \frac{1}{\sigma_j (2\pi)^{1/d}} e^{(-\frac{xij-\mu_j}{\sigma_j^2})} \qquad (2)$$

In the maximization step: we update the mean and variance, as per (3) and (4).

$$\mu_j^{n+1} = \frac{1}{N} \sum e_{ij} x_{ij} \qquad (3)$$

$$\sigma_j^{n+1} = \frac{1}{N} \sum e_{ij}\ (x - \mu_j^{n+1}) \qquad (4)$$

We iteratively repeat the execution of expectation and the maximization steps until the log-likelihood function converges or the maximum number of executions is reached. The log-likelihood function is defined by (5).

$$\ln \mathrm{p}(x_j | \mu, \sigma) = \sum_{i=1}^{N} \ln(\sum_{k=1}^{k} N(x_{ij}; \mu_j, \sigma_j)) \qquad (5)$$

After convergence is achieved, we used the probability density function to measure the trustworthiness of participants' observations.

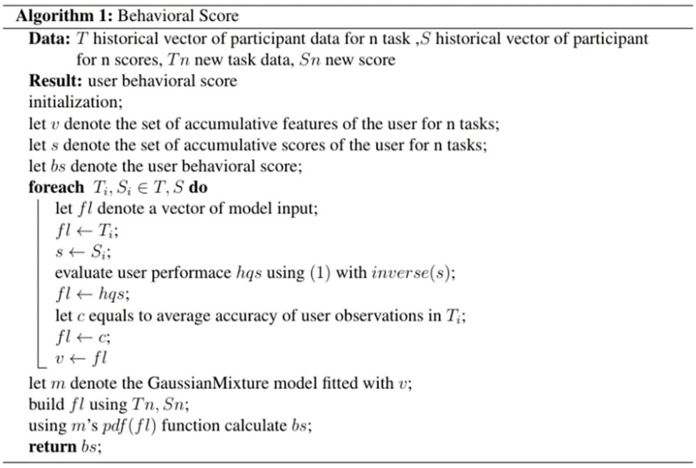
D. Participants' Reliability Determination Approach

In this section, we discuss the methods used to determine participants' reliability. We used two approaches to measure the reliability: 1) Participant's behavioural analysis based on his/her historical data; and 2) Input clustering based on the participants' historical input in the same task label. For both of our approaches, we used a Gaussian Mixture Model that follow the same approach as the Expectation Maximisation but with a different usage.

Behavioural Analysis Approach: To detect a user's behavioural changes, we've created a pool of activities combined with a consistency and efficiency parameters for a specified number of tasks, and supplied them to the model. Furthermore, we updated those values each time the user submits more data. In this case, a single cluster model can represent the user in every task type he participates in. Using Gaussian Model probability density function, we were able to generate a behavioural score each time a user submits data to a new task. User reliability can be deduced from the generated score knowing that when the model adapts to a user, a stable score will be generated each time he/her submits a new input, and an imbalanced score otherwise. Our behavioural scoring algorithm in detailed in Algorithm 1.

The participant's efficiency is calculated using a weighted average function that prioritise participant's recent scores, according to (6).

$$HQS = \frac{\sum_i^t \gamma^i S_i}{\sum_i^t \gamma^i} \qquad (6)$$



```
Algorithm 1: Behavioral Score
Data: $T$ historical vector of participant data for n task ,$S$ historical vector of participant
      for n scores, $Tn$ new task data, $Sn$ new score
Result: user behavioral score
initialization;
let $v$ denote the set of accumulative features of the user for n tasks;
let $s$ denote the set of accumulative scores of the user for n tasks;
let $bs$ denote the user behavioral score;
foreach $T_i, S_i \in T, S$ do
    let $fl$ denote a vector of model input;
    $fl \leftarrow T_i$;
    $s \leftarrow S_i$;
    evaluate user performace $hqs$ using (1) with $inverse(s)$;
    $fl \leftarrow hqs$;
    let $c$ equals to average accuracy of user observations in $T_i$;
    $fl \leftarrow c$;
    $v \leftarrow fl$
let $m$ denote the GaussianMixture model fitted with $v$;
build $fl$ using $Tn, Sn$;
using $m$'s $pdf(fl)$ function calculate $bs$;
return $bs$;
```

Participants' Clustering Approach: In our second approach we focused on the participant labelling claims. we trained a new clustering machine learning model by taking the historical users' activities combined with their performance and an additional consistency value in all of their participated tasks as input. The clustering technique involves the grouping of data. Given a set data point, each point can present multiple features, and the clustering model is able to classify each data point into a specific group. Data points that are in the same group share similarities. Using this model, we were able to label a newly submitted participant's input into the "reliable" or "non-reliable" category. By comparing the clustering model's output with the user's claim overtime, we are able to determine the trustworthiness and reliability of a participant.

V. Experimental results

In this section, we discuss the dataset used to validate our approach and present the obtained results and their analysis.

A. Dataset Used

In order to test our solution, we used the Mobile Data Challenge (MDC) dataset collected during the Lausanne Data Collection Campaign [12]. This campaign was launched by Idiap and NRC-Lausanne to collect real-time information from mobile users using the sensors embedded in their smartphones. The campaign ran for two years with nearly 200 users collecting information about their habits and movements, and is widely used in mobile computing research.

From the MDC dataset, we selected specific records that are useful for our use case, namely: 1) Users visits along with the place labels of each user; 2) Users' activities between the start and the end time of the visits. In order to supplement the data from the MDC dataset, we employed the Gaussian distribution to generate additional and equally distributed data for all the users – the Gaussian distribution having been widely used to describe sensing data in the past [13].

B. Test Scenarios and Results' Analysis

To validate our approach, we conducted two experiments using the extended MDC dataset extracted records. In the first experiment, we investigated the relation between the quality score and the behavior score, by comparing an honest and a malicious participant. Figures 5 and 6 depict the obtained results, for the honest and the malicious participants respectively. As observed in figure 5, the honest participant's behavioral score stabilizes after 7 iterations. This is not the case for the malicious participant (figure 6), in which the behavioral score shows spiking and a non-stable pattern. Furthermore, we notice that the quality score drops in the area with abnormal behavior, for the malicious participant.

In the second experiment, we investigated the effectiveness of our clustering algorithm for reliability analysis. As shown in figure 7, the machine learning model was able to correctly cluster participants into two distinct groups (reliable vs. non-reliable). As for the grey points, they represented users for which label data was manipulated to simulate malicious activity and fake claims. As shown in the figure, those cases were correctly classified as fake labels by the machine learning model, thus indicating that our model is able to distinguish correct and fake claims by participants.

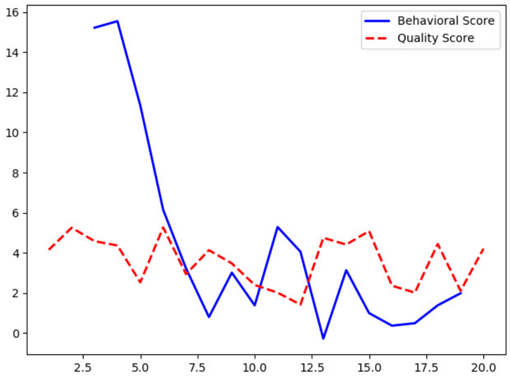


Fig. 5. Honest Participant behavioral and quality scores over

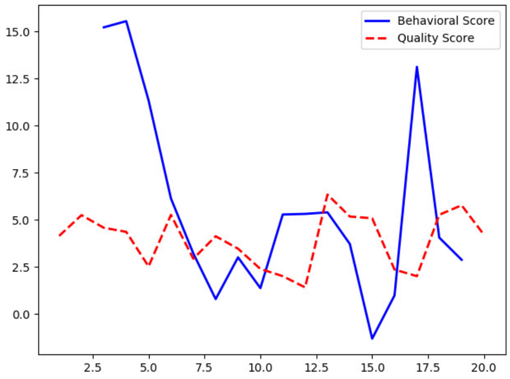


Fig. 6. Malicious Participant behavioral and quality scores over time

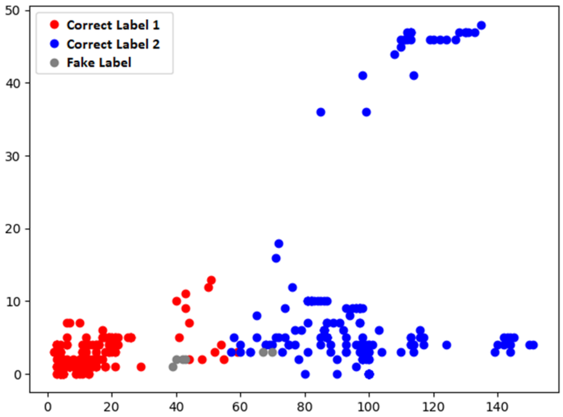


Fig.7 . Clustering model results

VI. Conclusions

Due to the dynamicity and complexity of mobile crowdsensing environments, ensuring their security and privacy aspects consists of big challenge. In this work, we focused on fake sensing activities and proposed a Blockchain based hybrid architecture that validates the contribution of participants based on an analysis of their behavior and their historical data quality scores. The proposed solution was implemented and validated using real life data collected from mobile users' activities and movements, and the results obtained are very promising.

References

[1] J. Yick, B. Mukherjee, D. Ghosal, "Wireless sensor network survey", *Computer Networks Journal*, Vol. 52, No. 12, pp. 2292 – 2330, 2008.

[2] B. Guo; Z. Yu, X. Zhou, and D. Zhang, "From participatory sensing to mobile crowd sensing," *in Proceedings of the Pervasive Computing and Communications Workshops (PERCOM Workshops)*, IEEE, 2014, pp. 593–598.

[3] A. Dua et al., "Towards Trustworthy Participatory Sensing," Proc. USENIX HotSec., pp. 1-6, 2009.

[4] X. O. Wang, W. Cheng, P. Mohapatra, et al., "Enabling reputation and trust in privacy-preserving mobile sensing," IEEE Transactions on Mobile Computing, vol. 13, no. 12, pp. 2777-2790, 2014.

[5] M. Shin, C. Cornelius, D. Peebles, A. Kapadia, D. Kotz, and N. Triandopoulos, "Anonysense: A system for anonymous opportunistic sensing," Journal of Pervasive and mobile Computing, vol. 7, no. 1, pp. 16–30, 2010.

[6] M. Li, L. Lai, N. Suda, et al., "Privynet: A flexible framework for privacy-preserving deep neural network training with a fine-grained privacy control," *arXiv preprint rXiv:1709.06161*, Sept. 2017.

[7] L. Pournajaf, L. Xiong, V. Sunderam, and S. Goryczka, "Spatial task assignment for crowd sensing with cloaked locations."

[8] K. Yang, K. Zhang, J. Ren, et al., "Security and privacy in mobile crowdsourcing networks: Challenges and opportunities," *IEEE Communications Magazine*, vol. 53, no. 8, Aug. 2015, pp. 75-81.

[9] M. Riahi, T. G. Papaioannou, I. Trummer, and K. Aberer. "Utility-driven data acquisition in participatory sensing". EDBT/ICDT, ACM, March 2013.

[10] K. Shilton, J. A. Burke, D. Estrin, M. Hansen, and M. Srivastava. "Participatory privacy in urban sensing". 2008.

[11] F. Hao, M. Jiao, G. Min, and L. Yang, "A trajectory-based recruitment strategy of social sensors for participatory sensing", *IEEE Communications Magazine*, Vol. 52 , No. 12, pp. 41-47, 2014.

[12] Laurila, Juha & Gatica-Perez, Daniel & Aad, Imad & Blom, Jan & Bornet, Olivier & Do, T.-M.-T & Dousse, Olivier & Eberle, Julien & Miettinen, Markus. (2012). "The mobile data challenge: Big data for mobile computing research". Nokia Research Center.

[13] C. Guestrin, A. Krause, and A. P. Singh. "Near-optimal sensor placements in gaussian processes". In ICML, 2005.