

# SHIELD+: A Improved Data Verification Framework with Reputation System

Wenqing Yan Panos Papadimitratos

Network Systems Security Group

KTH Royal Institute of Technology

Stockholm, Sweden

(wenqinyg, papadim@kth.se)

**Abstract -**

Without some scheme to process the usability of participants contributed data, the resulting summary statistics will be of little use to the end users. There are two main mechanisms can deal with the internal have this function – SHIELD and Reputation system.

## KEYWORDS

*Participatory Sensing Security; Reputation System; Machine learning*

## I. INTRODUCTION AND BACKGROUND

Nowadays, mobile phones not only provide the calling services. There are a batch of embedded sensors inside the equipment, which owns complex processing capabilities. Relying on the wireless transmission technology, the information collected by the equipment can be delivered to anywhere. These technological features have leaded to the appearance of a new paradigm known as participatory sensing network [1].

In participatory sensing, users collect data from their surrounding environment or their own behaviors using their mobile devices and transmit them to a campaign administrator using existing communication infrastructure (e.g., 3G service or WiFi access points) [2]. After the centralized storage and processing process, end users or other applications can access the results of the sensing tasks made by the application server. For example, the results may be displayed locally on the carriers' mobile phones or accessed by the larger public through web-portals depending on the application needs [3].

The core idea about PSN is users two-side roles. Except the user character to enjoy the result of centralized processing. They are also expected to contribute and share sensed data from their surrounding environments using their mobile phone. With more consumers, PS system can have much boarder spatial coverage. The broader the participation, the better the results [4]. However, the innate openness of PSN (participatory sensing network) makes it easy to collect corrupted data. There are two sources of contaminated data. First, honest users may inadvertently position their devices, then incorrect measurements are recorded, e.g., storing the phone in a bag while sensing urban noise information. Second, malicious users may pollute the sensor data by manipulating for their own benefits [1].

SHIELD is a Verification scheme to classify data sent from a given device based on measurement correlation with other devices to identify malicious nodes polluting the final results [4]. Reputation system empowers the campaign administrators to mark reputation scores to contributing devices, evaluating the trustworthiness so that corrupted/malicious contributions are identified [2]. A high reputation scores indicates that a device has been reporting highly reliable measurements in the past. Therefore, the administrator can rely more on sensor readings from that device in the future. Both of these two approaches have their advantages and disadvantages. In this paper, I want to combine their strengths, adding the idea of the reputation system to improve the capability of SHIELD. The new improved system named SHIELD+.

The rest of paper is organized as follows. Section II explains the reason why SHIELD+ need reputation mechanism. Section III explains the basic PS system and the adversary model. Section IV outlines an overview of SHIELD+, followed by a detailed description of its main processing phase. The last section introduces the performance evaluation method and the next-step work.

## II. MOTIVATION OF ADDING REPUTATION SYSTEM IN SHIELD

SHIELD operates on the reporting service in participatory sensing system, to assess and remove invalid, faulty data. As a good verification framework, ‘the deemed correct data can accurately represent the sensed phenomena, even when 45% of the reports are faulty’ [4]. Nevertheless, SHIELD owns two short boards. 1) it is agnostic to the cause of faulty reports. Some bad reports may be due to benign impairments of the PS honest clients, but SHIELD does not distinguish deliberately and unintentionally bad reports. In fact, the behavior of malicious users is different from the misbehaving users. Most adversaries operate sustained attacks to the system. If we want to take actions to prevent subsequent attacks, the first step is recognizing the malicious attacker or the polluted users. Therefore, recognition system

is necessary. With the help of this addition function, the

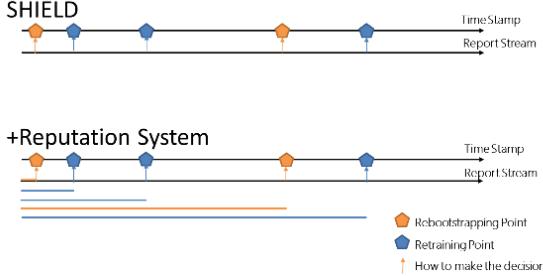


Fig.1 Difference between SHIELD and SHIELD+

special precautions, but also can filter out those potentially dangerous users during the preprocessing phase of raw data. Not only that, the outlier detection algorithm complexity and classification computation time will decrease. SHIELD use DBSCAN algorithm to cluster the input data, the operation complexity is  $O(r \cdot \log r)$ , which is related to the number of input reports  $r$ .

2) In SHIELD, if the statistical properties of the sensed phenomenon change, the system need to retrain or bootstrap again to build new classifier model. SHIELD uses predefined fix threshold to alert the system make the refreshing decision, which only based on single report from different device [4]. This refreshing mechanism is vulnerable. When there are some unstable honest users, the alert may be triggered easily, the system will be trapped in the bootstrapping or training phase. Besides, these two phases use efficient clustering method DBSCAN, which is a complex and high computational cost algorithm, so the system response capability will be affected.

To mitigate the above two points, we introduce the reputation mechanism in SHIELD+. Reputation system compute device reputation scores based on the device historical information as a reflection of the trustworthiness of the contributed data [2]. Making the refreshing decision based on device report history could enhance the robustness of the SHIELD. Fig.1 shows the main difference of SHIELD and reputation system. Instead of fix alarm threshold, SHIELD+ judge the conception drift based on the reputation scores.

Reputation system mark scores to the participants based on the quality of their contributions. Such pattern need to observe the reports submitted by each device for some time. The linkability between users and server expose the privacy risk of users. Some adversaries can exploit these links to de-anonymize [1]. SHIELD builds on anonymous collection of data to protect user privacy, but naive reputation system ignores to protect the users' anonymity. Therefore, SHIELD+ implements the IncogniSense – an anonymity-preserving reputation framework, which utilizes periodic pseudonyms generated using blind signature and transfers reputation between these pseudonyms [1].

system not only can find out those adversaries and take

### III. SYSTEM AND ADVERSARY MODEL

#### A. SYSTEM

In this paper, we use the basic Participatory Sensing (PS) system consisting of [5]:

**Users:** Participants using mobile devices (e.g., smart-phones, smart-vehicles, wearable sensor vests and armbands), equipped with multiple embedded sensors and navigation modules [5].

**Campaign Administrators:** Organization, public authorities or individuals, initiating data collection campaigns, by recruiting users and distributing sensing tasks to them [3].

**Identity & Credential Management Infrastructure:** It is responsible for the Authentication, Authorization and Access Control services by registering users and providing cryptographic credentials [6] [7].

**Reputation and Pseudonym Manager (RPM):** It takes charge of client pseudonyms system based on blind signatures [8]. Clients change the pseudonyms (also change the private key and public key) periodically, and RPM use blind signature to sign and authenticate the new pseudonyms. With the help of RPM, the user can transfer reputation scores associated with his current pseudonym to his next pseudonym [1].

**Reporting Service (RS):** The RS is an access control module that enable registered users to submit data and query the results of a sensing task [4]. SHIELD+ the same as SHIELD operates on the RS. SHIELD+ analyze and remove invalid faulty data with the help of machine learning classifiers and users' trustworthiness scores. Each user submits a report to RS. It includes a stream of measurements on the sensed phenomenon over a time interval  $p$ . Each report is like:

$$r_i = \{[v_1, v_2, v_3, \dots, v_n] | t | loc | \sigma_{PR_p} | C\}$$

$[v_1, v_2, v_3, \dots, v_n]$  is  $n$  measurements,  $v_i$  where  $i \in \{1, 2, 3, \dots, n\}$ ;  $t$  is the timestamp;  $loc$  represents a device location;  $\sigma_{PR_p}$  is a digital signature with private key  $PR_p$  which is the client new private key connected with the specific pseudonym  $p$ ; the corresponding public key  $PU_p$  included in the certificate  $C$ .

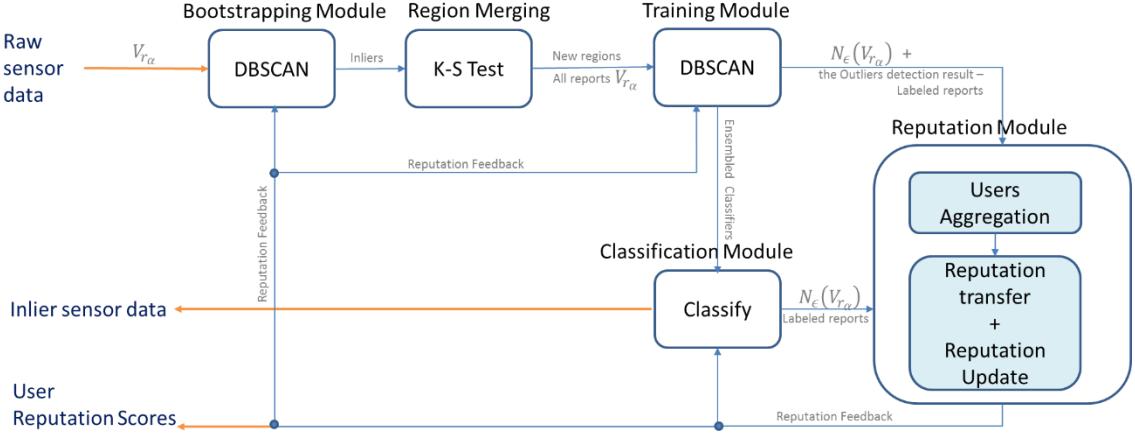
#### B. ADVERSARY MODEL

In this paper, we focus on the internal adversaries, which are active stakeholders of participatory sensing applications, i.e., participants, campaign administrators, and end users. (need to do more research). They have valid credentials, can submit authenticated faulty reports to the RS, can act individually or collectively [4].

### IV. SHIELD+ FRAMEWORKD

#### A. High-Level Overview

Compared with the original SHIELD, the new SHIELD+ system adds the E. Reputation Bootstrapping Phase and



Reputation Transfer & Updating Phase to mark reputation scores of contributing devices. Based on the reputation scores, we modify and improve the Classification Phase and the Concept Drift Detection Module.

Figure 2: System architecture with information flow

### B. Data Preprocessing

With the help of SHIELD mechanism, Reporting Service (RS) is an agent that discriminate the actual value of the sensed phenomenon, relying on multiple sources of evidence based on the reports. This process is essentially a decision making and a sensor fusion problem. Consistent with SHIELD, SHIELD+ uses *Dempster-Shafer Theory* (DST) to extract the evidence from report.

In this phase, each report is transformed into a probability mass, and computes three metrics a) the hypothesis,  $H_{max}$ , with the maximum belief, b) the belief,  $Bel(H_{max})$ , of this hypothesis, and c) the local conflict of the probability mass,  $LCon(m_c)$ . These are included in a 3-dimensional feature vector  $v_{r_a}$  (one for each report  $r_a$ ):

$$v_{r_a} = [H_{max}, Bel(H_{max}), LCon(m_c)]$$

$m_c$  denotes the probability mass derived from the user report [4].

### C. Bootstrapping Phase

For each spatial unit, the system waits until a sufficient number of reports  $\{r_a\}$  has been collected and then leverage the DBSCAN (density-based topological clustering) algorithm. The algorithm input is the feature vectors for report in  $\{r_a\}$ , named  $\{v_{r_a}\}$ . DBSCAN is a data clustering algorithm with the function of outlier detection. The output of the algorithm is a partition of  $\{v_{r_a}\}$  into inliers and outliers. The input of DBSCAN is all the feature vectors  $\{v_{r_a}\}$ , the maximum distance,  $\epsilon$ , and the *MinPoints* numbers. The distance function used in SHIELD is Canberra distance metric [4]:

$$d(v_{r_a}, v_{r_\beta}) = \sum_{i=1}^3 \frac{|v_{r_a}(i) - v_{r_\beta}(i)|}{|v_{r_a}(i)| + |v_{r_\beta}(i)|}$$

With the heuristic method introduced in [DBSCAN], we can compute the proper values of  $\epsilon$  and *MinPoints*. The  $\epsilon$ -Neighborhood of a point  $x$ ,  $N_\epsilon(x)$ , is defined by:

$$N_\epsilon(x) = \{y \in X : d(y, x) \leq \epsilon\}$$

The density of point  $x$ ,  $\rho(x)$ , is the number of points in its neighborhood area  $N_\epsilon(x)$ :

$$\rho(x) = |N_\epsilon(x)|$$

### D. Region Merging Phase and Training Phase

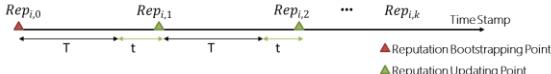
Leveraging the inlying reports of each spatial unit, the SHIELD+ then merges neighboring spatial units within which user reports follows almost the same distribution. In this phase, the system recursively calls the two sample *Kolmogorov-Smirnov* (K-S) test. The result is increasing the area of each spatial space and decreasing the number of spatial space [4].

Once the region merging phase done, the training take place for each formed region. Again, leveraging the DBSCAN clustering algorithm over reports from all the new merged spatial units. The output is a labeling of all user reports (of a region) into inliers and the outliers. Besides, there is one extra important output - the density of each input report,  $\{N_\epsilon(v_{r_a})\}$  [4].

### E. Reputation Bootstrapping Phase

After the Training Phase, each report has a label (inlier or outlier) and the density value  $\rho(x)$ . In DBSCAN algorithm the rule to classify the reports is: if  $\rho(x)$  is smaller than *MinPoints*, then this report is regard as outlier. Therefore, the density denotes the importance of each report. This can be the evidence to compute the reputation scores in the Reputation Bootstrapping Phase.

Firstly, for each formed region (Sec.D), the system merges the report according to the contributing user's ID. In the beginning of the bootstrapping (Sec.C), the system need to collect sufficient number of reports. If there are  $n$  users in the merged region. Each user may send one or more reports.



Based on the user ID, the system classifies the reports set into n subsets. Then, for each subset, (each subset represents a user,  $User_i$ ), compute the average density value,  $\bar{\rho}_{i,0}$ . With this density value, the system computes the initial reputation scores for every user in the Participatory Sensing Network. In order to simplify, we use  $\rho_{i,0}$ , instead of  $\bar{\rho}_{i,0}$  in the following paper.

Figure 3: Reputation updating cycle

In the reputation system. We tend to gradually build up trust in another user after several instances of trustworthy behavior. However, we rapidly tear down the reputation for this individual if we experience dishonest behavior on their part even in a handful of occasions. [reputation system page 5] According to the reputation behavior, SHIELD+ use the *Gompertz function* for computing reputation scores.

For user  $i$  :

$$R_{i,k}(\rho_{i,k}) = ae^{be^{c\rho_{i,k}}}$$

Where  $a$  is the upper asymptote;  $b$  controls the displacement along the x axis;  $c$  adjusts the growth rate of the function. These are the parameter of Gompertz function, and the input is the density value  $\rho_{i,k}$ . In the Reputation Bootstrapping Phase, the  $\rho_{i,k}$  is  $\rho_{i,0}$ .

#### F. Reputation Transfer & Updating Phase

The reputation scores need to be refreshed. Since, the density computing process is resource-intensive. SHIELD+ system set the time interval value  $T$  to refresh the reputation scores,  $Rep_{i,k}$ . Fig.3 shows the refreshing detail. After the  $T$  time, the system collects reports in the upcoming  $t$  time,  $\{r_a\}$ , and calculates the density of each report, forming the density set,  $\{\rho_{i,k}\}$ .  $k$  is an integer, representing the refreshing times. The Reputation Bootstrapping process is the 0<sup>th</sup> ( $k = 0$ ), then every updating process  $k$  plus one. We use the definition epoch  $k$  to represent the  $k^{\text{th}}$  time interval  $t$ .

In SHIELD+, we implement anonymity-preserving reputation framework with periodic pseudonyms. Interval  $T$  is also the period of validity of the pseudonym. After  $T$  time, every client generates new pseudonym  $p$  and corresponding new key pair  $(PR_p, PU_p)$ , with  $PR_p$  the private key and  $PU_p$  the public key and the system need transfer the previous reputation scores to the new pseudonym of the client. Then, the client sends the public key to the RPM for blind signature. Finally, the client uses the blindly signed pseudonym and the newly generated private key to report sensor readings to the Campaign Administrators and to transfer reputation to its next pseudonyms [1].

In this phase, each epoch, the system gets a new  $\rho_{i,k}$  for each user  $i$ . Then, the system transforms  $\rho_{i,k}$  to Reputation Scores,  $R_{i,k}$ , with *Gompertz function*. The input of the function needs to reflect the fact that reputation is the result of aggregating historical device information

(i.e.,  $\rho_{i,k}, k' = 1, 2 \dots k$ ). SHIELD implement the aggregating process in [2].

For user  $i$ :

$$\rho_{i,k}^{\text{norm}} = \frac{2(\rho_{i,k} - \min\{\rho_{i,k}\}_{i=1}^n)}{\max\{\rho_{i,k}\}_{i=1}^n - \min\{\rho_{i,k}\}_{i=1}^n} - 1$$

Where  $\max\{\rho_{i,k}\}_{i=1}^n$  and  $\min\{\rho_{i,k}\}_{i=1}^n$  represent the maximum and minimum density values from the density set in epoch  $k$ , respectively. Then the input of *Gompertz function* can be expressed as follows:

$$\rho'_{i,k} = \sum_{j=1}^k \lambda^{(k-j)} \rho_{i,j}^{\text{norm}}$$

Where the summation is used to facilitate the aggregation of historical information while the exponential term,  $\lambda^{(k-j)}$  with  $0 < \lambda \leq 1$ , reduce the impact of the past data.

#### G. Classification Phase

In the bootstrapping and Training Phase (Sec.C and D), the SHIELD system uses clustering unsupervised machine learning method to classify the inliers and outliers. As the high resource consumption ability of clustering, once we get the training result (Sec.D) for the merged spatial region, then the system changes to use classification method to find the outliers. With the input of the labeled reports, the system trains an ensemble of classifiers comprising a random forest, a naïve Bayes classifier and a nearest neighbor classifier [4]. Then, leveraging the ensemble classifier, the new raw data from the users can directly be grouped into inliers and outliers.

In SHIELD+, the new system keeps all the original processes in Classification Phase, but add one preprocessing step for the new raw data (the input of ensemble classifier). This preprocessing step is called Reputation Fliter. On the basis of the reputation scores variance between outlier and inliers, the system set a Reputation Threshold,  $RS$ . Through the Reputation Fliter, the system removes the reports from the users, which reputation scores  $Rep_{i,k}$  is lower than  $RS$ , where  $Rep_{i,k}$  is the latest reputation scores. Reputation Fliter, can relieve the load of the classifier and help the system to identify the malicious users and protect the final results from being polluted.

#### H. Concept Drift Detection Module

Consistent with SHIELD, this model is responsible for detecting the changes in the statistical properties of the sensed phenomenon [4]. However, instead of keeping monitors the disagreement between the probability mass, SHIELD+ uses Reputation scores to make the decision. More specifically, based on the reputation scores generated in Reputation Bootstapping phase, SHIELD+ set a predefined shreshold. If the user new score exceeds the threshold in the Reputation Updateing process, then an alert is triggered. Then, the system uses the area of concept drifts to decide to retrain the model or reboot the system. If the drift only occurs in one or a small number of regions. the system retrains (Training Phase in Sec.D). Otherwise, they

system has to be bootstrapped (from the Sec.C reboot the system).

### *I. Performance Evaluation and Next-step Work*

#### *Performance Evaluation*

#### *Next-step Work*

Do more research about adversary model and the details in blind signature.