Trust-3DM: Trustworthiness-Based Data-Driven Decision-Making Framework Using Smart Edge Computing for Continuous Sensing

HANANE LAMAAZI, RABEB MIZOUNI, HADI OTROK, (Senior Member, IEEE), SHAHTI SINGH, (Member, IEEE), AND ERNESTO DAMIANI, (Senior Member, IEEE)

Center for Cyber-Physical Systems, Department of Electrical Engineering and Computer Science, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates

Corresponding author: Hanane Lamaazi (hanane.lamaazi@ku.ac.ae)

This work was supported by the Center for Cyber-Physical Systems, Khalifa University, through the Internal Research Fund, under Grant 8474000137.

ABSTRACT Mobile Edge Computing (MEC) has been proposed as an efficient solution for Mobile crowdsensing (MCS). It allows the parallel collection and processing of data in real time in response to a requested task. A sensing task can be one-time or continuous, with multiple readings collected over time. Integrating MEC and continuous sensing in MCS is challenging due to many factors, including workers’ mobility, edge node placement, task location, Reputation, and data quality. In addition, guaranteeing cooperative communication in the presence of Anomalous data while maintaining a high quality of service (QoS) is a fundamental issue in continuous sensing. A stability-based edge node selection and anomaly detection-based decision-making framework for worker recruitment in continuous sensing is proposed to address these challenges. It can a) Select the most stable edge nodes in the area of interest (AoI), b) Dynamically cluster the workers according to their movement in the AoI, c) Locally detect and eliminate anomalies within the sensing data, and d) Adopt a feedback mechanism that ensures the cooperation between the edge nodes to eliminate untrustworthy workers in the whole sensing period and future tasks. A real-life dataset is used to evaluate the efficiency of the proposed framework. Results show that the framework outperforms the baselines by achieving higher QoS while introducing lower delay, energy consumption, and less resource consumption.

INDEX TERMS Smart edge computing, crowdsensing, distributed architecture, data quality, anomaly detection, trustworthiness.

I. INTRODUCTION

With the emergence of edge computing, handling heterogeneous data sources in crowdsensing has become a priority. Collecting data from multiple sources may produce low-quality data caused by device characteristics, environment change, and poor workers’ commitment to complete the sensing activity. Also, the limited energy provided by the device battery restricts the usage time, especially for big data processing [1].

However, the task computation intensity determines whether a task can be executed locally or offloaded remotely in single or multiple MEC servers [2]. Locating edge devices close to the data source makes it a powerful solution for handling complex tasks with a fast response time and timely computation [3]. Using parallel computation, MEC is considered an optimal solution for real-time applications while providing high optimization of communication performances and services [4], [5]. An Edge node can be a decision-maker that applies some assessment, detection, or prediction strategies to the sensed data from the workers [6]. The Edge node, deployed as a local or global entity, can play an important role in propagating information between different entities at
the same level, not only between the user and the server as in traditional work [7], [8]. It can ensure cooperation on multi-levels that can be edge-to-edge or edge-to-server communications to avoid spreading malicious behavior.

Because of prominent features of mobile edge computing (i.e., the mobility of edge node), traditional approaches may obtain less accurate data collection [9] in the MCS environments. The frequent change in the MCS ecosystem leads to the edge nodes’ disconnection. The data can become staled or uncompleted or completely forbid the edge nodes from disseminating the collected information to the server or the end-users.

Moreover, the mobility of the workers modifies their distribution and availability in the AoI. In this case, a set of changes may occur, including communication interruption, redundancy, poor coverage of the AoI, and an unbalanced distribution of workers. These problems lead to additional challenges in answering the requirements of the sensing task. Therefore, the primary question this work addresses is: **How to select the most stable edge nodes and dynamically place them in the AoI as local edge nodes to be adaptive to the workers’ movement?**

In such a dynamic environment, another key factor must be considered to achieve the required quality: cooperation between edge nodes.

On the other hand, cooperation refers to the coordination that edge nodes/servers can establish to leverage the decision-making process. This cooperation can happen at multi-levels to avoid disseminating malicious, faulty, or inaccurate data. Typically, a worker moving from one cluster to another can keep transmitting anomalous data if no edge node communication is planned to disseminate data and flag any abnormal behavior. Another question remains in this purpose which is: **How to ensure the cooperation between edge nodes to avoid the spread of anomalous behavior, represented by workers moving in the area of interest and connecting to different edge nodes during their travel?**

The importance of MEC in MCS is even more prominent when it comes to continuous crowdsensing. In this paradigm, workers are supposed to sense and send their readings many times over the sensing period, which generates an increased volume of data and causes communication overhead. In contrast with one-time sensing, where the data can be processed offline, continuous sensing requires a real-time data assessment to a) avoid unnecessary/incomplete data, b) detect anomalous readings, c) and cooperate with edge neighbors to stop the spread of anomalies or the selection of untrustworthy workers in the next time slot.

In the literature, many techniques have been proposed to assess the sensed data in MCS based on different data dimensions (e.g., radical, contextual, representational, etc.), data timeliness (e.g., latency, delay, and Age of Information which measures the task-oriented timeliness of packet delivery [10], [11]), and user perspectives (e.g., information inaccessible, insecure, error, irrelevant, inconsistent, etc.). In continuous crowdsensing, this assessment can further guide the next rounds of sensing. Additionally, crowdsourced data generally comes from multiple sources. While data coming from a single source may preserve its integrity and keep its desegregated form. However, data coming from multiple sources may lose its integrity and necessitate more consideration to detect inconsistencies. This would require an efficient data aggregation and analysis approach across all sources [12].

In MCS, trustworthiness is mainly related to the worker or the sensor’s behavior. A worker is trustworthy when he is highly committed to the assigned tasks. In contrast, a sensor is trustworthy when it provides accurate data or data with only slight perturbation and noise. Consequently, an anomalous sensor is a sensor that exhibits abnormal behavior or provides inconsistent readings. MCS systems are vulnerable to many data attacks, including (i) Data Corruption, which occurs when data loses integrity and becomes unusable or inaccessible to the user or the application [13] [14], and (ii) Data Exfiltration, which refers to any unauthorized transfer of data from one device to another [15], [16], and (iii) Data Disruption that occurs when a user loses access to its data for many reasons, such as viruses, software problems, hardware malfunctions, or even when the network is down, or the connection is lost [17], [18].

By considering these challenges, the research question that can be raised is: **How to continuously select the workers in the AoI while assessing in real-time their trustworthiness based on the quality of their provided data?**

To address these research questions, Trustworthiness-based Data-Driven Decision-Making using smart Edge Computing for Continuous Mobile Crowdsensing is proposed (Hybrid-based approach). This framework is built on top of OffSEC architecture [19] and thrives on maintaining high data quality in continuous sensing in the presence of **Multi-source data inconsistency**. The latter refers to any abnormal data or high outliers caused by faulty sensors, users’ mobility, connection loss, or even sensors specifications related to manufacturing. Also, it can be conducted by untrustworthy users who falsify the data to get benefits from the framework, such as payment or an increase in Reputation.

The main contributions this work provides are:

- Maximize AoI coverage while maintaining a continuous sensing process by deploying a stability-based EN selection using a location-based filtering method.
- Maintain a high QoS over the sensing period and tasks by selecting the best workers in terms of QoS using a greedy-based algorithm.
- Detect the non-stationary data-based anomaly using a non-regressive mechanism and classify selected workers as trustworthy and untrustworthy according to their provided readings with the overall outcomes.
- Ensure cooperation between edge nodes by deploying a feedback Mechanism based on the worker’s flag and Reputation update.
A real-life dataset is used to evaluate the efficiency of the proposed framework. The results are obtained at three levels: variation over time, over tasks, and the whole sensing period for all tasks. In conclusion, the proposed framework outperforms the baselines by achieving higher Quality of Service (QoS) while introducing lower delay and energy consumption.

II. RELATED WORK

Recent research has highlighted the crowdsensing challenges concerning network heterogeneity and data quality. However, these challenges are affected by 1) the type of sensing, one-time vs. continuous sensing, 2) Spatio-temporal crowdsensing entities distribution, and 3) worker’s behavior. If data quality does not satisfy the task requirements in one-time sensing, it cannot be changed since it is already collected. In contrast, continuous sensing can improve data quality over time by discriminating the problem source.

A. ONE-TIME VS. CONTINUOUS SENSING

One-time sensing could be an optimal solution for offline operations as training, testing, and assessing the data collected by the workers. Authors in [4] proposed a two-stage Data-Driven Decision-making Mechanism using smart edge computing (Smart-3DM). A group of workers is considered task domain-specific based on their outcomes. The group that provides high-quality data is considered for final selection and sends its collected data to a central edge node for extra processing. Multiple selections are processed locally by local edge nodes (LENs) and globally by the main edge (MEN) to reduce data computation complexity and increase data quality while attaining the task target. Similarly, one-time sensing is proposed in [19]. The authors adopted a distributed architecture (OffSEC) that offloads the server by assigning the selection of the workers to the edge.

The proposed approach overcomes the centralized MCS platform by improving the quality of selected workers while preserving the cost. The data is updated over time, and some workers can show some disloyalty to completing the task. However, opting for continuous sensing can be a good solution to overcome these challenges.

The authors [20] proposed a novel recruitment system for continuous sensing based on group stability (Stable-GRS). The most stable group of participants that provide a high quality of information (QoI) is selected using a genetic algorithm by considering the worker’s mobility. In this work, stability refers to the availability of a workers’ group in the AoI over the sensing period.

The proposed framework provides high QoI while saving costs compared to the individual-based recruitment system. Similarly, a privacy-preserving dynamic pricing mechanism is proposed [21]. It considers the participant’s sensing quality contribution while preserving the total cost allocated for participants’ recruitment. The entire cost is then optimized using the reinforcement learning technique after assessing the aggregated data provided by the workers, where only high-quality data is considered.

B. SPATIO-TEMPORAL MCS DISTRIBUTION

In addition, considering the Spatio-temporal distribution of the workers is very important. Every requested task depends on a specific sensing period/time slot and location. However, the workers’ movement in a particular AoI can decrease the probability of the worker’s commitment to completing the task.

Crowdsensing is location-based, and many attacks can be produced when sensing the data. The authors [22] proposed a smart strategy for attacks’ location identification to deal with this attack. It relies on a local configuration of the Self-Organizing Feature Map (SOFM) algorithm to locate where the fake tasks are centered. The improved version of SOFM is compared to the original one, which shows that it can achieve high detection accuracy while preserving resource consumption.

Similarly, the authors [23] involve the impact of the incentive mechanism on the workers’ behavior. First, the authors used the Perturbed game theory model to estimate the impact of incentive mechanisms on worker behavior while predicting their mobility patterns. Then, they deploy a distributed approximate algorithm that dynamically selects the participants according to their mobility patterns. They use different incentive strategies such as Hybrid-based-first, profit-first, and energy-efficient-first incentive mechanisms.

C. WORKER’S BEHAVIOR

In MCS, the workers’ Reputation is an important parameter that can be used to evaluate the trustworthiness of the data. Considering the fused data can help identify anomalous workers and improve data quality. The authors [24] propose a reputation-aware data fusion algorithm (CDR) that applies the Gompertz function to rate the trustworthiness of the reported data dynamically. The algorithm calculates the Reputation by evaluating the cooperation and reputation parameters of the correlated sensors. The space and time are discretized into sub-regions and epochs to assess the sensors. The proposed approach provides less Root Mean Square Error (RMSE) than the benchmarks leading to increased data prediction accuracy.

Similarly, a mathematical model for evolutionary dynamic sensing behaviors uses multiplex EGT [25]. The user’s behavior is incorporated into the system by evaluating their contribution to QoI and social honesty using evolutionary dynamic and statistical measurements, respectively, and integrating an incentive mechanism that considers the users’ reputation scores. The proposed system accurately discriminates user behavior in MCS environments compared to the baselines.

Another problem in MCS platforms is the cooperation between different MCS entities when using
TABLE 1. Existing solutions in the literature compared to the proposed approach.

| Ref  | Architecture | sensing | Execution | Features’ Name |
|------|--------------|---------|-----------|----------------|
|      |              | Centralized | Distributed | One-time | Continuous | Single Task | Multiple Tasks | Mobility | Reputation | Security | Data Assessment | Workers’ Behavior | Incentive Mechanism |
| [4]  |              | ✓        | ✓          |         |            |            |           |           |           |         |                |                    | ✓                     |
| [19] |              | ✓        | ✓          | ✓        |            |            |           |           |           |         |                |                    | ✓                     |
| [20] |              | ✓        | ✓          | ✓        |            |            |           |           |           |         |                |                    | ✓                     |
| [21] |              | ✓        |           | ✓        |            |            |           |           |           |         |                |                    | ✓                     |
| [22] |              | ✓        |           | ✓        |            |            |           |           |           |         |                |                    | ✓                     |
| [23] |              | ✓        |           | ✓        |            |            |           |           |           |         |                |                    | ✓                     |
| [24] |              |             | ✓        |           | ✓        |            |            |           |           |         |                |                    | ✓                     |
| [25] |              |             | ✓        |           | ✓        |            |            |           |           |         |                |                    | ✓                     |
| [26] |              |             | ✓        |           | ✓        |            |            |           |           |         |                |                    | ✓                     |
| [27] |              |             | ✓        |           | ✓        |            |            |           |           |         |                |                    | ✓                     |
|      | The proposed approach | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

D2D communication. The authors [26] proposed a multi-criteria decision system that switches between different communication modes, infrastructure, or D2D, relying on a reputation-based incentive mechanism to motivate users’ cooperation. The proposed work allows for achieving a high offloading ratio compared to a centralized platform. However, the participants can also react selfishly or rationally when collecting or transferring the data. The authors [27] propose a Hybrid-based-aware task allocation policy relying on a stability monitoring scheme that uses an online control strategy based on the current information to maximize social welfare and ensure network stability balance.

### III. SYSTEM MODELLING AND REQUIREMENTS

#### A. ARCHITECTURE MODEL

The proposed model is deployed on our pre-built architecture called OffSEC. It is a multi-layer edge-based framework consisting of an edge server, edge nodes, and workers. The edge servers are responsible for initiating or accepting incoming sensing tasks, clustering the connected edge nodes, and allocating edge nodes’ responsibilities. Edge nodes have two distinguished roles: local edge nodes (LENs) and the main edge nodes (MEN). LENs offloads the server by recruiting optimal workers and ensuring parallel computation of the collected data. MEN are the nearest LEN to the task with extra computational power. Each MEN, in addition to its role as LEN, aggregates the overall data from the neighboring LENs and reports the outcome to the edge server. Finally, the workers collect information using their devices and report their outcomes to the closest LEN. Figure 1 illustrates the continuous selection process. The proposed solution is a multi-layer framework where:

- **First Layer** Consists of selection and clustering of ENs by edge server.
- **Second Layer** Consists of local worker selection and data assessment offloaded by the edge server to LENs (steps 1-6).
- **Third Layer** Consists of MEN decision-making and the feedback mechanism (Steps 7-10).

1) **TASK MODEL**

The edge server defines the requirements for task $t_j$ over a sensing period, where a task is fulfilled when the sensing period is timed out. A sensing task is defined as $t_j = (L^T_j, S^T_j, QoS^T_j, K)$

where $L^T_j$ is the task coordinates, $S^T_j$ is the set of needed sensors, $K$ is the number of required workers, $QoS^T_j$ is the minimum required QoS per worker. However, the quality of data is evaluated in real time, where the selected workers within a cluster should provide consistent data that is not presenting any abnormal value.

2) **MOBILITY MODEL**

In the AoI, the workers move following a random mobility model. At each time epoch, the worker at location (X, Y) randomly moves toward the newly chosen direction to be reached at time $t + 1$. However, some participants can stand in one location for a short period, defined as “pause time.” During this time, the speed associated with the worker is equal to 0. A dataset containing the vehicular mobility trace of the city of Cologne, Germany, is used to model the workers’ mobility. It consists of participants’ IDs, locations, and speed at a specific period [28], [29].

3) **COVERAGE MODEL**

Considering the mobility of the workers in the AoI, selecting the most stable edge nodes is a requirement. First, the edge server creates a list of preferences based on the availability shared by potential edge nodes during the sensing period. Also, only edge nodes that maximize the coverage of AoI are considered. To evaluate the coverage achieved by a set of edge nodes, the AoI is divided into identical sub-regions with similar dimensions. A sub-area that contains at least one EN is considered covered and uncovered otherwise. The GPS location of EN is used to locate their related sub-regions. To validate the proposed framework, a set of parameters are considered, described as follows:

- **AoI boundaries**: (4000 to 25000) × (4000 to 25000)
- **AoI is divided into 25 identical sub-area**
The edge nodes coverage is evaluated as follows:

\[
\text{Coverage} = \frac{N_0 \text{ of Covered Sub Area in AoI}}{\text{Total } N_0 \text{ of Sub Area in AoI}}
\]  

(1)

where:
- Coverage \( \in [0, 1] \)
- Coverage is 1 when each sub-area in the AoI is covered by at least one edge node.

4) QUALITY OF SERVICE (QoS) MODEL

To fulfill the requirement of the sensing task in terms of QoS while selecting the optimal worker, each LEN considers the expected QoS required to execute the task. It is used as a selection condition that reflects a worker’s ability to perform the sensing task. Only workers that possess \( QoS_i^{Cr} \) (where \( i \) is the worker and \( Cr \) refers to its related cluster) greater or equal to the expected one are included in the selection process. The \( QoS_i^{Cr} \) is computed using the worker’s reputation \( (R_i^w) \), sensors availability \( (SA_i^w) \) in the worker’s device, the propagation delay \( Dp_i^L \) and the propagation energy \( Ep_i^L \). The \( QoS_i^{Cr} \) is a weighted sum equation where the weights \( w_1 - w_4 \) reflect the metric’s importance and are calculated using Eq (2):

\[
QoS_i^{Cr} = w_1 \times SA_i^w + w_2 \times R_i^w + w_3 \times \frac{1}{Dp_i^L} + w_4 \times \frac{1}{Ep_i^L},
\]  

(2)

where \( \sum_{i=1}^{4} w_i = 1 \) and \( 0 < w_i < 1 \).

The sensors availability is calculated as follows:

\[
SA_i^w = \frac{\sum_{i=1}^{m} SA_i}{S_f^w}
\]  

(3)

where \( m \) is the number of sensors available in a worker device, all variables are listed in Table 2.

The propagation delay is the time a LEN spends to forward the task to the workers within a cluster. It is calculated as:

\[
Dp_i^L = \frac{\text{Dist}(W_i, LEN)}{\text{Propagation Speed}}
\]  

(4)

where \( \text{Dist}(W_i, LEN) \) is the distance between every worker and its related LEN.

The propagation energy is the energy spent to deliver the requested task to the workers. It includes the energy consumed by a LEN in full power mode (CPU) and the transmit power spent during the offloading of the task [30]. It is calculated as follows:

\[
Ep_i^L = Dp_i^L + P_I
\]  

(5)

where \( P_I \) is the idle power of mobile devices.

5) REPUTATION MODEL

Reputation is updated at the end of each time slot; it measures the ability of a worker to provide accurate data over time. The cooperative score is calculated according to the worker’s outcomes. A faulty contribution of the worker leads to a cooperative score equal to 0, otherwise is equal to 1. The Reputation is then updated as follows:

\[
\text{Rep}_i = \alpha \times \text{Rep}_i + (1 - \alpha) \times \text{Cooperative Score}
\]  

(6)

where \( 0 < \alpha < 1 \)

In the proposed framework, the MEN is responsible for reputation updates and distribution to the neighboring LENs. The LENs then considers the updated Reputation of the workers in their next selection for future time slots and tasks.
B. SYSTEM REQUIREMENTS
To simulate continuous sensing, a task is fulfilled after the sensing period is over. The sensing period is segmented into slots with equal intervals of 1-second duration. The total sensing period per task is set up to 30 seconds. At a specific time slot, the Reputation of the selected workers is updated and considered in the next time slot when another set of workers is selected, ensuring the sensing process’s continuity over time, and satisfying the cooperation goal between LENs through the MEN. However, once the sensing period is over, the new workers’ reputation updates are sent to the server to be considered in the following tasks.

To model the proposed framework, a set of requirements are considered to select the most stable ENs that play the role of LENs and MEN, presented as follows:

1) For the LENs
   - The LEN should not move out of the AoI
   - The LEN should be available during the whole sensing period
   - The LEN must be connected during the entire sensing period
   - The LEN must be close to the task
   - The LEN must have a low displacement. The displacement is equal to 0 if the LEN is not moving, while it is equal to the reverse of the velocity if the LEN is moving.
   - The LEN must have High computation capabilities

2) For MEN
   - MEN should have the highest computational capabilities from the selected LENs.

3) For the worker
   - A worker must be close to the LEN
   - A worker can be connected to the server or discovered by the LENs
   - A worker should satisfy the task requirements in terms of QoS

In the AoI, four types of workers exist: Connected workers moving out of the AoI, Connected workers moving within the AoI, Non-connected workers moving out of the AoI, and Non-Connected workers moving within the AoI. As illustrated in figure 2 the most stable edge nodes are those moving within the AoI.

IV. PROPOSED APPROACH
This section describes the framework Establishment steps and continuous selection process.

A. STABILITY-BASED EDGE NODES
A stable communication between edge-to-server and edge-to-workers in a mobile environment should satisfy criteria related to the edge nodes’ movement in the AoI during the sensing period. In this work, edge node stability is considered. The most stable edge nodes available in the AoI during all the sensing periods are selected and added to the preference list as potential LENs. The AoI is split into several sub-area.

The edge nodes that cover most sub-region and move within the AoI while being available during the sensing period are recognized as the most stable edge nodes. The overall process is described in Algorithm 1.

Algorithm 1: Stability-Based Edge Nodes

```
Input : sensing Period, Participant’s mobility, AoI boundaries
Output: Most Stable EN
1. initialization
2. for every connected participant do
3.   Calculate Coverage according eq(1)
4.   Check the shared participant’s availability over sensing period
5.   end
6. if W_i is in AoI and W_i is connected over the sensing period then
7.   Add W_i to the list of preselected EN
8. end
9. end
```

B. TASK DISTRIBUTION AND EDGE NODES CLASSIFICATION
The edge server is responsible for distributing the task to the edge nodes. It calculates the potential LENs objective function, which consists of the residual energy and CPU of the edge node devices, The closeness of the edge node to the requested task, and the displacement of the edge node in the AoI and classifies them in ascending way. The overall process is described in Algorithm 2.

Algorithm 2: Task Distribution and Edge Nodes Classification

```
Input : sensing Period, Participant’s mobility, AoI boundaries
Output: Most stable EN
1. initialization
2. for every connected participant do
3.   Calculate Coverage according eq(1)
4.   Check the shared participant’s availability over sensing period
5.   end
6. if W_i is in AoI and W_i is connected over the sensing period then
7.   Add W_i to the list of preselected EN
8. end
9. end
```

C. DYNAMIC CLUSTERING
Since the EN and the workers are moving in the AoI, the clustering is dynamic and is defined according to the requested task, sensing time slot, and workers’ mobility patterns. The optimal number of clusters is determined using the silhouette mechanism. This number is then used to select the top potential LENs that can act as centroids to the clusters. Finally, the clustering is established using the K-means algorithm, where the available workers in the AoI are clustered with closet centroids and are considered for the selection process. The overall process is described in algorithm 3.
Algorithm 2: LEN’s Objective Function Computation

Input: Set of tasks $T_j$, EN’s $R^W_i$, $U^W_i$, $L_i$, $L_j^T$
Output: Closeness, Displacement, Objective function, LENs classification

1. initialization
2. calculate ENs closeness as defined in [19]
3. $C_{ij}^W = 1 / \sqrt{(x_i^w - x_j^t)^2 + (y_i^w - y_j^t)^2}$
4. calculate LENs Displacement
5. $Dis_{ij}^W$
6. if $V_i = 0$
7. $Dis_{ij}^W = 0$
8. else
9. $Dis_{ij}^W = 1 / V_i$
10. end
11. end
12. calculate EN’s objective function
13. $Objective\_function = w_1 \times R^W_i + w_2 \times U^W_i + w_3 \times Dis_{ij}^W + w_4 \times C_{ij}^W$
14. Sort LENs in ascending way according to their objective function

Algorithm 3: Dynamic Clustering and LENs and MEN Identification

Input: sensing time slot $t_s$, Classified EN list, Participants $W$
Output: Clusters, LENs, MEN, Workers

1. initialization
2. for every time slot $t_s$ do
3. Calculate the number of optimal clusters as
4. $best\_size = silhouette\_evaluation(L^W_i, \text{Euclidean\_dist})$
5. From the list of classified EN, select $LEN_{ij}$ as the top $best\_size$ ENs
6. if $LEN_{ij}$ = best_objective function
7. $LEN_{ij}$ is MEN
8. end
9. Construct the clusters as:
10. $Cr_{id} = kmeans(L^W_i, best\_size)$
11. Centroids = best_size $LEN_{ij}$
12. end

D. LOCAL SELECTION STRATEGY
At every time slot, every LEN selects a set of workers. The selection process uses a greedy algorithm where the best workers with high QoS are eligible to send their collected data. Once the workers are selected, the LEN assigns a label with the name “Flag” initiated with a 0 value. This flag is used to identify untrustworthy workers and as a cooperation index between LENs. This flag is updated later by the MEN when providing its feedback where trustworthy the workers have a positive flag; otherwise, it is negative. After the first selection, the LEN checks the workers’ flags. The LEN can find three types of flags: a flag with “NaN” value means that the worker is new and has never been selected by other clusters, a flag with a negative value means that the worker was identified as untrustworthy by one of the LENs and then they are eliminated from the selection, and flag with positive value means that the worker has been identified as trustworthy by one of the LENs. When the selection process is finished, every LEN collects the data from the eligible workers and applies its strategy to detect anomalies in the data. Then the LEN decision output is sent to the MEN for more processing. The selection process is described in algorithm 4.

E. LOCAL DECISION STRATEGY: DATA IN-DECISION OUT (DAI-DEO)
As explained previously, the LENs adopt the DAI-DEO strategy, shown in Algorithm 5. First, the LENs collected the data from the eligible workers. Then they extract some features from this data, such as average and standard deviation, and define upper and lower limits accordingly. Finally, the LENs
A set of evaluation metrics are considered to assess the efficiency of the proposed framework and categorized as follows: data-related, network-related, system-related, and worker-related.

- **Data-related metric**: the coefficient of variation (CV) is used to evaluate the data inconsistency. It is calculated as the ratio of standard deviation and average of the collected data, following the model defined in [4].
- **Network-related metrics**: contains QoS, network delay, and network energy consumption, following the model defined in [4].
- **System-related metrics** evaluate the overall resource consumption of the proposed framework. It includes Execution time (s), Memory utilization (RAM %), and processing power (CPU %), following the model defined in [4].
- **Worker-related metrics** include the number of selected workers, trustworthy workers, anomalous readings, and the workers’ Reputation. Most of the computed parameters follow the models provided in [19].

### Algorithm 4: Workers’ Greedy Selection

```plaintext
Algorithm 4: Workers’ Greedy Selection

Input : M^w_i, Flag, K
Output: anomalous W_i, Trustworthy W_i
1 initialization
2 Set Flag = 0
3 for each Cr do
4 Verify Flag
5 if – I <= Flag then
6 Keep W_i
7 else
8 Drop W_i
9 end
10 end
11 Calculate QoS^w_i according eq(2)
12 Sort QoS^w_i in ascending order
13 drop W_i that their QoS^w_i <= QoS^T
14 Select in greedy way W^G_i <= K / Number of Cr
15 Define Final list of W^G_i
16 Apply LEN Strategy as defined in algorithm 5
17 Send LENs strategy outputs to the MEN
18 end
19 Apply MEN Strategy as defined in algorithm 6
```

### Algorithm 5: LEN’s DAI-DEO Strategy

```plaintext
Algorithm 5: LEN’s DAI-DEO Strategy

Input : Aggregated_groups, Data_readings
Output: data_assessment_thresholds, anomalous_Workers, Trustworthy_workers
1 for Every_cluster do
2 Aggregate the worker’s outcomes
3 Calculate upper_threshold = data_readings(Avg) + 1 * data_readings(Std)
4 Calculate lower_threshold = data_readings(Avg) - 1 * data_readings(Std)
5 Untrustworthy W_i = Aggregated_group where [(data_readings > upper_threshold) || (data_readings < lower_threshold)]
6 Trustworthy W_i = Aggregated_group where [(data_readings <= upper_threshold) && (data_readings <= lower_threshold)]
7 end
```

### Algorithm 6: Cooperative Score and Reputation Rate

```plaintext
Algorithm 6: Cooperative Score and Reputation Rate

Input : Score, Untrustworthy W_i, Trustworthy W_i, Flag
Output: Reputation update, Flag update
1 initialization
2 Set Flag = 0
3 for Every Trustworthy W_i do
4 Assign Score = 1
5 Update the R^w_i according eq(1)
6 Increments Flag for Trustworthy W_i
7 end
8 end
9 for Every Untrustworthy W_i do
10 Assign Score = 0
11 Update the R^w_i according eq(1)
12 decrements Flag for Untrustworthy W_i
13 end
14 end
15 Send updates to the LENs
```

F. GLOBAL DECISION STRATEGY: DECISION IN-DICTION OUT (DEI-DEO)

The MEN receives all the LENs decisions. First, the MEN remove the redundant workers’ IDs, whether they are trustworthy or untrustworthy. This redundancy occurs because of the workers’ mobility, where a worker in a time slot can be detected and selected by more than one LEN. Consequently, redundant workers’ IDs are removed to reduce resource consumption while distributing the MEN feedback. After this filtering process, the MEN assigns a cooperative score to the final set of workers provided by all LENs, as shown in Algorithm 6. A null score is attributed to untrustworthy workers, and a positive otherwise. This cooperative score is then used to update the workers’ reputations. Also, the MEN update the workers’ flags. All the updates are then forwarded to all LENs for consideration in their next selections. These new updates are also reflected in the following tasks.

V. PERFORMANCE EVALUATION

A real dataset containing vehicular mobility traces of Cologne, Germany, is used [28], [29]. It includes participants’ IDs and locations over time: another real dataset, namely the Sarwat Foursquare dataset [31], [32] for social networking.


TABLE 3. Proposed Framework vs. Benchmarks.

| Similar Features | Hybrid-based | Local-based | Global-based | Task-based |
|------------------|--------------|-------------|--------------|------------|
| Architecture     |             |             |              |            |
| Execution        |             | Multi Tasks |              |            |
| Sensing          | Continuous  |             |              |            |
| Selection Technique | Greedy     |             |              |            |

| Distinguished Features | Decision-making | Hybrid | Local | Global | Task-based |
|------------------------|------------------|--------|-------|--------|------------|
| Recruitment Condition  | Page|QoS     | QoS   | QoS    | QoS        |
| Reputation Update      | Over Time/Task   | Over Time | Over Time | Over Time | Over Task |

TABLE 4. Simulation parameters.

| Simulation Parameter         | Value |
|-----------------------------|-------|
| Worker’s Parameters         |       |
| Workers Location (Lat, Long)| [31, ... 43] [129, ... 144] |
| Residual Energy             | [0.01, ... 1] Joule |
| CPU                         | [0.4, ... 1.0] GHz |
| Sensor Availability         | [0, ... 1] |
| Reputation                  | [0, ... 1] |
| Connectivity type           | 0 (Bluetooth), 1 (Wi-Fi) |

| Task Parameters             |       |
| Number of Tasks            | 5     |
| sensing period              | 30s   |
| Task Location (Lat, Long)   | [31, ... 43] [129, ... 144] |
| Number of Sensor type      | [2, ... 5] |
| Reputation                 | 0.5   |

| Validation Parameters |       |
|-----------------------|-------|
| α                     | 0.5   |
| Propagation Speed     | 30 m/s (Bluetooth), 100 m/s (Wi-Fi) |
| Bandwidth             | 6.4 bps (Bluetooth), 11 Mbps (Wi-Fi) |
| Bit_rate              | 2.1 Mbps (Bluetooth), 600 Mbps (Wi-Fi) |
| Transfer Data Speed   | 25 Mbps (Bluetooth), 250 Mbps (Wi-Fi) |
| latency               | 0.2 s (Bluetooth), 0.15 s (Wi-Fi) (average latency for Wi-Fi) |
| Data Size             | [2, ... 20] Gb |

applications is used to simulate the proposed framework. It includes data about the user’s device, such as energy and sensor availability [19]. In addition, the Stack Exchange Data Dump dataset [33] is used for the user’s Reputation. The rest of the parameters are randomly generated following a uniform distribution and following the communication technology standards [4], [19]. Multiple independent tasks are generated with different locations and requirements, including the number of required sensors per device, expected QoS, and the number of requested workers. Table 4 summarizes the simulation parameters, such as the AoI, sensing period, devices, communication technologies, and task requirements.

B. RESULTS AND ANALYSIS

1) RESULTS OVER TIME

This section evaluates the proposed framework and benchmarks over the sensing period and is presented in figure 3.

Reputation is the main parameter of the workers’ quality of service. Only workers with a high reputation are considered. A high workers’ QoS is achieved using a Hybrid-based approach as shown in figure 3a. Initially, the worker’s reputation is set to 0.5 for all approaches. The server and edge nodes know no historical information about the workers. For the first task along the sensing period, the Reputation is updated for all proposed solutions except for Task-based selection, where the reputation update is done after the first task is completed. However, as shown in figure 3b, the proposed approach outperforms the benchmarks where the workers’ Reputation increases over time and tasks. This is due to the adopted feedback mechanism relying on the workers’ contribution to overtime and task. At t=30s, the proposed approach provides a reputation that reaches a value close to 0.8. This new value is then used as an input for Task = 1, and this value increases till reaching a value more than 0.9 for task = 4.

2) RESULTS OVER TASK ID

This section evaluates the proposed framework and benchmarks over tasks’ IDs. In this work, the tasks are dependent since any update of workers’ Reputation in one task is also considered in future tasks as presented in figure 4. The Hybrid-based approach provides the highest workers’ Reputation compared to the benchmarks as presented in figure 4b. It starts with more than 0.7 in the first task, and this value increases till reaching a value of more than 0.8. However, Local-based and Task-based start with a value similar to the initial Reputation of 0.5. This value decreases for the Local-based and Task-based because they keep considering workers with low reputations since there is no cooperation between LENs. Without cooperation among LENs, the workers with anomalous readings are also included in the sensing process. Consequently, the QoS provided by the workers using Hybrid-based is higher than the benchmarks clearly shown in figure 4a. The benchmark provides a low QoS that does not exceed 0.4. However, the Task-based approach still performs better than the Global-based and Local-based approaches.

Figure 4c shows that the Hybrid-based approach provides low data variation compared to the benchmarks for all the tasks. This is related to the LENs cooperation that flags the workers with anomalous readings and removes them in future selections. However, the benchmarks have high variations because they keep considering workers with inconsistent data.

3) RESULTS OVER SIMULATION

This section evaluates the proposed framework and benchmarks over different densities. Every task requires a set of workers recruited to process the sensing and collect data. A worker can show trustworthy or untrustworthy behavior and provide accurate readings or anomalous ones. Table 5 wrap up the selection and the data-based anomaly detection process through the computation of the percentage of selected workers, percentage of trustworthy workers, and percentage of anomalous readings.

With a very low number of requested workers (e.g., 1000 every 30 seconds), Hybrid-based, Local-based, and Task-based cannot classify the workers as trustworthy or untrustworthy since the data variation is very low, close to zero. However, with the Global-based approach, the detection
is processed at the MEN level that receives all the LENs outcomes, increasing the probability of detecting anomalous data. Consequently, the Global-based approach is more suitable for data-based anomaly detection for low density. Moreover, it is clear that for the different numbers of requested workers, the Local-based approach selects fewer workers than the other approaches, with a percentage between 25% to 34%. Since the detection process is based on the consistency of the worker’s outcome and their contribution within the same cluster, local detection allows the discovery of more anomalies than global detection. Decreasing the Reputation of untrustworthy workers reduces the probability of choosing them in future selections. However, with the Global-based, Task-based, and Hybrid-based approaches, more workers are selected with an average of 90%. Then the probability of getting more trustworthy workers is achieved by more than 50%.

As illustrated in figure 5, the Hybrid-based approach shows a higher worker reputation than the benchmarks. This is due to the combination of local and global detection, which increases the reliability of the outcomes.
TABLE 5. Selection and detection percentages.

| Requested Work | Approach       | Selected Workers % | Anomalous Readings % | Trustworthy Workers % |
|----------------|----------------|--------------------|----------------------|-----------------------|
| 1000           | Hybrid-based   | 75                 | 0                    | 49                    |
|                | Global         | 75                 | 24                   | 0                     |
|                | Local          | 25                 | 0                    | 0                     |
|                | Task           | 75                 | 0                    | 0                     |
| 3000           | Hybrid-based   | 85.7               | 27.4                 | 38.3                  |
|                | Global         | 85.7               | 33.7                 | 52.0                  |
|                | Local          | 28.6               | 9.0                  | 19.6                  |
|                | Task           | 85.7               | 27.9                 | 37.9                  |
| 5000           | Hybrid-based   | 100.0              | 34.2                 | 65.8                  |
|                | Global         | 100.0              | 39.6                 | 60.4                  |
|                | Local          | 33.3               | 12.1                 | 21.2                  |
|                | Task           | 100.0              | 35.3                 | 64.8                  |
| 8000           | Hybrid-based   | 97.5               | 37.1                 | 60.4                  |
|                | Global         | 100.0              | 41.1                 | 58.9                  |
|                | Local          | 33.3               | 12.9                 | 20.5                  |
|                | Task           | 100.0              | 37.9                 | 62.1                  |

FIGURE 5. Average worker's reputation.

FIGURE 6. Average QoS.

FIGURE 7. Average delay (s).

FIGURE 8. Average energy consumption (J).

FIGURE 9. Coefficient of variation (%).

to the feedback mechanism adopted by the Hybrid-based, where the workers with inconsistent outcomes get a low cooperation score and then are excluded from the selection process. This allows for increasing the reputation when the number of requested workers is increased. In contrast, with Local-based, Global-based, and Task-based, the Reputation is decreasing because they keep considering the workers with low cooperation scores. These results impacted the QoS of the selected workers. The Hybrid-based approach provides the highest QoS compared to the benchmarks, as shown in figure 6. This is due to the higher Reputation of the workers and the closeness of the selected workers that require less propagation delay and energy, which positively impact the QoS.

Figure 7 depicts the average total delay for the four approaches, including communication and computation delay. Hybrid-based has a lower time delay than the other approaches due to the selection of the workers that rely on...
the QoS, including the propagation delay where the more the worker is close to the LEN more it has the priority to be selected. In addition to the feedback mechanism and the greedy selection, this helps to keep the most closed workers, which require less computation and propagation time to report the data. However, Task-based and Local-based spend approximately the same delay, while Global-based takes more time to compute and transmit the data. Processing all the computations at the MEN level is time-consuming for the computation and the transmission.

FIGURE 10. Resource consumption.

As is clearly illustrated in figure 8, increasing the number of requested workers leads to an increase in energy consumption. The Global-based approach has the smallest energy consumption compared to the other approaches. In global processing, the LEN oversees transferring of the collected data without processing it. In this case, the energy processing is not counted for at the LEN level, but it considers only the MEN capacity and the amount of data to be processed. However, the proposed approach consumes more power due to the deployment of the feedback mechanism that continuously updates the workers’ Reputation over time and tasks.

Figure 9 presents the coefficient of variation for the four approaches over the number of requested workers. As distinguished through the figure, when the number of selected workers is very low, the data is not varying from one source to another. The variation is close to zero for one reason: the selected workers are very close to the LEN and provide approximately similar information. However, when the number of requested workers increases, this also increases the possibility of having different data and then high variations. The Hybrid-based approach provides less variation than the benchmarks since the workers providing inconsistent data are excluded in the future selection.

C. FRAMEWORK RESOURCE CONSUMPTION

Three main metrics are measured, over different distributions, to evaluate the overall framework: Execution Time, CPU consumption, and RAM utilization.

Figure 10a shows the execution duration for the four approaches. Global-based and local-based show similar behavior in terms of execution time that decreases with the increase in the number of selected workers. However, the Task-based approach has an execution time from 900s to 1300s, while it is between 1000s and 1300s for the Hybrid-based approach. The response time of the Hybrid-based approach is significant compared to the other approaches for two reasons: First, the Hybrid-based requires extra time for the reputation update over time and task. Second, additional
time is needed for the filtration process based on the workers’ flags. With that, the approach still provides a good response time. Figure 10b shows the RAM utilization percentage. The four approaches allocate approximately the same memory during the simulation process. The results clearly show that RAM usage varies between 8% and 23%. The percentage is not linear with the increase of requested workers and depends more on the number of available workers within the cluster and their outcomes. In addition, All approaches require more CPU to run the selection process, as reflected in figure 10c. Hybrid-based and Local-based approaches require no less than 60% of the CPU for the computation and update process, and this value can reach more than 96%. However, the Task-based approach uses less CPU between 34% to 96%. Similarly, the global-based approach requires less CPU between 19% to 96%.

VI. CONCLUSION
Trustworthiness-based Data-Driven Decision-Making using smart Edge Computing for Continuous Mobile Crowdsensing is proposed to answer real-time requirements and Detect workers’ anomalous behavior when sensing the data in a continuous MCS environment. The proposed solution is a multi-layer framework consisting of first, selection, and clustering of ENs by edge server. Second, the edge server offloads the local selection of the workers and data assessment to LENs. Third, the MEN decision-making and cooperation between edge nodes using a feedback mechanism. Compared to the benchmark, the proposed framework improves the network performances by selecting optimal workers to perform the task with the highest Reputation, QoS, and providing a low coefficient of variation. In addition, the framework offers good resource consumption in terms of RAM and execution time. At the same time, it requires more CPU to handle the continuous updates of the workers’ reputations over time and tasks. The proposed approach can be improved to handle more complicated data problems in future work.

REFERENCES
[1] Q. Zhang, M. Lin, L. T. Yang, Z. Chen, S. U. Khan, and P. Li, “A double deep Q-learning model for energy-efficient edge scheduling,” IEEE Trans. Services Comput., vol. 12, no. 5, pp. 739–749, Sep. 2019.
[2] S. Nath and J. Wu, “Deep reinforcement learning for dynamic computation offloading and resource allocation in cache-assisted mobile edge computing systems,” Intell. Converged Netw., vol. 1, no. 2, pp. 181–198, 2020.
[3] X. Xu, H. Li, W. Xu, Z. Liu, L. Yao, and F. Dai, “Artificial intelligence for edge service optimization in Internet of Vehicles: A survey,” Tsinghua Sci. Technol., vol. 27, no. 2, pp. 270–287, Apr. 2022.
[4] H. Lamaazi, R. Mizoumi, H. Otrok, S. Singh, and E. Damiani, “Smart-3DM: Data-driven decision making using smart edge computing in hetero-crowdsensing environment,” Future Gen. Comput. Syst., vol. 131, pp. 151–165, Jun. 2022.
[5] A. Altaweel, C. Yang, R. Stoleru, S. Bhunia, M. Sagor, M. Maurice, and R. Blaock, “RSock: A resilient routing protocol for mobile fog/edge networks,” Ad Hoc Netw., vol. 134, Sep. 2022, Art. no. 102926.
[6] Y. Guo, H. Lamaazi, and R. Mizoumi, “Smart edge-based fake news detection using pre-trained BERT model,” in Proc. 18th Int. Conf. Wireless Mobile Comput., Netw. Commun. (Wi Mob), Oct. 2022, pp. 437–442.
[7] R. S. Alonso, I. Sittón-Candanedo, Ó. García, J. Prieto, and S. Rodríguez-González, “An intelligent edge-IoT platform for monitoring livestock and crops in a dairy farming scenario,” Ad Hoc Netw., vol. 98, Mar. 2020, Art. no. 102047.
[8] J. Yuan and X. Li, “A reliable and lightweight trust computing mechanism for IoT edge devices based on multi-source feedback information fusion,” IEEE Access, vol. 6, pp. 23626–23638, 2018.
[9] C. Yan, Y. Zhang, W. Zhong, C. Zhang, and B. Xin, “A truncated SVD-based ARIMA model for multiple QoS prediction in mobile edge computing,” Tsinghua Sci. Technol., vol. 27, no. 2, pp. 315–324, Apr. 2022.
[10] Z. Fang, J. Wang, J. Du, X. Hou, Y. Ren, and Z. Han, “Stochastic optimization-aided energy-efficient information collection in Internet of Underwater Things networks,” IEEE Internet Things J., vol. 9, no. 3, pp. 1775–1789, Feb. 2022.
[11] Z. Fang, J. Wang, Y. Ren, Z. Han, H. V. Poor, and L. Hango, “Age of information in energy harvesting aided massive multiple access networks,” IEEE J. Sel. Areas Commun., vol. 40, no. 5, pp. 1441–1456, May 2022.
[12] S. Salim, B. Turnbull, and N. Moustafa, “Data analytics of social media 3.0: Privacy protection perspectives for integrating social media and Internet of Things (SM-IoT) systems,” Ad Hoc Netw., vol. 128, Apr. 2022, Art. no. 102786.
[13] A. Alhusen and E. Arslan, “Avoiding data loss and corruption for file transfers with fast integrity verification,” J. Parallel Distrib. Comput., vol. 152, pp. 33–44, Jun. 2021.
[14] L. Xing and G. Levitin, “Balancing theft and corruption threats by data partition in cloud system with independent server protection,” Rel. Eng. Syst. Saf., vol. 167, pp. 248–254, Nov. 2017.
[15] F. Ullah, M. Edwards, R. Ramdhany, R. Chitchyan, M. A. Babar, and A. Rashid, “Data exfiltration: A review of external attack vectors and countermeasures,” J. Netw. Comput. Appl., vol. 101, pp. 18–54, Jan. 2018.
[16] D. Fadalakkarim and E. Bertino, “A-PANDEE: Advanced provenance-based anomaly detection of data exfiltration,” Comput. Secur., vol. 84, pp. 276–287, Jul. 2019.
[17] D. P. Jaiswal, H. Anand, S. M. Srinivasan, and M. Darayi, “A data-driven model to generate disruptive scenarios for infrastructure resilience studies,” Proc. Comput. Sci., vol. 185, pp. 248–255, Jan. 2021.
[18] Z. Yang, F. Liu, Z. Gao, H. Sun, J. Zhao, D. Janssens, and G. Wets, “Estimating the influence of disruption on highway networks using GPS data,” Expert Syst. Appl., vol. 187, Jan. 2022, Art. no. 115994.
[19] H. Lamaazi, R. Mizoumi, S. Singh, and H. Otrok, “A mobile edge-based Crowdsensing framework for heterogeneous IoT,” IEEE Access, vol. 8, pp. 207524–207536, 2020.
[20] R. Azzam, R. Mizoumi, H. Otrok, S. Singh, and A. Ouali, “A stability-based group recruitment system for continuous mobile crowd sensing,” Comput. Commun., J., vol. 119, pp. 151–169, Apr. 2018.
[21] X. Liu, F. Lu, H.-T. Zhu, B. Zhou, B. Xiao, and Y. Yan, “PrIDPM: Privacy-preserving dynamic pricing mechanism for robust crowdsensing,” Comput. Netw., vol. 183, Dec. 2020, Art. no. 107582.
[22] X. Chen, M. Simsek, and B. Kantarci, “Locally reconfigurable self organizing feature map for high impact malicious tasks submission in mobile crowdsensing,” Internet Things, vol. 12, Dec. 2020, Art. no. 100297.
[23] S. Li, W. Shen, M. Bilal, X. Xu, W. Dou, and N. Moustafa, “Fair and size-scalable participant selection framework for large-scale mobile crowdsensing,” J. Syst. Archit., vol. 119, Oct. 2021, Art. no. 102273.
[24] Y. Tang, S. Tasnim, N. Pissinou, S. S. Iyengar, and A. Shahid, “Reputation-aware data fusion and malicious participant detection in mobile crowdsensing,” in Proc. IEEE Int. Conf. Big Data (Big Data), Dec. 2018, pp. 4820–4828.
[25] A. Di Stefano, M. Scatà, B. Attanasio, A. La Corte, P. Liò, and S. K. Das, “A novel methodology for designing policies in mobile crowdsensing systems,” Pervas. Mobile Comput., vol. 67, Sep. 2020, Art. no. 101230.
[26] V. F. S. Mota, T. H. Silva, D. F. Macedo, Y. Ghamri-Doudane, and J. M. S. Nogueira, “Towards scalable mobile crowdsensing through device-to-device communication,” J. Netw. Comput. Appl., vol. 122, pp. 99–106, Nov. 2018.
[27] J. Duan, J. Lu, W. Jiang, and S. Yang, “Incentivizing fairness-aware task allocation in mobile crowdsensing with sweep coverage and stability control,” Appl. Soft Comput., vol. 97, Dec. 2020, Art. no. 106781.
[28] S. Upoor and M. Fiore, “Large-scale urban vehicular mobility for networking research,” in Proc. IEEE Veh. Netw. Conf. (VNC), Nov. 2011, pp. 62–69.
[29] S. Upoor, O. Trullols-Cruces, M. Fiore, and J. M. Barceló-Ordinas, “Generation and analysis of a large-scale urban vehicular mobility dataset,” IEEE Trans. Mobile Comput., vol. 13, no. 5, pp. 1061–1075, May 2014.
H. Lamaazi et al.: Trust-3DM: Trustworthiness-Based Data-Driven Decision-Making Framework

[30] H. Lamaazi, N. Benamar, and A. J. Jara, “RPL-based networks in static and mobile environment: A performance assessment analysis,” J. King Saud Univ.-Comput. Inf. Sci., vol. 30, no. 3, pp. 320–333, Jul. 2018.

[31] M. Sarwat, J. J. Levandoski, A. Eldawy, and M. F. Mokbel, “LARS: An efficient and scalable location-aware recommender system,” IEEE Trans. Knowl. Data Eng., vol. 26, no. 6, pp. 1384–1399, Jun. 2014.

[32] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel, “LARS: A location-aware recommender system,” in Proc. IEEE 28th Int. Conf. Data Eng., Apr. 2012, pp. 450–461.

[33] Stack Exchange Data Dump. Stack Exchange Data Dump, 2015.

HANANE LAMAAZI received the Ph.D. degree in network and computer sciences from the Faculty of Sciences, University of Moulay Ismail, Meknes, Morocco, in 2018. She was a Postdoctoral Fellow with the Center for Cyber-Physical Systems, Department of Electrical and Computer Engineering, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates. Her research interests include the Internet of Things (IoT), routing protocol (RPL), edge computing, and crowdsensing. She is a reviewer of several journals and a TPC member at different conferences.

RABEB MIZOUNI received the M.Sc. and Ph.D. degrees in electrical and computer engineering from Concordia University, Montreal, Canada, in 2002 and 2007, respectively. She is currently an Associate Professor with the Department of Electrical and Computer Engineering, Khalifa University of Science and Technology. Her research interests include deploying context-aware mobile applications, crowd sensing, software product line, and cloud computing.

HADI OTROK (Senior Member, IEEE) received the Ph.D. degree in electronics and communication engineering (ECE) from Concordia University, Montreal, Canada. He is currently an Associate Professor with the Department of ECE, Khalifa University of Science and Technology, and an Affiliate Associate Professor with the Concordia Institute for Information Systems Engineering, Concordia University, and the Electrical Engineering Department, Higher School of Technology (ETS), Montreal. His research interests include the domain of computer and network security, web services, ad hoc networks, application of game theory, and cloud security. He has co-chaired several committees at various IEEE conferences. He is an Associate Editor of the Ad-Hoc Networks (Elsevier), the IEEE COMMUNICATIONS LETTERS, and the Wireless Communications and Mobile Computing (Wiley).

SHAKTI SINGH (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in electrical and computer engineering from Purdue University, West Lafayette, IN, USA. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates. His research interests include semiconductor devices and integrated circuits, sensors, sensing technologies, crowdsourcing, crowdsensing, and the development of the IoT and wireless sensor networks.

ERNESTO DAMIANI (Senior Member, IEEE) joined the Information Security Group and Program, Khalifa University, as the Chair, and a Research Professor with the Etisalat British Telecom Innovation Center (EBTIC). He is on extended leave with the Department of Computer Science, Università Degli Studi di Milano, Italy, where he leads the SESAR Research Laboratory and the Head of the Ph.D. Program in computer science. His research interests include secure service-oriented architectures, privacy-preserving big data analytics, and cyber-physical systems security.