Hindawi
Security and Communication Networks
Volume 2021, Article ID 9678409, 11 pages
https://doi.org/10.1155/2021/9678409

Research Article

Enabling Fairness-Aware and Privacy-Preserving for Quality Evaluation in Vehicular Crowdsensing: A Decentralized Approach

Zhihong Wang,[1] **Yongbiao Li**,[1] **Dingcheng Li**,[1] **Ming Li**,[1] **Bincheng Zhang**,[2] **Shishi Huang**,[1] **and Wen He**[1]

[1]*The College of Information Science and Technology and the College of Cyber Security, Jinan University, Guangzhou 510632, China*
[2]*The College of Information, Xiamen University, Xiamen 361005, China*

Correspondence should be addressed to Yongbiao Li; liyongbiao@jnu.edu.cn

Received 9 June 2021; Revised 22 September 2021; Accepted 16 October 2021; Published 12 November 2021

Academic Editor: Leo Y. Zhang

With the rapid development of vehicular crowdsensing, it becomes easier and more efficient for mobile devices to sense, compute, and measure various data. However, how to address the fair quality evaluation between the platform and participants while preserving the privacy of solutions is still a challenge. In the work, we present a fairness-aware and privacy-preserving scheme for worker quality evaluation by leveraging the blockchain, trusted execution environment (TEE), and machine learning technologies. Specifically, we build our framework atop the decentralized blockchain which can resist a single point of failure/compromise. The smart contracts paradigm in blockchain enforces correct and automatic program execution for task processing. In addition, machine learning and TEE are utilized to evaluate the quality of data collected by the sensors in a privacy-preserving and fair way, eliminating human subject judgement of the sensing solutions. Finally, a prototype of the proposed scheme is implemented to verify the feasibility and efficiency with a benchmark dataset.

1. Introduction

Recently, mobile crowdsensing paradigm has significantly attracted attention from both the academic and industrial area [1, 2]. It is an essentially distributed problem-solving mechanism that leverages various sensing devices to collect valuable data in order to obtain a solution. There exist numerous famous crowdsensing platforms that cover from the global positioning system (GPS) to the weather report [3]. Particularly, due to the efficient sensing capability under the 5G network, mobile crowdsensing has been adopted in the vehicular network to assist autonomous driving [4]. It can be seen that, with the rapid development of mobile crowdsensing technology, the mobile sharing data in crowdsensing will play a critical role in digital society in the near future.

Generally, the architecture of mobile crowdsensing is mainly composed of three entities: the *requester, worker,* and a *crowdsensing service provider* (CSP). Specifically, a requester refers to the entity who has the requirement to obtain large scale data (or solutions) from the workers. He posts a task with certain incentives to the CSP. A worker refers to an entity who is equipped with a mobile sensor and willing to get a reward from the requester by providing valuable sensing data. The CSP mainly acts as an intermediary to receive sensing tasks from requesters and allocate them to the suitable workers. Despite the success of crowdsensing for accomplishing complex data collection tasks with the assistance of CSP, the requesters and workers are actually exposed to the potential threats on privacy and fairness issues [1, 4, 5].

A privacy issue: as for a requester, the sensing data collected from the workers are sensitive and valuable assets that he does not want to expose them to others, even for the CSP. This is because he has to pay for a solution, and such data may leak personal private information, e.g., his current location [6]. In practice, the sensing data is collected and stored by the CSP who is responsible for their security. Furthermore, if there exists any dispute between the requester and workers, the CSP will serve as an arbiter to judge the quality by accessing the data. As a matter of fact, CSP can be regarded as a trusted third party (TTP). That is, it will not preserve the privacy of the solutions and only give them to the requester who has paid for the workers. Unfortunately, numerous causes have shown that a fully TTP does not exist in reality. For instance, the leading IT company Facebook was reported to violate the privacy protection protocols again that it exposed 533 million personal data to the public website. Apart from the policy of the law, e.g., the General Data Protection Regulation (GPDR), additional technical measures should be designed to prevent the CSP from leaking the valuable data of the participants.

An unfairness issue: as for the quality evaluation of sensing data, there exists two intrinsic attacks that have an adverse effect on the fairness between the requester and worker, i.e., *false-reporting* and *free-riding* [7]. False-reporting is caused by a malicious requester who pays for workers after he has received the solutions. He may attempt to decrease the payment for workers by reporting that their collected data is low quality, no matter the real quality. On the flip side, free-riding refers to the attack that the requester pays for workers before he receives the data. In this payment model, a malicious worker may provide useless data after he receives the payment. In considering these two attacks, it is nontrivial to realize the fairness property in mobile crowdsensing.

To achieve privacy-preserving and fair quality evaluation in vehicular crowdsensing, there are some realistic challenges: (i) Firstly, neither the requester nor the CSP knows how to evaluate the collected data using a general purpose approach. Some simple crowdsensing tasks may know the types of the answer; e.g., a task has only a Boolean answer, while most of the tasks do not have exact answers or a range to be selected. In addition, (ii) the quality of workers' sensors may be different such that they may collect invalid or obviously wrong data. Several research efforts, such [7–10], have been made to tackle the quality evaluation while preserving privacy and fairness. Some of them utilize a truth discovery method which leverages a cloud to compute the result by performing secure computation [11]. Besides, the homomorphic encryption and zero-knowledge proof are used to protect the privacy of the data while obtaining the final computation result. Nonetheless, they either rely on a central TTP, or require high efficient cryptographic primitives. If a crowdsensing task is complex during the computation of the solutions, then these methods will introduce high computation cost. Other researchers attempt to handle this by introducing the reputation mechanism [7]; however, it requires the workers and requester to keep online for a long time. Dishonest workers or requesters may receive a task for once or register another account. Thus, we argue with the following: *can we design a privacy-preserving and fairness-aware quality evaluation method for complicated tasks in mobile crowdsensing?*

To tackle the abovementioned challenges, we proposed a decentralized quality evaluation scheme, named QuaEva, based on the blockchain technology, machine learning (ML), and trusted execution environment (TEE). In particular, QuaEva is designed atop our previous work [12] which utilizes the blockchain and smart contracts to accomplish a task for crowdsensing. More precisely, facing the challenges in terms of worker quality evaluation for complicated tasks, we leverage the off-the-shelf machine learning methods to evaluate a task solution without relying on human subjective judgement. The advantage lies in that if there are a large number of crowdsensing tasks to be mediated, then it can reduce the burden of human involvement. In the meanwhile, this design can provide an efficient way to improve the accuracy of the models. Current machine learning models can distinguish many objects with high accuracy, and the accuracy will grow gradually with the evolution of machine learning technology. Recently, a business model called machine learning model market arose [13, 14], which provides paid prediction service by leveraging the trained models. Moreover, to enforce fairness among requesters and workers without relying on a trusted third party (e.g., against a single point of failure), we resort to the blockchain and smart contracts. Therefore, QuaEva can be constructed as a decentralized architecture with immutability and decentralization. Furthermore, to preserve the privacy of the solutions, we introduce the TEE into the evaluation of sensing data quality. By doing so, a sensing data or solution can be evaluated within a TEE-secured environment. In a nutshell, our specific contributions can be depicted as follows:

(1) Privacy-preserving and decentralized quality evaluation framework: we propose a privacy-preserving and decentralized quality evaluation scheme named QuaEva, in which the privacy of sensing data can be preserved and the quality evaluation can be conducted in a decentralized way based on the blockchain and smart contracts.

(2) Fair quality evaluation without subjective grounds: by leveraging machine learning and TEE to mobile crowdsensing quality evaluation, we can detect useless sensing data in an efficient and fair way, instead of resorting to a TTP who might have subjective grounds to give the final results.

(3) Implementation on real world dataset: we implement a prototype of the proposed scheme on the real world dataset and conduct several experiments, demonstrating that QuaEva can achieve secure and fair quality evaluation in mobile crowdsensing.

The remainder of the paper is organized as follows. In Section 2, we present the background of blockchain and smart contract, machine learning, and TEE. In Section 3, we present the system model, security assumptions, and threat model. Then, in Section 4, the description of our proposed framework is given. In Section 5, we give the related work with quality evaluation in crowdsensing. The experimental results are presented in Section 6. Finally, we conclude in Section 7.

2. Background

In this section, we present the basic background of the building blocks used in this paper.

2.1. Blockchain and Smart Contract. Blockchain was first introduced by Nakamoto who aims to solve the issue of double-spending in Bitcoin, and it has been utilized in many applications [15–17]. It is essentially considered as a distributed ledger (DL) which consists of consecutive blocks. Each block mainly contains a block header and several transactions which happened recently. Compared with the classic distributed database, transactions in blockchain cannot be modified or deleted once they are recorded. In particular, a secure blockchain system satisfies the basic three security properties: *chain growth, chain quality,* and *common prefix* [18]. Based on such fundamental properties, the maintainers (also called blockchain nodes) of a blockchain system have a consistent overview of the blockchain. The initial blockchain system, i.e., Bitcoin, does not support Turing-complete smart contracts and thus has significant limitations when being applied in other applications, e.g., supply chain and decentralized finance (DiFi). Therefore, several efforts have been made to enable it to be adopted in different scenarios.

In fact, the key reason that blockchain has a huge influence in various areas is the capability of supporting smart contracts. Smart contract was first proposed by Nick Szabo in the 1990s. It refers to executing a program or an agreement automatically without illegal interference. In QuaEva, by leveraging smart contracts and the decentralized blockchain, we can enforce the process of crowdsourcing to complete quality evaluation without relying on a trusted party. Further, the collected sensing data can be guaranteed with integrity in the blockchain. The main challenge of blockchain-based crowdsourcing schemes lies in the quality evaluation of sensing data with fairness. Smart contracts can be utilized to prevent a crowdsensing system to give a subjective arbitration in case of dispute.

CrowdBC, proposed by Li et.al. [12] in 2018, is a blockchain-enabled decentralized framework for crowdsourcing. Compared with previous crowdsourcing platforms, CrowdBC allows requesters and workers to achieve fair crowdsensing tasks without relying on a central party by leveraging the underlying blockchain and smart contract technologies. More precisely, a requester posts a task by the Requester-Worker Relationship Contract (RWRC) which specifies the requirements of the task. Workers can accept this task by making a deposit in the RWRC. Besides, the cryptographic primitives are utilized to protect the privacy of the solutions. CrowdBC provides a future direction for crowdsensing by combining with the trustworthy blockchain technology. However, it leaves an open problem: *how to design a solution validation function that is able to evaluate the task solution correctly while preserving the fairness in the crowdsourcing.*

2.2. Machine Learning. With the advent of the digitalization era, machine learning (ML) has been recognized as a promising paradigm for data analysis and knowledge discovery. It is a branch of artificial intelligence (AI) [19]. A large amount of data has been generated and collected nowadays, providing us plenteous resources to train the accurate ML models. In addition, a collection of machine learning algorithms, such as deep learning, reinforcement learning, and federate learning [15], have been proposed that we can use them to make a prediction more easily. As a matter of fact, a well-trained ML model can be regarded as a *Judger* who can give more accurate judgement than the crowdsensing system in various applications, e.g., vehicle identification. Correspondingly, a large amount of data is collected that we can use to train a more accurate ML model, which has favorable results for both crowdsensing and ML [20, 21]. In particular, ImageNet is an image database that has collected hundreds and thousands of labelled images. The labelling task is completed in AMT, a public crowdsourcing platform. Many companies train some ML models and provide ML services for the public, such as Microsoft Azure and Google.

2.3. Trusted Executed Environment. Trusted Execution Environment (TEE) is an isolated secure environment which is designed to protect the confidentiality and integrity of loaded code and data. It can be used to prove the correct execution of a specific program. There are several types of TEE presently, such as Intel SGX and ARM TrustZone. Take the Intel SGX as an illustration; it creates a trusted sandbox environment called enclave in which a program can run securely. Specifically, a typical TEE protocol contains three phases: initialization, install, and resume. In the initialization phase, a key pair (the public key and private key) is generated by the hardware manufacturer, who embeds a private key into the enclave that no one can obtain. The install phase mainly loads a program *pram* to the TEE and outputs a (randomness) session id for identifying the *pram*. The last phase is responsible for executing *pram* with valid inputs. After the execution, the TEE outputs an attestation to authenticate the correct execution. Recently, there have been some attacks on TEE, e.g., side channel attack and rollback attack, while we argue that existing solutions are orthogonal with our work [22, 23].

Specifically, we list the operations of an enclave that will be used in our scheme. Scheduling operations: an instance of a SGX is scheduled by a host:

(1) $idx \leftarrow$TEE.install(pram): the host creates an enclave and starts an enclave instance by providing a

software code *pram*. The enclave returns a unique identifier idx after being created successfully.

(2) TEE.resume(idx, $inps$): this operation is used to resume the normal execution of an enclave with an input $inps$.

3. System and Security Model

In this section, we illustrate the system model and security model of QuaEva and specify the design goals. At the beginning, we list the notations used in this paper (cf. Table 1).

3.1. System Architecture. As shown in Figure 1, there are five types of roles in QuaEva: requester, worker, blockchain node, computation node, and storage node:

(1) *Requesters*, identified by $R = \{R_1, \ldots, R_n\}$, refer to the entities who post a crowdsensing task with the description of the tasks and an amount of rewards. R specifies the requirements of the data collection task in the description and converts the requirements as executable programs that will be loaded into the computation node (i.e., the TEE-powered host).

(2) *Workers*, identified by $W = \{W_1, \ldots, W_m\}$, refer to the entities who have certain type of mobile sensing device (e.g., a vehicle) as he intends to receive a sensing task for pursuit of task rewards. The capability of W is evaluated based on the historical data when W completes tasks.

(3) *Blockchain node*, identified by B_i, refers to a maintainer of the underlying blockchain. In QuaEva, a permissionless blockchain (e.g., Ethereum) or permissioned blockchain (e.g., Hyperledger Fabric) can be used to construct the underlying blockchain. Also, R and W can take part in the maintenance of blockchain as a blockchain node.

(4) *Computation node*, identified by C, refers to an entity who is empowered with a TEE that can provide the environment for secure computation. In a permissionless setting, the centralized cryptocurrency exchange can be recognized as such node. They mainly use their secure environment to provide secure computation services by attesting the computation results to the blockchain for a reward.

(5) *Storage node*, identified by S , refers to an entity that is responsible for storing the encrypted data. We do not specify a concrete data storage; any existing distributed storage system can be used in our scheme, e.g., the InterPlanetary File System (IPFS) [24].

As depicted in Figure 1, the basic architecture of our proposed vehicular crowdsensing system mainly consists of two components: (i) task management and (ii) solution management. The task management component is responsible for task posting, task receiving, and solution submission. The solution management is in charge of solution evaluation and reward payment. Specially, requesters interact with workers based on the decentralized blockchain system. They achieve their goals with fairness based on smart contracts.

Generally, the requesters and workers are required to register in QuaEva to get their credentials (i.e., a public key and private key) before participating in the data collection of mobile crowdsensing. By doing so, every participant can be evaluated by his/her historical behaviors. Specifically, when a requester posts a task, a ML model for evaluating its collected data is deployed in a TEE primarily. The requesters and workers are assumed to place dependence on this model to evaluate the data before a task begins. The hash value of a program and function are recorded in the blockchain, which can be used for workers to check the correctness of the ML results. In QuaEva, solutions (e.g., collected data) are verified by a TEE-powered server with a ML model rather than a third party, which is to decrease the security threats from false-reporting and free-riding attacks.

Security threats: in terms of security threats, we assume both requesters and workers might behave dishonestly. More concretely, a dishonest requester may attempt to reduce his cost by denying the contributions of workers after receiving the solutions, which is known as a false-reporting attack. On the other hand, a malicious worker may try to obtain the task rewards without contributing enough time and resources, which is known as a free-riding attack. These two attacks have an impact on the fairness of crowdsensing. In addition, we assume that a computation node might behave dishonestly or be compromised. In more detail, a malicious computation node C^* can feed old version data to the TEE, enforcing the ML model to give a low quality solution (data) evaluation, which is known as rollback (replay) attack [22]. Meanwhile, a compromised computing node can collude with a requester or a worker to gain profits.

Security assumption: here, we make the security assumptions as follows: the underlying blockchain system satisfies the majority of honest assumption; that is, given a majority of blockchain nodes are honest, the blockchain system can run with *persistence* and *liveness* [18]. These two basic security properties guarantee that a transaction posted by an honest user will be confirmed and become permanent after a period of time, e.g., 6 blocks in Bitcoin. In addition, the TEE executed in the computing node is secure in our scheme. We also observe that the side channel attack is a realistic attack that could leak the private key of the enclave. However, we argue that the protection mechanisms against side channel attack have been proposed recently and are orthogonal with our work. Besides, we assume that a ML model can return accurate evaluation results.

3.2. Design Goals. We summarize the security goals of QuaEva as follows:

(1) Privacy preservation: the privacy of workers' data, i.e., their collected data or completed task solution, can be preserved without relying on any central party

Table 1: The notations of explanation.

Notation	Explanation
λ	The system security parameter
$R = \{R_1, \ldots, R_n\}$	A set of requesters
$W = \{W_1, \ldots, W_m\}$	A set of workers
$[n]$, $[m]$	The tuple of $(1, \ldots, n)$, $(1, \ldots, m)$
$\{pk_{R_1}, sk_{R_1}\}$	The public and private key of a requester R_1
$\{pk_{W_1}, sk_{W_1}\}$	The public and private key of a worker W_1
$\{mpk, msk\}$	The public and private key of an Intel SGX enclave
$M_1 \| M_2$	The concatenation of messages M_1 and M_2
$H_0(\cdot)$	Noncryptographic hash functions
$\sum.KGen(1^\lambda)$	The key generation algorithm in the digital signature scheme
$\sum.Sign(M, sk)$	The signing algorithm on message M using the secret key in the digital signature scheme
$\sum.Verify(pk, \sigma, M)$	The verification of the signature σ using the public key and M in the digital signature scheme

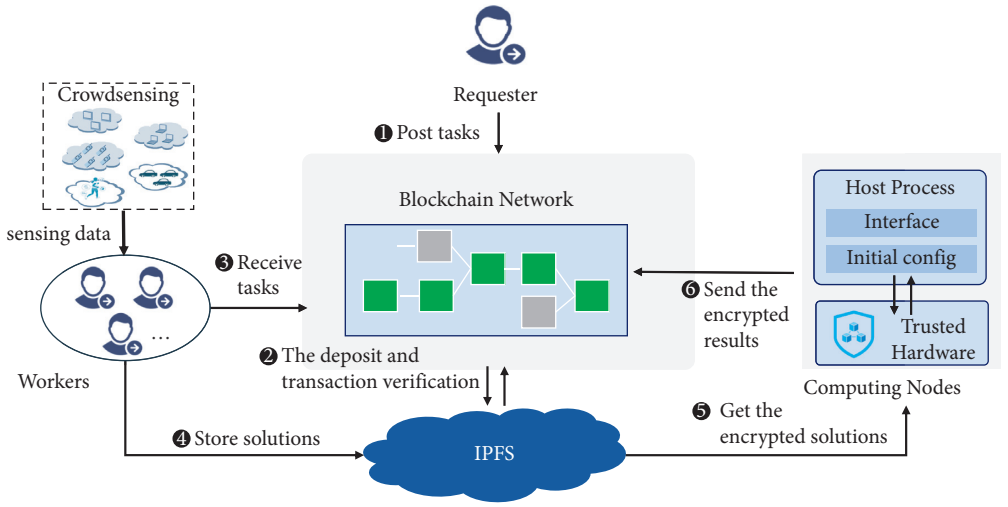

Figure 1: The system architecture of QuaEva.

(2) Fairness: our protocol can guarantee fairness by resisting the false-reporting and free-riding attacks [7]

(3) Reliable quality evaluation: the quality of the collected data during the mobile crowdsensing can be evaluated with a well-trained ML model which will give a reliable evaluation result to the workers

4. The Proposed Protocol QuaEva

4.1. The Design of QuaEva. Firstly, we present the overview design of the QuaEva protocol. To enable privacy-preserving and fairness solution evaluation, we leverage a TEE to construct an off-chain environment which can preserve the confidentiality and integrity of loaded programs. In practice, we adopt Intel's Software Guard Extensions (SGX) which has been widely used in both academic and industrial areas. The component of the computing node C contains two parts: a host process and an enclave, where the host process is mainly responsible for interacting with the blockchain and the distributed storage IPFS node, and the enclave refers to the secure environment for running the ML model.

As described in Figure 1, the proposed QuaEva protocol proceeds with four main phases: the tasking allocation phase, the task solving phase, the solution (data) evaluation phase, and the reward phase. During the first phase (steps 1–3), a requester posts a sensing task with some rewards to the RWRC contract. Meanwhile, he leverages the functionality of Intel SGX attestation to load a solution evaluation program prog to the enclave of C. Next, a set of qualified workers $\{W_1, \ldots, W_m\}$ who have registered in QuaEva (with deposits in RWRC contract) can receive this task. Specifically, these workers are required to satisfy the predefined conditions which are set in the RWRC contract; for instance, the value of reputation should be larger than a certain value. After completing the task, the workers can submit their solutions (or collected data) to the storage node S. Particularly, these data are encrypted under the public key of the enclave that can only be decrypted in the secure environment. In the meanwhile, a digest value of the solution is committed on the blockchain. Upon receiving the encrypted solutions, the enclave starts to verify them with the program prog . The evaluation result will be sent to the blockchain with a remote attestation by the computing node. The encrypted data is reencrypted in the TEE using the public key of the requester. According to the evaluation result, the reward assignment is executed automatically in the RWRC contact.

4.1.1. Smart Contracts Design. To avoid relying on a central party to conduct the quality evaluation of the solutions, QuaEva resorts to the smart contracts for ensuring the fairness and correctness of the process of crowdsensing. More concretely, the Turing-complete smart contracts enable us to depict any complex logic into contract code. Take Ethereum as an instance of the underlying blockchain system, a smart contract is converted into executable code in the EVM such that the miners can verify the correctness of the logic execution. One may think to use the smart contract to evaluate the solutions; however, due to the high on-chain transaction costs, it is unwise to put the evaluation work on-chain. Instead, we only write the core logic with smart contracts and store them in the blockchain layer, while other complex computations are put in the application layer. By doing so, we can significantly decrease the costs of on-chain transaction fees. In fact, at present, the smart contracts cannot support many complicated cryptography algorithms (e.g., Java and JavaScript); thus the off-chain solutions based on existing tools are a favorable choice in the blockchain. Different crowdsensing tasks have specific requirements; it is not easy to design on-chain quality evaluation functions for satisfying various requirements. To mitigate this challenge, we improve the smart contracts in CrowdBC. QuaEva also implements three types of smart contract: the User Register Contract (URC), User Summary Contract (USC), and Requester-Worker Relationship Contract (RWRC). Each smart contract is initialized and deployed in the blockchain for one time.

Specifically, to simplify the task posting, we devise a set of standard templates which are published by QuaEva for the tasks that have the similar logic. Each user registers with his/her addresses, profile, and pseudonym in the URC contract, where the address is corresponding to the public key of the user. In particular, in the RWRC contract, there exists a solutionEvaluate$(\cdot)$ function which is to evaluate the quality of a solution. This function can only accept inputs from the computing node C, which is achieved by verifying the signature of the transactions. Considering that C might behave dishonestly, the input is signed with the public key of the enclave. That is, the quality evaluation of a solution is realized privately in the enclave, and the result is sent to the RWRC contract with an authenticated attestation. In addition, there exists an algorithm checkWorkerQualification$(\cdot)$ that verifies the qualification of workers, e.g., checking if a worker satisfies the limited reputation value.

4.1.2. TEE-Powered Blockchain Oracle. Blockchain oracle serves as a data feed for a blockchain system. In QuaEva, the computing node C can be regarded as the blockchain oracle. More precisely, when a worker requests a reward payment after submitting the solution, he can send the request transaction to the blockchain, and the node C is notified by the "Event" mechanism in Ethereum. Then, C requests the encrypted data from the IPFS and sends it to the enclave for verification. The output of the enclave is sent back to the RWRC contract with remote attestation. Specifically, to defend against failure, we can build a distributed oracle network as in Chinklink [25], where a set of TEE-powered servers act as the blockchain oracle for feeding data.

4.2. Formal Protocol Specification. Compared with the conventional crowdsensing platform that utilizes money as a reward, QuaEva uses a cryptocurrency of the blockchain as the reward. Cryptocurrency can be obtained by mining or transacting with others. Following the assumption of [26], cryptocurrencies are fungible while they cannot be duplicated or forged. The owner who owns a private key can possess and transfer a cryptocurrency. Each party (a requester or a worker) has his own secure wallet to operate his coins. coins(v) is used to denote an item that the amount is v and coin$(v)_{t_0}$ denotes the cryptocurrency v to be locked in a smart contract for t_0 times. For simplicity, we utilize a supervised learning [20] to estimate the users' data qualities without knowing a ground truth (our approach can be extended to support other machine learning algorithms), where the quality level is labelled as $Q = \{q_1, \ldots, q_k\}$ in each task $\iota \in \{1, \ldots, k\}$ and q_ι represents the best quality. In particular, inspired by CrowdBC [12], we introduce the on-chain reputation management mechanism to evaluate the behaviors of workers, where a reputation value rep is updated periodically according to the historical tasks in smart contracts. As illustrated in Figure 2, the whole process of crowdsensing consists of five phases: initialization, task posting, task receiving, solution submission, and solution evaluation. In the following, we present the protocol specification.

4.2.1. Initialization. As a first step, each party who intends to participate in a crowdsensing task is required to register in the URC contract. By doing so, each participant can be evaluated with a historical profile, e.g., the skills, experiences, and task completion degree of workers. The public keys of registered parties are published in blockchain so that anyone can check the validation of an identity. Notice that QuaEva does not require a party to register an account using a true identity, but a pseudoanonymous account, which is essentially a similar design as in Bitcoin. In particular, someone might want to register with detailed personal information to increase the possibility of task receiving; QuaEva supports this type of information to be stored in the IPFS without introducing too much on-chain cost. Besides the user registration, the computing node C also obtains a key pair $(mpk_{\mathrm{TEE}}, msk_{\mathrm{TEE}})$ (which is usually embedded in the TEE by the manufacturer) and publishes the public key in this phase.

4.2.2. Sensing Task Posting. In this phase, a requester R_i can post a task to call for data from qualified workers. The task defines several parameters, including $\{\mathrm{des}, \mathrm{coin}(\pi_R + n * v_R)_{t_d}, mpk_{\mathrm{TEE}}, t_c, R_i^p, n, \mathrm{rep}\}$ and solutionEvaluate$(\cdot)$, where des refers to a short description of a sensing task, pk_R refers to the public key of a requester, n is the amount of required workers, solutionEvaluate$(\cdot)$ denotes a solution evaluation function which accepts inputs from the computing node C. Specifically, each evaluation function should be loaded into the TEE using TEE.install(solutionEvaluate$(\cdot)$) by the requester previously. A proof of the remote attestation is required to check the correctness of the evaluation function. It is committed on the RWRC contract that each worker can verify. Note that the evaluation function can only accept inputs from an attested

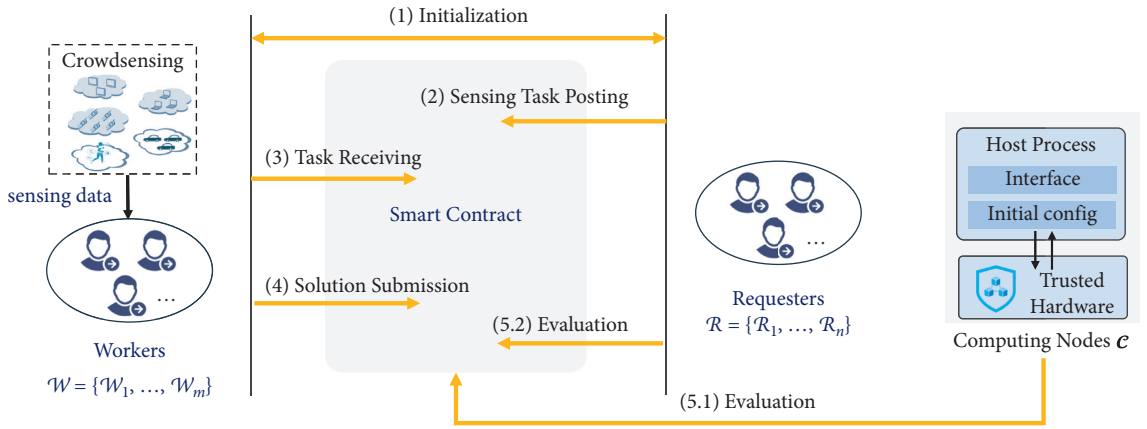

Figure 2: The process model of QuaEva.

secure processor TEE with a valid signature (using mpk_{TEE} to verify).

Specifically, the requester R_i needs to deposit a certain amount of cryptocurrency which is larger than the payment reward to achieve fairness. The remainder of the deposit will be sent back to the address of R_i if he behaves honestly. t_d refers to the time when workers are required to submit the data. t_c refers to the time when R_i is required to confirm the final evaluation results from the TEE. To prevent a low qualified worker from participating in this task, the requester can set a limited reputation rep in the RWRC contract.

4.2.3. Task Receiving. Then, a worker W_j can receive the task if he satisfies the condition of the task, e.g., a worker who drives a car and is present in the place where the requester intends to collect the data. Similar to the requester, the worker is also required to make a deposit in the RWRC contract. The deposit can be redeemed after the worker has submitted valid data in due time.

4.2.4. Solution Submission. After collecting the sensing data before t_d, the workers can submit their data to the storage node (i.e., the IPFS). In the meanwhile, a transaction with a data submission event is sent to the blockchain, and a hash value of the data is also committed in the RWRC contract. Note that an event of the RWRC contract is triggered to be sent to the C simultaneously (Ethereum solidity contract supports the Event mechanism.), which is to notify C that a sensing data has been submitted by a worker. Considering the privacy of the data, the worker encrypts the sensing data with the public key of the TEE (i.e., mpk_{TEE}) and signs it with his private key. The address of the data addr is sent to the RWRC contract so that the requester can download it.

4.2.5. Evaluation. In this phase, all submitted data are sent to the computing node C for quality evaluation. Specifically, due to the limited storage of TEE, we do not require C to evaluate these data simultaneously. More precisely, they can be split into multiple parts and evaluated separately. Inspired by proof of misbehavior in [27], as for the quality evaluation of data, it does not require all of the sensing data to be high quality, while if a part of it is evaluated as low quality, it represents that this worker provides a low quality data to this task and should not get reward from the RWRC smart contract. As mentioned before, the sensing data can be evaluated with a ML model, e.g., a capturing picture. The TEE decrypts the received data and determines the possibility of the accuracy in this task. C triggers the TEE to output an evaluation result that serves as an oracle for the blockchain. In particular, the TEE utilizes the public key of the requester to reencrypt the data after the evaluation, allowing the requester to decrypt the data by using his/her private key.

Afterwards, when the RWRC contract receives evaluation results from C, it checks if the results are signed with the TEE's public key, and it verifies the final results in solutionEvaluate$(\cdot)$ automatically. If there is no dispute between requester and worker on the solution result, the reward will be assigned to workers according to the output of the TEE. Otherwise, one of them who does not satisfy the result could post a transaction to the contract for arbitration by a third verifier. Such verifiers can only participate in the sensing task when there exists a dispute, and they can choose an authoritative party to act this third party previously.

4.3. Security Analysis

4.3.1. Fairness. In our scheme, we assume that there exists a predefined evaluation function that can automatically evaluate sensing data in the TEE. More precisely, assume that the solution is X, and then the miners can verify if solutionEvaluate$(\cdot)$ is equal to H or L according to the TEE. No one can modify the program which has been attested in the enclave. To reduce the size of on-chain data, the data can also be split into multiple parts, i.e., $X = \{X_1, X_2\}$. When the TEE outputs L result, it can upload a part of the data (e.g., X_γ) with the evaluation results to the RWRC contract. Thus, miners can verify such a part according to the output of the TEE. Notice that, during the process of task execution and result evaluation, there does not exist a trusted party to give subjective decisions. In addition, dishonest requesters or workers would be automatically punished by the RWRC

contract which pays the deposit to the other party. Based on the honest-majority assumption, the underlying blockchain system is secure, and the probability that malicious parties create a fork blockchain which is in their favor is negligible. Therefore, QuaEva is able to achieve fairness by using the trusted hardware and the blockchain technology.

4.3.2. Privacy Preservation. It is straightforward that the privacy of data can be protected once a trusted hardware is adopted. More concretely, to protect the privacy of data, our protocol requires that a worker submits an encrypted task solution with the public key of the TEE. The data is reencrypted with the public key of the requester in the TEE. All of the public keys are published and attested in the task. With the security assumption of TEE, the private key cannot be retrieved by anyone, even for the manufacturer. Therefore, the data in the TEE is protected from malicious users. In addition, the encrypted data is stored in the distributed storage and can be downloaded by a unique pointer which is committed on the blockchain.

4.3.3. Nonrepudiation. It is also straightforward that nonrepudiation can be achieved with the tamper-resistant blockchain technology. Specifically, the nonrepudiation represents that R and W should be authorized to post or receive tasks in the URC contract, and they cannot refute the participation in the latter. In addition, if a worker W_m submits a low quality solution with a poor contribution, W_m cannot deny the low quality submission, because it has been committed in the RWRC contract.

5. Implementation and Evaluation

In this section, we give the evaluation of QuaEva. As for the SGX environment, to avoid the complex development of Intel SGX based on SGX SDK (Intel SGX SDK for Windows v2.13.100.2), we initialize it with SGX SDK of version 2.5. We build a TEE environment on a server (Ubuntu18.04.4LTS, Intel(R) Core(TM) i5-7500 CPU @ 3.40GHZ). Specifically, we develop the secure computation program by using *python* programming language.

To show the practicability of the proposed scheme, we implement a local Ethereum test network in our server. We evaluate the performance of our scheme by considering the whole crowdsensing process. The transaction fee is defined as the same for different transactions. Specifically, there are 1 requester and 10 workers in our experiments. We conduct 100 times for the same picture capturing task and use the Multilayer Perceptron (keras) to recognize each collected picture in the TEE. The model used in the TEE is illustrated as in Table 2, where 4-layer MP networks are specified to conduct the image recognition.

As shown in Figure 3, we analyze the performance of on-chain transactions in the test network. Specifically, we record the time consumption for each transaction involved in the execution of quality evaluation. There are 6 types of transactions involved in the proposed scheme, i.e., deposit payment, task posting, data submission, evaluation requesting, evaluation submission, and proof of evaluation by TEE. The difficulty of the local Ethereum is relatively low as that each block is generated by taking about 4.756 seconds. Each block can only store about 376 transactions. We record the time cost that starts from a transaction being sent to the network and ends at it being written on the blockchain. Note that the average confirmation time for a specific transaction is about 9.428 seconds. Namely, each transaction takes about 2 blocks of time to be finally confirmed in the local Ethereum network. The average transaction throughput can be up to 70.08 TPS (transactions per second).

Specifically, during the process of quality evaluation, we load the image recognition model (i.e., the keras model) to the TEE [20]. Each sensing data of a task is encrypted from under the public key of the TEE and can be decrypted in the enclave in a privacy-preserving way. The evaluation output of the result is sent to the contract with an authenticated attestation. Specifically, we evaluate the performance of remote attestation, enabling an output from the TEE to be authenticated by the Intel Attestation Service (IAS). Each remote attestation takes 2.73 s on average, which is a little long compared with the execution of evaluation. However, we can combine a number of outputs as a whole result and then attest it from the IAS, which can significantly decrease the time cost.

In the TEE, we test 10,000 pictures which are collected from Minst https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz. The total time of the recognition takes 12,421 ms, which takes 1.24 ms on average for recognizing a single picture in the TEE. The accuracy of the recognition can be up to 97.67%. According to our experiments, we show that, by using smart contract, TEE, and machine learning technologies, it is able to achieve fair and privacy-preserving quality evaluation for the crowdsensing task.

6. Related Work

6.1. Crowdsensing. The concept of crowdsourcing was initialized by Howe in 2005 [28]. It contains several models such as crowdsensing, crowdfunding, and microcrowdsourcing. It represents a specific model in which individuals or organizations in all over the world are connected together, in which individuals are able to contribute their skills to obtain reward from a requester. As one of promising technologies, crowdsensing has attracted much attention over the past few years. The human intelligence-based crowdsensing consists of three groups of roles: requesters, workers, and a centralized crowdsensing system. It is mainly composed of three phases: data collection, data storage, and data upload. Currently, there exist lots of crowdsensing systems and their applications grow rapidly worldwide, such as WAZE, Google Maps, and Snapchat. These applications can collect valuable information such as weather and location.

6.2. Quality Evaluation. Despite the rapid development of crowdsensing, the issue of quality evaluation of the collected data has not been settled carefully and has drawn

Table 2: The description of Multilayer Perceptron used in the experiments.

Layer (type)	Output shape	Param
Flatten (flatten)	(None, 784)	0
Dense (dense)	(None, 128)	100480
Dropout (dropout)	(None, 128)	0
Dense_1 (dense)	(None, 10)	1290

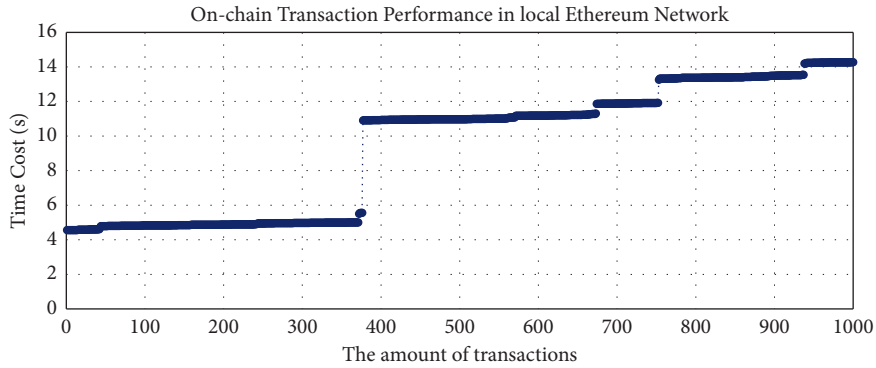

Figure 3: The time performance of a transaction to be recorded in the local Ethereum network.

much attention recently [2]. Many research works have been proposed that integrated an incentive, data aggregation, reputation management with data perturbation mechanism to achieve truthfulness and accurate aggregated results. Privacy preservation is a major research topic in crowdsensing, which is to protect the privacy of participants, e.g., location, name, and collected data. There exist three quality evaluation schemes in the crowdsensing [29, 30]. (i) The majority decision (MD): This scheme performs by aggregating all of the workers' data and the data that keep the same with the majority of participants will be the final output. (ii) The control group: This scheme generates a new validation crowdsourcing task for the collected data and selects several workers to perform the quality evaluation task. (iii) The gold standard: it proceeds by requiring workers to give a standard answer. Seldom ones of these schemes have considered the problem of worker quality evaluation and privacy preservation in a decentralized way. They mainly focus on the traditional crowdsensing architecture.

In addition, several schemes have been performed for quality control in crowdsensing, including incentive mechanism, worker selection, prior knowledge, and cheater detection. However, limitations have existed in prior schemes. Firstly, most of these methods focus on the simple tasks that can be evaluated by using the aggregation technique (AT) [31] or MD [30]. While it can address the scenario that the answers of sensing tasks have finite answers in the crowdsensing, as for the complicated skill-based tasks that do not have identical answers among the submitted result sets such as program development and diagram design, these methods cannot be applied. Secondly, requesters will generally afford some rewards for workers to get high quality data (solutions) generally. However, a main dilemma exists in the monetary incentive mechanism between the workers and the requesters. If the payment is paid before the task starts, workers may solve the task without effort, which is known as "free-riding" [7]. If the payment is paid after answers are submitted, the requester has the motivation to decrease the payment by giving an unreasonable evaluation, which is known as "false-reporting" [7]. Most current schemes are solving the dilemma based on the reputation system, but it is based on the hypothesis that workers and requesters may stay in the system for a long time, while this is not true that some dishonest users may use the crowdsensing system for one time. Lastly and importantly, the quality control approaches are executed in the crowdsensing system where workers and requesters believe that this central system will neither conduct dishonest activities nor be hit by the attackers. Unfortunately, it does not always be the case. Therefore, to accomplish the quality evaluation in a fair and privacy-preserving way, it is necessary to consider all of the security threats together.

6.3. Crowdsensing with Machine Learning. Machine learning has renovated many applications that attract considerable attention from both industrial and academic area [19, 20, 32–35]. There exist some works combining crowdsensing with machine learning. Guo et al. [34] proposed Bayesian-based predictive models that aim to accomplish the crowdsensing process within the crowdsensing architecture. Xiong et al. [35] proposed a crowdsensing method to collect large scale data to train the samples in machine learning. However, most existing methods have considered using machine learning to address the challenge of quality evaluation in the crowdsensing.

7. Conclusion

In this paper, we presented a fair-aware and privacy-preserving scheme for quality evaluation in crowdsensing. We analyze that the traditional quality evaluation functions are subjected to the weakness of human subjective intervention, single point of failure, and the complicated skill-based tasks cannot be evaluated by using conventional methods accurately. Hence, we solve the evaluation dilemma between the requester and workers with the trusted execution environment and machine learning based on our previous work [12]. Particularly, QuaEva does not rely on any third party to give judgement, and the data (i.e., the solutions) can be evaluated automatically with a committed evaluation function in the TEE. We believe that this design can provide a direction for the quality evaluation with ML and blockchain technologies.

Data Availability

The dataset analyzed during the current study are available in the Dataverse repository, https://www.kaggle.com/c/imagenet-object-localization-challenge/overview/description. These datasets were derived from the following public domain resources: https://github.com/lim60/crowdBC.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work of Zhihong Wang and Yongbiao Li was supported in part by the National Key Research and Development Plan of China under Grant 2020YFB1005600, in part by the National Natural Science Foundation of China under Grant nos. 61825203, U1736203, U2001205, 61732021, 62102166, 61902067, 62102165 and 62032025, in part by the Major Program of Guangdong Basic and Applied Research Project under Grant 2019B030302008, and in part by the National Joint Engineering Research Center for Network Security Detection and Protection Technology. The work of Dingchegn Li was supported in part by the National Natural Science Foundation of China. The work of Ming Li was supported in part by the National Natural Science Foundation of China, in part by Guangdong Basic and Applied Basic Research Foundation under Grant nos. 2020A1515111175 and 2019B1515120010, and by Guangdong Key Research and Development Plan 2020 under Grant nos. 2020B0101090002, 2020B0101090004, and 2020B0101360001.

References

[1] J. Ni, A. Zhang, X. Lin, and X. S. Shen, "Security, privacy, and fairness in fog-based vehicular crowdsensing," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 146–152, 2017.

[2] B. Zhao, S. Tang, X. Liu, and X. Zhang, "Pace: privacy-preserving and quality-aware incentive mechanism for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 1924–1939, 2020.

[3] X. Wang, Z. Ning, M. Zhou et al., "Privacy-preserving content dissemination for vehicular social networks: challenges and solutions," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1314–1345, 2018.

[4] Y. Hui, Y. Huang, Z. Su et al., "BCC: blockchain-based collaborative crowdsensing in autonomous vehicular networks," *IEEE Internet of Things Journal*, vol. 1, no. 1, p. 1, 2021.

[5] D. Dang, Y. Liu, X. Zhang, and S. Huang, "A crowdsourcing worker quality evaluation algorithm on mapreduce for big data applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 7, pp. 1879–1888, July 2016.

[6] X. Wang, Z. Liu, X. Tian, X. Gan, Y. Guan, and X. Wang, "Incentivizing crowdsensing with location-privacy preserving," *IEEE Transactions on Wireless Communications*, vol. 16, no. 10, pp. 6940–6952, 2017.

[7] Y. Zhang and M. Van der Schaar, "Reputation-based incentive protocols in crowdsourcing applications," in *Proceedings of the 2012 Proceedings IEEE INFOCOM. IEEE*, pp. 2140–2148, Orlando, FL, USA, March 2012.

[8] Z. Wang, J. Hu, R. Lv et al., "Personalized privacy-preserving task allocation for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1330–1341, 2018.

[9] J. Xiong, R. Ma, L. Chen et al., "A personalized privacy protection framework for mobile crowdsensing in iiot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2019.

[10] D. Liu, C. Huang, J. Ni, X. Lin, and X. S. Shen, "Blockchain-based smart advertising network with privacy-preserving accountability," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 3, pp. 2118–2130, 2020.

[11] A. Yang, J. Xu, J. Weng, J. Zhou, and D. S. Wong, "Lightweight and privacy-preserving delegatable proofs of storage with data dynamics in cloud storage," *IEEE Transactions on Cloud Computing*, vol. 9, no. 1, pp. 212–225, 2021.

[12] M. Li, J. Weng, A. Yang et al., "Crowdbc: a blockchain-based decentralized framework for crowdsourcing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 6, pp. 1251–1266, 2018.

[13] J. Weng, J. Weng, C. Cai, H. Huang, and C. Wang, "Golden grain: building a secure and decentralized model marketplace for MLaaS," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, p. 1, 2021.

[14] J. Weng, J. Weng, C. Cai, H. Huang, and C. Wang, "Fedserving: a federated prediction serving framework based on incentive mechanism," in *Proceedings of the IEEE INFOCOM 2021-IEEE Conference on Computer Communications. IEEE*, pp. 1–10, Vancouver, BC, Canada, May 2021.

[15] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, and W. Luo, "Deepchain: auditable and privacy-preserving deep learning with blockchain-based incentive," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2438–2455, 2019.

[16] M. Li, J. Weng, A. Yang, J.-N. Liu, and X. Lin, "Toward blockchain-based fair and anonymous ad dissemination in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11248–11259, 2019.

[17] M. Li, J. Weng, J.-N. Liu, X. Lin, and C. Obimbo, "Towards vehicular digital forensics from decentralized trust: an accountable, privacy-preserving, and secure realization," *IEEE Internet of Things Journal*, vol. 1, no. 1, p. 1, 2021.

[18] J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: analysis and applications," *Advances in Cryptology - EUROCRYPT 2015*, Springer, in *Proceedings of the Annual international conference on the theory and applications of cryptographic techniques*, pp. 281–310, April 2015.

[19] M. I. Jordan and T. M. Mitchell, "Machine learning: trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[20] X. Liu, W. Lu, W. Liu, S. Luo, Y. Liang, and M. Li, "Image deblocking detection based on a convolutional neural network," *IEEE Access*, vol. 7, pp. 26432–26439, 2019.

[21] A. Yang, J. Weng, K. Yang, C. Huang, and X. Shen, "Delegating authentication to edge: a decentralized authentication architecture for vehicular networks," *IEEE Transactions On Intelligent Transportation Systems*, vol. 1, pp. 1–15, 2020.

[22] S. Matetic, M. Ahmed, K. Kostiainen et al., "Rote: rollback protection for trusted execution," in *Proceedings of the 26th Usenix Security Symposium ({USENIX} Security 17)*, pp. 1289–1306, Vancouver, BC, Canada, April 2017.

[23] O. Oleksenko, B. Trach, R. Krahn, M. Silberstein, and C. Fetzer, "Varys: protecting SGX enclaves from practical side-channel attacks," in *Proceedings of the 2018 Usenix Annual Technical Conference ({USENIX} {ATX} 18)*, pp. 227–240, Boston, MA, USA, July 2018.

[24] J. Benet, "Ipfs-content addressed, versioned, p2p file system," 2014, https://arxiv.org/abs/1407.3561.

[25] L. Breidenbach, C. Cachin, B. Chan et al., "Chainlink 2.0: next steps in the evolution of decentralized oracle networks," 2021, https://chain.link/whitepaper.

[26] M. D. Bordo and A. T. Levin, "Central bank digital currency and the future of monetary policy," National Bureau of Economic Research, Tech. Rep., 2017.

[27] S. Dziembowski, L. Eckey, and S. Faust, "FairSwap," *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, vol. 1, pp. 967–984, Toronto Canada, October 2018.

[28] J. Howe, "The rise of crowdsourcing," *Wired magazine*, vol. 53, no. 10, pp. 1–4, Oct. 2006.

[29] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, "A survey on mobile crowdsensing systems: challenges, solutions, and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2419–2465, 2019.

[30] R. W. Ouyang, L. M. Kaplan, A. Toniolo, M. Srivastava, and T. J. Norman, "Parallel and streaming truth discovery in large-scale quantitative crowdsourcing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 10, pp. 2984–2997, Oct. 2016.

[31] G. Parent, H. Meng, G. A. Levow, M. Eskenazi, and D. Suendermann, *Crowdsourcing for Speech Processing: Applications to Data Collection, Transcription and Assessment*, John Wiley & Sons, Hoboken, New Jersey, United States, 2013.

[32] R. Kohavi and F. Provost, "Machine learning," *IEEE Transactions on Mobile Computing*, vol. 16, no. 4, pp. 934–949, 2017.

[33] G. Wang, T. Wang, S. Barbara, H. Zheng, and B. Y. Zhao, "Man vs. machine: practical adversarial detection of malicious crowdsourcing workers," in *Proceedings of the 23rd USENIX Security Symposium. Usenix Security*, San Diego, CA, August 2014.

[34] B. Guo, Z. Wang, Z. Yu et al., "Mobile crowd sensing and computing," *ACM Computing Surveys*, vol. 48, no. 1, pp. 1–31, 2015.

[35] H. Xiong, Y. Huang, L. E. Barnes, and M. S. Gerber, "Sensus: a cross-platform, general-purpose system for mobile crowdsensing in human-subject studies," in *Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing*, pp. 415–426, Heidelberg, Germany, September 2016.