
ISSN: 2321-2152
IJMECE
International Journal of modern
electronics and communication engineering
E-Mail
editor.ijmece@gmail.com
editor@ijmece.com
www.ijmece.com

ISSN2321-2152 www.ijmece

Vol 8, Issuse.3 Sep 2020

Crowdsensing using smart contracts that is both secure and private: Problems and solutions

B.Venkateswarlu[1],Dr.M.Bal Raju[2],Dr.M.Sreenivasulu[3],DrCh.Ramesh Kumar[4],

Abstract

With the emergence of Blockchain and smart contracts, numerous technologies and systems are empowered to automate commerce and make it easier to trade, monitor, and provide commodities, data, and services in a reliable and auditable manner. Blockchain and smart contracts. In the last several years, there has been a lot of interest in crowdsensing systems. Consumer electronics such as smartphones and Internet of Things gadgets are utilised to build large-scale sensor networks in crowdsensing systems. Here, we outline several critical aspects of crowdsensing systems that use smart contracts and Blockchain technology. With these systems, we also look at viable solutions that might solve serious security flaws. Businesses and social organisations are taking use of these gadgets for both profit and the benefit of the general population. Approximately 8 billion people throughout the globe have mobile phone subscriptions, with 5.5 billion of them having smartphones by the summer of 2020, according to current study (as of 2021). Increases in the number of IoT devices and 5G/6G networks are projected to drive these figures much higher in the years to come. Previously, we've seen crowdsensing systems used in environmental monitoring, transportation and entertainment, security, and healthcare. Peoplesensing systems have been established in several nations lately in response to the Coronavirus Disease 2019 (COVID-19) pandemic, not only to track the spread of the disease but also to cure it [3,4]. Crowd sensing is only one of several technologies that are having a profound effect on society. These are Blockchain and smart contracts. In addition to providing safe data storage, retrieval, and sharing, the blockchain provides a number of other services, including immutability and transparency as well as decentralisation and fault tolerance. [6]. These computer programmes, called smart contracts, allow Blockchain transactions to be automated via the declaration of business logic and code required to do certain activities when certain circumstances are satisfied [7]. With smart contracts, crowdsensing may increase data collecting and sharing as well as the establishment of decentralised marketplaces in which sensor data collectors can sell their data without the need for a central institution or broker [9,10]. There are other security concerns that must be addressed in this concept, though. These difficulties and their solutions are examined in this study.

Professor[1,2,3,4], Assistant Professor[1,2,3,4], ,Associate Professor[1,2,3,4],
Department of CSE Engineering,
Pallavi Engineering College,
Mail ID:bvenkat1109@gmail.com, Mail ID:drrajucse@gmail.com,
Kuntloor(V),Hayathnagar(M),Hyderabad,R.R.Dist.-501505.

Research contributions of this work

The following is a list of the most important findings from this study: • In this article, we examine the concepts of crowdsensing and smart contracts in more detail. When using smart contracts to enhance crowdsensing, we look at security and privacy concerns, and we provide solutions that may solve these concerns. Open issues that must be solved to deploy smart contracts for crowd sensing systems are discussed in this paper. The remainder of the paper is laid out as follows. There is a review of crowdsensing and smart contracts in Section 2. Smart contracts and crowd sensing technologies are examined in Section 3 for security concerns. Some proposed solutions to these problems are presented in Section 4. There are several unresolved concerns for future smart contract crowdsensing systems in Section 5**.**

Conclusions

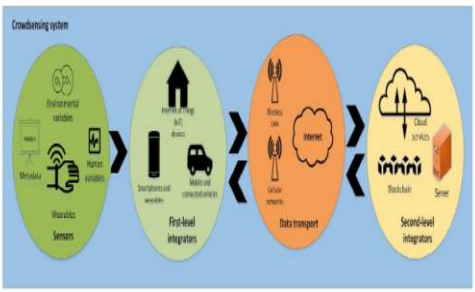
and future projects are discussed in Section 6. Smart contracts and the use of crowdsensing 2.1. Crowd-sensing technology. The Distributed Sensor Networks (DSN) programme launched in the United States in the 1970s was the beginning of current research in Wireless Sensor Networks (WSN). While this project utilised minicomputers and acoustic sensors, it was deemed state of the art at the time of its development. WSN technology and systems underwent a transformation as a result of the DSN project in the mid-to-late 1990s. Wireless networks comprising thousands of tiny devices could monitor broad regions of interest for months or even years if left unattended. Due to deployment and maintenance expenses, large-scale WSNs with thousands of units have not been practicable in practise, despite their theoretical potential. As a result of the billions of smartphones and other IoT devices owned by the general public, crowd-sensing systems have emerged in the first and second decades of the twenty-first century [12] to reduce the deployment and maintenance costs associated with the massive use of single WSN systems with thousands of devices. There are a wide range of applications for crowdsensing systems, from entertainment to transportation to environmental monitoring [13–19]. SARS-CoV-2, the virus that caused the Corona Virus Disease 19 (COVID-19) pandemic, prompted the adoption of crowdsensing devices created under the moniker "contact tracking applications" [3–5]. These systems avoid the high costs of deploying networks with thousands of sensors, particularly in metropolitan areas, by using consumer devices to perform sensing on a wide scale. The essential elements of crowdsensing systems are shown in Fig. 1 [2]. It's possible for sensors to gather data from observable real-world factors like as temperature and heart rate, as well as from human–computer interaction (HCI) or system operations (i.e., how much time a person logins to a website, oropens an application). Research and development of micrometer-scale machines (also known as Micro-Electro Mechanical Devices (MEMS) [20]) makes it feasible to integrate sensors for physical quantities into portable systems.

These devices take data from sensors and can conduct basic data filtering, data aggregation, and analysis. • First-level integrators. Smartphones, drones, and Internet of Things (IoT) gadgets are all examples of first-level integrator devices.Any communication technology that supports end-to-end communication is used in current crowdsensing systems, such as the Internet.Data from the first-level integrators is collected and analysed by second-level integrators. System integrators may either provide data to other organisations or assist users with data analytics, depending on the kind of system. These components are used by three types of users to gather data from first-level (or occasionally second-level) integrators through computer programmes. Included in this group are:

To acquire data from participants, task organisers are interested in deploying sensing tasks.nUser of a first-level integrator device doing sensing activities. • Participants.The data generated by a crowdsensing system may be of interest to other groups, such as government agencies.Data collecting in crowd sensing systems goes through a number of steps, as shown in Fig. 2.Participants are required to download sensingtasks from second-level integrators before task organisers may distribute them to participants.

• Data collection: First-level integrator devices are used to acquire data by performing sensing activities. First-level integrators may undertake data cleaning in addition to data collecting.

Here, first-level integrators broadcast data to second-level integrators either constantly or when contextual conditions are satisfied. • (i.e., reaching a specificlocation).



2

Fig. 1. Hardware components of crowdsensing systems.

In this step, the acquired data is analysed and shared by second-level integrator devices using Artificial Intelligence (AI) and statistical methodologies. According to the architecture of the crowd sensing system, participants or other entities may be given information created from the analysis of data. These difficulties, as well as remedies, may be found in Table 1 and Table 2, respectively. First- and second-level integrators have concerns about crowdsensing security and privacy. Issues relating to the confidentiality, integrity, and availability of data may be identified from a security standpoint. Security concerns include eavesdropping, data storage confidentiality, spoofing, authentication (for participants and sensors), exploiting operating system weaknesses, and denial of service attacks. Concerns about re-identification attacks, context-specific privacy (i.e., the identification of private situations), and exchanging data with external third parties are all important considerations from the standpoint of privacy. 2.2. Smart contracts and blockchain Using consensus or agreement among peers in a blockchain network, Blockchain implements a distributed ledger that stores data securely, immutably, and append-only [66]. Fig. 3 depicts the layers that make up a blockchain network's structure.

• A peer-to-peer (P2P) network: The P2P network guarantees that all blockchain nodes may communicate freely. There is no hierarchical organisation in this network of blockchain nodes.

The storage protocol is used to maintain the ledger in the global distributed ledger. For each individual user, he or she receives an individual public-key cryptographic pseudonym (address). A transaction is used to communicate between two addresses. Smart contracts are used to carry out data transactions on the global ledger. Applications: A blockchain network's Application Programming Interfaces (APIs) may be used for a wide range of applications. In addition to financial services, telemetry, copyright protection and digital document management platforms, there are a number of additional uses for these technologies. Figure 3 depicts the worldwide distributed ledger, which contains of cryptographically hashed blocks of data linked together. The system keeps (in all peers) transactions that have been validated using a predetermined set of criteria to identify whether transactions are genuine in any particular block in the chain. The blockchain records only legitimate transactions. To ensure the integrity of the data stored, a consensus algorithm conducted by all peers in the network chooses which block is to be chained to theledger [68]. This lets all peers to agree on a single version of the chain without a central authority. Various consensus algorithm models have been created, each with its own unique set of features and qualities. [69,70,71,72,73,74] Some examples of these models are PoF [69,70,71,72], PoS [71], PoA [72,73,74] and PoSpace [73,74]. As the first blockchain network to be publicly accessible, Bitcoin was founded in 2008 by someone (or a group of individuals) under the pseudonym Satoshi Nakamoto. Since then, a great deal of work has been done on blockchain technology in a variety of situations and scenarios. As of right now, there are two types of blockchain networks: open to the public and private. In the first category (public), these networks are accessible to the whole public, allowing anybody to join and run any form of application on top of them. Examples of typical public networks that are connected to the Internet

includeBitcoin and the3

Table 1
Security issues and solutions in crowdsensing systems.

Security issues			Solutions
Data confidentiality	Eavesdropping communication channels		Encryption through protocols such as Advanced Encryption Standard (AES) and Transport Layer Security (TLS)
	Data storage confidentiality		Database encryption using various methods at different granularities (e.g., table encryption, data encryption, disk encryption)
	Privacy		Different solutions based on the privacy issue (Table 2)
Data integrity	Spoofing		Estimation and filtering (e.g., methods such as kriging and Gaussian mixture models) [21,22]
			Anomaly detection (e.g., methods such as support vector machines, neural networks, Bayesian networks) [23]
	Authentication	User authentication	Biometric methods [24,25] Smart card authentication [26,27] Two-factor authentication [28–30]
		Sensor authentication	Secure brokering hardware [31,32] Trusted execution environments [33]
System availability	Availability at first-level integrators	Interference attacks on communication between sensors and first-level integrators	Frequency hopping, sensor repositioning, protocol modification, physical layer jamming avoidance (e.g., directional antennas, spread spectrum, and channel diversity) [34]
		Battery exhaustion attacks	Power-aware operating systems [35,36] Assessing power consumption of tasks before installation [37–39] Anomaly detection for power consumption at runtime [40,41]
		Operating system vulnerabilities	Static analysis [42] Dynamic analysis [43] Formal methods [44,45]
	Availability at second-level integrators	Elasticity	Hybrids between client-server and Peer-to-Peer (P2P) architectures [11] Cloud-based solutions [46,47]
		Denial of Service (DoS)	DoS countermeasures for cloud services and traditional network environments [48]

use of private clouds and intranets, and examples include R3 [77],and blockchain-as-a-service (BaaS) [78] corporations like IBM and Microsoft [79,80] establish their own platforms. Smart contracts are an add-on to the Blockchain technology. When it comes to automating transactions, smart contracts may be thought of as the processes that run on top of Blockchain. Blockchain-based smart contracts currently need the following elements:

For a certain blockchain platform, smart contracts are written in a specific programming language.

The sort of smart contracts that may be developed on a certain blockchain platform is determined by the programming language's properties (e.g., whether it is Turing complete). Solidity (forEthereum) and popular languages like Python, C++, GOlang, and JavaScript are examples of programming languages for smart contracts in blockchain systems.

In a platform based on distributed ledgers, smart contracts are stored and executed in the ledgers, storing data in the process. Ethereum is an example of a smart contract-enabled distributed-ledger platform.

It is a virtual computer that executes smart contracts at the network's edge, processing the contract's rules, even if they are recorded inthe distributed ledger itself. An example of a virtual computer for smart contracts is the Ethereum Virtual Machine (EVM). A smart contract's lifespan is shown in Fig. 4. A software developer creates the first contract in a computer language.

Table 2
Privacy issues and solutions in crowdsensing systems.

Privacy issues			Solutions
Re-identification	Re-identification from network identifiers		Disposable network identifiers [49] P2P anonymization [50] Double encryption (trusted broker) [51,52]
	Re-identification from task management	Authentication	Group authentication [51] Use of pseudonyms [52]
		Task distribution	Beacon-based distribution [51] Task downloading at crowded spaces [51] Anonymization network schemes [50]
		Data submission	Anonymization network schemes [50] Use of double encryption via brokers [51,52] Group-based signatures [51] k-anonymity [51] *l*-diversity [53] Use of pseudonyms [54] Micro-aggregation [55] Data aggregation [56]
Context privacy	Location privacy	Location-based queries	k-anonymity [57] Random noise in location submission [58] Use of known landmarks for location submission [59]
		Path privacy	Virtual fences [60] Fake locations [61] Cloaking regions [62]
	Sensor and metadata privacy		Allowing sensor data collection on task contexts only [63] Denying sensor data collection in contexts considered private [56]
External data sharing	Microdata release		k-anonymity [57] *l*-diversity [53] *t*-closeness [64]
	Statistical (summarized) data release		Differential privacy [65]

a creator or a creators (in this case a task manager). The deal was made public at that time. The smart contract will be executed on a participant's computer, which is a virtual machine linked to the blockchainP2P network. The smart contract will publish transactions back to the blockchain if it is required to do so by the smart contract. When Nick Szabo wrote about contractautomation [82], he was describing how to incorporate legally enforceable paper-based contracts in computer systems. Smart contracts were the result. Data sharing with everyone who met the contract's requirements was his objective. [67] Smart contracts, which are computer-executable, have the potential to be more useful than the legally binding paper-based contracts that came before them. Szabo conceived about smart contracts in 1997, but distributed ledgers and associated consensus mechanisms didn't come into existence until the end of the 2000s and early 2010s, when Szabo's original vision was realised [83]. In the current generation of smart contracts, blockchain technology is required. Because certain blockchains don't allow smart contracts (to be as functional as possible, as envisioned by Szabo). A good illustration of this is that, although Bitcoin is the most popular

blockchain (because to its usage in cryptocurrencies), it only enables the most fundamental forms of intelligent contract, such as exchanging bitcoin. Ethereum's smart contracts, on the other hand, are capable of supporting actions that go well beyond the simple exchange of cryptocurrencies [6]

Enhancing crowdsensing systems with smart contracts

Crowdsensing systems that rely on monetary incentives for data collection can benefit from the use of smart contracts in public blockchains, which create automated agreements between task organisers and participants that guarantee not only the completion of a data collection task, but also automated payments for those systems. Data may also be stored in the blockchain directly, ensuring that anybody can verify the tamper-proof guarantees provided by the blockchain. Crowdsensing systems with support for blockchain/smartcontracts fall into two broad types in terms of architectural models: Task organisers and participants in this category use smart contracts and blockchains to coordinate their sensing activities. In the blockchain, data acquired from participants and first-level integrators is saved and executed using smart contracts published by task organisers. Data from the blockchain may be downloaded by task organisers and participants can be compensated with bitcoin [84–88]. These crowdsensing systems use smart contracts and blockchains for certain of the activities (e.g., task allocation, data collecting, reward payment) whereas centralised crowdsensing architectures are used for other tasks [89–93]. Crowdsensing systems may benefit from the use of smart contracts, which can improve incentives, data quality, transparency, decen-tralization and fault tolerance. DDoS assaults, authentication, and privacy concerns may all be addressed by employing smart contracts and the blockchain technology (i.e., anonymization of users withoutusing third-parties because of their design).

Secure and privacy-preserving crowdsensing using smartcontracts: Issues

Using blockchain and smart contracts, we'll look at the challenges of developing crowd-sensing systems.

Before the development of general-purpose smart contracts as tools to enhance security, static analysis [42], dynamic analysis [43], and formal methods [44,45] for malware identification existed.

Static analysis's purpose is to identify any problems in the source code before it is executed [105]. Oyente is a smart contract tool that may be used for this purpose [94]. Based on the static analysis proposed by Luu et al., this tool [94] takes use of symbolic expressions to describe the programme variables and symbolic routes of smart contracts If a route fails to meet a constraint, it is called infeasible and discarded. An infeasible route indicates that the tool has discovered an issue with the application. Through the execution of code snippets, the purpose of dynamic analysis is to uncover faults and mistakes (or equivalent transformations). Manticore [106], Methryl [107], VerX [108] and KEVM [109] are instances of this approach in smart contracts. A smart contract is converted to a set of symbols and a set of symbolic pathways, which are then performed. Logic and requirements are used to demonstrate programme correctness in the third sort of method (formal meth-ods). F* functional programming language, VeriSol (111), VeriSolid (112) and SPIN (113) are only a few examples of formal specifications used in smart contracts. Static or dynamic code analysis isn't the only way to secure software in smartcontracts, but task managers' perceptions are. When task managers with poor conduct are punished by reputation in systems like SenseChain [87]CrowdBC [114], participants won't gather data on their behalf. For open mobile application marketplaces, similar ideas have been presented For instance, Android applications used to be published on official markets like Google Play without any type of vetting with limited success [115,116], which is no longer the case due to the implications of security violations and their consequences before they can be signalled as such. A.J. Perez and S. Zeadally Computer Science Review 43 (2022) 100450 Apps are presently being tested before they are released in the Google Play Market for Android devices. Integrity of data Participants who contribute fraudulent or misleading data for personal gain or to harm a system, whether accidentally or on purpose, violate data integrity, as previously discussed. Research in the literature has suggested ways to address this problem, such as using incentives and creating reputation measures for participants, requiring participants to submit some reimbursable deposits, and using trusted third-party verification [87–99]; these solutions have been found to be effective. This strategy (incentives and reputation) aims to give users with incentives if they contribute data that raises QoI, hence enhancing data integrity in the system[87]. Similar to this is the use of reputation metrics. By keeping track of participants' reputations and assigning assignments based on their ratings, task managers may ensure that only people who have a good reputation are assigned work. A subscription is required in the second category (reimbursable deposits) in order to gather data from participants. A monetary incentive is given to the participant, and the deposit is returned if the data is shown to be accurate. The task manager retains the deposit if the participant fails to provide high-quality data [114]. Also, the use of third-party verifiers has been suggested. Data integrity

requirements are met by using a third party to verify the participants' data in these systems. The task manager and participants may both put their faith in the third party that receives the data supplied by the participant [117]. 4.3. Protection of personal information ZKP (Zero Knowledge Proofs), external identity servers, and k-anonymity modifications are three of the most common ways to secure the privacy of crowdsenaring players utilising smart contracts. For example, Hawk [102] uses zero-knowledge proofs to verify the execution of smart contracts by storing transaction information encrypted on a blockchain. ZKP may be used to verify the execution of a smart contract while protecting the identities of the individuals involved. Second, systems employ external identity servers that let players to register outside the blockchain and get public/private keys created by a taskorganizer. The address that transactions on the blockchain utilise is created using these keys [114,118]. In the third category, smart contract and blockchain adaptations of k-anonymity (a mechanism for microdata release in databases [119]) have also been presented. When numerous participants trust each other, k-anonymity is utilised to establish k-anonymous groups and frameworks in which a single participant submits his/her acquired data to the blockchain under various blockchain identities (i.e., addresses) [120 and 121]. Using smart contracts to safeguard and protect the privacy of crowd-sensing will face future hurdles. 5.1. Smart contract software engineering techniques The usage of smart contracts in crowdsensing systems may expose participants to security and privacy problems due to a lack of standardisation in blockchain implementations. The absence of quality control in the blockchain development process might render it vulnerable to software flaws if standards aren't in place. There are no standards for evaluating the security or privacy safeguards in blockchain systems since each blockchain has its own set of protocols, specifications, programming languages, and tools. A minimal degree of security via design models is required for smart contracts, which are programmes that may make use of blockchain technology [122,123]. Therefore, greater study into the application of software engineering methods to smart contracts is required. Blockchain and smart contracts may be used to build more reliable systems for crowd sensing that participants can rely on. Smartcontracts, privacy, and the General Data Protection Regulation (GDPR) According to the European Union's General Data Protection Regulation (GDPR), online businesses collecting data from EU individuals regardless of their physical location must adhere to the same standards of privacy protection. The right to erasure [125] is one of the privacy safeguards provided by GDPR for EU individuals. It stipulates that EU people have the right to seek the deletion of any information on them that a service may have obtained (this right is alsooften known as the right to be forgotten). It is possible that if a European person engages in a blockchain-based crowdsensing system that collects human-centric data, such involvement may violate GDPR because of the blockchain's immutability, transparency and fault-tolerance properties. Several methods have been suggested to deal with this issue, including the use of smart contracts to prune a block from the blockchain if necessary [126,127,128]. Data may be wiped without the need for a hard-fork. For crowdsensing systems to meet legal criteria such as the right to erasure and other GDPR legal requirements and future privacy legislation, further research on smart contracts, digital wallets, and blockchains is required. Smart contracts based on blockchain technology for crowdsensing may be scaled up to a large extent. When it comes to the present version of smart contracts, a central authority isn't necessary since blockchains create an environment in which anybody can evaluate and verify the outcomes of a smart contract. Although current blockchains can validate, verify, and process a small number of transactions per second (Ethereum, while it supports general-purpose smart contracts, can processapproximately 15 transactions per second (around 1.3 million transactions per day) [129]), which does not scale well for the potential number of transactions that a crowdsensing system may generate. [129] Uber (a crowdsensing system for shareriding) completed 15 million trips per day in 2018 (about 176A.). If every ride is a completed transaction, as assumed by J. Perez and S. Zeadally in Computer Science Review 43 (2022), there will be 100450 transactions every second. That many transactions per second for a crowdsensing system would need an overhaul in the consensus methods and network design of a public distributed ledger to be scalable. [131]. When it comes to processing thousands of trans-actions per second, Ethereum began updating its protocols and network topology architecture in October 2021 by changing its con-sensus mechanism from Proof of Work (PoW) to Proof of Stake (PoS) combined with a technique to spread its network loadamong 64 parallel chains (a technique called shard chains) [132]. [132] In addition to reducing the overall power consumption of the whole Ethereum system, this move also increased the system's potential to grow. On the future years, it will be interesting to observe whether this sort of upgrade can effectively enable safe crowdsensing systems in public blockchains. Systems built on top of public blockchains will need more effort in order to grow. 5.4. Smart contracts for

crowdsensing systems without blockchain Because blockchains offer a tamper-proof environment for verifying a smart contract's functioning, execution, and outcomes without a central authority, the first generation of smart contracts were built. Smart contracts without the use of a blockchain might be an option as well. Ultimately the wearable/mobile device executes code in an untamperable environment called a Trusted Execution Environment (TEE) that proves that a transaction was successfully and securely completed every time a user pays for services or goods using an application for mobile payment (such as a mobile wallet or an NFC-enabled wearable/mobile device). Decentralized ledgers are not used to record the outcomes of mobile payment applications, but they may still be validated by a third-party (like a bank) and legally binding. These smart contracts may be used to create decentralised data marketplaces for crowd-sourced sensors, allowing participants to sell sensor data and be compensated in bitcoin. [6]. Using fiat money [133] to pay for incentives to engage in data markets, on the other hand, may not need the usage of blockchains. The creation of a specification for smartcontracts without a blockchain is still an active area of study. As a last point, These intriguing new applications have come to light because to the rise of Blockchain, cryptocurrencies, and smart contracts in recent years. Systems that use crowdsensing may greatly benefit from the use of smart contracts and blockchain technology. These systems need a high level of security and privacy. Some of these problems were discussed, as well as some potential solutions. Finally, we've outlined upcoming research difficulties that must be solved in the future in order to implement safe and private crowdsensing systems that make use of smartcontracts and blockchain technology.

References

1] A.J. Perez, S. Zeadally, Recent advances in wearable sensing technologies,Sensors. 21 (20) (2021) 6828.

[2] A.J. Perez, S. Zeadally, Design and evaluation of a privacy architecture forcrowdsensing applications, ACM SIGAPP Appl. Comput. Rev. 18 (1) (2018)7–18.

[3] D.A. Drew, L.H. Nguyen, C.J. Steves, C. Menni, M. Freydin, T. Varsavsky,C.H.Sudre, M.J. Cardoso, S. Ourselin, J. Wolf, T.D. Spector, Rapid imple-mentation of mobile technology for real-time epidemiology of COVID-19,Science (2020).

[4] H. Cho, D. Ippolito, Y.W. Yu, Contact tracing mobile apps for COVID-19: Privacy considerations and related trade-offs, 2020, arXiv preprintarXiv:2003.11511.

[5] J.H. Wright, R. Caudill, Remote treatment delivery in response to theCOVID-19 pandemic, Psychother. Psychosom. 89 (3) (2020) 1.

[6] Y. Kurt Peker, X. Rodriguez, J. Ericsson, S.J. Lee, A.J. Perez, A costanalysis of internet of things sensor data storage on blockchain via smartcontracts, Electronics 9 (2) (2020) 244.

[7] I. Bashir, Mastering Blockchain: Distributed Ledger Technology, De-centralization, and Smart Contracts Explained, Packt Publishing Ltd,2018.

[8] J.S. Park, T.Y. Youn, H.B. Kim, K.H. Rhee, S.U. Shin, Smart contract-basedreview system for an IoT data marketplace, Sensors 18 (10) (2018) 3577.

[9] A. Javaid, N. Javaid, M. Imran, Ensuring analyzing and monetizationof data using data science and blockchain in lot devices, (Doctoraldissertation, MS thesis), COMSATS University Islamabad (CUI), Islamabad44000, Pakistan, 2019.

[10] C.Y. Chong, S.P. Kumar, Sensor networks: evolution, opportunities, andchallenges, Proc. IEEE 91 (8) (2003) 1247–1256.

[11] A.J. Perez, M.A. Labrador, S.J. Barbeau, G-sense: a scalable architecture forglobal sensing and monitoring, IEEE Netw. 24 (4) (2010) 57–64.

[12] A.T. Campbell, S.B. Eisenman, N.D. Lane, E. Miluzzo, R.A. Peterson, People-centric urban sensing, in: Proceedings of the 2nd Annual InternationalWorkshop on Wireless Internet, 2006, p. 18.

[13] E. Kanjo, Noisespy: A real-time mobile phone platform for urban noisemonitoring and mapping, Mob. Netw. Appl. 15 (4) (2010) 562–574.
[14] W.Z. Khan, Y. Xiang, M.Y. Aalsalem, Q. Arshad, Mobile phone sensingsystems: A survey, IEEE Commun. Surv. Tutor. 15 (1) (2012) 402–427.

[15] N.D. Lane, S.B. Eisenman, M. Musolesi, E. Miluzzo, A.T. Campbell, Urbansensing systems: opportunistic or participatory? in: Proceedings of the9th Workshop on Mobile Computing Systems and Applications, 2008, pp.11–16, http://dx.doi.org/10.1145/1411759.1411763.

[16] N.D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, A.T. Campbell,A survey of mobile phone sensing, IEEE Commun. Mag. 48 (9) (2010)http://dx.doi.org/10.1109/MCOM.2010.5560598.

[17] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, L. Selavo, Real timepothole detection using android smartphones with accelerometers, in:Proceedings of 2011 International Conference on Distributed Computingin Sensor Systems and Workshops (DCOSS), 2011, http://dx.doi.org/10.1109/DCOSS.2011.5982206.

[18] D. Mendez, A.J. Perez, M.A. Labrador, J.J. Marron, P-sense: A participatorysensing system for air pol-lution monitoring and control, in: Proceedingsof the 2011 IEEE International Conference on Pervasive Computingand Communications (PERCOM), 2011, pp. 344–347,

[19] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E.Howard, R. West, P. Boda, PEIR, the personal environmental impact report,as a plat-form for participatory sensing systems research, in: Proceedingsof the 7th International Conference on Mobile Systems, Applications,and Services (MobiSys '09), 2009, pp. 55–68,

[20] M. Gadel-Hak (Ed.), The MEMS handbook, CRC Press, 2001.[21] D. Mendez, M. Labrador, K. Ramachandran, Data interpolation forparticipatory sensing systems, Pervasive Mob. Comput. 9 (1) (2013)132–148.