AI-Enabled Blockchain Consensus Node Selection in Cluster-Based Vehicular Networks

Khalil Saadat, *Member, IEEE*, Ning Wang, *Senior Member, IEEE*, and Rahim Tafazolli, *Senior Member, IEEE*

***Abstract*—In the scenario of highly mobile nodes, applying blockchain not only needs to ensure high performance and security but also needs to consider the low stability of consensus nodes. We introduce a novel framework to reduce the negative impact of low-stable consensus nodes. In addition, we have developed an intelligent algorithm for selecting the optimal number of Mobile Consensus Nodes (MCNs) considering various parameters including Stability of Node (SoN). The results demonstrate that the proposed scheme improves the average reputation and stability of consensus nodes by 6.8% and 17.5%, respectively while reducing the average message counts by 33.9%.**

***Index Terms*—Blockchain, AI, ANNs, consensus node selection algorithm, machine learning, mobile consensus nodes.**

I. Introduction

IN THE past few decades, communication networks have evolved to deal with the additional complexity of emerged applications. Designing such a modern network requires special attention to the network dynamicity in the presence of highly mobile nodes. The next-generation verticals, for example, Internet of Things (IoT) and Internet of Vehicles (IoV) encompass highly mobile nodes including buses and trains. Besides, more devices with sensitive data have joined the network. Hence, it is required to design a more secure solution for storing application-specific contents and exchanging configuration messages.

Traditional data storage solutions may not always satisfy the basic requirements of modern networks including security, availability, transparency, traceability, non-repudiation, integrity, and automation. For instance, the centralised network architecture makes the IoT and IoV nodes vulnerable to single-point-of-failure and distributed-denial-of-service attacks [1]. Blockchain, as an emerging technology, is capable of satisfying these basic requirements by establishing trust among untrusted parties. The next-generation networks can benefit from blockchain capabilities, but this novel technology still may not be a suitable alternative to traditional systems due to its scalability and flexibility issues. Existing blockchain scalability solutions have attempted to alleviate the impact of bottlenecks that exist in various blockchain layers. The scalability solutions can be categorised into three layers. (1) layer zero solutions which enhance the data propagation model, (2) layer one solutions such as Directed Acyclic Graph and Sharding which are also called on-chain solutions, and (3) layer two solutions such as side-chain, off-chain and cross-chain which are known as non-on-chain solutions [2], [3]. A common concept behind some of the above solutions is to split the network into multiple clusters (sub-networks).

Another issue in the context of blockchain is the flexibility of blockchain networks in environments which mandates cross-cluster movements. Mobile Roadside Units (mRSUs) have already been proposed by other researchers which can potentially reduce the high deployment and operational costs of Fixed Roadside Units (fRSUs) [4], [5]. Applying blockchain in such dynamic scenarios of Mobile Consensus Nodes (MCNs) requires intelligent mechanisms to address frequent cross-cluster movements in the presence of low-stable nodes. This is because every cross-cluster movement imposes additional overheads in terms of retrieving the new cluster's ledger and reassigning it to the new cluster's consensus nodes. Therefore, selecting low-stable MCNs who are willing to leave the cluster early should be avoided.

In this letter, we propose a novel framework called Mobility-aware Reputation-based Blockchain Framework (MRBF) to address the aforementioned issues of MCNs. In our framework, we introduce an Intelligent Consensus Node Selection Algorithm (ICNS) based on the Artificial Neural Network (ANN) to select the highly-reputed MCNs in a given cluster based on nodes' live and historical reputation information. In specific, we target low-stable MCNs with predictable mobility patterns. The reputation values of MCNs are built based on various parameters including Stability of Node (SoN). ICNS utilises an intelligent dynamic reputation threshold which is carefully designed to change dynamically according to the live number of MCNs in a given cluster. The intelligent threshold considers the blockchain performance and security metrics.

The contributions of our work are as follows: (1) We propose a novel framework designed for cluster-based environments with multiple mobile nodes involved in the consensus process. To the best of our knowledge, this is the first work which argues that the mobility patterns of consensus nodes can be utilised in the consensus node selection process. (2) We propose an ANN-based scheme for selecting MCNs based on crucial parameters including node's stability. The offline ANN model with high accuracy proves that the model can be used to assign high-accurate reputation values to MCNs. (3) We carried out performance evaluations to quantify the benefits of our scheme including the intelligent dynamic reputation threshold. Our initial results prove the capability of our scheme for selecting the optimal number of MCNs which reduces the average message counts by 33.9% while improving the average reputation and SoN values by 6.8% and 17.5%, respectively.

Manuscript received 2 December 2022; accepted 13 January 2023. Date of publication 23 January 2023; date of current version 6 June 2023. The associate editor coordinating the review of this article and approving it for publication was N. Passas. *(Corresponding author: Khalil Saadat.)*

The authors are with the 5GIC and 6GIC, Institute for Communication Systems, University of Surrey, GU2 7XH Guildford, U.K. (e-mail: khalil.saadat@surrey.ac.uk; n.wang@surrey.ac.uk; r.tafazolli@surrey.ac.uk).

Digital Object Identifier 10.1109/LNET.2023.3238964

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

II. Background and Related Work

At the core of every blockchain system is a consensus mechanism (CM). The ultimate goal of all CMs is to reach an agreement among untrusted parties in a distributed manner. Typically, a consensus is reached when some pre-defined conditions are met. After reaching a consensus, the data (in the format of transactions and blocks) is stored in the distributed ledger. We can divide the CMs into two categories: (1) conventional CMs; and (2) alternatives to conventional CMs. An example of conventional CMs is practical Byzantine Fault Tolerance (pBFT) which is more suited for small-scale and state-machine replication scenarios. Likewise, pBFT consensus is reached based on majority voting which requires knowing the number of nodes involved in the consensus process, so it is most suited for private or permissioned scenarios.

Among alternatives to conventional CMs, reputation or trust-based CMs have received considerable scholarly attention in recent years. This category is widely proposed for emerging application scenarios such as vehicular communications. The reputation or trust-based CMs are designed to improve blockchain's performance without sacrificing its security [6]. In principles, reputation or trust-based CMs are meant to consider a more concrete set of parameters for selecting blockchain consensus (miner) nodes. RBFT [7] and PoT [8] are examples of related works which attempted to select the consensus (miner) node based on pre-defined reputation/trust models. Another work [9] examines the scenario in which the geographical location of nodes is taken into account to prevent Sybil attacks and to decide whether to include a node in the "whitelist" or "blacklist." In contrast, our current work focuses on problems associated with assigning blockchain consensus tasks to the mobile consensus nodes (MCNs). In specific, we address the problems associated with low-stable MCNs.

III. Mobility-Aware Reputation-Based Blockchain Framework (MRBF)

A. Problem Formulation

First, as per our literature review, we have found out that existing CMs do not directly involve node stability in the selection of consensus nodes. However, application scenarios including vehicular communications may involve MCNs. The idea of mobile RSUS (mRSUs) have already been proposed by other researchers including [4] and [5]. MCN's mobility patterns may differ depending on the nature of MCN implementation which can be either predictable (up to some extent) or completely random. For instance, buses have predictable timetables with some level of inaccuracy (i.e., bus delays). In this letter, we argue that predictable mobility patterns of MCNs can be utilised in the blockchain design to select the most stable nodes.

Second, some researchers have proposed a variety of solutions including Sharding to split the entire blockchain network into multiple clusters (also called shards, geographical areas, or districts) to address the blockchain scalability issue. In the context of the dynamic environment of MCNs, we assume that the entire blockchain network is divided into multiple clusters, each dedicated to a specific geographical area traversed by the MCNs (for instance, two clusters as depicted in Figure 1). As

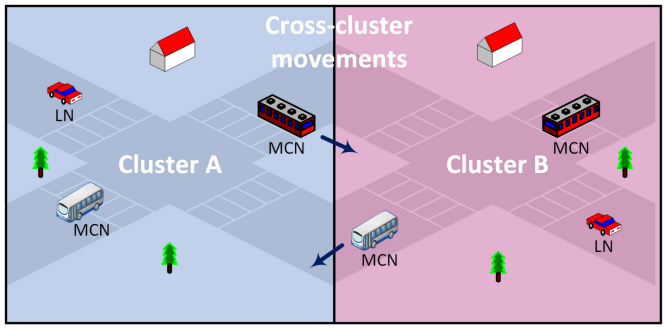

Fig. 1. Cluster-based blockchain in the dynamic environment of mobile consensus nodes (MCNs) and light nodes (LNs).

MCNs are moving frequently, there are occasions in which nodes are either leaving an existing geographical area (cluster) or joining a new geographical area (cluster).

Third, in the context of blockchain CMs, we consider reputation-based CMs as suitable alternatives to conventional CMs. In reputation-based CMs, consensus nodes with higher reputation values are regarded as more trusted nodes. Therefore, consensus nodes with higher reputation values are involved in the consensus procedures (i.e., block validation) more than other nodes with lower reputation values. The reputation-based CMs are proposed by other researchers, but we consider the unique characteristics of MCNs in reputation calculation.

As per the aforementioned three key findings, we formulate our problem in a cluster-based environment in which multiple MCNs are moving from one cluster to another. The frequent movements make a dynamic environment of moving nodes along with some uncertainty in terms of the instability of MCNs. In the following subsection, we define key components of our framework (MRBF) to tackle the aforementioned problems. As illustrated in Figure 1, we simplify the problem scenario in which a geographical area is split into two distinct districts (clusters). This letter is based on our initial problem formulation on reconfigurable blockchains for dynamic cluster-based applications [10].

B. Assumptions and Framework Components

In our proposed framework (Figure 2), we assume two types of blockchain nodes: (1) Mobile Consensus Node (MCN) and (2) Light Node (LN). MCN is defined as consensus nodes which are moving due to the nature of their implementation. For example, buses within a geographical area can become edge nodes, and blockchain can be a subcomponent of each individual edge node. Accordingly, MCNs are actively involved in the blockchain consensus process of collecting transactions, generating blocks, propagating blocks, rejecting/approving proposed blocks, and storing the approved blocks. In our definition, MCNs are nodes with certain movement patterns. These patterns are recognised based on the nature of the mobile nodes such as buses and trains that take similar routes. On the other hand, LNs are blockchain nodes which are typically less powerful in terms of computation, communication and storage capabilities. Therefore, LNs are not directly involved in the CM of chaining blocks. Instead, LNs, for example, autonomous vehicles, submit new

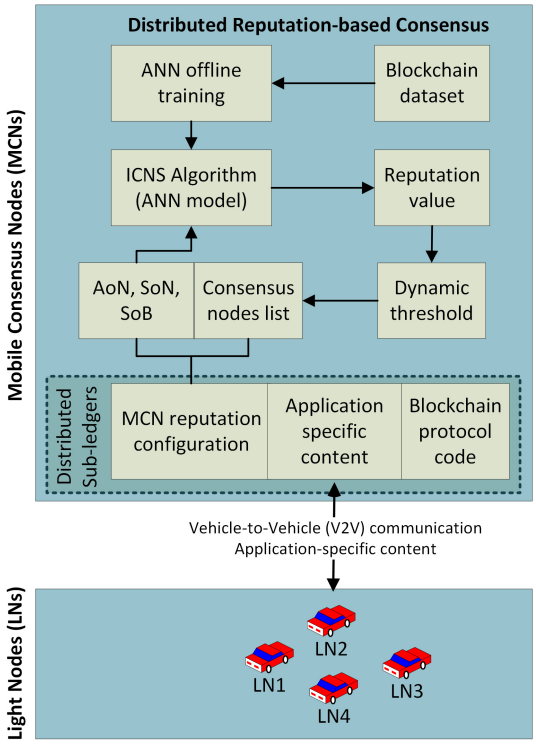

Fig. 2. Mobility-aware Reputation-based Blockchain Framework (MRBF).

transactions including application-specific contents (i.e., road-safety information) to a set of distributed MCNs for distributed consensus and content storage into the blockchain ledger.

In our system design, we assume an incentive mechanism (in a similar way to other consensus mechanisms including RBFT and PoT) to encourage MCNs to actively and trustfully participate in the consensus mechanism while ensuring the integrity of the distributed ledger by linking the incentive mechanism to the reputation index parameters. In other words, a higher reputation index parameter of individual MCN results in a higher reputation value for that particular node, therefore, a higher chance of validating blocks, and ultimately a higher chance of receiving rewards in terms of transaction fees or tokens. As depicted in Figure 2, reputation index parameters, in particular, the Stability of Node (SoN) are also stored in the distributed ledger. Ultimately, the reward mechanism could promote the MCNs with higher reputation value in the form of transaction fees or any other application-specific tokens to keep MCNs motivated to stick to the crucial honesty requirements. Furthermore, we assume that the clusters are static and have been defined based on some pre-defined criteria. Hence, we do not specifically focus on a particular node clustering algorithm. Instead, we focus on the dynamic behaviour of clusters in terms of frequent cross-cluster movements.

IV. Intelligent Consensus Node Selection Algorithm (ICNS)

A. Overview of ICNS

As indicated previously, some application scenarios require clustering the entire blockchain network to address the scalability issue of blockchain. On top of that, the dynamic nature of certain application scenarios mandates frequent movement of blockchain nodes, so cross-cluster movements are happening quite often. In the context of reputation-based consensus mechanisms, we envisage that there is a necessity to involve mobility parameters of consensus nodes in the selection of consensus nodes in each individual cluster. This necessitates designing an intelligent consensus node selection algorithm based on mobility parameters on top of other conventional reputation parameters. We have adopted ANN for assigning reputation values to MCNs based on a trained model.

B. Reputation Model Based on ANN

The ANN consists of three main layers: (1) input layer, (2) hidden layers, and (3) output layer. In our ANN, the input layer takes three reputation index parameters namely Age of Node (AoN), Stability of Node (SoN), and Success Rate of Block Proposals (SoB) which are defined in equations 1-3. As per the definition, T_{in} denotes the time that an MCN is expected to move into a new cluster. Conversely, T_{out} denotes the time that an MCN is expected to move out of the cluster.

$$SoN^{n,c} = T^{n,c}_{out} - T^{n,c}_{in} \tag{1}$$

where n denotes the values for an individual node in a given time while c denotes the values for that particular node in a given cluster at a given time. On the other side, T_{init} refers to the initial time in which the MCN joins the entire blockchain network. This time is used for calculating the AoN.

$$AoN^{n} = T^{n}_{in} - T^{n}_{init} \tag{2}$$

For SoB, we consider the total number of true block proposals (B_{true}) in comparison with the total number of proposed blocks (B_{total}) by a particular MCN since joining the network.

$$SoB^{n} = \frac{B^{n}_{true}}{B^{n}_{total}} \tag{3}$$

In our design, AoN and SoB are calculated in the global domain, so the historical values are kept even when the MCN moves to a new cluster. AoN and SoB jointly indicate MCN's trustworthiness. Adversely, SoN is cluster-specific, so it changes based on MCN's timetable information in a given cluster. SoN indicates the stability of nodes in terms of how long an MCN stays in a given cluster. The hidden layers in ANN find a model based on the input and output values given their reputation index parameters. The output layer consists of reputation values for each MCN. We define a reputation dataset along with the following formula to calculate reputation values considering reputation index parameters:

$$Rep^{n,c} = \frac{AoN^{n}}{AoN_{average}} + \frac{SoN^{n,c}}{SoN_{average}} + \frac{SoB^{n}}{SoB_{average}}, \tag{4}$$

where *Rep* denotes the reputation value. The average values for the entire dataset are used for calculating the reputation index parameters, which indicate the relative values of AoN, SoN, and SoB in proportion to the average of these values in the entire dataset. After creating the dataset based on the above method, the dataset is cleaned, and reputation index parameters are used in ANN's input layer. Accordingly, the reputation values are used in ANN's output layer. After splitting the dataset into train/test, we test the model to ensure its accuracy.

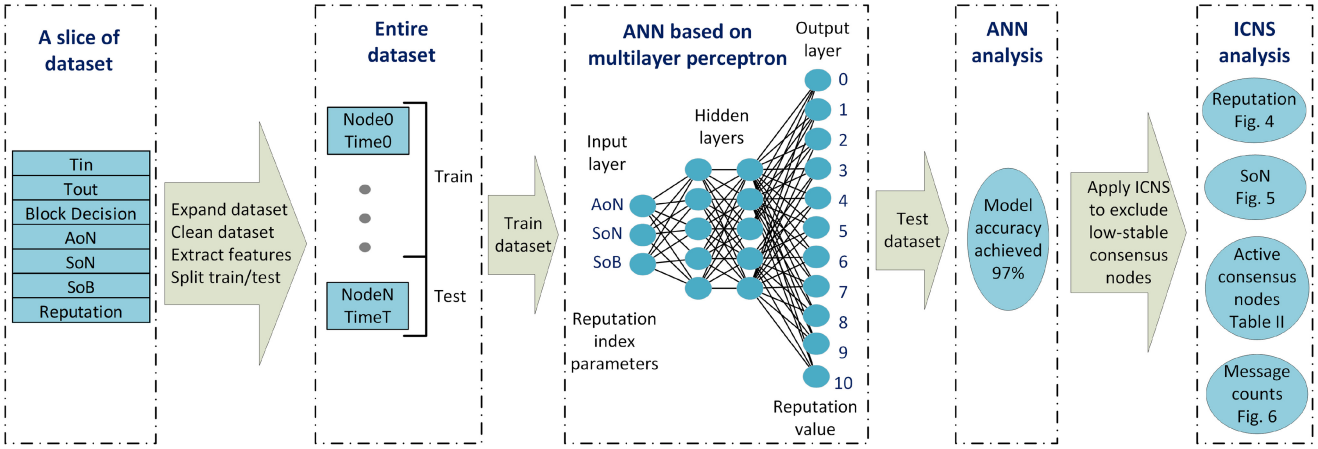

Fig. 3. Methodology for running experiments to analyse the impact of our proposed scheme.

TABLE I
pBFT MIN AND MAX NODE THRESHOLD CRITERIA

Active Consensus Nodes (# nodes)	Byzantine Fault Tolerance (# nodes)	Blockchain Confirmation Latency (seconds)	Blockchain Throughput (transactions per second)	Blockchain Message Counts (messages)
1	0	L1	T1	M1
4	1	L2	T2	M2
7	2	L3	T3	M3
10	3	L4	T4	M4
13	4	L5	T5	M5
16	5	L6	T6	M6
19	6	L7	T7	M7
22	7	L8	T8	M8

C. Intelligent Dynamic Reputation Threshold

ICNS requires a threshold mechanism to either include or exclude consensus nodes in/from the consensus mechanism. Unlike existing works on blockchain consensus (miner) node selection including [11], the threshold in our design is not static. This is because the threshold should intelligently consider local cluster considerations including the local MCNs' reputation index values for all nodes within the cluster. Besides, the dynamic reputation threshold should consider blockchain's live performance and security measures.

In order to maintain acceptable security and performance measures, we consider the minimum and maximum consensus node threshold. As a result of that, our intelligent dynamic reputation threshold changes dynamically based on the live number of nodes involved in the CM. The minimum node threshold ensures the security of the blockchain consensus mechanism while the maximum node threshold ensures satisfactory performance in terms of blockchain confirmation latency, throughput and message counts (communication overhead). Table I depicts the main criteria for defining minimum and maximum node threshold values. The actual values for node thresholds depend on the target security and performance requirements which are application-specific. These values are denoted as L1-L8 (blockchain confirmation latency), T1-T8 (blockchain throughput), and M1-M8 (blockchain message counts). In pBFT, the number of active consensus nodes determines the security of the system in terms of the maximum number of byzantine fault nodes tolerated in the network. Table I indicates up to 22 active consensus nodes, however, the corresponding values can be defined based on the fact that pBFT can tolerate one-third of byzantine/fault nodes.

TABLE II
ICNS VS NO SELECTION ALGORITHM

Evaluation parameter (average values)	**Result**		
	No selection	***ICNS***	***Improvement %***
Reputation	4.631	4.947	6.823
SoN	98.309	115.592	17.579
Active consensus nodes	19.662	16.050	18.369
Message counts	641.838	424.155	33.916

V. SIMULATION RESULTS AND DISCUSSION

We have conducted an experiment to evaluate how many nodes are being excluded/included with/without our algorithm. We aim to compare the impact of a scenario where ICNS is used for selecting consensus nodes versus a scenario without a consensus node selection mechanism. More precisely, we want to examine how much we managed to improve the average reputation and SoN values. Besides, we want to understand the impact of excluding low-reputation nodes on the overall blockchain performance including message overhead.

As demonstrated in Table II, we have successfully utilised ANN to select the optimal number of consensus nodes. This eventually reduces the average number of active consensus nodes and message overhead without a negative impact on the average reputation of the blockchain consensus nodes. More specifically, we have excluded nearly four nodes on average which was decided based on the live reputation values of nodes. The "improvement" column in Table II indicates the absolute values in terms of how much each evaluation parameter has improved during the simulation time. Table III presents the main simulation setting used in our experiment.

Figure 4 illustrates the average reputation values of all MCNs in a given cluster during the simulation time. As shown, these values are increasing over time, this is due to the nature of AoN value which keeps increasing as time increases. Figure 5 illustrates how the average SoN values are changing over time. As shown, there is no trend in the SoN values due to the nature of stability values which are cluster-specific. Figures 4 and 5 clearly prove our point that our scheme outweighs the scenario where no selection algorithm is in place. Figure 6 compares the number of consensus

TABLE III
SIMULATION PARAMETERS SETTINGS

Parameter	Value
Simulation time	1 week - 10080 mins
Number of clusters	2
Total number of MCNs in the network	40
Total number of events	200 per week per node
Frequency of inter-class movement events	100 per week per node
Frequency of block proposals	100 per week per node
Overall success rate of block proposal events	95% per node
Update period of reputation index parameters	Every 1-minute
Reputation integer values range	0 to 10
Minimum node threshold	7
Maximum node threshold	25

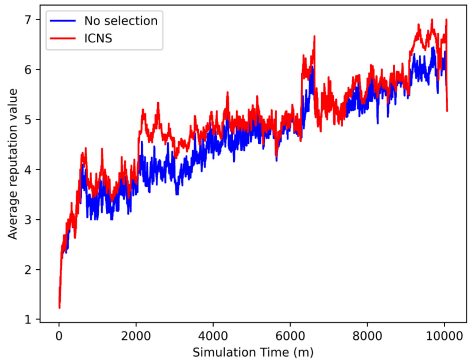

Fig. 4. Average reputation during simulation time.

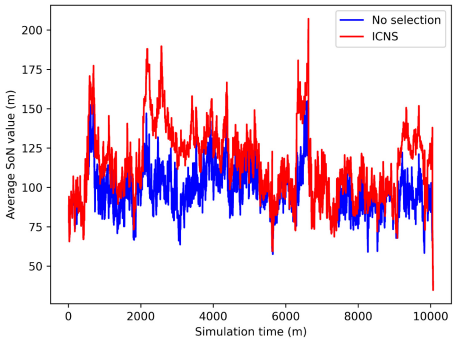

Fig. 5. Average SoN during simulation time.

messages exchanged throughout the simulation in our scheme versus the no-selection algorithm. This figure reveals the impact of excluding low-reputation nodes from the consensus mechanism in pBFT. Indeed, the number of messages in our scheme is less due to the reduced number of active consensus nodes while considering the consensus nodes' reputation index parameters.

VI. CONCLUSION

In this letter, an intelligent algorithm is introduced to accommodate the mobility of consensus nodes into the blockchain consensus mechanism. The simulation results indicate the capability of our proposed scheme for improving the average reputation and stability of blockchain consensus nodes

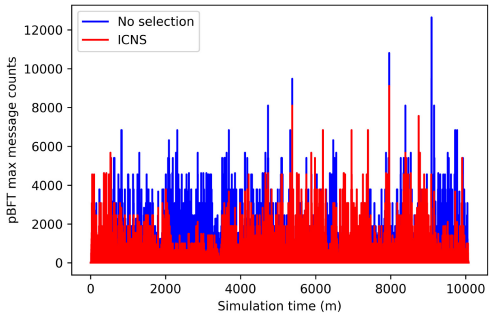

Fig. 6. Message counts during simulation time.

while reducing message counts in pBFT. This letter is distinguished from others as the stability of blockchain consensus nodes is considered in the selection process. Currently, we are expanding our simulation platform to (1) incorporate multiple clusters, (2) calculate the imposed overhead of ICNS, and (3) explore the impact of inaccurate mobility information.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of the University of Surrey 5GIC & 6GIC (http://www.surrey.ac.uk/ics) members for this work.

REFERENCES

[1] N. Giri, R. Jaisinghani, R. Kriplani, T. Ramrakhyani, and V. Bhatia, "Distributed denial of service (DDoS) mitigation in software defined network using blockchain," in *Proc. 3rd Int. Conf. I-SMAC (IoT Social, Mobile, Analytics Cloud) (I-SMAC)*, 2019, pp. 673–678.

[2] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: A survey," *IEEE Access*, vol. 8, pp. 16440–16455, 2020.

[3] A. Hafid, A. S. Hafid, and M. Samih, "Scaling blockchains: A comprehensive survey," *IEEE Access*, vol. 8, pp. 125244–125262, 2020.

[4] J. Lee, S. Ahn, and R. C. Pryss, "Adaptive configuration of mobile roadside units for the cost-effective vehicular communication infrastructure," *Wireless Commun. Mobile Comput.*, vol. 2019, Jul. 2019, Art. no. 6594084.

[5] Y. Cao and N. Wang, "Toward efficient electric-vehicle charging using VANET-based information dissemination," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 2886–2901, Apr. 2017.

[6] M. T. de Oliveira, L. H. A. Reis, D. S. V. Medeiros, R. C. Carrano, S. D. Olabarriaga, and D. M. F. Mattos, "Blockchain reputation-based consensus: A scalable and resilient mechanism for distributed mistrusting applications," *Comput. Netw.*, vol. 179, Oct. 2020, Art. no. 107367.

[7] K. Lei, Q. Zhang, L. Xu, and Z. Qi, "Reputation-based Byzantine fault-tolerance for consortium blockchain," in *Proc. IEEE 24th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, 2018, pp. 604–611.

[8] J. Zou, B. Ye, L. Qu, Y. Wang, M. A. Orgun, and L. Li, "A proof-of-trust consensus protocol for enhancing accountability in crowdsourcing services," *IEEE Trans. Services Comput.*, vol. 12, no. 3, pp. 429–445, May/Jun. 2019.

[9] L. Lao, X. Dai, B. Xiao, and S. Guo, "G-PBFT: A location-based and scalable consensus protocol for IoT-blockchain applications," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, 2020, pp. 664–673.

[10] K. Saadat, N. Wang, X. Wei, B. Da, and R. Tafazolli, "Reconfigurable blockchains for dynamic cluster-based applications," in *Proc. IEEE Int. Conf. Parallel Distrib. Process. Appl., Big Data Cloud Comput. Sustain. Comput. Commun., Social Comput. Netw. (ISPA/BDCloud/SocialCom/SustainCom)*, 2020, pp. 925–931.

[11] S. R. Maskey, S. Badsha, S. Sengupta, and I. Khalil, "Reputation-based miner node selection in blockchain-based vehicular edge computing," *IEEE Consum. Electron. Mag.*, vol. 10, no. 5, pp. 14–22, Sep. 2021.