Empirical Analysis of Sidechaining Models for QoS Aware Blockchain Deployments from a Pragmatic Perspective

Dharmendra Kumar Roy [1] Dr. Anurag Sharma[2]

[1] Research Scholar, CSE, *MATS University, Raipur(Chhattisgarh)*
[2]Associate Professor, CSE, *MATS University, Raipur(Chhattisgarh)*
roy.dharmendra@gmail.com[1], anusiraag@gmail.com[2]

Abstract: Blockchain based security deployments have gained a lot of popularity due to their immutable, transparency, traceability, and distributed computing characteristics. Blockchain models store data in the form of small chunks (called as blocks), which are linked with each other via rule-based unique hashes. A large number of consensus models are defined by researchers that allow addition of these blocks to the blockchain. These consensus models define block-level rules (mining rules) that must be satisfied before addition of blocks. Due to use of distributed computing, blockchains are capable of easily mitigating Finney, Distributed Denial of Service (DDoS), Masquerading, Sybil, and other attacks. But as the length of blockchain increases, the delay needed to add a block to the chain increases exponentially. This is due to the fact that while adding blocks, multiple delay components are needed to be incorporated, which include, hashing delay, encryption delay, hash validation (mining) delay, block read delay, block write delay, etc. Due to this exponential increase in delay, scalability & applicability of blockchain is reduced, which limits its adoption for high-speed large-scale deployments. To overcome this limitation, sidechains or blockchain shards were developed, which work via dividing the main blockchain into QoS-aware (Quality of Service) smaller chains. Each of these chains have the same immutability, traceability, transparency, and distributed computing characteristics from main blockchain, but require smaller delay for block mining & addition, which improves applicability of the underlying blockchain deployment. But a large number of sidechain models are proposed by researchers, and each of them varies in terms of computational complexity, security level, QoS performance, scalability, applicability, and other performance metrics. Due to such a wide variation in parametric performance, it is ambiguous for researchers & blockchain design engineers to identify most suited sidechain for their application-specific deployments. To overcome this limitation, a survey of different sidechaining models, along with their nuances, advantages, limitations, and future research scopes id discussed in this text. Based on this survey, readers will be able to identify different sidechaining solutions for their application-specific use cases. This text also compares the reviewed models in terms of different performance metrics, which will further assist researchers to select most optimum sidechaining models for their deployments. Furthermore, this text also evaluates a novel Model Rank Score (MRS) which combines various performance metrics in order to assist readers in selection of an optimum sidechaining model for context-sensitive use cases.

Keywords: Blockchain, Sidechain, Sharding, Machine, Learning, Delay, Security, MRS, QoS

DOI Number: 10.48047/nq.2022.20.19.NQ99223 **NeuroQuantology2022;20(19): 2614-2630**

eISSN 1303-5150

1. Introduction

Blockchains are defined as secure & immutable data-structures that cannot be easily compromised, and are highly distributed with traceability & trust characteristics [1]. The delay needed to add a block to single blockchains is defined via equation 1,

$$D(Add) = N * [D(Read) + D(Validate) + D(Mine)] + (N-1) * [D(Hash) + D(Encrypt)] + D(Write) \dots (1)$$

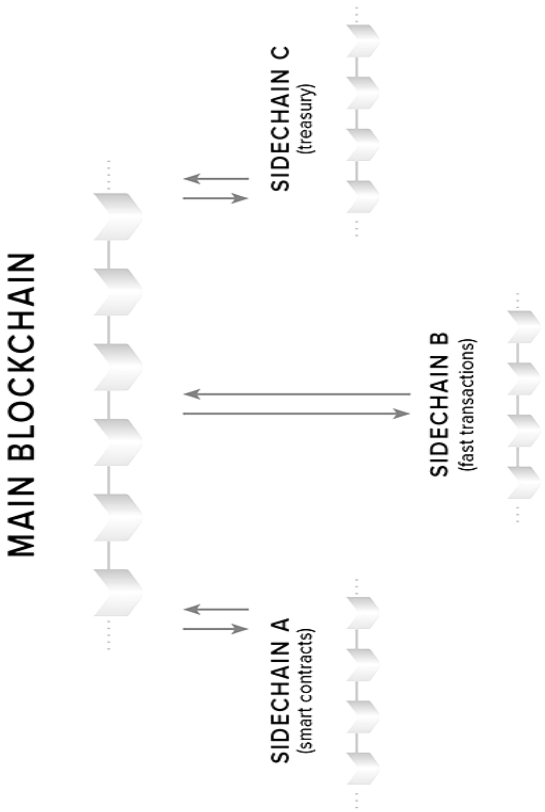

Figure 1. Example of sidechaining for context-sensitive scenarios

Where, $Add, Read, Validate, Mine, Hash, Encrypt\ \&\ Write$ represents various operations like addition of block into the chain, reading the block from the chain, validating the block, mining the block, hashing the block, encrypting the bock, and writing the block back to the blockchain. From this equation it can be observed that, delay needed to add a block exponentially increases w.r.t. blockchain size. Due to which researchers developed smaller sized blockchains, which were termed as sidechains or blockchain shards. A typical sidechain model is depicted in figure 1, wherein Main Blockchain is divided into 3 different sidechains, and each of them are context-sensitive. Sidechain A uses smart contracts, while Sidechain B uses faster consensus models, and Sidechain C uses high security encryption & hashing models, which assists in treasury-based deployments. Due to this division, value of N in equation 1 is reduced, which increases transactional speeds. Similar models [2, 3, 4], along with their nuances, advantages, limitations, and future research scopes are discussed in next section of this text. Based on this discussion readers will be able to identify high-performance models for their application-specific deployments. Followed by this, section 3 compares these models in terms of different performance metrics, that includes computational complexity, security level, QoS performance, scalability, and applicability. Upon referring this comparison, researchers can identify optimally performing models for their use cases. Finally, this text concludes with some interesting observations about the reviewed models, and recommends various methods to further improve their performance under different scenarios.

2. Literature Review

A wide variety of sidechaining models are available, which assist in improving QoS performance of secure blockchain deployments. Each of these models have their own characteristics, which makes it highly complex to identify optimum models for given application use cases. For instance, work in [2] proposes use of State Sharding using space aware representations (SSSAR) which evaluates a trade-off between size of data stored on the blockchain, and number of shards generated during the process. The model showcased reduced memory utilization with higher throughput when evaluated on Ethereum Networks. But the model requires state information to be available before deployment, which limits its scalability performance. To overcome this limitation, work in [3] proposes use of Polynomial Coded Blockchain Sharding (PCBS) that aims at achieving an equilibrium between security, scalability & decentralization across distributed deployments. The model uses node-level computations in order to evaluate verification functions which assist in deployment of coded transactions. The sharding process can be observed from figure 2, wherein each group of 10 transactions are combined to form a blockchain shard, which is encoded & decoded via verification functions.

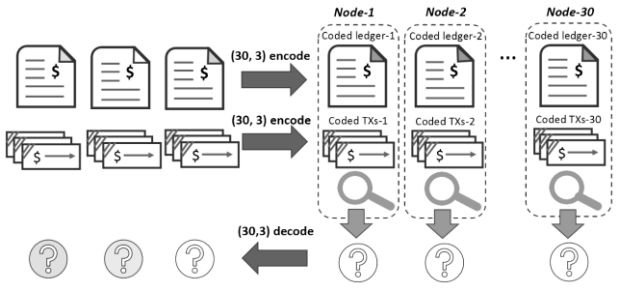

Figure 2. Design of Poly Shard Model [3]

Due to simplicity in sharding process, the model is capable of producing sidechains with low delay, and lower computational complexity, which makes it useful for a wide variety of low & medium scale applications. But as size of the chain increases, number of shards increase exponentially, which increases complexity of shard management & limits its scalability. To overcome this limitation, work in [4] proposes use of Fuzzy Inference Model (FIM) for improving sidechain creation capabilities. The model uses a combination of Bandwidth Consumption (B), Storage Consumption (S), & Calculation Consumption (C) in order to train a fuzzy rule engine, which assists in optimum splitting of blockchains. Due to use of BSC metrics, the model is highly reconfigurable and has lower complexity for creation of sidechains when compared with Serial Miner (SM), & Lock Miner (LM) Models, deployed on the same blockchain datasets. A similar model that uses Smart Contract-based Hierarchical Model for Group Key Agreement (SCHMGKA) [5] that assists in dividing blockchain shards into 2 levels. Initial level uses Group Controllers (GCs), while secondary level uses Sub Group Controllers (SGCs) to manage different sidechains. The model was deployed for Vehicular Adhoc Networks (VANETs), but can be extended to any wireless network deployment with multiple nodes. Due to division of sidechain management tasks, the model is capable of maintaining lower computational complexity, with high-speed mining operations, when compared with Multiple Attribute Authenticated & Contributory Group-based Key Agreement (MACGKA) based models.

An interesting model for shard creation based on service-awareness (SSA) is discussed in [6], which proposes use of securely scalable blockchains based on service requirements.

The model linearly increases number of chains w.r.t. number of services added into the deployments. A typical example of this can be observed from figure 3, wherein services like 'payments', 'domain name assignment', 'registration', and 'intellectual property' are added incrementally into the model, and based on these additions, number of sidechains are increased.

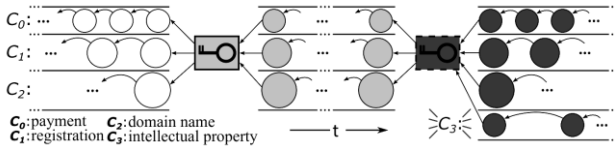

Figure 3. Use of incremental service-aware sharding [6]

The model showcases good performance under limited number of users, but the blockchain increases if a single service is used by majority users. This limitation can be removed via the work in [7] which proposes use of Federated Learning (FL) to create shards. The model uses Direct Acyclic Graph (DAG) with Model Aggregation, & Transaction Exchange to train the FL Models. It stores all data on Inter Planetary File System (IPFS), and mines blocks via Raft Consensus, which assists in improving shard-level reconfigurability under different scenarios. The model was observed to have better performance when compared with Asynchronous FL (AFL), and Federate Average (FA) learning models, which makes it highly useful for a wide variety of application deployments. A simpler version of this model is discussed in [8] which proposes use of splitting & aggregating blockchain (SAB) signatures for to split of merge blockchains. This process allows archiving sidechains which are rarely used, and splitting blockchains that are constantly updated during transactional communications. The model showcases lower complexity with higher speed, but requires constant chain reconfiguration, which limits it deployment capabilities. To overcome this issue, work in [9] proposes a simpler method to Reconfigure Blockchains via Clustering Process (RBCP). The model uses 3 phases, namely, presplitting, splitting & post-splitting to select cluster heads, that decide whether to split or merge blocks via multiple evaluation criterions. These criterion include, speed-awareness, energy awareness, security awareness, complexity awareness, etc. and can be tuned as per application requirements. The model is highly reconfigurable, which makes it useful for large-scale deployments, but requires complex management processes, which increases delay & reduces throughput w.r.t. number of split-based sidechains. Effect of this limitation can be reduced via use of Parallel Middle Blocks (PMBs) [10] which uses stochastic process for continuous sharding, thereby assisting in improving validation & shard creation performance under different network scenarios. This model's performance can be extended via work in [11], wherein researchers have proposed use of Two-way Peg Model (TPM), that dynamically estimates trust-levels for different miner nodes, and selects them for efficient mining process. Due to this dynamic miner selection process, the proposed model showcases lower storage cost, better security performance, and lower processing delay when compared with Proof of Work (PoW) based blockchain, and Threshold based Sidechain (TSC) Models, thereby making it useful for a wide variety of real-time deployments. Security of this model must be evaluated under different attacks, and can be extended via use of Membership Management with Reduction of Failure Probability (MMRFP) [12] for mitigating n/2 attacks. The model has higher complexity, but can be used for large-scale deployment due to constant computational requirements. The model doesn't incorporate fault tolerance, due

to which its applicability is limited to networks with minimum fault scenarios. To overcome this limitation, work in [13] proposes use of Trust-Based Shard Distribution (TBSD) which deploys Genetic Algorithm (GA) for stochastic trust estimation, and assists in reducing collusions for malicious miner nodes. The model uses a Trust Agent (TA) that performs estimation of Miner Trust Management, and Shard Distribution mechanisms via parallel processing operations. It combines leader selection, block commit operation, Reporting Subjective Consensus Opinion (RSCO), Local Consensus Result (LCR) formation with final node trust evaluation for estimation of optimum shards under real-time attack scenarios. Application of such models for E-Voting deployments can be reviewed from [14], wherein Proof of Stake (PoS) and Proof of Credibility (PoC) are combined to form shards based on share of voters as observed from figure 4, which assists in improving QoS performance under large-scale applications. It can be observed that due to stochastic clustering & stake based sharding, the model is capable of removing consensus done via malicious nodes, which assists in improving security against multiple attacks. But the model is highly stochastic in nature, which limits its reliability under higher scaled networks. To improve this reliability, work in [15] proposes use of Queueing Modelling with Game Theoretic Model (QM GTM) for continuous evaluation of blockchain shards via Nash Equilibrium analysis. The model uses a Polynomial Time (PT) method to obtain balance between shards creation & management performance, thereby assisting in improving scalability under dynamic communication environments. The model showcases lower delay than Centralized Optimization (CO), and Random with Uniform Distribution (RUD) due to inclusion of average transaction confirmation time (ATCT) between different shard creation processes.

Similar sharding models that use directed acyclic graph (DAG) [16], Blockchain-Based Federated Learning (BFL) [17], many objective optimization algorithm based on the dynamic reward and penalty mechanism (MaOEA-DRP) [18], and Reputation based High Incentive Blockchain (RHIB) [19] are proposed by researchers.

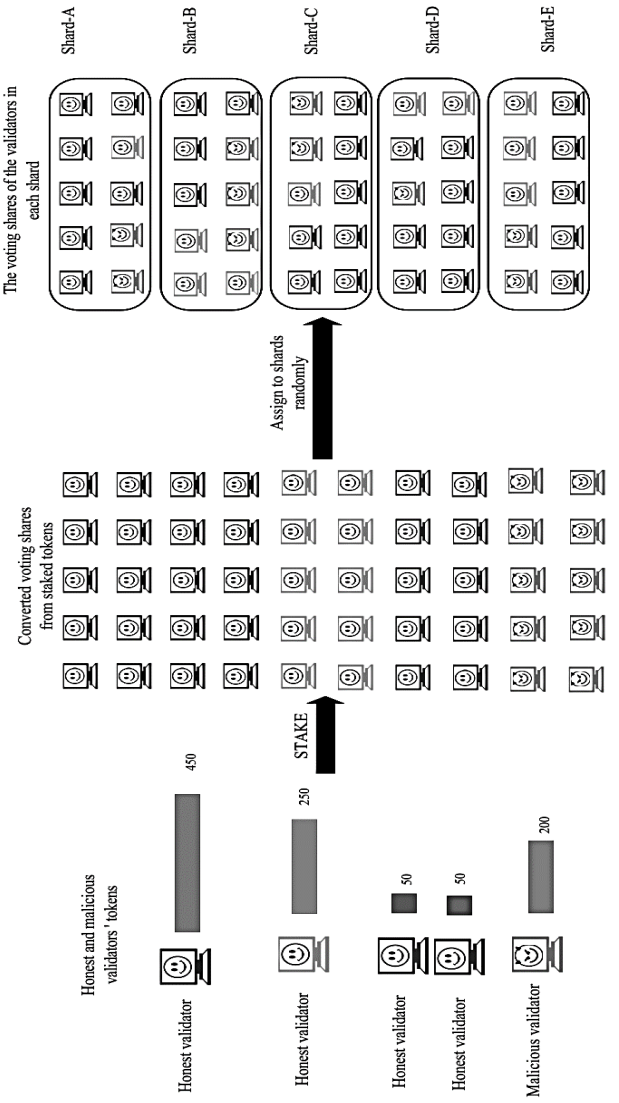

Figure 4. Use of sharding with Honest & Malicious Miner Nodes [14]

These models augment energy levels, mining delay, and throughput parameters during shard creation in order to reduce computational complexity, while maintaining high security performance. These models have higher complexity due to use of Machine Learning (ML) based methods, which requires high-power processing miner entities, which limits their scalability. To overcome this limitation work in [20] proposes use of Verifiable Stochastic Function (VSF) for improving self-balancing & operability equilibrium under different conditions. The model's performance is observed to be consistent across different scaled networks, and thus can be deployed for large-scale sidechain deployments. Extension to this model is discussed in [21], wherein researchers have proposed use of Two-Phase Cooperative Bargaining Game Model (TPCBGM) to improve shard creation efficiency. The model uses Weak Pareto Optimality (WPO), symmetrical performance, covariance with respect to translations, restricted monotonicity, and covariance with respect to positive affine transformations in order to create & manage sidechains. Due to which the model is able to scale even under multiple attack scenarios. To evaluate performance of such models, work in [22] proposes Chebyshev, Chvátal, and Hoeffding bounds for estimation of delay, energy consumption, and throughput bounds for different sidechain deployments. This model must be used for evaluating performance of highly efficient sidechaining models including, Software-Defined Networking based Low-delay, highly Secure & Reliable Model for Making Decisions with good Emergency Handling capabilities (SDN LSROM EH) [23], Reputation aware Aggregation Model via Secure & Self-Organized Scalable Sharded (RAMSSOSS) [24] blockchains deployed on edge computing devices, and a Deep Q Learning Network for Sharded Blockchains (DQLNSB) [25] that uses Deep Reinforcement Learning (DRL) for optimization of shards. These models stochastically augment multiple network metrics in order to estimate optimal combination of parameters for incrementally higher performance under different network scenarios. Efficiency of these models must be validated on different network types, and can be further evaluated via use of Joint Hypergeometric Distribution based attack simulation model [26] for network deployments. The model is capable of simulating multiple attacks, and then estimating their security performance under different network configurations. Such models must be used to estimate security performance of sharded chains. Similar sharded chain deployment methods like Validator Rotation via Hierarchical Game Theory Model (VR HGTM) [27], Permissionless Blockchains with Game Theoretic Model (PBGT) [28], Contract-Theoretic Pricing (CTP) for handling multiple transactions on sharded blockchains [29], and Extreme Learning Machine (ELM) [30] for classification of blocks into active & inactive blocks for sidechain creation & management are discussed, and validated under different simulation conditions.

Applications of such sidechains are discussed in [31, 32], wherein researchers have deployed ZyConChain for division of main blockchain into sidechain & state chains, and Hierarchical Consensus Mechanism for Service-Zone Sharding (HCM SZS) for multiple security applications. These models use different context-aware sidechain creation mechanisms, which assist in improving their application-specific performance. To further improve scalability of such models, work in [33, 34] propose use of Distributed Stochastic Generation (DSG) with Shard Reconfiguration, and use of Reputation based Sharding (RBS) for

better speed & lower energy consumption under different blockchain types. Such models use stochastic division of blocks into shards, which limits their reliability & consistency performance. To overcome this limitation work in [35] proposes use of Verifiable Secret Sharing (VSS) with controlled stochastic model for reducing stochasticity during shard selection, thereby improving efficiency during sharding process. Security & QoS extensions to these models are also discussed by researchers in [36, 37], wherein Garlic Onion Routing (GOR), and Off-Chain Computation Management (OCCM) methods are used to incorporate QoS-aware privacy models into existing sidechains. These models must be evaluated under different transaction types, and can be extended via use of load balancing (LB) [38], and Gas Consumption-Aware Relocation (GCAR) [39] for improving applicability & usability performance under different application-specific load conditions. Applications of such models that use Time Variant Multiple Objective based Particle Swarm Optimization (TVMOPSO) [40], combination of Para-Sharding with DAGs and Proof of Participation (PS DAG PoP) Consensus Model [41], Land Registry Search Model (LRSM) [42] based sidechaining, and Distributed Machine Learning (DML) [43] for improving security & scalability of blockchain based models are proposed, which assist in deployment of context-aware models, that can be scaled as per application requirements. Similarly, work in [44, 45] also propose use of DML for Permission Integrated Ring Alliance-based Training Estimation (PIRATE) model, and ML based Sharding Mechanisms (MLSM), which assist in improving blockchain deployment capabilities are also discussed and tested under different test cases. These models allow for highly augmented sharded application deployments, and have wide variation in real-time performance. Estimation of this performance in terms of computational complexity, mining delay, cost of deployment, scalability, & QoS performance is evaluated for different models, and can be observed from the next section of this text. Based on this evaluation, researchers will be able to identify best models for their application-specific deployments.

3. Result analysis & comparison

From the literature survey, it can be observed that existing shard creation & management models utilize Machine Learning (ML) to optimize performance under different application scenarios. Their performance also varies w.r.t. deployed scenario, which makes it highly ambiguous for researchers to identify best performing model for their application deployments. To reduce such ambiguities, this section compares the reviewed models in terms of computational complexity (CC), mining delay (D), cost of deployment (CD), scalability (S), & QoS (Q) performance metrics. Due to non-uniformity in simulation & deployment environments, this section evaluates these metrics in terms of fuzzy ranges of Low (L=1), Medium (M=2), High (H=3), and Very High (VH=4), which will assist readers to estimate sidechain creation performance on a uniform scale. Based on this evaluation criterion, table 1 summarizes performance of different models w.r.t. various evaluation metrics.

Model	CC	CD	D	Q	S
SSSAR [2]	H	M	M	H	L
PCBS [3]	VH	H	M	M	H

FIM [4]	M	M	L	M	H
SCHMGKA [5]	M	H	M	H	L
SSA [6]	M	L	M	H	L
FL DAG [7]	VH	VH	M	H	M
SAB [8]	L	M	M	H	L
RBCP [9]	M	L	M	M	H
PMBs [10]	H	H	H	M	L
TPM [11]	M	M	L	M	M
MMRFP [12]	H	H	M	H	L
TBSD [13]	M	H	L	H	M
PoSC [14]	H	H	M	L	L
QM GTM [15]	H	VH	M	M	H
DAG [16]	M	H	H	M	M
BFL [17]	VH	VH	L	M	M
MaOEA-DRP [18]	H	VH	M	H	H
RHIB [19]	H	H	M	H	M
VSF [20]	L	L	M	L	M
TPCBGM [21]	H	H	M	M	L
SDN LSROM EH [23]	H	M	M	H	H

RAMSSOSS [24]	H	H	L	H	M
DQLNSB [25]	VH	H	L	M	H
VR HGTM [27]	H	M	M	H	M
PBGT [28]	L	M	L	H	H
CTP [29]	M	H	M	H	L
ELM [30]	VH	H	M	M	L
ZyConChain [31]	M	H	M	M	L
HCM SZS [32]	H	VH	M	M	H
DSG [33]	L	L	M	L	M
RBS [34]	H	H	L	M	H
VSS [35]	H	H	H	M	H
GOR [36]	M	H	L	M	L
OCCM [37]	M	VH	L	M	H
LB [38]	H	H	M	L	L
GCAR [39]	H	H	H	M	L
TVMOPSO [40]	H	H	M	L	M
PS DAG PoP [41]	H	M	M	H	M
LRSM [42]	H	VH	L	M	L
DML [43]	VH	H	L	H	M

PIRATE [44]	VH	H	M	H	M
MLSM [45]	H	H	H	M	L

Table 1. Evaluation of different sidechaining models in terms of various performance metrics

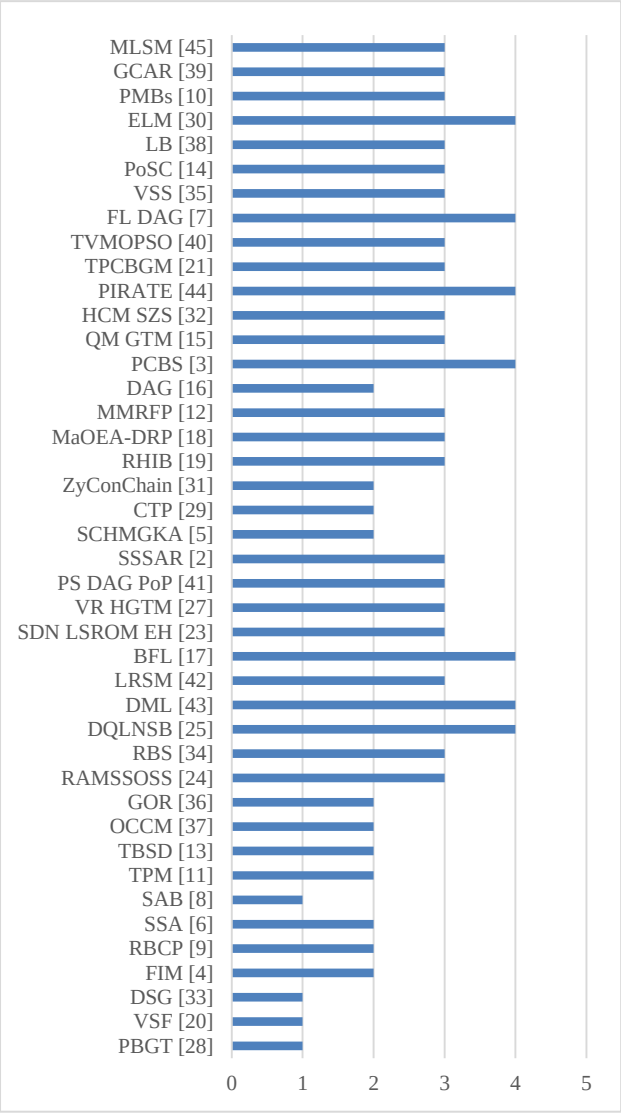

Figure 5. Computational Complexity of different sidechaining models

Based on this evaluation and figure 5, it can be observed that SAB [8], VSF [20], PBGT [28], and DSG [33] showcase lowest computational complexity, and thus can be used for deployment of low-complexity sidechains in Internet of Things (IoT) and other low power applications.

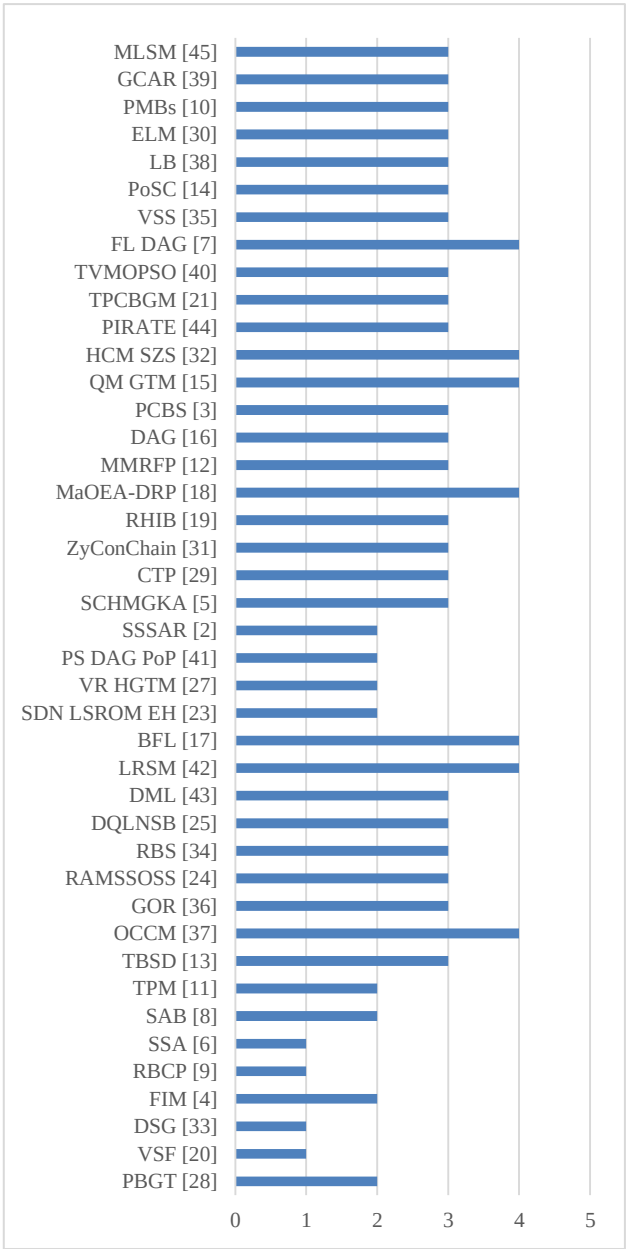

Figure 5. Deployment Cost of different sidechaining models

Similarly, based on table 1 and figure 6, it can be observed that SSA [6], RBCP [9], VSF [20],

DSG [33], SSSAR [2], FIM [4], SAB [8], TPM [11], and SDN LSROM EH [23] have lowest deployment cost, and thus can be used for low-cost applications including Wireless Sensor Networks (WSNs), Aerial Networks (ANs), etc.

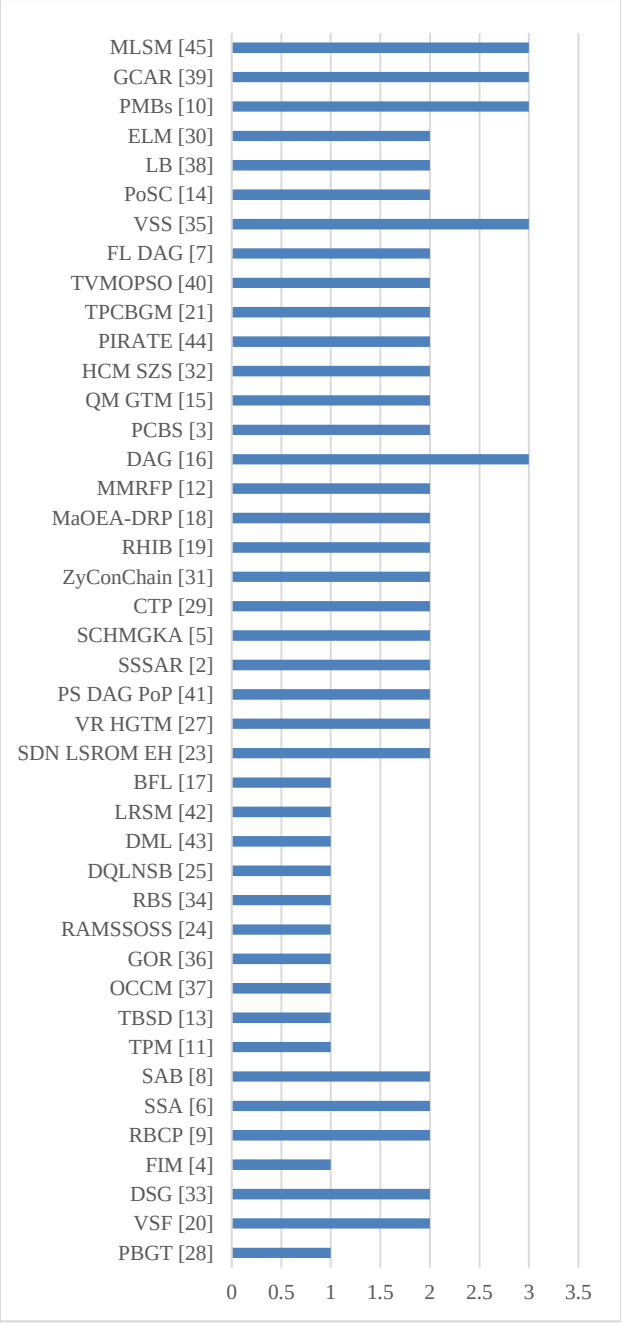

Figure 7. Delay for mining blocks of different sidechaining models

Similarly, based on table 1 and figure 7, it can be observed that FIM [4], TPM [11], TBSD [13], BFL [17], RAMSSOSS [24], DQLNSB [25], PBGT [28], RBS [34], GOR [36], OCCM [37]. LRSM [42], and DML [43] have lowest mining delay, and thus can be used for high-speed applications including Military Deployments, WSNs, and other context-sensitive deployments.

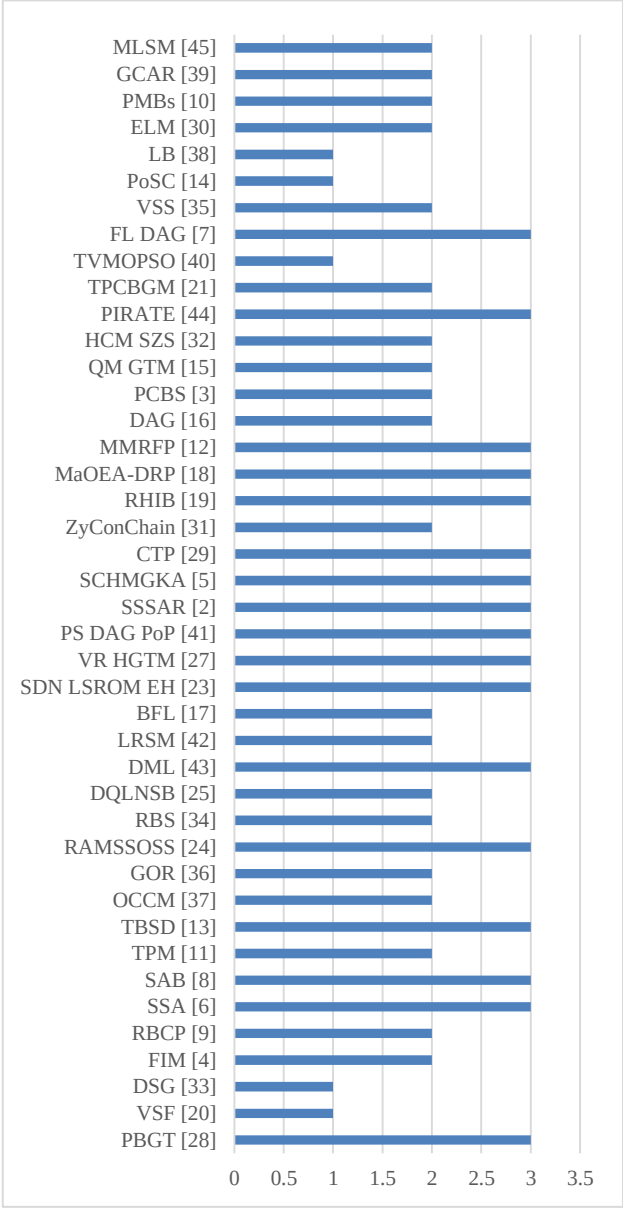

Figure 8. QoS performance of different sidechaining models

Upon referring table 1 and figure 8, it was also observed that SSSAR [2], SCHMGKA [5], SSA [6], FL DAG [7], SAB [8], MMRFP [12], TBSD [13], MaOEA-DRP [18], RHIB [19], SDN LSROM EH [23], RAMSSOSS [24], VR HGTM [27], PBGT [28], CTP [29], PS DAG PoP [41], DML [43], and PIRATE [44] had higher QoS performance, and thus can be used for applications that require lower energy consumption with better throughput performance levels.

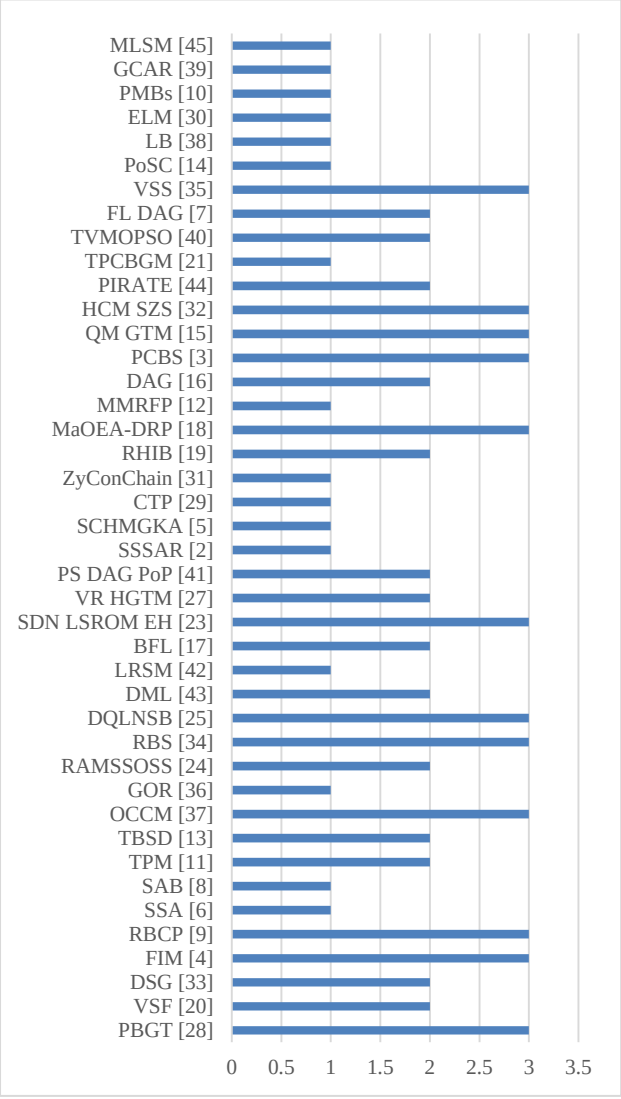

Figure 9. Scalability performance of different sidechaining models

Similarly, from table 1 and figure 9 it was observed that PCBS [3], FIM [4], RBCP [9], QM GTM [15], MaOEA-DRP [18], SDN LSROM EH [23], DQLNSB [25], PBGT [28], HCM SZS [32], RBS [34], VSS [35], and OCCM [37] had better scalability, and thus can be used for multiple real-time applications.

To further simplify this process of model selection, a novel Model Rank Score (MRS) is evaluated via equation 2, which combines multiple evaluation metrics to identify sidechaining models with better overall performance.

$$MRS = \frac{4}{CC} + \frac{4}{CD} + \frac{4}{D} + \frac{Q}{4} + \frac{S}{4} \dots (2)$$

This score was evaluated for each model, and can be observed from table 2 as follows,

Model	**ARS**
SSSAR [2]	6.33
PCBS [3]	5.58
FIM [4]	9.25
SCHMGKA [5]	6.33
SSA [6]	9.00
FL DAG [7]	5.25
SAB [8]	9.00
RBCP [9]	9.25
PMBs [10]	4.75
TPM [11]	9.00
MMRFP [12]	5.67

TBSD [13]	8.58
PoSC [14]	5.17
QM GTM [15]	5.58
DAG [16]	5.67
BFL [17]	7.00
MaOEA-DRP [18]	5.83
RHIB [19]	5.92
VSF [20]	10.75
TPCBGM [21]	5.42
SDN LSROM EH [23]	6.83
RAMSSOSS [24]	7.92
DQLNSB [25]	7.58
VR HGTM [27]	6.58
PBGT [28]	11.50
CTP [29]	6.33
ELM [30]	5.08
ZyConChain [31]	6.08
HCM SZS [32]	5.58
DSG [33]	10.75
RBS [34]	7.92
VSS [35]	5.25
GOR [36]	8.08
OCCM [37]	8.25
LB [38]	5.17
GCAR [39]	4.75
TVMOPSO [40]	5.42
PS DAG PoP [41]	6.58
LRSM [42]	7.08
DML [43]	7.58
PIRATE [44]	5.58
MLSM [45]	4.75

Table 2. MRS for different sidechaining models

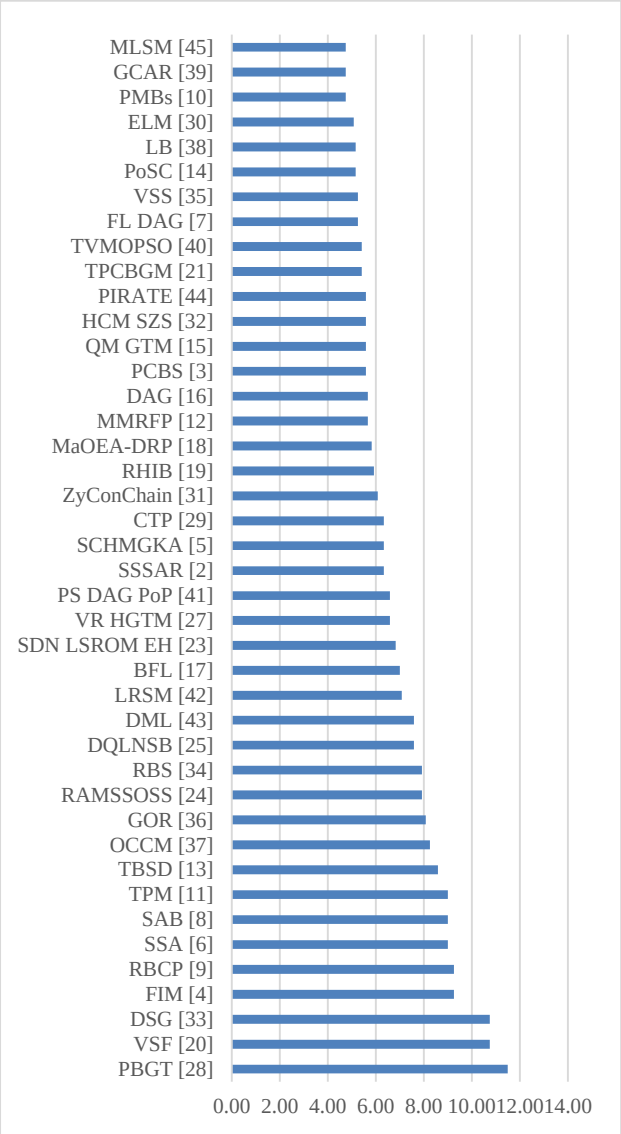

Figure 10. MRS performance of different sidechaining models

Based on this evaluation, and figure 10, it can be observed that PBGT [28], VSF [20], DSG [33], FIM [4], RBCP [9], SSA [6], SAB [8], TPM [11], TBSD [13], OCCM [37], and GOR [36] showcase better overall performance. These models must be used for low delay, low cost, low complexity, high QoS and high scalability sidechain application deployments.

4. Conclusion and future scope

This review estimates performance of different sidechaining models and compares them w.r.t. various statistical metrics. Based on initial estimation it was observed that ML based Models outperform other models in terms of scalability & QoS performance, but addition of privacy, application-based reconfigurability, and enhanced security are also needed for real-time deployments. This text compared the models in terms of computational complexity, mining delay, cost of deployment, scalability, & QoS performance, wherein, it was observed that SAB, VSF, PBGT, and DSG showcased lowest computational complexity, SSA, RBCP, VSF, DSG, SSSAR, FIM, SAB, TPM, and SDN LSROM EH have lowest deployment cost, FIM, TPM, TBSD, BFL, RAMSSOSS, DQLNSB, PBGT, RBS, GOR, OCCM, LRSM, and DML have lowest mining delay, SSSAR, SCHMGKA, SSA, FL DAG, SAB, MMRFP, TBSD, MaOEA-DRP, RHIB, SDN LSROM EH, RAMSSOSS, VR HGTM, PBGT, CTP, PS DAG PoP, DML, and PIRATE had higher QoS performance, while PCBS, FIM, RBCP, QM GTM, MaOEA-DRP, SDN LSROM EH, DQLNSB, PBGT, HCM SZS, RBS, VSS, and OCCM had better scalability, and thus can be used for multiple real-time applications. These metrics were combined to form a Novel Model Rank Score, which indicated that PBGT, VSF, DSG, FIM, RBCP, SSA, SAB, TPM, TBSD, OCCM, and GOR showcase better overall performance. These models must be used for low delay, low cost, low complexity, high QoS and high scalability sidechain application deployments. In future, researchers must validate performance of these models on multiple datasets, and fuse these models to develop hybrid ML based sidechaining techniques, which can be used for a wide variety of network deployments. Moreover, researchers can introduce low-power Convolutional Neural Network (CNN)

models in order to augment sidechaining performance via convolutional feature sets.

5. References

[1] Angtai Li, Guohua Tian, Meixia Miao & Jianpeng Gong (2022) Blockchain-based cross-user data shared auditing, Connection Science, 34:1, 83-103, DOI: 10.1080/09540091.2021.1956879

[2] A. Mizrahi and O. Rottenstreich, "State Sharding with Space-aware Representations," 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), 2020, pp. 1-9, doi: 10.1109/ICBC48266.2020.9169402.

[3] Songze Li, Mingchao Yu, Chien-Sheng Yang, Amir Salman Avestimehr, Sreeram Kannan, and Pramod Viswanath. 2021. PolyShard: Coded Sharding Achieves Linearly Scaling Efficiency and Security Simultaneously. Trans. Info. For. Sec. 16 (2021), 249–261. DOI:https://doi.org/10.1109/TIFS.2020.3009610

[4] Yan Wang, Jixin Li, Wansheng Liu, Aiping Tan, "Efficient Concurrent Execution of Smart Contracts in Blockchain Sharding", *Security and Communication Networks*, vol. 2021, Article ID 6688168, 15 pages, 2021. https://doi.org/10.1155/2021/6688168

[5] Naresh, VS, Allavarpu, VVLD, Reddi, S, Murty, PSR, Raju, NVSL, Mohan, RNVJ. A provably secure sharding based blockchain smart contract centric hierarchical group key agreement for large wireless ad-hoc networks. *Concurrency Computat Pract Exper*. 2022; 34(3):e6553. https://doi.org/10.1002/cpe.6553

[6] Gencer, Adem Efe & Van Renesse, Robbert & Sirer, Emin. (2017). Short Paper: Service-Oriented Sharding for Blockchains. 393-401. 10.1007/978-3-319-70972-7_22.

[7] Yuan, Shuo & Cao, Bin & Sun, Yao & Peng, Mugen. (2021). Secure and Efficient Federated Learning Through Layering and Sharding Blockchain.

[8] S. S. D. Selvi, A. Paul, C. P. Rangan, S. Dirisala and S. Basu, "Splitting and Aggregating Signatures in Cryptocurrency Protocols," 2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON), 2019, pp. 100-108, doi: 10.1109/DAPPCON.2019.00021.

[9] K. Saadat, N. Wang, X. Wei, B. Da and R. Tafazolli, "Reconfigurable Blockchains for Dynamic Cluster-based Applications," 2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), 2020, pp. 925-931, doi: 10.1109/ISPA-BDCloud-SocialCom-SustainCom51426.2020.00142.

[10] Gao, Yuefei & Kawai, Shin & Nobuhara, Hajime. (2019). Scalable Blockchain Protocol Based on Proof of Stake and Sharding. Journal of Advanced Computational Intelligence and Intelligent Informatics. 23. 856-863. 10.20965/jaciii.2019.p0856.

[11] M. Li, H. Tang, A. R. Hussein and X. Wang, "A Sidechain-Based Decentralized Authentication Scheme via Optimized Two-Way Peg Protocol for Smart Community," in IEEE Open Journal of the Communications Society, vol. 1, pp. 282-292, 2020, doi: 10.1109/OJCOMS.2020.2972742.

[12] Xu, Yibin & Huang, Yangyu. (2020). An n/2 byzantine node tolerate blockchain sharding approach. 349-352. 10.1145/3341105.3374069.

[13] J. Yun, Y. Goh and J. -M. Chung, "Trust-Based Shard Distribution Scheme for Fault-Tolerant Shard Blockchain Networks," in IEEE Access, vol. 7, pp. 135164-135175, 2019, doi: 10.1109/ACCESS.2019.2942003.

[14] Abuidris, Y., Kumar, R., Yang, T. and Onginjo, J. (2021), Secure large-scale E-voting system based on blockchain contract using a hybrid consensus model combined with sharding. ETRI Journal, 43: 357-370. https://doi.org/10.4218/etrij.2019-0362

[15] Canhui Chen, Qian Ma, Xu Chen, and Jianwei Huang. 2021. User Distributions in Shard-based Blockchain Network: Queueing Modeling, Game Analysis, and Protocol Design. In Proceedings of the Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc '21). Association for Computing Machinery, New York, NY, USA, 221–230. DOI:https://doi.org/10.1145/3466772.3467051

[16] Chaoxia Qin, Bing Guo, Yan Shen, Tao Li, Yun Zhang, Zhen Zhang, "A Secure and Effective Construction Scheme for Blockchain Networks", *Security and Communication Networks*, vol. 2020, Article ID 8881881, 20 pages, 2020. https://doi.org/10.1155/2020/8881881

[17] ScaleSFL: A Sharding Solution for Blockchain-Based Federated Learning, https://arxiv.org/abs/2204.01202

[18] X. Cai et al., "A Sharding Scheme-Based Many-Objective Optimization Algorithm for Enhancing Security in Blockchain-Enabled Industrial Internet of Things," in IEEE Transactions on Industrial Informatics, vol. 17, no. 11, pp. 7650-7658, Nov. 2021, doi: 10.1109/TII.2021.3051607.

[19] Han, R., Yu, J., & Zhang, R. (2020). Analysing and Improving Shard Allocation Protocols for Sharded Blockchains. *IACR Cryptol. ePrint Arch., 2020*, 943.

[20] S. Kim, "Two-Phase Cooperative Bargaining Game Approach for Shard-Based Blockchain Consensus Scheme," in IEEE Access, vol. 7, pp. 127772-127780, 2019, doi: 10.1109/ACCESS.2019.2939778.

[21] S. Kim, "Two-Phase Cooperative Bargaining Game Approach for Shard-Based Blockchain Consensus Scheme," in IEEE Access, vol. 7, pp. 127772-127780, 2019, doi: 10.1109/ACCESS.2019.2939778.

[22] C. Huang et al., "RepChain: A Reputation-Based Secure, Fast, and High Incentive Blockchain System via Sharding," in IEEE Internet of Things Journal, vol. 8, no. 6, pp. 4291-4304, 15 March15, 2021, doi: 10.1109/JIOT.2020.3028449.

[23] J. Ren, J. Li, H. Liu and T. Qin, "Task offloading strategy with emergency handling and blockchain security in SDN-empowered and fog-assisted healthcare IoT," in Tsinghua Science and Technology, vol. 27, no. 4, pp. 760-776, Aug. 2022, doi: 10.26599/TST.2021.9010046.

[24] A. Asheralieva and D. Niyato, "Reputation-Based Coalition Formation for Secure Self-Organized and Scalable Sharding in IoT Blockchains With Mobile-Edge Computing," in IEEE Internet of Things Journal, vol. 7, no. 12, pp. 11830-11850, Dec. 2020, doi: 10.1109/JIOT.2020.3002969.

[25] J. Yun, Y. Goh and J. -M. Chung, "DQN-Based Optimization Framework for Secure Sharded Blockchain Systems," in IEEE Internet of Things Journal, vol. 8, no. 2, pp. 708-722, 15 Jan.15, 2021, doi: 10.1109/JIOT.2020.3006896.

[26] A. Hafid, A. S. Hafid and M. Samih, "A Novel Methodology-Based Joint

Hypergeometric Distribution to Analyze the Security of Sharded Blockchains," in IEEE Access, vol. 8, pp. 179389-179399, 2020, doi: 10.1109/ACCESS.2020.3027952.

[27] J. Li, D. Niyato, C. S. Hong, K. -J. Park, L. Wang and Z. Han, "Cyber Insurance Design for Validator Rotation in Sharded Blockchain Networks: A Hierarchical Game-Based Approach," in IEEE Transactions on Network and Service Management, vol. 18, no. 3, pp. 3092-3106, Sept. 2021, doi: 10.1109/TNSM.2021.3078142.

[28] M. H. Manshaei, M. Jadliwala, A. Maiti and M. Fooladgar, "A Game-Theoretic Analysis of Shard-Based Permissionless Blockchains," in IEEE Access, vol. 6, pp. 78100-78112, 2018, doi: 10.1109/ACCESS.2018.2884764.

[29] J. Li, T. Liu, D. Niyato, P. Wang, J. Li and Z. Han, "Contract-Theoretic Pricing for Security Deposits in Sharded Blockchain With Internet of Things (IoT)," in IEEE Internet of Things Journal, vol. 8, no. 12, pp. 10052-10070, 15 June15, 2021, doi: 10.1109/JIOT.2021.3049227.

[30] D. Jia, J. Xin, Z. Wang and G. Wang, "Optimized Data Storage Method for Sharding-Based Blockchain," in IEEE Access, vol. 9, pp. 67890-67900, 2021, doi: 10.1109/ACCESS.2021.3077650.

[31] N. Sohrabi and Z. Tari, "ZyConChain: A Scalable Blockchain for General Applications," in IEEE Access, vol. 8, pp. 158893-158910, 2020, doi: 10.1109/ACCESS.2020.3020319.

[32] J. -Y. Kwak, J. Yim, N. -S. Ko and S. -M. Kim, "The Design of Hierarchical Consensus Mechanism Based on Service-Zone Sharding," in IEEE Transactions on Engineering Management, vol. 67, no. 4, pp. 1387-1403, Nov. 2020, doi: 10.1109/TEM.2020.2993413.

[33] Hung Dang, Tien Tuan Anh Dinh, Dumitrel Loghin, Ee-Chien Chang, Qian Lin, and Beng Chin Ooi. 2019. Towards Scaling Blockchain Systems via Sharding. In Proceedings of the 2019 International Conference on Management of Data (SIGMOD '19). Association for Computing Machinery, New York, NY, USA, 123–140. DOI:https://doi.org/10.1145/3299869.3319889

[34] Liu, Yizhong & Liu, Jianwei & Vaz Salles, Marcos Antonio & Zhang, Zongyang & Li, Tong & Hu, Bin & Henglein, Fritz & Lu, Rongxing. (2021). Building Blocks of Sharding Blockchain Systems: Concepts, Approaches, and Open Problems.

[35] Gang Wang, Zhijie Jerry Shi, Mark Nixon, and Song Han. 2019. SoK: Sharding on Blockchain. In Proceedings of the 1st ACM Conference on Advances in Financial Technologies (AFT '19). Association for Computing Machinery, New York, NY, USA, 41–61. DOI:https://doi.org/10.1145/3318041.3355457

[36] R. M. Parizi, S. Homayoun, A. Yazdinejad, A. Dehghantanha and K. -K. R. Choo, "Integrating Privacy Enhancing Techniques into Blockchains Using Sidechains," 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), 2019, pp. 1-4, doi: 10.1109/CCECE.2019.8861821.

[37] M. Westerkamp and J. Eberhardt, "zkRelay: Facilitating Sidechains using zkSNARK-based Chain-Relays," 2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), 2020, pp. 378-386, doi: 10.1109/EuroSPW51379.2020.00058.

[38] Okanami, Naoya & Nakamura, Ryuya & Nishide, Takashi. (2020). Load Balancing for

Sharded Blockchains. 10.1007/978-3-030-54455-3_36.

[39] Woo, S., Song, J., Kim, S. *et al.* GARET: improving throughput using gas consumption-aware relocation in Ethereum sharding environments. *Cluster Comput* **23,** 2235–2247 (2020). https://doi.org/10.1007/s10586-020-03087-1

[40] Nartey, Clement & Tchao, E. T. & Gadze, Dzisi & Akowuah, Bright & Nunoo-Mensah, Henry & Welte, Dominik & Sikora, Axel. (2022). Blockchain‐IoT peer device storage optimization using an advanced time‐variant multi‐objective particle swarm optimization algorithm. EURASIP Journal on Wireless Communications and Networking. 5. 10.1186/s13638-021-02074-3.

[41] Coutinho, K., Clark, P., Azis, F., Lip, N., & Hunt, J. (2021). Enabling Blockchain Scalability and Interoperability with Mobile Computing through LayerOne.X. *ArXiv, abs/2110.01398*.

[42] Yadav, Amrendra & Singh, Nikita & Dharmender, • & Kushwaha, Singh. (2022). Sidechain: storage land registry data using blockchain improve performance of search records. Cluster Computing. 10.1007/s10586-022-03535-0(.

[43] M. A. Cheema, H. Khaliq Qureshi, C. Chrysostomou and M. Lestas, "Utilizing Blockchain for Distributed Machine Learning based Intrusion Detection in Internet of Things," 2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS), 2020, pp. 429-435, doi: 10.1109/DCOSS49796.2020.00074.

[44] S. Zhou, H. Huang, W. Chen, P. Zhou, Z. Zheng and S. Guo, "PIRATE: A Blockchain-Based Secure Framework of Distributed Machine Learning in 5G Networks," in IEEE Network, vol. 34, no. 6, pp. 84-91, November/December 2020, doi: 10.1109/MNET.001.1900658.

[45] Y. Liu, F. R. Yu, X. Li, H. Ji and V. C. M. Leung, "Blockchain and Machine Learning for Communications and Networking Systems," in IEEE Communications Surveys & Tutorials, vol. 22, no. 2, pp. 1392-1431, Secondquarter 2020, doi: 10.1109/COMST.2020.2975911.