International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8
DOI: https://doi.org/10.17762/ijritcc.v11i8.7922
***Article Received**: 29 April 2023 **Revised**: 22 June 2023 **Accepted**: 20 July 2023*

LQMPCS: Design of a Low-Complexity Q-Learning Model based on Proof-of-Context Consensus for Scalable Side Chains

Anil Kumar Pandey[1], Ashish Kumar Misal[2]
[1]Department Of Information Technology
National Institute Of Technology Raipur
Raipur, India
anilpandey.nit@gmail.com
[2]Department Of Information Technology
National Institute Of Technology Raipur
Raipur, India
ashish.mishal153@gmail.com

Abstract—Single-chained blockchains are being rapidly replaced by sidechains (or sharded chains), due to their high QoS (Quality of Service), and low complexity characteristics. Existing sidechaining models use context-specific machine-learning optimization techniques, which limits their scalability when applied to real-time use cases. Moreover, these models are also highly complex and require constant reconfigurations when applied to dynamic deployment scenarios. To overcome these issues, this text proposes design of a novel low-complexity Q-Learning Model based on Proof-of-Context (PoC) consensus for scalable sidechains. The proposed model initially describes a Q-Learning method for sidechain formation, which assists in maintaining high scalability even under large-scale traffic scenarios. This model is cascaded with a novel Proof-of-Context based consensus that is capable of representing input data into context-independent formats. These formats assist in providing high-speed consensus, which is uses intent of data, instead of the data samples. To estimate this intent, a set of context-based classification models are used, which assist in representing input data samples into distinctive categories. These models include feature representation via Long-Short-Term-Memory (LSTM), and classification via 1D Convolutional Neural Networks (CNNs), that can be used for heterogeneous application scenarios. Due to representation of input data samples into context-based categories, the proposed model is able to reduce mining delay by 8.3%, reduce energy needed for mining by 2.9%, while maintaining higher throughput, and lower mining jitters when compared with standard sidechaining techniques under similar use cases.

Keywords-Blockchain, Sharding, Mining, PoC, Consensus, Q-Learning, LSTM, CNN, Intents.

I. INTRODUCTION

Many data security solutions, including cryptocurrencies like Bitcoin and Ethereum's smart contract service, rely on blockchain technology [1, 2, 3, 4]. As a result of the hash chain, the integrity of the blockchain's data is only as strong as the difficulty of tampering with individual blocks of information. There are a lot of technological hurdles that blockchain-based systems must overcome right now. Ethereum co-creator Vitalik Buterin claims that distributed ledgers can only have two out of the three desirable properties of decentralization, scalability, and security. Technically, scalability is a major obstacle for real-time applications and other industrial integrations of blockchain systems because of the exponential development of a blockchain's size, which makes it impossible to store, disseminate, verify, and add new blocks to the chain via Stackelberg Game (SG) [5, 6]. In Bitcoin and other traditional blockchains, a new block is added every 10 minutes or so. The block chosen for inclusion in the blockchain is the candidate block with the longest list of transactions and the highest response to the cryptographic hash computation (which is done using a method known as Proof of Work, or PoW). The miner who created the block is rewarded for their time and energy. There are many miners all vying to build the next block for the network. At regular intervals, miners produce blocks, which are then broadcast over the network, where other validation nodes check their integrity by comparing the block's headers and transaction list to those of other blocks [7, 8]. Whenever a majority of validator nodes agree that a block has a legitimate transaction list and the required minimum hash value, the block is added to the blockchain and validators maintain a shared distributed ledger. Therefore, all nodes examine each candidate block each time a block is created, which raises issues about scalability as the number of nodes grows owing to an increase in message overhead. Sharding is a mechanism that has been suggested as a way to address blockchain scalability problems SG [5].

Database sharding, in which a single database is divided into many shards so that separate transactions may be processed and verified in parallel, was the inspiration for blockchain sharding. In other words, numerous shards may execute transactions concurrently since each transaction is only sent to a small subset of the network (a shard). The number of transactions processed per second on a blockchain that has been sharded will grow in direct proportion to the number of shards that have been added (TPS) [9, 10, 11, 12]. However, as the number of shards grows, fewer nodes take part in the validation of a sharded block, making it simpler for malicious nodes to take over the consensus effort on a single shard via Graph Mining (GM) and Reputation Mining (RM) [13, 14, 15, 16]. That makes shard-based blockchains less safe, even if the number of malevolent participants is small. So, even a single fragment may be made more vulnerable to 51% assaults. Ethereum 2.0, Zilliqa, and ELASTICO are just a few examples of blockchain-based cryptocurrencies that employ sharding, and they all use a random shard distribution to provide a level playing field [17, 18, 19, 20]. This mechanism does not deter irresponsible conduct, and the shards are assigned using basic randomization, which is not secure enough to be employed in block consensus techniques protecting against a wide range of damaging assaults. On top of that, block validators' ability to protect themselves from malicious actions is severely limited while the network is reaching a consensus.Some potential works can be done on different domains using blockchain technologies. For instance,it can be applied over the works [21, 22, 23, 24, 25].

Separate approaches to blockchain sharding exist for splitting up transactions and for splitting up states. Transaction sharding allows for the processing of several transactions at once by a separate shard, which improves throughput performance [26, 27, 28, 29]. When more shards are employed, the performance boost is roughly proportionate to the total number of shards. Using ledger pruning, state sharding reduces the amount of data that must be stored by only keeping the atomic parts of a transaction. It is common practice to change which shards a node belongs to at the beginning of each epoch in both forms of sharding. The current sharding protocol uses a PoW-based random technique to allocate shards to nodes. Each node in ELASTICO participates in a proof-of-work (PoW) hashing contest to establish its unique identity before being arbitrarily placed into one of many shards SG [30, 31, 32, 33]. Similar to how hash value volatility is used during the peer discovery phase in Rapidchain's shard building approach [18], nodes on participating networks may be randomly allocated. However, these systems suffer hash computation overheads due to the fact thatPoW is used in the distributed shard grouping process to verify identities. However, the SSChain blockchain uses the current PoW algorithm for shard consensus and, because of its non-reshuffled structure, supports sharding of both transactions and states [34, 35, 36]. The security of the root chain may also be seriously compromised since SSChain operates on two separate chain architectures. Sharding protocols' block consensus methods may be broken down into the following steps [1, 19]: Each node, using a PoW-based randomization scheme, checks to see whether its neighbor is included in a shard. After nodes have been partitioned into separate shards, each shard will use intra-shard consensus to handle transactions separately. Transactions are distributed among shards based on the input address associated with each one. Once agreement has been reached within a shard, a consensus-building committee decides how the block to be added to the blockchain should be built. A hash of each SHA-256 transaction executed across all nodes is recorded in the connected block. Intra-shard takeover of a single shard presents a security issue due to the possibility that a rogue validator would cause a shard to lose consensus or process transactions incorrectly.

As a result, it is clear that existing sidechaining models use machine-learning optimization algorithms that are context-dependent, hence limiting their scalability for real-time applications. Furthermore, these models are quite intricate and need ongoing reconfiguration when employed in a dynamic setting. To address these problems, this study proposes a new kind of Proof-of-Context (PoC) consensus for scalable sidechains based on a revolutionary low-complexity Q-Learning Model. In section 3, we evaluated the proposed model's performance under varying real-time situations to that of popular sharding methods already in use. The paper concludes with observations on the proposed model's context and suggestions toimprove it for practical applications.

II. TYPE DESIGN OF THE PROPOSED LOW-COMPLEXITY Q-LEARNING MODEL BASED ON PROOF-OF-CONTEXT CONSENSUS FOR SCALABLE SIDECHAINS

Wherever On the basis of a review of existing sidechain techniques, it was determined that these models use context-specific machine-learning optimization techniques, limiting their scalability when applied to real-time use cases. In addition, these models are extraordinarily intricate and require constant reconfiguration when applied to dynamic deployment scenarios. This sectiondiscusses the design of a novel, low-complexity Q-Learning Model for scalable sidechains that is based on Proof-of-Context (PoC) consensus. Flow of the model is depicted in figure 1, where it can be observed that the proposed model initially describes a Q-Learning method for sidechain formation, which aids in maintaining a high degree of scalability even in scenarios involving large volumes of

***Article Received**: 29 April 2023 **Revised**: 22 June 2023 **Accepted**: 20 July 2023*

traffic. This model is cascaded with an innovative Proof-of-Context-based consensus that can represent input data in context-independent formats. These formats facilitate the provision of a rapid consensus based on data intent rather than data samples. To estimate this intent, a set of context-based classification models are employed, which aid in categorizing input data samples into distinct groups. These models include feature representation via Long-Short-Term-Memory (LSTM) and classification via 1D Convolutional Neural Networks (CNNs), which are applicable to a variety of application scenarios.

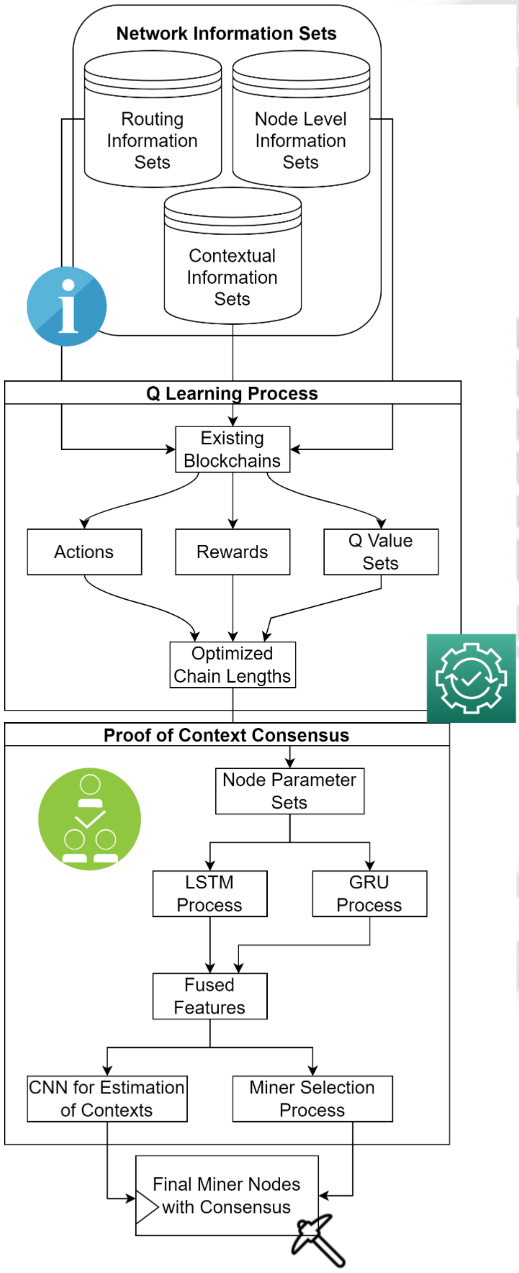

Fig. 1. Design of the proposed consensus model for scalable mining scenarios

Thus, all the collected input data samples are initially passed through a Q-Learning process, which assists in splitting current blockchain into multiple sidechains. To perform this task, the following process is used,

- Initially, find Q-Values of all the existing sidechains as per equation 1,

$$Q_i = \frac{1}{N_d}\sum_{l=1}^{N_d}(N_i + l) * d_l(read) + d_l(write) + (N_i + l - 1) * d_l(verify) * \frac{E(mining)}{E(Max)} \dots (1)$$

Where, are the current number of blocks in the chain, while are a set of dummy blocks which are added to the chain in-order-to find the block reading delay ( ), block writing delay ( ), and block verification delay ( ), which assist in estimation of the mining delay needed for addition of new blocks to individual chains. In this evaluation $E($ represents the delay needed for mining individual blocks, and the maximum delay needed for mining operations.

- Based on these values, identify new Q levels via equation 2,

$$Q(New) = Q(Current) + N_d\left(\frac{1}{d(Mining)} + Max(Q)\right) \dots (2)$$

Where, is the total delay needed for mining new blocks, and is represented via equation 3,

$$d(Mining) = \sum_{l=1}^{N_d}(N_i + l) * d_l(read) + d_l(write) + (N_i + l - 1) * d_l(verify) \dots (3)$$

- This new Q value is estimated for every sidechain (if there are no sidechains, then only single Q value is estimated by this process)
- As per this new Q value, a Q factor is estimated via equation 4,

$$Q_f = N_s * \frac{Q(New)}{\sum_{i=1}^{N_s} Q_i} \dots (4)$$

Where, represents total number of sidechains currently present in the network that are actively used for storing blocks.

- The current sidechain is split into 2 equal parts as per equation 5,

$$Q_f > 2 \dots (5)$$

- After this process, the Q value for all new sidechains is estimated, and the sidechain with minimum Q value is used for addition of new blocks

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8
DOI: https://doi.org/10.17762/ijritcc.v11i8.7922
***Article Received**: 29 April 2023 **Revised**: 22 June 2023 **Accepted**: 20 July 2023*

Table 1. Design of the context-specific block structure for PoC consensus

Prev. Hash	Source IP	Dest. IP	Other Network Info.
Nonce	Timestamp	Data Samples	Meta Data Samples
Context Flags	Meta Data of Context Flags	Current Hash	

Due to this process, the model is able to generate new sidechains, and can be scaled for larger number of nodes, which improves it usability for real-time scenarios. While adding blocks to the selected sidechain, a Proof-of-Context (PoC) based model is used, which assists in identification of miner nodes with context-specific intents. To perform this task, the model uses a contextual-block structure which can be observed from table 1.

The block structure stores the following information sets,

- Prev. Hash, which is hash of the previous blocks
- Source & Destination IP address
- Other Network Information, which stores additional information (like energy levels, network performance, etc.) about the network scenarios
- Nonce is a stochastic number which is estimated as per equation 6, and assists in uniquely identifying individual blocks

$$Nonce = STOCH\left(1, Max\left(Range\right)\right) \dots (6)$$

Where, represents a stochastic Markovian process for estimation of number sets, and is the range of available number sets. Nonce values are generated until current hash of the block is unique, when compared with existing blocks.

- Timestamp, which represents the timestamp at which the packet addition request has arrived for adding blocks
- Data Samples & Meta Data Samples, which represent values of data and their meta data value sets
- Context Flags & Meta Data of Context Flags, which contains information about the context of the miners
- Current Hash, which is the hash of current block, that is estimated using Secured Hash Algorithm (SHA) via equation 7,

$$Hash = SHA256\left(Block\right) \dots (7)$$

The context flags are used to represent miner nodes that have participated in the mining process. These miner nodes are selected via a combination of flexible-contextual feature information that is evaluated via a fusion of Long Short-Term Memory (LSTM) & Gated Recurrent Unit (GRU) operations. These operations assist in representing miner information into context-independent feature sets. To perform this task, the model initially extracts an initialization feature vector via equation 8,

$$i = var\left(x_{in} * U^{i} + h_{t-1} * W^{i}\right) \dots (8)$$

Where, are the contextual information sets about individual miners, and is estimated via equation 9,

$$x_{in} = \frac{1}{N(Mine)} \sum_{i=1}^{N(Mine)} \frac{d_i}{Max(d)} + \frac{E_i}{Max(E)} + \frac{Max(THR)}{THR_i} + \frac{100}{PDR_i} \dots (9)$$

Where, represents the delay needed for mining, energy needed for mining, throughput while mining, and packet delivery ratio while mining blocks. In equation 8, is the variance of the feature sets, and is estimated via equation 10, while are constants of the LSTM process.

$$var(x) = \frac{\left(\sum_{i=1}^{N(Mine)} \left(x_i - \sum_{j=1}^{N(Mine)} \frac{x_j}{N(Mine)}\right)^2\right)}{N(Mine)+1} \dots (10)$$

The initialization features are cascaded with scaled feature sets and operational feature sets , via equations 11 and 12,

$$f = var\left(x_{in} * U^{f} + h_{t-1} * W^{f}\right) \dots (11)$$

$$o = var\left(x_{in} * U^{o} + h_{t-1} * W^{o}\right) \dots (12)$$

To further augment these features, they are cascaded with convolutional features, that are estimated via equation 13, and converted into ternary output feature vectors via equations 14 & 15 as follows,

$$C_t' = tanh\left(x_{in} * U^{g} + h_{t-1} * W^{g}\right) \dots (13)$$

$$T_{out} = var\left(f_t * x_{in}(t-1) + i * C_t'\right) \dots (14)$$

$$h_{out} = \tanh\left(T_{out}\right) * o \dots (15)$$

Where, represents output kernel metric, that is updated for individual set of iterations. The output kernel metric is used to estimate GRU specific impedance and resistance features via equations 16 & 17, that assist in retaining high density feature sets.

$$z = var\left(W_z * \left[h_{out} * T_{out}\right]\right) \dots (16)$$

$$r = var\left(W_r * \left[h_{out} * T_{out}\right]\right) \dots (17)$$

Based on these features, the model estimates an output feature set via equation 18, and an updated feedback kernel metric via equation 19,

$$xout = (1-z) * h_t' + z * h_{out} \dots (18)$$

$$h_t' = tanh\left(W * \left[r * h_{out} * T_{out}\right]\right) \dots (19)$$

This process is repeated until variance levels of features between two consecutive evaluations is almost constant, which assists in estimation of highly variant context-independent feature sets. These feature sets are classified into 'selected', and 'non-selected' categories via equation 20, that uses an efficient set of SoftMax based activation layers.

$$c_{out} = SoftMax\left(\sum_{i=1}^{N_f} f_i * w_i + b\right) \dots (20)$$

Where,   represents total features estimated by the LSTM & GRU operations,   represents their feature values, while *w* and *b* represents individual weights and biases for different feature sets. These weights and biases are tuned by a standard 1D CNN model, that iteratively modifies these values to improve the accuracy of classification for different miners. The miners classified as 'selected' as used for the mining process. This assists in improving the mining performance for individual side chain block addition requests. This performance is evaluated in terms of mining delay, energy needed for mining, throughput levels, and PDR levels in the next section of this text.

III. RESULT ANALYSIS & COMPARISON

Before the proposed model is perceived to make use of Q-Learning for sidechain formation at the outset, which helps to keep scalability high even when dealing with massive amounts of traffic. In order to represent input data in context-independent formats, this model is cascaded with a novel Proof-of-Context based consensus. The intent of data, rather than data samples, is used by these formats to provide rapid consensus. To estimate this goal, we employ a suite of context-based classification models that aid in mapping input data samples to meaningful classes. Some of the models used here are 1D Convolutional Neural Networks (CNNs) for classification and Long-Short-Term-Memory (LSTM) for feature representation, both of which can be put to use in a wide variety of different kinds of applications. Performance of this model is estimated in terms of delay (D) needed for mining, energy (E) needed during mining, throughput (T) of mining, and packet delivery ratio (PDR) of mining request sets. These request sets were collected from the following sources,

- Cryptocurrency mining sets (https://www.kaggle.com/datasets/amritpal333/crypto-mining-data)
- Bitcoin mining sets (https://datahub.io/cryptocurrency/bitcoin)
- Cantora's Cryptocurrency Sets (https://datarade.ai/data-categories/cryptocurrency-data)
- AWS Public Blockchain Sets (https://registry.opendata.aws/aws-public-blockchain/)

All these sets were combined to form a total of 25k block addition requests, which were segregated into 60% for training, 20% for testing, and 20% for validation of the chains. Nearly 500 miner nodes were used for mining these blocks, and their 'selected', and 'non-selected' classes were decided based on their location and previous mining performance for initial block addition request sets. Based on this strategy, the delay of mining was estimated and compared with SG [5], GM [14], and RM [16] in table 2 where it was compared w.r.t. different Number of Blockchain Addition (NBA) Requests as follows.

Table 2. Delay needed for mining different blocks

NBA	D (ms) SG [5]	D (ms) GM [14]	D (ms) RM [16]	D (ms) Proposed
1250	0.98	0.99	1.01	0.71
2500	1.06	1.06	1.08	0.76
3750	1.13	1.12	1.14	0.8
5000	1.18	1.18	1.2	0.84
6250	1.24	1.24	1.28	0.9
7500	1.32	1.38	1.45	1.05
9375	1.5	1.69	1.8	1.31
11250	1.96	2.15	2.25	1.62
12500	2.46	2.59	2.66	1.89
15000	2.86	2.9	2.97	2.1
16250	3.09	3.2	3.29	2.34
17500	3.48	3.6	3.7	2.63
18750	3.91	4.03	4.15	2.95
21875	4.36	4.56	4.67	3.24
23750	4.7	5.03	5.16	3.53
25000	5.09	5.56	5.66	3.81

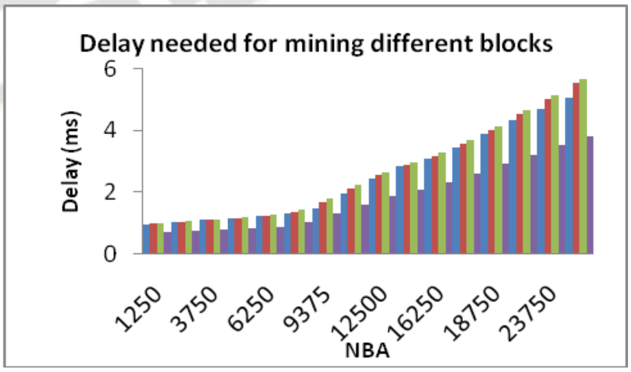

Fig 2: Delay needed for mining different blocks

As per this evaluation, it can be observed that the proposed model is able to reduce the mining delay by 10.5% when compared with SG [5], 12.4% when compared with GM [14], and 12.8% when compared with RM [16], which makes it useful for high-speed applications. The reason for this reduction in delay is use of Q-Learning for creation of contextual sidechains, which assists in reducing the effort during different mining operations. Similarly, the energy consumed during mining operations, can be observed from table 3. As per this evaluation, it can be observed that the proposed model is able to reduce the energy needed for mining by 15.4% when compared with SG [5], 19.5% when compared with GM [14], and 16.4% when compared with RM [16], which makes it useful for high lifetime applications. The reason for this reduction in energy consumption is use of Q-Learning for creation of contextual sidechains& use of LSTM & GRU based miner selection, which assists in identification of miner nodes with higher energy levels. Similarly, the throughput obtained during mining operations, can be observed from table 4

Table 3. Energy needed for mining different blocks

NBA	E (mJ) SG [5]	E (mJ) GM [14]	E (mJ) RM [16]	E (mJ) Proposed
1250	2.31	3.24	2.71	1.96
2500	2.7	3.56	2.93	2.12
3750	2.8	3.75	3.1	2.24
5000	2.99	3.98	3.28	2.37
6250	3.14	4.2	3.46	2.49
7500	3.32	4.4	3.61	2.6
9375	3.45	4.57	3.75	2.7
11250	3.59	4.74	3.9	2.82
12500	3.72	4.97	4.11	2.98
15000	3.94	5.32	4.39	3.17
16250	4.25	5.64	4.61	3.32
17500	4.52	5.9	4.79	3.44
18750	4.71	6.07	4.92	3.54
21875	4.83	6.28	5.11	3.68
23750	5.02	6.52	5.3	3.81
25000	5.22	6.75	5.48	3.94

Table 4. Throughput achieved while mining different blocks

NBA	T (kbps) SG [5]	T (kbps) GM [14]	T (kbps) RM [16]	T (kbps) Proposed
1250	316.1	294.6	317.99	454.65
2500	319.06	296.95	320.47	458.2
3750	321.14	299.17	322.94	461.88
5000	323.8	301.81	325.79	466
6250	326.84	304.5	328.68	470.06
7500	329.6	307.04	331.47	473.99
9375	332.36	309.59	334.27	477.92
11250	335.12	312.13	337.02	481.85
12500	337.88	314.68	339.77	485.78
15000	340.64	317.23	342.52	489.72
16250	343.4	319.82	345.27	493.65
17500	346.16	322.41	348.02	497.58
18750	348.91	325	350.77	501.51
21875	351.67	327.51	353.5	505.4
23750	354.43	330.01	356.22	509.29
25000	357.19	332.51	358.94	513.16

As per this evaluation, it can be observed that the proposed model is able to improve the throughput for mining operations by 19.5% when compared with SG [5], 23.5% when compared with GM [14], and 18.3% when compared with RM [16], which makes it useful for high data rate applications. The reason for this improvement in throughput is use of LSTM & GRU based selection of miner nodes, which assists in identification of contextual miner sets with higher temporal throughput levels. Similarly, the PDR obtained during mining operations, can be observed from table 5.

Table 5. PDR achieved while mining different blocks

NBA	PDR (%) SG [5]	PDR (%) GM [14]	PDR (%) RM [16]	PDR (%) Proposed
1250	74.53	75.44	77.56	86.59
2500	75.23	76.04	78.17	87.27
3750	75.72	76.6	78.78	87.96
5000	76.35	77.29	79.48	88.74
6250	77.07	77.97	80.18	89.52
7500	77.72	78.63	80.84	90.27
9375	78.36	79.28	81.52	91.01
11250	79.02	79.94	82.19	91.76
12500	79.67	80.59	82.86	92.52
15000	80.32	81.25	83.53	93.27
16250	80.97	81.9	84.21	94.01
17500	81.62	82.56	84.88	94.76
18750	82.27	83.21	85.55	95.52
21875	82.92	83.87	86.22	96.27
23750	83.57	84.52	86.89	97.02
25000	84.22	85.17	87.55	97.76

As per this evaluation, it can be observed that the proposed model is able to improve the PDR for mining operations by 12.5% when compared with SG [5], 10.4% when compared with GM [14], and 12.3% when compared with RM [16], which makes it useful for high consistency mining applications. The reason for this improvement in PDR is use of LSTM & GRU based selection of miner nodes, which assists in identification of contextual miner sets with higher temporal PDR levels. Due to these operations, the proposed model is able to improve the scalability performance for mining under real-time scenarios.

IV. CONCLUSION AND FUTURE SCOPE

It is believed that the proposed model utilizes Q-Learning for sidechain formation at the outset, which helps to maintain scalability even when massive amounts of traffic are present. This model is cascaded with a novel Proof-of-Context based consensus to represent input data in context-independent formats. These formats provide rapid consensus by relying on the intent of data rather than data samples. To estimate this objective, we employ a collection of context-based classification models that map input data samples to meaningful classes. These models include 1D Convolutional Neural Networks (CNNs) for classification and Long-Short-Term-Memory (LSTM) for feature representation, both of which are applicable to a broad range of applications. In terms of mining speed, it was discovered that the proposed model reduces mining delay by 10.5% when compared to SG [5], 12.4% when compared to GM [14], and 12.8% when compared to RM [16], making it suitable for high-speed applications. This reduction in delay is due to the use of Q-Learning for the creation of contextual sidechains, which aids in the reduction of effort during various mining operations. In terms of mining energy, it was observed that the proposed model reduces energy consumption by 15.4% when compared to SG [5], 19.5% when compared to GM [14], and 16.6% when compared to RM [16], making it suitable for long-lasting applications. This reduction in energy consumption is due to the utilization of Q-Learning for the creation of contextual sidechains and LSTM and GRU-based miner selection, which aids in the identification of miner nodes with higher energy levels. The proposed model improves the throughput of mining operations by 19.5% when compared to SG [5], 23.5% when compared to GM [14], and 18.3% when compared to RM [16], making it suitable for high data rate applications. This improvement in throughput is attributable to the use of LSTM and GRU-based miner node selection, which aids in the identification of contextual miner sets with higher temporal throughput levels. The proposed model improves the PDR for mining operations by 12.5% when compared to SG [5], 10.4% when compared to GM [14], and 12.3% when compared to RM [16], making it applicable for high consistency mining applications. This improvement in PDR is due to the use of LSTM and GRU-based miner node selection, which aids in the identification of contextual miner sets with higher temporal PDR levels. Due to these operations, the proposed model can enhance the scalability performance of mining in real-time scenarios.

In the future, the performance of this model must be validated for various use cases, and it can be extended through the application of hybrid bioinspired models that can incrementally tune mining performance in complex scenarios. Moreover, this performance can be enhanced by integrating deep learning models such as Auto Encoders (AEs) and Generative Adversarial Networks (GANs) that can anticipate mining requests and scale blockchains based on contextual parameter sets.

REFERENCES

[1] T. H. Tran, H. L. Pham, T. D. Phan and Y. Nakashima, "BCA: A 530-mW Multicore Blockchain Accelerator for Power-Constrained Devices in Securing Decentralized Networks," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 68, no. 10, pp. 4245-4258, Oct. 2021, doi: 10.1109/TCSI.2021.3102618.

[2] S. Hu, Y. Pei and Y. -C. Liang, "Sensing-Mining-Access Tradeoff in Blockchain-Enabled Dynamic Spectrum Access," in IEEE Wireless Communications Letters, vol. 10, no. 4, pp. 820-824, April 2021, doi: 10.1109/LWC.2020.3045776.

[3] J. Wang, J. Zhu, M. Zhang, I. Alam and S. Biswas, "Function Virtualization Can Play a Great Role in Blockchain Consensus," in IEEE Access, vol. 10, pp. 59862-59877, 2022, doi: 10.1109/ACCESS.2022.3176349.

[4] Y. Chen, H. Chen, M. Han, B. Liu, Q. Chen and T. Ren, "A Novel Computing Power Allocation Algorithm for Blockchain System in Multiple Mining Pools Under Withholding Attack," in IEEE Access, vol. 8, pp. 155630-155644, 2020, doi: 10.1109/ACCESS.2020.3017716.

[5] S. Jiang, X. Li and J. Wu, "Multi-Leader Multi-Follower Stackelberg Game in Mobile Blockchain Mining," in IEEE Transactions on Mobile Computing, vol. 21, no. 6, pp. 2058-2071, 1 June 2022, doi: 10.1109/TMC.2020.3035990.

[6] C. Chen, X. Chen, J. Yu, W. Wu and D. Wu, "Impact of Temporary Fork on the Evolution of Mining Pools in Blockchain Networks: An Evolutionary Game Analysis," in IEEE Transactions on Network Science and Engineering, vol. 8, no. 1, pp. 400-418, 1 Jan.-March 2021, doi: 10.1109/TNSE.2020.3038943.

[7] Y. Zuo, S. Jin, S. Zhang and Y. Zhang, "Blockchain Storage and Computation Offloading for Cooperative Mobile-Edge Computing," in IEEE Internet of Things Journal, vol. 8, no. 11, pp. 9084-9098, 1 June1, 2021, doi: 10.1109/JIOT.2021.3056656.

[8] H. Sun, W. Liu, L. Qi, X. Ren and Y. Du, "An Algorithm for Mining Indirect Dependencies From Loop-Choice-Driven Loop Structure via Petri Nets," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 52, no. 9, pp. 5411-5423, Sept. 2022, doi: 10.1109/TSMC.2021.3126473.

[9] M. Debe, K. Salah, R. Jayaraman, I. Yaqoob and J. Arshad, "Trustworthy Blockchain Gateways for Resource-Constrained Clients and IoT Devices," in IEEE Access, vol. 9, pp. 132875-132887, 2021, doi: 10.1109/ACCESS.2021.3115150.

[10] Y. Zuo, S. Jin, S. Zhang, Y. Han and K. -K. Wong, "Delay-Limited Computation Offloading for MEC-Assisted Mobile Blockchain Networks," in IEEE Transactions on Communications, vol. 69, no. 12, pp. 8569-8584, Dec. 2021, doi: 10.1109/TCOMM.2021.3113390.

[11] K. Lei, M. Du, J. Huang and T. Jin, "Groupchain: Towards a Scalable Public Blockchain in Fog Computing of IoT Services

Computing," in IEEE Transactions on Services Computing, vol. 13, no. 2, pp. 252-262, 1 March-April 2020, doi: 10.1109/TSC.2019.2949801.

[12] S. M. Alrubei, E. Ball and J. M. Rigelsford, "A Secure Blockchain Platform for Supporting AI-Enabled IoT Applications at the Edge Layer," in IEEE Access, vol. 10, pp. 18583-18595, 2022, doi: 10.1109/ACCESS.2022.3151370.

[13] Y. Liu, S. Ke, Z. Fang, M. H. Cheung, W. Cai and J. Huang, "A Storage Sustainability Mechanism With Heterogeneous Miners in Blockchain," in IEEE Journal on Selected Areas in Communications, vol. 40, no. 12, pp. 3645-3659, Dec. 2022, doi: 10.1109/JSAC.2022.3213309.

[14] B. Guidi, A. Michienzi and L. Ricci, "Steem Blockchain: Mining the Inner Structure of the Graph," in IEEE Access, vol. 8, pp. 210251-210266, 2020, doi: 10.1109/ACCESS.2020.3038550.

[15] H. Yin et al., "Proof of Continuous Work for Reliable Data Storage Over Permissionless Blockchain," in IEEE Internet of Things Journal, vol. 9, no. 10, pp. 7866-7875, 15 May15, 2022, doi: 10.1109/JIOT.2021.3115568.

[16] C. Tang, L. Wu, G. Wen and Z. Zheng, "Incentivizing Honest Mining in Blockchain Networks: A Reputation Approach," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 1, pp. 117-121, Jan. 2020, doi: 10.1109/TCSII.2019.2901746.

[17] L. -e. Wang, Y. Bai, Q. Jiang, V. C. M. Leung, W. Cai and X. Li, "Beh-Raft-Chain: A Behavior-Based Fast Blockchain Protocol for Complex Networks," in IEEE Transactions on Network Science and Engineering, vol. 8, no. 2, pp. 1154-1166, 1 April-June 2021, doi: 10.1109/TNSE.2020.2984490.

[18] G. Li et al., "GT-Chain: A Fair Blockchain for Intelligent Industrial IoT Applications," in IEEE Transactions on Network Science and Engineering, vol. 9, no. 5, pp. 3244-3257, 1 Sept.-Oct. 2022, doi: 10.1109/TNSE.2021.3099953.

[19] C. Qiu, H. Yao, C. Jiang, S. Guo and F. Xu, "Cloud Computing Assisted Blockchain-Enabled Internet of Things," in IEEE Transactions on Cloud Computing, vol. 10, no. 1, pp. 247-257, 1 Jan.-March 2022, doi: 10.1109/TCC.2019.2930259.

[20] Y. -F. Wen and C. -Y. Huang, "Exploration of Mined Block Temporarily Holding and Enforce Fork Attacks by Selfish Mining Pool in Proof-of-Work Blockchain Systems," in IEEE Access, vol. 10, pp. 61159-61174, 2022, doi: 10.1109/ACCESS.2022.3181186.

[21] G. Urkude and M. Pandey, “AgriSense: Automatic Irrigation Utility System Using Wireless Sensor Network and Web of Things,” in 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), 2019, pp. 1–6.

[22] G. Urkude and M. Pandey, “AgriOn: A comprehensive ontology for Green IoT based agriculture,” J. Green Eng., vol. 10, no. 9, pp. 7078–7101, 2020.

[23] G. Urkude and M. Pandey, “Contextual triple inference using a semantic reasoner rule to reduce the weight of semantically annotated data on fail–safe gateway for WSN,” J. Ambient Intell. Humaniz. Comput., Jan. 2021.

[24] G. Urkude and M. Pandey, “Design and Development of Density-Based Effective Document Clustering Method Using Ontology,” Multimed. Tools Appl., Apr. 2022.

[25] G. Urkude and M. Pandey, “A novel semantic representation approach for web documents using deep entity linking and multidocument support,” Int. J. Commun. Syst., vol. 35, no. 8, p. e5119, Feb. 2022.

[26] T. Wang, X. Bai, H. Wang, S. C. Liew and S. Zhang, "Game-Theoretical Analysis of Mining Strategy for Bitcoin-NG Blockchain Protocol," in IEEE Systems Journal, vol. 15, no. 2, pp. 2708-2719, June 2021, doi: 10.1109/JSYST.2020.3004468.

[27] F. Jameel, M. A. Javed, S. Zeadally and R. Jäntti, "Efficient Mining Cluster Selection for Blockchain-Based Cellular V2X Communications," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 7, pp. 4064-4072, July 2021, doi: 10.1109/TITS.2020.3006176.

[28] W. Liu, B. Cao and M. Peng, "Blockchain Based Offloading Strategy: Incentive, Effectiveness and Security," in IEEE Journal on Selected Areas in Communications, vol. 40, no. 12, pp. 3533-3546, Dec. 2022, doi: 10.1109/JSAC.2022.3213324.

[29] X. Zhang, S. Jiang, Y. Liu, T. Jiang and Y. Zhou, "Privacy-Preserving Scheme With Account-Mapping and Noise-Adding for Energy Trading Based on Consortium Blockchain," in IEEE Transactions on Network and Service Management, vol. 19, no. 1, pp. 569-581, March 2022, doi: 10.1109/TNSM.2021.3110980.

[30] C. Tang, C. Li, X. Yu, Z. Zheng and Z. Chen, "Cooperative Mining in Blockchain Networks With Zero-Determinant Strategies," in IEEE Transactions on Cybernetics, vol. 50, no. 10, pp. 4544-4549, Oct. 2020, doi: 10.1109/TCYB.2019.2915253.

[31] S. Guo, Y. Dai, S. Guo, X. Qiu and F. Qi, "Blockchain Meets Edge Computing: Stackelberg Game and Double Auction Based Task Offloading for Mobile Blockchain," in IEEE Transactions on Vehicular Technology, vol. 69, no. 5, pp. 5549-5561, May 2020, doi: 10.1109/TVT.2020.2982000.

[32] D. C. Nguyen, P. N. Pathirana, M. Ding and A. Seneviratne, "Privacy-Preserved Task Offloading in Mobile Blockchain With Deep Reinforcement Learning," in IEEE Transactions on Network and Service Management, vol. 17, no. 4, pp. 2536-2549, Dec. 2020, doi: 10.1109/TNSM.2020.3010967.

[33] A. Asheralieva and D. Niyato, "Throughput-Efficient Lagrange Coded Private Blockchain for Secured IoT Systems," in IEEE Internet of Things Journal, vol. 8, no. 19, pp. 14874-14895, 1 Oct.1, 2021, doi: 10.1109/JIOT.2021.3071563.

[34] K. Nicolas, Y. Wang, G. C. Giakos, B. Wei and H. Shen, "Blockchain System Defensive Overview for Double-Spend and Selfish Mining Attacks: A Systematic Approach," in IEEE Access, vol. 9, pp. 3838-3857, 2021, doi: 10.1109/ACCESS.2020.3047365.

[35] X. Ding, J. Guo, D. Li and W. Wu, "An Incentive Mechanism for Building a Secure Blockchain-Based Internet of Things," in IEEE Transactions on Network Science and Engineering, vol. 8, no. 1, pp. 477-487, 1 Jan.-March 2021, doi: 10.1109/TNSE.2020.3040446.

[36] Q. Wang, T. Xia, D. Wang, Y. Ren, G. Miao and K. -K. R. Choo, "SDoS: Selfish Mining-Based Denial-of-Service Attack," in IEEE Transactions on Information Forensics and Security, vol. 17, pp. 3335-3349, 2022, doi: 10.1109/TIFS.2022.3202696.