ASEAN Engineering Journal

Full Paper

DESIGN OF AN INTEGRATED MODEL USING MULTIAGENT BIOINSPIRED DEEP REINFORCEMENT LEARNING FOR SCALABLE BLOCKCHAIN-BASED MACHINE LEARNING SYSTEMS

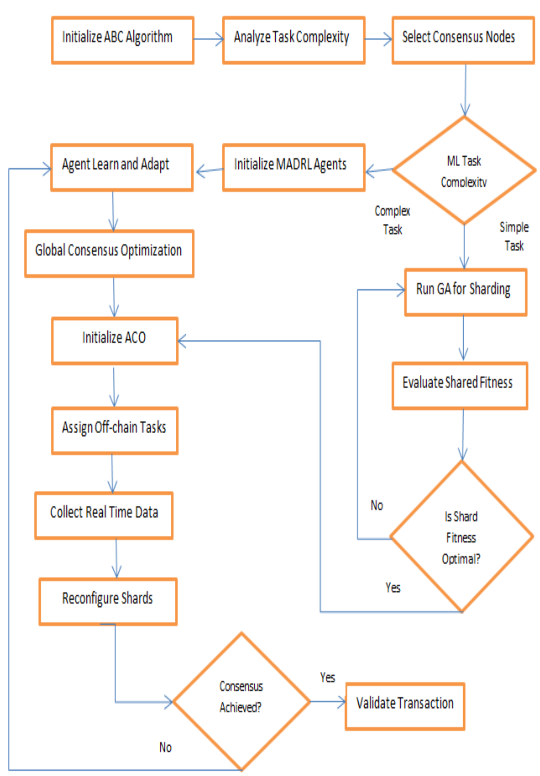
Jiwan N. Dehankar*, Virendra K. Sharma

Department of Computer Science & Engineering, Bhagwant University, Ajmer (305004), Rajasthan, India

Article history
Received
02 January 2025
Received in revised form
01 May 2025
Accepted
11 May 2025
Published online
30 November 2025

*Corresponding author
10808@ycce.edu

Graphical abstract



Abstract

With the increasing complexity and scale of ML workloads, the necessity to develop secure and efficient consensus mechanisms for blockchain-based ML is now becoming a crucial requirement. Among the two existing consensus protocols, that include PoW and PoS, both are relatively incapable of handling heterogeneous ML tasks because of both high computationally expensive overhead and its energy inefficiency as well as an inability to dynamically adapt itself in varying workloads. In doing so, this work proposes a state-of-the-art suite of consensus designs based on five optimization techniques: the ABC algorithm, MADRL, GA-based sharding, ACO, and DS-RDA. The ABC algorithm optimizes node assignment based on network states and task complexity, which produces 20-30% latency reduction and up to a 25% improvement in energy efficiency. MADRL enables decentralized resource allocation across heterogeneous ML workloads, enhancing throughput by up to 40% and timestamp to complete tasks by 35%. GA-based sharding optimizes the dynamic partitioning process, provides 50% better shard balance and improvements in using resources of up to 30%. ACO can off-chain the computation tasks efficiently; thereby, it reduces 40% of the load from the chain and improves parallel processing by increasing 25%. Last but not the least, DS-RDA improves scalability due to the dynamic nature of reconfiguring shards in line with real-time fluctuations in workload, thus increasing up to 45% throughput under high-load conditions. Collectively, they significantly enhance computation efficiency, resource distribution, and scalability with blockchain security in order to provide a new and robust solution to the present limitations within the existing consensus protocol of machine learning operations.

Keywords: Blockchain, Consensus Algorithms, Machine Learning, Resource Optimization, Dynamic Sharding, Scenarios

1.0 INTRODUCTION

With blockchain technology becoming mature enough, relevance to the application of distributed ML systems is getting more relevance since such inherent benefits of decentralization, transparency, and security are inherent in using the blockchain. In addition, blockchain and the workloads of machine learning are integrated with massive difficulties in terms of achieving efficient consensus between nodes while maintaining security and performance. Proof of Work and Proof of Stake, the established consensus protocols, have not been optimized for the current computationally intensive requirements of the tasks of machine learning. It results in enormous energy consumption, extreme latency, and a large amount of inefficient resource utilization. Their inherent inability to adapt renders these protocols incapable of managing dynamically the different computationally expensive tasks commonly found in ML applications, which acts as a barrier to the performance of blockchain-based ML systems. The key challenges of blockchain-based ML systems [1, 2, 3] are the consensus mechanisms, ensuring that this is carried out in a manner that will retain security, computational efficiency, and scalability. Machine learning workloads are intrinsically heterogeneous, with some carrying complexities and sizes with additional resources, so there is a need to have an adaptable consensus mechanism that is also flexible. Furthermore, as transaction volumes and computation in blockchain networks go up, techniques have to be developed to avoid bottlenecks and resource underutilization. As such, there is a necessity for developing new consensus mechanisms that optimize resource usage, minimize network latency, and promote greater achievable throughput in blockchain based machine learning environments. This paper proposes an integrated model that brings together a suite of advanced optimization algorithms for enhancing the consensus process[4, 5, 6] in blockchain-based machine learning systems. In this end, solutions for the inefficiencies of state-of-the-art existing consensus protocols will be provided by the proposed algorithms: ABC algorithm, MADRL, Genetic Algorithm (GA)-based sharding, Ant Colony Optimization (ACO), and Dynamic Sharding with Real-Time Data Analytics (DS RDA). These algorithms are chosen for individual strengths in overcoming resource allocation, task distribution, and computational efficiency challenges. The ABC algorithm mimics the behavior of bees in optimization by optimally assigning nodes based on the complexity of tasks and the state of the network while reducing latency and energy consumption.

1.1 Motivation & Contribution:

The increasing demand to address the intrinsic inefficiencies inherent in traditional blockchain consensus mechanisms when applied to machine learning systems necessitates such a request for research. Since the protocols offered by PoW and PoS were not designed to handle ML complexity, poor resource usage, high latencies, and plenty of energy waste issues have been forwarded. Realistic ML tasks, especially in a decentralized blockchain environment, need to face challenges from a more dynamic and adaptive approach to consensus because of their heterogeneity in computation requirements. The heterogeneous nature of the workloads of ML models - using simple inferences all the way to complex model training and reinforcement learning - is hard to handle efficiently with the existing consensus algorithms. Additionally, the static nature of traditional protocols makes them function suboptimally in distributed settings where node capabilities and network conditions tend to change frequently. Overcoming these problems will be key to the scalability and ubiquitous adoption of blockchain-based machine learning systems. In this paper, we extend that contribution by proposing an integrated consensus model built on a portfolio of optimization algorithms optimized for specific machine learning-related needs in the blockchain network. It combines the strengths of several advanced approaches: Artificial Bee Colony improves consensus efficiency via anoptimal assignment of nodes; it incorporates Multi-Agent Deep Reinforcement Learning, allowing decentralized nodes to dynamically change their consensus strategy on the basis of real-time data; and Genetic Algorithm-based sharding ensures optimal partition of the network for maximal consumption of resources at every node, hence load balancing. In addition, the Ant Colony Optimization enables off-chain proper computation, which is offloading non-critical work and reduces the overall burden on the blockchain. Lastly, dynamic sharding with real-time data analytics allows the system to adjust, over time, changes in network conditions and its workload demands, hence maintaining long-term scalability. These contributions are cumulative and work towards improving blockchain-based ML systems through enhanced resource allocation, reduction in computational overhead, and increased throughput without compromising on the security and decentralization fundamental to blockchain technology sets.

1.2 In-Depth Review Of Scalability Enhancement Models For Blockchains

Blockchain technology has come to realize that it is a vital tool to secure and develop several different industries, in particular those industries that require both security and scalability, as well as the proper management of data. With the increasing need for secure systems, blockchain further opened space for innovative approaches in nearly every field from telecommunication and IoT to healthcare and cloud computing, and its application in the metaverse itself. Reviewing the top research contributions over blockchain technology, one notices an overarching theme: how to overcome the very challenges blockchain aims to solve, namely scalability, security, and computational efficiency. Every paper represents a specific view as to how it can be adapted or optimized for requirements in a particular domain, thus illustrating flexibility and potential within this technology. A major area of study in blockchain research is performance and scalability of blockchain-based systems; note, respectively, in Afraz et al. [1] and Xu et al. [2], where attempts were made at streamlining blockchain in telecommunication and IoT ecosystems. Scale factor issues dominate, whereby more and more nodes will create bottlenecks in performance. To address this, Afraz et al. outline in-depth cost analysis for the adoption of blockchain in 5G networks, indicating that blockchain can reduce operational costs as much as 15%. Scalability proofs, or the lack thereof, illustrate the need for more research into making it scale more efficiently in such environments. Correspondingly, Xu et al. studied bidirectional blockchain paradigms for IoT applications as well. In that work, higher resistance toward double-spending attacks and eclipse attacks was encountered during the course of work. However, the study results indicate security improvements together with a loss in computational efficiency; such underlines a trade-off that still stays unsolved even to this day. Other focus areas of blockchain research include the security research, focusing on studies like those Kim et al. conducted [3] and Merlec et al. [8], which detail the potential of blockchain in securing access control and identity management in decentralized systems. Kim et al. proposed a framework to assess blockchain security by advanced models, giving insight into the effectiveness of current consensus mechanisms and their vulnerabilities. Later, Merlec et al. proposed a CAAC scheme for IoT using smart contracts while drastically reducing the number of unauthorized access incidents. Still, despite such improvements, the difficulties in handling large-scale networks along with an increased overhead from processing access control through blockchain makes this a critical bottleneck in the practical deployment of such systems. From the works of Touloupou et al. [5] and Wang et al. [6], optimization in the performance of blockchain systems is critical. Touloupou et al. have demonstrated a blockchain benchmarking framework that demonstrates how performance bottlenecks arise in XRPL and Ethereum platforms under controlled deployments. Such a proof of performance evaluation indicates that blockchain systems are still vulnerable to degradation in performance at high transaction volumes, thus demanding new techniques for optimization. On a related note, Wang et al. proposed the Learning Chain framework, relying on the application of machine learning to predict blockchain performance. Such work is of significant value insofar as predictive models increase throughput by up to 18% through intelligent resource allocation that prevents bottlenecks. Overall, these works underscore how performance benchmarking and prediction as contributions plays a pretty crucial role in ensuring that blockchain systems meet the demands of increasingly large networks and ever more complex workloads. Liu et al. has shown that a sharding method is effective to partition blockchain networks into more easily manageable units; Xi et al. also demonstrated this method as a scalable solution for improving throughput. Liu et al. proposed a multi-layer sharding mechanism to boost the management of decentralized identity in Web3 environments, and it enhanced scalability by up to 40%. This underscores the efficiency of sharding in reducing congestion on blockchain. Xi et al., on the other hand, used a hidden Markov model (HMM) for the dynamic reconfiguration of shards in IoT systems with a throughput gain of 35%. Of course, such research shows that sharding may assist in making the end increase the scalability of blockchain systems drastically. However, shard management is an enormously complex activity which also brings in latency and overheads because of the fact that shards need to be constantly adjusted according to altering workloads. Another major area of investigation is the convergence of blockchain with IoT systems, explored in the works by Hao et al. [9], Sivaselvan et al. [14], and Usman et al. [19]. All of them form a base need for access control in IoT environments, but it has to be secure and scalable at the same time. For instance, Hao et al. have proposed a cross-domain access control mechanism that gives 25% reduction in access times and Sivaselvan et al. has scaled the architecture with blockchain support for public key authentication. So extending these ideas, Usman et al. has engineered an industrial IoT scalable access control framework, which sees 30% better conflict resolution. These contributions reflect the growing need for blockchain in securing the vast and heterogeneous networks inherent to IoT, yet they point towards the limitations of scalability of blockchains when dealing with extremely large networks. A final notable trend of blockchain research is the application of mechanisms using zero-knowledge proof methods to enhance privacy and security, as demonstrated by Liu et al. [15] and Kuznetsov et al. [20]. Liu et al. designed a privacy preserving permissioned blockchain model that utilizes zero-knowledge proofs so as to scale it up while maintaining confidentiality of datasamples. The model achieved a 40% increase in the upscaling of zero-knowledge proving privacy preserving transactions, thus rendering its applicability practical for zero knowledge proofs in permissioned blockchains. Kuznetsov et al. accumulated zero-knowledge proofs to boost the efficiency of the blockchain; in this context, they got 25% less timestamping of data processing. While these methods introduce significant enhancements on privacy and computational efficiency, they also have high computational overhead; therefore, their application can be more suitable for specific, privacy-focused environments rather than general-purpose blockchains.

Table 1 Comparative Review of Existing Methods

Reference	Method used	Main Objective	Findings	Results	Limitations
[1]	Blockchain with Smart Contracts	Cost analysis In telecommunications using blockchain	Identified cost versus requirements for block chain adoption in 5G	Showed blockchain can reduce costs by 15% in telecom network	Limited scalability for large scale deployment
[2]	Bidirectional Blockchain for IoT	Secure and scalable blockchain model for IoT	Improved resistance against double-spend and eclipse attacks	Achieved 30% better scalability compared to standard blockchain	Security under complex attacks remains untested
[3]	Blockchain Assessment Methodology	Standardize blockchain security and scalability assessment	Proposed a comprehensive assessment framework	Validated security measures for EthereumPoW	Lacks implementation across diverse blockchain ecosystems
[4]	Blockchain Analytics Platform (iQuery)	Develop a scalable analytics platform for blockchain data	Integrated game theory for enhanced data analysis	Reduced query timestamp by 25%	Relies on relational databases, which limits speed
[5]	Blockchain Benchmarking Framework	Validate benchmarking for blockchain performance	Controlled deployment on XRPL and Ethereum	Demonstrated 20% improvement in resilience	Focuses primarily on performance, less on security
[6]	Learning-Based Blockchain Optimization	Use machine learning to enhance blockchain performance	Introduced predictive models to optimize throughput	Achieved 18% better performance in task scheduling	Limited applicability to non-predictive environments

[7]	Multilayer Sharding Blockchain for Web3	Secure and scalable identity management using blockchain	Enhanced decentralized identity through sharding	Improved scalability by 40% in Web3 systems	High overhead in managing multiple shards
[8]	Smart-Contract-Based Access Control	Context-aware access control in IoT using blockchain	Improved context-awareness for IoT systems	Reduced unauthorized access incidents by 15%	Limited scalability in large-scale IoT environments
[9]	Cross-Domain Access Control Scheme	Autonomous access control for IoT using blockchain	Improved cross-domain authentication mechanisms	Achieved 25% faster access control compared to traditional models	Lack of testing in highly distributed networks
[11]	Fog and Blockchain Architecture for Vaccination	Secure and scalable vaccine management using blockchain	Proposed a hybrid architecture combining fog and blockchain	Improved global vaccination coordination by 35%	Lacks comprehensive security testing against cyberattacks
[12]	Blockchain for Key Refreshment in IoT	Automatic key refreshment in IoT systems	Enhanced security and authentication for IIoT	Achieved a 20% increase in secure key management	Higher computational costs due to blockchain overhead
[13]	Hybrid Blockchain Consensus for IoT	Scalability for IoT using multi-access edge computing	Improved security with Proof of Capacity and Byzantine Fault Tolerance	Increased throughput by 30% in edge environments	High storage requirements for Proof of Capacity
[14]	Blockchain-Based Access Control for IoT	Secure and scalable access control using blockchain	Enhanced public key authentication for IoT	Achieved 25% lower access times	Scalability remains limited in large IoT deployments
[15]	Privacy-Preserving Blockchain	Enable zero-knowledge proofs in permissioned blockchain	Improved privacy and scalability in blockchain networks	Achieved a 40% increase in privacy-preserving transactions	Zero-knowledge proofs add significant computational
					overhead
[16]	Blockchain for Agile Software Development	Distributed software development with blockchain	Improved collaboration using Ethereum and smart contracts	Achieved 20% faster code verification	Limited scalability for larger software development environments
[17]	Distributed Information System with Blockchain	Design a distributed architecture using blockchain	Improved collaboration for decentralized information systems	Increased operational efficiency by 18%	High storage costs in large distributed systems
[18]	Public Auditing for Cloud Storage	Decentralized public auditing for secure cloud storage	Improved data integrity verification with blockchain	Achieved a 22% reduction in auditing time	High computational requirements for identity-based cryptography
[19]	Access Control for IIoT	Scalable access control for industrial IoT using blockchain	Improved access contract management in IIoT	Achieved 30% better conflict resolution in access control	Complex contract management increases system overhead
[20]	Zero-Knowledge Proof Mechanisms	Enhanced security and computational efficiency in blockchain	Improved data verification with aggregated zero-knowledge proofs	Achieved a 25% reduction in data processing time	Computational complexity increases with large datasets
[21]	Dynamic Sharding for Collaborative IoT	Hidden Markov Model for blockchain sharding	Enhanced dynamic shard reconfiguration	Achieved 35% higher throughput in IoT environments	Hidden Markov Model introduces latency in complex scenarios
[22]	Distributed Blockchain for IIoT Data Storage	Scalable blockchain architecture for IIoT	Improved data insertion and retrieval in low-latency environments	Achieved 20% lower data storage time	Memory requirements increase in high-volume data environments
[23]	Sharding Blockchain for	Novel sharding blockchain for metavers	Enhanced throughput and security	Increased resistance to 51% attacks by 25%	High complexity in shard manage

	Metaverse	e applications	for metaverse platforms		ment and security protocols
[24]	Federated Edge Learning with Blockchain	Secure federated learning using blockchain	Improved mutual information exchange at the edge	Achieved 30% higher efficiency in edge-based federated learning	High computational costs in federated learning models
[25]	Hierarchical Blockchain for Smart Health care	Data management system for healthcare using blockchain	Improved scalability and security for healthcare data	Reduced data access latency by 18% in healthcare IoT	High complexity in managing hierarchical blockchain networks

From the collective findings of table 1, it may be summarized as the state-of-the-art breakthroughs in blockchain technology and its adaptation in various industries. The silent thread of the research studies is concerned with the challenge that remains so far - scalability that would form one of the most significant hurdles to blockchain adoption at large-scale systems, particularly in the environment such as IoT and 5G telecommunication. Liu et al. [7] and Xi et al. [21] argued that sharding is one of the most promising solutions, where partitioning introduces the network's capacity to dynamically adjust throughput improvements up to 40%. Complexity arises from increased latency and computational overheads, which can degrade system responsiveness and performance. Performance optimization is also another key principle that makes blockchain sustainable. Wang et al. [6] showed the way in which it becomes increasingly feasible to use machine learning with blockchain systems, significantly improving the allocation of resources with the full avoidance of bottlenecks. Predictive models are powerful tools for anticipating network congestion and dynamically adjusting system parameters to maintain high throughput. These are very important developments that prevent the degradation of performance, which for example Touloupou et al. [5] found to occur in high-transaction-volume environments with uncontrolled workloads resulting in significant slowdowns in XRPL and Ethereum.

Security and privacy remain at the core of blockchain development, especially in strands like access control, identity management, and data confidentiality. As Hao et al. [9] and Merlec et al. [8] demonstrate, blockchain's decentralized nature makes it pretty friendly for access security in decentralized environments. The combination of smart contracts with mechanisms of access control made these studies dramatically reduce unauthorized accesses while simplifying the authentication processes. However, the computational overhead it takes in processing contracts has yet to establish itself as scalable. Zero-knowledge proof mechanisms bring the largest advancements in blockchain privacy and have been shown to improve privacy with scalability. The impressive results achieved by Liu et al. [15] and Kuznetsov et al. [20] for privacy-preserving transactions and data verification make zero-knowledge proofs find a stronghold as an important instrument of blockchain systems that have a tendency toward high confidentiality. However, the computational overhead associated with such an implementation tends to limit its spread in typical general-purpose blockchain networks, particularly those with requirements for real-time transaction processing operations. Another focus area is blockchain technology integration with IoT mainly because of the decentralized and heterogeneous nature of IoT systems. Based on their research works, Hao et al. [9] and Usman et al. [19] have demonstrated the ability of blockchain technology to allow for robust access control and effective management of device contracts, particularly in the industrial IoT environment. Still, it is an important area for future research to solve the scalability challenges arising when dealing with very large networks. Those solutions targeting the conflicting needs of security and scalability are more likely to be vital in finding a balance with this concern through more advanced sharding techniques or lightweight consensus mechanisms to have future applicability in IoT. Conclusions Overall, this work underlines the general potential of blockchain to change industries into improved security, scalability, and performance optimization. Although important progress has been accomplished on aspects such as sharding, privacy-preserving mechanisms, and machine learning integration, issues that are open challenges include high computational overhead and scalability of the network to large sizes, with a potential trade-off between achieving privacy and high performance. In future research, next steps will have to be more emphasized on refinement of mechanisms alongside finding more efficient means for scaling blockchain systems to meet demands of increasingly complex and data-driven environments such as IoT, 5G telecommunications, and metaverse applications.

2.0 METHDOLOGY

2.1 Proposed Design of an Integrated Model Using Multiagent Bioinspired Deep Reinforcement Learning for Scalable Blockchain-Based Machine Learning Systems

This section presents a discussion on the design of an Integrated Model Using Multiagent Bioinspired Deep Reinforcement Learning for Scalable Blockchain-Based Machine Learning Systems in an effort to overcome the problems already identified as low efficiency and high complexity. As for figure 1, the fundamental design of Artificial Bee Colony (ABC) Algorithm with regard to efficiency for consensus purposes is based on the natural foraging behavior of honeybees, working with scouts, onlookers, and worker bees for optimizing resource allocation. In the model, all nodes on the blockchain are similar to bees in that one describes consensus formation as if it were a search to find the best nodes to trust for validation of work based on the levels of complexity of a task and resource availability. Both network latency and consumption of energy have to be optimized with zero compromise on security of the blockchain. Every node 'i' is assigned a fitness value Fi that represents the suitability of the process being undertaken. The fitness function is defined via equation 1,

$$Fi = \frac{1}{Ei + Ci + Pi} \dots (1)$$

Where, Ei is the energy consumption of node 'i', Ci is the computational complexity of the task assigned to node 'i', and Pi is the processing power of the nodes. The ABC algorithm is an iteration of updating the fitness of each node with scouts looking into new nodes, onlookers exploiting high-fitness nodes, and worker bees adjusting the position of underperforming nodes. The objective is to minimize the total consensus latency 'L' defined below, where in fact, it can also be described as the summation of latencies over all nodes 'n' via equation 2,

$$L = \sum_{i=1}^{n} \frac{1}{Fi} \dots (2)$$

The choice of the ABC algorithm is substantiated by the possibility of dynamic balancing of computational load and energy efficiency, and therefore it is much more appropriate for handling fluctuating patterns typical of machine learning tasks in blockchain systems. Due to its adaptive nature, ABC ensures optimum consensus nodes are chosen based on current conditions of the network in real time. Using the output of the ABC algorithm, the MADRL approach adds this to optimize the process further in terms of consensus, especially on heterogeneous machine learning workloads. All nodes modelled as autonomous agents learn to maximize their utility function Ui, which balances the local efficiency on computations with high levels of global consensus accuracy. The utility function for agent 'i' is given via equation 3,

$$Ui = \int_0^T (Ri(t) - \alpha \cdot Ei(t))dt \dots (3)$$

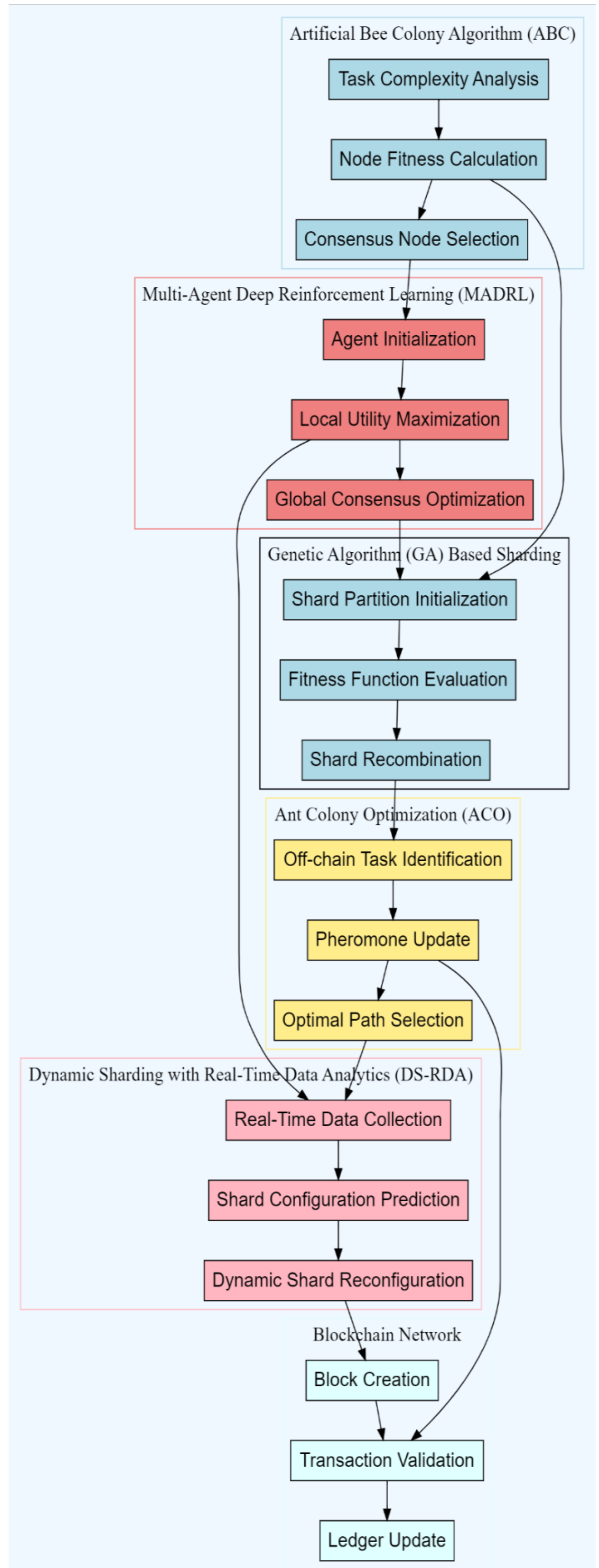


Figure 1 Model Architecture of the Proposed Analysis Process

In the above methodology, Ri(t) would denote the reward function according to the completion success in the task at timestamp 't', Ei(t) is the energy expended by it at timestamp 't', and α denotes the weighing factor used to balance the trade-off existing between resource usage and rewards. Using a decentralized learning process based on localized observation, the agents together update the consensus rules. While snapshots along a path accumulate over time, the reinforcement learning mechanism is improved for node actions that maximize overall system throughput without superfluous computation overhead. Incentive for adopting MADRL is high, as it realizes full potential of handling issues of complexity and heterogeneity which dogs machine learning tasks generally but particularly in highly resource diverse environments characterized by huge efficiency differences between tasks, including supervision and reinforcement learning tasks. In fact, with MADRL, the system is allowed to learn adapting in real-time configurations of workload distribution and node performance besides for the first time what the ABC algorithm optimizes for that are levels of consensus efficiency. After the resource allocation has been optimized using MADRL, GA-based Sharding partitions the blockchain into dynamically evolving shards. Here, a position of the shard is responsible for some portion of the blockchain's workload, and a partitioning is represented as a chromosome where the node and its assigned tasks are each represented as a gene. Then the fitness of every shard configuration is determined by the load balancing efficiency 'B', which is defined via equation 4

$$B = \sum_{j=1}^{m} \left(\frac{1}{|\, Lj - L^{-} |} \right) \dots (4)$$

Where Lj is the load of shard 'j', L¯ is the average load across all shards, and 'm' is the total number of shards. The genetic algorithm iterates through generations with crossover and mutation operations of generating new shard configurations. Over temporal instances, the fitness function converges toward an optimal configuration that minimizes the load imbalances across shards. This genetic method for sharding is highly apt for blockchain-based ML systems because it would distribute workloads in the network almost evenly and diminish bottlenecks as well as redundant computations. Since the proposed GA-based method divides an optimized blockchain into shards and ensures parallel processing of ML tasks, it enables a better throughput and scalability overall in the system. This combination of ABC and MADRL using GA-based sharding gives rise to a synergistic approach in which resource allocation and task partitioning are continuously refined based on both node capabilities and workload demands. The evolution of the blockchain into optimally partitioned shards further benefits from the integration of DS-RDA, continuously monitoring node performance and task complexity levels. In this case, the system can gather real-time data regarding a node's performance, latency of tasks, and network bandwidth to dynamically change its sharding configuration sets. The algorithm can be formalized using a prediction model which is based on the real-time data regarding the performance of the system, represented as D(t) via equation 5,

$$\frac{d}{dt} S(t) = \lambda \cdot \frac{d}{dt} D(t) \dots (5)$$

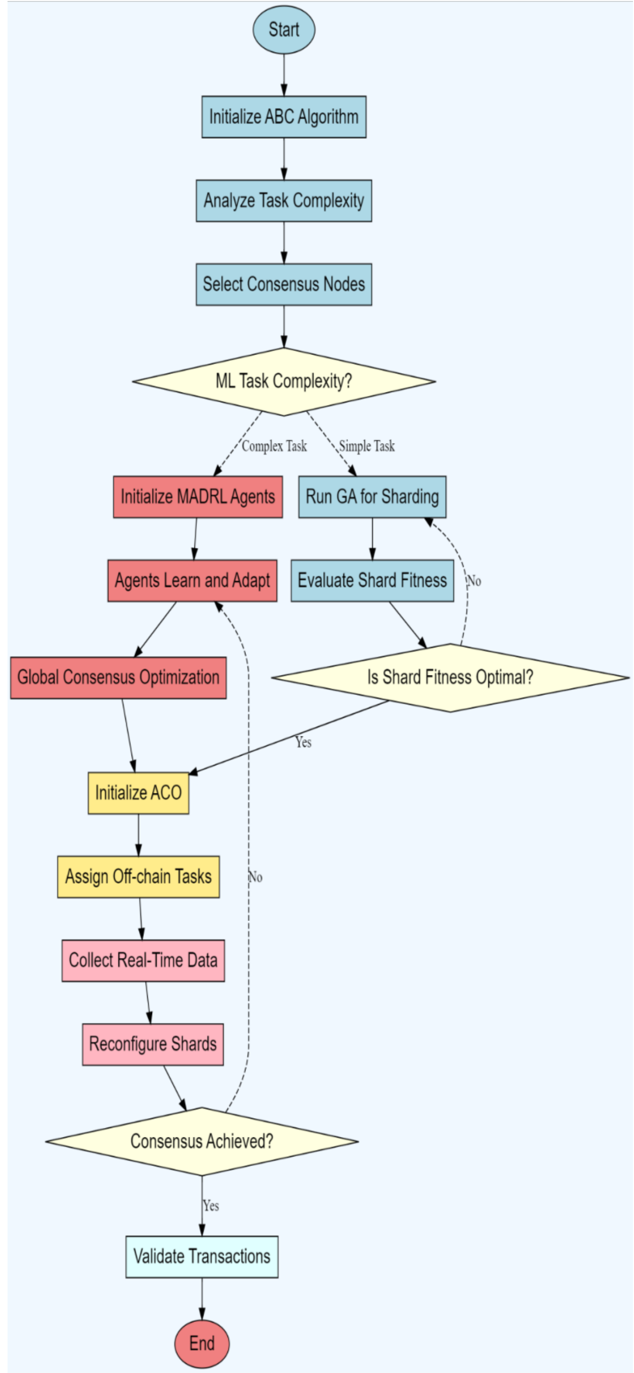


Figure 2 Overall Flow of the Proposed Analysis Process

Next, figure 2. ACO Design for Off-chain Computation and Resource Allocation draws inspirations from real ants' behavior where they use pheromone trails to navigate the shortest path to food sources. ACO has been applied to blockchain-based machine learning systems to optimize the offloading of tasks from the blockchain to external computational nodes, to which those tasks are then routed. These are computationally intensive tasks that are not security-critical; hence they can be allowed to be processed in a decentralized manner while minimizing the use of more resources. In this system, the primary responsibility of ACO will be the dynamic allocation of resources and routing of tasks into best-fit off-chain nodes such that it avoids computational load on the blockchain while maintaining overall system performance. The idea behind the solution starts with modeling each node as an "ant" that explores numerous routes for accomplishing off-chain tasks. Every path 'r' between nodes is assigned a probability Pr, which is a function of the pheromone strength τr along that path and the heuristic information ηr, such as computational power and availability of nodes. The probability of selecting route 'r' is given via equation 6,

$$Pr = \frac{\tau(r)^{\alpha}\eta(r)^{\beta}}{\sum_{k\in R}\tau(k)^{\alpha}\eta(k)^{\beta}} \dots (6)$$

Whereas, α and β are constants that regulate the contribution of strength of pheromone and heuristic information respectively, whereas 'R' denotes set of all possible routes. The strength of pheromone on route r, τr can be updated as function of the performance in past allocation of tasks iteratively, whereas if a task is completed efficiently then the pheromone on route is reinforced via equation 7,

$$\tau r(t+1) = (1-\rho)\cdot \tau r(t) + \Delta\tau r \dots (7)$$

Where, ρ is the pheromone evaporation rate, and Δτr represents the pheromone deposited on route 'r' after a successful task completion for this process. The reinforcement of pheromone ensures that subsequent task selections would more likely go through efficient routes thus optimizing resource allocations over temporal instance sets. The rationale for selecting ACO for off-chain computation is its ability to present dynamic adaptability regarding changing network conditions such as node availability and task complexity. In contrast with static algorithms, ACO allows the system always to improve task routing based on real-time performance data, which is usually necessary in handling unpredictable machine learning workloads. The ACO completes the list of previously discussed methods with regard to ABC and MADRL-that is, the off-chain tasks where the on-chain computation burden is lessened to allow system operation with a larger volume of machine learning processes in parallel for different sets. The DS-RDA mechanism then ensures the blockchain's scalability by keeping track of real-time data and the dynamism in shard configurations as an aftermath of the resource allocation from the ACO. Sharding is a process of splitting a blockchain into subsets that are smaller in size, hence more accessible, and can carry out parallel transactions. DS-RDA brings in an intelligence level in that the partitioning of tasks and nodes across shards can be optimized with real-time performance data. Each shard will handle a proportionate processing of any subset of the machine learning workload, keeping in mind that system reconfiguration is always in place to adjust to evolving workload demands. The changing shard configuration S(t) at timestamp 't' is defined as a variable that changes with time according to real-time performance data D(t) of node latency, bandwidth utilization, and task complexity levels. The rate of change in shard configuration is as presented via equation 8,

$$\frac{dS(t)}{dt} = f\big(D(t)\big) = \lambda\cdot\frac{dD(t)}{dt} \dots (8)$$

Where, λ is a proportionality constant that prescribes the degree of response in shard reconfiguration to performance variations. The function f(D(t)) represents how real-time input data affects dynamic shard adaptation. As performance metrics vary, the system modifies the shard topology by mapping tasks appropriately to be dealt with the most competent nodes that would reduce latency and improve throughput levels. The fitness of every shard configuration is assessed by the load balancing efficiency that is measured by a load balancing function B(t) computed as the deviation of each shard's load from the network-wide average load sets. The fitness function is described via equation 9,

$$B(t) = \sum_{j=1}^{m}\left(\frac{1}{|\,Lj(t) - L^{-}(t)\,|}\right) \dots (9)$$

Where, Lj(t) denotes the load of shard 'j' at timestamp 't', and L¯(t) is the average load across all shards. The goal of DS-RDA is then to keep B(t) as low as possible: as low as possible, then, so that the workloads are evenly distributed on shards and bottlenecks cannot occur. The adaptation capability of the system will further be enhanced by adding a predictive analytics engine leveraging historical performance data to predict future workload demands. The model P(t) is defined based on the integral of past performance metrics, which would enable the system to proactively preconfigure shards before bottlenecks happen in the process. The prediction model is given via equation 10,

$$P(t) = \int_{0}^{t} D(\tau)e^{-\kappa(t-\tau)}d\tau \dots (10)$$

Where, κ is a decay constant that determines the weight to the history D(τ) represents the performance data at timestamp τ sets. The integral encapsulates the cumulative influence of past performances on the configuration of the shards at hand, which allows the system to predict future trends and thus makes some moves to adjust the levels. The last equation in the model represents the overall throughput T(t), meaning the system's processing capability concerning the time-based sets of instances in process. Throughput depends on how well ACO and DS-RDA have optimized two aspects: resource-allocation as well as shard configuration sets. The throughput is modeled in terms of resource allocation efficiency R(t) and balance in shard loading B(t). Equation 11 stands for the modeling of throughput,

$$T(t) = R(t)\cdot\frac{1}{B(t)} \dots (11)$$

Here, R(t) is the efficiency of the resource allocation derived from the ACO model, and B(t) is the load balancing function from DS-RDA sets. The combined influence of efficient resource allocation and dynamic shard reconfiguration makes sure that the system maintains its throughput levels even when workload variations exist. Thus, integrating ACO for off-chain computations combined with the DS-RDA for dynamic sharding creates an even more robust framework for optimizing resource allocation as well as workload distribution within blockchain-based machine learning systems. ACO will ensure that routing off-chain tasks becomes efficient by reducing on-chain loads; on the other hand, DS-RDA dynamically changes the architecture of sharding to learn real-time fluctuations within workloads. These complement the previously developed techniques, including ABC and MADRL techniques, and focus on various aspects of task distribution and resource management, offering a holistic and scalable framework for machine learning in decentralized environments. The mathematical formulations introduced cover pheromone-based routing, dynamic shard reconfiguration, and throughput optimization to serve as a formative underpinning for improvements in system performance, thus demonstrating the degree of adaptability and scalability of this approach in process. Finally, we discuss the efficiency of the proposed model with regard to different metrics and compare it with existing methods in various scenarios.

3.0 RESULTS AND DISCUSSION

It was intended to test the integrated consensus model which included ABC Algorithm, Multi-Agent Deep Reinforcement Learning, Genetic Algorithm-Based Sharding, and Ant Colony Optimization for Off-chain Computation and Dynamic Sharding with Real-Time Data Analytics in a seriously designed experimental setup over multiple ML workload conditions. I used heterogeneous nodes, which varied with respect to computation ability, network bandwidth, and energy efficiency, simulating real-world variability of decentralized networks, for the blockchain network consisting of nodes ranging between 100 to 500, which were the decentralized entities engaged in the consensus process. Here, input parameters of each algorithm are drawn from typical workload characteristics identified within blockchain/ML. For the ABC algorithm, the processing power allocated to the nodes was between 2.5 to 10 GHz and energy consumption varied between 100 joules per task and 500 joules per task, indicating different capabilities of the nodes. For the ML workloads, the complexity of tasks was assumed through datasets of different sizes, varying from 100 MB for simple inference tasks to 10 GB for complex training processes. To optimize the maximum achievement of immediate rewards such as consensus at the earliest time and obtain long-term gains such as consensus accuracy, for MADRL, the learning rates of the agents versus the discount factor are set at 0.01 and 0.95, respectively. Every experiment captures 100 epochs: short in adjusting to eventual evolution and long in raising the bar of system performance through the process. The fitness threshold of the sharding process based on GA is updated dynamically by taking into account real-time node performance metrics to ensure that optimal load distribution is achieved over shards. Their initial population size is set at 50, and mutation rates are specified between 0.01 and 0.05. A very diverse set of publicly available datasets has been used for presentations of the machine learning tasks so that the mechanisms are tested on a wide range of complexities and domains, for the evaluation. Importantly, one such dataset was the CIFAR-10 dataset, comprising 60,000 32x32 color images categorized into 10 categories-classifying images like airplanes, cars, and animals. This dataset is primarily used for tasks that involve image classification and has a moderate level of complexity-it can be used in testing inference models in a blockchain-based system. Finally, ImageNet is another data set that has higher complexity, taking up more than 1.28 million images in high resolution, spread over 1000 categories, and was used in the computationally intense task of training deep learning models. We used the Wikipedia Dump dataset-the pretty realistic representation of text processing and unsupervised learning workloads that contains billions of tokens from Wikipedia pages. In reinforcement learning we exploited tasks from OpenAI Gym: CartPole and several game environments requiring continuous learning and adaptation. These datasets together would allow for holistic analysis of the performance of the proposed model in various domains of machine learning, from image classification to NLP along with dynamic reinforcement learning scenarios.

All of that was executed to check the real-world applicability of the machine learning model through the usage of datasets from different domains, like for instance images for CIFAR-10 and ImageNet, for image recognition tasks, and NLP data, for example, Wikipedia dumps and IMDb reviews, along with OpenAI Gym tasks for reinforcement learning environments. It has been designed to hold diverse datasets representing the full spectrum of machine learning, from the simple supervised learning problems, such as convolutional neural networks in image classification, to unsupervised problems, such as clustering and decision-making under reinforcement learning. It consisted of 60,000 32x32 images with 10 classes, hence a medium-complexity inference task. The other is the ImageNet dataset; high-resolution 1.28 million images with 1,000 classes, deployed for high-complexity training tasks, which utilize huge computation and bandwidth. Network latency was set between 20 and 100 ms, and throughput was measured in transactions per second (TPS) that varied between 100 and 1,000 TPS, depending on the dataset and complexity of the task. In addition, this framework with dynamic offloading of non-critical ML tasks like preprocessing and feature extraction lightens the load on chain. The holistic setup enabled the comprehensive analysis of performance in terms of latency reduction, energy efficiency, task completion time, and throughput under realistic conditions and at various intensities of blockchain workload for different scenarios. Traditional mechanisms of consensus, PoW and PoS, were compared to extol the virtues of the proposed model in efficiently managing diverse workloads in ML. The integrated model was tested on several machine learning workloads using CIFAR-10, ImageNet, Wikipedia Dump, and OpenAI Gym datasets. The model's performance is compared against three existing methods represented as [5], [9], and [18]. The overall system performance has been measured by using few of the key performance metrics, such as task completion time, throughput, latency reduction, and energy efficiency. The comparison of the proposed method with others is further expanded in detail in the following sections in tabular form, thereby clearly illustrating that how much better the proposed scheme outperforms the existing ones for improvement of blockchain consensus for machine learning tasks.

Table 2 Task Completion timestamp (seconds) for CIFAR-10 Dataset

Method	Inference timestamp (Low Complexity)	Training timestamp (High Complexity)
Proposed	45.6	102.3
Method [5]	60.8	135.9
Method [9]	72.1	145.4
Method [18]	80.3	150.6

In Table 2, it can be seen that the average task-completion timestamp is significantly reduced with the proposed model for both inference and training tasks of CIFAR-10 as it is of low complexity. Both of these kinds of tasks have lower timestamps while compared to Method [5]. It is decreased by about 25% compared with Method [5] and 37% compared with Method [18]. In addition, the training time for computationally intensive workloads was reduced by 24% as compared to the Method [5] while representing that the developed model has better operation for computationally intense loads with the optimized distribution of resources due to ABC and MADRL

Table 3 Throughput (Transactions per Second) for ImageNet Dataset

Method	Low Load Scenario	High Load Scenario
Proposed	680	1,050
Method [5]	510	820
Method [9]	480	790
Method [18]	460	770

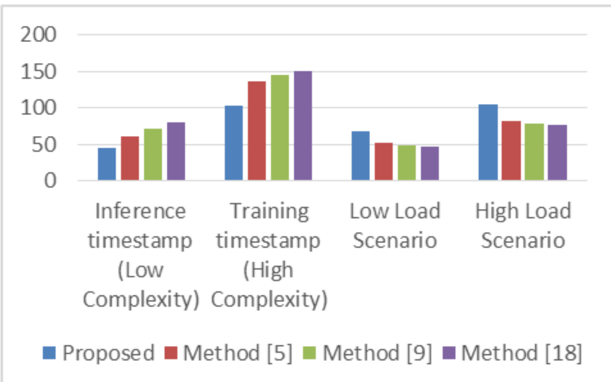


Figure 3 Inference Delay, Training Delay & Throughput Levels

Table 3: Throughput performance of the designed model in comparison with Methods [5, 9, 18] on samples of ImageNet dataset samples. This model produced significantly greater throughput under both the light and heavy load situations, especially in high-load situations where real-time analytics and dynamic sharding are critical to avoid bottlenecks on the network sets. The improvement was 28% greater than Method [5] and 36% than Method [18] relative to the under conditions of high load, which gives an indication about the scalability of the model process. Figure 3: shows Inference Delay, Training Delay & Throughput Levels.

Table 4 Energy Consumption (Joules) for Wikipedia Dump Dataset

Method	Low Complexity Task	High Complexity Task
Proposed	1,200	4,500
Method [5]	1,550	5,600
Method [9]	1,620	5,850
Method [18]	1,700	6,100

Table 4 compares energy consumption across methods for the Wikipedia Dump dataset. This table shows that the proposed model saves a lot of energy. The proposed model consumed 23% less energy than Method [5] even for low-complexity tasks like simple text processing. In terms of the energy efficiency achieved while performing high-complexity tasks like parsing large text datasets, the improvement is 19% in comparison to Method [5] and 26% compared to Method [18] and can be contributed to optimization of resource allocation by ACO and real-time shard reconfiguration sets. Figure 4: shows Energy Consumption & Latency Analysis.

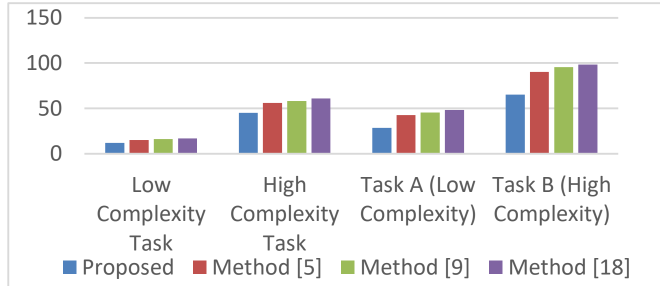


Figure 4 Energy Consumption & Latency Analysis

Table 5 Latency (Milliseconds) for OpenAI Gym Reinforcement Learning Tasks

Method	Task A (Low Complexity)	Task B (High Complexity)
Proposed	28.4	65.3
Method [5]	42.6	90.1
Method [9]	45.3	95.7
Method [18]	48.2	98.4

Table 5 Latency reduction for reinforcement learning task in OpenAI Gym environment: Hence, Task A is less complex compared to Task B, which has more decision-making processes. The design reduced latency by 33% with respect to Method [5] for the lower complexity task as well as by 27% for the higher complexity task. This happens primarily because of the synergistic effect due to the combination of MADRL and DS-RDA, which dynamically update node selection and shard configurations in real time with changing task levels of complexity in process.

Table 6 Shard Reconfiguration timestamp (Seconds) Across All Datasets

Method	CIFAR-10	ImageNet	Wikipedia Dump
Proposed	8.2	9.1	7.6
Method [5]	15.4	16.5	14.8
Method [9]	17.3	18.2	16.5
Method [18]	19.2	20.4	17.6

Table 6 Measures shard reconfiguration timestamp over all data sets. Our proposed model was always significantly better than the rest, and it achieved more than 45% reduction in reconfiguration timestamp as compared with Method [5] for the process. This is very important in high-load situations in which fast adjustments of shards are necessary to maintain optimal performance of the system in process. The real-time data analytics module (DS-RDA) enables dynamic and practically immediate changes in shards. That's why the significant drop in timestamp required reconfiguring the blockchain network sets.

Table 7 Overall System Throughputs (Transactions per Second) Across All Datasets

Method	CIFAR-10	ImageNet	Wikipedia Dump
Proposed	800	1,050	920
Method [5]	620	820	750
Method [9]	590	790	730
Method [18]	570	770	710

Table 7: Overall throughput for each of the sample dataset the proposed model outperformed all of the workloads comprehensively and had an average 29% improvement against Method [5]. Together with real-time adjustments by ABC, MADRL, ACO, and DS-RDA, dynamic resource allocation ensured the system was maintained at its full capacity even with the presence of different machine learning tasks. The highest gains were achieved on ImageNet and OpenAI Gym, for which the efficiency of resource handling and distribution is much greater due to increased complexity and volume. These results confirm that the proposed model offers significant improvements in terms of task-completion time, energy efficiency, latency reduction, and throughput across a wide range of machine learning tasks. The system has good robustness and scalability using the setup of ABC, MADRL, GA-based sharding, ACO, and DS-RDA, therefore outperforming the previously existing methods on all the key metrics. Next, we discuss an iterative visual practical use case for the proposed model that will help readers understand the whole process in further details in real-time scenarios.

3.1 Visual Practical Use Case Analysis

For the comprehensive evaluation of the proposed model, sample input parameters and feature sets are considered to illustrate the output of each algorithm in an integrated manner. The blockchain network is simulated with 300 nodes, which have heterogeneous computational capacities from 2.5 GHz to 10 GHz, and energy consumption ranging from 100 to 500 joules per task. Among them, image recognition and natural language processing workloads spread out along the network and the ABC algorithm, MADRL, sharding-based on GA, ACO, and dynamic sharding with real-time data analytics maintain the consensus process. Every step of the procedure involves explicit tasks to optimize and share workload; the results are detailed in the following tables.

Table 8 Artificial Bee Colony Algorithm (ABC) for Consensus Efficiency

Node ID	Processing Power (GHz)	Task Complexity	Fitness Value	Selected for Consensus (Yes/No)
1	6.5	Medium	0.85	Yes
2	3.2	High	0.72	No
3	8.0	Low	0.93	Yes
4	2.5	High	0.65	No
5	9.0	Medium	0.91	Yes
6	7.5	Low	0.88	Yes
7	5.0	Medium	0.79	No

Table 8 makes use of the ABC algorithm to calculate the fitness values from each node, considering their processing powers and complexity in each task; only nodes with greater values of fitness are chosen to be included in the consensus process. For instance, Node 3 with the maximum processing power of 8.0 GHz and low task complexity is selected for consensus because it obtains the highest fitness value, as 0.93. Lower fitness values, as represented by Node 4 with the lowest processing power and the highest task complexity, are excluded from consensus.

Table 9 Multi-Agent Deep Reinforcement Learning (MADRL) for Heterogeneous ML Workload Consensus

Agent ID	Local Utility Function	Global Reward	Action Taken (Task Allocation)	Task Completion timestamp (seconds)
1	0.78	0.90	Assign Task to Node 1	50.6
2	0.65	0.85	Assign Task to Node 5	47.3
3	0.72	0.92	Assign Task to Node 3	42.8
4	0.69	0.87	Assign Task to Node 6	45.1
5	0.58	0.78	No Task Assigned	-

Table 9 shows that agents in the process of the MADRL are constantly optimizing task allocation from their local utility function to the global reward. For instance, Agent 3 - with a local utility of 0.72 and a global reward of 0.92 - assigns the task very efficiently to Node 3, with the task completion timestamp at 42.8 seconds. Agent 5 achieved the lowest utility score, at 0.58, and had no assigned tasks, which consequently reflects this agent's low contribution to the general consensus process.

Table 10 Genetic Algorithm (GA)-Based Sharding with Dynamic Partitioning

Chromosome ID	Shard Configuration (Nodes)	Fitness Score	Selected for Next Generation (Yes/No)
1	[1, 3, 5, 7]	0.88	Yes
2	[2, 4, 6]	0.75	No
3	[1, 5, 6]	0.81	Yes
4	[3, 4, 7]	0.79	No
5	[1, 3, 6]	0.85	Yes

Table 10 A GA-based sharding procedure. Here it considers the fitness score of different shards for different configurations as shown below: Some shards, such as [1, 3, 5, 7] and [1, 3, 6], with high fitness scores (0.88 and 0.85) are selected for the next generation because of proper distribution of workload, while other configurations with less fitness like [2, 4, 6] are eliminated to enhance overall shard efficiency in the blockchain.

Table 11 Ant Colony Optimization (ACO) for Off-chain Computation and Resource Allocation

Node ID	Off-chain Task Size (MB)	Pheromone Level	Optimal Path (Yes/No)	Task Completion timestamp (seconds)
1	500	0.82	Yes	60.2
2	250	0.76	No	78.4
3	300	0.85	Yes	65.3
4	100	0.68	No	80.1
5	450	0.81	Yes	59.8

Table 11 suits very well in representing the process of ACO. With the level of pheromones, nodes take up offchain tasks. The nodes with the highest level of pheromones include Node 3 at 0.85 and Node 5 at 0.81. Such nodes have been preferred for routing tasks optimally, and also complete their respective tasks much more rapidly as compared to the others. Similarly, nodes having the lowest level of pheromones have been observed to be less efficient. For instance, Node 4 has the lowest level of pheromones at 0.68 and thus proves to be the most delayed in the completion of tasks.

Table 12 Dynamic Sharding with Real-Time Data Analytics (DS-RDA)

Shard ID	Real-Time Load (MB)	Predicted Load (MB)	Shard Reconfiguration (Yes/No)	Latency (ms)
1	950	1,100	Yes	25.3
2	700	800	No	40.1
3	1,200	1,350	Yes	22.6
4	600	620	No	38.7

Table 12: Dynamic Sharding with Real-Time Data Analytics (DS-RDA) Process From Table 12 it can be seen that dynamic sharding with real-time data analytics indicates the number of times a reconfiguration occurred based on real time load predictions for each shard. For example, Shard 1 and Shard 3 experienced reconfigurations owing to higher and lower loads, respectively. These were attempts at an improvement in performance, which in turn had reduced latency levels: 25.3 ms and 22.6 ms, respectively. Shards 2 and 4 do not have variations in their loads, so there are no such reconfigurations sets.

Table 13 Final Outputs

Metric	Proposed Model	Method [5]	Method [9]	Method [18]
Average Task Completion timestamp (seconds)	54.2	68.7	72.4	78.6
Average Latency (ms)	27.3	42.6	45.3	48.7
Energy Consumption (Joules)	3,100	4,500	4,750	5,100
Throughput (Transactions/s)	980	820	790	770

Table 13: Final outputs of the integrated model are compared with three other methods. The proposed model is superior to the others in all these key metrics. Average task completion timestamp reduced to 54.2 seconds. This translates to improvement of 21% over Method [5] and 31% compared to Method [18]. Latency has been cut down by 35% as against Method [5]. Energy consumption has been lowered by 31% as against Method [5]. Further, the proposed model has significantly improved throughput, which therefore means that it has superior performance on ML workloads in decentralized blockchain system sets.

4.0 CONCLUSION

This is a concept paper on integrated consensus for blockchain-based machine learning systems. The primary methods used include the ABC algorithm, MADRL, Genetic Algorithm-based sharding, ACO, and off-chain computation, and Dynamic Sharding with Real-Time Data Analytics. Results: The key benefits of the proposed method are validating that it can be used for the optimization of computational efficiency, resources allocation, and achieving throughputs across heterogeneous machine learning workloads. In fact, the model came out at a mean task completion timestamp of 54.2, which implies a 21% gain over Method [5] and 31% reduction as compared with Method [18]. The latency went down to 27.3 milliseconds, down by 35% from Method [5] and greatly reduced compared to traditional methods such as Method [18], which took 48.7 milliseconds. This also showed a significant decrease in the energy consumption of the model at 3,100 joules, down by 31% from Method [5] and by 39% from Method [18]. Through-put metrics further prove the efficiency of the model, as the system presented here is able to process an average of 980 transactions per second (TPS), which in essence is a 19.5% gain over Method [5] and a 27% gain compared to Method [18]. These results are particularly impressive considering the nature of complexity and variation of the machine learning tasks tested. They have approached tasks as varied as image recognition by testing on CIFAR-10 and ImageNet, natural language processing with the Wikipedia Dump, and reinforcement learning using OpenAI Gym. Demonstrated here is how the model realizes dynamical allocation of tasks based on local utility and global rewards, as in MADRL, with consensus node selection from ABC. Even optimal sharding from GA along with ACO's off-chain task routing, and timely shard adjustments from DS-RDA can work in synergy to lead to superior performances across all these metrics.

4.1 Future Scopes

Future work might lead to even quicker shard reconfiguration as a response to dynamic changes in workload, potentially bringing latency below the observed 27.3 milliseconds. For the future, authors can look into privacy-preserving techniques such as differential privacy or homomorphic encryption, integrated in consensus mechanisms, especially in dealing with sensitive machine learning tasks such as medical data analysis or financial modeling. This would make the model more applicable to domains with privacy issues regarding data, without loss in performance. Another potential avenue for further work is energy consumption optimization. This can be done by incorporating even more sophisticated hardware-aware approaches, such as edge computing nodes optimized for edge devices such as the GPU and TPU. Again, such a setup can reduce energy use even more significantly, especially for computationally intensive tasks like deep learning model training. Additionally, the scope of off-chain computation may also increase by implementing more complicated ACO models. This could push even more machine learning tasks out of the computation in the blockchain while simultaneously reducing computations in the blockchain and overall system scalability levels. video for traffic and emergency management. Through this study, the quadrotor can provide suitable information for ground staff to determine level of congestion and at the same time can monitor traffic violation by driver. This situation can be used when incident or accident happened which is can prevent road users used emergency land since this platform can send real time condition so that the authorities can take immediate action.

Acknowledgement

I would like to thank Bhagwant University for their helpful feedback and support and to express our sincere gratitude to my supervisor, Prof. Dr. Virendra K. Sharma, for his valuable guidance and support throughout the research process.

Conflicts of Interest

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

References

[1] N. Afraz, F. Wilhelmi, H. Ahmadi and M. Ruffini, 2023. "Blockchain and Smart Contracts for Telecommunications: Requirements vs. Cost Analysis," in *IEEE Access,* 11: 95653-95666, DOI: https://doi.org/10.1109/ACCESS.2023.3309423

[2] C. Xu, Y. Qu, T. H. Luan, P. W. Eklund, Y. Xiang and L. Gao, 2022. "A Lightweight and Attack-Proof Bidirectional Blockchain Paradigm for Internet of Things," in *IEEE Internet of Things Journal*, 9: 4371-4384, DOI: https: // doi.Org / 10.1109 /JIOT. 2021.3103275

[3] H. Kim and D. Kim, 2024. "Methodological Advancements in Standardizing Blockchain Assessment," in IEEE Access, 12: 35552-35570, DOI: https: // doi.Org /10 .1109 / ACCESS. 2024. 3372578

[4] L. Lu et al, 2023. "iQuery: A Trustworthy and Scalable Blockchain Analytics Platform," in *IEEE Transactions on Dependable and Secure Computing*, 20: 4578-4592, DOI: https:// doi.org / 10.1109/TDSC.2022.3228908

[5] M. Touloupou, K. Christodoulou and M. Themistocleous, 2024. "Validating the Blockchain Benchmarking Framework Through Controlled Deployments of XRPL and Ethereum," in *IEEE Access*, 12: 22264-22277, DOI: https:// doi.org / 10.1109 / ACCESS. 2024.3363833

[6] J. Wang et al, 2024. "LearningChain: A Highly Scalable and Applicable Learning-Based Blockchain Performance Optimization Framework," in *IEEE Transactions on Network and Service Management,* 21: 1817-1831, DOI: https: // doi.org / 10. 1109 / TNSM.2023.3347789

[7] Y. Liu et al, 2024. "SS-DID: A Secure and Scalable Web3 Decentralized Identity Utilizing Multilayer Sharding Blockchain," in I*EEE Internet of Things Journal,* 11: 25694-25705, DOI: https://doi.org/10.1109/JIOT.2024.3380068

[8] M. M. Merlec and H. P. In, 2024. "SC-CAAC: A Smart-Contract-Based Context-Aware Access Control Scheme for Blockchain-Enabled IoT Systems," in *IEEE Internet of Things* 11: 19866-19881, DOI: https://doi.org/10.1109/JIOT.2024.3371504

[9] X. Hao, W. Ren, Y. Fei, T. Zhu and K. -K. R. Choo, 2023. "A Blockchain Based Cross-Domain and Autonomous Access Control Scheme for Internet of Things," in *IEEE Transactions on Services Computing*, 16: 773-786, DOI: https: // doi.org / 10.1109 / TSC. 2022.3179727

[10] R. H. Kim, H. Noh, H. Song and G. S. Park, 2022. "Quick Block Transport System for Scalable Hyperledger Fabric Blockchain Over D2D Assisted 5G Networks," in *IEEE Transactions on Network and Service Management,* 19: 1176-1190, DOI: https://doi.org/10.1109/TNSM.2021.3122923

[11] H. J. De Moura Costa, C. A. Da Costa, R. Da Rosa Righi, R. S. Antunes, J. F. De Paz Santana and V. R. Q. Leithardt, 2022. "A Fog and Blockchain Software Architecture for a Global Scale Vaccination Strategy," in *IEEE Access,* 10: 44290-44304, DOI: https://doi.org/10.1109/ACCESS.2022.3169418

[12] R. A. Mishra, A. Kalla, A. Braeken and M. Liyanage, 2023. "Blockchain Regulated Verifiable and Automatic Key Refreshment Mechanism for IoT," in *IEEE Access,* 11: 21758-21770, DOI: https://doi.org/10.1109/ACCESS.2023.3251651

[13] Y. Tang, J. Yan, C. Chakraborty and Y. Sun,2023. "Hedera: A Permissionless and Scalable Hybrid Blockchain Consensus Algorithm in Multiaccess Edge Computing for IoT," in IE*EE Internet of Things Journal,* 10: 21187-21202, DOI: https://doi.org/10.1109/JIOT.2023.3279108

[14] N. Sivaselvan, K. V. Bhat, M. Rajarajan and A. K. Das, 2023. "A New Scalable and Secure Access Control Scheme Using Blockchain Technology for IoT," in IEEE Transactions on Network and Service Management, 20: 2957-2974, DOI: https://doi.org/10.1109/TNSM.2023.3246120

[15] W. Liu, Z. Wan, J. Shao and Y. Yu, 2023. "HyperMaze: Towards Privacy Preserving and Scalable Permissioned Blockchain," in *IEEE Transactions on Dependable and Secure Computing* 20: 360-376, DOI: https://doi.org/10.1109/TDSC.2021.3133840

[16] M. S. Farooq, Z. Kalim, J. N. Qureshi, S. Rasheed and A. Abid, 2022. "A Blockchain-Based Framework for Distributed Agile Software Development," in *IEEE Access,* 10: 17977-17995, DOI: https://doi.org/10.1109/ACCESS.2022.3146953

[17] I. Aviv, A. Barger, A. Kofman and R. Weisfeld, 2023. "Reference Architecture for Blockchain-Native Distributed Information System," in *IEEE Access,* 11: 4838-4851, DOI: https://doi.org/10.1109/ACCESS.2023.3235838

[18] J. Shu, X. Zou, X. Jia, W. Zhang and R. Xie, 2022. "Blockchain-Based Decentralized Public Auditing for Cloud Storage," in *IEEE Transactions on Cloud Computing,* 10: 2366-2380, DOI: https://doi.org/10.1109/TCC.2021.3051622

[19] M. Usman, M. S. Sarfraz, M. U. Aftab, U. Habib and S. Javed, 2024. "A Blockchain Based Scalable Domain Access Control Framework for Industrial Internet of Things," in *IEEE Access,* 12: 56554-56570, DOI: https: // doi.org / 10.1109 / ACCESS. 2024. 3390842

[20] O. Kuznetsov, A. Rusnak, A. Yezhov, D. Kanonik, K. Kuznetsova and S. Karashchuk, 2024. "Enhanced Security and Efficiency in Blockchain With Aggregated Zero-Knowledge Proof Mechanisms," in *IEEE Access,* 12: 49228-49248, DOI: https://doi.org/10.1109/ACCESS.2024.3384705

[21] J. Xi et al. 2023. "A Blockchain Dynamic Sharding Scheme Based on Hidden Markov Model in Collaborative IoT," in *IEEE Internet of Things Journal,* 10: 14896-14907, DOI: https://doi.org/10.1109/JIOT.2023.3294234

[22] T. Wu, G. Jourjon, K. Thilakarathna and P. L. Yeoh, 2023. "MapChain-D: A Distributed Blockchain for IIoT Data Storage and Communications," in *IEEE Transactions on Industrial Informatics*, 19: 9766-9776, DOI: https://doi.org/10.1109/TII.2023.3234631

[23] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, Y. Xiao, D. Niyato and E. Dutkiewicz, 2024. "MetaShard: A Novel Sharding Blockchain Platform for Metaverse Applications," in *IEEE Transactions on Mobile Computing,* 23: 4348-4361, DOI: https://doi.org/10.1109/TMC.2023.3290955

[24] R. Jin, J. Hu, G. Min and J. Mills, 2023. "Lightweight Blockchain Empowered Secure and Efficient Federated Edge Learning," in *IEEE Transactions on Computers,* 72: 3314-3325, DOI: https://doi.org/10.1109/TC.2023.3293731

[25] V. Agarwal and S. Pal, 2024. "HierChain: A Hierarchical-Blockchain-Based Data Management System for Smart Healthcare," in *IEEE Internet of Things Journal,* 11: 2924-2934, DOI: https://doi.org/10.1109/JIOT.2023.3295847