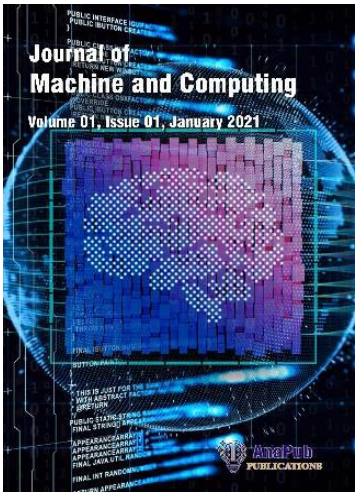
Journal Pre-proof

Efficient Event Transactions in VANET's Using Reinforcement Learning Aided Block Chain Architecture

Shaik Mulla Almas, Kavitha K and Kalavathi Alla



DOI: 10.53759/7669/jmc202505052

Reference: JMC202505052

Journal: Journal of Machine and Computing.

Received 22 June 2024

Revised form 15 August 2024

Accepted 04 October 2024

Please cite this article as: Shaik Mulla Almas, Kavitha K and Kalavathi Alla, "Efficient Event Transactions in VANET's Using Reinforcement Learning Aided Block Chain Architecture", Journal of Machine and Computing. (2025). Doi: https:// doi.org/10.53759/7669/jmc202505052

AnaPub
PUBLICATIONS

Efficient Event Transactions in VANET's Using Reinforcement Learning Aided Block Chain Architecture

Shaik Mulla Almas[1*], K. Kavitha[2], Kalavathi Alla[3]

[1*]Research Scholar, Department of Computer Science and Engineering, Annamalai University, India.
[2]Associate Professor, Department of Computer Science and Engineering, Annamalai University, India.
[3]Professor, Department of Information Technology, Vasireddy Venkatadri Institute of Technology, India.

[1*]*mullaalmas27@gmail.com* [2]kavithacseau@gmail.com [3]kalavathi_alla@yahoo.com

ABSTRACT

Vehicular Ad-Hoc Networks (VANETs) have emerged as a pivotal technology for enhancing road safety and traffic management through real-time vehicle-to-vehicle (V2V) communication. However, the dynamic and open nature of VANETs introduces challenges related to data security, privacy, and trust among vehicles. To address these challenges, the integration of blockchain technology into VANETs has gained considerable attention. In this study, we introduce Vehicular chain Reinforcement Learning (RL), a Blockchain-based VANET system that employs artificial intelligence (AI), Deep Reinforcement Learning (DRL), to create a flexible, knowledgeable, collaborative, and secure network for the VANET industry. The framework brings together a wide variety of VANET systems by utilizing Blockchain technology and an intelligent decision-making RL algorithm that operates online. The goal is to optimize the network's behavior in real time, with privacy and security of Vehicles data as primary concerns. The proposed Blockchain Manager (BM) intelligently adjusts blockchain setup to optimize security, latency, and cost. In the realm of Reinforcement Learning (RL), the DQN framework introduces Deep Q-Network (DQN), Double Deep Q-Network (DDQN) and Dueling DQN (DQDQN) techniques to efficiently solve the Markov Decision Process (MDP) optimization model. The proposed approaches and two heuristic ones are thoroughly compared. The suggested methods achieve real-time adaptation to system state convergence, maximum security, minimal latency, and low cost.

Keywords: VANET, Block chain, Block Manager, Reinforcement learning, Deep Q-Network (DQN)

1. INTRODUCTION

Focusing on the global impact of VANET systems and their effect on people's standard of living is essential. Due to an increase in the number of Vehicles with event transactions, it is becoming increasingly difficult to employ the conventional VANET model to provide round-the- clock monitoring. Direct interaction between doctors and Vehicles during illness epidemics raises concerns about instability, scalability, and delays in receiving critical services. As a result, both the Vehicles and the doctors face a higher risk of dying. More than 10 million Indian vehicles have accidental records that limit their ability to function according to the Traffic control laws. Obviously, these figures rise precipitously due to the developing intelligent transport management systems. It is of the utmost importance to establish an VANET system that eliminates the requirement for Vehicles and physicians to meet one another in person. Researchers are looking into approaches to decentralize the connecting of several parties while yet considering these constraints. In 2008, a distributed ledger, often known as a blockchain, was first presented as a means of ensuring the dependability and security of data that is exchanged across several participants. As discussed in, this exciting technology was use in a variety of fields, including but not limited to Industry 4.0 and the IoT, the financial sector, and the academic world.

Blockchain features allowed it “to overcome central challenges in these applications. Due to the characteristics that it possesses, Blockchain was able to overcome major problems in several applications. These characteristics can be summarized as follows: It eliminates the need for a third party while at the same time fostering confidence amongst diverse entities subject to a variety of rules and regulations**.** Data recovery is made simpler because all entities involved in the Blockchain have access to a copy of the ledger. In this way, the newly added block is irretrievable and better fraud detection is achieved [1].

To create these Blockchain systems, a consensus algorithm and smart contract are used. Blockchain's consistency and integrity are safeguarded by the consensus algorithm. There are a few different consensus algorithms that have been researched and written about, including Proof of Work (PoW), Proof of Stake (PoS), Practical Byzantine Fault Tolerance (PBFT), Delegated Proof of Stake (DPoS), and others. Without the need for a middleman, smart contracts enable the autonomous execution of business logic in response to predefined criteria. Smart contracts enjoy the same security assurances as the blockchain ledger since they are executed as transactions on top of the ledger. Miners are responsible for checking the legitimacy of transactions before they are included in a confirmed block. With guidance from Blockchain companies, miners can reliably enforce smart contract regulations. Due to Vehicles misidentification and event records being duplicated between vehicles, traditional VANET systems experience redundancy issues. The VANET industry has been an early adopter of blockchain technology due to the many ways in which it can be utilized to improve upon the inefficiencies of current VANET systems. The VANET sector is predicted to become the largest Blockchain market by 2022, with revenues of over $500 million.

VANET systems benefit from blockchain technology because of its ability to reduce the likelihood of inconsistencies in medical data, resulting in higher-quality data, shorter processing times, fewer human processing processes, and lower reconciliation costs. Moreover, Blockchain capabilities such as accessibility, trust, openness, traceability, and auditability can be effectively implemented in VANET delivery systems. When conducting an analysis of medical data, it can be helpful to link data and events from a variety of entities to understand the factors that contribute to medical phenomena like virus infections. When dealing with complicated transactions while adhering to all the essential privacy and security regulations, there are issues that can occur. When optimizing for Blockchain, the writers solely take latency and security into account. The price, however, is a factor that must not be disregarded.

To enhance blockchain efficiency, it is necessary to update the Blockchain configuration adaptively based on the characteristics of incoming transactions, a task that requires a learning- assisted decision-making strategy [2]. Rapid advancements in Artificial Intelligence (AI) in recent years attest to the technology's prowess in efficiently absorbing and applying lessons from large datasets. To enable the construction of intelligent health systems across a variety of disciplines, including the VANET industry, AI methodologies were used extensively. Data analysis, preprocessing, recognition, categorization, drug discovery, etc. were only some of the many uses. The Artificial Intelligence (AI) technique known as Machine Learning (ML) known as Reinforcement Learning (RL) has found usage in medical settings. RL is a technology that is decision-driven and learns the dynamics of its surroundings as well as the links between the states of its components. Since RL approaches include both the immediate (short-term) reward at a given state and the discovery of a long-term policy that optimizes the system's benefit over time, they have the potential to outperform conventional methods of decision-making.

Deep Learning was combined with traditional RL to create Deep Reinforcement Learning, or DRL for short, to improve RL's overall performance”. A decision can be made in real time by Deep Reinforcement Learning (DRL) based on a model that has been trained. This paradigm enables us to achieve our objective of maximizing system security while simultaneously reducing latency and costs, and achieving this optimal balance between these competing system goals is our primary objective.

Within the scope of this investigation, we present Health chain-RL, an effective and decentralized VANET Blockchain architecture. Health chain-RL makes use of Deep Reinforcement Learning (DRL), which enables the network's behavior to be dynamically modified. This paradigm enables us to achieve our objective of maximizing system security while simultaneously reducing latency and costs, and achieving this optimal balance between these competing system goals is our primary objective. Here is a rundown of the major contributions:

A multi-goal optimization framework, Blockchain-RL is being developed for use in VANET systems. "It establishes a relationship between characteristics like the number of transactions per block and the age of a transaction and blockchain setup aspects like the priority of transactions and the security of data. The purpose of Blockchain-RL is to boost the effectiveness of VANET networks such as to:

- Introduce the reputation of Blockchain miners; consider the temporal elements of Blockchain; and formulate the Markov Decision Process (MDP) of our suggested Health chain-RL [3].
- Optimise latency, security, and cost in real-time while considering the requirements of Blockchain entities and have been tasked with proposing an intelligent manager that is based on reinforcement learning techniques such as Deep Q-Network (DQN) and Dueling Double Deep Network. This will allow to optimize these factors by taking into account the requirements of Blockchain entities.
- Compare the suggested Health chain-RL to other methods, such as the Greedy and Random- selection methods, while demonstrating the superior performance of the proposed BM".

2. RELATED WORK

Table 1. Blockchain-powered applications employing Deep Reinforcement Learning.

AUTHORS	FIELD	TRADE-OFF OBJECTIVE	RL-APPROACH
Zhang, D., Zeng, Z., Sudhan, A.	Vehicular Ad Hoc Networks	Trust features of Blockchain nodes and vehicles, consensus nodes, Blockchain computational capability	DDQN
Abulkuda[illegible] K.	Industrial Internet of Things	Parity across regions and overall energy use	Distributed DQN
Li[illegible] Y., Wang, S., Zhao, Q., [illegible]u, S., Zhou, A., Ma, [illegible], Yang, F.	Vehicular Edge Computing	Energy required for transmission, data stored in cache, and delay in sending data all add up.	DQN
Lian[illegible] F., Yu, W., Liu, X., Griffith, D., Golmie, N.	Industrial Internet of Things	Flexibility, independence, delay, and safety	DQN
Xia, X., Chen, F., He, Q., Grundy, J., Abdelrazek, M., Jin, H.	Wireless Networks	Consumption of resources, costs, and caching	DQN
Guth, S., et al.	IoT Monitoring Applications	Accountability, lag time, and price	DQN

2.1 Blockchain Technology in VANET

Blockchain is ideally suited for use in VANET applications due to its features, which are required to uphold a high level of confidentiality when exchanging Vehicles data and medical records with one another. The authors propose a distributed event record ledger constructed on the MATLAB software. This will allow for diverse VANET operators to have access to Vehicles information in real time. Unfortunately, it has shortcomings in a variety of areas, including Vehicles identity, key replacement, and scalability, among others. The proposed architecture that is built on the Blockchain that safeguards the confidentiality of Vehicles event records and prohibits potentially harmful parties from having unauthorized access to those records. The proposed framework for Parallel Healthcare System (PHS) Blockchain has its own benefits such as scalability, latency, and security, because it is based on artificial systems, computational experimentation, and parallel execution, yet it has shortcomings. A dual Blockchain infrastructure is utilized by both the BSPP and the BLOCHIE VANET systems respectively. Both approaches come with several problems, including low scalability, high latency, high computational cost, and inadequate storage space. The private blockchain architecture for VANET known as Vehicular chain has problems with scalability and adds additional responsibilities, such as needing Vehicles to provide clearance [4]. The OmniPHR framework promotes interoperability among different providers to access health record, solves the scalability problem that Vehicular chain was having, although Vehicles authentication is still necessary. As an illustration, quite a few of the other suggested Blockchain systems in the VANET industry, such as, have problems with the scalability of their administrative processes.

2.2 Enhancing VANETs Using Reinforcement Learning

In "a Markov decision process (MDP), a transition to a new state is said to have occurred when a decision-maker (the agent) chooses an action for a given state while interacting with the environment (the formulation of the issue). This is because the decision-maker has moved on to a new state, Markov decision process (MDP). At the same time, the agent is rewarded monetarily for the work that he or she has done. As a result, the MDP consists of the following five basic components: the agent, the environment, the states, the actions, and the reward. A unique approach to solving Markov decision processes (MDPs) that use artificial intelligence (AI) is called reinforcement learning (RL), and it is a subfield of the area of machine learning (ML)". The major objective of the agent is to engage with its environment in a manner that contributes to the accomplishment of its other primary objective, which is to maximize its utility by adhering to a behavior policy. In the second stage, you will examine the policy that is the focus of your attention and decide on the most effective next move for a particular state. This later technique ends up being the one that is better in the long run, thus it is the one that we will implement.

2.3 Off-policy Learning

During its training, the agent may select either the on-policy or the off-policy instructional method. The concept known as on-policy learning describes a circumstance in which the desired policy and the actual behavior are completely congruent with one another. Off-policy learning is the term used to describe the alternative (e.g., Q-Learning). Q-learning is a well-known example of an off-policy learning algorithm in reinforcement learning. An agent is a piece of software that takes in information about a policy's value function and then tries to optimize that policy by analyzing it and making changes where necessary. Off-policy learning, on the other hand, involves the agent learning the value function in a manner that is distinct from the action itself. This is accomplished by iteratively updating the policy in the course of explorationin order to find the most effective policy [5].

2.4 Q-Learning approach

In particular, the Q-Learning approach is investigated in this work. This is a method in which an agent attempts to determine the most appropriate response for any given set of circumstances and then records this data in a Q-table. Medical imaging research and clinical concept extraction are only two examples of the kinds of challenges that neural networks may help with. This is only one example of how

doctors might benefit from using deep q-networks (DQN).

3. SYSTEM MODELING AND ANALYSIS

Our goal in creating Vehicular chain-RL was to create a safe, adaptable, and web-based platform where many parties could safely share and use VANET information. Fig-1 depicts the structure, which advocates the implementation of a consortium medical Blockchain across multiple VANET organizations. In accordance with their predetermined eligibility in the smart contract, these entities will have access to the distributed ledger where the medical data is stored, share it with other entities, and process it. In addition, any organization may operate its own private network to gather, process, and prepare Blockchain-bound transactions holding crucial data. Transaction data might be gathered, processed, and processed using this network. Some local network data may be preprocessed using AI techniques including summarization, clustering, and compression. However, this article will primarily focus on optimizing Blockchain networks by striking a balance between security, latency, and cost in light of the constraints imposed by transactions, specificallythe security and urgency levels.

Blockchain managers have been proposed as a means of dealing with the time-sensitive nature of medical data, protecting that data from unauthorized access, and optimizing all the aims at once [6]. In this study, we present a smart Blockchain manager that utilizes reinforcement learning methods to respond to the ever-changing state of the system and anticipate its behavior in the future.

Then, the Blockchain optimization problem is resolved, along with the Blockchain entities, the Blockchain network, the intelligent Blockchain manager, and the Blockchain itself.

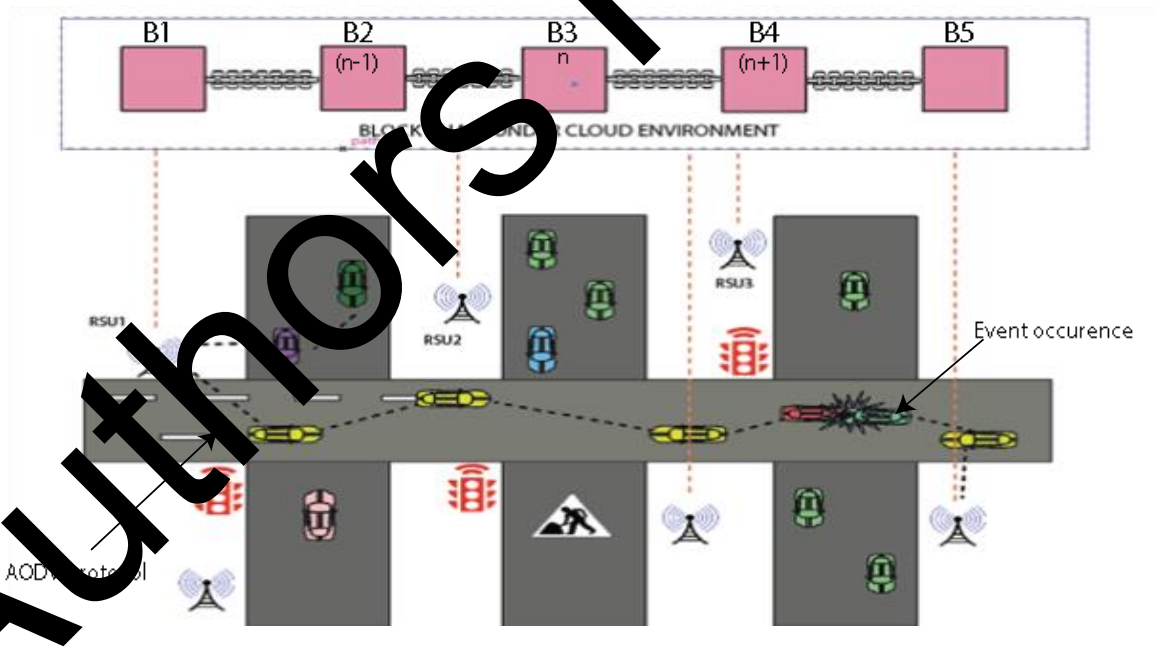


Fig. 1 Architecture of Vehicular Chain-Reinforcement Learning

3.1 Entities of a Blockchain

Several interested parties may collaborate on this framework's creation to speed up the creation of a decentralized VANET system that is also scalable, safe, and smart. To conduct research, review data, and adopt new health rules, these organizations can either share their VANET data with the blockchain or access the data that is already there. Possible participants in such a framework include medical facilities, pharmaceutical stores, insurance providers, and the Ministry of Public Health (MOPH**) [7].** In the form of a smart contract, Blockchain presents the underlying business logic that makes the technology work. This logic encompasses all the organizations' stipulations, guidelines, hierarchies of authority, and order of importance levels. Each party must approve the transaction and then apply the smart contract before it can be recordedon the Blockchain ledger.

3.2 Blockchain Technology.

Blockchain, a distributed ledger technology, facilitates the secure movement, storage, and processing of Vehicles data between institutions. Instead of using Proof of Stake (PoS), a consensus mechanism called Delegated Proof of Stake (DPoS) can be used to guarantee scalability and shield the Blockchain from detrimental usage and centralization. Voting and elections are used to select miners who maintain low operating expenses. With the Blockchain setup described in this paper, the trade-off between cost, latency, and security may be adjusted to suit the needs of the various entities storing transactions [8]. The important variables are the total number of miners and the average number of transactions per block. Thosetwo factors are determined by the smart Blockchain manager.

3.3 Blockchain Manager of Intelligent Blockchain — an Optimizer

One of the most important parts of the proposed architecture is the Blockchain Manager (BM). A certain time step's worth of transactions will be gathered, and from there the number of transactions allowed in a block will be determined. Next, a set number of miners validate a block using their storage space, processing power, and transaction fees. While it may be tempting to keep adjusting those settings, doing so can incur unnecessary computational and financial costs and should be avoided. Unless an unexpected occurrence occurs, all parties involved can agree on how often the Blockchain configuration should be updated. A smart contract on the blockchain or a timed algorithm could specify certain actions to be done at specific intervals. To ensure the integrity of the Blockchain, either one entity must assume the role of BM (which is not recommended for security reasons) or the role can be shared among multiple entities in the same chain and rotated at regular intervals. This circulation should occur on a predetermined and agreed-upon schedule. For the sake of consistency and safety, the proposed Vehicular chain-RL framework implements a circulation protocol.

Our goal is to find the sweet spot between security, latency, and cost in the Vehicular chain-RL protocol by optimizing the provided attributes transaction latency, cost, and security. Information on a transaction's security, timeliness, and age are summarized using various data compression, classification, and event detection methods (local network). How long a transaction must sit in limbo before it can be put to the Blockchain is described by the "age" concept (ledger). In Section 4 we go into greater depth. Here are some examples where urgency and safety play a role: Urgent transactions, like emergency alerts, may call for little security and short latency. If more miners are needed to keep the Bitcoin network running, transaction fees and transaction times for high- security payments may rise.

Considering three competing goals, the suggested framework allows us to attempt to translate the Vehicle's circumstances into several modes of Blockchain configuration. Safety, Delay, and Money [9]. At a given time step t, the utility multi-objective function is represented by Eq. (1)

$$min_{m_i}, \quad p\left(\frac{L}{l_{max}}\right) + q\left(\frac{s_{max}}{S}\right) + \left(\frac{C}{c_{max}}\right) \tag{1}$$

$$subject\ to \quad 1 \le m_i \le M_{max}$$

$$1 \le tr_i \le T_{max}$$

$$A_t \le A_{th}$$

Where by $1 \le m_i \le M_{max}$ and $1 \le tr_i \le T_{max}$ are constraints on the chosen number of miners m_i and the number of transactions tr_i, respectively.

The M_{max} miners and T_{max} transactions in a single block. The transaction age (A_t) should not exceed the threshold (A_{th}) above which a transaction is rejected from the transmission queue and not included in a block and forwarded to the network. According to the needs of the system administrator, the relative importance of latency, security, and cost is determined by the weighting factors p, q, and r, the sum of which equals one. When determining what features a system administrator needs, it is possible to consider both business logic and the data's inherent characteristics. With the maximum values of the objectives in mind, we were able to create equations that were uniform in their units and scaled to the same dimensions. Maximum latency (l_{max}), maximum security (s_{max}) and maximum cost (c_{max}) [10].

$$S = S_c m B M^q \tag{2}$$

Where S_c is a system coefficient; mBM is the number of miners picked by the BM; and q is an indicator factor demonstrating the scale of the network with a value that is greater than or equal to two, can be used to identify the security (S).

The total amount of time that it takes to create, verify, broadcast, and upload a block is denoted by the symbol L in Equation 3.

$$L = \left(\frac{t_b S_t}{D_{tr}}\right) + max_i\left(\frac{G}{a_i}\right) + E t_b S_t m + \frac{V_f}{U_{tr}} \tag{3}$$

t_b is the number of transactions that are included in each block, S_t is the size of each transaction, G is the number of computational resources required to verify a block, a_i is the number of computational resources that miner I possesses and E is a predefined parameter that is described in greater detail in. The size of the verification feedback is denoted by V_f, the uplink transmission rate from the miners to the BM is denoted by U_{tr} and the downlink transmission rate from the BM to the miners is denoted by D_{tr}.

$$U_{tr} = blog(1 + SN_u) \tag{4}$$

$$D_{tr} = blog(1 + SN_d) \tag{5}$$

In equations (4) and (5), b denotes the bandwidth, while SN_d and SN_u stand for the signal-to-noise ratio of the downlink and the uplink respectively.

The final objective that we are aiming to achieve via reducing costs is the cost C_{min}, Which is represented in Equation (6).

$$C_{min} = \frac{\sum_0^m CC_i}{t_b} \tag{6}$$

In this equation, the computing cost for each miner m is denoted by CC_i, where I am the number of selected transactions. As $CCi = a_i \times r_i$, CC_i is proportional to the product of its available resources (a_i) and the cost of utilizing those resources (r_i).

3.4 Strategy Based on Deep Reinforcement Learning (RL)

The architecture shown in was used as a starting point and tweaked such that the system could be used in a variety of online environments. This study aims to find the optimal Blockchain configuration for a given set of Vehicles conditions by considering the tension between the three main constraints of any VANET system: privacy, speed, and cost. Its goal is to ensure a responsive, smart, and safe VANET system. We classify this optimization issue as NP-hard. The problem was solved by the authors in using a Greedy strategy that ignored the time-dependent nature of receiving transactions within the Blockchain framework, hence negatively affecting latency and the system's long-term viability. The Greedy method is expensive and unreliable in real-time since it requires solving the optimization at each time step. To get beyond these restrictions on speed, complexity, and future aggregated performance, researchers have turned to reinforcement learning methodologies, particularly Q learning and its derivatives. The configurations of Blockchain will be determined at regular intervals by examining the BM queue of pending transactions [11]. The utility function represents the trade-off in a consortium, considering security, speed, and efficiency. A Markov Decision Process model is used to explain the multi-objective optimization issue (MDP) State space (SS), action space (AS), state transitions (Ts), reward function (RF), and discount factor (D_f) $D_f \in [0, 1)$. The agent receives a snapshot of the environment's state, represented as es ∈ SS, at regular intervals of time, denoted by t. To get a reward $rt \in$ R, the agent must carry out an action ea ∈ AS in accordance with a policy (a|s). Therefore, P (s' |es, ea) = Ts (es, ea, s') is the transition probability from state es to state s ′ when ea is the initial state. We introduce the concept of deep expectation, where the reward for performing an action a in a state s while adhering to a certain policy π is encapsulated by the function $Q\pi$ (es, ea) = $E_{\pi,P}$[rt |st = es, at = ea]. The MDP is then solved using Deep Reinforcement Learning (DRL). After the MDP is defined, Deep Reinforcement Learning is used to solve it (DRL). Following this, we will describe in detail the structure of our aims-based approach to education. This encompasses the SS, AS, Ts, and MORF of our optimization issue, or state space, action space, environment, and goal function.

4. PROPOSED METHODOLOGIES FOR REINFORCEMENT LEARNING

Blockchain Manager is unaware of the model for state transitions, a model-free technique like Deep Q-Network (DQN) must be employed to arrive at an estimate for Q. There is no assurance that this approach will converge, thus it's useful to add enhancements that speed up the weight convergence and stabilize training in a neural network. When it comes to model-free algorithms, experience replay (Rp) is a key idea utilized to facilitate training, both for the present experience and the agent's accumulated history of experiences. Using these previously collected samples of experience during updates not only improves data efficiency, "but also guarantees that the correlation between different samples of experience in the update is minimized using uniform sampling, which in turn reduces the variance. Using experience replay and soft updates to the neural networks, Fig. 2 depicts the interaction flowchart between an agent (BM) and the environment (environment in Bitcoin. The agent takes suitable action to reflect the share of state- selected transactions and miner needed to validate them. The environment's evaluation of the agent's action and its subsequent impact on reward and state will be fed back to the agent. Initially, multiple random actions are taken to explore the state space and determine the ideal action for that state. An experience replay memory will be created to record all information regarding previous encounters, including the state, the action made, the reward received, and the subsequent state [12]. As often as N time steps, the target network's settings will be adjusted. Making the proper choice and responding quickly to unexpected changes are crucial in VANET applications, where they can have a major impact on the health of the Vehicles and the effectiveness of the system. Traditional methods of decision-making struggle to keep up with the rapid pace and high stakes of today's VANET systems. Therefore, the purpose of this research is to explore the online decision-making performance and flexibility of state-of-the-art off-policy techniques. Double Q- Network (DQN) and its

variants are investigated. These include the Double Deep Q-Network (DDQN) and the Dueling Double Q-Network (D3QN). When neural networks are used in place of a Q-table, as they are in the original DQN model, the table is no longer needed. Concerns of overconfidence and false positives associated with DQN are especially pressing in the context of VANET systems. Since DQN suffers from being too optimistic, DDQN is an improvement. D3QN considers not just the value of the state in relation to the action to be taken, but also the probability of being in that condition. Thus, D3QN employs a neural network model with a unique structure. Each method is broken down into its component parts below. Traditional Q- learning relies on Q- tables for action estimation, but for model-free results, researchers at Google have recommended using a neural network instead. Machine learning models are used by DQN as function approximators instead of traditional lookup tables. The online network will undergo continuous gradient descent updates, while the parameters of the target network are changed after a predetermined number of episodes.

In the case of DQN, Y is denoted by the equation (7), and the parameters are held constant at some previously determined values *.

$$Y_k^{DQN} = r + Df_{max}Q(s'.a';\theta*) \quad (7)$$

The revised version of the loss function, denoted by Lf, considers the historical data on Y, DQN that has been stored in the replay buffer, denotedby Rp (8). Equation (9) describes the Q function,

$$Lfk(\theta k) = E_{(s,s,r,s}F)\sim P(R\epsilon)[(Yk^{DQN} - Q(s, \theta k))^2] \quad (8)$$

$$Qk+1(s_t, a_t, \theta_t) = (s_t, \theta_t) + Yk^{DQN} - (s_t, a_t, \theta_t)) \quad (9)$$

Where k is the episode duration, D_f is the discounting factor that prevents the BM from relying solely on future rewards, and the settings of the neural network. Rectified linear units (ReLU) make use of the activation function (∂), a positive learning function defined by ∂ = max (0, x). The φ value denotes the learning rate, which indicates the degree to which the most recent estimate is modified in relation to the update target. The smooth-ef1 loss function is developed and refined with the aid of Adam Optimizer. Online Sample-based learning is made possible using an intermediate estimate Y of the rewards of the future state. Adam is utilized as an alternative to standard stochastic gradient descent (SGD) techniques. The Q-Learning approach that is utilized determines how Y is defined.

4.1 Double Deep Q-Network (DDQN)

In DDQN, the maximum operator is used by both the online and target networks at the same time, but in DDQN, the maximum operator is used in a separate fashion. The two methods differ in how they alter the target network [13]. Because of this, the estimated actions are unrealistically positive. Most of the time, an overly optimistic problem will manifest itself in the form of a false-positive issue when it comes to large- scale issues. To choose the appropriate action with the highest Q-value, DDQN favors using online networks. With an eye toward the following state's expected Q-values, the target network prioritizes the action that necessitates the most data. After a set number of cycles, the online network's data will be used to adjust the parameters of the desired network. On the other hand, the internet network will be upgraded in accordance with the optimizer (e.g., Adam Optimizer). Because of this, the problem of over optimism will be mitigated, and the phase of learning will become steadier and more dependable. Equation (10) stands for the Y_k^{DDQN}, and Equation (11) is the action- value function Q for the DDQN taking Y DDQN into consideration.

$$Y_k^{DDQN} = r + D(s', arg_{max}(Q(s'.a';\theta^*))_k \quad (10)$$

$$Q_{k+1}(s_t, a_t, \theta_t) = (s_t, a_t, \theta_t) + \varphi\left(Y_k^{DDQN} - Q_{k+1}(s_t, a_t, \theta_t)\right) \quad (11)$$

4.2 Dual-Depth Q-Network Battle (D3QN)

We presented a new network architecture called the Dueling Double Deep Q-Network (D3QN). Since many states' action choices are roughly equivalent, the motivation for suggesting D3QN is that doing so may slow down learning. Two streams can be estimated using the suggested dueling neural network model. In this study, we apply the idea of an advantage function to DDQN, where the approach of applying advantage function A produces identifiability concerns and consequently inhibits the recovery of both the V_s and A_f. T the average of Af, shown to increase the stability of the optimization. Action-state Q function in D3QN is represented by Eq. (12), which incorporates the value and advantage functions.

$$Q(s, a;\ \theta, x", y") = V_s(s, \theta, x") + ((A_f(s, a;\ \theta, x", y") - \frac{1}{|A_f|}\sum_a^{A_f} A_f(\ldots a;\ \theta, x\ldots) \quad (12)$$

Parameters for the combined streams A_f and V_s are denoted by x" and y" in Eq. (12).

$V_s(s)$ = E[Q^* $(s,\ a)$]is a representation of the state-value function V_s. It is important to note that over optimism is not an issue for many uses because high performance can still be achieved. But lowering it will greatly steady the educational process.

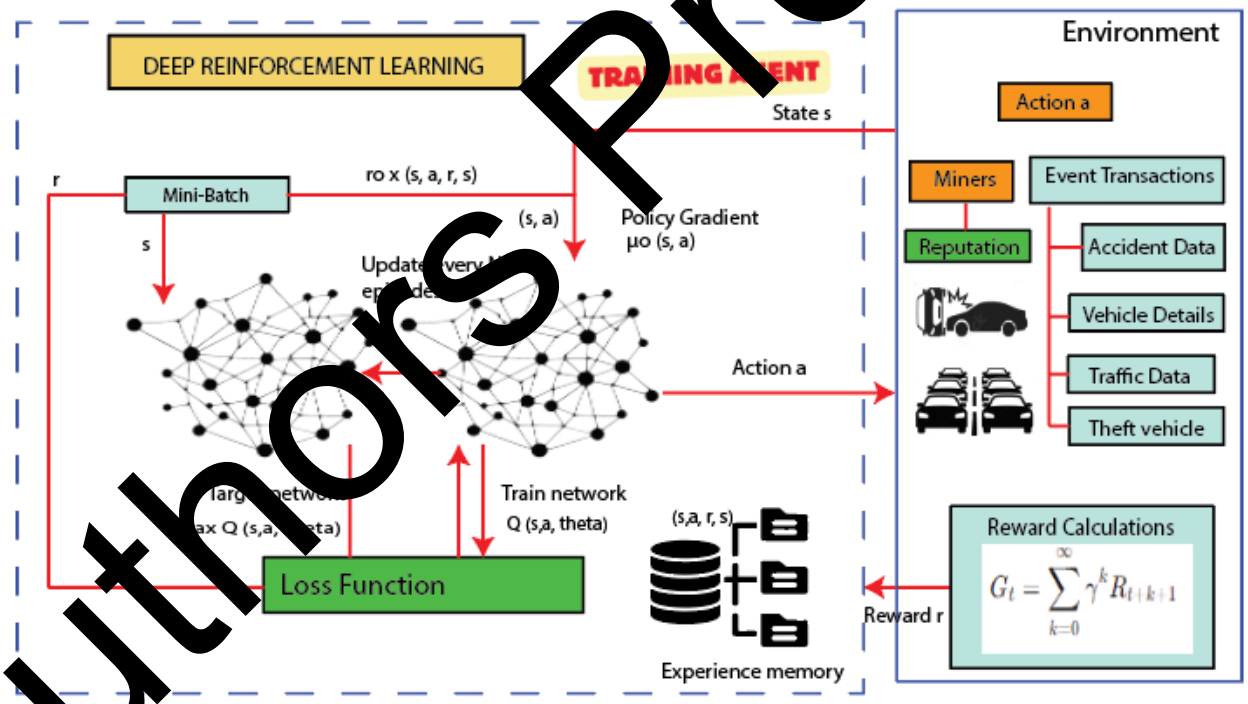


Fig. 2. Vehicular chain-RL system's Deep Q-Learning flowchart

5. IDEAS PROPOSED DURING INSTRUCTION

First, we have Algorithm 1, which depicts the BM's training procedures in full. Together with the initialization of the replay memory Rp, where the tuples of experience data are stored, the weights of the online and target neural networks are also set. Through social interaction, an agent in the BM model accumulates a set of experience tuples across a succession of states. The agent "picks actions in each

state at random with probability or according to a greedy policy (ϵ) to ensure the quality of its investigation of the actions available in that state. It also involves adjusting the weights of both the online and target neural networks until line7. To facilitate BM environment learning, ϵ is initially set to 1, and then begins to decline over time [14]. Exploitation behavior is represented by the best actions (those with the highest Q-value), while exploration behavior is shown by random activities. Acting is done to control the cost-benefit ratio of the situation. The tuples of state-action transitions are recorded in the replay memory Rp and later used as experience data in the optimization process to refine the estimation of Q. As an option, we can consider requesting irrational subsets of Rp's experiences of ρ. The TD-target Y is computed for each experience tuple i in the subset ρt to arrive at the updated estimate when θ^* is considered. This procedure aids in stabilizing and bringing about convergence in the learning process (experience replay). Whether a DQN, DDQN, or D3QN model is employed, the resulting TD-target Yi is determined by the formula. Adam optimizer is then used to fit to $Y\ i$ with a soft update of θ^* applied after a predetermined number of iterations to account for the most recent information about the environment. Soft updates to the target network can also help stabilize the learning process Convergence occurs when and only when *$\theta \sim \theta$*".

Algorithm 1: Methods Used in BM Training (agent)

Input: Artificial Environment Modeler

Output: θ1: The approximation's NN parameters $Q*$

1: $R_p \leftarrow$ Setting the amount of the replay memory as the initial N.

2: $\theta \leftarrow$ Randomize internet network settings to start.

3: $\theta^* \leftarrow \theta$ Setup the parameters for the intended network.

4: for $episodes = 1 : E$ do

5: To set the initial condition $s_0 \leftarrow \langle [S_0, U_0, a_0], [R_1 \cdots R_M] \rangle$

6: for $t = 1 : K$ do /** Environmental interaction **/

7: Specify the Update Operation for the State a_t

$$\begin{cases} \text{Random with probability } \epsilon \\ \text{greedy policy,} \quad \text{otherwise} \end{cases}$$

Determine, using at, how many transactions tr and how many miners m were chosen.

Use Eqs. (7) and (8) as guides.

8: Perform at and evaluate s_{t+1} and r_t

Rewards can be calculated using Eqs. (9) and (10). trt and mt from a_t

9: Achieve Environmental Status ← Boolean

10: Replace Experience with a New Tuple in the Playback Memory

Rp ← (s_t , a_t , r_t , s_{t+1}, $done$)

11.Update State $s_t \leftarrow s_{t+1}$

12: /**Updating the estimates**/ Just pick a sample at random $\rho \subseteq R_p$

$\rho \leftarrow$ { s_t , a_t , r_i , s_{t+1},done} $|\rho|^1$

13: Determine Q-targets for the selected strategy by using the target network.

For DQN, DDQN,and D3QN based equations, see Eqs. (15), (18), and (20).

14: Fit θ to Target Y_t using Adam Optimizer

15: Update the target network at each target step. $\theta^* \leftarrow r\theta + (1 - r)$*16:

end for

17: end for

18: return $\theta_1 \sim \theta$

5.1. Methods Proposed in Real Time

The neural network parameters were saved after completing algorithm 1 and reaching convergence, allowing their subsequent use in a real-time setting. This means that the agent must be prepared to adapt to sudden changes in the environment (such as an increase in miner pricing or the loss of some miners) to maintain the convergence state. One run through the neural network is performed at each time step in Algorithm 2, which is a representation of the steps as they occur in real time. The BM will always take the course of action that maximizes its expectedprofit over the long run.

Algorithm 2: Directly Monitored BM in Real Time (agent)

Input: NN parameters after training $\theta 1$

Output: State update

1: $done \leftarrow$ Set the initial environment status to false.

2: Initialize State st

3: while! done do

4: Find a * Consider that $a^* \leftarrow$ * and m^*

5: Assuming that you know the reward Q^* (, at , θ)

6: Observe s_{t+1}, and r_t

7: Environment State done ← Boolean

8: To the Replay Memory, Insert the Experience of a Tuple. $R_p \leftarrow$ (, a_t , r_t , s_{t+1}, $done$)

9: Update State $s_t \leftarrow s_{t+1}$

10: end while

6.EVALUATION OF PERFORMANCE

Vehicular chain-RL is tested through computational simulations to see how well it can adapt to novel conditions, how much action-time it requires to achieve a desired result, and how well its proposed strategies converge on the reward function.

6.1 Experiment Setup & Design Procedure

We utilize a BM queue size of Q and a multiplicative factor (per episode) for epsilon decay of 0.999 to train our Vehicular chain-RL over 104 episodes (E). It's vital to keep in mind that as the number

of miners grows, so will the risks, delays, and expenses associated with using the network. However, the delay and overall cost will rise as the number of transactions rises. Since consistency is of utmost importance, we've decided to give equal importance to all three goals (p, q, r). Table 2 details the weighting factor values and other optimization parameters in terms of safety, delay, and expense.

Table 2. Variables used in optimization.

Variables	Values
B	0.52MHz
SNu	12dB
SNd	10dB
Q	2.01
Vf	0.52Mb
St	1.0Kb
Sc	1.0
p,q,r	0.342

6.2 Convergence in Rewards

The hyper parameters of the four Q-Learning methods are adjusted in Algorithm 1's training procedure using the values given in Table 3. Experiments and observations were used to determine these parameters, which were optimized for performance. In Figure 2, we see the training reward for a total of 104 episodes over four different neural network depths (plain vanilla DQN, DQN with ten hidden layers, DDQN, and D3QN). Rewards values displayed here are averaged over the past 50 episodes. Using the graphic, we can see that the proposed methods do, in fact, converge. For the first 1000 episodes, all methods operate randomly with ϵ =1, allowing the agent to learn about its environment through trial and error. Subsequently, the agent gradually refines its random policy until it converges on the optimal version, using the exponential decay ϵ of described above [15]

Table 3. Configuration Settings

Variables	Values
α_i	20.01
α_j	10.2
E	10^4
	10^3
D_f	0.92
ϵ	1, by means of 0.9999 Decay
hl	4.0 otherwise 10.0
Q-Network	When hl= 4 → 70,45,45,79,150
Neurons/layers	When hl=10→70,45,45,45,45,45,45,45,45,45,79,150
φ	3X 10^{-4}
$\|\rho\|$	45
$\|R_p\|$	10^5
r	10^{-3}
target steps update (soft)	4

7. RESULTS AND DISCUSSION

To gauge the usefulness of Vehicular chain-policy RL, we have done a battery of experiments. Q-Learning methods have values that are close to one another, unlike the Greedy and RS policies. Vanilla DQN, by leveraging online and targeted networks, was able to reap the highest benefit (26,900). It is shown in below graph, how much each policy earned in rewards during the testing (real-time) episodes. We have a closer look at the payoff when latency, security, and cost are all considered. No goal may be pursued with more importance than the others. Considering this, any strategy's aggregate of these three competing goals needs to be less than or equal to one. The suggested multi-objective optimization paradigm involves a trade-off, while the reward function tends to maximize. It's important to tighten up security without compromising on speed or budget. Therefore, the incentive attempts to optimize the three competing goals while considering the limits imposed by the application. By assigning penalties to ensure that the requirements are met, the DQL algorithm and its variant find the policy that maximizes the long- term trade-off described by Eq. (10). To maximize safety, DQL methods consider not only the required level of security but also the time-sensitive nature of the transactions at hand. Therefore, it has the lowest latency when compared to the Greedy and RS methods. As was previously introduced, the reward function is directly affected by the reputation of the chosen miners. Since the RS strategy takes the best action given the current state, it achieves a high latency level, while the greedy approach, which simply considers the immediate future, is unable to manage the trade-off between security and latency. Greedy and RS methods have a hefty price tag compared to how well they perform. Increasing the security results in a higher cost objective, as the cost objective stated by Eq. (8) is directly related to the miners' cost. The delay goal is also strongly impacted by the volume of transactions that are chosen [16]. Therefore, the strategy should consider such details and maximize security while minimizing latency and expense. Figure 6 depicts the average accumulated rewards for all methods, demonstrating how similarly they function across the various Q-learning methods. As a matter of fact, the RS strategy has the worst performance, with the Greedy approach coming in second. In a stationary setting, Vanilla DQN performed best, with an average accumulated reward of 13450.

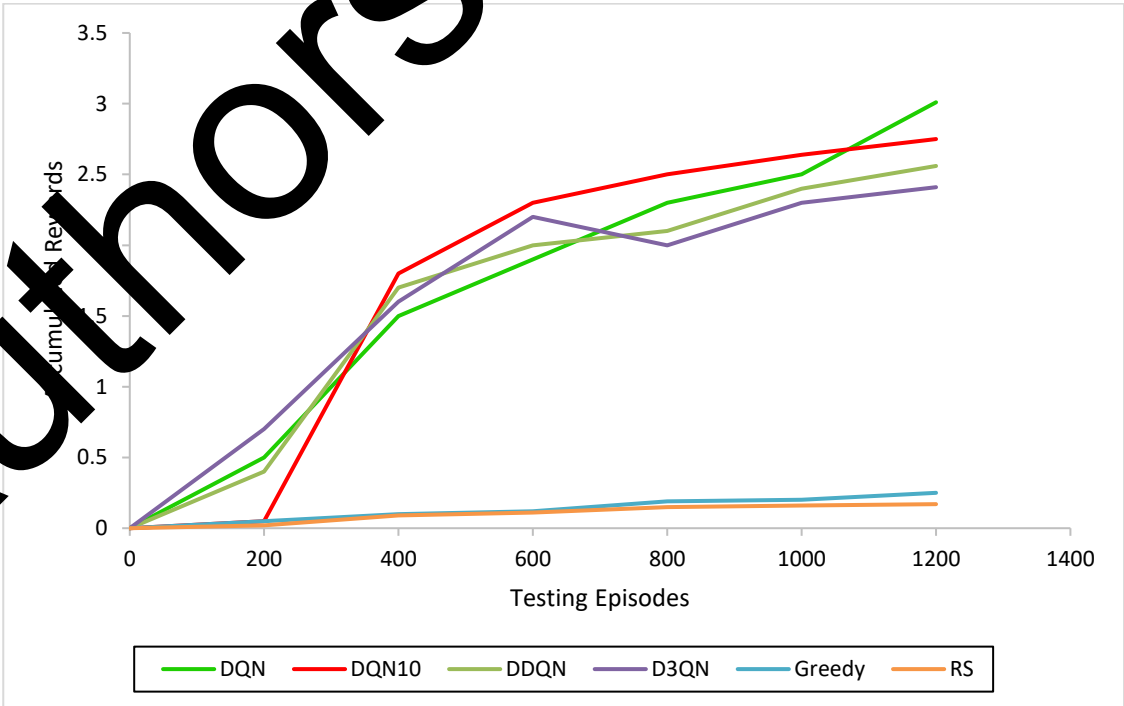
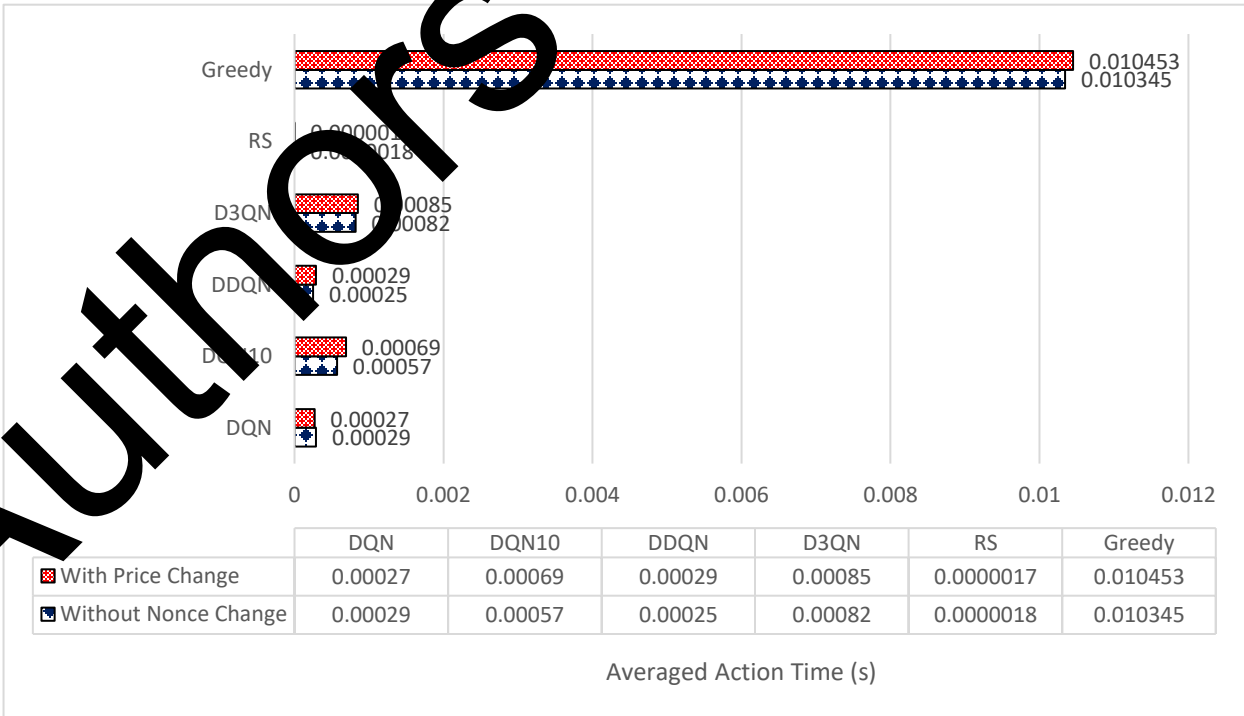


Fig. 3. Sum of all policies' rewards from their testing (in-the-wild) episodes.

7.1 Action-time

Time-to-decision is viewed as crucial in VANET applications, especially in emergency situations. The time it takes for each technique to decide on an action after a series of trials is as follows: The Greedy strategy uses the most time because it simply does one pass over all the available actions to identify the optimal one considering the immediate payoff. In VANET systems, when quick decision-making is essential, the Greedy method may be judged untrustworthy. The Random Selection (RS) method, on the other hand, is lightning quick. It picks an action at random, without considering any of the potential benefits, later state transmission, entity-level requirements, or necessary tries to successfully solve the environment, resulting in little accumulated reward over time. Taking the proper action with a focus on the future and the reward now is crucial in VANET systems, making the RS approach an undesirable choice. Our proposed training techniques perform similarly to one another, especially when there are no unexpected shifts in the environment. Proposed methods must account for the unexpected price shift while still rewarding miners for their future work. While vanilla D3QN may prove to be the fastest way to perform major system changes, doing so is more labor-intensive in practice. The vanilla DQN technique may be the most suitable option in a static system or while implementing labor-intensive changes. As we saw however, the odds of making a mistake are larger in DQN. Therefore, the characteristics of the system are relevant in determining the gravity of the problem. Considering their individual strengths and weaknesses as well as the system requirements of Vehicular chain-RL, DQN, DDQN, and D3QN can all be successfully implemented [17]. If the methodologies presented in this work are to be used in Vehicular chain-RL, the system administrator can use the following guidelines to make an informed decision: Even while D3QN increases the time it takes to perform an operation; it is the preferred option when working with a dynamic and unpredictable environment. When compared to alternative methods, DQN's time to choose an action is the fastest. However, when there is an abrupt shift in the system, the temptation to act inappropriately increases.



	DQN	DQN10	DDQN	D3QN	RS	Greedy
With Price Change	0.00027	0.00069	0.00029	0.00085	0.0000017	0.010453
Without Nonce Change	0.00029	0.00057	0.00025	0.00082	0.0000018	0.010345

Fig. 4. Simulated Policy Response Time in s, both with and without a shift in miners' pricing.

8. CONCLUSION

We provide Vehicular chain-RL, which enables the safe, adaptive, and flexible exchange of medical data and Vehicles information among a wide range of entities. An "intelligent Blockchain Manager (BM) is introduced to address the trade-off between system security, latency, and cost. The Blockchain Manager is implemented with one of three reinforcement decision-making algorithms (DQN, DDQN, or D3QN) depending on the needs of the VANET application and the robustness of the platform. For the time being, just one method may be used to put the smart Blockchain manager into action. The DQN method is useful when only little changes are made to the system, but the D3QN method can handle frequent fluctuations and converge smoothly in real time, as shown by the experiments. In the proposed Reinforcement Learning (RL) methods, the Block Manager (BM) takes on the role of agent determining parameters like transaction volume per block size and the required number of miners for validation. The simulation findings show that reinforcement learning approaches are superior to greedy and random selection (RS) methods. Despite heuristic approaches, which make a rapid decision-making that results in an immediate drop in the accumulated reward, the given models consider the temporal characteristics and future behavior of the Blockchain to arrive at a sub-optimal solution. When compared to DQN, DDQN, and D3QN, the Greedy approach's processing overhead makes it unsuitable for usage in real-time or mission-critical circumstances".

REFERENCES

[1] A. Hasselgren, K. Kralevska, D. Gligoroski, S.A. Pedersen, A. Faxvaag, Blockchain in healthcare and health sciesnces—A scoping review, Intl. J. Med. Inform. 134 (2020) 104040.

[2] M. Liu, Y. Teng, F.R. Yu, V.C.M. Leung, M. Song, Deep reinforcement learning based performance optimization in blockchain-enabled internet of vehicle, in: ICC 2019 - 2019 IEEE International Conference on Communications, ICC 2019, pp. 1–6.

[3] M. Liu, F.R. Yu, Y. Teng, V.C. Leung, M. Song, Performance optimization for blockchain-enabled industrial Internet of Things (IIoT) systems: A deep reinforcement learning approach, IEEE Trans. Ind. Inf. 15 (6) (2019) 3559–3570.

[4] S. Tanwar, K. Parekh, R. Evans, Blockchain-based electronic healthcare record system for healthcare 4.0applications, J. Inform. Secur. Appl. 50 (2020) 102407.

[5] F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Y. Wang, Q. Dong, H. Shen, Y. Wang, Artificial intelligence in healthcare: past, present and future, Stroke Vascular Neurol. 2 (4) (2017) 230–243.

[6] Y. Li, X. Liang, Z. Hu, E.P. Xing, Hybrid retrieval-generation reinforced agent for medical image report generation, in: Advances in Neural Information Processing Systems, 2018, pp. 1530–1540.

[7] Y. Ling, S.A. Hasan, V. Datla, A. Qadir, K. Lee, J. Liu, O. Farri, Diagnostic inferencing via improving clinical concept extraction with deep reinforcement learning: A preliminary study, in: Machine Learning forHealthcare Conference, 2017, pp. 271–285.

[8] S.M. Shortreed, E. Laber, D.J. Lizotte, T.S. Stroup, J. Pineau, S.A. Murphy, Informing sequential clinical decision-making through reinforcement learning: an empirical study, Mach. Learn. 84 (1–2) (2011) 109–136.

[9] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT press, 2018.

[10] H. Fan, L. Zhu, C. Yao, J. Guo, X. Lu, Deep reinforcement learning for energy efficiency optimization in wireless networks, in: 2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis, ICCCBDA, 2019, pp. 465–471.

[11] H. Lee, J. Kim, J. Lee, Resource allocation in wireless networks with deep reinforcement learning: A circumstance-independent approach, IEEE Syst. J. 14 (2) (2020) 2589–2592.

[12] D. Zhang, F.R. Yu, R. Yang, Blockchain-based distributed software-defined vehicular networks: A dueling deep Q-learning approach, IEEE Trans. Cogn. Commun. Netw. 5 (4) (2019) 1086–1100.

[13] Y. Dai, D. Xu, K. Zhang, S. Maharjan, Y. Zhang, Deep reinforcement learning and permissioned blockchain for content caching in vehicular edge computing and networks, IEEE Trans. Veh. Technol. 69(4) (2020) 4312–4324.

[14] Al Belushi, Y. Y. O., Dennis, P. J., Deepa, S., Arulkumar, V., Kanchana, D., & Ragini, Y. P. (2024, February). A Robust Development of an Efficient Industrial Monitoring and Fault Identification Model using Internet of Things. In 2024 IEEE International Conference on Big Data & Machine Learning (ICBDML) (pp. 27-32). IEEE.

[15] Y. Dai, D. Xu, S. Maharjan, Z. Chen, Q. He, Y. Zhang, Blockchain and deep reinforcement learning empowered intelligent 5G beyond, IEEE Netw. 33 (3) (2019) 10–17.

[16] N. Mhaisen, N. Fetais, A. Erbad, A. Mohamed, M. Guizani, To chain or not to chain: A reinforcement learning approach for blockchain-enabled IoT monitoring applications, Future Gener. Comput. Syst. 111 (2020) 39–51.

[17] T.T. Anh, N.C. Luong, Z. Xiong, D. Niyato, D.I. Kim, Joint time scheduling and transaction fee selection in blockchain-based RF-powered backscatter cognitive radio network, 2020, ArXiv preprint arXiv:2001.03336.