Network analysis in a peer-to-peer energy trading model using blockchain and machine learning

Saurabh Shukla [a,*], Shahid Hussain [b], Reyazur Rashid Irshad [c], Ahmed Abdu Alattab [c], Subhasis Thakur [d], John G. Breslin [d], M Fadzil Hassan [e], Satheesh Abimannan [f], Shahid Husain [g], Syed Muslim Jameel [h]

[a] Department of Computer Science, Indian Institute of Information Technology Lucknow (IIITL), Lucknow, UP, India
[b] Innovative Value Institute, School of Business, National University of Ireland Maynooth, W23 F2H6, Maynooth, Co. Kildare, Ireland
[c] Department of Computer Science, College of Science and Arts, Sharurah 68341, Najran University, Najran, Kingdom of Saudi Arabia
[d] Data Science Institute, University of Galway, H91 TK33, Galway, Ireland
[e] Department of Computer and Information Sciences, Universiti Teknologi PETRONAS (UTP), Seri Iskandar, Malaysia
[f] Amity School of Engineering and Technology (ASET), Amity University Maharashtra, Mumbai, India
[g] CAS-Ibri University of Technology & Applied Sciences, Oman
[h] National College of Ireland, Dublin, Ireland

A R T I C L E I N F O

Keywords:
Peer-to-peer energy trade
Blockchain
Smart grids
Smart meters
Cyber security
Energy
Cloud computing
Machine learning
Latency
Reinforcement learning
Neural networks
Q-learning

A B S T R A C T

Existing technology like smart grid (SG) and smart meters play a significant role in meeting the everlasting demand of energy consumption, supply, and generation for peer-to-peer (P2P) energy trading between different distributed prosumers. Whereas blockchain when used with P2P energy trading plays a major role in cost and security by eliminating any involvement of outsiders and third parties. However, existing works related to the blockchain with P2P energy trading are engaged in increasing the cost related to resource allocation, latency, computational processing, and large network setup. The objective of this paper is to design and develop a three-tier architecture, an analytical model, and a hybrid algorithm for network analysis in a blockchain-based P2P energy trading system using reinforcement learning (RL) and feed forward neural network (FFNN) techniques. In this model, we will examine the various parameters and tradeoffs which affect the delay, throughput, and security in P2P energy trading. This will lead to profitable P2P energy trading between different distributed prosumers. By analyzing the simulation results of the proposed model and algorithm by benchmarking with the existing state-of-the-art techniques it's clear that the proposed algorithm shows marked improvement over network latency generated results. The simulation of the model is conducted using the iFogSim simulator, Ganache with Ethereum platform, Truffle, Python editor tool, and ATOM IDE with solidity.

1. Introduction

The advancement in technology like blockchain with Peer-to-Peer (P2P) energy trading has grown the interest of the market holders for secure transactions and trading between prosumers. But this amalgamation of blockchain with energy trading has increased the network delay, and cost of resource allocation, and decreased the throughput [1]. Here, P2P energy trading between consumers and prosumers is based on blockchain and machine learning techniques. A variety of statistical indicators are used in this paper to assess the suggested predictive model's performance [2]. The efficiency of a blockchain platform is also assessed in terms of latency, throughput, and resource allocation. Hence, energy trading transactions and reward design concepts are majorly centred on smart contracts [3]. In our proposed analytical model, Q-learning with Neural Network (NN) is utilized to map the current state to the associated action. Following that, the expected return is used to calculate the action value to receive the goal observation information from the environment, and to provide the current environment's status information. The markov decision process (MDP) specifies the agent's interaction environment with Q-learning.

This paper proposes a blockchain-based network system in an MDP context. Moreover, a time component for network latency, state transition probability, and network size makes up the state of the system. Depending on the situation, the time component for the network refers to hourly, half-hourly, daily, or yearly information. The actor participants in energy trading are represented by agent A in the MDP tuple. Whereas R is the reward for the monetary incentive for a specific action, such as cost reduction, throughput maximization, or minimum delay [4].

See Fig. 1 in which the P2P energy trading system model is discussed using blockchain and Q-Learning MDP with NN to meet the latency requirement for energy trading involving smart meters.

Fig. 1 shows the P2P energy trading model. The task required to design the model for network and latency optimization with secure communication. In this system, end-users acting as prosumers and consumers can sell the electrical energy as per their needs and requirements depending upon how much energy is in stock available for the auction and bidding. These users are directly connected to utility offices or the control centre. Blockchain-based small servers embedded with machine learning(ML) algorithms acting at the edge of networks transfer the electrical data in a single hop count. Transmission and processing of critical real-time data occur on small servers rather than on the cloud.

We have used an assumption of the Proof-of-Work (PoW) blockchain in P2P energy trading. Some of the factors which lead to the poor performance of the P2P energy trading model are highlighted below:

- ***Resource allocation cost:*** The cost of resource allocation in a blockchain-based P2P energy trading system depends upon blockchain mining. In this model, miners compete for rewards [6,7]. Whereas blockchain rewards the miner depending upon the task to resolve. Significant investment is involved in resource allocation during blockchain mining [8,9]. This process is required to complete the P2P energy using blockchain[10].
- ***Network cost:*** P2P energy trading with blockchain requires the storage of a large number of transaction details that occurred between prosumers including block ids, gas limit to mine, gas price information, network IDs, and mining status. Moreover, the network setup consists of large data transmission between smart meters, prosumers, and consumers. This P2P trading system also consists of various distributed networks such as household area networks (HAN), building area networks (BAN), neighborhood area networks (NAN), and local area networks (LAN) which further increases the size of the network. It further raises the network cost and security issues in a blockchain-based P2P trading model [11].
- ***Trading and transaction throughput***: The throughput value is calculated by measuring the success rate of transactions related to energy trading between producers and consumers. Similarly, throughput values are also measured for electric data transmission from smart meter devices to local producers and consumers [12]. A blockchain-based P2P energy system is required to record such events promptly without any computational and network delay [13]. It's an important factor for efficient throughput analysis.

In current existing techniques, peer nodes can send 25–200 transactions per second. By expanding the number of peer nodes, the network traffic volume can also be enhanced. This resulted in a significant reduction in network throughput [14]. The network latency will increase as the number of peer nodes grows [15]. Therefore, to overcome the issue of large network size and delay a model is proposed in the blockchain environment for efficient P2P energy trading. Some of the major highlights of the proposed research work are listed below:

- Network latency minimization.
- Proof-of-work (PoW) validation for successful transactions.
- Processing time minimization.
- Packet error minimization.

The main objective of this paper is to develop a machine-learning model for network analysis in blockchain-based P2P energy trading between different prosumers. We analyze the different parameters and tradeoffs related to profitable P2P energy trading for network size and delay. Our main contribution is as follows:

- We designed a three-tier blockchain-based architecture for efficient P2P energy trading.
- We developed an analytical model for network analysis of blockchain-based P2P energy trading using a hybrid of reinforcement learning (RL) and feed forward neural network (FFNN) technique.
- Next, we designed and developed a novel hybrid reinforcement learning feed forward neural network (HRLFFNN) algorithm for P2P energy trading using a machine learning approach. The algorithm works to minimize the overhead related to high network delay,

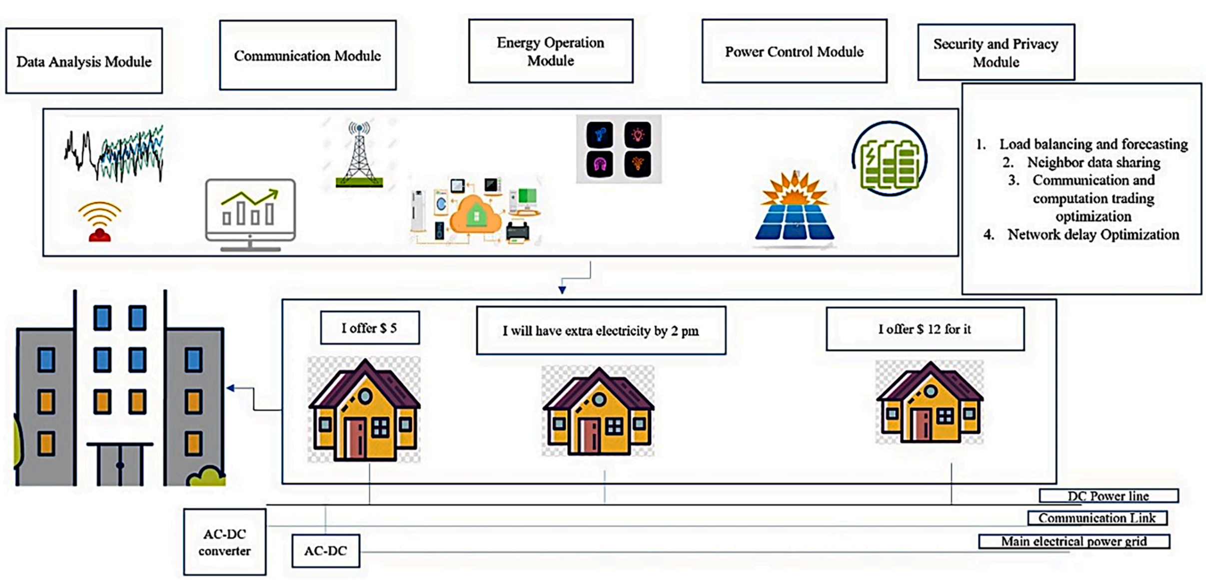

Fig. 1. Peer-to-Peer energy trading [5].

resource allocation, computational processing, and large network setup.

The paper is organized as follows: in Section 2 we discuss the background and related work for P2P energy trading using blockchain, whereas, in Section 3 we present the three-tier architecture, system model, and an analytical model for network analysis. In Section 4 we present a mathematical framework. In Section 5 we discuss the progression approach of evolution in FFNN. Furthermore, in Section 6 we present a novel hybrid machine-learning algorithm. In Section 7 the results and discussion along with benchmarking are discussed and in Section 8 we conclude the paper.

2. Background and related works

This section discussed the existing state-of-the-art techniques related to the role of blockchain and its effects on the P2P energy trading system in smart grid networks. Furthermore, the section highlights the existing issues such as

- The increase in the cost concerning resource allocation.
- Computational processing
- Large network setup

Some of the previous works tried to use machine learning techniques with blockchain in a P2P energy trading network system but they are unable to address the issue of large network setup and delay in a transaction which further leads to high energy trading costs [16]. In [17], the authors proposed an algorithm for negotiation in a P2P electricity market. The authors identified various computational challenges for negotiation in a P2P network. They tried to resolve the issue of communication delay between prosumers and consumers. Whereas in [5], the authors proposed a decentralized P2P architecture for efficient energy trading in real-time. Furthermore, they proposed a model for P2P trading to address the issue of security and a privacy-aware environment. A Multi-Agent System (MAS) was also presented concerning security and sustainability.

In [18], the author proposed a novel mechanism for P2P energy trading based on machine learning and blockchain hybrid techniques. Their proposed mechanism provides a predictive energy platform for trading which supports real-time negotiation and scheduling of energy resources in a distributed environment. The blockchain-based machine learning peer-2-peer (BMLP2P) model consists of two modules one is a blockchain-based energy trading, and another is smart contract-based predictive analysis. Using these modules authors were able to provide real-time energy consumption monitoring and prediction of short-term energy usage.

In [19], the authors proposed a decentralized mechanism for efficient P2P energy trading considering the privacy of prosumers and loss of powers and associated network fees. Therefore, they formatted a P2P energy trading model for network and grid usage. The model solves the problem of social welfare maximization. Next, an electrical distance approach was proposed to calculate network fees for the P2P energy trade. At last, they proposed a novel decentralized technique.

Similarly, in [20], the authors proposed an energy trading model to include other agents such as electricity suppliers. A non-cooperative gaming strategy was used between the suppliers and the community households. Furthermore, they proposed a distributed algorithm for the equilibrium in games.

In [21], the authors proposed a novel decentralized model for P2P energy trading using blockchain technology to overcome the challenges related to economic efficiency, information privacy, and the inter-temporal dependencies of storage devices. The model facilitated efficient and fraud-resilient trading while eliminating intermediaries' costs. They used an ant-colony optimization method for short-term auctions in the market layer. This helps in securing the privacy of the agents involved. Whereas the blockchains layer was able to provide automation, security, and real-time settlement using the implementation of smart contracts.

In [1], the authors proposed an agent-based model for P2P energy trading in a blockchain environment to highlight the trade-off between network size and delay. They were able to develop a method of efficient cost analysis for P2P energy trading using blockchain. They identified various factors for maximum throughput and profit. In [22], the authors proposed an automatic peer-to-peer energy trading deep q-learning (P2PET-DQL) technique. Their proposed model used markov decision process (MDP) which was further solved by reinforcement learning (RL) technique. They adopted a long-short-term delayed reward approach to maximize the profit in P2P energy trading. When compared with other existing techniques their proposed approach showed better results.

In [23], the author proposed a multi-agent-based deep reinforcement learning (MADRL) algorithm for P2P energy trading. This algorithm facilitates load balancing among the prosumers, consumers, and efficient utilization of resources with an optimal policy. Blockchain was used to guarantee the integrity, privacy, and security of energy trading and data transaction. In [24], the authors proposed a realistic P2P energy trading model for microgrids with deep neural network (DNN) and RL algorithms, to address the issue of decision-making in microgrids for local P2P energy trading. Their proposed decision-making process for energy trading was based on the concept of MDP. The MDP was used to find the optimal strategies. Their modified algorithm helps microgrids to utilize their resources to make better results. Moreover, their proposed Peer-to-Peer Deep Neural Network Reinforcement Learning (P2P-DNNRL) model was a hybrid of RL and NN in a blockchain environment.

In [25], the authors proposed an optimization model and blockchain architecture to manage crowdsourcing in a P2P energy trading transaction. Next, they designed and developed a two-phase operation algorithm that focuses on day-ahead scheduling and control of distributed energy resources. The proposed algorithm was able to work for real-time operation in distributed networks. In [26], the authors proposed an RL-based energy trading approach for microgrids which uses a Deep Q-Network to improve the utility of the microgrids. The proposed approach was able to reduce power losses and average power plant schedules. Their proposed novel reinforcement learning deep q -network (RLDQN) was able to minimize the network cost and network delay but it lacks real-world implementation.

In [27], the authors proposed a blockchain-enabled framework for P2P energy trading between prosumers and consumers in a Smart Grid (SG) network. They further designed a decentralized mechanism for market settlement. Which further helps in efficient negotiation. Performance and proximity were identified as parameters for selecting the partner in trading. A proof-of-location (PoL) algorithm was proposed to identify the location without disclosing the identity.

Similarly, in [28], the authors proposed a scheme called Energy Trading (ET) which consists of a smart contract-based secure energy trading approach for P2P in the SG system in the Ethereum platform. This scheme was proposed to overcome the issue of security, privacy, latency, and real-time settlement of the transaction in energy trading. In [29], the authors proposed a blockchain-based system for load balancing in a hybrid-decentralized P2P energy trading in SG to overcome the issue of trust and privacy. In this model, each agent can interact with the other without involving a third party. Furthermore, they proposed three smart contracts for P2P and prosumers requests. The model was able to reduce the cost of energy trading.

In [30], the authors proposed a P2P energy trading market platform with multiclass energy management to coordinate between different prosumers with heterogeneous preferences. They proposed a new concept of energy, which allows energy to be treated as a heterogeneous product. This minimizes the cost linked with the losses and degradation of the battery. Furthermore, it provides a distributed optimization mechanism for scalability and user data privacy. In [31], the authors proposed an SC-based architecture for P2P energy trading in a decentralized manner without the involvement of any third party. Furthermore, they proposed a framework for direct energy trading between prosumers and consumers in a blockchain environment.

In [32], the authors used a hyperledger fabric (HF) for secure and intelligent communication in P2P energy trading. They designed a novel framework for improving resource utilization and handling the energy crisis. The authors were able to overcome the issue related to a bottleneck in trading. In [33], the authors proposed a distributed electricity trading system for P2P electricity sharing between producers and consumers. The system consists of two layers. In the first layer, a multi-agent system was designed and developed to support the trading network. Next, an agent coalition system was created for electricity trading negotiation.

In [34], the authors proposed a fuzzy multi-objective programming-based model using blockchain technology for P2P energy trading. The proposed approach was a meta-heuristic technique to overcome the issue of investment cost and maintain a balance between power supply and demand owing. The use of blockchain technology ensures the security, privacy, and sustainability of participants in the microgrids. There's no role of central authority in energy system control and flow; instead, participants play a major role in it. Their proposed model was able to maximize the demand satisfaction of customers. The model was solved by a genetic algorithm.

Most of the existing state-of-the-art techniques such as BMLP2P, P2PET-DQL, MADRL, P2P-DNNRL, and RLDQN highlighted the issue of high network cost, large network size, and network delay in blockchain environment for P2P energy trading. The existing algorithms were unable to minimize the large network setup and delay for efficient P2P energy trading. The above-mentioned techniques use machine learning algorithms in a blockchain environment. However, our proposed novel algorithm and analytical model when compared with the existing techniques were able to minimize the network delay and size in a P2P energy trading network.

3. Three-Tier architecture and analytical model

In this section, we discussed the proposed Three-tier architecture and the analytical model. The architecture and model were designed to minimize the network delay for P2P energy trading in a blockchain environment. See Fig. 2 for the architectural design.

Fig. 2 illustrates the blockchain-based three-tier architecture for transaction allocation in different processes. This allocation of the transaction is conducted using Q-learning with the neural network (NN) approach to solve MDP for efficient P2P energy trading in a blockchain environment. The architecture consists of smart meters used by different peers, data connectors, a meter data management system (MDMS), and servers which consist of different blocks to store multiple transactions, and the records of these transactions are further communicated or sent to MDMS and cloud data centres using a communication network.

The main objective of this paper is to develop a network analysis method by minimizing the network delay and size in P2P energy trading using a blockchain technique, where each peer executes some process in parallel. These processes receive and record messages from neighbours. Moreover, these messages are checked for new transactions. Similarly, for a new transaction and block, a new process is invoked. The system contains the collection of the new transaction from each process and all the new transactions are placed in a queue.

The queue model will follow the first in first out (FIFO) process based on the assumptions made. Once the new transaction is added to a queue its information is sent to the local neighbours. There are a total of 4 processes and every process has different roles one process is used for creating and storing the transactions. Another process is used for validating the blocks, similarly, one process is used for storing the blocks in the blockchain. The fourth process is used for validating the PoW.

Here in this system, the transaction latency is the amount of time it takes to complete a transaction in a blockchain network. The transaction broadcast, submission, and consensus times all contribute to transaction delay. Similarly, the transaction roundtrip time is calculated as the time it takes for a transaction to complete from submission to execution. As the volume of user requests on the blockchain network grows, transaction latency increases.

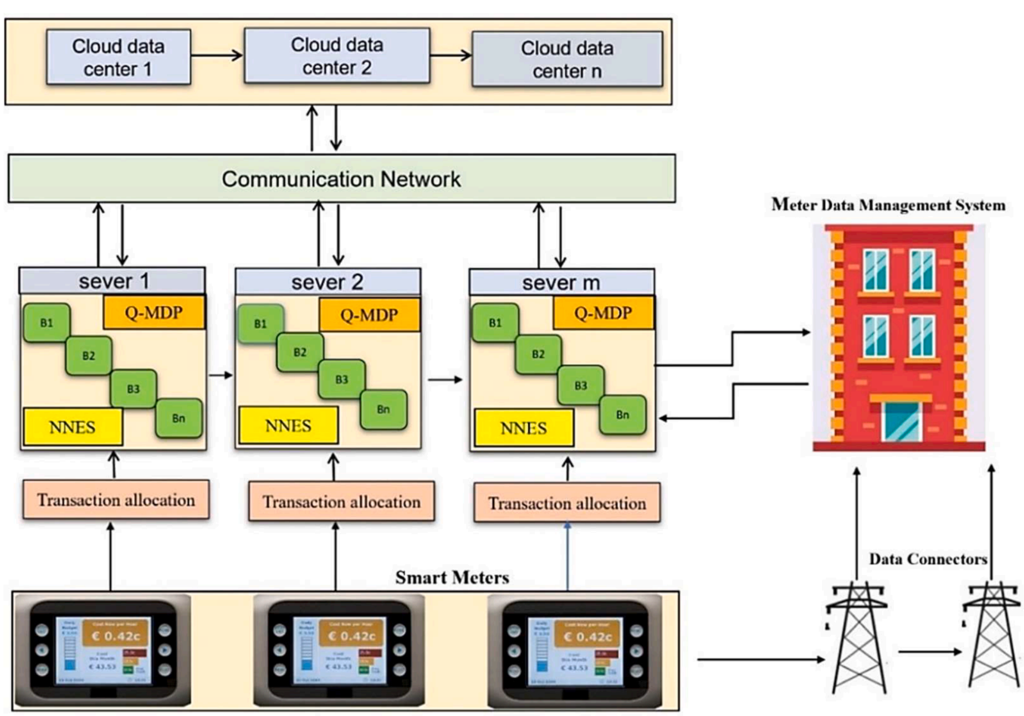

Fig. 2. Three-Tier Architecture.

The reinforcement learning algorithm was to minimize the network delay by allocating transactions to different blocks. Q-learning MDP is used for modelling in a dynamic environment by gathering feedback from experience. To account for the dynamic behaviour of the blockchain-based P2P energy trading system, the suggested approach requires a Q-learning MDP. Because of the continually changing incoming transaction requests at blocks, the P2P system was unable to forecast the transition probabilities.

Therefore, to address the issue, Q-learning MDP was used to create a decision-making process. It also uses quality action to enhance the total payout for the blockchain-based P2P system. RL helps to make the most use of available resources [35–37]. For high efficiency and decentralized intelligence, RL can provide rapid and intelligent decision-making. See Fig. 3 for the proposed novel ML-based blockchain model.

In Fig. 3 we illustrated an advanced system model. The model consists of a hybrid machine learning technique i.e., RL and NN algorithm working in a blockchain environment for efficient P2P energy trading. The model consists of three layers the first layer consists of smart meters deployed at the prosumer's sites where electrical data is transacted through data connectors and each transaction is linked with the attached timestamp. The transactions form a ledger and are then allocated to blockchain-based servers where different processes are allocated to different processors executing in the blocks which are acting as miners. The miners work to minimize the network size and delay by using Q-learning MDP and FFNN which further utilize the concept of neural network evolution strategies (NNES). The data is then transmitted to end-users in a single-hop count where the blockchain system is deployed at the edge of networks.

See Fig. 4 for the blockchain-based transaction settlement.

Fig. 4 shows the contract chain and the ledger chain in combination to perform a blockchain-based transaction settlement. Different verifiers are used to verify the transactions occurring between the prosumers. The hash digest of each contract is connected to the hash digest of the previous contract. Similarly, the hash digest of each ledger is connected to the hash digest of the previous ledger. Each contract consists of a block number, hash digest number, time stamp, and contract number. Whereas each ledger consists of a block number, hash digest number, timestamp, and ledger number. Moreover, Fig. 4 explains the transaction settlement and verification using blockchain.

See Fig. 5 for hybrid RL with FFNN (Feedforward Neural Network).

Fig. 5 shows the mapping of feedforward NN with RL. In which the current states are mapped to the corresponding action to calculate the action value based on the return from NN to RL. Hence once the target information is obtained from the environment. A piece of status information is provided from the current environment of the RL to the deep NN [38].

In the analytical model, RL with FFNN is utilized to map the current state to the associated action. The action value is then determined using the expected return. After that, get the goal observation information from the environment, and then provide the present environment's status information. The MDP is a specification of the environment in which the agent interacts and defines as a tuple of $(S, A, P, \ R, \gamma)$ where S is the state, A is the agent, P is the probability matrix, R is the reward function, γ is the discount factor. An analytical model is designed using MDP; the state S consists of a time component of network size, and an energy generation component.

The network's time component refers to hours and minutes, with information varying depending on the problem. The energy production and consumption patterns are referred to as the time component. The actor participating in P2P energy trading is represented by agent A in the MDP tuple. R stands for the monetary incentive for a specific action, such as network delay minimization.

The state transition probability is defined as P_i: $S_i \times A_i \times S_i \rightarrow [0, 1]$ Here, the probability is denoted as $P_i(s'|s,a)$. where action a in a P2P energy trading energy network could include the decision to trade energy, etc. And the reward is defined as R_i: $S_i \times A_i \rightarrow R_i$. The system's main goal is P2P transaction allocation on each block node by minimizing network delay and probability for transaction allocation.

The instant reward in a particular state s is

$$R_i(s,a) = U_i(s,a) \ + \ T_{pi}(s, \ a) - (N_{Di}(s,a) + O(s,a) \ ), \tag{1}$$

where '$U_i(s,a)$' is the instant utility, $N_{Di}(s,a)$ is the instant network

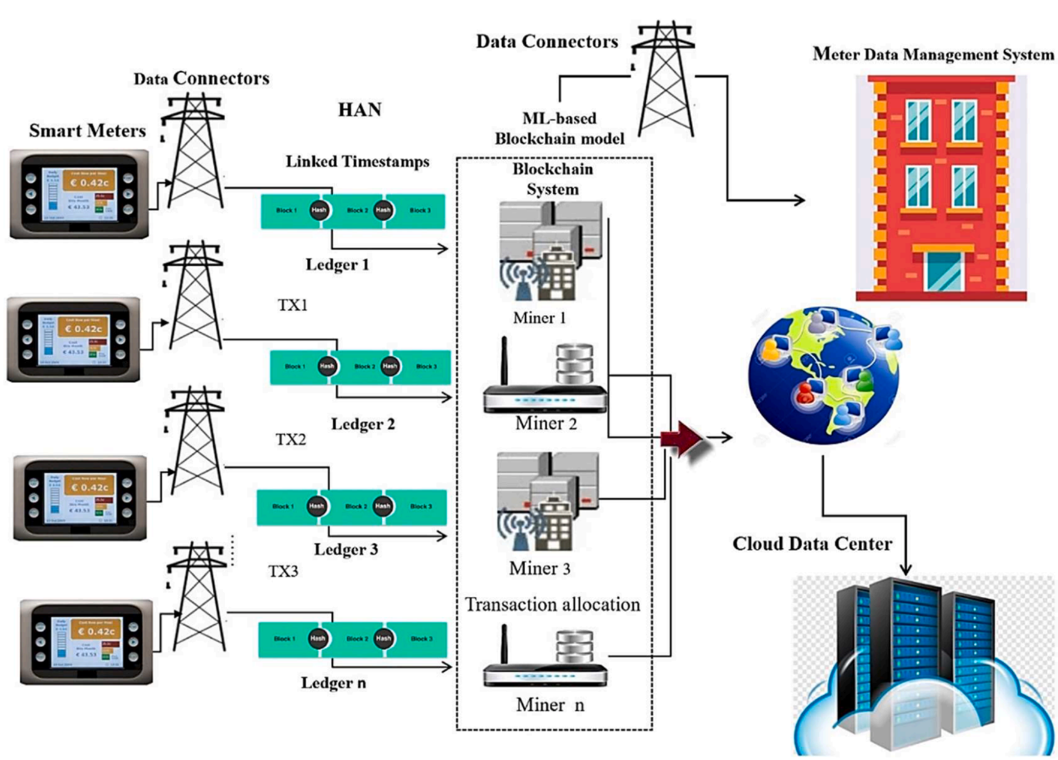

Fig. 3. Advanced machine learning-based blockchain model.

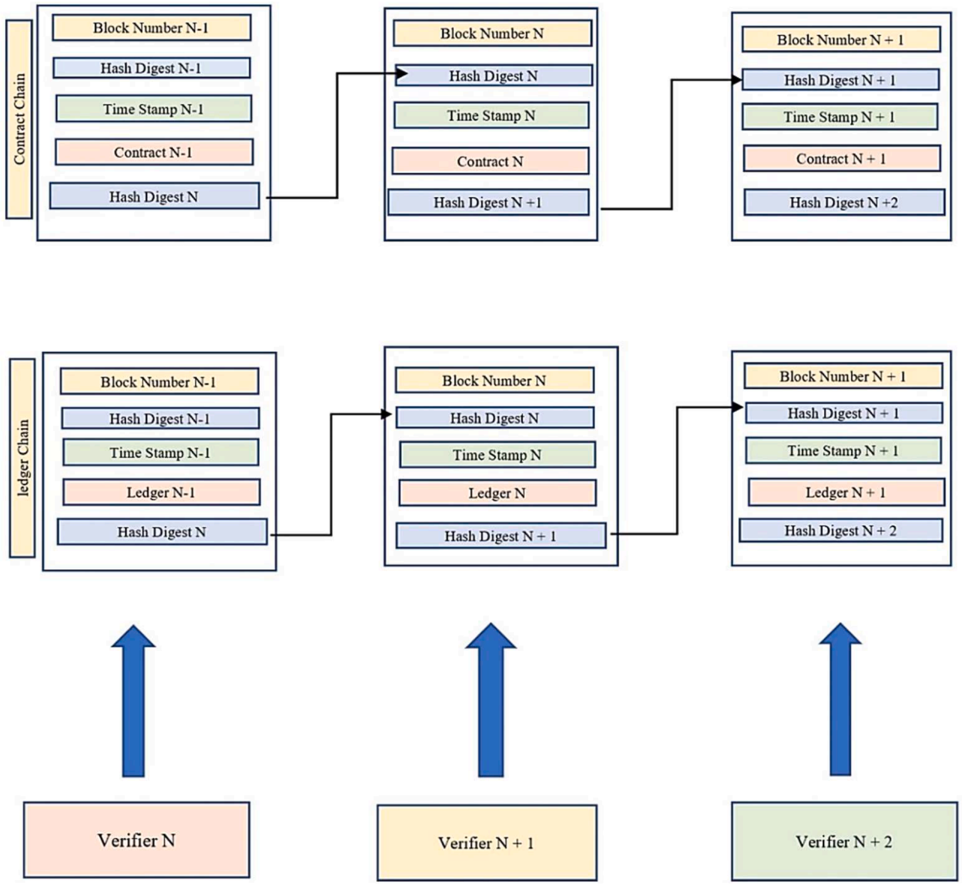

Fig. 4. Blockchain-based transaction settlement.

delay, $T_{pi}(s, \ a)$ is the instant throughput, and $O(s, a)$ is the transaction allocation probability function, in combination, respectively. In Eq. (1) $R_i(s,a)$ the instant reward is inversely proportional to the immediate system throughput, network delay, and transaction allocation probability function, and is directly proportional to the immediate utility and throughput. The instantaneous reward in the proposed analytical model is the ideal value of instant throughput assessed in terms of maximum throughput and minimal network delay in a given state s.

The instant utility is computed as

$$U_i(s,a) = r_{iu}\log\left(1+n^{1,2}+n^{2,3}+n^{3,4}\right) \tag{2}$$

where 'r_{iu}' is the reward utility. The number of transactions sent by a process P_1 to P_2 is $n^{1,2}$. The number of transactions sent by a process P_2 to P_3 is $n^{2,3}$. Similarly, the number of transactions sent by a process P_3 to P_4 is $n^{3,4}$. In Eq. (2) the evaluation of the predicted future reward is based on the instant utility function. The instant utility indicates how good the suggested system's reward is in future. The logarithmic sum of the number of transactions at each process is used to calculate immediate utility.

$$N_{Di}(s,a) = \chi_N\cdot(P_1+P_2+P_3+P_4\ )/\left(n^{1,2}+n^{2,3}+n^{3,4}\right), \tag{3}$$

where 'χ_N' is the network weight. Here, P_1, P_2, P_3, and P_4 are the processes. In Eq. (3) $N_{Di}(s,a)$ is the immediate network delay generated by taking action a from the state s . Immediate or instant network delay is the result of a combination of four independent processes that execute the transaction as well as block mining. Whereas $n^{1,2}$, $n^{2,3}$, and $n^{3,4}$ are the number of transactions.

$$T_{pi}(s,\ a) = N_{Di}(s,a) + \ D_{P2P} \tag{4}$$

$T_{pi}(s, \ a)$ is the throughput and D_{P2P} is the distance between two peers.

The probability for the transaction allocation $O(s, a)$ is calculated as

$$O(s,a) = \frac{\chi_{Ni}\ (n^{1,2}\cdot P_{1.\ allocation}\cdot P_{2.\ allocation}\ + n^{2,3}.P_{2.allocation}.P_{3.allocation} + n^{3,4}P_{3.allocation}.P_{4.allocation}\ )}{n^{1,2}\ +\ n^{2,3}\ +\ n^{3,4}}, \tag{5}$$

Eq. (5) calculates the transaction allocation probability function associated with the number of transactions allocated to different processes.

$$P_{i.allocation} = \frac{\max\left(0, \lambda_i - \left(Q_{i,\ \max} - Q_i^{'}\right)\right)}{\lambda_i}, \tag{6}$$

Eq. (6) is derived from Eq. (5). Eq. (6) calculates the transaction allocation probability $P_{i.allocation}$ for the individual processes depending upon the size of the transaction request and waiting time at the queue

$$Q_i^{'} = \min\left(\max(0, Q_i - s_i) + n^{i,j}, Q_{i,\ \max}\right), \tag{7}$$

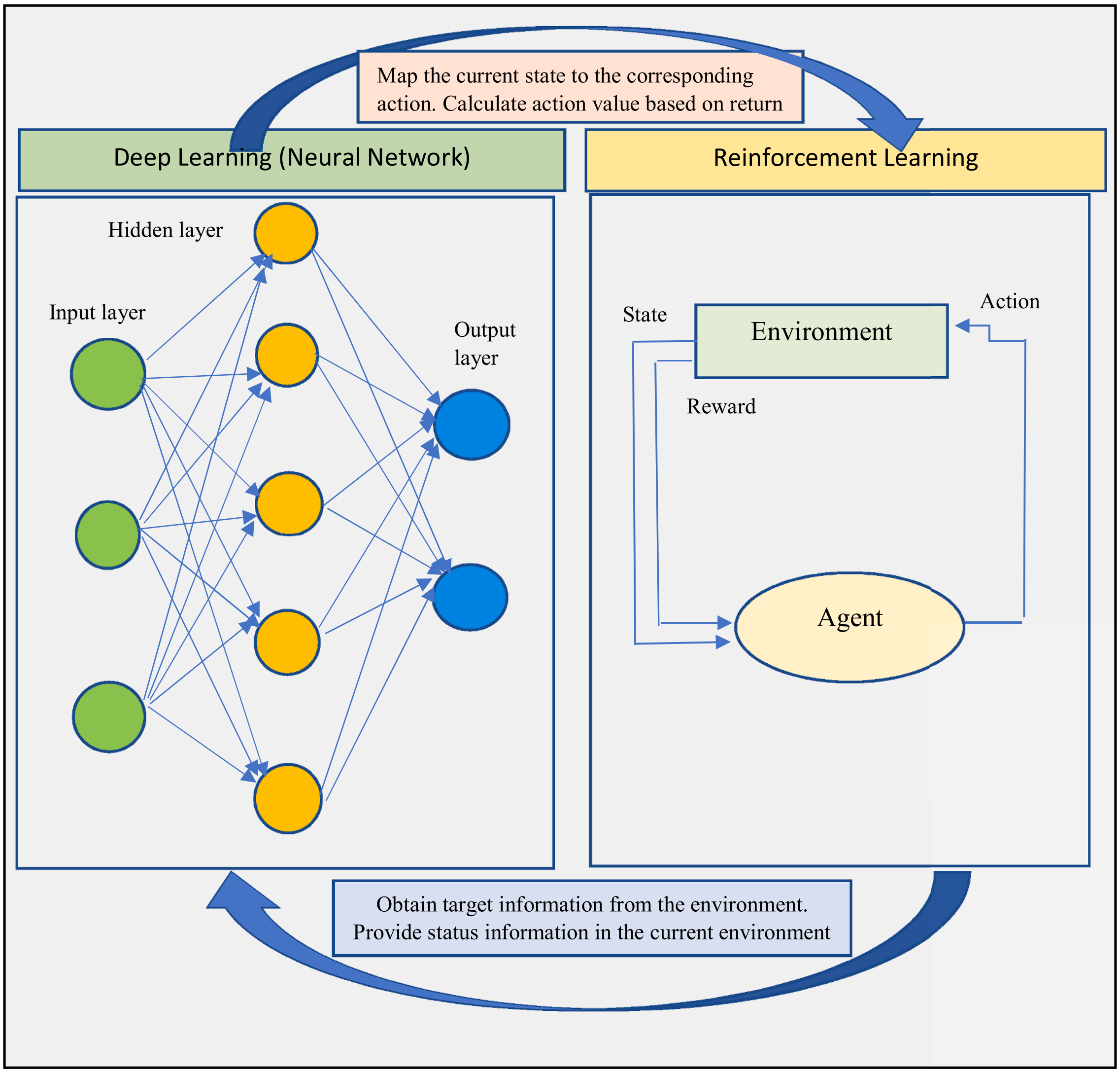

Fig. 5. Feedforward neural network with reinforcement learning.

Where χ_{Ni} is the transaction allocation weight. The service rate of a process P_i is s_i. The total count of transactions for processing at a process P_i is $n^{i,j}$. The transaction arrival rate at the process P_i is λ_i. $Q_i^{'}$is the next queue state.

Individual processes will send transaction traffic rates using a one-hop transmission channel. It's critical to certify the quality of service (latency requirement) for both the producer and the consumer. Prosumers encounter network delays as a result of big transactions and significant trading traffic. The goal of the proposed method is to maximize throughput by minimizing the network delay in a P2P energy trading system using blockchain. Eq. (8) defines the value function by satisfying the equation for bellman [39,40]:

$$\begin{aligned} \nu^*(s) &= \max_a \mathrm{E}(R_{i_{t+1}} + \gamma_i \nu^*(S_{i_{t+1}})|S_{i_t} = s, A_{i_t} = a) \\ &= \max_a \sum_{s',r} p_i(s', r|s, a)[r + \gamma_i \nu^*(s')] \end{aligned} \tag{8}$$

Where '$\nu^*(s)$' is the optimum value, R_i is the reward.

4. A mathematical framework for network delay minimization to maximize throughput

The system is unable to accurately forecast probability in maximal events. RL is offered as a way to discuss this constraint. The loss of transactions in RL is solved by paying attention to the backdrop details [41].

Different processes in a blockchain-based P2P energy trading system continuously detect the current state of the proposed system s with an action a. After then, there will be a transition. Q-function is defined as follows [39,40]:

$$Q(s,a) \leftarrow (1-\alpha_i)Q(s,a) + \alpha_i\left[R_i(s,a) + \gamma_i \max_{a' \in A_{i_{s'}}} Q(s',a')\right], \tag{9}$$

where 'α_i' is the learning rate. The value α_i is measured by calculating the difference between the new and the old Q- value. After a state change, Eq. (9) updates the value of the Q function. The system's process keeps track of the current state and action. The blocks continue to collect transactions.

Eq. (1) is used to determine the proposed approach's estimated reward function. The next state $s'$is obtained after identifying its three components. The adjacent process has the function of transmitting transactions for allocation to other processes, whereas the state's neighbouring queue is an arbitrary unit. The transaction size is determined in the next phase after the transaction arrives. For the i-th service transaction allocation and distribution phases, a small network delay and size are necessary.

We considered allocating transactions to various processes. Let (s_i^d, c_i^d, τ_i^d) denote a three-dimensional vector of the i-th transaction, where $\langle s_i^d, c_i^d, \tau_i^d \rangle$ are the transaction size, P2P system complexity, and network delay of the transaction that occurred between the processes. When the i-th transaction is allocated to the j-th process; then the network delay for the particular transaction is defined by

$$N_{ij} = \frac{s_i^d c_i^d + b_j^s}{f_j^{CPU}} \tag{10}$$

In Eq. (10) $f_j^{CPU}$and b_j^s are the processor frequency and storage size of the block j. The network delay minimization function is characterized by

$$D(t) \stackrel{\Delta}{=} min \sum_{i=1}^{\Omega(t)} \sum_{j=1}^{\Psi} y_{ij} N_{ij} \tag{11}$$

$$\sum_{i=1}^{\Omega(t)} y_{ij} = 1, \forall j \in \Psi, \tag{12}$$

$$N_{ij} \leq \tau_i^d, \forall j \in \Psi, \tag{13}$$

$$y_{ij} \in \{0, 1\}, \forall i \in \Omega(t), \forall j \in \Psi, \tag{14}$$

where '$\Omega(t)$' and 'ψ' are the sets of transactions and processes, and $y_{ij}$is the case for the allocation of transactions. respectively. N_{ij} is the network delay. The network delay reduction function (F_Δ) is expressed by

$$(F_\Delta) = \lim_{t \to \infty} \frac{1}{t} \sum_{i=1}^{t} D(i) \tag{15}$$

To fully process the upload of transactions to processes, the system's network delay between prosumers and consumers reaches its maximum. The greedy strategy is used to make the decision, which decreases network delay when transactions are uploaded. The training environment in our method is a system made up of procedures and blocks.

Values can be used to specify and express the status of the system. These variables are (1) demand (complexity and the number of transactions for allocation), (2) remaining transactions in the process queue, (3) time consumed from the last transaction to the present transaction, and (4) series of requirements from the final transaction. We need to measure the time taken by the process to finish the computation of the preceding remaining transaction in the queue when the transaction is uploaded for allocation to the process and blocks. In a blockchain-based P2P energy trading system, calculate the computation and network delay of the procedure and send the value as a $[K \times 1]$vector. Following that, we determine how much time the procedure will allocate if the arriving transaction is approved. The value is preserved in the second vector of $[K \times 1]$comparable size.

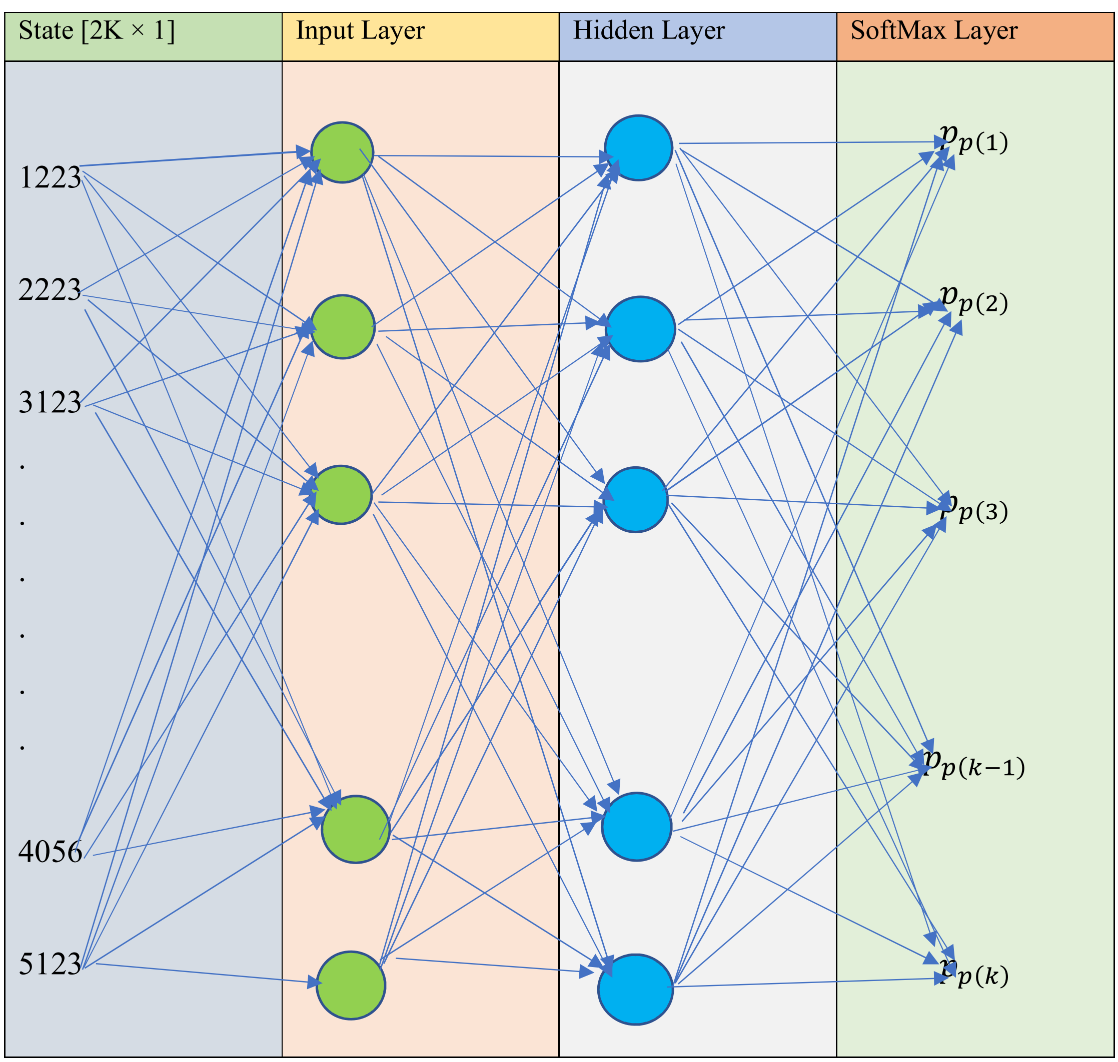

Fig. 6. The FFNN states and layers.

We generate a vector $[2K \times 1]$ that exemplifies the condition of the system near a given time by combining the two vectors. It defines the size of the state with a total number of $2K$' nodes in the input layer of the FFNN system. H^j denotes the node in the hidden layer. The relationship between the input layer and the hidden layer is defined as $[M \times K]$. To allocate transactions, we measure the computational and network delay in milliseconds for processes and blocks. Transactions need a small number of megacycles to allocate; as a result, there is a difference in estimated network delay between the processes. In Eq. (16) the value of the node in the hidden layer is calculated as

$$H^j = \sum_{i=1}^{2K} \left(W_{i,j}^{(1)} \times Z^{(i)} \right) \tag{16}$$

The node '$\widehat{T}^{(f)}$' value in the output layer is

$$\widehat{T}^{(f)} = \sum_{j=1}^{M} \left(W_{j,f}^{(2)} \times Z^{(i)} \right) \tag{17}$$

The probability of selecting the node with minimum latency is calculated as

$$p_{p(i)} = \frac{\widehat{T}^{(i)}}{\sum_{f=1}^{k} \widehat{T}^{(f)}} \tag{18}$$

See Fig. 6 shows the different layers used in the FFNN.

Fig. 6 shows the structures of the different layers and states along with the different node values in the FFNN model.

5. Progression approach for the evolution

The difficulties of transaction selection and allocation to processes and blocks can be determined using an RL model. The goal is to choose an action that reduces the long-network delay and size of the blockchain-based P2P network system [42]. The reward generated is defined as

$$\text{Reward}_i = \frac{1}{N(t)}, \tag{19}$$

where $N(t) = N(t-1) + N_{ij}^{(t)}$, $N(0) = 0$

Here, $N_{ij}^{(t)}$ is the latency. From Eq. (15) (F_Δ) minimizes the delay of the P2P system. Hence, the reward is defined as

$$Reward_i = \frac{1}{\sum_{f=t-k}^{t} N_{ij}(F)} \tag{20}$$

Eq. (20) is derived from Eqs. (1) and (2). The main goal of the proposed study is to solve the MDP in a blockchain-based P2P energy trading system by minimizing the high network delay by maximizing the predicted future benefit using the Q-learning method. According to Eq. (20), the reward is inversely proportional to the system network delay, i.e., the reward is highest when the network delay is lowest and vice versa. In addition, Eq. (20) is mathematically and statistically proven. FFNNs are trained using Neuro Evolution (NE), also known as neural network evolution [41,43].

Another method for training feed-forward neural networks is Neuro Evolution (NE), or neural network evolution, which is inspired by biological evolution. Each iteration now has a NN allocated to it, and a new generation is created from the NN. This generation is based on the NN [41,44]. The children are chosen based on a bigger reward for renewing the NN [45]. Evolutionary techniques are used to update the NN.

The proposed novel algorithm demonstrates the mechanism for transaction allocation and selection. The suggested innovative algorithm employs the greedy and FFNN approaches, with judgments made using the greedy strategy to reduce the schema's network delay during transaction allocation. In an RL context, evolution techniques update the FFNN [41,43,46].

The mean reward is calculated as $(Mean\_reward_i)$. Where the weight matrix is defined as

$$W_{j,f}^{(i)} = W_{j,f}^{(i)} + \alpha_i \times \sum Gain_i^{(H)} \times W_{j,f}^{(i)(H)}, H = [1, 2 - - - - -, M], \tag{21}$$

6. Hybrid reinforcement learning feed forward neural network algorithm (HRLFFNN)

In this section, we presented a novel HRLFFNN algorithm for efficient P2P energy trading between producers and consumers in a blockchain environment, where a transaction has occurred between two users to broadcast the transaction to all prosumers. The negotiation protocol of two users begins in a blockchain environment. Transactions are validated using a consensus algorithm by prosumers. Several new blocks are created for the verified transactions. Next, the blocks are attached to the respective platforms. A method to execute a trading algorithm using smart contracts in a blockchain environment consists of prosumers as participants.

Prosumers will be informed about the set of trades from different prosumers to buy energy. The meter readings will be uploaded to the smart contract after a specific time interval, and the smart contract will settle payment between the prosumers after verifying the prosumers' energy supply-demand information. T is the number of transactions required for each smart contract execution.

To collect energy supply-demand information, at most n transactions are transmitted to a smart contract. The smart contract then receives at least n transactions from the smart meters to collect actual energy generation and consumption data. The large network size can affect the performance of the P2P network with blockchain. A large number of peers involving various LAN, HAN, and BAN, with smart meters and transactions over distant peers. Which further increases the network diameter. *High network latency* affects the performance of a P2P energy trading system with blockchain, as the time needed to disseminate the number of transactions and blocks depends upon the service and network delay. The blockchain system is divided into splits into several blockchains.

Each peer executes some process in parallel. Here the process receives messages from neighbours and checks for new transactions and new blocks. Information for a new transaction and the new block is transferred to the other process. The process gathers new transactions in a queueing model which is first in first out (FIFO). This establishes the proof-of-work (PoW) protocol. At last, examine the block uses. There is a fee associated with each transaction. Miner gets a reward for publishing a new block. Blockchain processes a new block in a different time interval. The algorithm works in a blockchain environment for secure P2P energy trading using a digital signature. RL is used with the greedy method for transaction allocation and selection in the blockchain environment of P2P energy trading.

HRLFFNN Algorithm symbols notations

FCn: Final contract
$Cn_{i,j}^{QK}$: Contract
o': Counteroffer
o: Offer
Θ_K: Task
S_m: Smart Meters
K_{sy}: Symmetric key
A_{sy}: Asymmetric key
$Encrypt_{Sy}$: Symmetric encryption
$Encrypt_{Asy}$: Asymmetric encryption
C: Ciphertext
C_K: Cipher key
$PrVT_K$: Private key
PuB_K: Public key
C_s: Cloud server
BC_n: Blockchain nodes

(*continued on next page*)

(*continued*)

E_{Dp}: Electrical energy data packet
SPARK: Real-Time Analyzer (RTA)
S_m _*D*: Smart meter data
Hc: Hash code
k: Key
C_t: Ciphertext
T_S: Timestamp
C_k: Cipher key
D: Memory
MG: Micro Grid
P2P: Peer-2-Peer
Q: Quality of action
s: State
a: Action
r: Reward
NN: Neural network
A_{PuB_K}: Asymmetric public key
A_{PrV_K}: Asymmetric private key

HRLFFNN Algorithm Steps

Requirement: Smart meter devices, P2P energy trading data, blockchain nodes, and prosumers.
Step 1: Creation of blockchain-based P2P system.
Step 2: Prosumer selection.
Step 3: Retrieve trading data.
Step 4: Start the energy trading process.
Step 5: Encryption Process.
Step 6: Symmetric Key generation.
Step 7: Signature generation and use of Diffie-Hellman key exchange.
Step 8: Hash code generation.
Step 9: Ring formation by mixing signature.
Step 10: Final contract creation.
Step 11: Decryption process.
Step 12: Data allocation at the blockchain nodes.
Step 13: Next, check the availability of free processors.
Step 14: Attached timestamp of the energy trading data.
Step 15: Formation of ledgers.
Step 16: To perform mining and data allocation at the individual block nodes.
Step 17: Verification of keys.
Step 18: Send the hash code to the miners.
Step 19: Initialization of Q-values.
Step 20: Execute action and observe the reward.
Step 21: Store transitions.
Step 22: Selection of trading strategy by applying the e-greedy algorithm.
Step 23: Update NN weight.

HRLFFNN Algorithm

START
(blockchain-based *P2P* system is created)
function Encrypt (E_{Dp})
Observe actual energy generation and demand
Send the intended amount of energy trading $x(k)$
for $j = 1, 2, \ldots, N$ do
Receive the intended amount of energy trading from the neighbour's MG $x(k)$
Sell the amount of energy
else
Purchase Y_{ij} amount of energy from *MG*
end
Input request, from the requester
Check the status of *P2P* contracts
if *requester* == *registered*
Store input values
else
Register requester
Store input values
end if
if *requester* == *seller*
P2P.seller()
else if *requester* == *buyer*
P2P.buyer()
end if
end if

(*continued on next column*)

(*continued*)

if depict energy
buyEnergy()
else if surplus energy
sellEnergy()
end if
trading_result()
end if
function *trading_result*(){
Start energy transaction
Store results
end function}
for task Θ_k, seller j generates an offer o to buyer i
if *i* accepts o then
generate a contract $Cn_{i,j}^{Q_k}$ based on o
$FCn_{i\Theta_k}^{tmp} = FCn_{i,\Theta_k}^{tmp}$ o $Cn_{i,j}^{\Theta_k}$;
$FCn_{j,\Theta_k}^{tmp} = FCn_{j,\Theta_k}^{tmp}$ o $Cn_{i,j}^{\Theta_k}$;
return;
else
end if
if S_m confirms E_{Dp} storage over blockchain then
Generate a K_{sy}
for each E_{dp} do ($S_m <- C_T$)
$C_t + T_S <- E_{Dp}$
$C_t <- Encrypt_{sy}(E_{Dp}, K_{sy})$
$C_k <- Encrypt_{Asy}(K_{sy}, FCn_{i\Theta_k}^{tmp} PuB_K, FCn_{j,\Theta_k}^{tmp} PuB_K)$
else
Operation is not performed
end if
end function
function *SIGNATURE* (E_{Dp})
Generation (A_{PuB_K}, A_{PrV_K})
$H_c <-$ Hash for E_{Dp}
Digital signature design using H_c and signed $SPrVT_K$
Share $SPuB_K$to the BC_n
Ring formation
else
operation is not performed
end if
end function
function *Decrypt* ($C_T, C_K, FCn_{i\Theta_k}^{tmp} PrVT_K, FCon_{j,\Theta_k}^{tmp} PrVT_K, K_{sym}$)
$K_{sy} <- Decryption_{Asy}(C_k, FCn_{i\Theta_k}^{tmp} PrVT_K, FCn_{j,\Theta_k}^{tmp} PrVT_K)$
$E_{Dp} <- Decryption(C_T, K_{sy})$
end function
Initialize *Q*-values $Q(s, a)$ arbitrarily for all state-action pairs.
for each step until learning do
Choose an action a in the current state S based on *Q*-value estimate $Q(s, a)$
Select action a and observe the outcome state S' and reward r
Initialize trading replay memory D to capacity N.
Calculate *Utility*
Store transition and update parameters
Update the system using a policy gradient
Collect the current demand rates
Forecast energy
Reset Q
end for
for $t \epsilon T$ do
Forecast *R(t), D(t),* and observe *S(t)*
From experience sequence *Q (t)*
Input $Q(t)$ with Θ and get Q
Choose trading strategy X (t) using e-greedy
for $j\varepsilon N$ do
Receive the intended energy $X(t)$ from *MG*
end for
end for
Applying ∊- *greedy algorithm*
Transaction allocation
$Q(s,a) \leftarrow (1-\alpha_i)Q(s,a) + \alpha_i[R_i(s,a) + \gamma_i \max_{a' \in A_{i_s}} Q(s', a')]$
Calculate $Mean\_reward_i$
Evaluate Parent NN
Obtain the *NN* output
Update parameters
Update the NN weights
END

7. Results and discussion

This section discusses the performance evaluation and analysis along with a simulation overview and settings for the proposed model for generated results and state-of-the-art comparison.

7.1. Performance evaluation and analysis

In this section, the execution of the proposed novel Hybrid Reinforcement Learning Feed Forward Neural Network (HRLFFNN) algorithm is evaluated and analysed. A benchmarking of the existing techniques was conducted to verify the proposed algorithm and model for examining the robustness of the performance measures.

7.1.2. Simulation overview and simulation settings

The performance of the ML-based blockchain model that incorporates the proposed algorithm is analysed through simulation and experiments conducted for 10 epoch cycles. The baseline for this simulation is the minimum network setup, network delay/latency, minimum error, and minimum computational processing of transactions in a blockchain-based server. To simulate the ML-based blockchain model, iFogSim an open-source software, Ganache, truffle in Ethereum platform and solidity-based ATOM editor tool is used. First, it is required to install the Geth-go Ethereum node to interact with the smart contract.

In the iFogSim simulator, the optimum minimum network latency for Quality-of-Service (QoS) requirement in P2P energy trading is generated with 7 smart meter devices after 400 iterations.

Next Ganache was issued to inspect blocks and transactions. This will help to mine these transactions in a block and network id. Ganache is used to see the account address, timestamp, energy traded, amount, and several transactions along with the private key. Moreover, it will help to set the mining block time. Next, installing Node JS (i.e., the server-side JavaScript application) helps to interact with the Ethereum node. Using Truffle, we have compiled and tested the contracts, we have also used ATOM a text editor tool to edit the smart contracts written in solidity language. The algorithm was implemented using Netbeans and Python with several main packages, modules, and classes. See Fig. 7 for the Graphical User Interface (GUI).

Fig. 7. shows the physical topology configurations built- in the iFogSim simulator. The configuration in Fig. 7 is solely based on the concept of a proposed system and analytical model. Fig. 7 shows the various smart meters deployed at the edge of the local networks for energy trading between different households. The smart meters and prosumers are connected to various data connectors and servers with distributed blockchain servers such as m-0–1, m-0–0, m-1–1, m-1–0, d1, and d2. Furthermore, these blockchain servers are connected to proxy servers and cloud servers. See Fig. 8 for the transaction record.

Fig. 8. shows the block with the transaction record. The interface consists of block id, nonce number, transaction data, and the block's hash value. The transaction record consists of a timestamp, the amount of energy traded and the transaction amount. See Fig. 9. User Interface (UI).

Fig. 9. shows the UI for node information, balance, and account of the producers and agents involved in the P2P energy trading. Furthermore, it also shows the transfer amount from the source address to the destination address with information related to the transaction hash for the individual transaction that occurred during trading. See Fig. 10 UI for the Ganache simulator.

Fig. 10. shows the use of the Ganache simulator for the deployment of smart contracts in the Ethereum platform. The simulator shows the address, balance measurable using Ether, Transaction cost (Tx), Index value, and secure private key associated with each address involved in a P2P energy trading. A 95% confidence interval is included in Figs. 11-15 for the true value parameters. See Fig. 11 for network latency.

Fig. 11. shows a comparison of network latency between the proposed algorithm and blockchain-based cloud server for P2P energy trading. The blockchain server processes the incoming transactions and sends the transaction to another node. These nodes send transactional data to the cloud servers and utility centres. The proposed algorithm minimizes the network. See Fig. 12. for the successful transaction.

Fig. 12 shows the successful transaction between different prosumers for efficient energy trading. The number of blocks is increased in different physical topology configurations. The proposed algorithm easily outperforms the blockchain-based cloud server in a successful

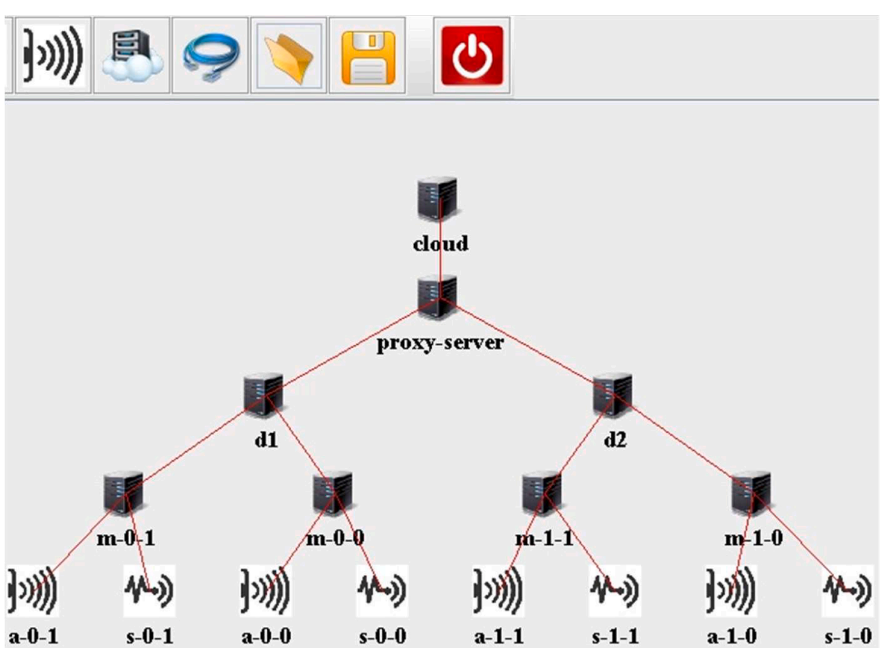

Fig. 7. GUI for physical topology configuration.

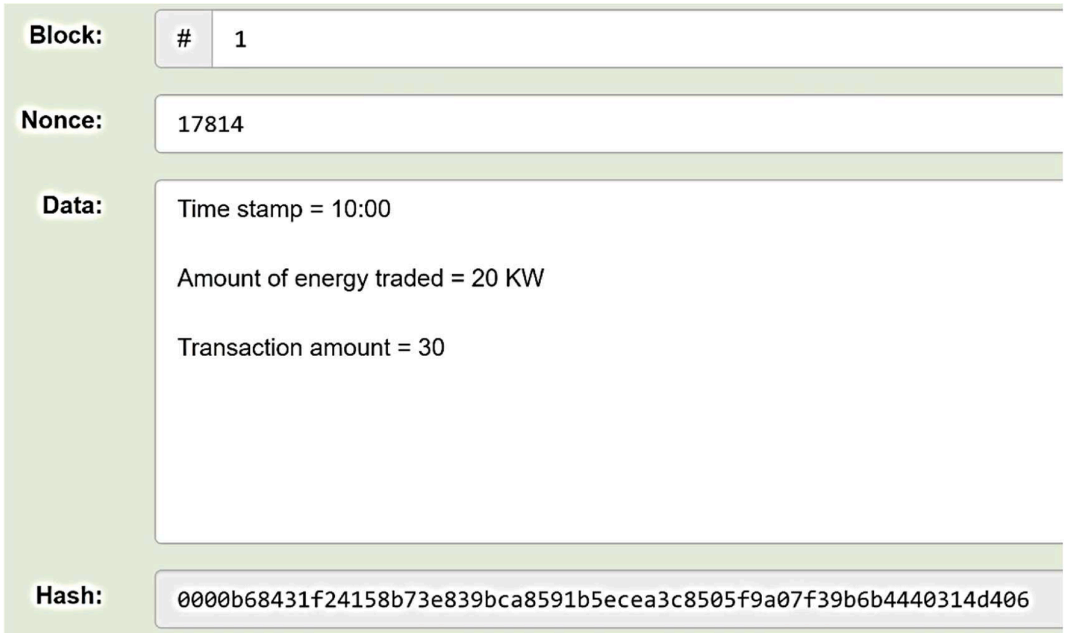

Fig. 8. Block with the transaction record.

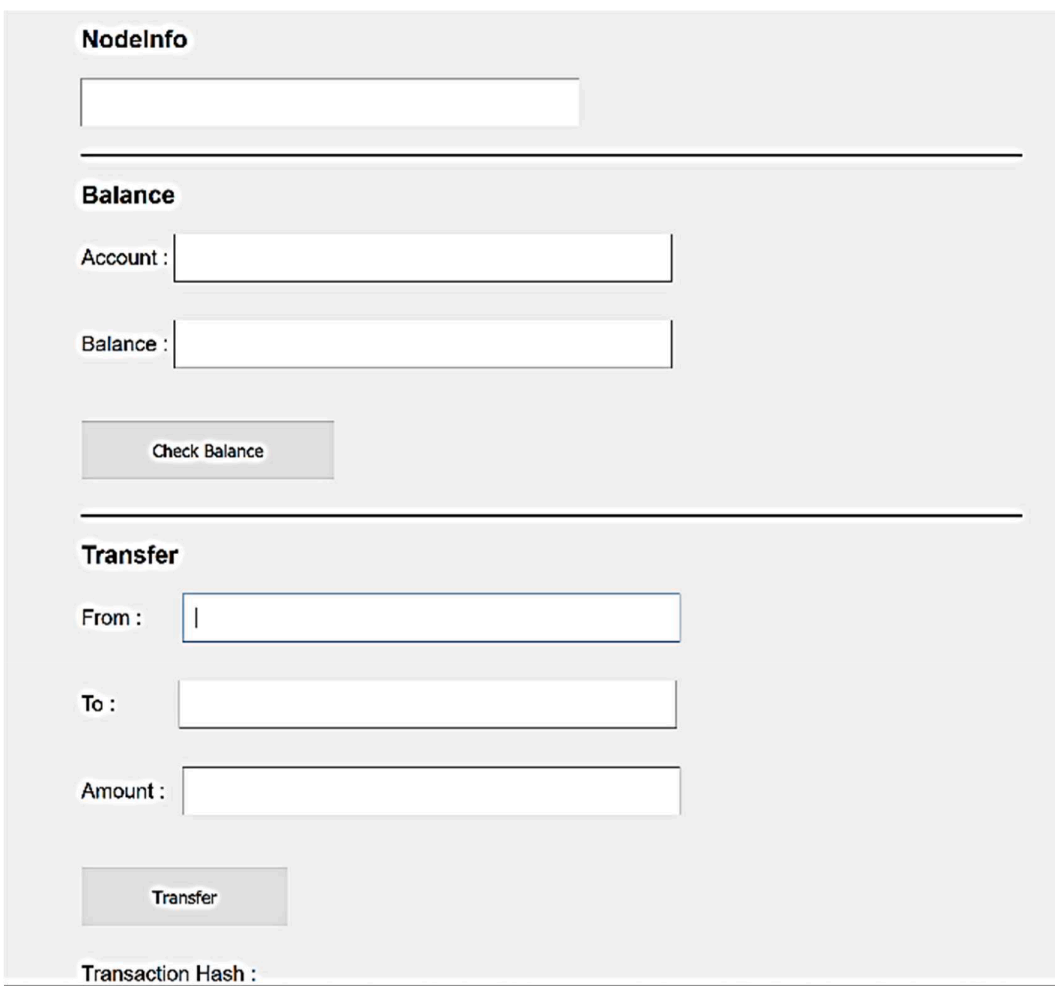

Fig. 9. User Interface for transaction account balance and transfer information.

Fig. 10. Use of Ganache simulator for deploying contracts in Ethereum.

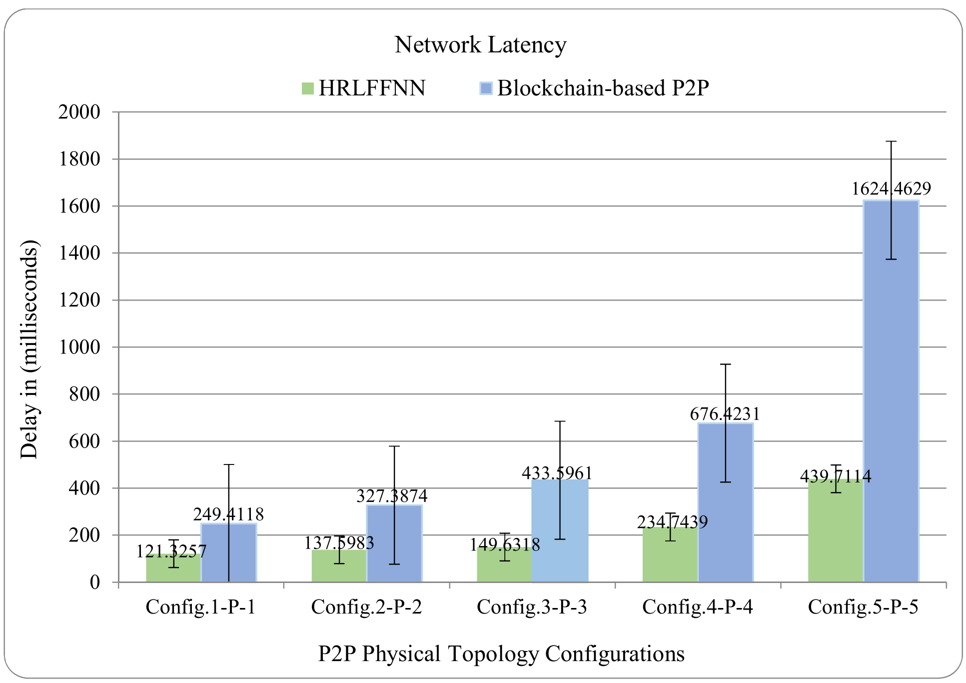

Fig. 11. Network latency comparison for the proposed algorithm and the blockchain-based cloud.

transaction to validate the PoW. See Fig. 13. for the processing time.

Fig. 13 shows the processing time comparison between the proposed algorithm and the blockchain-based cloud server. The processing time required by the blocks for processing the transactions is much lower than the blockchain-based cloud servers. The minimum processing time using the HRLFFNN algorithm is 331 ms and the maximum processing time using the HRLFFNN is 516 ms. The proposed algorithm easily outperforms the existing blockchain-based server. See Fig. 14. for the packet error.

Fig. 14 shows the packet error using the HRLFFNN algorithm and blockchain-based cloud server at different intervals of time. The figure shows the packet error in the proposed algorithm is much lower when compared to the blockchain-based technique. The minimum number of error packets in HRLFFNN at a time interval of 10 min is 9. Whereas the maximum number of error packets at a time interval of 50 min is 31. Hence, the proposed algorithm easily outperforms the blockchain-based server for error packets during efficient P2P energy trading. See Fig. 15. for the benchmarking.

In Fig. 15 the performance of the proposed HRLFFNN algorithm is evaluated based on the value of network latency. The novel HRLFFNN algorithm easily outperforms the other existing works like BMLP2P, P2PET-DQL, MADRL, P2P-DNNRL, and RLDQN. The HRLFFNN algorithm shows a network latency of 183.1783 ms. Whereas BMLP2P and P2PRT-DQL show a latency of 214.7601 ms and 247.4186 ms respectively. MADRL and P2P-DNNRL show a latency of 289.5086 ms and 310.8725 ms. Similarly, RLDQN shows a latency of 342.2022 ms. The latency value of HRLFFNN is minimum when compared with other blockchain-based machine-learning techniques. The proposed algorithm yields marked improvement over other state-of-the-art techniques and algorithms.

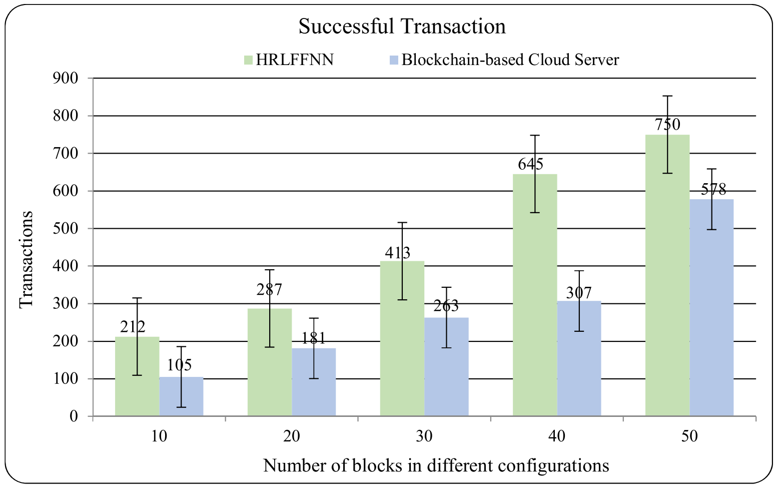

Fig. 12. PoW for successful transaction in the proposed algorithm and blockchain-based cloud.

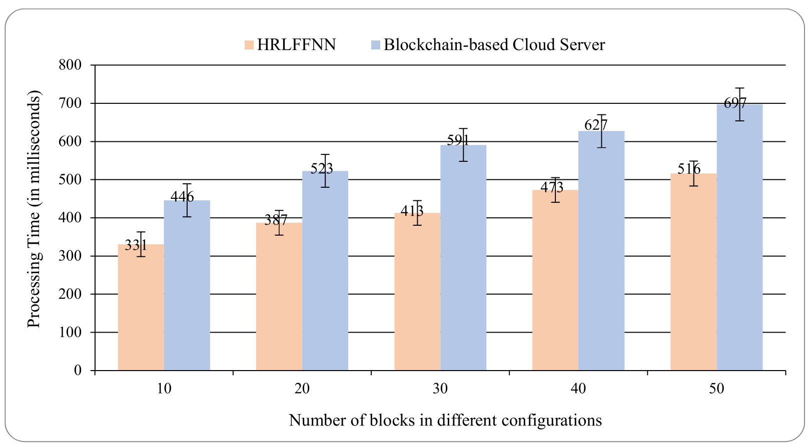

Fig. 13. Processing time comparison for the proposed algorithm and blockchain-based cloud.

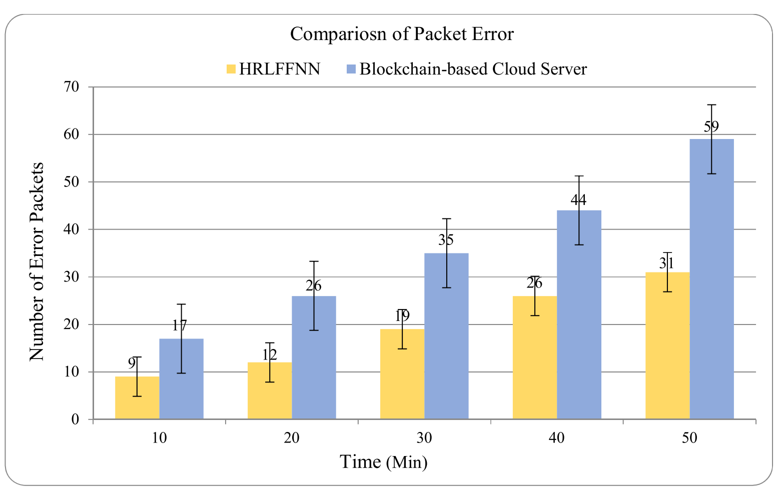

Fig. 14. Packet error comparison in the proposed algorithm and blockchain-based cloud.

8. Conclusion

P2P energy trading involves a large volume of transactions between different distributed producers and consumers. This trading involves several smart meter devices which further generate a large volume and veracity of electrical data. Using P2P energy trading prosumers are involved in energy trading i.e., buying and selling. Nowadays, existing technologies and smart cities society are using blockchain techniques for secure and efficient P2P energy trading. Blockchain avoids the involvement of any third party, outside attackers, hackers, False Data Injection

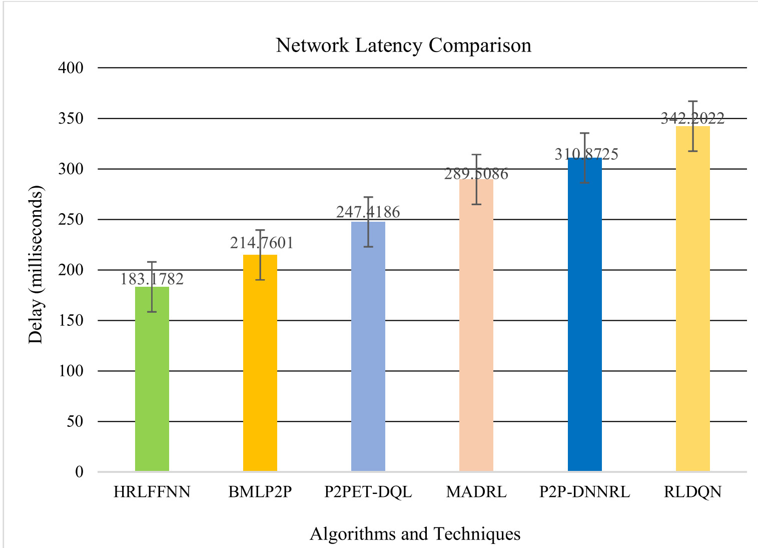

Fig. 15. State-of-the-art techniques comparison for network latency.

(FDI), and several other anomalies. However, P2P energy trading involves a large network setup including LAN, HAN, BAN, and WAN. Moreover, with the inclusion of blockchain technology, the P2P energy trading network setup increases along with the system complexity which in turn increases the delay and network latency of the system.

Therefore, to minimize the network setup and network delay. We proposed a blockchain-based 3-tier architecture, machine learning-based blockchain analytical model and HRLFFNN algorithm to increase the throughput by minimizing the packet error, network delay, and processing time of successful transactions. When compared for performance analyses and an evaluation with the existing technologies such as BMLP2P, P2PET-DQL, MADRL, P2P-DNNRL, and RLDQN using blockchain and machine learning techniques the proposed algorithm easily outperforms them in terms of network delay. The proposed model and algorithm successfully address the problem of high network latency in P2P energy trading. Future research work requires the implementation of the analytical model and algorithm in a real-world environment.

Author statement

We wish to submit a revised article entitled “Network Analysis in a Peer-to-Peer Energy Trading Model using Blockchain and Machine Learning” for publication in the Computer Standards and Interfaces

We confirm that this work is original and has not been published elsewhere, nor is it currently under consideration for publication elsewhere. Its submission for publication has been approved by all authors.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

The authors are thankful to the Deanship of Scientific Research at Najran University for Funding this work, under the Research Groups Funding program grant code (NU/RG/SERC/12/13).

References

[1] S. Thakur, J.G. Breslin, Cost Analysis of Blockchains-based Peer to Peer Energy Trade, in: 2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), IEEE, 2020.

[2] S. Hussain, et al., Fuzzy logic weight based charging scheme for optimal distribution of charging power among electric vehicles in a parking lot, Energies 13 (12) (2020) 3119.

[3] S. Hussain, M.A. Ahmed, Y.-C. Kim, Efficient power management algorithm based on fuzzy logic inference for electric vehicles parking lot, IEEE Access 7 (2019) 65467–65485.

[4] Saurabh Shukla, Mohd Fadzil Hassan, Low Tang Jung, Azlan Awang, Muhammad Khalid Khan, An analytical model to minimize the latency in healthcare internet-of-things in fog computing enviornment.https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0224934, 2019. 14(PLOS ONE 14(11): e0224934): p. 1–31.

[5] J. Kalbantner, et al., P2PEdge: a decentralised, scalable P2P architecture for energy trading in real-time, Energies 14 (3) (2021) 606.

[6] S. Shukla, et al., A blockchain-enabled fog computing model for peer-to-peer energy trading in smart grid, in: International Congress on Blockchain and Applications, Springer, 2021.

[7] C. Ge, et al., Secure keyword search and data sharing mechanism for cloud computing, IEEE Transac. Depend. Secure Comp. 18 (6) (2020) 2787–2800.

[8] C. Ge, et al., Revocable attribute-based encryption with data integrity in clouds, IEEE Transac. Depend. Secure Comp. 19 (5) (2021) 2864–2872.

[9] C. Ge, et al., A verifiable and fair attribute-based proxy re-encryption scheme for data sharing in clouds, IEEE Transac. Depend. Secure Comp. 19 (5) (2021) 2907–2919.

[10] A. Yazdinejad, et al., Secure intelligent fuzzy blockchain framework: effective threat detection in iot networks, Comp. Industry 144 (2023), 103801.

[11] S. Shukla, et al., Identification and authentication in healthcare internet-of-things using integrated fog computing based blockchain model, Inter. Things 15 (2021), 100422.

[12] N. Wang, et al., Distributed machine learning for energy trading in electric distribution system of the future, Electr. J. 34 (1) (2021), 106883.

[13] C. Ge, et al., Revocable identity-based broadcast proxy re-encryption for data sharing in clouds, IEEE Transac. Depend. Secure Comp. 18 (3) (2019) 1214–1226.

[14] C. Ge, et al., Attribute-Based Proxy *Re*-Encryption With Direct Revocation Mechanism for Data Sharing in Clouds, IEEE Transac. Depend. Secure Comp. (2023).

[15] A.H. Wheeb, Performance analysis of VoIP in wireless networks, International Journal of Computer Networks and Wireless Communications (IJCNWC) 7 (4) (2017) 1–5.

[16] C. Ge, et al., Attribute-Based Encryption with Reliable Outsourced Decryption in Cloud Computing Using Smart Contract, IEEE Transac. Depend. Secure Comp. (2023).

[17] F. Moret, et al., Negotiation algorithms for peer-to-peer electricity markets: computational properties, in: 2018 Power Systems Computation Conference (PSCC), IEEE, 2018.

[18] F. Jamil, et al., Peer-to-peer energy trading mechanism based on blockchain and machine learning for sustainable electrical power supply in smart grid, IEEE Access 9 (2021) 39193–39217.

[19] A. Paudel, et al., Peer-to-peer energy trading in smart grid considering power losses and network fees, IEEE Transac. Smart Grid 11 (6) (2020) 4727–4737.

[20] M. Zhang, et al., Energy trading with demand response in a community-based P2P energy market, in: 2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), IEEE, 2019.
[21] A. Esmat, et al., A novel decentralized platform for peer-to-peer energy trading market with blockchain technology, Appl. Energy 282 (2021), 116123.
[22] J.-G. Kim, B. Lee, Automatic P2P energy trading model based on reinforcement learning using long short-term delayed reward, Energies 13 (20) (2020) 5359.
[23] Y. Xu, et al., Deep reinforcement learning and blockchain for peer-to-peer energy trading among microgrids, in: 2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics), IEEE, 2020.
[24] T. Chen, S. Bu, Realistic peer-to-peer energy trading model for microgrids using deep reinforcement learning, in: 2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe), IEEE, 2019.
[25] S. Wang, et al., Energy crowdsourcing and peer-to-peer energy trading in blockchain-enabled smart grids, Systems 49 (8) (2019) 1612–1623.
[26] L. Xiao, et al., Reinforcement learning-based energy trading for microgrids, arXiv preprint (2018). arXiv:1801.06285.
[27] M. Khorasany, et al., Lightweight blockchain framework for location-aware peer-to-peer energy trading, Int. J. Elect. Power Energy Syst. 127 (2021), 106610.
[28] A. Kumari, et al., ET-DeaL: a P2P smart contract-based secure energy trading scheme for smart grid systems, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2020.
[29] R. Khalid, et al., A blockchain-based load balancing in decentralized hybrid P2P energy trading market in smart grid, IEEE Access 8 (2020) 47047–47062.
[30] T. Morstyn, M.D. McCulloch, Multiclass energy management for peer-to-peer energy trading driven by prosumer preferences, IEEE Transac. Power Syst. 34 (5) (2018) 4005–4014.
[31] D. Han, et al., Smart contract architecture for decentralized energy trading and management based on blockchains, Energy 199 (2020), 117417.
[32] A. Lohachab, et al., Performance evaluation of Hyperledger Fabric-enabled framework for pervasive peer-to-peer energy trading in smart Cyber–Physical Systems, Future Generation Comp. Syst., 118 (2021) 392–416.
[33] F. Luo, et al., A distributed electricity trading system in active distribution networks based on multi-agent coalition and blockchain, IEEE Transac. Power Syst. 34 (5) (2018) 4097–4108.
[34] Y.-C. Tsao, V.-V. Thanh, Toward sustainable microgrids with blockchain technology-based peer-to-peer energy trading mechanism: a fuzzy meta-heuristic approach, Renew. Sustain. Energy Rev. 136 (2021), 110452.
[35] R.K. Naha, et al., Fog Computing: survey of trends, architectures, requirements, and research directions, IEEE access 6 (2018) 47980–48009.
[36] M. Aazam, E.-N. Huh, Fog computing and smart gateway based communication for cloud of things, in: 2014 International Conference on Future Internet of Things and Cloud, IEEE, 2014.
[37] A. Gowri, P.S. i Bala, An agent based resource provision for iot through machine learning in fog computing, in: 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), IEEE, 2019.
[38] Naveed, N., H.T. Madhloom, and M.S. Husain, Breast cancer diagnosis using wrapper-based feature selection and artificial neural network.
[39] T.P. Le, N.A. Vien, T. Chung, A deep hierarchical reinforcement learning algorithm in partially observable Markov decision processes, IEEE Access 6 (2018) 49089–49102.
[40] W.T. Scherer, S. Adams, P.A. Beling, On the practical art of state definitions for Markov decision process construction, IEEE Access 6 (2018) 21115–21128.
[41] L. Mai, N.-N. Dao, M. Park, Real-time task assignment approach leveraging reinforcement learning with evolution strategies for long-term latency minimization in fog computing, Sensors 18 (9) (2018) 2830.
[42] Saqib E. Awan, Mohammed Bennamoun, Ferdous Sohel, Frank M. Sanfilippo, Benjamin J. Chow, Grish Dwivedi, Feature selection and transformation by machine learning reduce variable numbers and improve prediction for heart failure readmission or death, PloS one (2019) 13.
[43] Z. Michalewicz, Evolution Strategies and Other methods, in Genetic algorithms+ Data structures= Evolution Programs, Springer, 1994, pp. 167–184.
[44] A. L`heureux, et al., Machine learning with big data: challenges and approaches, IEEE Access 5 (2017) 7776–7797.
[45] G. Manogaran, et al., Wearable IoT smart-log patch: an edge computing-based Bayesian deep learning network system for multi access physical monitoring system, Sensors 19 (13) (2019) 3030.
[46] M.G.R. Alam, Y.K. Tun, C.S. Hong, Multi-agent and reinforcement learning based code offloading in mobile fog, in: 2016 International Conference on Information Networking (ICOIN), IEEE, 2016.