IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2021.Doi Number

Towards Designing a Trusted Routing Scheme in Wireless Sensor Networks: A New Deep Blockchain Approach

Saad M. Darwish
[1]Department of Information Technology, Institute of Graduate Studies and Research, Alexandria University, Alexandria, Egypt
Corresponding author: Saad M. Darwish (saad.darwish@alexu.edu.eg)

ABSTRACT Routing is a critical process in Wireless Sensor Networks (WSNs) since it is responsible for data transmission to base stations. Routing attacks are capable of completely destroying and degrading the function of WSNs. A trustworthy routing system is critical for ensuring routing security and WSN efficiency. Numerous studies have been conducted to increase trust between routing nodes, including cryptographic techniques, and centralized routing decisions. Nonetheless, the majority of routing methods are impractical in practice, since it is difficult to identify untrusted activities of routing nodes effectively. Meanwhile, there is no efficient method of preventing malicious node attacks. As a result of these issues, this article offers a trusted routing method that combines deep-chain and Markov Decision Processes (MDPs) in order to enhance the routing security and efficiency of WSNs. To authenticate the process of transmitting the node, the proposed approach utilizes a Proof of Authority (PoA) method inside the blockchain network. The validation group necessary for proofing is selected using a deep learning methodology that focuses on the properties of each node. MDPs are then used to choose the appropriate next hop as a forwarding node capable of transferring messages simply and securely. According to testing data, our routing system still performs well in a 50% malicious node routing environment when compared to existing routing algorithms.

INDEX TERMS Wireless Sensor Networks, Trusted Routing, Deep-Chain, Blockchain, Markov Decision.

I. INTRODUCTION

The multi-hop routing mechanism is a fundamental component of WSN technology. Nonetheless, the dispersed and dynamic characteristics of WSN render multi-hop routing susceptible to a variety of attack patterns, compromising security [1]. A malicious node may emit erroneous queue length information in order to increase the likelihood of receiving packets, hence altering the routing schedule of other routing nodes. Current routing algorithms have difficulty identifying such malicious nodes, since it is difficult to distinguish between two routing nodes' real-time changes in routing information [2].

When a malicious node receives data packets from a neighbor node, it discards them immediately rather than forwarding them to the next-hop neighbor node. This results in a data "black hole" in the network, which is difficult to detect in WSNs for routing nodes (see Fig. 1) (3). These malicious nodes might be external attackers or legal internal nodes that have been intercepted by external attackers. Trust management has been a common method of assuring the routing network's security in recent years. This approach enables the routing node to identify reasonably trustworthy routing connections efficiently. On the other hand, its use is constrained by the fact that the trust values of nearby routing nodes may be accessible by just one routing node that does not fully adhere to the distributed multi-hop WSN.

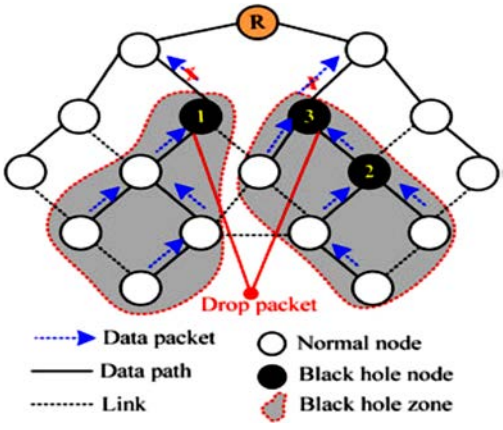

FIGURE 1. **Black hole attack.**

There has been a significant amount of study in recent years on blockchain technology and routing algorithms [4].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098933, IEEE Access

The blockchain is a decentralized database that is maintained by several nodes and is essentially concerned with problems of trust and security. The blockchain relies on four critical technological features to deliver reliable and secure services [1]: (1) Distributed ledger stores all the transactions on the blockchain. Every node maintains a full ledger; no data tampering is possible. Every node may be used to monitor the legality of transactions. (2) Asymmetric encryption and authorization technique: Information recorded on the blockchain is public, but account identifying information is encrypted and may only be accessed by the data owner, thereby protecting the security of data and personal privacy. (3) The consensus method is how all nodes on the network agree on the transaction's legitimacy, making tampering difficult. See [5] [6] for different types of consensus methods. (4) Smart contracts, which use secure and uncorrupted data to automatically run the prescribed instructions by a blockchain miner. A smart contract's execution outcome modifies the ledger state on the blockchain network. Since this has been certified by a particular consensus procedure, the material cannot be manipulated or tampered with.

Proof of Authority (PoA) is a Byzantine consensus technique that is used for authorization and private blockchain technology [6]. The method is based on a group of reputable entities (i.e., authorities) referred to as validators. Validators gather, construct, and add blocks to the chain in response to consumer transactions. As a result, we must pay special attention to the selection of validators. Recent advances in reinforcement learning have enabled wireless nodes to watch and acquire information from their effective local operational environment, learn, and make efficient routing choices on the fly [6]. A common decision-making strategy is to determine the optimal next-hop based on the present situation. Numerous academics have identified Markov Decision Systems (MDSs) as one of the most appropriate decision-making techniques for a random dynamic approach to solve this problem. Each hop in the routing process may be thought of as a state in this case, with each hop choosing one of the best hops. Then, by making consecutive judgments, messages may be sent effectively and securely to their destination [7].

We present a novel trustworthy routing system for WSNs based on blockchains and Markov Decision Processes in this study. We use proof of authority inside the blockchain network to validate the node transmission phase in particular. To do this, a deep neural network is employed to choose the salient nodes that represent the node-dependent validators' features. Through the attributes associated with each node, a deep-learning model augments the collection of validators. The technique leverages the decentralized, tamper-resistant, and traceable nature of blockchain transactions to increase the integrity of routing information across routing nodes. The MDPs model is used to assist routing nodes in making more informed routing choices and selecting the most dependable and efficient routing links.

This study expands substantially on our conference paper [39]. In comparison to this condensed version, more information on the proposed strategy are offered, as well as a more detailed performance assessment. Additionally, we include a more extensive literature analysis to contextualize the proposed strategy and make the study more self-contained. As a result, this edition of the paper presents a more comprehensive and methodical explanation of the earlier work.

The rest of this article is as follows: Along with some preliminary information, Section 2 summarizes current strategies for a reliable routing method in WSNs. Section 3 discusses the suggested trustworthy routing model. Section 4 presents many experimental results that demonstrate the suggested model's efficacy. Finally, in Section 5, we will complete the paper and outline future goals.

II. BACKGROUND AND SURVEY

A. PRELIMINARIES

1) WIRELESS SENSOR NETWORK

The wireless sensor network examined in this article is primarily utilized for event detection and data collecting. Due to the short communication distance between sensor nodes, data is often sent through many hops to the base station. Routing algorithms prioritize determining the safest route between sources and sink nodes. Assume that in a sensor network with an area of $L \times L$, there are a number of common nodes and a single sink node in the center, and that after deployment, all nodes remain stationary [8]. All sensor nodes are isomorphic; they all have routing, transmitting, and receiving functions; any two nodes may interact in a single-hop or multi-hop fashion, and their start states are identical. The nodes' initial load is zero, and their initial energy is equal to E_0. The manner of information sharing between nodes is as follows [9]. That is, each node has a unique identifier ID; this node stores information such as residual energy, packet IDs, next-hop IDs, and sender IDs. This information is updated in real-time in response to changes in the forward neighbor. It is worth noting that the nodes' traffic queues are restricted, and data packets are handled in the first-in-first-out (FIFO) manner.

The forward neighbor node set contains nodes that are neighbors of node i within the maximum communication radius R. The following definition applies to the forward neighbor node set [10]:

$$\mathrm{FN}(i) = \{a | d_{in} \leq R.\, d_{as} < d_{is}\} \quad (1)$$

where node a is any forward neighbor node of node i, d_{ia} is the distance from node i to node a, d_{as} and d_{is} are the distances from node i and node a to the sink node respectively, R is the maximum communication radius of node i. In this case, for any node $a \in \mathrm{FN}(i)$, the energy consumption during the communication between node a and sink node is the forward energy consumption which denoted as e_{as}.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098933, IEEE Access

This work takes into account the geographic connection between the current node i, the forward neighbor node a, and the sink node s, as well as the energy state of each node, in order to construct a data transmission channel that consumes less energy and has a shorter latency. The closer node i's forward neighbor node is to the sink node, and the smaller the straight-line distance d_{is}, the fewer hops it takes forward along the route, and the quicker data can be transferred to the sink node. Simultaneously, from the standpoint of residual energy, it is anticipated that the lower the energy consumption e_{ia}, the greater the forward distance d_{as}, and node a may endure more forward energy consumption e_{as}. On the other hand, when the residual energy E_i is plentiful and the forward neighbor node a's residual energy E_a is typically little, it is intended that node i may share as much transmission energy consumption as feasible. The transmission energy efficiency ratio is calculated based on the aforementioned two factors to reflect nodes' capacity to balance energy consumption during data transfer. It is stated as follows:

$$p(a) = \frac{d_{is}}{d_{ia}+d_{as}} \cdot \left(\frac{e_{as}}{E_i} + \frac{e_{ia}}{E_a}\right) \quad (2)$$

$$p(a) = \frac{p(a)}{max\{p(a)\}} \quad (3)$$

Where $P(a)$ is a benefit index, the greater the value of $P(a)$, the more efficiently energy is used to ensure that data is transferred along the shortest route feasible [11].

2) BLOCKCHAIN

Assume that $G(.)$, and $H(.)$ are cryptographic hash functions with output in the range $\{0.1\}^k$. (Assume both are SHA-256.). A block is a triple of the type $B =< s.x.ctr >$, where $s \in \{0.1\}^k$, $x \in \{0.1\}^*$, and $ctr \in \{0.1\}^\ell$ meet the $validBlock^D(B)$ predicate [12]:

$$\left(H\left(ctr.G(s.x)\right) \leq D\right) \quad (4)$$

$D \in \mathbb{N}$ is the difficulty level, and ℓ with $q \leq 2^l$ is used to guarantee that ctr is sufficiently short, e.g., $\ell = 32$. The limitations on ctr are rather arbitrary; in general, any subset of all bit strings with at least q items may be restricted to ctr. In general, a lower D will be more challenging; the likelihood of success will be $D/2^k$. Here, s represents the connection to the previous block in the chain, x represents the chain's additional content, and ctr represents the freedom to discover a block meeting the predicate $validBlock^D(B)$.

- A series of blocks is referred to as a blockchain. Its rightmost block, marked head, is the chain's head or end $head(\mathcal{C})$.
- By convention, an empty string ε is also a chain $head(\varepsilon) = \varepsilon$.
- A chain $\mathcal{C}$ with $head(\mathcal{C}) =< s'.x'.r' >$ may be made longer by appending a valid block $B =< s.x.r >$ with $s = H(r'.G(s'.x'))$. For $\mathcal{C} = \varepsilon$ any block may extend it, i.e., there is no constraint on, for example, set $s = 0$. The expanded chain $\mathcal{C}_{new} = \mathcal{C}B$ now contains$(\mathcal{C}_{new}) = B$.
- $len(\mathcal{C})$ is the *length* of a chain.
- We indicate $\mathcal{C}^{\lceil\kappa}$ of length m, for any $k \in \mathbb{N}^0$ the chain by deleting k times its head, i.e., pruning the k rightmost blocks.
- For $k \geq len(\mathcal{C}).\mathcal{C}^{\lceil\kappa} = \mathcal{E}$.
- If $\mathcal{C}_1$ is a prefix of $\mathcal{C}_2$, i.e., $\exists k \in \mathbb{N}^0\left[\mathcal{C}_1 = \mathcal{C}_2^{\lceil\kappa}\right]$, we denote $\mathcal{C}_1 \preccurlyeq \mathcal{C}_2$.

Fig. 2 depicts the Blockchain's primary component [13]. Most importantly, the consensus process is the mechanism through which every accounting node comes to an agreement on the efficacy of an interruption avoidance transaction. Bitcoin chains have variable difficulty, i.e., D varies between blocks and is determined by the chain's contents up to that point (which includes time stamps). In general, since the number of nodes and the hashing power per node every round are constant, it is believed that the difficulty D is constant as well.

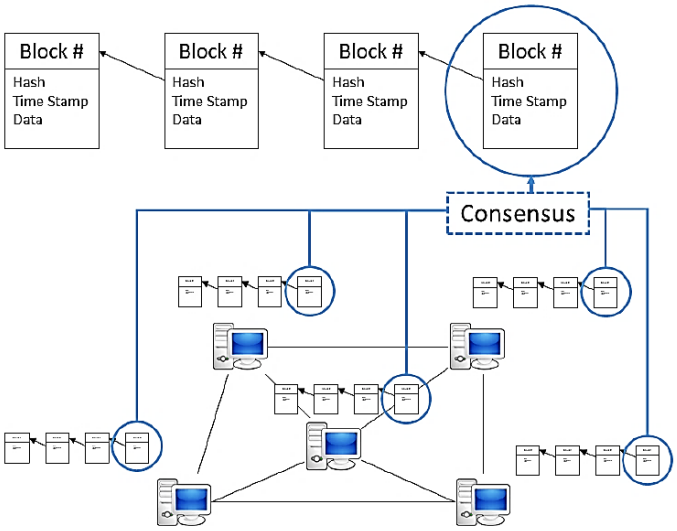

FIGURE 2. **Key elements of blockchain systems [13].**

3) CONSENSUS IN BLOCKCHAIN

The blockchain network's nodes operate autonomously, with no central authority overseeing them. In an ideal world, all network members would always agree on the same new block to be added to the blockchain, and the network would consist of a single blockchain. However, in actuality, nodes may get detached from the network or may even behave maliciously in Byzantine situations. As a result, a fault-tolerant consensus process that is agreed upon by all nodes in the blockchain is necessary to settle any possible disagreement [14].

The consensus procedure is composed of five stages: Propose, Prevote, Pre-commit, Commit, and NewHeight. The proposer broadcasts a suggestion to its peers during the Propose stage. A proposal contains the block, the signatures of the validators who verified the block, and the proposer's signature. If the proposer secured a block during the previous round's pre-commit, that block will be utilized for the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098933, IEEE Access

proposal. In the absence of this, a new block will be formed. During this time frame, all nodes will chatter about the proposal to their surrounding peers.

Each validator will vote for a block and inform their neighbors during the Prevote stage. A vote contains the hash of the voted block, the voter's signature, the kind of vote - prevote or pre-commit - as well as round and height information. The blocks to be included are picked in the following order: (1) a locked proposed block from a previous round, and (2) a valid and approved block from the current proposal. If neither is available, the neighbors are notified through a special NIL prevote. All nodes will communicate all round prevotes to their surrounding peers.

The validator examines if it has gotten more than two-thirds of prevotes for an approved block during the Pre-commit stage. If there is one, the validator releases the current lock and signs and broadcasts a pre-commit vote for this block, rather than locking it. Additionally, the validator wraps the locked block's prevotes into a proof-of-lock that will be used to build the block in the following Proposal. If there are less than two-thirds prevotes, the validator will not sign or lock any block. During this time frame, all nodes will communicate to all surrounding peers all pre-commits for the round.

If the node receives more than two-thirds of pre-commits for a given block at the end of Pre-commit, it will continue to the Commit stage. Otherwise, it advances to the following round's Propose stage. Two simultaneous requirements must be satisfied in the Commit stage before the consensus method may cycle back to the Propose stage. To begin, the node must have received the block from one of its peers in order to sign and broadcast the commit to the rest of the peers. Second, the node must wait until the network has received at least two-thirds of the block's commits. Once these conditions are met, the node will set the Commit Time property to the current time and proceed to the NewHeight phase, where it will remain for a specified period. The goal is to enable nodes to wait for further commitments to the committed block that were missed during Pre-commit owing to network latency difficulties. After the specified timeframe has expired, the algorithm resumes from Propose. If a node obtains more than 2/3 commitments for a specific block at any point throughout the consensus process, it will immediately enter the Commit stage.

***Definition* 1:**
Blockchain ( ) = (||*x*: {0...N-1} @ (Propose(*x*); Prevote(*x*); Pre-commit(*x*); PreparePOL(*x*); Commit(*x*))); *NextRound* (); Where P; Q represents process P followed by process Q and $P \parallel Q$ represents synchronous processes P and Q.

In-process Prevote (*x*) simulates malicious node behavior; an honest node verifies the proposed block as-is, but a malicious node with the aim to replace the proposed block broadcasts a different block from the one it got. A malicious node voting for an illegal block is mimicked as voting for an already-existing duplicate block in the chain. Pre-commit(x) is analogous to Prevote(*x*), which is defined by a comparable sequence of two operations. The first procedure, Pre-commit (*x*), finds the first block with a confidence level of at least 2/3 using the votes obtained in the Prevote(x) phase. BroadcastPrecommits ( ) is a similar method to BroadcastPrevotes ( ). PreparePOL(*x*) generates the proof of lock, collects the signatures of validators who voted for this block, and saves them inside the block. Commit(x) adds a pre-committed block to the chain that has at least 2/3 consensus. If the block is not NIL, the node adds it to the chain. This concludes the first round of consensus [14] [15].

4 DEEP NEURAL NETWORKS

Convolutional Neural Networks (CNNs) have shown remarkable success in a variety of fields of computer vision and pattern recognition research, including image classification, object detection, and scene segmentation [16]. Typically, a CNN receives an order of tensors as input. The input is then processed successively. A single processing step is often referred to as a layer, which may be a convolution layer, a pooling layer, a normalization layer, a fully connected layer, or a loss layer, among others.

$$x^1 \rightarrow \boxed{w^1} \rightarrow x^2 \rightarrow .. \rightarrow x^{L-1} \rightarrow \boxed{w^{L-1}} \rightarrow x^L \rightarrow \boxed{w^L} \rightarrow z \quad (5)$$

The above equation explains how a CNN operates in a forward pass, layer by layer. The input x^1 is processed in the first layer, which is represented by the first box. We refer to the parameters involved in the processing of the first layer collectively as a tensor w^1. The first layer's output is x^2, which also serves as the input for the second layer's processing. This procedure continues until all levels in the CNN are complete, at which point x^L is returned. However, an extra layer is included for backward error propagation, a technique for learning optimal parameter values for the CNN.

Assume that the issue at hand is a classification problem using C classes. A frequently used technique is to output x^L as a C-dimensional vector with the prediction encoded in the i-th element (posterior probability of x^1 comes from the i-th class). To convert x^L to a probability mass function, we may configure the (L-1)-th layer's processing as a softmax transformation of x^{L-1}. The output x^L may take on other forms and meanings in different applications. The last layer is a loss layer. Assuming that t is the goal (ground-truth) value for the input x^1, a cost or loss function may be utilized to quantify the disparity between the CNN prediction x^L and the target t. For instance, a simple loss function might be used.

$$z = \frac{1}{2} \parallel t - x^L \parallel^2 \quad (6)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098933, IEEE Access

After learning all of the parameters for a CNN model $w^1 \ldots w^{L-1}$, we are ready to utilize the model for prediction. Prediction requires just that the CNN model be run forward, in the direction indicated by the arrows in Equation 5. Consider the categorization issue. We start with the input x^1 and send it through the first layer's processing (the box with the parameters w^1) to get x^2; x^2 is then transferred to the second layer, and so on. Finally, we get $x^L \in \mathbb{R}^C$, which computes the posterior probability of x^1 being classified as C. We may produce the CNN prediction as:

$$\arg\max_i x_i^L \tag{7}$$

It is worth noting that the loss layer is not required for prediction. It is only beneficial when we are attempting to learn CNN parameters via the use of a training set of examples [17].

5 MARKOV DECISION PROCESS FRAMEWORK

MDP offers a mathematical framework for simulating how a decision-maker would behave in a scenario that is partly controlled by the environment and partly random [18]. Specifically, the decision-maker A_t makes a decision action at time t. At time t +1, the environment feeds back to the maker a new state S_{t+1} and a reward R_{t+1} according to the state S_t and the reward R_t obtained from the environment,(see Fig. 3). The Markov decision process is usually defined by a five tuple $< T.S.A(i).p(\cdot\,|i.a).r(i.a) >$ where,

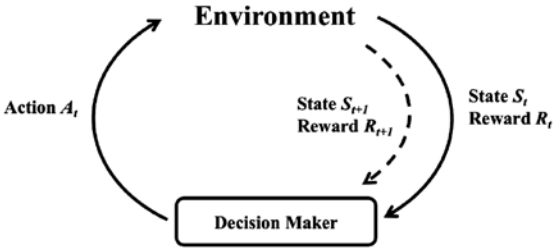

FIGURE 3. **Interaction of Markov decision process [18].**

- T is a collection of decision epochs that might be limited or infinite in length;
- S denotes the state space, and its member i is referred to as the system state.
- $A(i)$ is defined as the action space in state i, and $a \in A(i)$ is defined as the action that is permissible when the system is in state i. At each decision epoch, the chosen action sequence is referred to as a policy, which is a mapping from the state space to the action space.
- $p(\cdot|i,a)$ is the transition probability distribution, and $p(j|i,a)$ is the transition probability function from state i to state j after a is taken;
- $r(i, a) : S \times A(i) \rightarrow R$ is called the reward function, When positive, $r(i, a)$ may be regarded as income, and when negative as cost. In general, the reward received depends on the next state j, when the state of the system at current decision epoch is i and action a is selected.

$$r(i.a) = \sum_{j \in S} r(j|i.d) \cdot p(j|i.a) \tag{8}$$

The assumption in an MDP is that the decision-maker is capable of transiting from any state to any other state in a single step. And MDPs' central problem is to determine a policy for the decision-maker [18].

MDP is a Markov chain extension that includes additional action space and rewards to incentivize decision makers to discover the best method. In contrast, if just one action occurs for each state and all rewards are identical, MDP collapses to a Markov chain. The authors in [18] offered a Markov chain-based approach for studying blockchains, which was used to dissect the consistency of PoW-based systems with an astonishing and precise assertion. However, this strategy evaluates the whole state of the blockchain and disregards interactions between malicious and honest participants. In this work, we want to investigate the security provisions of PoA from the standpoint of an attacker who imitated malicious actions in order to maximize profit using the MDP framework [19].

B. RELATED WORK

This section will discuss many established trustworthy routing solutions for enhancing route security and dependability. Following that, we discuss some relevant methods to blockchain development routing methods. Finally, we investigate existing systems that use MDP in order to make the appropriate decision about message delivery. By and large, all trust models in WSNs fall into two categories: central models and distributed models [20]. The base station or a specialized trustable interface performs the action of aggregating and integrating the trust values of sensor nodes in central trust models. However, in distributed trust architectures, sensor nodes collect trust values on their own. Different approaches, technologies, and procedures for establishing trust have been suggested in WSNs, including fuzzy logic, probabilistic, and deterministic methodologies [20].

The authors of [18] employed fuzzy logic to create a mechanism for evaluating trust in WSNs. The reputation values of nodes are used to calculate the reputation values of pathways in this manner. Then, for packet transmission, the route with the greatest reputation value is chosen. The fuzzy logic-based trust model is considered to be one of the core models; it should be mentioned that central models use a lot of energy. One advantage of fuzzy reasoning is that it is well-suited for very complicated systems whose actions are difficult to deduce. Additionally, the authors in [21] introduced a lightweight, low-energy adaptive hierarchy clustering technique for detecting suspicious node-to-node interactions.

Numerous proposals have been made recently to create a robust spatial routing algorithm for a wireless sensor network that can identify and communicate data about an incident to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098933, IEEE Access

the base station [22]. The authors in [23] designed a safe routing protocol using hierarchical routing algorithms based on numerous criteria such as the distance between nodes and the base station, the distribution density of nodes, and the residual energy of nodes. In [24], the author proposed a secure communication and routing architecture based on the routing protocol's security architecture.

Several academics have recently combined the tamper-proof and traceable characteristics of blockchain technology with routing algorithms in order to improve the stability of routing nodes. De la Rocha et al. [25] presented the trustworthy public key management framework. The method eliminated the need for central authentication and provide a decentralized inter-domain routing network by substituting a blockchain protocol for traditional public key infrastructures. Li et al. [26] created a concurrent, multi-link blockchain-based communications network. The nodes may be classified as malicious or benign, depending on the methodology used to link the interrelated factors and the behavioral features of the blockchain-based data routing nodes. Ramezan et al. [27] developed a blockchain-based contractual routing system for networks with untrusted nodes using smart contracts. The critical principle is that the source node confirms the arrival of each hop routing to the smart contract, and malicious behavior nodes are recorded. The following packets will no longer pass through a malicious node that has been setup. A malicious node equipped with the token's algorithm, on the other hand, may fraudulently report that the packets were received. As a result, there are safety concerns.

As mentioned in [30], several studies have described a signal to noise ratio-based dynamic clustering-based routing system for wireless sensor networks. For the security of routing protocols, the authors used a cluster-based symmetric key cryptography technique. To address the problem in WSN, they created a unique bio-inspired trustworthy routing architecture combining ant colony optimization and Physarumautonomic optimization. The neighbor's conduct was observed with the purpose of assessing trust, and trust-based information was obtained. Another group of academics in [31] published a comprehensive study on the energy-efficient encryption and decoding algorithms for various keys. The introduced mechanism is responsible for encryption and decryption utilizing the DES and RSA algorithms. The quality of channels in wireless sensor networks may be enhanced by encrypting the data using various keys.

As described in [32-36], several authors suggested securing ad hoc on-demand distance vector, a secure routing protocol based on initial encryption that can withstand certain routing assaults while ensuring the integrity and acknowledgement of identification. Other authors introduced an energy-aware secure routing architecture that preserves a trusted environment, isolates misbehaving nodes, and has a minimal control cost. The authors devised an intrusion-tolerant routing system for wireless sensor networks. Although a malicious node may compromise certain nodes in close area, it cannot cause extensive network disruption.

To protect data from eavesdropping assaults, several researchers have suggested a safe multipath routing protocol in sensor networks that use random network coding in directed diffusion routing. Current works addressed the issue of colluding and coordinated black hole attacks and suggested an approach that can be included into the ad hoc on-demand distance vector and secure ad hoc on-demand distance vector protocols. Numerous up-to-date papers have combined multipath routing with a feedback mechanism to identify nodes that lose packets and choose a different path for data transmission, therefore avoiding misbehaving nodes.

The trust and energy-aware routing protocol is a safe routing framework that is based on three distinct frameworks and represents the trust value. The weighted trust and contributed residual energy are two of them, while the hop count is the third. A new trust model is predicted [36] that is built on message trust, data trust, and energy trust. The trustworthiness of data is determined by three factors: trust evaluation, fault tolerance, and data consistency. Energy trust identifies denial-of-service attacks by recognizing the node that spends the most energy in comparison to other nodes.

Recent developments in MDP solvers have enabled the solution of large-scale structures and sparked interest in WSNs in the future. For example, the authors of [28] established a WSN-controlled transmission power-level routing protocol using MDPs. The chosen power source is chosen by determining the best strategy for the MDP setup. The authors in [29] presented past work that examined relay selection in WSNs using an MDP. Additionally to selecting from the explored relay nodes, a transmitting node may choose to continue searching for other relay choices. The node selects the reward to be dispersed to accessible relays throughout the probing process. The states are the finest historical recompense, as well as the recompense for unproven relays in earlier levels. The MDP formulation is then solved using the Bellman equation. Different indicators, such as transmission latency, energy usage, and anticipated network congestion, might be considered while making a choice. For further information, see [28].

To summarize, although the majority of secure protocols provide protection against replay and routing table poisoning attacks, they lack significant protection against black-hole attacks. Current blockchain-based routing systems rely on the proof of work concept to authenticate transactions (packets) in order to handle additional overhead. In comparison to previous protocols, the proposed approach utilizes proof of authority for authentication, which takes less computing time due to its reliance on a small number of key nodes (validators). The novelty here is the utilization of the deep neural network selecting the validators based on their node’s features. These validators are then utilized by MDP for choosing secure path.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098933, IEEE Access

III. THE PROPOSED FRAMEWORK

The primary objective of this suggested method is to build a reliable, trustworthy routing protocol for wireless sensor networks by integrating deep chain and Markov decision-making to provide secured routing. The suggested scheme's basic architecture is illustrated in Fig. 4, and it is composed of three phases: building a node data structure, selecting a validator through a deep learning model, and optimizing the next hop via MDP. Each of these stages is discussed in depth in the following subsections.

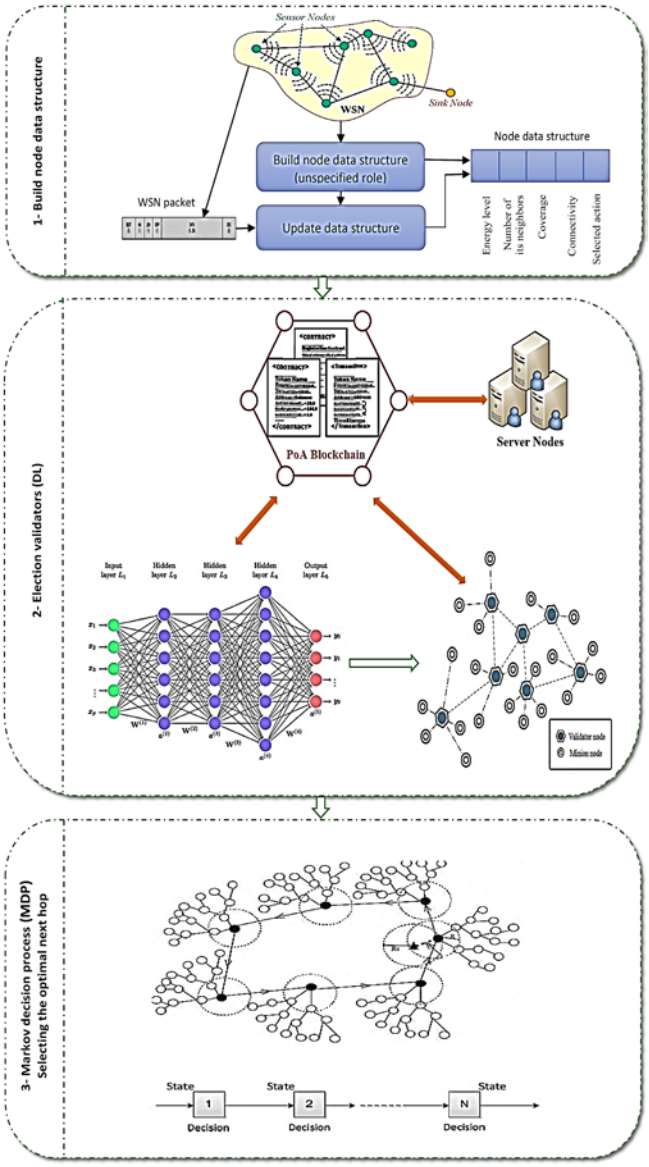

FIGURE 4. **The proposed trusted routing scheme.**

We assume in this paper that the blockchain network is either a trusted routing node or that it rejects packets delivered by other routing nodes. Malicious routing nodes may publish erroneous routing information to the routing network, such as queue length information, thus interfering with the routing scheduling process. Additionally, they may serve as black hole attack nodes, refusing to forward data. However, we exclude collusion attacks between two routing nodes in order to execute incorrect blockchain transactions. Additionally, we believe that a routing node may function solely as a normal or malicious node, implying that attacks are far from intermittent. Meanwhile, we disregard the sporadic aberrant behavior produced by the node's performance (e.g., a node does not send a message in time or loses the wireless spectrum). Herein, the server nodes are often static, while the routing nodes may be dynamic. However, the entrance and departure of nodes have no effect on our scheme, since our blockchain-based system's status information is likewise constantly updated.

A. STEP 1: Build Node Data Structure

At first, all sensors operate in the same manner and serve no use as validators or slave nodes. They are not anonymous sensors; they have a unique identification (e.g., anonymous addresses). Each packet in a transmission is the same size. There are two types of data transmission in a wireless sensor network: direct transmission and multi-hop data transfer. Multi-hop data transfer is utilized in this instance. With symmetrical communication, each cell in the WSN starts with the same amount of energy and remains static. During initialization, the function of any node that is originally set to unstated is converted into the validator or minion. Each node in the network maintains a data structure including a variety of information on the node property, such as the chosen action (validator or not), the energy level, the coverage, the connectivity, and the number of its neighbors, as shown in Fig. 5. For more information, see [37].

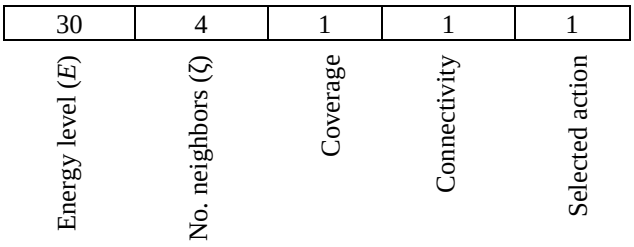

FIGURE 5. **Filled node's data structure.**

B. STEP 2: Validators Election using Deep Neural Learning

After establishing the data structure for each node, the characteristics of these nodes are utilized to determine the most significant nodes that will act as validators in the blockchain proof framework's authentication network. A deep neural network is used to make the selection. Deep learning techniques are used to learn functional hierarchies, in which features are constructed on higher levels using minor levels. The activation potentials supplied by each of the first hidden layer's unique input measurements are utilized to choose the most appropriate functions. The features are selected to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098933, IEEE Access

provide more accurate classifications than the high-dimensional initial characteristics. The stacked RBMS (Deep Belief Network) is used as a BlackBox with its default settings in this paper. For more information, see [16] [38] [39].

Consider a sample space Z of the type $X \times Y$ and an ordered training set $S = ((x_i, y_i))_{i=1}^{m}$, where $x_i \in X$ represents the data and $y_i \in Y$ represents the associated label. Assume that *H* is a hypothesis space (e.g., a particular neural network architecture parameterized by a weight vector *w*). If the network calculates a function from *X* to *Y*, we shall abbreviate it as f_w; for example, $f_w(x) = y$. There is a loss (or risk) function $\ell : H \times Z \dashrightarrow \mathbb{R}$ such that we may get a loss $\ell(w.(x,y))$ given a hypothesis w and a sample$(x,y) \in Z$. Consider the situation in which we want to reduce the average loss over the training set *S*.

$$L_s(w) = \frac{1}{m} \sum_{i=1}^{m} \ell(w.(x_i . y_i)) + \lambda \mathcal{R}(w) \qquad (9)$$

In the above equation, $\lambda > 0$ and the term $\mathcal{R}(w)$ is referred to as the regularizer; the latter attempts to impose a concept of "simplicity" on *w*. Due to the fact that *S* is constant, we may represent $\ell_i(w) = \ell(w.(x_i, y_i))$ as a function of just *w*. The training issue is to discover a w that minimizes $L_s(w)$; in other words, we want to solve the optimization problem described below.

$$\min_{w \in H} L_s(w) \qquad (10)$$

This optimization problem is also sometimes termed empirical risk minimization. After completing this step, the selected nodes will be determined which will be used in blockchain-based routing network. These selected nodes will be acted as validator and routing nodes.

C. STEP 3: Blockchain-Based Routing Networks

To increase the trustworthiness and robustness of routing information, we integrate the blockchain, which is essentially a distributed ledger with tamper-proof, decentralization, and information traceability characteristics, into the wireless sensor network and use blockchain token transactions to record node-related information [39]. The primary structure is composed of two components: the routing network itself and the blockchain network. Packets are sent from the source terminal to the destination terminal through a routing node, which then chooses the next-hop routing node based on the routing policy received from MDPs. The MDP requests and gathers pertinent routing network status information from the blockchain network on a continuous basis. The packets will be sent to the target routing node and subsequently to the destination terminal after continuous transmission. Each blockchain system uses a unique consensus method to guarantee the transaction's fairness. We picked the PoA consensus method for our blockchain network because it is more efficient at processing transactions. Two distinct types of entities are specified in our concept for the PoA-based blockchain network.

- Validator: validators are pre-authenticated nodes on the blockchain that have advanced authorization and are in charge of the PoA Blockchain's verification job. Their particular responsibilities include the execution of smart contracts, the verification of blockchain transactions, and the release of blockchain blocks. As described in step 2, a new validator may be introduced via the election of verified validators through a deep belief network. Even if a malicious validator exists, it is limited to attacking one of the contiguous blocks, at which time the malicious validator may be thrown out by other validator votes.
- Minion: minions are less-privileged nodes that are unable to conduct verification work in the PoA blockchain as validators. Each routing node in our system is likewise a minion with less privileges on the PoA blockchain, and it also has a unique blockchain address. They may start token contracts, activate certain contract functionalities, and access the blockchain for transaction details.

On the blockchain network, we utilize various blockchain tokens to represent the various packets that need to be sent to target nodes, with *n* unit tokens representing n unit related packets. The essence of a token is that it is a representation of the digitized data included in the smart contract's associated packets. Token contracts may be initiated by routing nodes to create tokens and map the state information of associated packets. They will exchange tokens through the token contract in order to transfer tokens depending on the transmitted and received packets. The consensus method between server nodes prevents malicious nodes from revising the token transactions arbitrarily; to some degree, the token properly reflects the packet transmitted between the routing nodes.

After joining the blockchain-based routing network, each routing node is registered on the registration contract. When the routing node gets data from its offspring, it forwards the packets and drops the data. However, in the case of the Blockchain's next-hop routing node. They must then validate the routing information on the blockchain, which includes the address of the next-hop routing node, the amount of packets delivered to the next node, and the timestamp. The routing information is then verified and updated on the blockchain by the server nodes through the blockchain consensus process. The proposed approach is consistent with the idea described in the article [1] about the implementation of a blockchain-based routing network.

D. STEP 4: Next Hope Selection using MDPS

MDP is used to determine the optimum strategy for maximizing a value function, which is defined as the expected sum of rewards at all decision epochs in finite horizon issues, or as the anticipated total discounted reward or the expected average reward in infinite horizon problems [40]. When using MDP theory to opportunistic routing, the following issues must be considered: how the state is defined and how the choice is made. In general, the process of packet

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098933, IEEE Access

forwarding from one node to another may be thought of as a state change. Due to the fact that the packet must reach the target node in the fewest feasible hops, we examine only the finite horizon scenario; therefore, the set of decision epochs is represented by $T = \{0, 1, 2,..., M\}$. $S = \{1, 2,..., N\}$. N is the state space, with system state i defined as the ID of the node to which the packet belongs at a decision epoch t. A packet produced by the source node is sent to the destination node through many hops, which implies that the initial state (any node in the network) passes through several stages to reach the termination state, which in this case an absorption state is matching to the destination node.

Following that, we examine what actions are possible when the system's state at decision epoch t is i. In opportunistic routing, suitable candidate forwarders should be chosen from among neighbors and prioritized in the sender's perspective. However, from the receiver's perspective (the candidate nodes that received the packet), a coordinate mechanism is required to determine whether or not to transmit the packet in response to other nodes' replies. The article makes the assumption that a flawless coordination mechanism is utilized between the candidate nodes, i.e. that packets are sent in exact accordance with the candidate nodes' priorities. As a result, we examine just the former choice scenario, in which the accessible action space consists of all potential ordered subsets of the sender's neighbor node set.

Define as $F_i = \{i_1, i_2, \ldots, i_{|Fi|}\}$ as the forwarder set of node i, then the adjacency matrix $F = (f_{ij})_{N\times N}$ is represented the forwarders set of all nodes, if $j \in F_i$, $f_{ij} = 1$, otherwise, $f_{ij} = 0$. Accordingly, assume that $\omega_i = \{\omega_{ii_1}.\omega_{ii_2}.\cdots.\omega_{ii_{|F_i|}}\}$ is one ordered set of F_i, where $\omega_{ij} > \omega_{ik}$ means node j has the higher priority than node k to forward the packet from node i. Therefore, $W = (\omega_{ij})_{N\times N}$ is defined as the priority matrix. We assign the priority to every neighbor node so that the assignment of priority does not depend on the size of forwarder set $|F_i|$. When the packet is located at node i, an appropriate action $a(i) = (F_i, \omega_i)$ is selected for node i to broadcast the packet, where the action $a(i)$ is only related to the state i and not affected by the epoch t. For the sink node N, the only decision can be made is terminating the packet forwarding, i.e. $a(N) = F_N = \emptyset$.

Let $s = l_0, l_1, ..., l_M = N$ be the node sequence of packet delivered from source node s to sink node N, thus $\pi = (a(l_0), a(l_1), ..., a(l_M))$ is called a forwarding strategy for opportunistic routing. Because of the randomness of forwarding nodes and relay hops in opportunistic routing, we use (F,W) to redefine the data forwarding strategy from a global perspective, which takes into account of all the possible strategy π. In other words, no matter which nodes the packet pass traveled before and which node the packet locates at now, the candidate forwarders of next hop and their priorities can be determined only according to the global strategy (F,W). Given a forwarding strategy (F,W), the whole stochastic process of the packet generated from the source node to the destination node is specified, which can be transferred into a Markov chain with one absorbing state. According to the basic forwarding rules of opportunistic routing, the candidate node with the highest priority is responsible for forwarding the packet if it receives the packet successfully. If not, the candidate node with the secondary priority will take over forwarding the packet, and so on. The packet will be retransmitted when all the candidate nodes fail to receive the packet. Therefore, in opportunistic routing the probability of a packet delivered from node i to node j is calculated as follows,

$$q_{ij} = \begin{cases} p_{ij}. & j \in F_i \text{ and } w_{ij} = \max w_i \\ p_{ij}\prod_{k\in F_i}(1-p_{ik}). & j \in F_i \text{ and } w_{ij} < w_{ik} \\ \prod_{k\in F_i}(1-p_{ik}). & j = i \\ 0. & otherwise \end{cases} \quad (12)$$

where p_{ij} is the packet delivery probability when node i transmits packets to node j successfully.

As the state in MDP corresponds to the node's ID which the packet locates at, the transition probability between states is affected by the current state and the actions taken. In the finite horizon Markov decision process, it is necessary to ensure that the packets arrive at the destination node within a limited number of hops and to avoid the unrestricted retransmission at one node, thus the transition probability between any two system states is defined as follows,

$$p\big(j|i.(F.W)\big) = \begin{cases} \frac{p_{ij}}{1-\prod_{l\in F_i}(1-p_{il})}, & j \in F_i \text{ and } \omega_{ij} = \max \omega_i \\ \frac{p_{ij}\prod_{k\in F_i}(1-p_{ik})}{1-\prod_{l\in F_i}(1-p_{il})}, & j \in F_i \text{ and } \omega_{ij} = \max \omega_i \\ 0. & otherwise \end{cases} \quad (13)$$

where $1-\prod_{l\in F_i}(1-p_{il})$ denotes the probability that at least one candidate node in the forwarder set F_i receives the packet successfully. Therefore, the state transition probability $p\ (j|i,\ (F,W))$ is a conditional probability based this event, and satisfies the constraint

$$\sum_{j\in S} p(j|i.(F.W)) = 1. \quad (14)$$

In this paper, the decision cost is considered as the number of packet transmission from the current sender to the destination node under a forwarding strategy. The expected transmission count (ETX) is a metric of single hop link between two nodes. In opportunistic routing, ETX denotes the expected value of the total number of transmissions for successfully transmitting a packet, and the routing path which has a smaller ETX will induce a lower number of retransmission. In ExOR [1], ETX is calculated based on Dijkstra algorithm, however, the best candidate nodes may not be found by that method. In order to better reflect the random characteristics of packet forwarding, we refer to the expected any-path transmissions (EAX) [11] which is a modification of ETX. Given the global forwarding strategy

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098933, IEEE Access

(F,W), the expected any-path transmissions for node i is defined as follows,

$$\mathrm{EAX}_i(F.W) = 1/[1-\prod_{l\in F_i}(1-p_{il})] + \sum_{j\in S} p(j|i.(F.W))\mathrm{EAX}_j(F.W) \quad (15)$$

That is, the EAX of node i to the sink node is consisted of two parts: one is the expected retransmission count of single hop from node i to its candidate forwarders set F_i, another is the expected sum value of the EAX from all the nodes in F_i to the sink. In opportunistic routing, we desire to find an optimal forwarding strategy (F^*,W^*), thus each node maintains a routing table to forward data. When the EAX of each node is minimized, the performance of the whole network is optimized. We define the negative value of the expected retransmission count for node i as the immediate reward obtained by the system state i when take the action $a(i)$ at epoch t.

$$r_t\big(i.a(t)\big) = -\frac{1}{1-\prod_{l\in F_i}(1-p_{il})} \quad (16)$$

Accordingly, the value function V (.) which should be maximized in the finite horizon problem is defined as follows,

$$\begin{aligned} V(s) &= \mathrm{E}[\textstyle\sum_{k=0}^{M} r_k(l_k.a(l_k))]. \\ &= r_0\big(l_0.a(l_0)\big) + \mathrm{E}[\textstyle\sum_{k=1}^{M} r_k(l_k.a(l_k))]. \\ &= r\big(s.a(s)\big) + \textstyle\sum_{j\in S} V(j)\cdot p(j|s.a). \\ &= -\mathrm{EAX}_s(F.W) \end{aligned} \quad (17)$$

The key question of WSN routing is how best to find the next step in every hop. As mentioned in the literature, the key impacts on next-hop decision-taking involve trust, congestion probability and distance to the target". Readers looking for more information regarding how to compute these factors can refer to [39]. The optimal next step is a standard decision-making mechanism focused on the current circumstances, and we are implementing MDPs to address the issue as it is one of the better choices for a random dynamical system. Any hop on the route can be seen as a state; each hop is determined to pick one of the next best hops.

The decision-making in each stage relies on the current scenario, and the entire routing method is efficient in chain decisions. Because hops are not infinite from source to destination, we follow a final Markov decision to solve it. The simple principle is that to find a sequence of better hops by candidates; we must use optimal decision metrics in the routing process as a criterion for the decisions to construct a Finite Markov Decision control system. As the network of the wireless sensor is a global network, central computation does not appeal to accomplish one path. Every node is, therefore, responsible for measuring and making decisions in any hop. Thereby, we find the decision of next-hop as a one-phase decision-making process; the purpose of the decision is to optimize the reward for each move.

IV. EXPERIMENTAL RESULTS

We constructed a prototype and compared its performance to that of existing state-of-the-art reinforcement learning-based routing algorithms, the trust-based algorithm, and the blockchain-based algorithm. We compared our system to a standard PoW-based blockchain system to determine our system's performance in terms of latency, consumption, and throughput. We created a PoA consortium blockchain and simulated a single server that would update the chain's transactions. The MDPs may receive all of the routing information they needed from public blockchain transactions. The consortium blockchain was developed using Solidity 0.8.4 to ensure the integrity of Ethereum transactions. We used the blockchain-based routing algorithm as a performance test [1]. To replicate actual packet arrival rates, we use the same setup as in [1], with 32 terminals in a 16 × 16 matrix randomly broadcasting packets to the destination point using a Poisson distribution with one packet per slot.

Additionally, we simulated 16 ×16 routing nodes that were capable of receiving and delivering actual packets in a maximum of one packet/slot depending on the routing strategy given by the MDPs model. Finally, the experiment collected data on average packet latency, transaction latency, and energy usage. In the experiments, there were 25% and 50% malicious nodes in the 16 ×16 routing nodes. The malicious nodes attempted to manufacture fake queue length information and use the BP algorithm weakness to cheat more packets or function as a black hole node and broadcast no packets. The server node equipment are configured as follows: CPU 2.6 GHz, RAM 16 GB, Storage 1 TB, Network 1000 Mb, OS Ubuntu Server 19.04. Whereas the sensor node devices' detailed configurations are as follows: CPU 1.2 GHz, RAM 1 GB, Storage 16 TB, Network 100 Mb, OS TinyOS Alliance 2.1.2.

A. EXPERIMENT 1: COMPARATIVE ANALYSIS-ROUTING WITH MALICIOUS NODES

To understand whether malicious nodes may alter the routing scheduling algorithms, we ran an experiment that compared the Trust-based backpressure algorithm (TB-BP), the Q-Learning backpressure algorithm (QL-BP), and the Reinforcement learning and blockchain-based algorithm (RLBC) to our system. For further information on the comparative methodologies, see [1]. The comparison studies shown that our method outperformed TB-BP, QL-BP, RLBC, and RLBC in the malicious routing environment as a function of packet arrival rate and average latency. As seen in Fig. 6, our technique outperforms the TB-BP method in a routing environment with 25% malicious nodes, saving about 74% of the time when compared to the TB-BP method, 58% when compared to the QL-BP methodology, and 21% when compared to the RLBC methodology. Additionally, we conducted comparative experiments in a routing environment with 50% malicious routing nodes (see Fig. 7) and discovered that it reduces delay by approximately 82 % when compared

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098933, IEEE Access

to the TB-BP algorithm, 66 % when compared to the QL-BP algorithm, and 28% when compared to the RLBC algorithm.

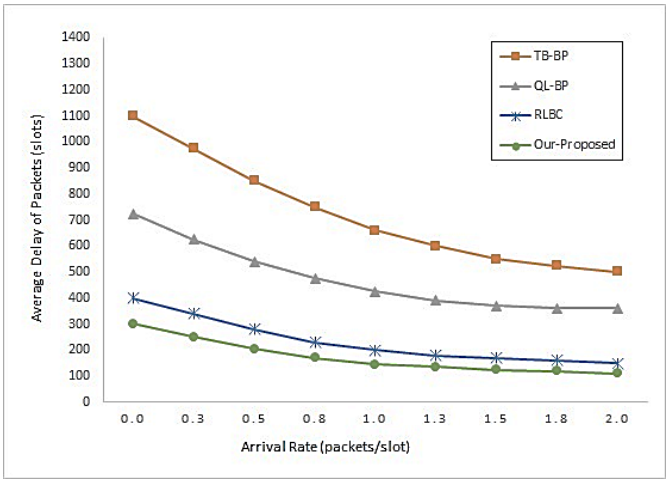

FIGURE 6. **Average delay of packets with 25% malicious nodes.**

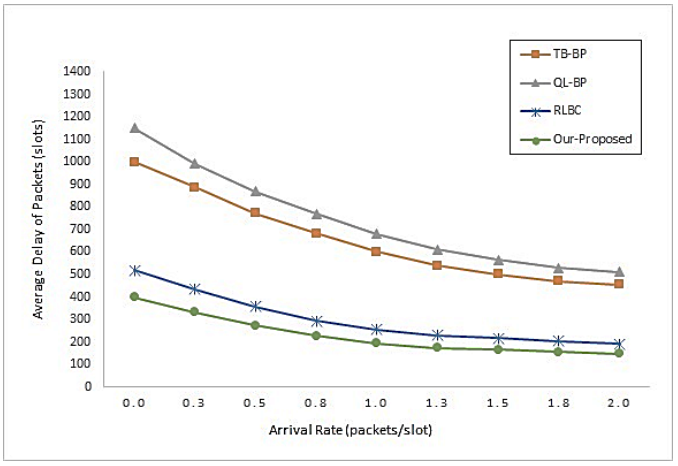

FIGURE 7. **Average delay of packets with 50% malicious nodes.**

The experimental results demonstrate that our technique is not sensitive to malicious node impact in terms of average packet delay, and its efficacy demonstrates that it is conceivable to utilize it to enhance the routing algorithm's performance. While both the proposed system and the RLBC algorithm rely on the blockchain network to determine trust nodes and are based on the PoA algorithm, the comparative system identifies validators randomly, in contrast to the proposed system, which selects validators using deep learning (PoA-DL), which has the effect of determining the best trust nodes for paths that are not exposed to at least one attack.

B. EXPERIMENT 2: SYSTEM EFFICIENCY WITH POA-DL.

In the second set of experiments, we compared our blockchain system based on the PoA-DL consensus mechanism, which employs deep learning to determine validators, to a traditional PoA-based blockchain system, which employs a random selection of validators, and to a traditional PoW-based blockchain system. Throughout the investigation, we captured experimental data such as transaction delay and throughput. We used transaction packaging time as a proxy for average token transaction delay. We measured the latency of token transactions on PoA-DL, PoA, and PoW blockchain systems as the arrival rate increased. The results of the experiment are shown in Fig. 8.

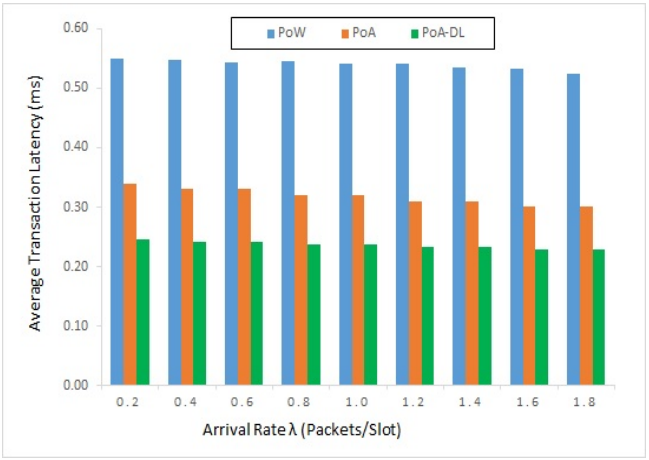

FIGURE 8. **Average transaction latency for both PoA and PoW-based blockchain systems.**

As can be seen, the transaction's latency is pretty steady and does not vary much with the arrival rate. Our PoA-DL blockchain system had an average transaction latency of roughly 0.24 milliseconds, whereas the PoA blockchain system had an average transaction delay of roughly 0.32 milliseconds. Whereas the PoW blockchain technology has a latency of roughly 0.55 ms. Results indicate that our blockchain system, which is based on the PoA-DL consensus mechanism, can reduce around 25% and 56% of transaction delay, respectively, when compared to PoA and PoW. Such a delay in token transactions is acceptable since it has a negligible effect on the routing schedule. It is both feasible and efficient to gather and maintain routing scheduling information using our PoA-DL blockchain solution. The proposed approach is efficient since the most secure nodes are chosen by applying deep learning methods to choose the best validators. Later, these nodes will be utilized by MDP to determine the optimal routing route; since there is no risk of assaulting these nodes, transaction latency will be decreased.

C. EXPERIMENT 3: TOKEN TRANSACTION THROUGHPUT WITH POA-DL.

The final set of experiments validated the proposed trusted routing scheme's efficiency in terms of token transaction throughput. The throughput of token transactions demonstrates the blockchain system's capacity to manage concurrent token transactions. The results in Fig. 10 demonstrate that as the rate of synchronous requests grows, the token transaction throughput climbs steadily, and the curve gradually flattens out as the throughput reaches its peak. The token transaction throughput of our blockchain system using the PoA-DL consensus mechanism is stable at 3630

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098933, IEEE Access

concurrent requests per second, while the symbolic transaction throughput of the RLBC comparative system using the PoA consensus mechanism is stable at 3,300 concurrent requests per second, and the classic blockchain system using the PoW consensus mechanism is only stable at around 1,500 concurrent requests per second. The experimental results demonstrate that the PoA-DL-based method has a more efficient transaction processing capacity while dealing with concurrent searches due to the restricted number of validators. It is appropriate and legitimate to use the PoA-DL algorithm as the blockchain system's consensus mechanism. This PoA-DL blockchain-based routing scheduling technique is capable of successfully coping with the routing environment's high concurrent request volume.

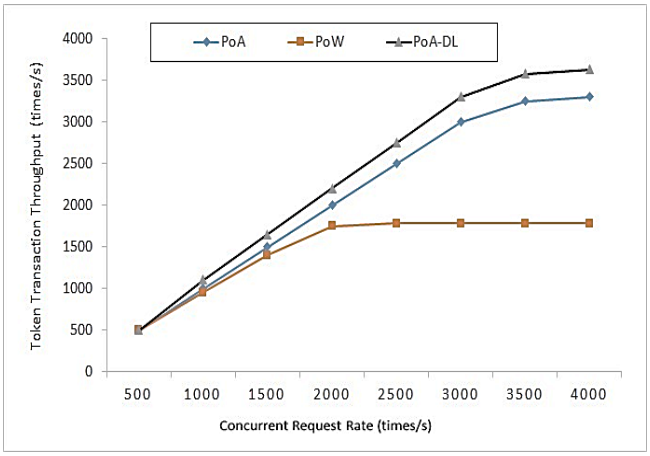

FIGURE 9. **Throughput of transaction token for both PoA-DL, PoA and PoW-based blockchain systems.**

Due to the fact that the proposed model uses MDP for routing rather than reinforcement learning as in [1], the solution of an MDP model, referred to as a policy, may be implemented using a routing lookup table. This table may be easily saved in the sensor node's memory for online operations. As a result, the MDP model can be applied to even the smallest and most resource-constrained nodes without requiring excessive computation. Additionally, near-optimal solutions may be constructed to approach optimum decision policies, allowing for the creation of WSN algorithms that are less computationally intensive [28]. The reinforcement learning-based routing, on the other hand, is based on modifying the weight matrix to attain the required performance. In general, constructing an ideal weight matrix is a difficult task that is often solved by trial and error. To summarize, using MPD for routing improves the model's transaction throughput [18].

V. CONCLUSIONS AND FUTURE WORK

In this study, we offered a trusted routing method that improves the performance of the routing network by combining deep-chain and Markov decision processes. We employ the blockchain token to represent the routing packets, and each routing transaction is confirmed by validator nodes before being distributed to the blockchain network. By making each routing transaction tracker traceable and tamper-resistant, routing nodes will be able to monitor dynamic and trustworthy routing information on the blockchain network. Additionally, we design the MDP model in order to ensure fast route discovery and to avoid routing links to hostile nodes. Our test results indicate that our schema is capable of readily removing hostile node attacks, and the device's latency is exceptional. In the future, we want to utilize our technique to test the efficacy and portability of other route scheduling techniques than the backpressure technique. Additionally, we want to add data validation technologies based on the blockchain.

REFERENCES

[1] J. Yang, S. He, Y. Xu, L. Chen, and J. Ren, "A trusted routing scheme using blockchain and reinforcement learning for wireless sensor networks", Sensors, vol. 19, no. 4, pp. 1-19, 2019.

[2] Z. Jiao, B. Zhang, C. Li, and H. T. Mouftah, "Backpressure-based routing and scheduling protocols for wireless multihop networks-a survey", IEEE Wireless Communications, vol. 23, no. 1, pp. 102-110, 2016.

[3] F. Ahmed, and Y.-Bae Ko, "Mitigation of black hole attacks in routing protocol for low power and lossy networks", Security and Communication Networks, vol. 9, no. 18, pp. 5143-5154, 2016.

[4] F. Abdel-Fattah, K. A. Farhan, F. H. Al-Tarawneh, and F. AlTamimi, "Security challenges and attacks in dynamic mobile ad hoc networks MANETs", in Proc. IEEE Conf. (JEEIT), Jordan, pp. 28-33, 2019.

[5] A. Angrish, B. Craver, M. Hasan, and B. Starly, "A case study for blockchain in manufacturing "FabRec" A prototype for peer-to-peer network of manufacturing nodes", Procedia Manufacturing, vol. 26, no. 1, pp. 1180-1192, 2018.

[6] L. M. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms", in Proc. IEEE Conf. (MIPRO), Croatia, pp. 1545-1550, 2018.

[7] E. K. Wang, Z. Nie, Z. Du, and Y. Ye, "MDPRP: Markov decision process based routing protocol for mobile WSNs", in Proc. Springer Conf. on Geo-Informatics in Resource Management and Sustainable Ecosystem, Singapore, pp. 91–99, 2016.

[8] Y. Lv, Y. Liu, and J. Hua, "A study on the application of WSN positioning technology to unattended areas" IEEE Access, vol. 7, pp. 38085-38099, Mar. 2019.

[9] A. Ahmed, K. Abu Bakar, M. Ibrahim Channa, K. Haseeb and A. W. Khan, " TERP A trust and energy aware routing protocol for wireless sensor network", IEEE Sensors Journal, vol. 15, no. 12, pp. 6962-6972, Aug. 2015.

[10] D. Zhang, G. Li, K. Zheng, X. Ming, and Z.-H. Pan, "An energy-balanced routing method based on forward-aware factor for wireless sensor networks", IEEE Transactions on Industrial Informatics, vol. 10, no. 1, Feb. 2014.

[11] L. Tang, Z. Lu and B. Fan, "Energy efficient and reliable routing algorithm for wireless sensors networks", Applied Sciences, vol. 10, no. 5, pp. 1 - 16, 2020.

[12] J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: Analysis and applications" in Proc. Annual inter. Springer Conf. on the theory and applications of cryptographic techniques, Germany, pp. 281-310, 2015.

[13] W. Cai1, Z. Wang, J. B. Ernst, Z. Hong, C. Feng, and V. Leung, "Decentralized applications the blockchain-empowered software system", IEEE Access, vol. 6, pp. 53019-53033, 2018.

[14] W. Thin, N. Dong, G. Bai, and J. S. Dong "Formal analysis of a proof-of-stake blockchain", in Proc. IEEE Conf. (ICECCS), Australia, pp. 197-200, 2018.

[15] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and Chen Qijun, "A review on consensus algorithm of blockchain", in Proc. IEEE Conf. (SMC), Canada, 2017, pp. 2567-2572.

[16] S. A. Seshia, A. Desai, T. Dreossi, D. J. Fremont, S. Ghosh, E. Kim, S. Shivakumar, M. V. Chanlatte, and X. Yue, "Formal specification for deep neural networks" In Proc. Springer Inter. Conf. Symposium on Automated Technology for Verification and Analysis, Cham, pp. 20-34, 2018.

[17] T. Dreossi, A. Donzé, and S. A. Seshia, "Compositional falsification of cyber-physical systems with machine learning components", Journal of Automated Reasoning, vol. 63, no. 4, pp. 1031-1053, 2019.

[18] X. Liu, G. Zhao, X. Wang, Y. Lin, Z. Zhou, H. Tang, and B. Chen, "MDP-based quantitative analysis framework for proof of authority", in Proc. Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Guilin, China, 2019, pp. 227-236.

[19] L. Kiffer, R. Rajaraman, and a. shelat, "A better method to analyze blockchain consistency", in Proc. ACM SIGSAC Conf. Computer and Communications Security, Toronto, Canada, 2018, pp. 729–744.

[20] A. Beheshtiasl, and A. Ghaffari, "Secure and trust-aware routing scheme in wireless sensor networks", Wireless Personal Communications, vol. 107, no. 4, pp. 1799-1814, 2019.

[21] Y. Arfat, and R. A. Shaikh, "A survey on secure routing protocols in wireless sensor networks", Inter. Journal of Wireless and Microwave Technologies (IJWMT), vol 6, no. 3, pp. 9-19, 2016.

[22] Y. Wang, Z. Ye, P. Wan, and J. Zhao, "A survey of dynamic spectrum allocation based on reinforcement learning algorithms in cognitive radio networks", Artificial Intelligence Review, vol. 51, no. 3, pp. 493-506, 2019.

[23] C. Deepa, and B. Latha, "HHCS: Hybrid hierarchical cluster based secure routing protocol for Wireless Sensor Networks", in Proc. Inter. Conf. on Information Communication and Embedded Systems (ICICES2014), India, pp. 1-6, 2014.

[24] F. Khan, "Secure communication and routing architecture in wireless sensor networks", in Proc. IEEE Inter. Conf. on Consumer Electronics (GCCE), Japan, pp. 647-650, 2014.

[25] A. R. G-Arevalillo, and P. Papadimitratos, "Blockchain-based public key infrastructure for inter-domain secure routing." In Proc. Inter. Workshop on Open Problems in Network Security (iNetSec), Sweden, pp. 20-38, 2017.

[26] J. Li, G. Liang, and T Liu, "A novel multi-link integrated factor algorithm considering node trust degree for blockchain-based communication", KSII Transactions on Internet and Information Systems, vol. 11, no. 8, pp. 3766- 3788, 2017.

[27] G. Ramezan, and C. Leung, "A blockchain-based contractual routing protocol for the internet of things using smart contracts", Hindawi, Wireless Communications and Mobile Computing, vol. 2018, pp. 1- 15, 2018.

[28] M. Abu Alsheikh, D. T. Hoang, D. Niyato, H-P. Tan, and S. Lin, "Markov decision processes with applications in wireless sensor networks: A survey." IEEE Communications Surveys and Tutorials, vol. 17, no. 3, pp. 1239-1267, 2015.

[29] W. Rehan, S. Fischer, M. Rehan, and M. Rehmani, "A comprehensive survey on multichannel routing in wireless sensor networks", Journal of Network and Computer Applications, vol. 95, pp. 1- 25, 2017.

[30] T. Kalidoss, L. Rajasekaran, K. Kanagasabai, G. Sannasi, and A. Kannan,"Qos aware trust based routing algorithm for wireless sensor networks", Wireless Personal Communications, vol. 110, no. 4, pp. 1637-1658, 2020.

[31] R. S. Raghav, K. Thirugnansambandam, and D. K. Anguraj, "Beeware routing scheme for detecting network layer attacks in wireless sensor networks", Wireless Personal Communications, vol. 112, no. 4, pp. 2439-2459, 2020.

[32] S. Taterh, Y. Meena, and G. Paliwal, "Performance analysis of ad hoc on-demand distance vector routing protocol for mobile ad hoc networks", In Computational Network Application Tools for Performance Management, pp. 235-245. Springer, Singapore, 2020.

[33] A. Ahmed, K. Abu Bakar, M. I. Channa, A. W. Khan, and K. Haseeb, "Energy-aware and secure routing with trust for disaster response wireless sensor network", Peer-to-Peer Networking and Applications, vol. 10, no. 1, pp. 216-237, 2017.

[34] S. Sharma, A. V. Singh, and V. Dattana, "A survey of IoT routing protocols based on security and trust management", In Proc. IEEE Inter. Conf. on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, pp. 623-629, 2020.

[35] M. Boulaiche, "Survey of secure routing protocols for wireless ad hoc networks", Wireless Personal Communications, vol. 114, no. 1, pp. 483-517, 2020.

[36] S. Prabhu, and M. Anita.E.A, "Trust based secure routing mechanisms for wireless sensor networks: A survey", In Proc. IEEE Inter. Conf. on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, pp. 1003-1009, 2020.

[37] S. M. Darwish, M. N. El-Dirini, and I. A. Abd El-Moghith, "An adaptive cellular automata scheme for diagnosis of fault tolerance and connectivity preserving in wireless sensor networks", Alexandria Engineering Journal, vol. 57, no. 4, pp. 4267-4275, 2018.

[38] N. R. Sabar, A. Turky, A. Song, and A. Sattar, "An evolutionary hyper-heuristic to optimise deep belief networks for image reconstruction", Applied Soft Computing vol. 97, part B, pp. 1-24, 2020.

[39] I. A. Abd El-Moghith, and S. M. Darwish, "A deep blockchain-based trusted routing scheme for wireless sensor networks" In Proc. Springer Inter. Conf. on Advanced Intelligent Systems and Informatics, China, pp. 282-291, 2020.

[40] J. Hao, X. Jia, Z. Han, B. Yang, and D. Peng, "Design of opportunistic routing based on markov decision process", in Proc. IEEE Conf. Chinese Control Conference (CCC), China, pp. 8976-8981, 2017.

SAAD M. DARWISH received the B.Sc. degree in statistics and computer science from the Faculty of Science, Alexandria University, Egypt, in 1995, the M.Sc. degree in information technology from the Department of Information Technology, Institute of Graduate Studies and Research (IGSR), University of Alexandria, in 2002, and the Ph.D. degree from Alexandria University, for a thesis in image mining and image description technologies. Since June 2017, he has been a Professor with the Department of Information Technology, IGSR. He has supervised around 60 M.Sc. and Ph.D. Students. He is the author or the coauthor of more than 50 articles publications in prestigious journals and top international conferences and received several citations. His research interests include image processing, optimization techniques, security technologies, database management, machine learning, biometrics, digital forensics, and bioinformatics. He has served as a reviewer for several international journals and conferences.