electronics

MDPI

Article

A Novel Blockchain Approach for Improving the Security and Reliability of Wireless Sensor Networks Using Jellyfish Search Optimizer

Viyyapu Lokeshwari Vinya [1], Yarlagadda Anuradha [2], Hamid Reza Karimi [3,*], Parameshachari Bidare Divakarachari [4] and Venkatramulu Sunkari [5]

[1] Department of Computer Science and Engineering, Vardhaman College of Engineering, Shamshabad, Hyderabad 501218, India
[2] Department of Computer Science and Engineering, Gayatri Vidhya Parishad College of Engineering (Autonomous), Madhurawada, Visakhapatnam 530048, India
[3] Department of Mechanical Engineering, Politecnico di Milano, 20156 Milan, Italy
[4] Department of Electronics and Communication Engineering, Nitte Meenakshi Institute of Technology, Bengaluru 560064, India
[5] Department of Computer Science and Engineering, Kakatiya Institute of Technology and Science, Warangal 506015, India
* Correspondence: hamidreza.karimi@polimi.it

Abstract: For the past few years, centralized decision-making is being used for malicious node identification in wireless sensor networks (WSNs). Generally, WSN is the primary technology used to support operations, and security issues are becoming progressively worse. In order to detect malicious nodes in WSN, a blockchain-routing- and trust-model-based jellyfish search optimizer (BCR-TM-JSO) is created. Additionally, it provides the complete trust-model architecture before creating the blockchain data structure that is used to identify malicious nodes. For further analysis, sensor nodes in a WSN collect environmental data and communicate them to the cluster heads (CHs). JSO is created to address this issue by replacing CHs with regular nodes based on the maximum remaining energy, degree, and closeness to base station. Moreover, the Rivest–Shamir–Adleman (RSA) mechanism provides an asymmetric key, which is exploited for securing data transmission. The simulation outcomes show that the proposed BCR-TM-JSO model is capable of identifying malicious nodes in WSNs. Furthermore, the proposed BCR-TM-JSO method outperformed the conventional blockchain-based secure routing and trust management (BSRTM) and distance degree residual-energy-based low-energy adaptive clustering hierarchy (DDR-LEACH), in terms of throughput (5.89 Mbps), residual energy (0.079 J), and packet-delivery ratio (89.29%).

Keywords: block chain; jellyfish search optimizer; routing; security; trust management; wireless sensor network

Citation: Vinya, V.L.; Anuradha, Y.; Karimi, H.R.; Divakarachari, P.B.; Sunkari, V. A Novel Blockchain Approach for Improving the Security and Reliability of Wireless Sensor Networks Using Jellyfish Search Optimizer. *Electronics* **2022**, *11*, 3449. https://doi.org/10.3390/electronics11213449

Academic Editor: HungYu Chien

Received: 3 October 2022
Accepted: 22 October 2022
Published: 25 October 2022

1. Introduction

Recent developments have shown that WSNs are crucial for expansion of many applications, including those in the military, the healthcare industry, and industrial monitoring [1]. Randomly dispersed sensor nodes (SNs) with limited energy, storage, and computational resources make up this self-organized network [2]. The SNs track several variables, including pressure, moisture, and heating rate, and subsequently transmit the information to base stations (BSs) [3]. Threats to security are the main problems with WSNs. The cause of this is that SNs have limited resources and are vulnerable to attack [4]. There are typically two sorts of attacks carried out in WSNs: internal attacks and external attacks. In internal attacks, SNs act selfishly to protect their energy and storage, in contrast to external attacks, when the attackers seize control of the SNs to carry out destructive operations [5]. Therefore, it is essential to locate and eliminate the malicious nodes from the

system [6]. By establishing smart contracts, which are written agreements that govern the entire system, the blockchain technology offers a potential solution to the aforementioned problems. It was first established and made up of nodes that monitor a distributed ledger's status. Public, private, and consortium networks are the three main forms of blockchain networks [7,8]. Any node can join the fully decentralized network by joining the public blockchain, which is completely decentralized. Only the chosen nodes may participate in the private blockchain's permission-based system [9]. The consortium blockchain is a partially decentralized system that is overseen by numerous companies. Consensus is used by blockchain miners to ensure data integrity [10].

The system employs a variety of consensus techniques, including proof of authority (PoA), proof of work (PoW), proof of stake (PoS), and others [11,12]. In a PoW network, the nodes choose the miner nodes by solving a mathematical puzzle; the first node that obtains the answer to the puzzle will add a new block to the blockchain [13]. Implementing the PoW involves a lot of calculation. Blocks and transactions in PoA are verified by validators, which are nodes that have been chosen earlier [14]. High computational resources are thus not necessary for the selection of miners. The miners that have the most coins confirm the blocks in a PoS system. A blockchain is a useful tool for maintaining a distributed ledger of transactions across multiple groups [15,16]. Since the blockchain is unchangeable, the data cannot be altered. As the units in a blockchain are connected by hashes, the transaction data in a blockchain are secure [17]. The block body contains the transactions, while the block header contains the hashes of the Merkle tree and preceding blocks [18]. Without validation, the attackers use the network's resources to fake the addresses and positions of the trustworthy nodes [19,20]. The routing method suffers when there are intrusions in the system. The attackers corrupt the data and transmit false sensed data that reduces the performance of the system [20]. Nodes could act selfishly and decline the data packets after being authenticated. As a result, the trust value is used to determine the nodes' level of trust. A safe routing technique using blockchain-based encryption and trust assessment is suggested to address the existing challenges. The points mentioned below are the contributions of this article

- The suggested BCR-TM-JSO method designates CH selection and routing from common nodes. The transport of data from SNs to BS in real time and with minimal energy consumption is guaranteed by the suggested routing method. The aggregator nodes (ANs) serve as data-delivery relay nodes.
- The Rivest–Shamir–Adleman (RSA) mechanism ensures secure and trustworthy data transfer. Without the need for authentication, any node can simply access the network's resources and information. To protect the network from outsiders, node authentication is crucial.
- As a result of the victim nodes dropping data packets, there are more retransmissions and delays. Consequently, reliable nodes are used to accomplish secure and effective routing.

The remainder of this paper is structured as follows. Section 2 presents a literature review of blockchain-based routing in WSN. Section 3 defines the preliminaries of the system model and attacker model, respectively. Section 4 explains the process of the proposed method along with mathematical equations. Section 5 elaborates the simulation results and the comparative analysis of the proposed method. Finally, the conclusion is stated in Section 6.

2. Literature Review

Blockchain-based secure routing and trust management (BSRTM) has been demonstrated in WSNs by Saba Awan et al. [21]. In this article, a concept for encryption and trust assessment was put out using a blockchain to hold the identities of ANs and SNs. ANs and SNs were authenticated in private and public blockchains, correspondingly. To remove malicious nodes from the system, the trust levels of SNs were determined. Consideration was given to the SNs' trust values and residual energy when performing secure routing

within the system. Additionally, RSA, an asymmetric key encryption algorithm, was employed to secure the communication protocol. This suggested paradigm used BSs, which have powerful computational capabilities, to identify and validate ANs.

A blockchain technology method has been demonstrated by Sung-Jung Hsiao and Wen-Tsai Sung [22] to improve the data security of WSNs. This study builds a secure WSN framework using data transfer and blockchain-based technology. The current wireless technology uses a blockchain-based approach to strengthen transmission dependability and is built on IoT architecture. This research methodology transforms the blockchain-based transaction ledger into a database of sensor data. As a result, the suggested system collects and evaluates sensor data to increase the dependability of the wireless sensing network. Additionally, this cutting-edge blockchain-based system has the ability to handle a private cloud end. However, as the amount of data grows, the amount of time required for the computation process grows, decreasing the quality of the data transfer.

A trusted distributed routing strategy for ESN combining blockchain and the Deep Convolutional Neural Network (DCNN) algorithm has been presented by M Revanesh and Venugopalachar Sridhar [23]. The distributed routing information in the WSN is managed using a blockchain technique, and the Salp Swarm Optimization algorithm is used to achieve the best routing. Using DCNN, the best routing choices were made while taking into account the variances in routing information between the nodes. In order to create a reliable routing method for WSN, this article intercorrelates the blockchain notion with DCNN. In the PoA blockchain, every routing node was represented as a minion with limited privileges and rights but a distinct address.

In order to improve the security and effectiveness of WSNs' routing, Ibrahim A. et al. [24] have developed a trusted routing system that integrates deep blockchain and Markov decision processes (MDPs). The suggested method makes use of a PoA technique within the blockchain network to authenticate the transmission of the node. The correct next hop was then selected using MDPs as a forwarding node capable of swiftly and securely delivering messages. The MDPs' design was adopted to help routing nodes choose the most trustworthy and effective route connections and to make more informed routing decisions. Backward error propagation requires an additional layer to accomplish the operation.

In the system, Sana Amjad et al. [25] showed how to use DDR-LEACH, a low-energy adaptive clustering hierarchy for blockchain-based authentication and CH selection. Based on maximum residual energy, degree, and minimum distance from BS, the DDR-LEACH protocol was utilized to replace CHs with regular nodes. Additionally, using the blockchain to store a significant amount of data is highly expensive. The inter-planetary file system (IPFS), which operates in a superior fashion to the current encryption systems, was employed to ensure data security. Furthermore, employing a consensus protocol approach to validate transactions required a significant computing investment. The suggested manner in this case was using the PoA consensus technique. However, PoA has a high processing cost and blockchain data storage is quite expensive.

The blockchain method has been used by Rekha Goyat et al. [26] to demonstrate a secure localization strategy for WSNs based on trust evaluation. This study focuses on the development of blockchain technology and the trust-evaluation algorithm. Each beacon node's trust value was calculated during the localization system based on several trust measures, and the relevant weights were then continuously changed. Higher trustworthy beacon nodes were then solely chosen for the mining process. Beacon nodes with the greatest blockchain trust levels carry out the mining operation. Proof of work (PoW) consensus is primarily utilized as a mathematical problem for certifying newly created blocks, although PoW demands intensive processing and substantial resources to carry out the activity of mining, with high energy consumption.

A routing technique that offers a shared memory across the network's nodes by utilizing blockchain technology has been demonstrated by Hilmi Lazrag et al. [27]. The technology is quite young; it originally gained popularity in the world of cryptocurrencies, and recently, it has been used in numerous industries where it has demonstrated its effec-

tiveness. Presented is a methodology that takes advantage of the potential of Blockchains in order to obtain a better resolution to the imbalanced traffic load, the high levels of disturbance, and the security problems. The strategy entails using blockchain to store all network activity as a shared memory across nodes. Despite the positive outcomes, using a cost function to decide on a route and make decisions is only presented as a proof of concept.

Utilizing a red deer algorithm (RDA)-based clustering technique with blockchain-enabled safe data transmission, known as RDAC-BC, Gia Nhu Nguyen et al. [28] have proven energy-efficient and secure clustering-based network connectivity in pervasive wireless devices. Their suggested RDAC-BC method conducts node setup and clustering utilizing the RDAC methodology. Following the selection of the CHs, secure data transfer using a blockchain is carried out between the cluster members (CMs) and the CHs. The use of blockchain and RDAC platform contributes to increased security and energy efficiency. Consequently, the suggested RDAC-BC uses significantly less energy than the traditional algorithms.

3. Preliminaries

The encrypted routing and trust values are transmitted from BSs to other network nodes using the suggested BCR-TM-JSO protocol. Additionally, the blockchain's consensus processes verify each and every communication between nodes. The ANs authenticate and authorize the SNs whenever they communicate with them, allowing the SNs to send packets to ANs. Additionally, the BSs authenticate the ANs before allowing them to communicate with other ANs or BSs. The connections are recorded on the blockchain when the identities of the nodes have been verified. The blockchain cannot be used to remove transaction data. Utilizing these structural qualities, malevolent nodes manipulating the routing information or trust value can be quickly identified [29]. Because of the blockchain's traceability and accessibility characteristics, BCR-TM-JSO can identify rogue nodes. Thus, the blockchain offers secure routing as well as a reliable trust-evaluation system for malicious node detection.

3.1. Attacker Model

The WSN is susceptible to a wide range of assaults, wherein the malicious nodes acquire the identities of an intermediate node or inappropriately construct several identities before joining the network. The rogue nodes listen in on node-to-node communication before broadcasting false information throughout the network. There are several ways to carry out the attack. Direct communication between malicious and authorized nodes is one of them [30]. Another technique is when a rogue node assumes a genuine node's identity and broadcasts false information on their behalf. Another threat is denial of service (DoS) [31], in which hostile nodes launch an attack to drain the resources of trustworthy nodes. The network's connectivity and efficiency are impacted by this attack.

3.2. System Model

This study suggests a trust assessment model and blockchain-based encryption for a secure routing method in WSNs. The following are some of the model's assumptions.

- A unique Ethereum address is shared by all ANs, SNs, and BSs.
- All of the BSs and ANs are valid.
- The objective metrics are not susceptible to severe system parameters or external variables.

In this study, data transmission over the network is secured and made reliable using the Rivest–Shamir–Adleman (RSA) mechanism. In the suggested approach, the data are initially sensed by the SNs and transmitted to the related ANs. The data are then received by the ANs and sent to the nearby BSs. In our architecture, various network nodes are registered and authenticated using two blockchains. The public blockchain stores the ANs' identities, and they are permitted access to and membership in the private blockchain. Additionally, as illustrated in Figure 1, ANs that are a member of the private blockchain carry out the registration and authentication of the SNs.

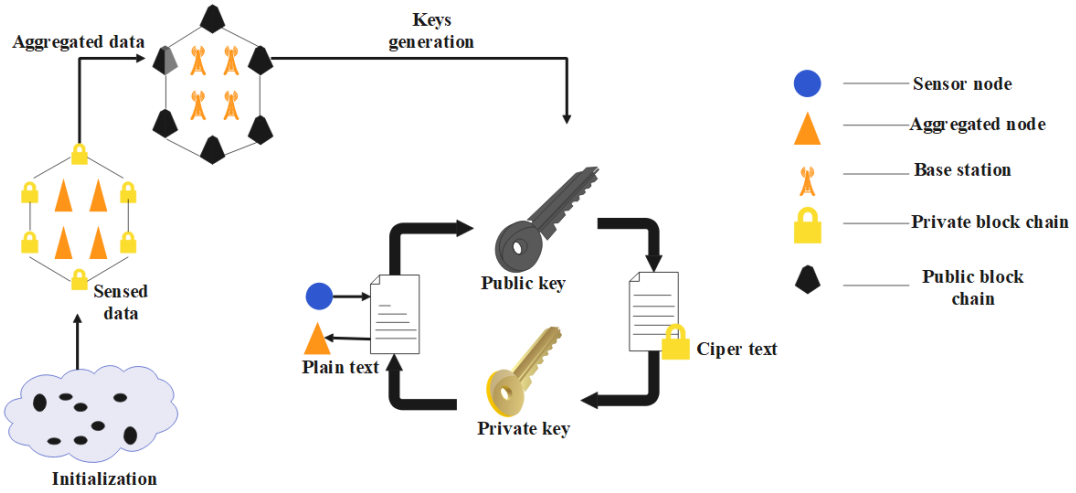

Figure 1. An overall flow of blockchain based WSN.

3.3. RSA Cryptography

The RSA public key cryptosystem was established by Rivest, Shamir, and Adleman, and is extensively exploited for secure data transmission [32]. This research assesses the development of a number of symmetric and asymmetric encryption methods known as RSA in order to determine whether the encryption ratio is high when both encryption techniques are used. As per various criteria, every method has its own benefit. Because the key length in an asymmetric encryption technique is long, breaking the code in RSA is difficult. When it comes to throughput, more throughput means less power usage. The RSA algorithm is the better option in symmetric key encryption approaches and more secure in the asymmetric encryption technique because it generates keys by factoring high prime numbers. As a result, RSA is determined to be the superior answer in this manner. The comprehensive description of RSA is specified as follows.

1. Key Generation

Step 1: Initially, select two dissimilar and large primes p and q and calculate $n = p \times q$ and Euler's totient function is represented as $\varphi(n) = (p-1) \times (q-1)$.

Step 2: Select a public key parameter e (i.e., $1 < e < \varphi(n)$ and $\gcd(\varphi(n), e) = 1$).

Step 3: Calculate the private key parameter d (i.e., $(e \times d) - 1 = 0 \; mod \; \varphi(n)$) by means of the extended Euclidean procedure. $\left(e, n \in Z \times Z^*_{\varphi(n)}\right)$ and $\left(d, n \in Z \times Z^*_{\varphi(n)}\right)$ are stated as public/private key pairs.

2. Encryption

The function of encryption $e_h : Z_n \rightarrow Z_n$ is stated as

$$e_h(M) : b = M^e mod \; n$$

3. Decryption

The function of decryption $d_h : Z_n \rightarrow Z_n$ is stated as

$$d_h(M) : M = b^d mod \; n$$

The steps included in the proposed model are initialization, registration, and authentication of the nodes

3.3.1. Initialization

In this phase, the network's established nodes are all initialized, and RSA is utilized to transmit data securely. The public and private keys are generated separately for each node. Whereas the private key of every node is deliberately private and only known

by authorized nodes, the public key of every node is saved on the BS, where the public blockchain is implemented. Additionally, it is assumed in the proposed work that ANs aggregate forwarding data and that BSs and ANs are both trusted nodes. The acquired data is encrypted by the SNs using the ANs' public keys before being sent to the ANs. Using their private keys, the ANs decrypt the data packets. While transmitting network packets from ANs to BSs, the same procedure is used.

3.3.2. Registration

The public blockchain and private blockchain are used to register the ANs. When the aforementioned procedures are successfully completed, the public blockchain retains a track of the names of ANs during the registration process. Nobody else can maliciously alter the identities of ANs when they are stored in the blockchain. In a similar manner, the blockchain offers a trustworthy authentication method in the WSN. In contrast, ANs are removed from the network if their identities cannot be verified. SNs may join the blockchain after finishing the registration process. SNs are connected to the appropriate ANs following their placement. Registering the nodes reduces external attacks.

3.3.3. Authentication

Whenever SNs and ANs interact, the ANs use a private blockchain to verify the SNs' identities. Additionally, BSs use a public blockchain to leverage AN authentication while communicating with ANs. Mutual authentication is carried out between Ans, since they make requests to the BSs whenever they connect with one another [33].

4. Proposed Method

4.1. Jellyfish Search Optimizer (JSO)

Around a world, jellyfish can be found in water that is different depths and degrees. They have a bell-like shape; many of them have a diameter of less than a centimeter, while others are rather enormous. Each of the numerous species demonstrates a unique adaption to the maritime habitat. These organisms are able to appear practically anywhere in the water due to this phenomenon, which works in conjunction with each jellyfish's individual movements within the swarm and the ensuing tidal currents to generate jellyfish blooms. The ideal place would be determined by comparing food quantities, because the amount of food at each spot a jellyfish visits fluctuates. As a result, a novel method is created that is motivated by the searching and swimming of jellyfish in the ocean [34]. This is called a jellyfish search optimizer (JSO). The behavior and motion of jellyfish in sea are theoretically modelled in the following sections, and an optimization method based on the numerical model is then created [35].

The jellyfish are drawn to the tidal currents because they are rich in nutrients. By combining all the trajectories from every jellyfish in the sea to the jellyfish that really is right now in the optimal position, the trend of the ocean current is identified. Equation (1) reproduces the ocean current,

$$\overrightarrow{trend} = \frac{1}{n_{pop}} \sum \overrightarrow{trend_i} = \frac{1}{n_{pop}} \sum (X^* - e_c X_i) = X^* - e_c \frac{\sum X_i}{n_{pop}} = X^* - e_c \mu \quad (1)$$

The number of jellyfish is stated as n_{pop}; e_c is the variable that controls the attraction, and X^* represents the jellyfish presently in the best position in the swarms. The mean location of all jellyfish is μ, and the distance among a jellyfish's present ideal position and the average position is represented as df. Fix the df as $e_c \mu$, because, based on this statement, mean location comprises the probability of all jellyfish, therefore Equation (2) is represented as

$$df = e_c \mu \quad (2)$$

Thus, ($\overrightarrow{trend}$) is defined and updated as Equation (3),

$$\overrightarrow{trend} = X^* - df \tag{3}$$

The variance of the jellyfish has a uniform spatial pattern in all coordinates. Thus Equation (4) is written as

$$df = \beta \times \sigma \times rand^f(0,1) \tag{4}$$

$$\text{Fix } \sigma = rand^{\alpha}(0,1) \times \mu \tag{5}$$

$$\text{Later, } df = \beta \times rand^f(0,1) \times rand^{\alpha}(0,1) \times \mu \tag{6}$$

For further updating, Equation (6) is modified as

$$df = \beta \times rand(0,1) \times \mu \tag{7}$$

$$\text{Here, } e_c = \beta \times rand(0,1) \tag{8}$$

Therefore,

$$\overrightarrow{trend} = X^* - \beta \times rand(0,1) \times \mu \tag{9}$$

Currently, the new position of every jellyfish is specified through Equation (10)

$$X_i(t+1) = X_i(t) + rand(0,1) \times \overrightarrow{trend} \tag{10}$$

Equation (10) is updated as Equation (11),

$$X_i(t+1) = X_i(t) + rand(0,1) \times \overrightarrow{trend} \times X^* - \beta \times rand(0,1) \times \mu \tag{11}$$

where $\beta > 0$ is stated as distribution coefficient, which is connected to measurement of $\overrightarrow{trend}$. β is fixed to 3.

Jellyfish move in swarms in both passive (type A) and active (type B) ways. Many jellyfish originally display type A movement whenever the swarm has just been established. They gradually start to move more in a type B manner. The movement of jellyfish within their own places is known as type A movement, and the most recent position of every jellyfish is provided by Equation (12),

$$X_i(t+1) = X_i(t) + \gamma \times rand(0,1) \times (U_b - L_b) \tag{12}$$

The upper and lower bound are stated as U_b and L_b; $\gamma > 0$ is declared the motion coefficient, $\gamma = 0.1$ is attained from the Equation (12). This effort is deliberated as an operative manipulation of local exploration, which is expressed in Equation (13)

$$\overrightarrow{Step} = X_i(t+1) - X_i(t) \tag{13}$$

$$\text{Here, } \overrightarrow{Step} = rand(0,1) \times \overrightarrow{Direction} \tag{14}$$

$$\overrightarrow{Direction} = \begin{Bmatrix} X_j(t) - X_i(t) \; if \; f(X_i) \geq f(X_j) \\ X_i(t) - X_j(t) \; if \; f(X_i) < f(X_j) \end{Bmatrix} \tag{15}$$

where f is a function of position X. Therefore,

$$X_i(t+1) = X_i(t) + \overrightarrow{Step} \tag{16}$$

Using a temporal control system, the type of motion over time is determined. The time control mechanism consists of constant C_o and a time-control function $c(t)$ to control

the jellyfish's ability to travel either inside the jellyfish swarm or in accordance with the ocean current. The random value of the time-control function varies over time from 0 to 1. The jellyfish float together with the current of the ocean when its value reaches C_o. They go inside the swarm when its value is less than C_o. The time control varies at random between zero and one, and the precise value of C o is unknown. C_o is therefore set to 0.5, the average of zero and one, which is stated as Equation (17)

$$C(t) = \left| \left(1 - \frac{t}{Max_{iter}}\right) \times (2 \times rand(0,1) - 1) \right| \tag{17}$$

Jellyfish populations are typically started at random and have drawbacks, such as slow divergence and a propensity to get stuck at a local optimum due to poor variability. One of the simplest chaotic maps, the logistic map, is created to increase the diversity of the initial population. This map offers less chance of nonlinear equations and more varied initial populations than random selection. As a result, this map is written as Equation (18),

$$X_{i+1} = \eta X_i(1 - X_i),\ 0 \le X_0 \le 1 \tag{18}$$

The position's chaotic value for the ith jellyfish is stated as X_i; the initial population is stated as X_0, which is held in the range of 0 to 1; η is fixed as 4.0.

Oceans can be found all over the earth. Since the earth is spherically symmetric, a jellyfish would eventually get back to the contrary bound whenever it leaves the search region, which is expressed as Equation (19),

$$\begin{cases} X'_{i,d} = (X_{i,d} - U_{b,d}) + L_b(d)\ if\ X_{i,d} > U_{b,d} \\ X_{i,d} = (X_{i,d} - L_{b,d}) + U_b(d)\ if\ X_{i,d} < L_{b,d} \end{cases} \tag{19}$$

Position of ith jellyfish in dth dimension is stated as $X_{i,d}$; U_b and L_b are represented as upper and lower bounds. The updated position is specified as $X'_{i,d}$.

4.2. CH Selection

The BCR-TM-JSO is employed for CH routing and selection. The proposed BCR-TM-JSO is selected because it comes with collision avoidance as standard, making it very helpful for creating congestion-aware routing in unforeseen circumstances.

4.2.1. Fitness Function

The BCR-TM-JSO technique makes use of the trust measure, QUN (queue utilization between nodes), network quality, and distance. As a consequence, the provided cost metrics are used throughout the routing process to stop rogue nodes by placing a large emphasis on the trust value for the trust metric. Subcategories of the cost indicators such as QUN, link quality, and distance are provided to increase the overall efficiency in terms of energy usage and overhead.

a Trust metric

Trust has been utilized to improve security against malicious nodes and is a significant cost value for routing through an IoT-based WSN. An IoT-based WSN's mobile nodes communicate with one another by depending on the mutual trust those nodes have built up over time. Direct trust is a packet-forwarding method depending on the number and percentage of data packets received (RP_{ij}) and delivered (TP_{ij}). Equation (20) is used to calculate the direct trust degree.

$$Trust = \frac{TP_{ij}}{RP_{ij}} \tag{20}$$

b QUN and link quality

QUN is demarcated as the ratio amongst the number of packets engaged in the queue of *j*th node $\left(QUN_j\right)$ and quantity of obtainable packets in the particular node's queue (QUN_{total}). Equation (21) is utilized to calculate the QUN.

$$QUN = \frac{QUN_j}{QUN_{total}} \tag{21}$$

The fraction of transmissions and retransmissions requirements for successful data transfer between nodes i and j is referred to as link quality.

$$Link\ quality = \frac{1}{f \times r} \tag{22}$$

f and r state the forward and reverse information delivery.

c Distance

To identify the direct route over the IoT-WSN, the Euclidean distance between the nodes is decided by Equation (23).

$$Distance = \sqrt{\left(x_j - x_i\right)^2 + \left(y_j - y_i\right)^2} \tag{23}$$

$x_i,\ x_j$ and $y_i,\ y_j$ are the coordinates of the nodes i and j, respectively.

In BCR-TM-JSO, a weight value (δ) is considered for each cost value to convert into a single cost value as shown in the Equation (24).

$$c_k = \delta_1 \times Trust + \delta_2 \times QUN + \delta_3 \times Link\ quality + \delta_4 \times Distance \tag{24}$$

Thus, 0.35, 0.25, 0.25 and 0.15 are the weighted mean for $\delta_1,\ \delta_2,\ \delta_3$ and δ_4, respectively. While creating the data transmission, the malicious nodes that cause packet drops are avoided by employing the trust metric. The QUN and link quality are then considered in the cost measure to create a route with less traffic. This increases packet delivery and lowers the possibility of a network collision. Additionally, to lower the nodes' energy consumption, the shortest distance is found.

4.2.2. Route Selection

In WSN, a client that receives data registers its ID in the routing table. Once all the information has been collected, the client starts sending it to the intended location. In the event that the source consumer obtains several detection warnings from its neighbors, BCR-TM-JSO assesses the route with the highest pheromone rate. Clients with a one-hop connectivity to the router send data directly, whereas clients more distant from the router use the BCR-TM-JSO for modular connection. Data packets are therefore sent along the route with the strongest signal. After the routing path has been constructed using BCR-TM-JSO, the transmission of data packets begins. Data are transported via a network with less congestion when the queue length used in routing is used to provide a collision-free path. The route's mean is then calculated based on the packets transmitted and received and the link quality. As a result, the optimum data-transmission path is determined by utilizing the aforementioned fitness characteristics.

4.3. Blockchain Data Structure for Malicious Node Detection

This study suggests a block data structure centered on a malicious node identification blockchain to best understand the blockchain. It is distinct from conventional WSNs in which it cannot consistently be discovered. The data structure in this case is mostly split into two sections. On the one side, the block header primarily carries the earlier block's hash value, which is used to link the current block to it and maintain the blockchain's integrity. The major data for the wireless sensor node in the block, including its position, ID, and condition, is instead contained in the block body. The block's hash value is created

using this data, the preceding block's hash value, and a random number. Here, Hash1 is the hash pointer of the D1 sensor data, Hash12 is the hash pointer of Hash1 + Hash2, the layer stack is then added, and lastly, the distinct Merkle tree is produced. In addition to using a "block + chain" chain-data structure, it also employs a hash pointer to create a Merkle tree to store the data gathered by each block. When using such a data structure, it is feasible to modify the block's hash pointer after any block's contents are modified, ensuring that the data cannot be altered. In order to further limit the likelihood of malicious exploitation, assure security and integrity, and enhance the simplicity of the detection phase, the data are also collected by numerous sensors utilizing this data format whenever the entire network is deployed. The blockchain's data format for identifying nodes is shown in Figure 2.

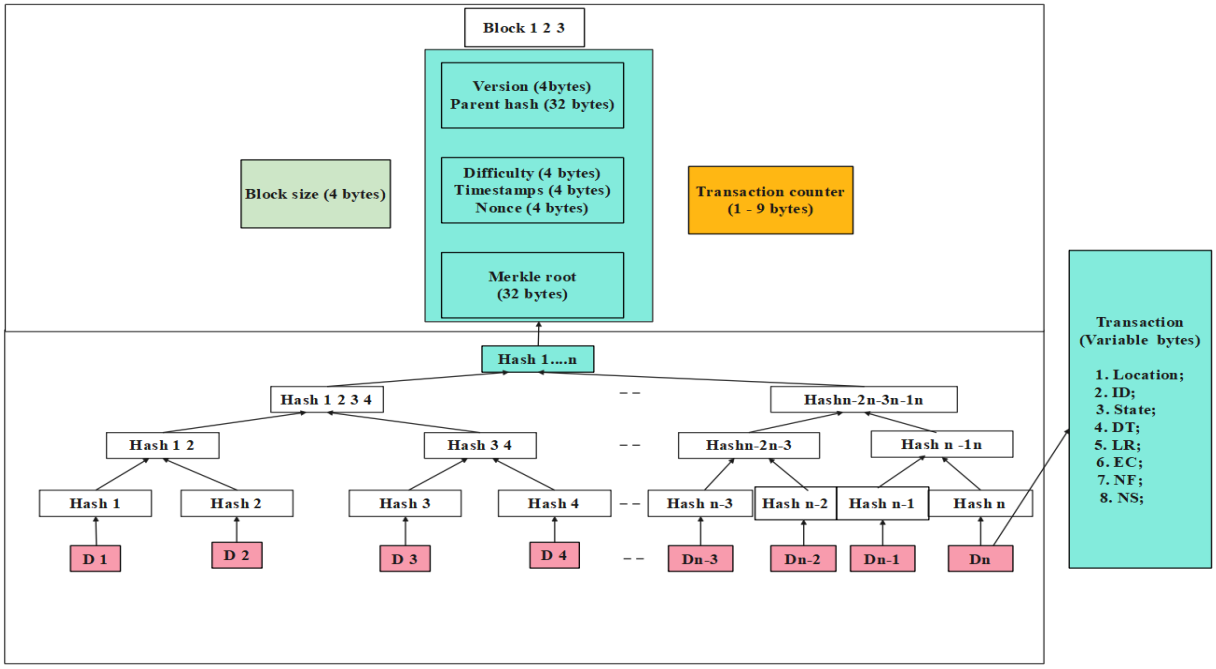

Figure 2. Data structure of blockchain for node detection.

The blockchain is a decentralized public ledger that has a rigorously encrypted method for member interoperability. A blockchain is really merely a data structure comprised of a singly linked list of network nodes, each of which contains a number of blocks that each include the cryptographic information from the block before it. The structure of block header and Merkle root function is shown in Figure 3. In Figure 3, when block header is created, a new block is created, and a hash is generated using pairs of keys and a digital signature. A hashing technique is then used to continuously construct new blocks using the hash of the previous block. A hash is generated using the private, public keys and the hash guarantees the integrity of the blocks.

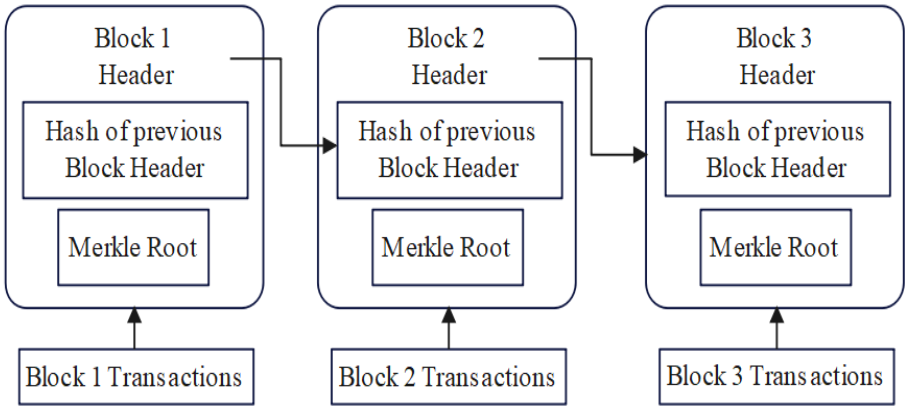

Figure 3. Structure of block header and Merkle root function.

4.4. SHA256 Algorithm

The SHA256 hash algorithm creates a message digest, which is a hash value with a length of 256 bits, for messages that are no longer than 264 bits in length. A 64-character hexadecimal string is able to describe the 32-byte array known as the (e digest. Five stages make up the SHA256 algorithm's computation:

1. Increase the input data's number of 0 bits till it reaches 448 bits. (After that, increase the input data's length by 64 bits to 512 bits.
2. Separate the 512-bit data into 16 classes: $M_0 - M_{15}$.
3. Reset the vectors $K_0 - K_{63}$ and $h_0 - h_7$, and its initial values of A, B, C, D, E, F, G, and H are set as $h_0 - h_7$.
4. Fix the variable t to loop from 0 to 63; then mentioned as Equation (25)

$$\begin{aligned} B_{t+1} &= A_t C_{t+1} = B_t,\ D_{t+1} = C_t,\ F_{t+1} = E_t,\ G_{t+1} = F_t, H_{t+1} = G_t, \\ A_{t+1} &= H_t + \sum_{1}(E_t) + Ch(E_t, F_t, G_t) + K_t + W_t + \sum_{0}(A_t) + Maj(A_t, B_t, C_t) \\ E_{t+1} &= H_t + \sum_{1}(E_t) + Ch(E_t, F_t, G_t) + K_t, W_t,\ D_t \end{aligned} \tag{25}$$

5. Consider $h_0 = h_0 + A_{63}, h_1 = h_1 + B_{63}, h_2 = h_2 + C_{63}, h_3 = h_3 + D_{63}, h_4 = h_4 + E_{63}, h_5 = h_5 + F_{63}, h_6 = h_6 + G_{63}, h_7 = h_7 + H_{63}$. Output $h_0 - h_7$.

From the above-stated process, $\sum_1(E_t)$, $\sum_0(A_t)$, $Maj(A_t, B_t, C_t)$, $Ch(E_t, F_t, G_t)$ are logical functions, and W_t is updated as Equation (26)

$$W_t = \begin{cases} M_t, & 0 \le t \le 15 \\ \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + (W_{t-16}), & 16 \le t \le 63 \end{cases} \tag{26}$$

It is clear that the SHA256 execution provides a solution that involves updating the variables of A and E, which calls for a number of addition processes and 64 iterations. Therefore, a key factor in lowering the algorithm's running time will be the optimization of these two variables.

4.5. Blockchain-Integrated WSNs

Data transfer on the blockchain is extremely safe when the features cannot record data. This security technique is one of the WSN's advantages. As shown in Figure 4, a block in the chain is linked to a sensor device including such sensors as 1, 2, 3, 4, or 5. Additionally, these blocks gather information gathered from the sensors of many other blocks in addition to gathering and storing data from their own sensor devices. As a result, each block stores both its own sensor information and the sensor information of neighboring blocks with the non-existence of a central block.

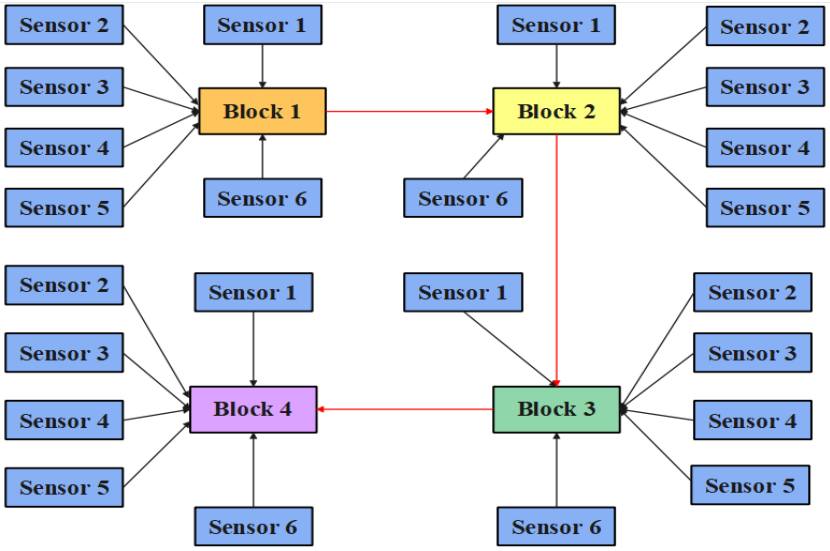

Figure 4. WSN blockchain technology.

The links between the chain's building blocks are depicted in Figure 4. A block has fields such as previous hash and current hash. In order to keep things simple, consider that the codes produced by hash functions take the form of a short code snippet made up of numbers and symbols, as illustrated in Figure 5.

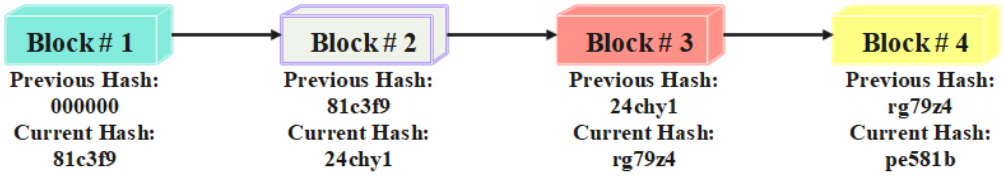

Figure 5. Connection of WSN nodes in the blockchain.

5. Simulation Results

This section briefly discusses the simulation-based evaluation of the suggested model (BCR-TM-JSO). A trust-evaluation mechanism is used in our suggested model to find the network's malicious SNs. Additionally, a technique for authentication is offered to protect our network from intrusion. A routing system is also suggested to provide real-time, energy-efficient data delivery from SNs to BSs. On the provision of authentication, the proposed model is contrasted with the current paradigm. Since the impact of authentication cannot be clearly seen, it is assessed based on network lifespan, energy usage, and throughput. With the use of both PoW and PoA consensus methods, the network as a whole is verified.

5.1. Simulation Setup

A 64-bit operating system, an Intel(R) Core (TM) i5-5200U CPU running at 2.20 GHz, and 8 GB of RAM are the system requirements for the simulation scenario. SNs are immobile during the simulations. Table 1 provides the system configuration for the model assessment.

Table 1. Simulation parameters.

Parameters	Values
Initial energy of SNs	0.05 J 5.2
Sensing area	100×100 m^2
BSs	2
Deployment	Random
SNs	100
ANs	4
Network Interface	Wireless
CH	4

5.2. Quantitative Analysis of Proposed BCR-TM-JSO with Existing Methods

The proposed BCR-TM-JSO method's effectiveness is authenticated by utilizing different optimization methods including the reptile search algorithm (RSA) and artificial gorilla troops optimizer (AGTO) by means of the packet-delivery ratio (PDR). In this study, a performance analysis is performed for various node counts, and is shown in Table 2. Table 2 clearly shows that the proposed BCR-TM-JSO achieved better PDR when compared with existing RSA and AGTO methods.

Table 2. Analysis of PDR with various optimization techniques.

Number of Nodes	Packet-Delivery Ratio (%)		
	Existing RSA	Existing AGTO	Proposed BCR-TM-JSO
10	82.34	85.46	95.44
20	80.25	83.29	93.48
30	78.81	81.49	91.18
40	75.77	80.47	89.40
50	72.03	79.58	86.11
60	70.29	78.17	83.31
70	69.47	76.74	81.98
80	68.63	74.41	80.45
90	67.11	72.28	78.88
100	66.82	71.89	77.13

5.3. Performance Analysis

For the developed system assessment, various performance measures are taken into account, and they are listed below.

5.3.1. Throughput

The network performance for authenticated and unauthenticated nodes is shown in Figure 6. Table 3 displays the throughput results using the current BSRTM [21] and DDR-LEACH [25] techniques.

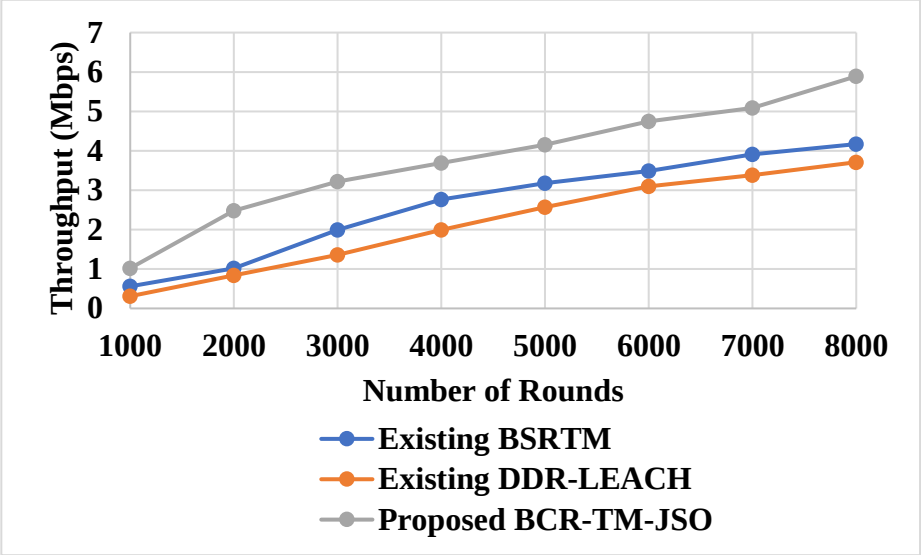

Figure 6. Performance of throughput with existing methods [21,25].

Table 3. Performances of throughput.

Number of Rounds	Throughput (Mbps) Existing BSRTM [21]	Existing DDR-LEACH [25]	Proposed BCR-TM-JSO
1000	0.56	0.31	1.02
2000	1.02	0.84	2.48
3000	1.99	1.36	3.22
4000	2.76	1.99	3.69
5000	3.18	2.57	4.15
6000	3.49	3.10	4.75
7000	3.91	3.38	5.09
8000	4.17	3.71	5.89

A network without authentication allows any hostile node to join and carry out nefarious operations, while a system with authentication only allows authenticated and registered nodes to join. Figure 6 demonstrates the network's throughput eventually rises as the number of rounds rises.

5.3.2. FPR and FNR

Figure 7 displays how various malicious nodes affect the FPR (False Positive Rate) and FNR (False Negative Rate). FPR and FNR rise together with the number of malicious nodes. The cause is that a lot of rogue nodes spread a lot of false information throughout the network. The network's detection performance declines as the number of rogue nodes rises. Additionally, the detection accuracy decreases as the number of malicious nodes exceeds 20, resulting in higher FPR and FNR. Table 4 displays the results for FPR and FNR.

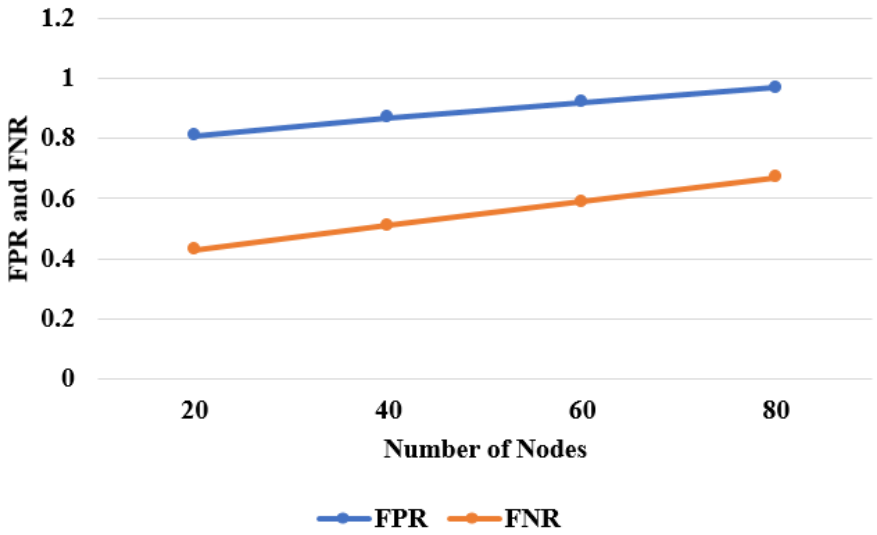

Figure 7. Performance of FPR and FNR.

Table 4. Performances of FPR and FNR.

Number of Malicious Nodes	Proposed BCR-TM-JSO	
	FPR	FNR
20	0.81	0.43
40	0.87	0.51
60	0.92	0.59
80	0.97	0.67

5.3.3. Network Lifetime

This is the window of time wherein the network is active. The number of active nodes affects the network longevity. Table 5 displays the network lifetime performances in terms of dead nodes. Figure 8 compares the suggested model's network lifetime to that of RSA [21].

Table 5. Performances of network lifetime in terms of dead nodes.

Number of Rounds	Dead Nodes	
	Existing BSRTM [21]	Proposed BCR-TM-JSO
1000	12	7
2000	29	16
3000	45	24
4000	62	31
5000	71	38
6000	79	49
7000	88	56
8000	97	67

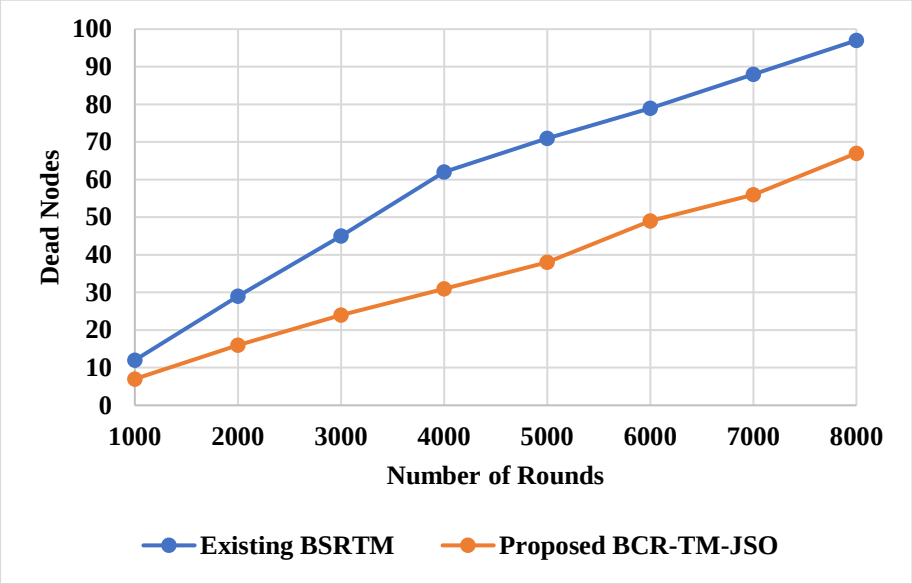

Figure 8. Performance of lifetime in terms of dead nodes with existing BSRTM [21].

Additionally, this illustrates the period of network stability and demonstrates how the distance among SNs and BSs prevents direct communication amongst them. The BSRTM [21] lacks an authentication system, which allows malevolent nodes to join the network, assume the identities of genuine nodes, and transmit false information, which degrades the efficiency of the network. The suggested model involves performing node validation. As a result, no external malicious nodes are permitted to join the network. Following authentication, self-centered nodes are identified based on their trust values, and they are subsequently removed from the network. As a result, the suggested model's network lifetime outperforms BSRTM's [21]. Table 6 displays the network lifetime performances in terms of remaining energy. The lifespan of a network that is dependent on the SNs' remaining energy is shown in Figure 9.

Table 6. Performances of network lifetime in terms of residual energy.

Number of Rounds	Residual Energy (J)	
	Existing BSRTM [21]	Proposed BCR-TM-JSO
1000	0.052	0.071
2000	0.037	0.054
3000	0.023	0.039
4000	0.010	0.021
5000	0.008	0.010
6000	0.007	0.009
7000	0.007	0.008
8000	0.007	0.008

These numbers merely show how easily our suggested model can be used; the stability period and remaining energy are unaffected by the blockchain technology. The blockchain makes the network's interactions between nodes transparent, which further assists in preventing outside nodes from intercepting them. Additionally, only the nodes that have already been authenticated can participate in network communication. While the suggested model uses less energy, the BSRTM [21] with no authentication allows rogue nodes to join in the network and transmit enormous amounts of incorrect information to the forwarder nodes, which use more energy to move packets.

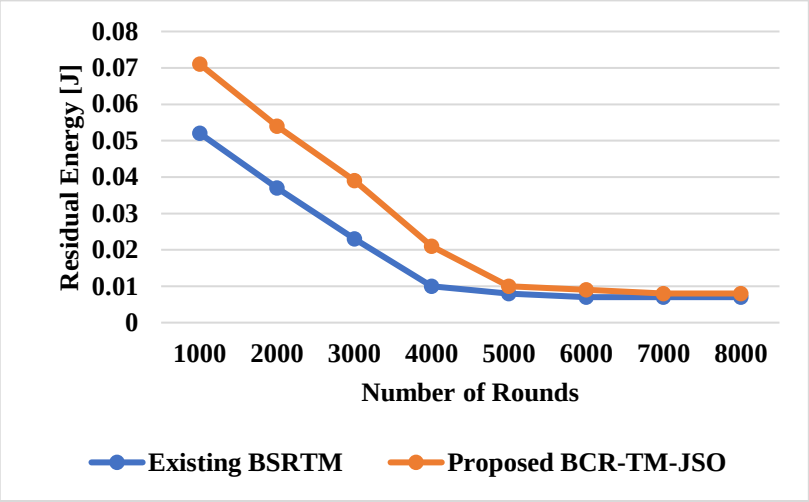

Figure 9. Performance of lifetime in terms of residual energy with existing BSRTM [21].

5.3.4. Consumed Time

Figure 10 displays the time required to evaluate trust across various node counts. As can be seen from the graph, it takes 0.22 milliseconds to calculate the consumed time for a trust evaluation of 10 nodes, whereas it takes 14.46 milliseconds for a trust evaluation of 100 nodes. The DT, FR, and RT values, all of which depend entirely on a node's behavior, are taken into account by the trust-evaluation method. Furthermore, it is evident from the image that as the number of nodes increases and the trust evaluation does as well, the computing cost of the network rises. Table 7 displays the results of time consumed during the transmission.

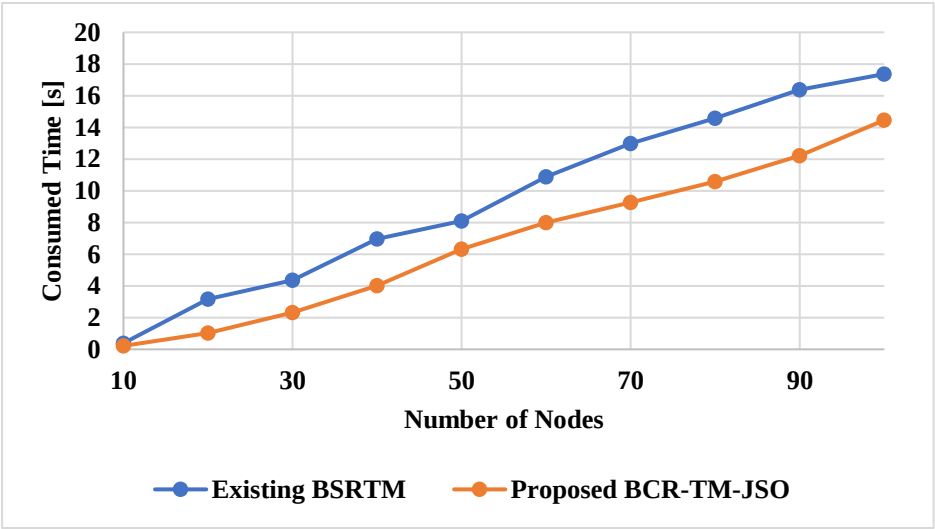

Figure 10. Performance of consumed time with existing BSRTM [21].

Table 7. Performances of consumed time.

Number of Nodes	Consumed Time (s)	
	Existing BSRTM [21]	Proposed BCR-TM-JSO
10	0.39	0.22
20	3.18	1.03
30	4.37	2.33
40	6.97	4.01
50	8.10	6.33
60	10.88	7.99
70	12.99	9.28
80	14.59	10.59
90	16.38	12.22
100	17.37	14.46

5.3.5. Energy Consumption

Depending on function in the network, various nodes have varying levels of trust. Every node's trust value is compared to a predetermined threshold; nodes with high trust values are regarded as legitimate nodes, whereas nodes with low trust values are regarded as malevolent nodes. In addition, the graphic shows how much energy was expended when determining the trust values of various network nodes. Figure 11 displays the computed trust values for several nodes, taking into account performance data. Table 8 displays the energy-consumption effectiveness.

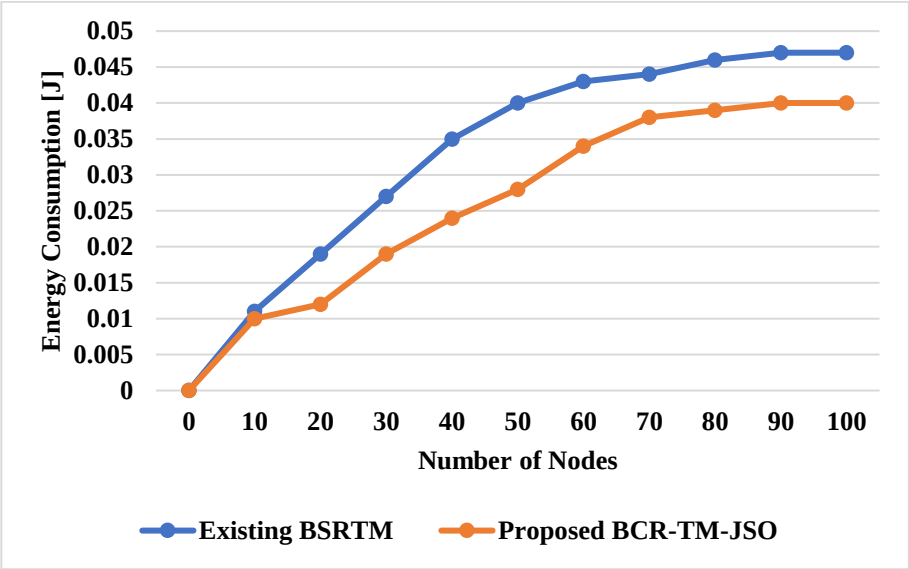

Figure 11. Performance of Energy Consumption with existing BSRTM [21].

Additionally, as the number of nodes rises, so does the overall energy used by the network. After some time, the network's overall energy use starts to gradually rise until it nearly stabilizes. The PoW consensus technique is used by the blockchain-based trust-evaluation model to validate communications and add blocks to the network.

Table 8. Performances of energy consumption.

Number of Nodes	Energy Consumption (J)	
	Existing BSRTM [21]	Proposed BCR-TM-JSO
0	0	0
10	0.011	0.010
20	0.019	0.012
30	0.027	0.019
40	0.035	0.024
50	0.040	0.028
60	0.043	0.034
70	0.044	0.038
80	0.046	0.039
90	0.047	0.040
100	0.047	0.040

5.3.6. Residual Energy

The residual energy is taken into account in each cycle of the analysis of SN energy usage. The residual energy of the SNs decreases as the number of rounds rises. An SN is propagated as a dead node in a network if its energy falls below a predetermined threshold. Table 9 compares the results of residual energy using the existing BSRTM [21] and DDR-LEACH [25] technologies.

Table 9. Performances of residual energy.

Number of Rounds	Residual Energy (J)		
	Existing BSRTM [21]	Existing DDR-LEACH [25]	Proposed BCR-TM-JSO
1000	0.052	0.031	0.079
2000	0.037	0.026	0.071
3000	0.023	0.013	0.058
4000	0.010	0.009	0.042
5000	0.008	0.005	0.020
6000	0.007	0.002	0.010
7000	0.007	0.002	0.009
8000	0.007	0.002	0.009

The measurement of residual energy is shown in Figure 12. Unnecessary data are discarded using a large quantity of AN energy, which reduces the network lifetime. Furthermore, data loss occurs as a consequence of interactions with the malicious nodes. As a result, this model proposes an authentication and trust-evaluation technique. The nodes are initially verified by their MAC addresses. Thereafter, every SN's value is calculated, and nodes whose trust values fall below certain levels are eliminated from the system.

In malicious attacks, the attackers fabricate data and transmit undesirable data, which uses a large quantity of the ANs' energy. The suggested model performs better than a model with bad nodes. Every SN in the suggested model sends the sensed data to the AN that it is connected to. On the basis of DT, FR, and RT, the ANs calculate trust values for SNs. The number of significant and failure communications is calculated based on trust value. The SNs with trust values below the cutoff are then eliminated from the network after the trust value has been calculated. As a result, the suggested framework uses less energy and predicts the longest possible network lifetime.

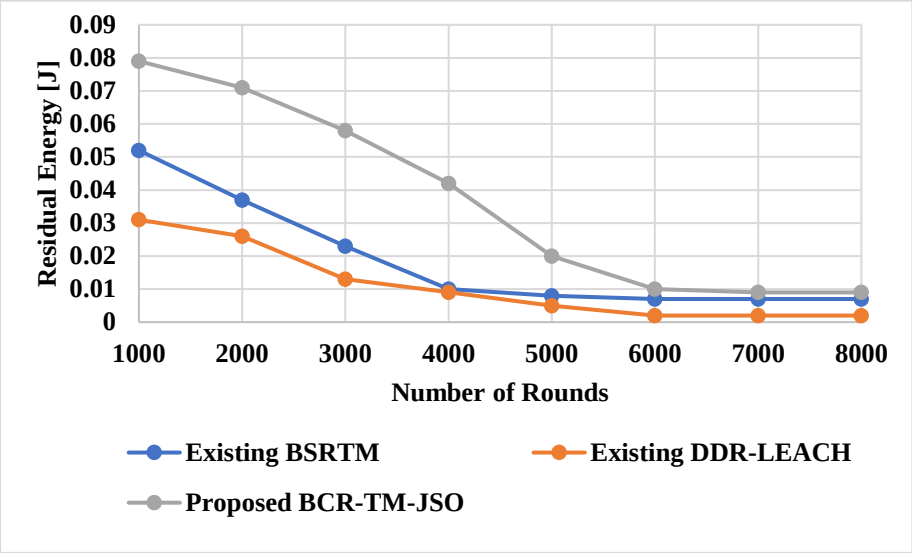

Figure 12. Performance of residual energy with existing methods [21,25].

5.3.7. Packet-Delivery Ratio

This is the proportion of data packets sent through the SNs to data packets successfully established at the BSs. Table 10 displays the results of PDR using the existing BSRTM [21] and DDR-LEACH [25] technologies. The PDR is depicted in Figure 13 in relation to the quantity of rounds. As the number of rounds rises, the quantity of data packets falls.

Table 10. Performances of PDR with hierarchical routing protocols.

Number of Rounds	Packet-Delivery Ratio (%)		
	Existing BSRTM [21]	Existing DDR-LEACH [25]	Proposed BCR-TM-JSO
1000	82.82	85.14	89.29
2000	78.25	81.29	86.02
3000	73.84	77.99	82.48
4000	71.77	73.77	80.40
5000	68.03	70.58	76.18
6000	66.29	68.82	73.37
7000	64.47	66.19	71.30
8000	62.63	64.47	69.93

The distribution of packets involves a large number of SNs, which lowers the computational burden placed on a single SN. The suggested framework improves the chance that data packets will be successfully received. As a result, PDR has a high value in the first round. They continue sending packets to the BSs as long as all SNs are still functional. The figure also shows the PDR in relation to a round number. The SNs' limited battery capacity prevents them from being supplied with energy. As a result, as the number of rounds rises, they become more likely to die, which causes PDR to fall. There are few packets that reach the BS when there are malicious nodes present. Attackers manipulate or randomly forward the packets. The ANs in the suggested paradigm calculate the SNs' trust values. The data packets are then forwarded only via trusted SNs. Therefore, in the suggested model, high PDR suggests minimal packet loss.

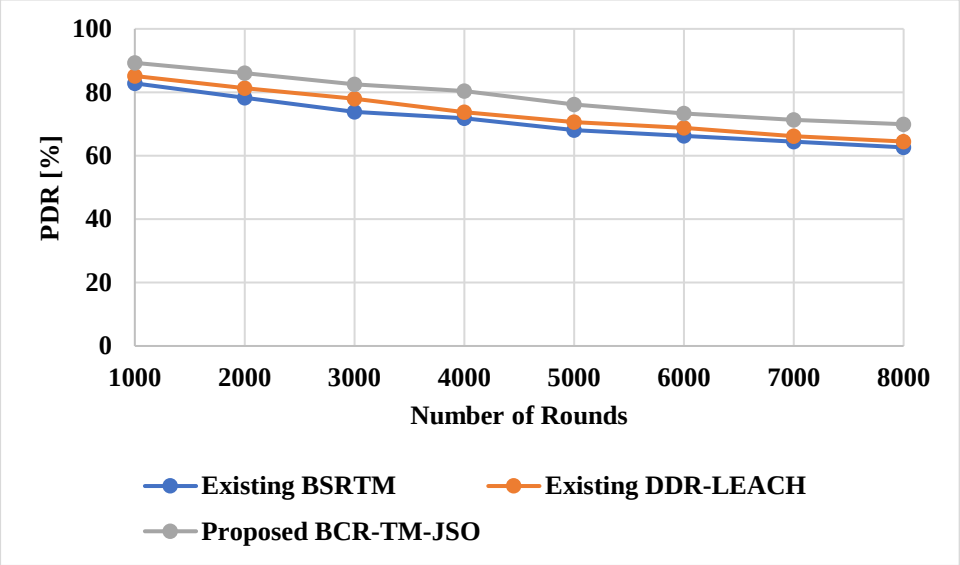

Figure 13. Performance of PDR with existing methods [21,25].

6. Conclusions

With changing environments, wireless sensor networks (WSNs) are vulnerable to a variety of malicious attacks, and they rely primarily on the authentication and encryption mechanism to overcome this difficulty. Due to the real-time modification of routing information, the most common routing techniques in studies are fallbacks describing the malicious nodes on networks. Consequently, an inter-correlated routing method that is trustworthy and reliable is needed. This study introduces a jellyfish search optimizer (BCR-TM-JSO) for malicious node detection in WSN based on the blockchain routing and trust model. The WSN's distributed routing information is managed using a blockchain method, and jellyfish search optimizer is used to ensure the best routing possible. Whereas the public blockchain is implemented on BSs, the private blockchain is implemented on ANs. Only a small number of SNs remain active in the network as SNs gradually lose energy and begin to die. Additionally, RSA is utilized for data-packet encryption and decryption for secure routing. Comparing the simulation results of the proposed BCR-TM-JSO to those of the established blockchain based secure routing and trust management (BSRTM) and the distance, degree, and residual energy-based low-energy adaptive clustering hierarchy (DDR-LEACH), the proposed BCR-TM-JSO produced a better performance in comparison of throughput (5.89 Mbps), residual energy (0.079 J), and packet-delivery ratio (89.29%). In future, the suggested work will be tested with many networks in real-world applications.

Author Contributions: The investigation, resources, data curation, writing—original draft preparation, writing—review and editing, and visualization were conducted by V.L.V. and Y.A. The conceptualization, and software were conducted by V.S. The validation, formal analysis, methodology, supervision, project administration, and funding acquisition of the version to be published were conducted by H.R.K. and P.B.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Italian Ministry of Education, University and Research through the Project "Department of Excellence LIS4.0-Lightweight and Smart Structures for In-dustry 4.0" and in part by the Horizon Marie Sklodowska-Curie Actions program (101073037).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: This work was partially supported by the Italian Ministry of Education, University and Research through the Project "Department of Excellence LIS4.0-Lightweight and Smart Structures for Industry 4.0" and in part by the Horizon Marie Sklodowska-Curie Actions program (101073037).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sanchez, G.; Alma, E.; Rivas-Araiza, E.A.; Gonzalez-Cordoba, J.L.; Toledano-Ayala, M.; Takacs, A. Blockchain mechanism and symmetric encryption in a wireless sensor network. *Sensors* **2020**, *20*, 2798. [CrossRef] [PubMed]
2. Lin, Y.; Mukhtar, H.; Huang, K.T.; Petway, J.R.; Lin, C.M.; Chou, C.F.; Liao, S.W. Real-time identification of irrigation water pollution sources and pathways with a wireless sensor network and blockchain framework. *Sensors* **2020**, *20*, 3634. [CrossRef] [PubMed]
3. Gopalakrishnan, K. Security vulnerabilities and issues of traditional wireless sensors networks in IoT. In *Principles of Internet of Things (IoT) Ecosystem: Insight Paradigm*; Springer: Cham, Switzerland, 2020; pp. 519–549.
4. Liu, Z.; Wang, D.; Wang, J.; Wang, X.; Li, H. A blockchain-enabled secure power trading mechanism for smart grid employing wireless networks. *IEEE Access* **2020**, *8*, 177745–177756. [CrossRef]
5. Maseleno, A.; Hashim, W.; Perumal, E.; Ilayaraja, M.; Shankar, K. Access control and classifier-based blockchain technology in e-healthcare applications. In *Intelligent Data Security Solutions for e-Health Applications*; Academic Press: Cambridge, MA, USA, 2020; pp. 151–167.
6. Humayun, M.; Jhanjhi, N.Z.; Hamid, B.; Ahmed, G. Emerging smart logistics and transportation using IoT and blockchain. *IEEE Internet Things Mag.* **2020**, *3*, 58–62. [CrossRef]
7. Zhang, J.; Zhong, S.; Wang, T.; Chao, C.H.; Wang, J. Blockchain-based systems and applications: A survey. *J. Internet Technol.* **2020**, *21*, 1–14.
8. Youliang, T.; Wang, Z.; Xiong, J.; Ma, J. A blockchain-based secure key management scheme with trustworthiness in DWSNs. *IEEE Trans. Ind. Inform.* **2020**, *16*, 6193–6202.
9. Ge, C.; Yin, C.; Liu, Z.; Fang, L.; Zhu, J.; Ling, H. A privacy preserve big data analysis system for wearable wireless sensor network. *Comput. Secur.* **2020**, *96*, 101887. [CrossRef]
10. Soundararajan, R.; Palanisamy, N.; Patan, R.; Nagasubramanian, G.; Khan, M.S. Secure and concealed watchdog selection scheme using masked distributed selection approach in wireless sensor networks. *IET Commun.* **2020**, *14*, 948–955. [CrossRef]
11. Nancy, P.; Muthurajkumar, S.; Ganapathy, S.; Santhosh Kumar, S.V.N.; Selvi, M.; Arputharaj, K. Intrusion detection using dynamic feature selection and fuzzy temporal decision tree classification for wireless sensor networks. *IET Commun.* **2020**, *14*, 888–895. [CrossRef]
12. Menaria, V.K.; Jain, S.C.; Raju, N.; Kumari, R.; Nayyar, A.; Hosain, E. NLFFT: A novel fault tolerance model using artificial intelligence to improve performance in wireless sensor networks. *IEEE Access* **2020**, *8*, 149231–149254. [CrossRef]
13. Kalidoss, T.; Rajasekaran, L.; Kanagasabai, K.; Sannasi, G.; Kannan, A. QoS aware trust based routing algorithm for wireless sensor networks. *Wirel. Pers. Commun.* **2020**, *110*, 1637–1658. [CrossRef]
14. Hakak, S.; Khan, W.Z.; Amin Gilkar, G.; Imran, M.; Guizani, N. Securing smart cities through blockchain technology: Architecture, requirements, and challenges. *IEEE Netw.* **2020**, *34*, 8–14. [CrossRef]
15. Simaiya, S.; Lilhore, U.K.; Sharma, S.K.; Gupta, K.; Baggan, V. Blockchain: A New Technology to Enhance Data Security and Privacy in Internet of Things. *J. Comput. Theor. Nanosci.* **2020**, *17*, 2552–2556. [CrossRef]
16. Yazdinejad, A.; Parizi, R.M.; Dehghantanha, A.; Zhang, Q.; Raymond Choo, K.K. An energy-efficient SDN controller architecture for IoT networks with blockchain-based security. *IEEE Trans. Serv. Comput.* **2020**, *13*, 625–638. [CrossRef]
17. Naveed, Q.N.; Alqahtani, H.; Ullah Khan, R.; Almakdi, S.; Alshehri, M.; Abdul Rasheed, M.A. An intelligent traffic surveillance system using integrated wireless sensor network and improved phase timing optimization. *Sensors* **2020**, *22*, 3333. [CrossRef] [PubMed]
18. Manjula, V.; Thalapathi Rajasekaran, R. Security vulnerabilities in traditional wireless sensor networks by an intern in IoT, blockchain technology for data sharing in IoT. In *Principles of Internet of Things (IoT) Ecosystem: Insight Paradigm*; Springer: Cham, Switzerland, 2020; pp. 579–597.
19. Ramasamy, L.K.; Firoz, K.K.P.; Imoize, A.L.; Ogbebor, J.O.; Kadry, S.; Rho, S. Blockchain-based wireless sensor networks for malicious node detection: A survey. *IEEE Access* **2021**, *9*, 128765–128785. [CrossRef]
20. Lee, C.C. Security and privacy in wireless sensor networks: Advances and challenges. *Sensors* **2020**, *20*, 744. [CrossRef]
21. Awan, S.; Javaid, N.; Ullah, S.; Ullah Khan, A.; Qamar, A.M.; Choi, J.G. Blockchain Based Secure Routing and Trust Management in Wireless Sensor Networks. *Sensors* **2020**, *22*, 411. [CrossRef]
22. Hsiao, S.; Sung, W.T. Employing blockchain technology to strengthen security of wireless sensor networks. *IEEE Access* **2021**, *9*, 72326–72341. [CrossRef]
23. Revanesh, M.; Sridhar, V. A trusted distributed routing scheme for wireless sensor networks using blockchain and meta-heuristics-based deep learning technique. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4259. [CrossRef]
24. Abd, E.; Ibrahim, A.; Darwish, S.M. Towards designing a trusted routing scheme in wireless sensor networks: A new deep blockchain approach. *IEEE Access* **2021**, *9*, 103822–103834.

25. Amjad, S.; Abbas, S.; Abubaker, Z.; Alsharif, M.H.; Jahid, A.; Javaid, N. Blockchain Based Authentication and Cluster Head Selection Using DDR-LEACH in Internet of Sensor Things. *Sensors* **2022**, *22*, 1972. [CrossRef] [PubMed]
26. Goyat, R.; Kumar, G.; Alazab, M.; Saha, R.; Thomas, R.; Rai, M.K. A secure localization scheme based on trust assessment for WSNs using blockchain technology. *Future Gener. Comput. Syst.* **2021**, *125*, 221–231. [CrossRef]
27. Lazrag, H.; Chehri, A.; Saadane, R.; Rahmani, M.D. Efficient and secure routing protocol based on Blockchain approach for wireless sensor networks. *Concurr. Comput. Pract. Exp.* **2021**, *33*, e6144. [CrossRef]
28. Nguyen, G.N.; Ho Le Viet, N.; Francis Saviour Devaraj, A.; Gobi, R.; Shankar, K. Blockchain enabled energy efficient red deer algorithm-based clustering protocol for pervasive wireless sensor networks. *Sustain. Comput. Inform. Syst.* **2020**, *28*, 100464. [CrossRef]
29. Almaiah, M.A. A new scheme for detecting malicious attacks in wireless sensor networks based on blockchain technology. In *Artificial Intelligence and Blockchain for Future Cybersecurity Applications*; Springer: Cham, Switzerland, 2021; pp. 217–234.
30. Ávila, K.; Sanmartin, P.; Jabba, D.; Gómez, J. An analytical survey of attack scenario parameters on the techniques of attack mitigation in WSN. *Wirel. Pers. Commun.* **2021**, *122*, 3687–3718. [CrossRef]
31. Sengupta, J.; Ruj, S.; Bit, S.D. A comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT. *J. Netw. Comput. Appl.* **2020**, *149*, 102481. [CrossRef]
32. Yudistira, R. AES (Advanced Encryption Standard) and RSA (Rivest–Shamir–Adleman) Encryption on Digital Signature Document: A Literature Review. *Int. J. Inf. Technol. Bus.* **2020**, *2*, 26–29.
33. Cui, Z.; Xue, F.; Zhang, S.; Cai, X.; Cao, Y.; Zhang, W.; Chen, J. A hybrid blockchain-based identity authentication scheme for multi-WSN. *IEEE Trans. Serv. Comput.* **2020**, *13*, 241–251. [CrossRef]
34. Raja, L.; Periasamy, P.S. A Trusted Distributed Routing Scheme for Wireless Sensor Networks Using Block Chain and Jelly Fish Search Optimizer Based Deep Generative Adversarial Neural Network (Deep-GANN) Technique. *Wirel. Pers. Commun.* **2022**, 1101–1128. [CrossRef]
35. Chou, J.S.; Truong, D.N. A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. *Appl. Math. Comput.* **2021**, *389*, 125535. [CrossRef]