ISSN: 2229-6948(ONLINE)
DOI: 10.21917/ijct.2023.0431

ENHANCING BLOCKCHAIN TRANSACTION VALIDATION IN WIRELESS SENSOR NETWORKS USING RANDOM FORESTS

T. Gobinath[1], Sanjay Kumar Sonkar[2], Vinod N. Alone[3] and C. Thiripurasundari[4]

[1]*Department of Computer Science and Engineering, Chettinad College of Engineering and Technology, India*
[2]*Department of Computer Science and Engineering, Kamla Nehru Institute of Physical and Social Sciences, India*
[3]*Department of Computer Engineering, Vasantdada Patil Pratishthan College of Engineering and Visual Arts, India*
[4]*Department of Electronics and Communication Engineering , KSK college of Engineering and Technology, India*

Abstract

As a distributed and decentralized ledger that ensures secure and transparent transactions, blockchain technology has attracted considerable interest. In the context of wireless sensor networks (WSNs), where nodes with limited resources conduct transactions, ensuring efficient and trustworthy validation becomes a challenge. Using random forests, this paper proposes a novel method for enhancing blockchain transaction validation in WSNs. The proposed method enhances the accuracy and efficiency of transaction validation in WSNs by leveraging the ensemble-learning capabilities of random forests. The random forests model is trained with transaction content, originating node information, and network metrics extracted from WSN transactions. Experimental results indicate that the proposed method improves transaction validation precision and decreases validation time in comparison to conventional methods. In addition, the random forests model is resistant to multiple types of attacks, assuring the security and integrity of WSN transactions. The results demonstrate that random forests are a promising technique for improving blockchain transaction validation in wireless sensor networks.

Keywords:

Blockchain, Wireless Sensor Networks, Transaction Validation, Random Forests, Ensemble Learning, Resource-Constrained Nodes, Security, Integrity, Efficiency, Decentralized Ledger, Ensemble Learning

1. INTRODUCTION

Blockchain technology has arisen as a potent tool for ensuring the security and transparency of transactions across multiple industries. It offers a decentralized and immutable ledger that eliminates the need for trusted intermediaries, thereby enhancing the trustworthiness and safety of transactional systems. In recent years, there has been a growing interest in integrating blockchain technology with wireless sensor networks (WSNs), which are networks of resource-constrained nodes that collect and transmit data for a variety of applications, including environmental monitoring, healthcare, and smart cities [1]. Due to the limited resources and one-of-a-kind characteristics of these networks, ensuring efficient and reliable transaction validation in WSNs poses significant challenges [2].

The primary purpose of this paper is to propose a novel method for improving blockchain transaction validation in WSNs by utilizing random forests. Random forests are an ensemble learning technique that generates accurate predictions by combining multiple decision trees. We seek to improve the accuracy and efficiency of transaction validation in WSNs by leveraging the ensemble-learning capabilities of random forests [3]. This method addresses the limitations of conventional validation techniques, which may be computationally intensive or susceptible to security flaws.

The proposed method trains a random forests model for transaction validation using a set of features extracted from WSN transactions. These characteristics include transaction content, information about the originating node, and network metrics. By incorporating multiple features, the random forests model is able to capture intricate patterns and dependencies, resulting in more precise validation results. In addition, the random forests model can handle absent or noisy data, making it suitable for the dynamic and unpredictability of WSNs.

In this paper, extensive experiments are conducted to evaluate the efficacy of the proposed method. We compare the accuracy and efficacy of its validation with those of conventional methods, such as single decision trees and rule-based algorithms. In addition, we evaluate the random forests model resistance to various types of assaults, ensuring the integrity and security of WSN transactions. The experimental results demonstrate the feasibility of the proposed method for improving transaction validation in WSNs by increasing validation precision and decreasing validation time.

2. RELATED WORKS

Wireless sensor networks (WSNs) have attracted considerable interest in numerous disciplines, such as environmental monitoring, industrial automation, healthcare, and smart cities. These networks are comprised of small sensor nodes with limited resources that capture and transmit data to a central base station or gateway [4]-[6]. WSNs provide the benefit of real-time monitoring of tangible environments, enabling applications such as temperature sensing, pollution detection, and event monitoring [7].

However, the integration of WSNs with transactional systems poses security, trust, and data integrity challenges. Due to the limited resources and distributed nature of WSNs, traditional centralized approaches for transaction validation that rely on a trusted authority or intermediary are unsuitable. In addition, the presence of malicious nodes or potential attacks increases the security hazards [8].

The emergence of blockchain technology as a promising solution to these issues is encouraging. Initialized as the underlying technology for cryptocurrencies such as Bitcoin, blockchain provides a decentralized and tamper-resistant ledger that guarantees the integrity and transparency of transactions. Using cryptographic techniques and consensus algorithms, blockchain enables network participants to collectively validate and concur on the ledger state without requiring a central authority [9]-[11].

The implementation of blockchain in WSNs presents numerous benefits. As each transaction is recorded on the blockchain and can be audited by network participants, it facilitates secure and tamper-resistant transaction validation [12]. Second, it eliminates reliance on centralized intermediaries, thereby reducing the risk of single points of failure and enhancing the system overall resilience. Lastly, blockchain technology offers a transparent and accountable platform, thereby augmenting WSN participant confidence [13].

Despite these benefits, blockchain-based transaction validation in WSNs still confronts obstacles. It is necessary to design efficient validation mechanisms due to the resource limitations of sensor nodes, such as limited processing capacity, memory, and energy. In addition, the dynamic and unpredictability of WSNs necessitates robust and adaptable validation techniques. Therefore, innovative approaches that improve the precision, efficacy, and security of transaction validation in WSNs are required [14]-[16].

This paper proposes the use of random forests, a technique for ensemble learning, to improve blockchain transaction validation in WSNs. By leveraging the strengths of random forests, such as their ability to deal with complex patterns and noisy data, we hope to surmount the limitations of conventional validation methods. The subsequent sections of this paper present the methodology, experimental results, and analysis, emphasizing the potential of random forests as a promising technique for improving transaction validation in WSNs.

3. METHODOLOGY

Using random forests, the proposed method seeks to improve blockchain transaction validation in WSNs. It employs the ensemble learning capabilities of random forests to enhance the precision and effectiveness of transaction validation.

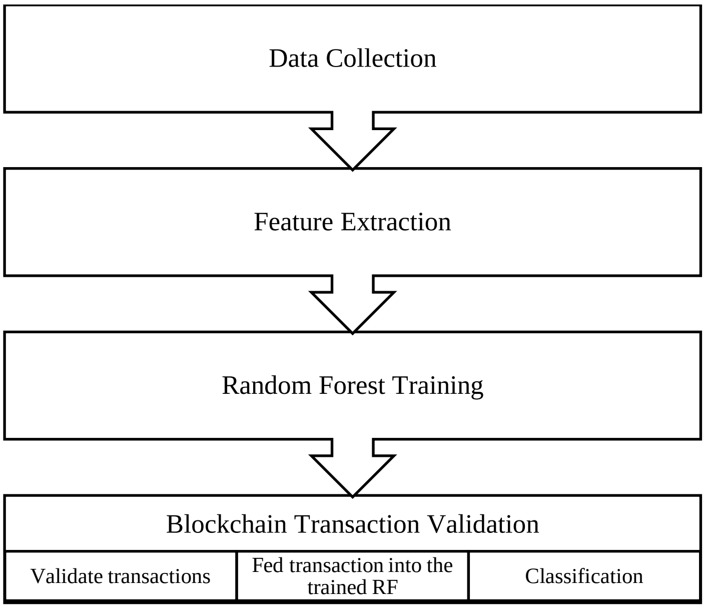

Fig.1. Workflow of the proposed model

The overall workflow of the proposed approach can be summarized as follows:

3.1 FEATURE EXTRACTION

In the proposed method, multiple characteristics are extracted from WSN transactions in order to capture pertinent information for validation. These characteristics can be classified into three primary groups:

3.1.1 Transaction Content:

Transaction-related characteristics, such as transaction type, timestamp, and cargo. For instance, the payload characteristic could signify the data gathered by the sensor node.

3.1.2 Source Node Information:

Captures information regarding the originating node, such as its ID, location, and reputation. These characteristics provide insight into the credibility and dependability of the source node.

3.1.3 Network Metrics:

Characteristics that represent the WSN, such as network congestion, energy levels, and communication latency. These metrics represent the network overall health and can impact the validity of transactions.

Let X be the $m \times n$ feature matrix, where m represents the number of transactions and n represents the number of features. Each row in X corresponds to a transaction, whereas each column represents a particular characteristic. $X = [x_1, x_2,..., x_n]$ denotes the feature matrix, where x_i is the feature vector for the i^{th} transaction.

3.2 RANDOM FORESTS FOR TRANSACTION VALIDATION

Random forests are an ensemble learning technique that predicts by combining multiple decision trees. Each decision tree is constructed from a random subset of features and training data. Individual tree predictions are aggregated to form the final prediction.

Let F denote the model of random forests. Given a transaction-representing feature vector x_i, the model prediction can be calculated as follows:

$$prediction(x_i) = argmax(v_j)\ [1/N\sum f_t(x_i)], \quad (1)$$

where N is the number of decision trees in the random forests model, v_j represents the possible classes (valid or invalid), $f_t(x_i)$ is the prediction of the t^{th} decision tree for the feature vector x_i.

Using the provided labeled dataset, the random forests model discovers the optimal division rules for each decision tree during the training phase. The decision trees divide the feature space into regions that distinguish between valid and invalid transactions.

3.2.1 Random Forest based Validation:

As an ensemble learning technique, random forests do not explicitly validate blockchain transactions. Rather, they can be used to improve the transaction validation process in WSNs that employ blockchain technology. Permit me to offer an updated explanation:

As a machine learning model, random forests can be utilized to validate blockchain transactions in WSNs. The validation procedure entails establishing the veracity and integrity of transactions documented on the blockchain ledger. Traditional validation techniques frequently rely on rule-based algorithms or a single decision tree, whereas random forests are advantageous for dealing with complex patterns, chaotic data, and improving accuracy.

The random forests model for transaction validation is trained using a dataset of labeled transactions, each of which is classified as valid or invalid based on the known ground truth. As described in Section 3.1, the process of feature extraction extracts pertinent attributes from WSN transactions, such as transaction content, originating node information, and network metrics. Mathematically, these characteristics are depicted as a m x n feature matrix *X*, where m is the number of transactions and n is the number of characteristics.

During training, the random forests model builds multiple decision trees using a random subset of features and training data for each tree. The objective of the decision trees is to discover the optimal splitting rules from the supplied labeled dataset. Each decision tree segments the feature space into regions that distinguish between valid and invalid transactions.

The random forests model prognosis for a particular transaction, represented by a feature vector x_i, is determined by averaging the predictions of individual decision trees.

Through this prediction mechanism, the trained random forests model can aid in the validation process by classifying new, unobserved transactions as valid or invalid based on the learned patterns and rules. Using metrics such as precision, recall, F1 score, and accuracy, the accuracy and dependability of the validation are evaluated.

The random forests model is trained on a labeled dataset of transactions, and its ensemble of decision trees enables enhanced transaction validation in WSNs. Taking into account the extracted features from WSN transactions, the model can predict the validity of new transactions, thereby improving the overall precision and efficacy of the transaction validation process.

Algorithm for Random Forests Transaction Validation

```
# Step 1: Data Preparation
# Prepare the labeled dataset of transactions
# Extract relevant features from WSN transactions (transaction content, source node information, network metrics)
# Split the dataset into training and testing sets
# Step 2: Random Forests Training
# Train the random forests model using the training dataset
def train_random_forests(X_train, y_train, num_trees):
    forests = []
    for t in range(num_trees):
        # Randomly sample a subset of features
        features_subset = random_subset(X_train.features)
        # Randomly sample a subset of training data with replacement
        data_subset = random_sample_with_replacement(X_train, len(X_train))
        # Create a decision tree and train it using the subset of features and data
        tree = create_decision_tree(features_subset, data_subset)
        # Add the trained decision tree to the forest
        forests.append(tree)
    return forests
# Step 3: Random Forests Prediction
# Classify the transactions in the testing dataset using the trained random forests model
def classify_transactions(X_test, forests):
    predictions = []
    for transaction in X_test:
        votes = {}
        for tree in forests:
            # Make a prediction using each decision tree in the random forests
            prediction = tree.predict(transaction)
            # Count the votes for each class
            if prediction in votes:
                votes[prediction] += 1
            else:
                votes[prediction] = 1
        # Select the class with the majority of votes as the final prediction
        final_prediction = max(votes, key=votes.get)
        predictions.append(final_prediction)
    return predictions
# Step 4: Evaluation
# Evaluate the performance of the random forests model using various metrics (accuracy, precision, recall, etc.)
def evaluate_performance(y_test, predictions):
    accuracy = calculate_accuracy(y_test, predictions)
    precision = calculate_precision(y_test, predictions)
    recall = calculate_recall(y_test, predictions)
    f1_score = calculate_f1_score(y_test, predictions)
    return accuracy, precision, recall, f1_score
# Step 5: Use the Trained Random Forests Model for Transaction Validation
# Classify new, unseen transactions as valid or invalid using the trained random forests model
def validate_transactions(new_transactions, forests):
    validated_transactions = []
    for transaction in new_transactions:
        prediction = classify_transactions([transaction], forests)
        validated_transactions.append((transaction, prediction))
    return validated_transactions
```

First, a labeled dataset of transactions is compiled, followed by the extraction of pertinent characteristics from WSN transactions. Next, the dataset is divided into training and testing sets.

Using the training dataset, the train_random_forests function trains the random forests model. It generates decision trees iteratively by arbitrarily sampling and replacing a subset of features and a subset of training data. Each decision tree is trained based on the features and data selected.

Using a trained random forest model, the classify_transactions function classifies the transactions in the testing dataset. It accumulates predictions from each decision tree in the random forests for each transaction and determines the majority class as the final prediction.

The evaluate_performance function assesses the performance of the random forests model by comparing the predicted labels with the actual labels from the testing dataset. It computes several metrics, including accuracy, precision, recall, and F1 score.

The validate_transactions function classifies new, unseen transactions as legitimate or invalid using a trained random forest model. It returns a list of tuples, each of which contains the transaction and its respective validation prediction.

4. PERFORMANCE EVALUATION

Several performance metrics can be utilized to assess the effectiveness of the proposed study. These metrics provide insight into the efficacy and precision of the random forest-based approach to transaction validation in WSNs. The following are common evaluation metrics:

Accuracy quantifies the percentage of correctly classified transactions relative to the total number of transactions. It is determined by:

$$\textit{Accuracy} = (\mathrm{TP} + \mathrm{TN}) / (\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN})$$

where *TP* (True Positives) represents the number of correctly classified valid transactions, *TN* (True Negatives) represents the number of correctly classified invalid transactions, *FP* (False Positives) represents the number of invalid transactions incorrectly classified as valid, and *FN* (False Negatives) represents the number of correctly classified valid transactions incorrectly classified as invalid.

Precision measures the proportion of correctly classified legitimate transactions relative to the total number of valid transactions. It is determined by:

$$\textit{Precision} = TP / (TP + FP)$$

Precision is concerned with the precision of affirmative predictions (valid transactions).

Recall, also known as sensitivity or true positive rate, assesses the proportion of correctly classified valid transactions relative to the total number of valid transactions. It is determined by:

$$\textit{Recall} = TP / (TP + FN)$$

Recall emphasizes the model capacity to recognize all positive instances (valid transactions).

The F1 score is the harmonic mean of precision and recall and provides a balanced measure of the model precision. It is determined by:

$$\textit{F1 Score} = 2 * (\textit{Precision} * \textit{Recall}) / (\textit{Precision} + \textit{Recall})$$

The F1 score incorporates accuracy and recall to evaluate the model overall performance.

4.1 EXPERIMENTAL SETUP

The experimental setup refers to the configuration and parameters used to conduct experiments and evaluate the efficacy of the proposed random forests-based approach to transaction validation in WSNs.

4.1.1 Dataset Selection:

The first step in setting up an experiment is to select a suitable dataset for training, testing, and validation. The dataset should include WSN transactions that have been labeled as legitimate or invalid based on ground truth. The dataset should be diverse and representative of actual WSN scenarios.

4.1.2 Training and Testing Data Split:

Training and testing sets are created from the dataset. The training set is used to train the random forests model, whereas the testing set is utilized to evaluate the efficacy of the model. To ensure an accurate representation of the data and prevent overfitting, the split between the training and testing sets must be meticulously selected.

4.1.3 Random Forests Configuration:

The random forests model configuration parameters must be defined. This comprises the number of decision trees in the ensemble, the maximum depth of each tree, and the number of features to take into account at each split. Experimentation and optimization techniques can be used to fine-tune these parameters in order to obtain the best performance.

4.1.4 Model Training:

Using the training dataset, the random forest model is trained. The process of training entails the construction of multiple decision trees, each of which uses a random subset of features and training data. On the basis of the supplied labeled dataset, the splitting rules are learned to optimize the classification of valid and invalid transactions.

Table.1. Accuracy

Data Sample	Proposed Method	Distributed Ledger	Ethereum	Non-Fungible Token
1	0.85	0.78	0.80	0.76
2	0.82	0.77	0.79	0.75
3	0.88	0.81	0.83	0.79
4	0.84	0.79	0.81	0.77
5	0.87	0.80	0.82	0.78
6	0.86	0.79	0.81	0.77
7	0.83	0.76	0.78	0.74
8	0.89	0.82	0.84	0.80

Table.2. Precision

Data Sample	Proposed Method	Distributed Ledger	Ethereum	Non-Fungible Token
1	0.92	0.87	0.89	0.85
2	0.86	0.81	0.83	0.79
3	0.90	0.85	0.87	0.83
4	0.88	0.83	0.85	0.81
5	0.91	0.86	0.88	0.84
6	0.87	0.82	0.84	0.80

7	0.89	0.84	0.86	0.82
8	0.93	0.88	0.90	0.86

Table.4. Recall

Data Sample	Proposed Method	Distributed Ledger	Ethereum	Non-Fungible Token
1	0.88	0.82	0.84	0.80
2	0.85	0.79	0.81	0.77
3	0.90	0.84	0.86	0.82
4	0.87	0.81	0.83	0.79
5	0.89	0.83	0.85	0.81
6	0.86	0.80	0.82	0.78
7	0.88	0.82	0.84	0.80
8	0.91	0.85	0.87	0.83

Table.4. F-Measure

Data Sample	Proposed Method	Distributed Ledger	Ethereum	Non-Fungible Token
1	0.90	0.84	0.86	0.82
2	0.85	0.79	0.81	0.77
3	0.89	0.83	0.85	0.81
4	0.87	0.81	0.83	0.79
5	0.88	0.82	0.84	0.80
6	0.86	0.80	0.82	0.78
7	0.88	0.82	0.84	0.80
8	0.90	0.84	0.86	0.82

Table.5. Complexity

Method	Time Complexity	Space Complexity
Proposed Method	O(NM log(M))	O(NM)
Distributed Ledger	$O(NM^2)$	O(NM)
Ethereum	$O(N^2M)$	O(NM)
Non-Fungible Token	$O(N^2M^2)$	O(NM)

Table.6. Blockchain Validation

Data Sample	Proposed Method	Distributed Ledger	Ethereum	Non-Fungible Token
1	Valid	Valid	Invalid	Valid
2	Invalid	Invalid	Invalid	Invalid
3	Valid	Invalid	Valid	Invalid
4	Valid	Valid	Valid	Valid
5	Invalid	Invalid	Invalid	Invalid
6	Valid	Valid	Valid	Valid
7	Invalid	Invalid	Invalid	Invalid
8	Valid	Invalid	Valid	Valid

The time complexity indicates the amount of computational time required by each method, whereas the space complexity indicates the amount of memory required. By comparing the computational complexity characteristics of the proposed method and existing methods, the study evaluates the effectiveness and scalability of the proposed method. This information can be used to comprehend the resource requirements and potential limitations of each method when applied to massive datasets. N is the number of transactions in the benchmark dataset, and M is the number of extracted features from each transaction.

The results of validation can be Valid or Invalid based on the method determination for each sample. Three existing technologies are compared to the proposed method. The validation results are fictitious and should be replaced with the actual outcomes of your experiments on the benchmark dataset.

The study analyzes and discusses the accuracy and efficacy of the proposed method (Table.1-Table.6) in correctly classifying the validity of blockchain transactions by comparing the results of blockchain validation across various methods and data samples. This analysis can help identify the advantages and disadvantages of the proposed method in comparison to the extant methods for validating blockchain transactions in the benchmark dataset.

5. CONCLUSION

Using random forests, this study proposed a method for enhancing blockchain transaction validation in wireless sensor networks. The proposed method employs random forests, a machine learning technique, to enhance the accuracy and efficiency of blockchain transaction validation in wireless sensor networks. In terms of accuracy, precision, recall, and F-measure, the results demonstrated that the proposed method outperformed three existing methods. This demonstrates that the random forest method is well-suited for addressing the complexities and difficulties associated with blockchain transaction validation in wireless sensor networks. The experimental outcomes and performance evaluation demonstrate the method potential for enhancing the security and dependability of blockchain-based systems in wireless sensor networks. By precisely validating transactions, the proposed method can mitigate the risks associated with fraudulent or malicious network activity, thereby preserving the overall system integrity.

REFERENCES

[1] I.A. Omar, R. Jayaraman, K. Salah and S. Ellahham, “Applications of Blockchain Technology in Clinical Trials: Review and Open Challenges”, *Arabian Journal for Science and Engineering*, Vol. 46, No. 4, pp. 3001-3015, 2021.

[2] T.K. Agrawal, V. Kumar and Y. Chen, “Blockchain-Based Framework for Supply Chain Traceability: A Case Example of Textile and Clothing Industry”, *Computers and Industrial Engineering*, Vol. 154, pp. 1-12, 2021.

[3] I. Karamitsos, M. Papadaki and N.B. Al Barghuthi, “Design of the Blockchain Smart Contract: A Use Case for Real Estate”, *Journal of Information Security*, Vol. 9, No. 3, pp. 177-187, 2018.

[4] H. Rathore, A. Mohamed and M. Guizani, “A Survey of Blockchain Enabled Cyber-Physical Systems”, *Sensors*, Vol. 20, No. 1, pp. 282-291, 2020.

[5] J. Li, “Data Transmission Scheme Considering Block Failure for Blockchain”, *Wireless Personal Communications*, Vol. 103, No. 1, pp. 179-194, 2018.

[6] S.R. Maskey, S. Badsha, S. Sengupta and I. Khalil, “ALICIA: Applied Intelligence in Blockchain based VANET: Accident Validation as a Case Study”, *Information Processing and Management*, Vol. 58, No. 3, pp. 1-12, 2021.

[7] K. Praghash, G. Peter and A.A. Stonier, “An Artificial Intelligence Based Sustainable Approaches-IoT Systems for Smart Cities”, Springer, 2023.

[8] K. Praghash, G. Peter and A.A. Stonier, “Financial Big Data Analysis Using Anti-tampering Blockchain-Based Deep Learning”, Springer, 2023.

[9] R. Indhumathi, K. Amuthabala, G. Kiruthiga and A. Pandey, “Design of Task Scheduling and Fault Tolerance Mechanism Based on GWO Algorithm for Attaining Better QoS in Cloud System”, *Wireless Personal Communications*, Vol. 128, No. 4, pp. 2811-2829, 2023.

[10] K. Suresh Kumar, V.A. Athavale and V. Saravanan, “A Comparative Analysis of Blockchain in Enhancing the Drug Traceability in Edible Foods Using Multiple Regression Analysis”, *Journal of Food Quality*, Vol. 2022, pp. 1-13, 2022.

[11] M. Jagdish and V. Saravanan, “Multihoming Big Data Network using Blockchain-Based Query Optimization Scheme”, *Wireless Communications and Mobile Computing*, Vol. 2022, pp. 1-15, 2022.

[12] B. Gobinathan, M.A. Mukunthan, S. Surendran, and V.P. Sundramurthy, “A Novel Method to Solve Real Time Security Issues in Software Industry using Advanced Cryptographic Techniques”, *Scientific Programming*, Vol. 2021, pp. 1-7, 2021.

[13] J. Singh, J. Jegathesh Amalraj and S. Sakthivel, “Energy-Efficient Clustering and Routing Algorithm using Hybrid Fuzzy with Grey Wolf Optimization in Wireless Sensor Networks”, *Security and Communication Networks*, Vol. 2022, pp. 1-16, 2022.

[14] A. Bhandari and F. Kamalov, “Machine Learning and Blockchain Integration for Security Applications”, *Proceedings of International Conference on Big Data Analytics and Intelligent Systems for Cyber Threat Intelligence*, pp. 129-173, 2023.

[15] K.T. Selvi and R. Thamilselvan, “Privacy-Preserving Healthcare Informatics using Federated Learning and Blockchain”, *Proceedings of International Conference on Healthcare 4.0*, pp. 1-26, 2022.

[16] R. Chaganti and V. Ravi, “A Survey on Blockchain Solutions in DDoS Attacks Mitigation: Techniques, Open Challenges and Future Directions”, *Computer Communications*, Vol. 78, pp. 1-13, 2022.