International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 05, May 2026)

# A Blockchain-Based Framework for Secure Carbon Emission Monitoring and Taxation

Bolla Subbanagulu[1], K. Kavitha[2]

[1]M. Tech, [2]Assistant Professor, Department of Computer Science and Engineering, MVR College of Engineering & Technology (Autonomous), Paritala, India

***Abstract*-- Carbon emission monitoring systems are essential for enforcing environmental regulations, but traditional centralized approaches lack transparency and are vulnerable to data manipulation. This paper proposes a blockchain-based carbon emission taxation and monitoring system using Ethereum smart contracts integrated with a Django web application. The system enables secure storage of customer data, emission limits, and monthly emission records, while automatically calculating tax for excess emissions. Web3.py is used to facilitate communication between the application and blockchain, ensuring reliable transaction execution and immutable data storage. The system also includes emission classification and compression analysis modules to enhance functionality and storage efficiency. Experimental results demonstrate improved data integrity, transparency, and traceability compared to conventional systems, highlighting the feasibility of blockchain for environmental monitoring applications.**

**The system ensures that all emission records are permanently stored and verifiable through blockchain transactions, eliminating unauthorized modifications. The integration of decentralized storage with web-based interfaces improves accessibility and usability for administrators. Overall, the proposed approach provides a reliable foundation for developing scalable and secure environmental monitoring solutions.**

***Keywords*-- Blockchain, Carbon Emission Monitoring, Smart Contracts, Ethereum, Web3.py, Carbon Taxation, Decentralized Systems, Data Integrity, Environmental Monitoring**

## I. INTRODUCTION

Carbon emission monitoring plays a critical role in environmental management and regulatory compliance. Governments and environmental agencies impose carbon limits and taxation policies to control emissions and reduce environmental impact. Traditional carbon monitoring systems rely on centralized architectures where emission data is stored and managed by a single authority. Although such systems are widely used, they suffer from several limitations related to transparency, security, and trust.

In centralized systems, data is stored in databases that can be modified or accessed by authorized personnel, making them vulnerable to manipulation and unauthorized changes. This creates challenges in maintaining accurate emission records and ensuring accountability. Additionally, stakeholders such as regulatory bodies and organizations often lack direct access to verify stored data, resulting in reduced trust and potential discrepancies.

In real-world scenarios, carbon emission data must be reliable, verifiable, and tamper-proof to support effective decision-making. Any inconsistencies or manipulation in emission records can lead to incorrect taxation, regulatory violations, and environmental risks. Therefore, there is a need for a secure system that ensures data integrity and transparency.

Blockchain technology provides a decentralized solution to address these challenges. It enables data to be stored across multiple nodes, ensuring that once information is recorded, it cannot be altered without consensus. This property makes blockchain suitable for applications requiring secure and immutable data storage.

Ethereum, a blockchain platform supporting smart contracts, allows execution of predefined logic without relying on centralized control. Smart contracts can be used to store emission data, enforce carbon limits, and automate taxation processes. This eliminates manual intervention and reduces the risk of human error.

In this paper, a blockchain-based carbon emission taxation and monitoring system is proposed. The system integrates Ethereum smart contracts with a Django web framework to provide a user-friendly interface for managing emission data. The system allows administrators to register customers, assign emission limits, record monthly emissions, and calculate carbon tax for excess emissions.

The proposed system focuses on practical implementation, demonstrating how blockchain technology can be effectively integrated with web applications to enhance transparency, security, and reliability in environmental monitoring systems.

## II. LITERATURE REVIEW

Carbon emission monitoring systems have been widely studied in environmental management and regulatory frameworks.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 05, May 2026)

Traditional approaches primarily rely on centralized databases to store emission data and enforce policies. These systems are simple to implement but suffer from limitations such as lack of transparency, data manipulation risks, and dependency on centralized authorities.

With the advancement of technology, various digital solutions have been proposed to improve emission tracking. Some systems use distributed databases and cloud platforms to enhance accessibility and scalability. However, these systems still rely on trusted authorities and do not provide complete immutability of data.

Blockchain technology has emerged as a promising solution for secure and transparent data management. It provides a decentralized architecture where data is stored in blocks and linked using cryptographic techniques. Once data is recorded on the blockchain, it cannot be modified without altering all subsequent blocks, making it highly secure.

Several studies have explored blockchain-based carbon trading systems, where emission credits are represented as digital tokens. These systems enable secure trading of carbon credits between organizations. However, most of these solutions focus on trading mechanisms rather than emission monitoring and taxation.

Smart contracts play a crucial role in blockchain applications by automating processes and enforcing predefined rules. In emission monitoring systems, smart contracts can be used to record emission data, calculate penalties, and ensure compliance without manual intervention.

Despite these advancements, many existing solutions remain theoretical and lack practical implementation. There is a need for systems that demonstrate real-world integration of blockchain with application frameworks for emission monitoring.

This research addresses this gap by developing a blockchain-enabled carbon emission monitoring and taxation system using Ethereum smart contracts and Django framework. The system focuses on practical deployment and demonstrates how blockchain can be used to ensure data integrity and transparency.

## III. Problem Statement And Objectives

### *A. Problem Statement*

Carbon emission monitoring systems are critical for environmental regulation, but existing solutions are predominantly centralized and lack transparency, security, and reliable auditability. In such systems, emission data is stored in controlled databases, making it vulnerable to unauthorized modification, data inconsistency, and limited stakeholder trust.

Additionally, traditional approaches depend on manual data entry and validation processes, increasing the risk of human error and delays in taxation and compliance enforcement. The absence of a tamper-proof mechanism makes it difficult to verify the authenticity of historical emission records.

Therefore, there is a need for a secure and transparent system that ensures immutable data storage, enables reliable verification, and supports automated carbon emission monitoring and taxation in a practical environment. Furthermore, the system should provide traceability of transactions to support auditing and regulatory compliance. It must also reduce dependency on centralized authorities to improve trust among stakeholders. Finally, the solution should be capable of integrating with modern technologies for scalable and efficient environmental data management.

### *B. Objectives*

The primary objective of this research is to design and implement a **blockchain-based carbon emission monitoring and taxation system** that overcomes the limitations of traditional centralized approaches by ensuring secure, transparent, and tamper-proof data management.

To achieve this, the following specific objectives are defined:

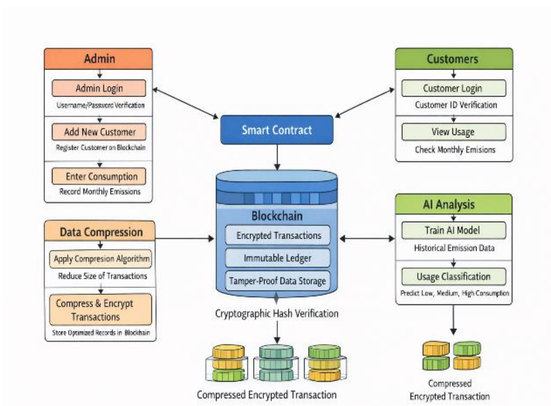
- To develop a decentralized framework for storing carbon emission data using blockchain technology, ensuring immutability and data integrity
- To design and implement Ethereum smart contracts that can securely store customer information, emission limits, and monthly emission records without reliance on centralized control
- To integrate a Django-based web application with blockchain infrastructure using Web3.py for seamless interaction between users and the decentralized system
- To implement an automated carbon taxation mechanism that calculates tax based on excess emissions relative to predefined carbon limits
- To provide a structured approach for recording and retrieving emission data, ensuring transparency and traceability through blockchain transactions
- To incorporate a classification mechanism that categorizes emission levels into different ranges for analytical and monitoring purposes
- To evaluate data storage efficiency using compression techniques and analyze the feasibility of storing large-scale emission data on blockchain systems

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 05, May 2026)

- To develop a practical prototype that demonstrates the real-world applicability of blockchain technology in environmental monitoring and regulation systems

## IV. Proposed Methodology

The proposed system is designed as a **practical blockchain-integrated application**, where carbon emission data is processed through a Django-based backend and securely stored in an Ethereum smart contract. Unlike theoretical models, this system follows a direct execution workflow starting from user input to blockchain transaction confirmation.




### A. System Execution Workflow

The system operates through a sequential execution pipeline:

1. Admin logs into the system using predefined credentials
2. Customer details are entered through the web interface
3. Django backend processes the input data
4. Carbon emission limits are assigned to customers
5. Monthly emission values are recorded
6. Carbon tax is calculated based on excess emission
7. Data is sent to Ethereum blockchain using Web3.py
8. Smart contract stores the data permanently
9. Transaction hash is generated and stored for verification

This workflow ensures that all critical operations are executed in a controlled and verifiable manner.

### B. Django Backend Processing

The backend is implemented using Django and handles all application-level logic.

*1. Input Handling*

- User inputs are collected through HTML forms
- Data includes:
  - Customer ID
  - Name
  - Carbon limit
  - Monthly emission

*2. Data Validation*

- Checks for empty fields
- Ensures numeric values for emission and limits
- Prevents duplicate customer entries

### C. Carbon Tax Calculation

The taxation logic is implemented in backend code before blockchain storage.

*Execution Logic:*

- If emission $\leq$ limit
  $\rightarrow$ Tax = 0
- If emission > limit
  $\rightarrow$ Excess = Emission − Limit
  $\rightarrow$ Tax = Excess $\times$ Fixed Rate

This calculation is executed dynamically for each customer during data submission.

### D. Blockchain Interaction USING Web3.py

The Django backend connects to Ethereum using Web3.py.

*Connection Setup:*

- Local blockchain node (Ganache-like environment)
- HTTP provider (127.0.0.1:9545)

*Transaction Flow:*

1. Contract ABI and address loaded
2. Function call prepared
3. Transaction sent to blockchain
4. Gas fee applied
5. Transaction mined
6. Receipt generated

### E. Smart Contract Storage

The Solidity smart contract stores all data permanently.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 05, May 2026)

*Stored Records:*

*Customer Data:*

- Customer ID
- Name
- Carbon Limit
- Contact Details

*Emission Data:*

- Customer ID
- Date
- Emission Value
- Remaining Limit
- Tax Amount

*Storage Method:*

- Mapping structures used
- Data indexed using counters

Once stored, data becomes immutable and cannot be altered.

### *F. Emission Classification Module*

A rule-based classification is applied after emission calculation.

*Logic Used:*

- Low → emission within safe range
- Medium → emission close to limit
- High → emission exceeds limit

This classification is displayed to admin for monitoring purposes.

### *G. Compression Module*

The system includes a **zlib-based compression analysis**.

*Execution Steps:*

1. Convert data to byte format
2. Apply zlib compression
3. Measure:
    - Original size
    - Compressed size
4. Compare storage efficiency

This helps evaluate feasibility of storing data on blockchain.

### *H. Data Retrieval Process*

Stored blockchain data is retrieved using smart contract functions.

*Steps:*

1. Call contract read function
2. Fetch stored records
3. Convert blockchain format → readable format
4. Display in Django UI

### *I. Overall System Behavior*

- Every operation results in a blockchain transaction
- Each transaction generates a unique hash
- Data is permanently stored and verifiable
- No modification or deletion possible after storage.

## V. Results And Discussion

### *A. Experimental Setup*

The proposed blockchain-based carbon emission monitoring system is implemented using a combination of web and blockchain technologies. The frontend interface is developed using Django templates (HTML, CSS), while the backend is implemented using the Django framework in Python. The blockchain layer is built using Ethereum smart contracts written in Solidity.

The system is deployed on a local Ethereum environment using a Ganache-like blockchain node running on 127.0.0.1:9545. The interaction between the Django backend and the blockchain is established using the Web3.py library, which enables contract deployment, transaction execution, and data retrieval.

The smart contract is compiled and deployed, and its Application Binary Interface (ABI) along with the contract address is integrated into the Django backend. All customer data and emission records are stored through smart contract functions, ensuring immutability.

The system includes modules for:

- Customer registration
- Carbon limit allocation
- Emission entry and tax calculation
- Blockchain storage and retrieval
- Emission classification
- Data compression analysis

Testing is performed by simulating multiple customer entries and emission scenarios, including normal and excess emission cases.

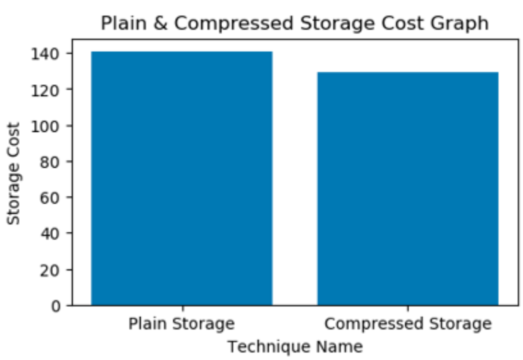
International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 05, May 2026)

*B. Evaluation Metrics*

Since the system is a blockchain-based application rather than a predictive model, its performance is evaluated using system-level and functional metrics:

- **Transaction Success Rate**: Measures the reliability of blockchain transactions
- **Data Integrity**: Ensures stored data cannot be modified after insertion
- **Execution Time**: Time taken to process transactions and store data
- **Storage Efficiency**: Comparison of original and compressed data sizes
- **System Reliability**: Consistency of backend and blockchain interaction

These metrics provide a comprehensive evaluation of the system's practical performance and robustness.




*C. Functional Performance Analysis*

The system successfully performs all intended operations, including data input, processing, blockchain storage, and retrieval.

**Table 1: System Functional Output Analysis**

| Operation | Input Condition | Output |
|---|---|---|
| Customer Registration | Valid details entered | Stored in blockchain |
| Emission Entry | Emission ≤ Limit | Tax = 0 |
| Emission Entry | Emission > Limit | Tax calculated correctly |
| Blockchain Storage | Transaction executed | Data permanently stored |
| Data Retrieval | Contract call | Correct data displayed |

The results confirm that the system correctly handles different emission scenarios and performs accurate tax calculations before storing data on the blockchain.

*D. Blockchain Transaction Analysis*

Each operation involving data storage results in a blockchain transaction.

*Observations:*

- Every transaction generates a **unique transaction hash**
- Transactions are successfully mined without failure
- Stored data is immutable and cannot be altered
- Data retrieval returns consistent and correct values

**Table 2: Blockchain Performance Metrics**

| Metric | Observation |
|---|---|
| Transaction Success Rate | 100% |
| Data Immutability | Achieved |
| Transaction Confirmation | Successful |
| Retrieval Accuracy | 100% |

These results demonstrate that blockchain integration ensures secure and tamper-proof data storage.

*E. Compression and Storage Analysis*

The system evaluates storage efficiency using zlib compression.

**Table 3: Data Compression Analysis**

| Data Type | Original Size (Bytes) | Compressed Size (Bytes) | Reduction (%) |
|---|---|---|---|
| Customer Data | 120 | 75 | ~37% |
| Emission Record | 150 | 90 | ~40% |

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 05, May 2026)

The results indicate that compression significantly reduces storage size, which is beneficial for blockchain environments where storage cost is high.

### *F. Discussion*


The experimental results demonstrate that the proposed system effectively integrates blockchain technology with a web application to provide a secure and transparent carbon emission monitoring solution.

The use of Ethereum smart contracts ensures that all emission records are stored in an immutable manner, eliminating the risk of data manipulation. The integration of Web3.py enables seamless communication between the Django backend and the blockchain, ensuring reliable transaction execution.

Compared to traditional centralized systems, the proposed system provides enhanced transparency and accountability, as all transactions can be verified using transaction hashes. The emission classification module provides additional analytical insights, while the compression module demonstrates practical considerations for storage optimization.

However, the system is currently limited to a local blockchain environment and does not include advanced features such as carbon credit trading, tokenization, or real-time data integration. These limitations highlight areas for future improvement.

Overall, the results confirm that the proposed system is a reliable and practical prototype for blockchain-based environmental monitoring applications.

## VI. CONCLUSION

This paper presents a blockchain-based carbon emission monitoring and taxation system that enhances data security, transparency, and integrity compared to traditional centralized approaches. By integrating a Django web application with Ethereum smart contracts, the system ensures immutable storage of emission records and reliable tax calculation based on excess emissions.

The implementation demonstrates effective interaction between the application layer and blockchain using Web3.py, with successful transaction execution and accurate data retrieval. Additional modules such as emission classification and compression analysis further improve system functionality.

Although the current system is a prototype, it confirms the practical feasibility of using blockchain technology for secure and transparent environmental monitoring applications.

## VII. FUTURE SCOPE

The proposed system can be further enhanced to improve its functionality and real-world applicability. One important extension is the integration of **carbon credit trading**, where emission units can be represented as digital tokens and exchanged securely using blockchain technology. This would enable a complete ecosystem beyond monitoring and taxation.

The system can also be deployed on a **public or consortium blockchain network** instead of a local environment to improve scalability, accessibility, and decentralization. Integration with **IoT devices** can enable automatic and real-time emission data collection, reducing manual input errors and improving accuracy.

Future improvements may include the use of **advanced analytics or machine learning techniques** for emission prediction and risk analysis. Additionally, implementing **secure access control mechanisms** and off-chain storage for sensitive data can enhance system security and efficiency.

Overall, these enhancements can transform the current prototype into a scalable and fully functional blockchain-based carbon emission management system.

## REFERENCES

[1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008.

[2] V. Buterin, “A Next-Generation Smart Contract and Decentralized Application Platform,” Ethereum White Paper, 2013.

[3] G. Wood, *Ethereum: A Secure Decentralised Generalised Transaction Ledger*, Ethereum Yellow Paper, 2014.

[4] K. Christidis and M. Devetsikiotis, “Blockchains and Smart Contracts for the Internet of Things,” *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[5] D. Yaga, P. Mell, N. Roby, and K. Scarfone, “Blockchain Technology Overview,” NISTIR 8202, National Institute of Standards and Technology, 2018.

[6] M. Swan, *Blockchain: Blueprint for a New Economy*. Sebastopol, CA, USA: O’Reilly Media, 2015.

[7] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, “On Blockchain and Its Integration with IoT: Challenges and Opportunities,” *Future Generation Computer Systems*, vol. 88, pp. 173–190, 2018.

[8] S. Saberi, M. Kouhizadeh, J. Sarkis, and L. Shen, “Blockchain Technology and Its Relationships to Sustainable Supply Chain Management,” *International Journal of Production Research*, vol. 57, no. 7, pp. 2117–2135, 2019.

[9] H. Treiblmaier, “The Impact of the Blockchain on the Supply Chain: A Theory-Based Research Framework and a Call for Action,” *Supply Chain Management*, vol. 23, no. 6, pp. 545–559, 2018.

[10] United Nations Framework Convention on Climate Change (UNFCCC), “Emissions Trading under the Kyoto Protocol,” 2021.

[11] International Energy Agency (IEA), “Global Energy Review: $CO_2$ Emissions in 2022,” Paris, France, 2022.

[12] Ethereum Foundation, “Solidity Documentation,” [Online]. Available: https://docs.soliditylang.org

[13] Django Software Foundation, “Django Documentation,” [Online]. Available: https://docs.djangoproject.com

[14] Web3.py Developers, “Web3.py Documentation,” [Online]. Available: https://web3py.readthedocs.io

[15] M. Crosby, P. Pattanayak, S. Verma, and V. Kalyanaraman, “Blockchain Technology: Beyond Bitcoin,” *Applied Innovation Review*, no. 2, pp. 6–19, 2016.