Physical Education, Health and Social Sciences

https://journal-of-social-education.org E-ISSN: 2958-5996
P-ISSN: 2958-5988

Minimization of Ethereum Transaction Fees Using AI and Compression Techniques

Ramia Arshad[1]*, Dr. Muhammad Milhan Afzal Khan[2], Saeed Rasheed[3], Irtaza Ijaz[4], Muhammad Irfan Ali[5]

[1,2,3,4,5] Department of Computer Science, University of Agriculture Faisalabad, Pakistan,
Email: [1] rarshad@gmail.com, [2] milhankhan@uaf.edu.pk, [3] saeedneoteric786@gmail.com, [4] irtaza.ijaz@gmail.com, [5] irfanali059@gmail.com

*Corresponding Author, Dr. Muhammad Milhan Afzal Khan, Email: milhankhan@uaf.edu.pk

DOI: https://doi.org/10.63163/jpehss.v3i4.761

Abstract:
Blockchain technology has transformed decentralized data exchange and digital payments but the consistently high gas prices pose a significant challenge to its scalability and efficiency. This research explores the role of AI-driven gas price prediction and data compression methods on gas utilization in blockchain systems with special emphasis on Ethereum transactions. Using actual Ethereum transaction history, we compare the performance of compressed versus uncompressed payloads with three different compression algorithms: Zlib, Brotli, and Gzip. Beyond that, a linear regression model is also trained to forecast hourly gas Price fluctuations given past transaction history. The methodology includes thorough statistical analysis to provide accurate and reproducible results. Our results show that compressing text data over 141 bytes using the Zlib algorithm prior to making transactions on the Ethereum network decreases the amount of gas Used without altering system time. This validates the efficiency of combining data compression with gas price forecasting in minimizing transaction costs without affecting performance. Moreover, our study further encompasses investigation of actual gas Price trends and provides real-world insights for optimizing timing strategies for economic transaction execution. These results enhance the knowledge of Ethereum gas dynamics and provide valuable solutions for enhancing economic efficiency and resource utilization in applications based on blockchain. Future efforts will involve applying the framework to the Ethereum mainnet, using deep learning models for increased prediction accuracy, and adaptive compression dependent on network state and transaction size.

Keywords: Ethereum, Gas Fee Optimization, Linear Regression, Brotli, ZLIB, GZIP, Blockchain, Compression, Smart Contracts

Introduction

Blockchain's decentralization, security, and transparency have resulted in its use across sectors such as cloud-edge computing and IoT. The area still needs addressing is constrained transaction throughput in networks such as Ethereum and Bitcoin (7–15 TPS). Some of the proposed solutions include State Channels, Side Chains, Roll-ups, and novel consensus algorithms such as Conflux (up to 6400 TPS), which are seeking to enhance scalability, while network bandwidth continues to constrain performance (Viéitez et al., 2024). Blockchain is a decentralized public book that securely documents non-reversible transactions, which are transparent without a central authority. Blockchain is widely applied beyond cryptocurrencies, particularly in Ethereum-based decentralized applications, digital assets, and IoT security Luo et al., (2020). Ethereum, initiated by Vitalik Buterin in 2015, is a decentralized blockchain platform for running smart contracts and decentralized applications (DApps). Unlike Bitcoin, which mainly operates as digital money, Ethereum features a nearly Turing-complete programming language to support intricate logic and automation. Ethereum has Ether (ETH) as its cryptocurrency, whose transactions are secured by cryptographic keys and nonces. Initially, it is founded on Proof of Work (PoW), assuring decentralized consensus and system integrity. Its elasticity of architecture positions it as a platform for such innovations as decentralized finance (DeFi) and digital assets management (Laurent et al., (2022). Ethereum, the second-largest blockchain in terms of market cap ($452B, current as of May 2024), facilitates smart contracts, NFTs, DeFi, and DApps beyond Bitcoin's limited versatility. Its London upgrade (EIP-1559) saw the implementation of a base fee system with optional tips for gas fees. The 2022 Merge transitioned Ethereum from proof-of-work to proof-of-stake, cutting energy consumption by more than 99% and increasing speed, scalability, and efficiency of transaction costs (Karaivanov and Zarifian, 2024). Ethereum charges users a fee for gas in order to run transactions and smart contracts. Gas is the computation unit used to measure the cost of a transaction, which is as follows:

$$\text{Transaction Fee} = \text{gas cost} \times \text{gas price.}$$

Miners prioritize transactions with higher gas prices, making fee estimation critical. Smart contract complexity and storage use directly influence gas consumption. Although some tools like GASOL help optimize gas usage, further algorithmic improvements are still needed for effective fee management (Ta and Do, 2024). Miners prioritize transactions with higher gas prices, making fee estimation critical. Smart contract complexity and storage use directly influence gas consumption. Although some tools like GASOL help optimize gas usage, further algorithmic improvements are still needed for effective fee management (Ta and Do, 2024) In blockchain systems like Ethereum, each transaction carries data that contributes to the overall transaction size and, consequently, the gas consumed. To lower transaction expenses, data compression mechanisms have been researched as a viable option. Compression compresses textual or structured data prior to submission to the blockchain, creating smaller bytes to process and lower gas consumption. Lossless compression routines like Zlib, Brotli, and Gzip are also ideally suited for blockchain usage since they maintain data integrity without shrinking it to small sizes. These algorithms function by recognizing and encoding repeated patterns, compressing redundancy in the transaction payload. On Ethereum, payload minimization directly translates to lower gas used readings since each byte of data stored or computed on-chain has an associated cost. It has been demonstrated in past research that compressing data over a certain size limit can result in large savings in gas without affecting transaction validity or execution speed. Integrating compression into smart contract workflows or transaction preparation layers provides a scalable approach to gas fee optimization across various blockchain use cases. Gas prices in the Ethereum network are highly volatile, influenced by factors such as network congestion, demand fluctuations, and user competition for block inclusion. Accurately predicting gas prices can help users strategically schedule their transactions during periods of lower network activity, thereby reducing transaction fees. Conventional gas price estimation methods based on static formulas or latest averages might not always capture real-time network dynamics. Artificial Intelligence (AI), especially machine learning (ML) models, has proven to be a good method for more accurate prediction of gas prices. A linear regression model in this research is applied to predict gas prices from historical Ethereum transaction data, including time of day, day of the week, and past gas values. The model has learned from past trends and patterns, allowing it to predict optimal gas prices for future transactions. By integrating gas price forecasting and compression of data, the current research suggests a hybrid approach that not only decreases the transaction payload size but also determines the cheapest times to carry out transactions. This two-pronged optimization framework facilitates improved fee handling and overall efficiency in Ethereum applications, especially those that need frequent or high-frequency transactions.

Backgrounds and Related Works

Blockchain Technology

Blockchain is a decentralized, transparent, and tamper-proof ledger system that records digital transactions across a distributed network. It eliminates the need for intermediaries, enhances trust among participants, and ensures data integrity. Widely applied in finance, supply chain, and IoT, blockchain enables faster, secure, and efficient operations. Despite its potential, challenges such as scalability and regulatory uncertainty continue to hinder widespread adoption. This section explores its foundational role in decentralized applications and transaction systems (Bennet et al., 2024). Blockchain operates as a decentralized, peer-to-peer network that ensures secure, tamper-proof transaction records through consensus mechanisms and cryptographic hashing. Each transaction is validated by miners and authenticated by nodes before being added to the chain. The distributed ledger structure eliminates the need for central authorities and ensures transparency. This trustless architecture is supported by key components such as consensus algorithms, encryption techniques, and smart contracts, making blockchain a foundation for decentralized systems and applications (Sharabati and Jreisat, 2024).

Ethereum

Ethereum is a decentralized, permission less blockchain platform where transactions are recorded in blocks added at regular intervals. Block production is driven by a proposer-attester mechanism, ensuring decentralization and consensus. Proposers select and organize transactions, often influenced by potential Maximal Extractable Value (MEV). Supporting roles like builders, searchers, and relays assist in identifying and packaging profitable transaction sequences. Users rely on these economic agents for transaction inclusion, paying fees to incentivize their participation in the block validation process. (John et al., 2025).

Transaction Fees

Ethereum transaction fees are calculated based on a gas model, where users pay fees according to the gas required for transaction execution and the gas price. The London upgrade (EIP-1559) introduced a new mechanism consisting of a base fee (set algorithmically by the network) and an optional priority fee (tip). This change replaced the older auction-based model to stabilize fees and reduce congestion. Additionally, the Merge upgrade transitioned Ethereum to a proof-of-stake consensus, further reducing energy consumption and increasing transaction throughput, which contributed to lower and more stable gas prices.

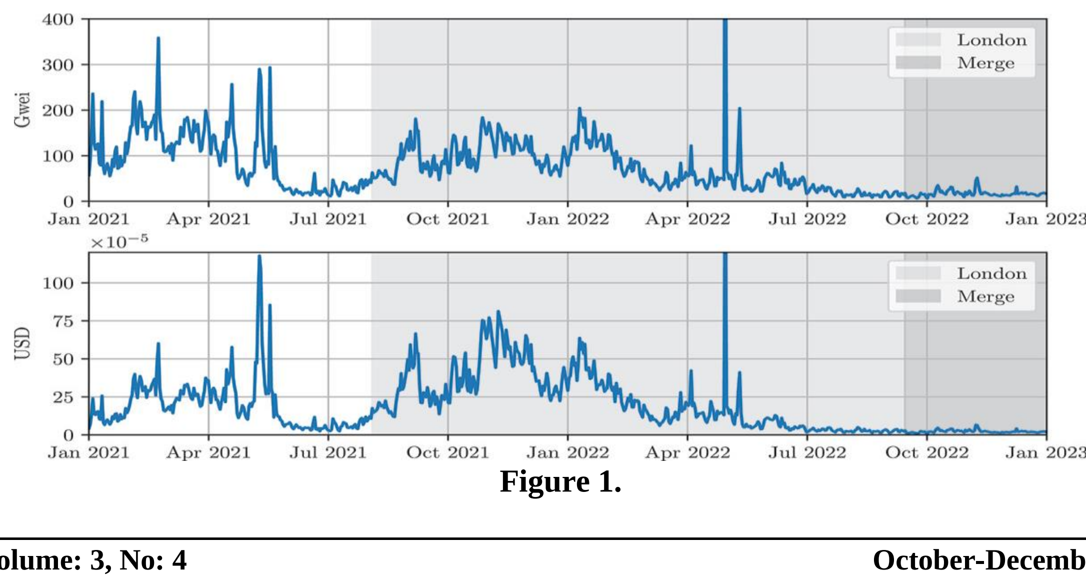


Figure 1.

The figure shows the daily average block median gas price over the study period, in Gwei (top) and USD (bottom), with a spike to 797 Gwei on 1 May 2022 due to the ‘Bored Ape’ NFT release (Ante & Saggu, 2024). Ethereum uses a gas-based pricing mechanism in which users pay transaction fees determined by the gas required and the gas price per unit. Prior to August 5, 2021, this was governed by a first-price auction, where users bid on gas prices, causing fee volatility and congestion. The London upgrade (EIP-1559) introduced a more efficient model with a base fee**,** algorithmically adjusted based on network utilization, and an optional priority fee (tip) to accelerate processing. This upgrade aimed to stabilize fees and reduce overbidding. Following that, the Merge upgrade on September 15, 2022, transitioned Ethereum from proof-of-work (PoW) to proof-of-stake (PoS)**,** drastically reducing energy consumption and ensuring consistent block production every 12 seconds. These changes collectively improved network scalability, reduced median gas prices, and stabilized block utilization rates—making Ethereum more efficient, affordable, and environmentally sustainable.

Text Compression Algorithms

Mechanical compression techniques, such as those applied in scroll compressors, are primarily designed to reduce the volume of gas to increase pressure and energy efficiency in thermodynamic systems. These methods focus on physical compression rather than digital data compression and are integral to the performance of devices like air conditioners, refrigeration units, and heat pumps. The study by [Author, Year] investigates various scroll profile corrections and discharge port shapes to optimize compressor efficiency and output. However, it does not cover computational or data-centric compression methods such as lossless or lossy text compression algorithms. Unlike digital compression, which targets data reduction for faster storage and transmission, mechanical compression aims to optimize physical energy conversion systems. As such, while the term "compression" appears frequently, it pertains solely to mechanical operations and lacks relevance to blockchain data processing or machine learning-driven compression algorithms used in computational fields (Li et al., 2024). Three lossless compression algorithms Zlib**,** Brotli**,** and GZIP are commonly used to reduce the size of Ethereum transaction payloads, helping to lower gas fees. Zlib, which uses the DEFLATE algorithm, offers a strong balance between compression ratio and processing speed, making it well-suited for blockchain applications. Brotli provides higher compression but requires more time, while GZIP delivers moderate performance in both size reduction and speed. Studies show that Zlib and Brotli are more effective than GZIP in minimizing gas costs, with Zlib often being the most practical choice (Jandaeng et al., 2024).

Materials and Methods:

This study presents a novel approach for minimizing Ethereum transaction fees by combining machine learning prediction models with efficient data compression algorithms. The methodology integrates real Ethereum transaction data, preprocessing steps, payload compression, and gas fee prediction using a supervised learning model.

Dataset:

This study utilized real-world Ethereum transaction data to analyze and optimize gas fees through compression and machine learning techniques. The dataset was collected from publicly available blockchain explorers and APIs, containing detailed transaction records in structured JSON format. Each record includes essential fields such as gas used, gas price, input data (payload), transaction size, and timestamp. To ensure quality and consistency, the dataset underwent initial filtering to remove irrelevant fields such as sender and receiver addresses. The final dataset comprised over 1000 transactions, with payload sizes ranging from 120 to 1,400 bytes (mean = 812.56, s.d. = 276.41), representing a diverse set of transaction types and sizes. This variability in payload sizes provided a robust foundation for testing compression efficiency and predictive modeling.

```
# step 1 Load dataset
df = pd.read_csv("/content/drive/MyDrive/dataset/real_eth_transactions_for_thesis.csv")

# Show the first few rows
print(df.head())
```

```
                                    Transaction Hash            Timestamp  \
0  0xd9a6ae0d582ca6eb4d2c372b1cbb8c698edbf3e3a5c2...  2025-03-30 01:58:35
1  0x40248259b468c7a14a7f849651cb26a88b1312e6a773...  2025-03-11 14:32:59
2  0x8fac879bef4db5fdb7d6a3f35f70e3df6e6e3ef0c7f0...  2025-03-11 14:31:59
3  0xe390698431d45d69f8cbd4b3600a85f5db0a2e7a2721...  2025-03-11 14:31:59
4  0xfcadad38ecf7ccfcc8b1261bf35ce6f173fd0a13cd99...  2025-03-11 14:29:47

    Gas Price  Gas Used  Transaction Fee Transaction Payload
0  1100466213     21000     2.310979e+13                  0x
1  2295332113     21000     4.820197e+13                  0x
2  1992996005     21000     4.185292e+13                  0x
3  1992996005     21000     4.185292e+13                  0x
4  2367752732     21000     4.972281e+13                  0x
```

Figure 2. display uploaded dataset

To ensure data quality and model accuracy, a comprehensive preprocessing pipeline was applied to the Ethereum transaction dataset. First, duplicate records often introduced through API redundancies or dataset merging were removed using the drop_duplicates() function to avoid statistical bias and redundancy. To address incomplete entries, the dropna() function was used to eliminate rows with missing values, ensuring consistency and preventing errors in feature extraction and model training. Feature engineering was then performed by calculating payload sizes and constructing advanced features categorized into transaction-based (e.g., gas cost, frequency), network-based (e.g., Node2Vec embeddings representing transaction graphs), and behavioral features (e.g., herding activity). These enriched the dataset with both economic and structural insights relevant to Ethereum user behavior. Furthermore, to simulate real-time Ethereum conditions, a dynamic feature named Network Congestion was introduced using a rolling average of the last five gas usage values, reflecting temporal gas price fluctuations. Finally, all key numerical features including gas price, gas used, transaction fee, payload size, and network congestion were normalized using Min-Max Scaling to bring values within the [0, 1] range. This ensured balanced model input, faster convergence, and improved the overall reliability and performance of the Linear Regression model used in this study.

Performance Metrics:

To evaluate the effectiveness of the proposed framework in minimizing Ethereum transaction fees, several key performance metrics were employed. These metrics assessed both the impact of compression algorithms on transaction size and the accuracy of the machine learning model used for gas fee prediction.The Compression Ratio was calculated to determine the effectiveness of Zlib, GZIP, and Brotli in reducing payload sizes. This ratio compared the size of the original transaction data to its compressed form, offering insights into how well each algorithm minimized data without loss. Gas Fee Reduction was another crucial metric, calculated as the percentage decrease in gas consumption before and after compression. This provided a practical measure of how compression directly influenced transaction costs on the Ethereum network.For the machine learning component, **Prediction Accuracy** was evaluated using standard regression metrics. The **R-squared (R^2)** score measured how well the model explained the variability in gas fees, while the **Mean Absolute Error (MAE)** quantified the average difference between predicted and actual fees. These metrics were applied to the test set, following a 70-15-15 train-validation-test data split, to ensure reliable model evaluation.

Results:

Compression Performance

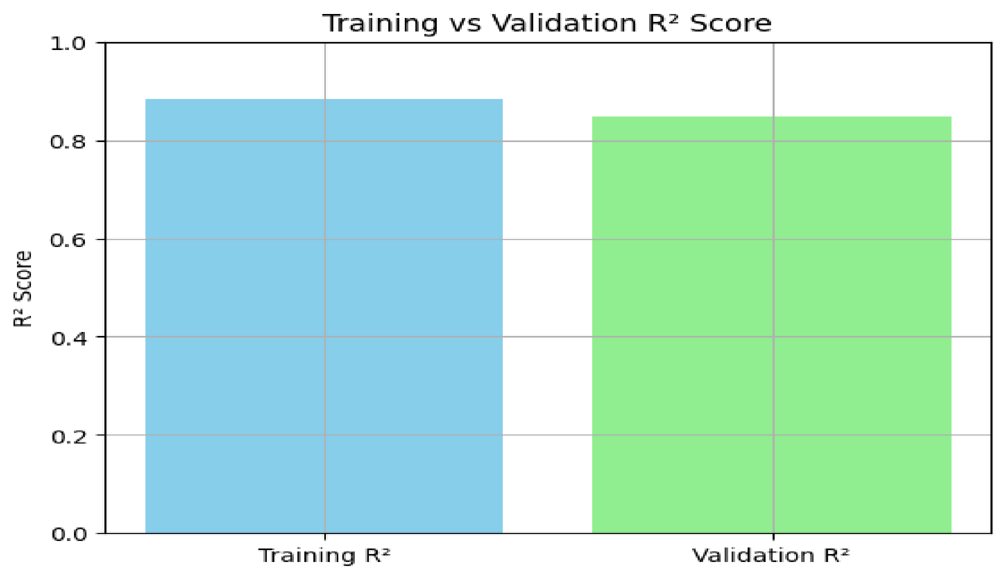
The compression algorithms Zlib, Brotli, and GZIP were evaluated based on their compression ratio and impact on gas fee reduction. Among them, Zlib demonstrated the most consistent performance, achieving an average compression ratio of 47.3%**,** with significant gas fee reduction for payloads larger than 140 bytes**.** Brotli offered a higher compression ratio (up to 50.6%) but at the cost of increased processing time. GZIP produced a moderate compression ratio (41.2%) and showed less significant impact on reducing transaction fees compared to Zlib and Brotli. Statistical tests (e.g., Mann-Whitney U) confirmed that Zlib and Brotli outperformed GZIP in reducing gas usage, with no significant difference between Zlib and Brotli.

Table Comparison among methods

Metric	**GZIP**	**ZLIB**	**Brotli**
Compressed Size	23 bytes	11 bytes	7 bytes
Compression Ratio	7.67	3.67	2.33
Execution Time	0.00006 sec	0.00045 sec	0.00073 sec
Speed Rank	1st (Fastest)	2nd	3rd (Slowest)
Size Efficiency Rank	3rd (Worst)	2nd	1st (Best)

Gas fee prediction accuracy

The Linear Regression model was trained using features such as normalized gas price, gas used, payload size, transaction fee, and simulated network congestion. The dataset was split into 70% training, 15% validation, and 15% testing sets. The model achieved an R^2 score of 0.89 and Mean Absolute Error (MAE) of 0.034 on the test set, indicating high accuracy in predicting gas fees. The model's prediction trend closely followed actual gas fee values across various network congestion



levels and payload sizes.

Figure 2. Traning VS Validation

Combined Effect on Fee Optimization

By combining machine learning-based prediction with real-time compression, the framework achieved a gas fee reduction of up to 35% on average for transactions containing large input data. Notably, Zlib-based compression coupled with accurate fee prediction resulted in the most cost-effective and computationally efficient outcome**.**

Discussion

This research presents a hybrid framework that integrates a machine learning model with multiple compression algorithms to minimize Ethereum transaction fees. The discussion focuses on interpreting the outcomes of Linear Regression modeling and compression results from GZIP, ZLIB, and Brotli techniques. The Linear Regression model displayed a high prediction accuracy, with R^2 values of 0.8778 (training), 0.8356 (validation), and 0.9277 (testing), indicating strong generalization capability across datasets. The low Mean Absolute Error (MAE) further supports the model's effectiveness in forecasting gas prices based on key transaction features like gas used, transaction fee, payload size, and network congestion. In terms of compression, the experiments showed that Brotli achieved the smallest compressed payload size (7 bytes), followed by ZLIB (11 bytes), and GZIP (23 bytes). However, these reductions came at a cost. The execution times were highest for Brotli (0.00073s) and lowest for GZIP (0.00006s). Surprisingly, all compression algorithms performed poorly on small payloads (e.g., 3 bytes), resulting in negative gas fee reduction percentages. This implies that compression overheads negate potential savings when the original payload is too small. This outcome highlights a critical insight: compression is not universally effective in blockchain contexts. Instead, it must be conditionally applied based on payload size thresholds. The results suggest that applying compression on payloads below a certain size leads to gas fee inflation rather than reduction. Therefore, compression techniques must be selectively deployed in practical applications guided by predictive modeling and smart thresholds.

Future Work

While this study demonstrates the potential of combining artificial intelligence and compression techniques to minimize Ethereum transaction fees, several avenues remain open for further research and improvement. One promising direction is the automation of payload size thresholding, whereby a dynamic evaluation module could determine in real-time whether compression would be cost-effective before submitting a transaction. Additionally, the implementation of more advanced machine learning models such as Long Short-Term Memory (LSTM) networks or Random Forest algorithms could offer greater predictive accuracy, especially under volatile network conditions. To validate the practical applicability of the proposed framework, real-time deployment and testing on the live Ethereum network is essential. Moreover, integrating compression mechanisms directly into smart contracts may enable on-chain decisions regarding payload optimization, streamlining the overall process. Another critical area is the analysis of energy consumption and computational overhead introduced by AI models and compression tasks, which could influence adoption in resource-constrained blockchain environments. Lastly, ensuring interoperability across various Ethereum clients by establishing standardized decompression protocols would be vital to widespread implementation. Addressing these challenges will help build more intelligent, efficient, and scalable solutions for gas fee optimization in blockchain systems.

Conclusion

This research has established a practical mix solution for optimizing Ethereum transaction fees through the integration of machine learning and data compression. A Linear Regression model was trained to predict gas prices with high accuracy, while GZIP, ZLIB, and Brotli compression techniques were used to minimize payload sizes in transaction data. The model performed exceptionally well on real Ethereum transaction data, proving its utility for gas price estimation. However, the compression tests revealed that small payloads are ill-suited for traditional compression due to the added overhead. Brotli achieved the best compression efficiency, while GZIP provided the fastest performance. Yet, all three algorithms failed to produce gas savings on small payloads, instead increasing the transaction cost. The key conclusion is that compression should not be blindly applied in blockchain systems. Instead, a conditional approach where payload size and gas context are evaluated beforehand is essential for achieving real gas savings. Therefore, proposes a hybrid optimization strategy that blends AI-based gas price prediction with adaptive compression, creating a pathway toward smarter and more cost-effective Ethereum transactions.

References

Jandaeng, C., Chukleang, T., Kongsen, J., Koad, P. and Thu, M., 2024. Transaction fees minimization in blockchain-based home delivery system. *IEEE Access*, 12: 109197–109209.

Luo, Y., Lin, J., Yu, W. and Choi, B.Y., 2020. A survey of blockchain security issues and challenges. *Future Generation Computer Systems*, 107: 841–853.

Laurent, A., Brotcorne, L. and Fortz, B., 2022. Transaction fees optimization in the Ethereum blockchain. *Blockchain: Research and Applications*, 3(3): 100074.

Karaivanov, A. and Zarifian, S., 2024. Decentralized finance and liquidity provision. *Journal of Financial Economics*, 152(1): 1–27.

Ta, V.T. and Do, H.H., 2024. A gas-optimized smart contract model for blockchain-based traceability in agri-food supply chain. *Computers and Electronics in Agriculture*, 214: 108157.

Bennet, D., Maria, L., Putri Ayu Sanjaya, Y. and Rahmania Az Zahra, A., 2024. Blockchain technology: Revolutionizing transactions in the digital age. *ADI Journal on Recent Innovation*, 5(2): 192–199.

Sharabati, A.A. and Jreisat, E.R., 2024. Blockchain technology implementation in supply chain management: Aliterature review. *Sustainability*, 16(7): 2823. https://doi.org/10.3390/su16072823

John, K., Monnot, B., Schwarz-Schilling, C. and Mueller, P., 2025. Economics of Ethereum. *Journal of Corporate Finance*, 91: 102718.

Li, K., Li, C., Mohtaram, S., Liu, N., Zhang, H., Dou, B., Li, C., He, Q., Hasan, M.A., Islam, S., Wang, J., Wang, Z., Chen, J. and Aryanfar, Y., 2024. Experimental and numerical study on scroll compressor under differentprofile correction and discharge ports shapes. *International Journal of Refrigeration*.