IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 13 July 2024, accepted 4 August 2024, date of publication 7 August 2024, date of current version 16 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3439683

RESEARCH ARTICLE

Transaction Fees Minimization in Blockchain-Based Home Delivery System

CHANANKORN JANDAENG[1,2], **TIRAJET CHUKLEANG**[3], **JONKSUK KONGSEN**[2], **PEERAVIT KOAD**[1,2], **AND MAY THU**[4]

[1]Informatics Innovative Center of Excellence (IICE), Walailak University, Nakhon Si Thammarat 80160, Thailand
[2]School of Informatics, Walailak University, Nakhon Si Thammarat 80160, Thailand
[3]Management of Information Technology, School of Informatics, Walailak University, Nakhon Si Thammarat 80160, Thailand
[4]Faculty of Engineering, Cambodia University of Technology and Science, Phnom Penh 121003, Cambodia

Corresponding author: Chanankorn Jandaeng (cjundang@gmail.com)

ABSTRACT This study investigates the impact of Zlib compression on gas consumption within blockchain systems, focusing particularly on Ethereum transactions. By employing the Ethereum simulator Ganache, we simulate 100 realistic home delivery system datasets to evaluate the performance of compressed versus uncompressed data. The methodology encompasses rigorous statistical analysis to ensure robust results. Our findings reveal that using the Zlib algorithm to compress textual data exceeding 141 bytes before submitting transactions on the Ethereum network reduces the gasUsed while maintaining the system time unchanged. This demonstrates the effectiveness of data compression in optimizing transaction costs without affecting operational efficiency. Additionally, our research extends to analyzing real gasPrice trends on the Ethereum network. We propose a non-linear regression model that accurately predicts hourly gasPrice variations based on the day of the week and the specific time. This provides a valuable tool for users to plan transactions strategically. These insights enhance the understanding of blockchain dynamics and offer practical solutions for improving economic and system efficiency in blockchain operations.

INDEX TERMS Text compression algorithm, transaction fees, blockchain, logistics, home delivery.

I. INTRODUCTION

As a subset of logistics, home delivery services streamline the process from order placement to recipient pickup [1]. Smart logistics leverages technology for optimized delivery, enhancing resource utilization, cost-effectiveness, and security [2], [3]. Blockchain adoption addresses security and privacy concerns, ensuring enhanced security, transparency, and sustainability in supply chain logistics [4].

Blockchain technology, introduced through Bitcoin [5] and later enhanced by Ethereum [6], is a decentralized and tamper-resistant framework crucial for ensuring integrity and security in Industry 4.0 logistics. Its adoption mitigates risks, counters fraud, and introduces transparency and trust, essential in fostering sustainable and resilient supply chain practices [7]. Blockchain technology enhances logistics by providing traceability [8], transparency [9], reliability [10], and end-to-end oversight. Tailored systems and intelligent contracts amplify efficiency, and decentralized features bolster security [11]. Despite its potential to streamline operations and fortify security, implementing blockchain in logistics faces challenges such as a lack of shared trust among stakeholders, complexity in integration, and barriers in reverse logistics [12] including high costs and stakeholder resistance [13]. Addressing confidentiality, privacy [14], data integrity, and scalability is crucial to navigating the complexities of blockchain implementation and carefully evaluating its benefits and costs in the logistics industry [15].

Various strategies have been proposed to mitigate transaction fees within the Ethereum network, including optimizing fees by establishing the minimum price users should pay for timely transaction processing [16]. Another strategy involves shifting a substantial portion of contract execution off-chain, significantly decreasing gas usage [17]. Adopting

The associate editor coordinating the review of this manuscript and approving it for publication was Vlad Diaconita.

an optimistic rollup (ORU) technique facilitates computation delegation from the main Ethereum blockchain to an untrusted remote system, resulting in a 20 times reduction in transaction fees [18]. Additionally, a max-min fairness-based algorithm has been devised to equitably distribute Ether, further reducing transaction costs and averting the exhaustion of the block gas limit.

Home delivery service customers face variable fees influenced by the product's value and blockchain technology. In addition to the delivery fees, users bear Transaction fees, increasing the overall service cost. Sustainable adoption strategies target mitigating blockchain-related service fees or reducing transaction costs [19].

The Ethereum Improvement Proposals 1559 (EIP-1559) upgrade uses a constant learning rate algorithm to calculate the base fees, addressing congestion issues. Transaction fees in Ethereum and similar blockchain systems are user-determined and comprise base fees and an optional tip [20]. Blockchain systems aim to optimize fees for timely transactions while minimizing user costs [16].

Notably, compression techniques play a crucial role in mitigating transaction fees in blockchain systems. SymmeProof integrates vector compression and inner-product range proof methods, resulting in efficient, non-interactive zero-knowledge range proofs and reduced communication costs [21]. Another method focuses on removing transaction data with a high ratio of spent transaction outputs and compressing fixed-length fields, achieving a notable compression ratio of 96.90% and substantial storage space savings for Bitcoin full nodes [22]. These methods underscore the use of compression techniques to effectively minimize transaction fees in blockchain systems.

In contrast, blockchain in-home delivery is discouraged due to high transaction fees, extended response times from processing large data sizes [19], and the limitations of decentralized approaches, which often lack trust and may necessitate reliance on third parties [23]. Additionally, deploying blockchain in-home delivery may require private chains, limiting global availability and scalability [24]. Although home security improvements are uplifting, it is essential to recognize that blockchain technology is not invincible to security breaches. Despite their much-lauded security measures, smart home devices are susceptible to viruses and hackers [25].

This research proposes a method to address challenges in employing blockchain for home delivery, focusing on minimizing gas costs using data length reduction. Specifically, the suggested approach involves compressing data before entering the blockchain network. However, this approach introduces an overhead in execution time. Therefore, we must demonstrate that our approach is relatively low on execution time. Moreover, the research recommends optimizing costs further by identifying optimal transaction fees and refining transaction responsiveness. Ultimately, these strategies aim to enhance blockchain technology's efficiency and cost-effectiveness in home delivery.

II. BACKGROUNDS AND RELATED WORKS

A. BLOCKCHAIN TECHNOLOGY

Various information methodologies have emerged in response to prevalent challenges, with blockchain as a prominent solution. Nakamoto introduced blockchain technology to establish an electronic monetary system devoid of intermediaries [5]. This approach entails encrypting transactions within blocks, interconnecting them, and systematically appending new blocks. Employing a distributed consensus algorithm to encrypt block data ensures data security.

Blockchain operates as a peer-to-peer network utilizing the TCP protocol and featuring a randomized topology, where each node autonomously peers with others [26]. Decentralized transactions are facilitated through blockchain technology, which boasts a sophisticated technological infrastructure. Upon a client or application publishing a transaction to the blockchain network, miners validate it while nodes authenticate transactions. Each miner node scrutinizes the transaction's legitimacy, ensuring the sender possesses adequate funds to cover processing fees or, in the case of cash transfers, sufficient funds. Additionally, it verifies recipient details. Subsequently, the miner node constructs a block comprising validated transactions. However, the blockchain network selects only one miner's block through a consensus mechanism. The block is disseminated to all nodes upon election, rewarding the blockchain network and transaction verification fees. The block is anchored to the network via a link or chain maintained by the block's hash function. Any alterations to the block result in hash value modification, revealing transaction inconsistencies.

B. ETHEREUM

Ethereum represents a distributed computing platform comprising a decentralized, automated, and democratic network of computers, gaining widespread acceptance across various domains [6]. At the core of Ethereum lies the smart contract, a groundbreaking concept in blockchain technology. Leveraging Ethereum's self-executing and event-triggered features, smart contracts enable online actions to transpire sans reliance on trusted intermediaries. With a robust community and information exchange platform, Ethereum thrives as an innovation hub. Smart contracts encapsulate rules and obligations, comprising addresses, states, and functions. These contracts are identified by unique addresses within the Ethereum Blockchain Network. Ethereum facilitates the execution of smart contracts and the deployment of distributed applications, whereas Solidity smart contract acts as the backend, complemented by a web application frontend.

C. TRANSACTION FEES

The transaction fee is the cost of executing a transaction on the Ethereum blockchain platform, calculated based on (1).

$$TransactionFree = gasUsed \times gasPrice, \tag{1}$$

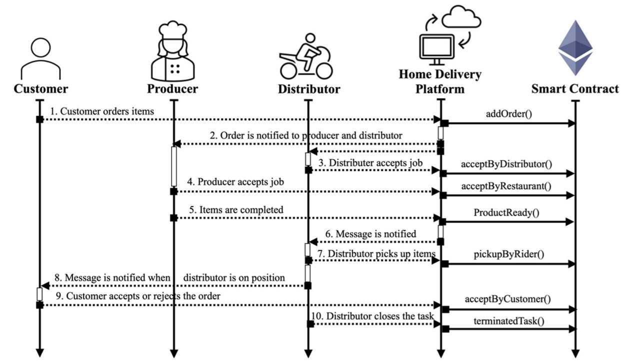


FIGURE 1. The sequence diagram illustrates the process of home delivery using blockchain technology. [19].

when gasUsed refers to the number of gas units required to execute the transaction, while gasPrice refers to the amount the transaction issuer is willing to pay for each gas unit.

In Ethereum, miners process transactions and receive mining rewards directly from these transaction fees. It is important to note that a higher gas price does not necessarily imply a higher transaction priority. Blockchain-powered application developers face challenges in setting optimal gasPrice due to the cyclic dependency between gasUsed and gasPrice. The actual gas usage of a transaction can only be known after it is processed.

The gasUsed in the Ethereum network has multifaceted impacts. Miners prioritize transactions based on gasPrice rather than transaction fees, leading to no guarantee of higher priority with a higher fee. This challenges developers as the exact gas usage of contract transactions is known only after processing. Gas optimization techniques offer potential solutions, like deploying fee-less smart contracts using Solidity. The gasUsed of Ethereum is affected by gas costs for Ethereum virtual machine (EVM) operations, gas-expensive smart contract patterns, and alignment with computational costs.

D. BLOCKCHAIN-BASED HOME DELIVERY SYSTEM

The home delivery system incorporates a carefully designed data structure and smart contract for data integrity and security assurance. We previously proposed blockchain technology's application in enhancing security and transparency for home delivery logistics [19]. It aims to address non-repudiation issues and improve overall security. A robust security policy framework is introduced to govern blockchain usage, fostering trust and accountability. Through blockchain implementation, we proposed specific mechanisms to prevent nonrepudiation issues for home delivery systems and demonstrated this system in the Ethereum network. We comprehensively analyze transaction costs and response times to evaluate the system's performance. However, the transaction fees and the response time are too high, more than acceptable for the business to invest.

In Figure 1, the interactions between the home delivery platform and the smart contract encompass functions with varying transaction fee characteristics. Expressly, the addOrder() function incurs a higher transaction fee contingent upon data length, while other functions primarily involve updating the order's status and incur lower fees. Consequently, this research analyzes transaction fees and response times about the addOrder() function within this context.

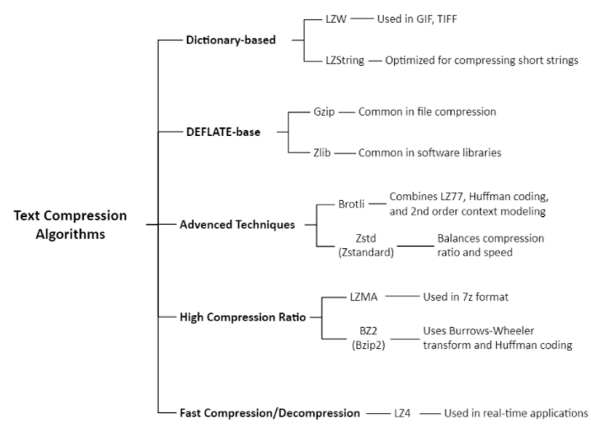


FIGURE 2. Taxonomy of text compression algorithms.

E. TEXT COMPRESSION ALGORITHMS

Text compression algorithms are techniques employed to reduce text data size while maintaining original content. These algorithms fall into two categories: lossless, which aims to minimize data size without losing information, and lossy, which sacrifices information for higher compression ratios [27]. However, this paper solely focuses on lossless text compression techniques. The significance of data compression and the need for effective data management in the contemporary data-intensive era is of utmost importance, as compression enhances storage and transmission capabilities, resulting in faster data transfer and reduced storage space requirements while also enabling efficient search within compressed files, potentially improving search speed compared to uncompressed text [28].

Figure 2 shows the taxonomy of the text compression algorithm. A text compression algorithm that integrates dictionary-based approaches, DEFLATE-based methods, advanced techniques, high compression ratios, and fast compression/decompression capabilities aims to efficiently reduce the size of textual data while maintaining a balance between compression effectiveness and processing speed. Dictionary-based algorithms like LZ77 and LZ78 replace repetitive patterns concerning dictionary entries, while DEFLATE combines LZ77 with Huffman coding for efficient compression.

Advanced techniques such as the Burrows-Wheeler Transform further enhance compression efficiency. Achieving a high compression ratio minimizes redundancy, while fast compression/decompression algorithms prioritize speed without compromising compression effectiveness. Together, these components enable the algorithm to significantly reduce the size of compressed text while ensuring rapid processing, which is crucial for various applications, from data storage to real-time data processing.

Optimizing gas usage in Ethereum involves various techniques and tools to reduce the cost of smart contract development and transactions. GaSaver automates the detection of gas-expensive patterns in Solidity code [29]. Super-optimization explores all possible instruction sequences to find the most efficient translation [30]. Static profiling identifies gas-expensive fragments and optimizes gas consumption [31]. MadMax utilizes static program analysis to detect gas-focused vulnerabilities in smart contracts [32]. Lastly, the ATOM smart contract architecture supports fast contract updates and efficient execution to minimize gas usage, update latency, and ledger size [33]. Zamani et al. proposes an algorithm that reduces gas consumption in blockchain networks by compressing data through DNA-to-ASCII conversion, delta computation, and LZW compression [34]. This approach lowers gas usage, reduces fees, improves scalability, and enhances data security through blockchain-based immutability.

III. MATERIALS AND METHODS

A. RESEARCH FRAMEWORKS

In this study, we have developed and outlined a methodology to minimize transaction fees within a blockchain-based home delivery system by employing data compression and gas price reduction techniques. Implementing these techniques may improve the overall service's response time.

The proposed system, as depicted in Figure 3, consists of several distinct components, including data generation, a text compression algorithm, gas price balancing, and transaction fee minimization. We generated a dataset comprising 10,000 orders in JSON format. The dataset contained customer information with varying text lengths in fields such as name, address, telephone number, and item quantity. Firstly, we identified an optimized text compression algorithm tailored to our objectives, which included reducing gas usage and system times during compression and decompression processes. Secondly, we examined the influence of fluctuating gas prices on transaction fees and their consequent impact on the system time of the Ethereum network to enhance transaction cost efficiency. Finally, we proposed an algorithm integrating the selected compression method with strategic gas pricing strategies. This approach aims to minimize transaction fees while ensuring acceptable response times on the Ethereum network. Our research contributes significantly to blockchain technology and decentralized applications by addressing the efficient management of transaction costs and response times.

B. SIMULATING TEXT COMPRESSION ALGORITHM

This section aims to identify an optimal data compression algorithm by evaluating various factors, including data reduction efficiency, compression and decompression times, and blockchain resource utilization. Key considerations include balancing processing time and compression ratio. Additionally, it is important to assess the impact of gasUsed on blockchain transaction efficiency and resource costs. These factors are guiding principles for selecting the most suitable algorithm, ensuring optimal performance and resource utilization within the blockchain ecosystem.

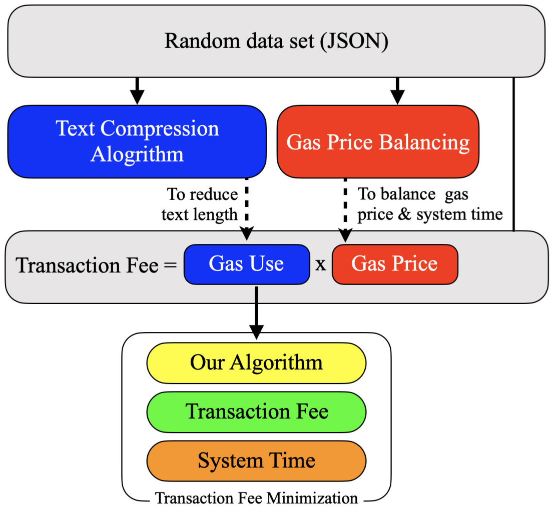


FIGURE 3. Architecture of our proposed method.

1) DATASET

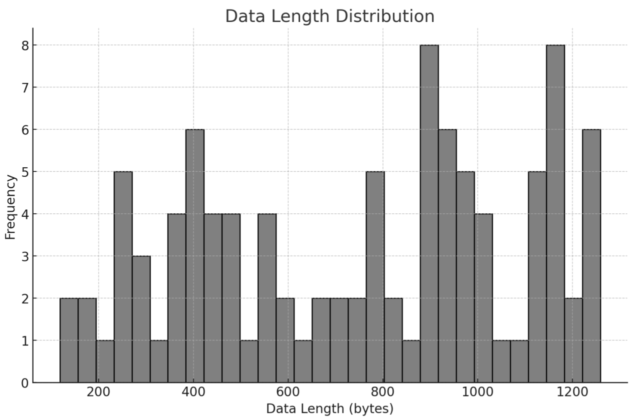
This experiment simulated the ordering process of items from a restaurant by generating data representing the restaurant, customer, and items ordered. Subsequently, the data was randomly matched to create a dataset comprising 100 transactions in JSON format. A transaction includes information such as the ordered items (name, quantity, price), details about the restaurant (name, telephone number), and customer information (name, location, telephone number). The transaction's data length in the data set varies between 117 and 1,254 bytes (mean = 746.42, s.d. = 334.69), as illustrated in Figure 4.

Figure 4 illustrates a representative transaction sample containing details of two distinct items, including the customer's precise address and telephone contact, as well as the appellation and contact number of the restaurant. Notably, the textual content of this example totals 333 bytes, a size typical of data parcels encountered in food delivery transactions.

2) PERFORMANCE METRICS

This experiment aimed to evaluate the gasUsed, system time (ST), and compression ratio (CR) for a transaction. The performance metrics were as follows:

- *gasUsed* is a fundamental Ethereum metric that measures the computational costs incurred during smart contract execution. Users pay for gas, which serves as a reward for miners. Gas costs can vary depending on



(a) The histogram of data length (N=100, average=746.42, s.d. = 334.69)


```
{
  "Menu": [
    {
      "menuName": "Special fish cake",
      "menuPrice": "70",
      "Qty": 3
    },
    {
      "menuName": "Chicken curry",
      "menuPrice": "40",
      "Qty": 2
    }
  ],
  "Customer": {
    "Name": "Chanankorn Jandaeng",
    "Location": "Sukhumwit [redacted] 10260",
    "Phonenumber": "0849999999"
  },
  "Restaurant": {
    "Rest_name": "ABC Restaurant",
    "Rest_phone": "0876322222"
  }
}
```


(b) The example of dataset represented in the JSON format (333 bytes).

FIGURE 4. The 100 orders (transactions) of data set.

coding style and data length, and accurate metrics are obtained from the Web3 Library for precise assessment.

- *Compression Ratio (CR)* is a measure of the effectiveness of the compression process. It is calculated by comparing the original data's size to the compressed data's size. The compression ratio can be expressed using the following formula:

$$CR = \frac{c_{post}}{c_{pre}}, \tag{2}$$

where c_{post} represent the volume after compression and c_{pre} represent the volume before compression.

- *System Time (ST)* refers to the duration starting from the initiation of a transaction. This begins with the commitment to the API that invokes the smart contract. It ends when the caller receives a notification indicating the successful completion of the process. The system time consists of three components: compression time (t_c), waiting time (t_w), block time (t_b) and execution time (t_e) in the blockchain, decompression time (t_d), and ϵ denotes other latency time in the transaction. The empirical evaluation is conducted within an isolated system, excluding the influence of network latency:

$$ST = t_c + t_w + t_b + t_e + t_d + \epsilon \tag{3}$$

3) TESTBED AND SCENARIOS

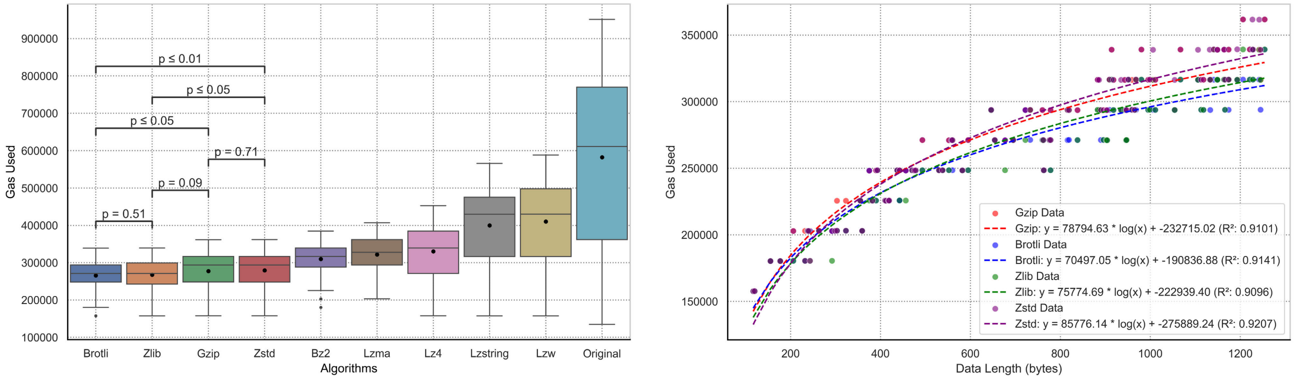
To conduct a comprehensive comparison involving the metrics of gasUsed, system time (ST), and compression ratio (CR) across nine distinct text compression algorithms (LZW, LZString, GZip, Brotli, BZ2, LZ4, LZMA, Zlib, and Zstd), a dataset comprising 100 transactions are subjected to compression with each algorithm before submission onto the Ethereum network. Subsequently, the compressed transactions are transmitted onto the Ethereum network for execution, following which the resulting transactions are retrieved and decompressed. All performance metrics are evaluated and analyzed using statistical methods. These methods compare the mean values and model regression functions based on hypothesis tests.

Ganache Ethereum blockchain has been selected as the designated platform for executing transactions to mitigate potential network delays. The smart contract is developed using the Solidity programming language, ensuring seamless integration with the Ethereum network via Web3.py. Furthermore, the installation and invocation of the web API are performed on the networked computer. The invocation procedures utilize the Python programming language within the Flask framework to ensure efficient interaction with the Ethereum network.

4) RESULTS: GAS USED

The study investigates different data compression algorithms by analyzing gasUsed metrics. Figure 5 compares the average gasUsed between text compression algorithms and the data variance. This arrangement facilitates the comparison of variability and central tendency across algorithms. The algorithm demonstrating the most efficient gasUsed performance, characterized by the lowest average gas usage, is positioned at the leftmost side of the graph.

This analysis is crucial in algorithm selection across various scenarios, considering gasUsed requirements and compression efficiency. Four algorithms exhibiting the lowest average gas usage were selected for normal distribution analysis. However, the utilization of Shapiro-Wilk statistical tests indicated that these four algorithms significantly deviate from a normal distribution ($p < 0.05$), thus requiring further examination. The gasUsed of Brotli, Zlib, GZip, and Zstd algorithms does not follow a normal distribution ($K = 10.0167, p = 0.0184$). Consequently, all pair comparisons were conducted using pairwise Mann-Whitney U tests. The statistical analysis uncovers notable performance discrepancies between Brotli, Gzip, and Zstd. At the same time, no substantial difference is observed between Brotli and Zlib before the p adjustment. Considering gas consumption,



(a) The box plot of algorithm and gasUse **(b) Regression analysis of Brotli, Zlib, Gzip, and Zstd algorithms**

FIGURE 5. The comparison of gasUsed between text compression algorithm and the original data.

Brotli and Zlib emerge as suitable candidates for the specific application under scrutiny.

We analyzed the trend lines of gasUsed across the four algorithms relative to data length to assess and validate the performance of both algorithms accurately. Figure 5 demonstrates that the gasUsed of all algorithms exhibits growth following logarithmic functions, with R^2 values exceeding 0.9000. Let $H_\beta, H_\chi, H_\gamma, H_\zeta$ represent the gasUsed of Brotli, Zlib, Gzip, and Zstd, respectively, as a function of data length x in bytes. The regression functions are expressed as follows:

$$\begin{aligned} H_\beta(x) &= 70,497.05log(x) - 190,836.88, R^2 = 0.9141, \\ H_\chi(x) &= 75,774.69log(x) - 222,939.40, R^2 = 0.9096, \\ H_\gamma(x) &= 78,794.05log(x) - 232,715.02, R^2 = 0.9101, \\ H_\zeta(x) &= 85,776.14log(x) - 275,889.24, R^2 = 0.9207 \end{aligned} \tag{4}$$

This study examines the gas usage efficiency of Brotli, Zlib, Gzip, and Zstd over the Ethereum network, utilizing statistical analysis to evaluate their performance. The Shapiro-Wilk test revealed that the gas usage data for distinct algorithms does not conform to a normal distribution, indicating unusual patterns. The Kruskal-Wallis H test identified significant differences in mean gas usage among various algorithms. Further pairwise comparisons via the Mann-Whitney U test showed significant differences between Brotli, Gzip, and Zstd. However, no significant difference was observed between Brotli and Zlib. An intersection analysis between Brotli and Zlib determined that both algorithms exhibit equivalent gas usage at a data length of approximately 439 bytes, quantified at approximately 237,980 gas.

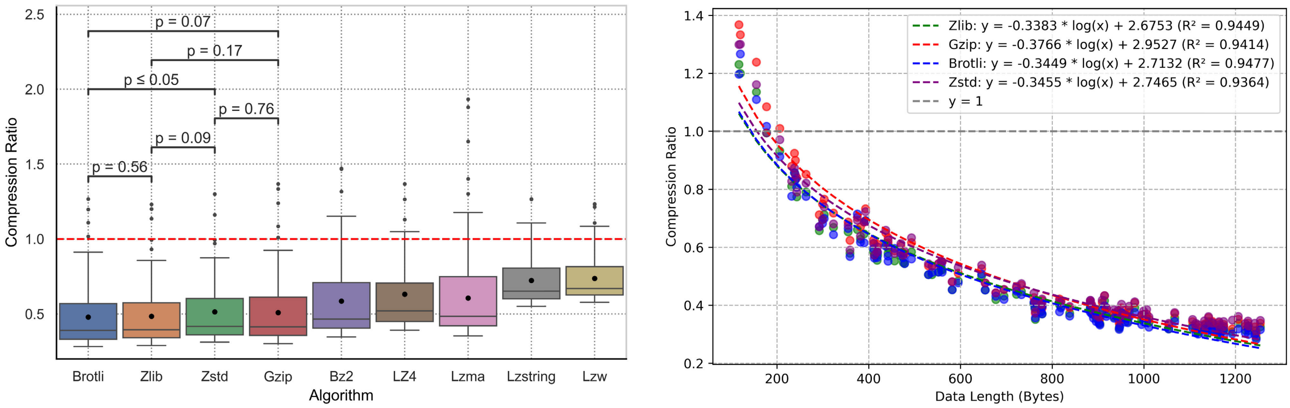
5) RESULTS: COMPRESSION RATIO

This study compared the efficiency of different data compression algorithms to determine the most effective one for this research. Figure 6 presents the results, showing the distribution of compression ratio (CR) values for each algorithm using a boxplot method and arranging the average CR in ascending order. The findings indicate that the Brotli and Zlib algorithms have similar and lower distribution spreads than others, with closely matched CR values.

Figure 6 utilizes boxplot graphs to illustrate the varying compression rates across multiple algorithms, emphasizing differences in efficiency. The algorithms are ranked based on average compression ratios, highlighting their ability to reduce data. The main observation is that the algorithm with the lowest average compression ratio performs most efficiently. The size of the boxplots and whisker lengths indicate the diverse data handling capabilities of the algorithms, with outliers suggesting potential significant deviations in compression performance under specific conditions. This analysis is crucial for selecting the most suitable algorithm for an application, considering the need for high compression performance and versatility in processing various data types. This comparison assists in choosing the most effective algorithm to meet specific compression requirements and data handling expectations.

This analysis investigates the relationship between data size and compression ratio within the framework of the Zlib algorithm, utilizing a mathematical model to elucidate this relationship as depicted in Figure 6b. The study indicates that the model achieved the highest R^2, indicating robust adaptability. Notably, the logarithmic model emerged as the best fit for the data, with an $R^2 = 0.9449$. This high coefficient of determination underscores the model's effectiveness in capturing significant variation in compression rate relative to data size.

Further insights were obtained through additional analysis, where a horizontal line was drawn at $y = 1$ on the graph to demarcate the point where the compression ratio equaled the original data size, indicating negligible compression. These findings carry significant implications for compression algorithms in long-length data management. They provide a nuanced understanding of algorithmic performance across different data sizes. Consequently, this elucidation enables informed algorithm selection tailored to the specific data



(a) The box plot of algorithm and compression ratio

(b) The non-linear regression model between data length and compression ratio

FIGURE 6. The comparison of compression ratio between text compression algorithm and the original data.

characteristics. The study compares Zlib and Brotli algorithms for text compression, finding Zlib comparable but faster. It highlights a threshold for effective compression, below which overhead increases. Emphasizing the importance of model selection aids in optimizing compression strategies. These insights enable informed decision-making and adopting efficient data management practices in technical contexts.

The study compares the Zlib and Brotli algorithms for text compression, noting that Zlib is comparable to Brotli but operates faster. It identifies a critical threshold for effective compression, below which overhead increases. Model selection is emphasized when optimizing compression strategies. These insights facilitate informed decision-making and the adoption of efficient data management practices in technical contexts.

6) RESULTS: SYSTEM TIME

From the perspective of system time, the investigation examines the compression and decompression times of various algorithms quantified in milliseconds, as shown in Figure 7. The findings reveal discernible discrepancies in processing duration among the algorithms. Some algorithms, such as Zstd, Zlib, and Gzip, exhibit distributions characterized by minimal variance and low median values, indicative of superior compression and decompression speeds. Conversely, other algorithms, such as Brotli, Bz2, and Lzma, display notable outliers and elevated data variance, suggesting varying responsiveness contingent upon the nature of the compressed data. This comprehensive examination provides insights into the performance and stability of each algorithm, facilitating informed algorithm selection for diverse compression scenarios.

Two algorithms (Brotli and Zlib) demonstrate the least gasUsed values and are singled out for further consideration. Therefore, examining boxplot graphs depicting the performance of the Brotli and Zlib compression algorithms, organized by averages, underscores a notable contrast in performance and execution time stability. The averages and medians of work times serve as indicators of overall speed and productivity, with algorithms boasting lower values being preferable for applications necessitating heightened performance and swift response times. Analysis of the distribution of work time, including outlier detection, provides insights into the stability and reliability of each algorithm across diverse application scenarios. These findings should be considered pivotal when selecting a data compression algorithm tailored to the application's specific needs and performance requirements.

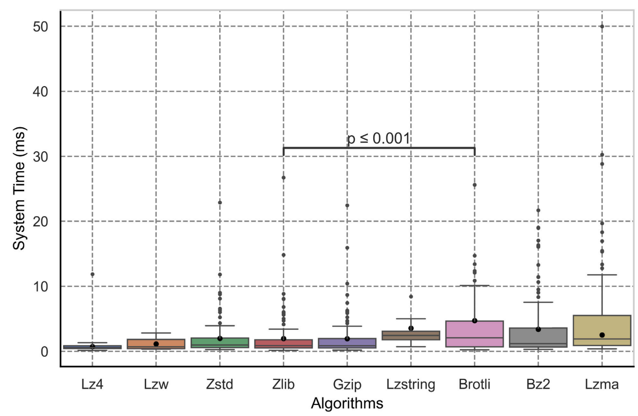


FIGURE 7. System Times of Brotli and Zlib are not normal distributions ($p < 0.05$). Moreover, the system time of Zlib is significantly less than Brotli ($p < 0.05$).

C. BALANCING GAS PRICE AND SYSTEM TIME

Transaction fees are determined by gasPrice and gasUsed. Our research aims to reduce gasUsed through text compression. The gasPrice is predetermined for each transaction. Because of the correlation between gasPrice and system time, this experiment investigates their relationship and recommends adjusting gasPrice to decrease transaction fees.

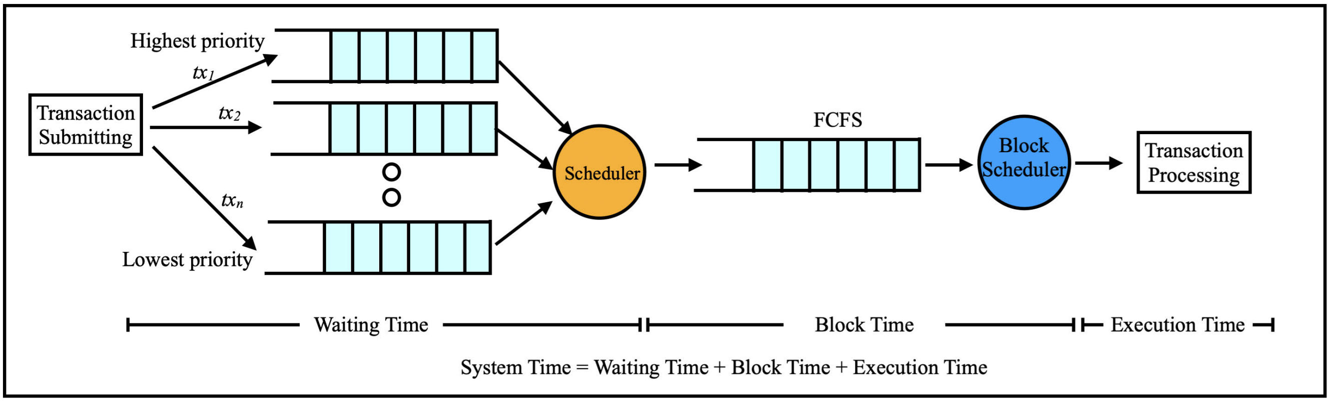


FIGURE 8. The queuing model of the Ethereum platform and its system time. In addition, the compression time and decompression time are added to the system time when evaluating the performance of our approach.

TABLE 1. Initial parameter of the experiment II.

Parameters	Initial data
dataset	100 transactions
data length	117 to 1,254 bytes
GasPrice	[20, 21, 22, 23, ..., 30] GWei
block time	12 s
Gas Limit	2,5000,000
wallet	100 addresses

1) TESTBED AND SCENARIOS

The outcomes of this investigation are pivotal in elucidating the influence of gasPrice settings on the efficiency of blockchain transactions. This research is crucial for optimizing transaction fees in blockchain application development. Furthermore, it underscores the value of utilizing a testnet like Ganache for risk-free experimentation and development, providing valuable insights into the performance and scalability of blockchain-based applications. Table 1 outlines the parameters pertinent to this study.

Table 1 elucidates the dataset, detailing 100 transactions alongside their respective data lengths. To experiment, the average gas price sourced from EtherScan is set at 25 GWei. Thus, this study systematically varies this parameter within a range of $\pm$ 10 percent of the average value. Moreover, relying on EtherScan data (https://etherscan.io), the block time is 12 seconds. The experiment initiates the gas limit at 2,500,000 to accommodate rapid block creation.

2) PERFORMANCE METRIC: SYSTEM TIME

Our study investigates the relationship between gasPrice and transaction system time within the Ethereum platform and found that Quan-Lin Li modeled the blockchain queue system based on a single queue and two stages of batch services [35]. On the other hand, this paper models Ethereum's transaction scheduling system and employs two distinct queue models, as Figure 8 illustrates. The first model utilizes a multilevel queue system, akin to scheduling arrival transactions based on a Poisson distribution. Within this model, the multilevel queue is segmented based on gasPrice, with transactions featuring the highest gasPrice prioritized for processing. In instances where transactions possess identical gasPrice, a first-come-first-serve (FCFS) approach is employed. In contrast, the second queue model adopts a fixed-length queue structure contingent upon reaching either block time or gas limit thresholds.

Ethereum's scheduler comprises two queueing systems. When transactions share the same gasPrice value, the scheduling mechanism follows an FCFS approach. This entails ensuring that the sum of all gasUsed within a block does not surpass the gasLimit. Ethereum initiates a new block if this limit is exceeded and processes the preceding block. Similarly, if the gasUsed amount is insufficient, Ethereum generates a new block and processes the preceding block upon reaching the block time threshold. Although such scenarios are rare due to varying gasPrice values among transactions, a multilevel queuing mechanism addresses this. Within multilevel scheduler implementations, transactions with higher gasPrice values are prioritized. If the sum of gasUsed exceeds the gasLimit, Ethereum creates a new block. In cases where the highest queue level exhausts its transactions, the scheduler selects transactions from the next highest queue level. While awaiting the creation of a new block, if a transaction with a higher gasPrice emerges, the scheduler elevates the priority of subsequent transactions, resulting in prolonged waiting times for lower-priority transactions.

In this context, system time represents the duration elapsed from transaction initiation to completion and subsequent inclusion within a block. The timestamps documenting these intervals are sourced from two distinct origins: start time is logged through code execution, while termination time is recorded via the inherent data structure of the Web3.py library.

3) RESULTS: SYSTEM TIME

This research investigates the relationship between gasPrice, denoted in Gwei, and the system time required for logging data into a smart contract. A notable inverse correlation

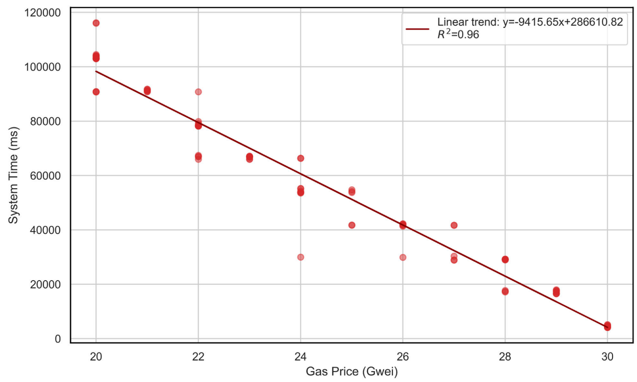


FIGURE 9. The linear regression function between gasPrice (GWei) and system times (ms) and the compression ratios ($y = -9,415.65x + 286,610.81$, $R^2 = 0.9600$).

between gasPrice and system time is elucidated through linear trend line analysis, as depicted in Figure 9. This suggests that transactions with higher gasPrice values are processed more expeditiously and are promptly incorporated into blocks. This analysis's derived $R^2 = 0.9600$ underscores a robust correlation, providing further evidence that gasPrice significantly influences transaction execution speed within the network.

This experiment highlights a linear correlation between system time and the variable gasPrice. As gasPrice increases, system time accelerates correspondingly, as evidenced by an approximate increment of 9 milliseconds from the baseline observation.

D. FINDINGS

This study delves into the correlation between data volume and the efficacy of the Zlib algorithm's compression, employing a combination of simulation and mathematical analysis to elucidate this relationship. The aim is to provide insights into predicting the interplay between these factors. Leveraging a logarithmic model, the research achieves an $R^2 = 0.9449$, indicating the robust adaptability of the model in elucidating variations in compression rates relative to data size. The analytical process incorporates introducing a horizontal line at $y = 1$ on the graph in Figure 6, delineating the threshold beyond which compression efficacy diminishes, thereby facilitating a clearer comprehension of the algorithm's performance across different data magnitudes.

The findings underscore the logarithmic model's efficacy in accurately capturing the relationship between data size and the compression efficiency of the Zlib algorithm. Consequently, this research furnishes significant insights into selecting optimal compression strategies within blockchain scheduler management. The study contributes to the broader discourse on applying compression algorithms by highlighting the importance of algorithmic efficiency in handling expansive datasets. It emphasizes the necessity of tailored algorithm selection based on specific data characteristics.

Furthermore, this research advocates for gasPrice as the primary determinant for transaction processing. The methodology involves gathering daily average gasPrice data and estimating system time, encompassing compression, waiting, blocking, execution, and decompression time within the blockchain network. Minimizing system time for transactions necessitates a strategic approach, wherein the gasPrice associated with a transaction is adjusted to secure placement in the higher priority queue.

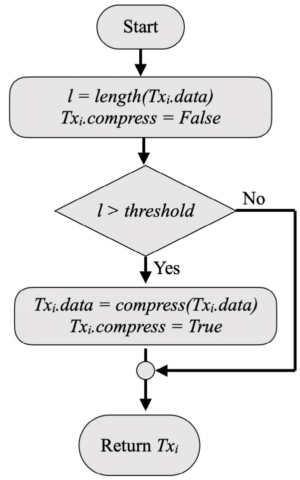


FIGURE 10. Flowchart of transaction fees minimization.

IV. TRANSACTION FEES MINIMIZATION

This section introduces an algorithm to minimize the Ethereum transaction fees by integrating a data compression mechanism before transmitting transactions to the Ethereum network. The experimentation conducted herein predominantly evaluates performance metrics. The gasUsed is quantified to indirectly assess the fees, while potential time increments resulting from implementing the proposed algorithm are estimated.

A. PROPOSED ALGORITHM

Utilizing the linear regression function depicted in Figure 6, we ascertain a threshold for determining whether textual data requires compression before transaction submission to the blockchain network. The compression ratio (CR) of the Zlib algorithm is expressed by the equation $y = -0.3383 \log(x) + 2.6753$, with a coefficient of determination (R^2) of 0.9449. When the compression ratio equals 1.0, the corresponding data length x is calculated as $exp[(1.0 + 0.3383)/2.6753] \approx 141$ bytes. Consequently, data exceeding 141 bytes undergo compression, while data below this threshold remains uncompressed. The flow chart of our algorithm is shown in Figure 10.

B. TESTBED AND SCENARIOS

The dataset utilized in this experiment is structured in JSON format, mirroring the setup of the previous experiment. It encompasses 100 orders with data lengths ranging

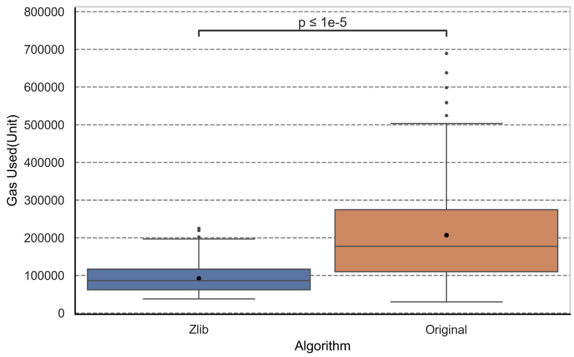


FIGURE 11. Average gasUsed between the *Zlib* algorithm and the original data. The gasUsed of are not normal distribution (*p < 0.05*), and paired comparisons with pairwise Mann-Whitney U tests (*U = 6,894.0, p < 0.05*).

from 115 to 1,278 bytes. To mitigate the influence of external network latency, the smart contract is deployed on Ganache, facilitating the isolation of variables impacting transaction speeds and reliability.

In the experiment, these 100 orders are the control variable submitted to the blockchain network. Subsequently, the same dataset undergoes processing via flowchart in Figure 10 to assess and potentially compress text based on length. The subsequent section will meticulously collect and analyze performance metrics resulting from these operations, such as gasUsed and system time.

Although the primary objective of this paper revolves around minimizing transaction fees, it is imperative to note that gasUsed directly corresponds to these fees, as articulated in (1). For this experiment, the gasPrice is held constant at 25 GWei, aligning with the average gasPrice. The performance metrics under this study are gasUsed and system time, with further elaboration on their significance provided in Section III-B2 of the manuscript.

C. RESULTS: GAS USED

The experimental investigation compared gas consumption with and without data compression, showing a notable decrease in gas usage when the compression algorithm was employed. This reduction is visually demonstrated, as compression results in shorter data length and lower gas consumption than uncompressed data. Figure 11 highlights a clear difference between the original text and the algorithm's results. Further statistical analysis confirms a significant difference between the approaches, including a normal distribution test and a pairwise Mann-Whitney U test conducted on the mean gasUsed *(n = 100)*. Specifically, the Mann-Whitney U test, with a sample size of 100, indicates a substantial disparity in the average gasUsed *(p < 0.05)*, validating the effectiveness of the compression algorithm in reducing gas consumption.

This experiment concludes that text compression techniques can effectively decrease gasUsed in transactions. Additionally, if the length of the text is below 141 bytes, the algorithm will bypass text compression due to a compression

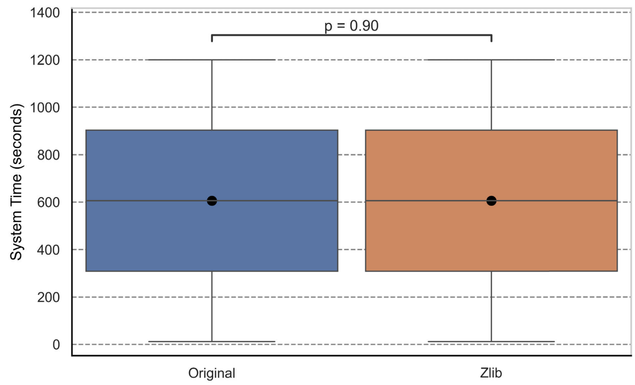


FIGURE 12. Average system time between the *Zlib* algorithm and the original data. The *system time* of are not normal distribution (*p < 0.05*), and the paired comparison with pairwise Mann-Whitney U tests (*U = 4,950.0, p > 0.05*) indicates that the average system time is not different significantly.

ratio exceeding 1.0, potentially leading to an increase in gasUsed.

D. RESULTS: SYSTEM TIME

The comparison of average system time between the original baseline and data processed through our algorithm yields a statistically insignificant result ($p > 0.05$). This indicates that incorporating our algorithm, which compresses textual data before transaction submission to the Ethereum network, has no discernible impact on overall processing time. Furthermore, a similar lack of significance is observed when comparing system time with overall time, as depicted in Figure 12.

V. DISCUSSIONS

A. TEXT COMPRESSION ALGORITHMS

This paper endeavors to reduce transaction fees within the home delivery system, a topic explored in our previous research. The first factor of transaction fees is gasUsed for each transaction. The data length causes an increase or decrease in gas usage. Our dataset comprises textual data in JSON format stored in our smart contract. We inspect to identify the most suitable text compression algorithm for our dataset. Among the top-performing algorithms for minimizing gasUsed in our dataset are the DEFLATE-based algorithms (Zlib and Gzip) and advanced techniques (Brotli and Zstd algorithms).

In theory, Brotli offers a higher compression ratio, but its compression process tends to be slower than DEFLATE-based algorithms. The Zstd strikes a balance between compression ratio and speed. Conversely, Zlib and Gzip algorithms prioritize speed while adjusting compression ratios. The Zlib is typically utilized for library compression, while Gzip is favored for file compression.

Our study's empirical findings indicate that both Brotli and Zlib algorithms exhibit the lowest average gas usage, with no significant difference. Due to its designed fast processing, the Zlib algorithm boasts a shorter system time compared

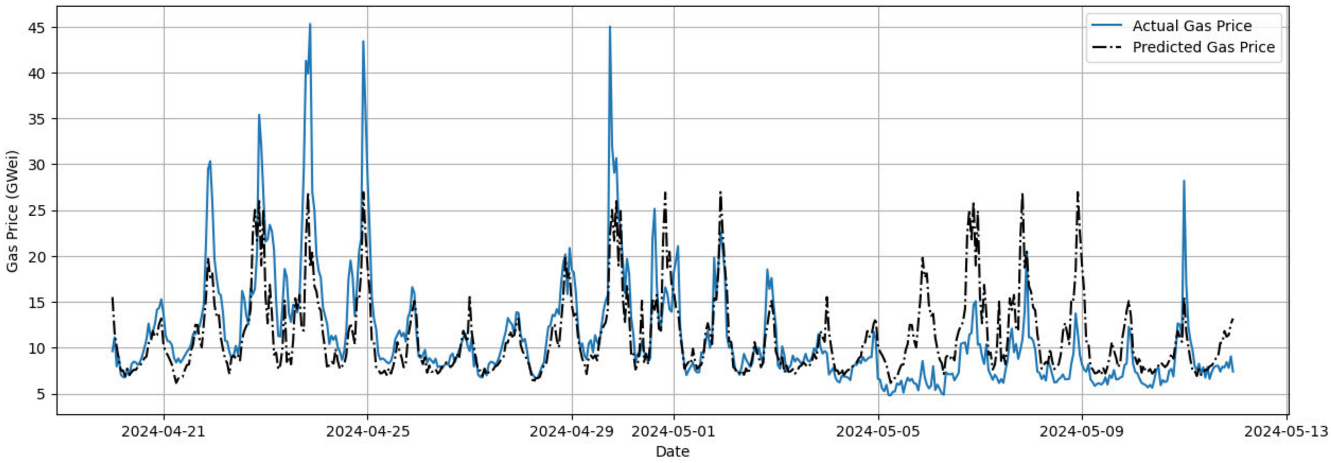


FIGURE 13. The non-linear regression function between average gasPrice (GWei), week of day, and hours (RMSE = 4.67).

to Brotli. Considering the trade-off between gasUsed and system time, the Zlib algorithm emerges as the optimal solution for our work based on this dataset.

B. TRANSACTION FEES MINIMIZATION

The second determinant of transaction fees lies in gasPrice. Our paper illustrates the correlation between gasPrice and system time, revealing that system time indirectly increases with gasPrice. Transactions with higher gasPrice are prioritized over those with lower gasPrice, ensuring timely processing. To expedite transaction processing, a transaction can set its gasPrice higher than the average gasPrice at the time of submission.

In our analysis, we examined gasPrice trends on the useweb3.xyz website between April 20, 2024, and May 12, 2024. We propose the non-linear regression model between the day of the week and hours to predict gasPrice with Random Forest Regression ($RMSE = 4.67$). Figure 13 illustrates the periodic growth of gasPrice, with a noticeable 24-hour cycle. GasPrice tends to start near its peak during midday and gradually decrease until reaching its lowest point around 10 AM. Subsequently, gasPrice ascends to its peak in the evening, particularly from 8 PM to 12 AM. Additionally, gasPrice remains elevated during weekdays, reflecting higher network activity.

As depicted in Figure 9, our linear regression model indicates that for every 1 GWei reduction in gasPrice. System time is expected to increase by approximately 18.38 % relative to the baseline and decrease when gasPrice rises. This insight enables us to estimate system time and balance response time and fees incurred by the customer.

The distinction between public and private blockchain networks presents an intriguing issue. While it is true that transaction fees in private blockchains are non-existent, this benefit is offset by increased hardware costs, the requirement for specialized knowledge to maintain the blockchain server, and the need for redundancy to prevent server failures. These factors create a competitive landscape between cloud-based and server-based systems. Our research explores the feasibility of cloud-based solutions, acknowledging the necessity of transaction fees. Thus, we aim to investigate methods to reduce these transaction fees effectively.

C. LIMITATIONS

Using text compression techniques, we proposed, demonstrated, and evaluated transaction fee minimization for Ethereum blockchain technology. We simulated 100 realistic datasets and experimented with research for our framework on the local blockchain network. The effects of network speed and hardware resources were not evaluated. Thus, the experimental results regarding system time cannot be used as a reference in real-life applications.

The limitation of our experiment is that we execute all test cases on the blockchain simulation named Ganache. Testing on a real blockchain network can only evaluate the gas used. We cannot estimate the system time because the network bandwidth affects the experiment more than the response time of operation in the blockchain network.

VI. CONCLUSION

This study meticulously examines the efficacy and implications of data compression within blockchain systems, specifically focusing on gas consumption and comparing Zlib-compressed data to its original counterpart. We systematically evaluate our approach and draw meaningful conclusions by simulating 100 realistic datasets on the Ethereum simulator Ganache and employing rigorous statistical methodology. Our findings suggest that employing the Zlib algorithm for compressing textual data sets exceeding 141 bytes before transaction submission to the Ethereum network results in lower gasUsed than the baseline while system time remains relatively unchanged. Additionally, our analysis includes an investigation into the real gasPrice trends of the Ethereum network, culminating in the proposal of a non-linear regression model that predicts gasPrice for each hour based on the day of the week and time of day.

This research delves into the efficacy of Zlib data compression within blockchain systems, specifically focusing on Ethereum transactions. Utilizing the Ganache simulator to analyze 100 realistic datasets, we demonstrate that compressing textual data exceeding 141 bytes significantly reduces gas consumption without impacting system time. This study also introduces a non-linear regression model that predicts hourly gas prices based on weekly time cycles, providing a strategic tool for optimizing transaction costs. Our findings highlight the benefits of data compression in improving economic efficiency and system performance in blockchain operations, offering valuable insights and methodologies for both practical applications and further academic exploration.

REFERENCES

[1] Y. Q. Zhao, S. W. Zhu, J. F. Yu, and A. P. Wang, "Semantic service registration and retrieval for smart home," *Appl. Mech. Mater.*, vols. 263–266, pp. 1740–1744, Dec. 2012. [Online]. Available: https://www.scientific.net/AMM.263-266.1740

[2] J. Zhan, S. Dong, and W. Hu, "IoE-supported smart logistics network communication with optimization and security," *Sustain. Energy Technol. Assessments*, vol. 52, Aug. 2022, Art. no. 102052. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2213138822001047

[3] B. Jebari, M. Ghogho, and K. Ibrahimi, "Secure and privacy-preserving E-mobility service based on blockchain and hybrid smart contracts," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Jun. 2023, pp. 333–338.

[4] Y. Issaoui, A. Khiat, A. Bahnasse, and H. Ouajji, "Smart logistics: Study of the application of blockchain technology," *Proc. Comput. Sci.*, vol. 160, pp. 266–271, Jan. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050919316825

[5] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[6] V. Buterin. (Aug. 2023). *A Next-Generation Smart Contract and Decentralized Application Platform*. [Online]. Available: https://ethereum.org/whitepaper

[7] E. Tijan, S. Aksentijević, K. Ivanić, and M. Jardas, "Blockchain technology implementation in logistics," *Sustainability*, vol. 11, no. 4, p. 1185, Feb. 2019. [Online]. Available: https://www.mdpi.com/2071-1050/11/4/1185

[8] "Application of blockchain technology in logistics and supplychain management," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 11, no. 3, pp. 1727–1732, Mar. 2023. [Online]. Available: https://www.ijraset.com/best-journal/application-of-blockchain-technology-in-logistics-and-supplychain-management-466

[9] J. J. O'Brien, R. Cantrell, M. Brian, and M. Todd, "Blockchain—Based system and method for crowdsourced delivery," U.S. Patent 2019 0 266 568 A1, Aug. 29, 2019.

[10] A. Nazarian, M. Arana, and D. L. Prestegard, "Blockchain configuration for secure content delivery," U.S. Patent 10 992 454 B2, Apr. 27, 2021.

[11] B. Putta and D. C. Kar, "Enhancing supply chain efficiency through blockchain integration," in *Handbook of Research on Blockchain Technology and the Digitalization of the Supply Chain* (Advances in Logistics, Operations, and Management Science), T. Najar, Y. Najar, and A. Aloui, Eds., Hershey, PA, USA: IGI Global, Apr. 2023, pp. 180–205. [Online]. Available: https://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-6684-7455-6.ch009

[12] S. Nguyen, P. S.-L. Chen, and Y. Du, "Blockchain adoption in container shipping: An empirical study on barriers, approaches, and recommendations," *Mar. Policy*, vol. 155, Sep. 2023, Art. no. 105724. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0308597X23002579

[13] R. Jayanthi, S. Sundararajan, T. Alam, N. Garg, R. R. Singh, and M. Udhayamoorthi, "Building trust management using blockchain technology," in *Proc. Int. Conf. Sustain. Comput. Data Commun. Syst. (ICSCDS)*, Erode, India, Mar. 2023, pp. 1303–1309. [Online]. Available: https://ieeexplore.ieee.org/document/10104956/

[14] A. U. Nwosu and S. B. Goya, "Blockchain-based efficient and secured framework for logistic management," in *Proceedings of Third Doctoral Symposium on Computational Intelligence* (Lecture Notes in Networks and Systems), vol. 479, A. Khanna, D. Gupta, V. Kansal, G. Fortino, and A. E. Hassanien, Eds., Singapore: Springer, 2023, pp. 57–70. [Online]. Available: https://link.springer.com/10.1007/978-981-19-3148-25

[15] M. H. Naseem, J. Yang, T. Zhang, and W. Alam, "Utilizing fuzzy AHP in the evaluation of barriers to blockchain implementation in reverse logistics," *Sustainability*, vol. 15, no. 10, p. 7961, May 2023. [Online]. Available: https://www.mdpi.com/2071-1050/15/10/7961

[16] A. Laurent, L. Brotcorne, and B. Fortz, "Transaction fees optimization in the Ethereum blockchain," *Blockchain: Res. Appl.*, vol. 3, no. 3, Sep. 2022, Art. no. 100074. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S209672092200015X

[17] J. Yang, S. Gao, G. Li, R. Song, and B. Xiao, "Reducing gas consumption of Tornado cash and other smart contracts in ethereum," in *Proc. IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Dec. 2022, pp. 921–926.

[18] Z. Ye, U. Misra, J. Cheng, W. Zhou, and D. Song, "Specular: Towards secure, trust-minimized optimistic blockchain execution," 2022, *arXiv:2212.05219*.

[19] T. Chukleang and C. Jandaeng, "Security enhancement in smart logistics with blockchain technology: A home delivery use case," *Informatics*, vol. 9, no. 3, p. 70, Sep. 2022. [Online]. Available: https://www.mdpi.com/2227-9709/9/3/70

[20] A. Kiayias, E. Koutsoupias, P. Lazos, and G. Panagiotakos, "Tiered mechanisms for blockchain transaction fees," 2023, *arXiv:2304.06014*.

[21] S. Gao, Z. Peng, F. Tan, Y. Zheng, and B. Xiao, "SymmeProof: Compact zero-knowledge argument for blockchain confidential transactions," *IEEE Trans. Depend. Sec. Comput.*, vol. 20, no. 3, pp. 2289–2301, Jun. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9786741/

[22] K. Zhou, C. Wang, X. Wang, S. Chen, and H. Cheng, "A novel scheme to improve the scalability of Bitcoin combining IPFS with block compression," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 4, pp. 3694–3705, Dec. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9861741/

[23] X. S. Ha, T. H. Le, T. T. Phan, H. H. D. Nguyen, H. K. Vo, and N. Duong-Trung, "Scrutinizing trust and transparency in cash on delivery systems," in *Security, Privacy, and Anonymity in Computation, Communication, and Storage* (Lecture Notes in Computer Science), vol. 12382. Cham, Switzerland: Springer, 2021, pp. 214–227. [Online]. Available: http://link.springer.com/10.1007/978-3-030-68851-615

[24] V. F. da Silva, M. N. Coelho, B. N. Coelho, V. N. Coelho, and I. M. Coelho, "A home ledger approach for IoT enabled devices," in *Proc. 31st Int. Symp. Comput. Archit. High Perform. Comput. (SBAC-PAD)*, Campo Grande, Brazil, Oct. 2019, pp. 227–233. [Online]. Available: https://ieeexplore.ieee.org/document/8924162/

[25] R. Huang, J. Xiao, S. Cheng, R. Zhang, H. Zeng, X. Huang, and Z. Liao, "A new home-based pension mutual aid framework based on blockchain," in *Mobile Networks and Management* (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering), vol. 474, Y. Cao and X. Shao, Eds., Cham, Switzerland: Springer, 2023, pp. 32–47. [Online]. Available: https://link.springer.com/10.1007/978-3-031-32443-7

[26] J. E. de Azevedo Sousa, V. Oliveira, J. Valadares, G. D. Gonçalves, S. Moraes Villela, H. Soares Bernardino, and A. Borges Vieira, "An analysis of the fees and pending time correlation in ethereum," *Int. J. Netw. Manage.*, vol. 31, no. 3, May 2021, Art. no. e2113.

[27] M. Crochemore and T. Lecroq, "Pattern-matching and text-compression algorithms," *ACM Comput. Surveys*, vol. 28, no. 1, pp. 39–41, Mar. 1996, doi: 10.1145/234313.234331.

[28] B. Stecuła, K. Stecuła, and A. Kapczyński, "Compression of text in selected languages—Efficiency, volume, and time comparison," *Sensors*, vol. 22, no. 17, p. 6393, Aug. 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/17/6393

[29] Z. Zhao, J. Li, Z. Su, and Y. Wang, "GaSaver: A static analysis tool for saving gas," *IEEE Trans. Sustain. Comput.*, vol. 8, no. 2, pp. 257–267, Apr. 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9946440/

[30] E. Albert, P. Gordillo, A. Hernández-Cerezo, A. Rubio, and M. A. Schett, "Super-optimization of smart contracts," *ACM Trans. Softw. Eng. Methodology*, vol. 31, no. 4, pp. 1–29, Oct. 2022. [Online]. Available: https://dl.acm.org/doi/10.1145/3506800
[31] J. Correas, P. Gordillo, and G. Román-Díez, "Static profiling and optimization of Ethereum smart contracts using resource analysis," *IEEE Access*, vol. 9, pp. 25495–25507, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9348894/
[32] N. Grech, M. Kong, A. Jurisevic, L. Brent, B. Scholz, and Y. Smaragdakis, "MadMax: Analyzing the out-of-gas world of smart contracts," *Commun. ACM*, vol. 63, no. 10, pp. 87–95, Sep. 2020. [Online]. Available: https://dl.acm.org/doi/10.1145/3416262
[33] T. Li, Y. Fang, Z. Jian, X. Xie, Y. Lu, and G. Wang, "ATOM: Architectural support and optimization mechanism for smart contract fast update and execution in blockchain-based IoT," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 7959–7971, Jun. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9520821/
[34] R. Zamani, M. Parsa-Moghaddam, and M.-R. Haghifam, "Dynamic characteristics preserving data compressing algorithm for transactive energy management frameworks," *IEEE Trans. Ind. Informat.*, vol. 18, no. 11, pp. 7587–7596, Nov. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9689947/
[35] Q.-L. Li, J.-Y. Ma, and Y.-X. Chang, "Blockchain queueing theory," 2018, *arXiv:1808.01795*.

CHANANKORN JANDAENG received the joint B.Sc. and M.Sc. degree in computer science and Ph.D. degree in computer engineering from the Price of Songkla University, Thailand, in 2005 and 2012, respectively. He is currently a Researcher and a Lecturer with the School of Informatics, Walailak University, Thailand. His research interests include the Internet of Things, cybersecurity, intelligent network monitoring systems, data science, and artificial intelligence applied in cybersecurity and network monitoring.

TIRAJET CHUKLEANG received the B.Sc. degree in information technology from Suratthani Rajabhat University, Suratthani, Thailand, in 2020, and the M.Sc. degree in management of information technology from the School of Informatics, Walailak University, Nakhon Si Thammarat, Thailand, in 2024. During his master's studies, he worked as a Research Assistant, Contributing to the School's Academic Service. He specializes in computer programming with modern tools such as Node.js and Python. His research interests include distributed applications and the application of blockchain technology.

JONKSUK KONGSEN received the B.Sc. degree in chemical technology from Chulalongkorn University, in 1988, and the M.Sc. degree in computer and engineering management from Assumption University, in 1992. She is currently working as a Lecturer with the School of Informatics, Walailak University, Thailand. Her research interests include business information systems, community-based tourism, and pattern matching.

PEERAVIT KOAD received the Ph.D. degree in computational science from Walailak University, Nakhon Si Thammarat, Thailand. His research interests include science, technology, and innovation, especially in multi-paradigm data science and artificial intelligence, which can be applied in many fields that require statistical data analysis. He also received scholarships from the Junior Science Talent Project (JSTP) under the National Science and Technology Development Agency (NSTDA).

MAY THU received the B.E. degree in information technology from Hmawbi Technology University, Hmawbi, Yangon, Myanmar, in 2012, the M.E. degree in information technology from Yangon Technology University, Yangon, in 2014, and the Ph.D. degree in computer engineering from the Prince of Songkla University, Hat Yai, Thailand, in 2021. She is currently a Research Associate and a Lecturer with the Faculty of Engineering, Cambodia University of Technology and Science (CamTech), Phnom Penh, Cambodia. Her research interests include machine vision and image processing, complex data classification, human behavior analysis, cognitive science, machine learning, deep learning, and artificial intelligence.