sensors

MDPI

Article

Visualization with Prediction Scheme for Early DDoS Detection in Ethereum †

Younghoon Park * and Yejin Kim

Division of Computer Science, Sookmyung Women's University, Seoul 04310, Republic of Korea; yejinkim@sookmyung.ac.kr

* Correspondence: yh.park@sookmyung.ac.kr; Tel.: +82-2-2077-7326

† This paper is an extended version of our paper published in International Conference on Green and Human Information Technology, Bangkok, Thailand, 31 January–2 February 2023.

Abstract: Blockchain technologies have gained widespread use in security-sensitive applications due to their robust data protection. However, as blockchains are increasingly integrated into critical data management systems, they have become attractive targets for attackers. Among the various attacks on blockchain systems, distributed denial of service (DDoS) attacks are one of the most significant and potentially devastating. These attacks render the systems incapable of processing transactions, causing the blockchain to come to a halt. To address the challenge of detecting DDoS attacks on blockchains, existing visualization schemes have been developed. However, these schemes often fail to provide early DDoS detection since they typically display only past and current system status. In this paper, we present a novel visualization scheme that not only portrays past and current values but also forecasts future expected system statuses. We achieve these future predictions by utilizing polynomial regression with blockchain data. Additionally, we offer an alternative DDoS detection method employing statistical analysis, specifically the coefficient of determination, to enhance accuracy. Through our experiments, we demonstrate that our proposed scheme excels at predicting future blockchain statuses and anticipating DDoS attacks with minimal error. Our work empowers system managers of blockchain-based applications to identify and mitigate DDoS attacks at an earlier stage.

Keywords: blockchain; visualization; DDoS; polynomial regression; coefficient of determination

Citation: Park, Y.; Kim, Y. Visualization with Prediction Scheme for Early DDoS Detection in Ethereum. *Sensors* **2023**, *23*, 9763. https://doi.org/10.3390/s23249763

Academic Editors: Muhammad Khalil Afzal, Byung-Seo Kim and Rehmat Ullah

Received: 18 October 2023
Revised: 26 November 2023
Accepted: 5 December 2023
Published: 11 December 2023

1. Introduction

Recently, blockchain has become one of the essential data integrity solutions in many security-required applications [1,2]. These applications collect and manage data, which is safeguarded by blockchain-based security tools. In most cases, the data in such applications holds significant value, making financial or reputational damage a possibility in the event of a security breach. Therefore, while blockchain effectively preserves data integrity, additional security measures are necessary.

Within blockchain-based security systems, various types of attacks may emerge, including data leakage, theft of electronic wallets, and denial of service (DDoS) attacks. One common occurrence in these attacks is traffic flooding to the blockchain server, which can manifest not only during an attack but also before attacks commence. Attackers generate a large number of transactions to disrupt blockchain servers, resulting in massive traffic floods and gas consumption. Early detection is crucial in such scenarios, as identifying attack indicators in advance can minimize the extent of damage.

Several previous works have been developed to detect these types of attacks on blockchain-based systems. Most of these works involve measuring values related to attacks on blockchain systems [3]. For instance, Ji et al. measured the number of transactions to detect attacks on the Bitcoin network [4]. Similarly, Simon et al. proposed an intrusion

detection system for the Ethereum network by analyzing mining rewards [5]. Additionally, various other previous works focus on detecting DDoS attacks on blockchain systems. However, these schemes merely present and anticipate values without visualizing the system's status, making them less accessible to non-IT experts, such as system managers.

To address this issue, several visualization schemes that display the status of blockchain systems have been proposed. Many monitoring tools with visualization features for blockchains are available, such as Etherscan [6] and Ethviewer [7]. Furthermore, visualization tools for healthcare data management systems based on Hyperledger [7] and Logchain [8] have been introduced. Some applications that display the connection information of blockchain nodes have also been developed [9,10]. Nevertheless, these previous visualization schemes assist in attack detection but do not allow system managers to recognize attack indicators in advance due to their lack of predictive capabilities [11].

In this paper, we introduce a novel visualization tool for blockchain systems that includes a prediction scheme. Like existing visualization schemes, our proposed tool primarily provides visualization features that depict the current system status. In our research, we focus on the Ethereum private network as the blockchain and measure values relevant to DDoS attacks on the blockchain system, including gas consumption and the number of transactions in each block. We specifically anticipate the future number of transactions in each block and identify DDoS attacks by comparing the expected data with the actual data. Additionally, we measure the gas usage for block generation and compare it to the average to determine the occurrence of DDoS attacks. We validate the accuracy of our tool using various statistical methods.

The remainder of this paper is structured as follows: In Section 2, we first discuss the relevant studies, and Section 3 introduces the essential components of the proposed algorithms. Section 4 provides a system model, and Section 5 details our new visualization scheme. In Section 6, we demonstrate that our proposed scheme effectively detects DDoS attacks in the early stages. Finally, we conclude this paper in Section 7.

2. Related Works

In this section, we review previous works related to visualization schemes for blockchain.

2.1. Visualization for Blockchain-Based Systems

While there are several security visualization schemes for general network systems, those specifically designed for blockchains have been relatively scarce. In the early stages, graph-based visualization schemes for cryptocurrency were proposed. Works such as [12–14] focused on data analysis and forensic investigation of transactions but did not consider security-related values. Tharani et al. [9] later proposed an advanced version that detects malicious transactions in Bitcoin, including ransomware and illicit advertisements. Similarly, refs. [15,16] presented visualization schemes for Ethereum, addressing fraud and vulnerability issues in transactions and smart contracts.

Another type of visualization scheme involves displaying network topologies. Delgado-Segura et al. introduced a technique for finding the Bitcoin network topology using orphan transactions [17], and Johnson et al. proposed a scheme that visualizes the mining pool and detects DDoS using a game-theoretic approach [18]. Miller et al. developed an advanced topology for the mining pool, highlighting influential nodes [19]. Additionally, Maeng et al. visualized the topology for the Ethereum network [20]. However, these visualization schemes primarily depict the current status without predicting future values.

2.2. Prediction for Blockchain Values

Various monitoring tools exist for blockchain systems, such as [21,22]. In [23], the authors presented a monitoring system for Bitcoin that detects illegal proof-of-work blockchains. Chen et al. proposed a monitoring technology that detects Ponzi schemes on public blockchains. Additionally, monitoring tools for smart contracts in Ethereum were provided [24,25]. However, these tools lack a prediction scheme for fast attack detection.

Research on predicting the future status of blockchains has also been conducted. Zheng et al. defined metrics for measuring the current status of the blockchain [3], and Ji et al. proposed predicting the number of transactions in the blockchain based on machine learning [4]. Simon et al. anticipated mining rewards using polynomial regression for Ethereum. However, they did not consider time in predicting future values, possibly because the studied blockchains were public [5].

3. Backgrounds

3.1. Blockchain Structure

Blockchain is a chain of blocks that allows for the secure storage of all committed transactions using shared and distributed networks [26,27]. To achieve this, decentralization techniques and robust cryptographic schemes are employed. These measures guarantee the integrity of the data within the blockchain, ensuring that data cannot be tampered with illegally. Blockchain is characterized by four key concepts: Decentralization, Persistency, Anonymity, and Auditability [27,28].

Decentralization is a critical feature that safeguards data within the blockchain system. In traditional security schemes, important data are often managed in a centralized manner, where data are stored in a central database, and keys for encryption/decryption and digital signatures are generated by a central server. Additionally, security policies and key management methods are determined by the server, and users obtain keys from the server and must adhere to its rules.

However, centralized systems have inherent problems, including a single point of failure and concentration of power:

- Single point of failure: In centralized security systems, essential information, including security-required data and keys, is stored on a central server. If an attacker breaches the central server and gains control, they can illegally access and modify the stored data. Furthermore, an attack on the central server can lead to a complete system shutdown. Consequently, the central server becomes a prime target for attackers, and real-world security breaches on central servers are well-documented.
- Concentration of power: In most security systems, the central server has administrator authority. Therefore, the system manager possesses unrestricted access to data, enabling them to view and modify data as they see fit. Even without malicious intent, if the central server is compromised, all data on the server is at risk. These issues arise because administrative power is concentrated solely within the central server.

To address these limitations of traditional security systems, blockchain technology has been proposed. Blockchain ensures the perfect integrity of stored data through decentralization and robust cryptographic measures. Due to these properties, blockchain is now widely employed in systems where data integrity is paramount.

In a blockchain-based security system, data are stored in a decentralized manner across multiple nodes (assuming there are n nodes) instead of relying on a single server. This means that even if a user managing one node attempts to illegitimately alter data, the integrity of the data remains intact because the data on the other $n - 1$ nodes remain unaltered. For an attacker to modify blockchain data, more than half of the n nodes must be compromised. As the number of nodes grows and they are distributed globally, it becomes increasingly challenging for attackers to compromise more than half of the nodes.

However, the blockchain system must also ensure data integrity even when the number of nodes is small. To achieve this, an additional security measure is employed: the consensus algorithm. In blockchain systems, data or their hash values are stored within each block. To prevent illegal modifications, each block must be linked to the blockchain using a predetermined algorithm known as the consensus algorithm. Various consensus algorithms exist, such as Proof of Work (PoW) and Proof of Stake (PoS). Connecting blocks with the consensus algorithm is often a complex process. Thus, if an attacker attempts to modify data within the blockchain, they must regenerate blocks from the point of data

modification to the end block using the challenging consensus algorithm. This task is nearly impossible, ensuring the perfect integrity of the data.

3.2. DDoS on Blockchain

The three fundamental elements of security are confidentiality, integrity, and availability. As previously mentioned, blockchain effectively ensures the integrity of stored data. Furthermore, in most blockchain-based systems, additional encryption and key management schemes are implemented to enhance confidentiality. Consequently, most attacks on blockchain systems aim to disrupt availability, typically through Distributed Denial of Service (DDoS) attacks.

In the context of blockchain, the most common method for launching a DDoS attack involves flooding the network with transactions. Legitimate users generate normal transactions to input data into each block or send cryptocurrency to other users. However, in a DDoS attack, spam or fake transactions are sent to the blockchain server, leading to the rejection of normal transactions or even a system shutdown.

Several examples of DDoS attacks on blockchain systems have been documented. A DDoS attack on the Solana blockchain occurred for 17 h in December 2021 [29], and another attack took place in January 2022 [30]. Multiple attacks on the Ethereum blockchain have also been reported [31,32]. Table 1 presents well-known examples of DDoS attacks on blockchain systems.

Table 1. Well-known Examples of DDoS on Blockchain.

Date	Blockchain	Attack
February 2021	EXMO	Service stopped for 5 h
December 2021	Solina	Network was offline for 17 h
December 2021	Arbitrum	Network was offline for 45 min
2022	Solina	DDoS was occurred for 3 times

Numerous solutions exist to prevent or mitigate DDoS attacks on blockchain systems. These solutions can be categorized into proactive and reactive methods. Proactive methods include retaining sufficient storage, memory, and network bandwidth, among other strategies. Reactive methods aim to mitigate DDoS attacks after they occur. Within the realm of reactive solutions, visualization tools that display the blockchain server's status have been developed to detect attacks earlier. However, existing visualization tools only reveal the current state of the blockchain, which may not be sufficient for rapid DDoS response. In this paper, in addition to the current status, we present future values for the blockchain status to enable quicker responses to DDoS attacks.

4. System Architecture

In this section, we introduce the entire system architecture. Figure 1 provides an overview of the entire system. The blockchain server generates various data while it runs, and the Decentralized Application (DApp) collects the data essential for predicting attacks. The collected data are stored in a database, and the prediction module calculates the future values for possible attacks. The results of the predictions are also stored in the database. Finally, the collected and predicted data are presented in plots to anticipate possible attacks using the visualization scheme.

Now, let us describe the detailed roles of the components.

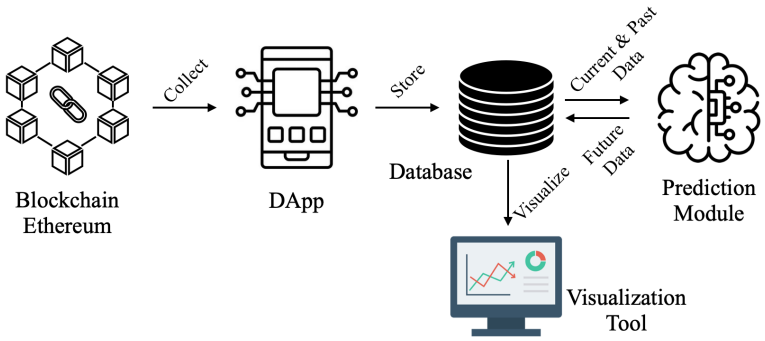


Figure 1. Overview of the Entire System.

- Blockchain Server: The blockchain server is a powerful computer that runs a blockchain, generating various data. This server must be connected to the internet. Among the data generated, some are related to the blockchain, and others are not. The DApp, which will be explained in the next paragraph, collects data that are essential for predicting future attacks. In this paper, we use Ethereum for a private blockchain and employ Geth to build the blockchain network. The collected data are closely related to attack attempts. For example, `gasUsed` represents the amount of gas used to create a new block.
- Decentralized Application (DApp): The DApp is a web application that operates directly on the blockchain or communicates directly with the blockchain RPC interface as a decentralized client [33]. The DApp's purpose in this paper is to collect blockchain information related to various attacks. The collected data for this paper are listed in Table 2. Additionally, for this work, we build the DApp with Node.js v18.16.1 and Web3 v0.20.6.
- Database: After the DApp collects data for intrusion detection, it sends the data to the database. The database collects information from the initial state to the current state of the blockchain. Furthermore, this database stores future data expected by the prediction module. Afterward, the collected and expected data are delivered to the monitoring tool to visualize the current state and check for any attacks.
- Prediction Module: The prediction module in this work aims to forecast the future values based on the current and past values on the blockchain server. The prediction module receives the current and past values for the blockchain server's status from the database. After the prediction, the future values are sent back to the database. The detailed procedure for the prediction module will be shown in Section 5.
- Monitoring Tool: After the database collects not only past and current data but also expected future data, it sends them to the monitoring tool. The monitoring tool displays this data through plots over time. Additionally, our monitoring tool issues a warning when it suspects an intrusion attack based on abnormally high future values. In this work, we build the monitoring tool with Node.js v18.16.1 and Grafana v9.2.1.

Table 2. List of Collected Data from Ethereum Server.

Collected Data	Meaning
`blockNumber`	Each block's number
`difficulty`	An integer value how difficult it is to mine a block
`gasUsed`	A total gas used by all transactions in a block
`size`	The size of a block
`timestamp`	An unix timestamp when a block was collected
`totalDifficulty`	An integer value of how difficult it is to mine whole blocks from initial block to the current block
`transactionNumber`	The number of transactions in a block

5. Proposed Detection Scheme

In this subsection, we will explain the detailed procedures of the system. The process can be classified into three steps. First, data collection from the Ethereum blockchain and saving the extracted data to the Maria DB are operated. Then, to use multiple linear regression to predict the number of transactions and the amount of gas, we selected particular data based on the degree of correlation. Finally, we visualize the current data in a solid line and the future data in a dotted line.

5.1. Data Collection

Several models have been proposed for monitoring the blockchain system. Sayadi et al. detect abnormal electronic transactions in the Bitcoin system [34], and they extract Bitcoin transaction data, block size, difficulty, hash rate, transaction volume, median time for transaction confirmation, and the number of unique Bitcoin transactions per day. In addition, in [35], the authors extract the number of transactions and the volume of gas consumption.

We determine the data to be extracted from the blockchain by referring to these previous studies and select data for future data prediction additionally. For the data collection process, we use the Web3 Node.js library to connect to the Ethereum node via HTTP. We extract blocks and transaction data from Ethereum using the `Web3.eth.getBlock()` function. Finally, we collect the data listed in Table 2: `blockNumber`, `difficulty`, `gasUsed`, `size`, `timestamp`, `totalDifficulty`, and `transactionNumber`. Among them, `gasUsed` and `transactionNumber` are used to predict attacks earlier.

The collected data are delivered to a database. In our work, we employ MariaDB to store them. The saved data are used for forecasting future values and will be displayed in plots by a monitoring tool.

5.2. Prediction with Future Values

DDoS attack on blockchain is carried out as an attack through transaction flooding. By producing a lot of spam or false transactions, the system needs more time to validate false transactions, which results in a network overload. Therefore, for anomaly detection of DDoS attacks, we focus on the number of transactions per block, `transactionNumber`. In this paper, we detect the DDoS attack by predicting the future `transactionNumber` values.

In addition, gas consumption in the Ethereum refers to a blockchain's transaction fee. In this work, we collected it as a name of `gasUsed`. This gas is paid to network validators for the requested services to the blockchain. If the network is congested, gas prices might be high and thus gas consumption will increase. Therefore, we focus on not only the number of transactions but also the gas consumption to check the traffic. In this work, we predict the DDoS attack by measuring `gasUsed` value.

We first choose the values that are used to predict the future values of `transactionNumber`. In terms of the future `transactionNumber`, we select current and past `difficulty`, `gasUsed`, `size`, `transactionNumber`, and time value. Of course, the current and past `transactionNumber` values influence the future `transactionNumber` value. In addition, `transactionNumber` also depends upon the time, because in most cases, more transactions emerge in the daytime, and the amount of the transactions decreases at night. Then, we demonstrate that the other properties `difficulty`, `gasUsed`, and `size` is also strongly related to `transactionNumber` by experiments.

Figure 2 indicates scatter plots for `transactionNumber` subject to `difficulty`, `gasUsed`, and `size`. In addition, as Table 3, we calculate the correlations between `transactionNumber` and `difficulty`, `gasUsed`, and `size`. From them, we conclude that `gasUsed` and `size` are strongly related to `transactionNumber`. In addition, although `difficulty` has a relatively lower correlation to `transactionNumber` than other factors, the correlation value is near 0.5, so we can conclude that `difficulty` has also a relation to `transactionNumber`.

Using the above four blockchain values and time, we predict the future `transactionNumber`. In the polynomial regression, for modeling, the prediction module gets the four blockchain value from MariaDB and calculates the coefficients of the polynomial regression

in Figure 1. After the modeling is down, the prediction module collects the current and past `transactionNumber`, `difficulty`, `gasUsed`, and `size`, and measures current time. With these information, the module calculates the future `transactionNumber` using the polynomial regression.

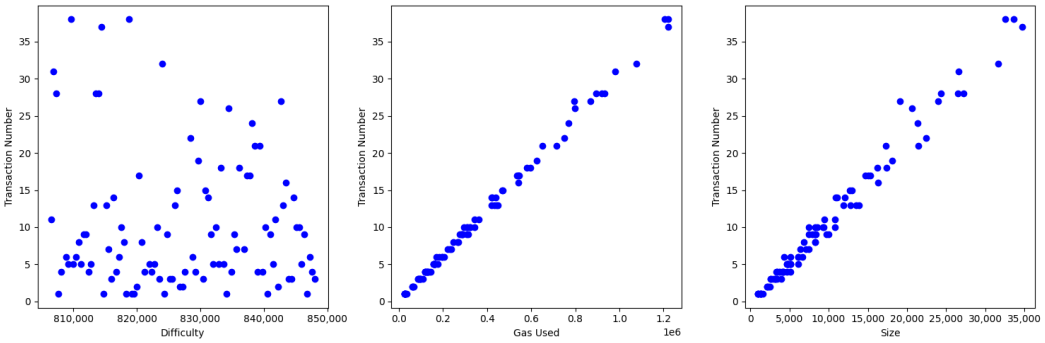
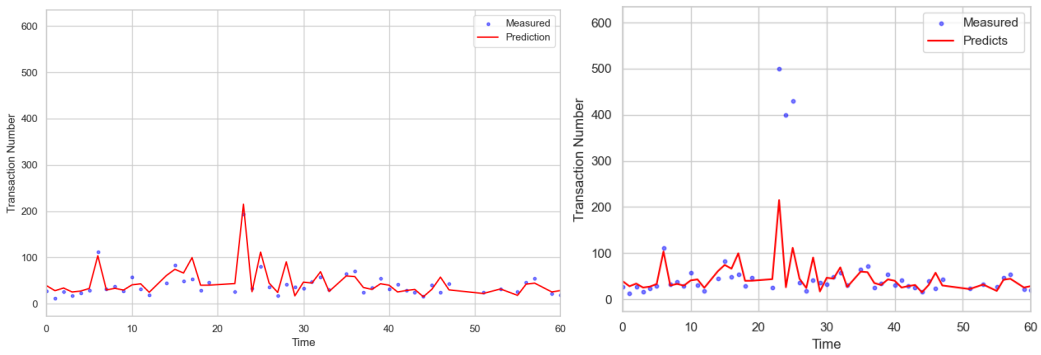


Figure 2. Scatter plots for `transactionNumber` subject to `difficulty`, `gasUsed`, and `size`.

Table 3. Correlations between `transactionNumber` and `difficulty`, `gasUsed`, `size`.

	`difficulty`	`gasUsed`	`size`
`transactionNumber`	0.4835	0.9996	0.9998

Figure 3 indicates the visualization results for an hour that show not only the past and current `transactionNumber`, but also the future values. The blue dots mean the current values and the red lines are the graphs for future values. Figure 3a is the result for the normal case. As the figure, the predicted values are nearly matched to the measured ones. From this, our proposed scheme predicts `transactionNumber` well. In addition, for the second case, we observe a DDoS attack on the blockchain system at 20 min. As Figure 3b there are gaps between the blue dots and red line around 20 min. By using this, we can predict the DDoS early.



(**a**) (**b**)

Figure 3. Visualization for `transactionNumber` for 1 h; (**a**) Case for normal transactions; (**b**) Case for DDoS attack.

Figure 4 is the graph for `transactionNumber`. The solid line is the graph for current and past values, and the dotted line is one for future values. We draw this graph using Grafana with the data in MariaDB. By using this graph, we can expect the near future `transactionNumber` value, so we will be able to detect early when a DDoS attack begins.

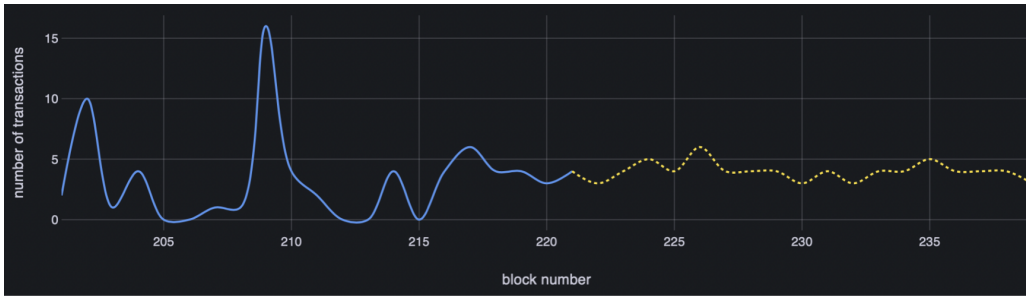


Figure 4. Graph for past, current and future values of `transactionNumber`.

5.3. Detection with Statistical Method

In the previous subsection, we provide a prediction scheme by expecting `transactionNumber` with a visualization tool. However, gaps between the expected value and the actual measured value can occur due to various normal reasons. Therefore, a false positive can be determined. To overcome this misjudgment, we supplement one more detection scheme using a statistical method.

In this paper, we provide another detection scheme using the interquartile range (IQR) method with `gasUsed` value. In Ethereum, 'gas' refers to the computational cost of processing transactions on the network. It is consumed when a transaction is submitted or when cryptocurrency is transmitted to other users. Therefore, if the amount of the gas becomes much larger than the average amount, we can expect that a DDoS attack will occur. In fact, if the gas is consumed more than the expected amount, the transaction or sending the cryptocurrency can become invalid. Therefore, measuring the usage of gas is another important concern in predicting a DDoS attack on the blockchain.

In this work, we first measure the usage of gas for several days and calculate the average amount. Next, we use the Interquartile Range (IQR) method to detect DDoS attacks. In our experiment, the top quarter amount of the gas usage is 4,566,790, so we determine that there is a DDoS attack on our blockchain system when the amount of gas usage is larger than that value. We demonstrate the validity of this method for detecting DDoS attacks with fewer false positives and false negatives in the next section.

6. Discussion

In this section, we demonstrate that our proposed prediction scheme, utilizing both visualization and statistical-based methods, accurately detects DDoS attacks. To assess the performance of the prediction scheme using the visualization tool, we examine the discrepancies between the expected and real values. Additionally, we verify the validity of the statistical method by calculating accuracy and F1 score.

We will first discuss the accuracy of the prediction scheme. As previously mentioned, to evaluate the effectiveness of our prediction scheme, we calculate the difference between the measured values and the expected values. This involves computing the Mean Squared Error (MSE) and the coefficient of determination (R^2). As our proposed scheme predicts more accurate values, MSE must decrease. In addition, because the MSE is in the numerator of the equation of R^2, R^2 approaches to 1 as our work expects well. Anticipating accurate future values implies that our scheme effectively detects normal cases, leading us to conclude that the likelihood of false positives in DDoS detection is low.

The MSE is calculated as follows:

$$MSE = \frac{1}{2}\sum_{i=0}^{n}(y_i - \hat{y}_i)^2,$$

where n is the number of measurements, y_i is the measured i-th value, and $\hat{y}_i$ is the expected i-th value. In the prediction scheme, we calculate the expected value of `transactionNumber`, making y_i represent the i-th `transactionNumber`, and $\hat{y}_i$ represent the expected value. To obtain the MSE, we run the Ethereum server for one hour, measure the real `transaction-`

`Number`, and calculate the expected `transactionNumber`. This experiment is conducted to demonstrate that our proposed prediction scheme performs well under normal circumstances, with normal transactions.

Additionally, we calculate the coefficient of determination (R^2). We collect data in the same manner and compute the following equation:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{(y_i - \overline{y})^2},$$

where $\overline{y}$ is the average of y_i values. In other words, $\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$.

Table 4 displays the MSE and the Coefficient of Determination for `transactionNumber`. Smaller MSE values indicate more accurate predictions. The calculated MSE is 0.0215, significantly smaller than the average value of `transactionNumber`. As shown in Table 3, the average number of transactions in each block is approximately 30. This low MSE demonstrates the accuracy of our polynomial-regression-based prediction scheme. We can further support this by calculating the coefficient of determination. As the scheme makes more accurate predictions, the coefficient of determination approaches 1. In Table 4, the R^2 value is 0.9999, indicating that the prediction scheme works effectively.

Table 4. Mean Squared Error and Coefficient of Determination for `transactionNumber`.

	MSE	R^2
`transactionNumber`	0.0215	0.9999

Now, we propose an alternative scheme that predicts DDoS attacks by monitoring `gasUsed` using a statistical method. To validate the performance of this second scheme, we calculate True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) to derive metrics like $Accuracy$, $Precision$, $Recall$, and $F1-Score$. These metrics are calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \; Precision = \frac{TP}{TP + FP},$$

$$Recall = \frac{TP}{TP + FN}, \; F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}.$$

These metrics, all ranging from 0 to 1, approach 1 as the prediction scheme more accurately identifies DDoS attacks.

From the results shown in Table 5, our statistical-based DDoS prediction scheme successfully detects attacks with a high degree of accuracy.

Table 5. Performance Evaluation of Statistical Method.

	Accuracy	**Precision**	**Recall**	**F1-Score**
Value	0.8667	0.8	0.8	0.8

7. Conclusions

In this paper, we have presented a DDoS prediction tool that leverages both visualization techniques and a statistical method. Our prediction tool comprises a visualization module with value expectation techniques and a detection scheme based on a statistical method. The visualization module provides a clear presentation of the current and past values of the number of transactions in each block, as well as the expected future values. To accomplish this, we employ polynomial regression to forecast these future values. In addition, our statistical method focuses on measuring the amount of gas used in creating each block. By calculating the gap between the current amount and the average amount, we can detect DDoS attacks at an earlier stage.

Our work demonstrates that the prediction scheme in the visualization tool accurately predicts future values, as indicated by the small Mean Squared Error (MSE). Furthermore, we have shown that the statistical method can effectively detect DDoS attacks with a low rate of false positives and false negatives. With the implementation of our tool, we can identify DDoS attacks on blockchain systems at an early stage, thereby mitigating potential damage caused by such attacks.

Author Contributions: Conceptualization, Y.K.; Methodology, Y.K.; Software, Y.K.; Investigation, Y.P.; Writing—original draft, Y.P.; Supervision, Y.P.; Project administration, Y.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2022-00207391, Development of Hashgraph-based Blockchain Enhancement Scheme and Implementation of Testbed for Autonomous Driving).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

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

References

1. Zhang, Y.; Xu, C.; Lin, X.; Shen, X. Blockchain-Based Public Integrity Verification for Cloud Storage against Procrastinating Auditors. *IEEE Trans. Cloud Comput.* **2021**, *9*, 923–937. [CrossRef]
2. Li, S.; Xu, C.; Zhang, Y.; Du, Y.; Chen, K. Blockchain-Based Transparent Integrity Auditing and Encrypted Deduplication for Cloud Storage. *IEEE Trans. Serv. Comput.* **2023**, *16*, 134–146. [CrossRef]
3. Zheng, P.; Zheng, Z.; Luo, X.; Chen, X.; Liu, X. A Detailed and Real-Time Performance Monitoring Framework for Blockchain Systems. In Proceedings of the 2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP), Gothenburg, Sweden, 25 May–3 June 2018; pp. 134–143.
4. Ji, S.H.; Baek, E.J.; Shin, M.G.; Park, J.S.; Kim, M.S. A Study on the Prediction of Number of Bitcoin Network Transactions Based on Machine Learning. *KNOM Rev.* **2019**, *22*, 68–76.
5. Simon, Jeyasheela Rakkini.; Geetha, K. Block Mining reward prediction with Polynomial Regression, Long short-term memory, and Prophet API for Ethereum blockchain miners. *ITM Web Conf.* **2021**, *37*, 01004. [CrossRef]
6. Available online: https://etherscan.io/ (accessed on 18 October 2023).
7. Available online: http://ethviewer.live/ (accessed on 18 October 2023).
8. Song, J.; Nang, J.; Jang, J. Design of Anomaly Detection and Visualization Tool for IoT Blockchain. In Proceedings of the 2018 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 12–14 December 2018; pp. 1464–1465. [CrossRef]
9. Tharani, J.S.; Charles, E.Y.A.; Hóu, Z.; Palaniswami, M.; Muthukkumarasamy, V. Graph Based Visualisation Techniques for Analysis of Blockchain Transactions. In Proceedings of the 2021 IEEE 46th Conference on Local Computer Networks (LCN), Virtually, 4–7 October 2021; pp. 427–430. [CrossRef]
10. Shrestha, A.K.; Vassileva, J. Bitcoin Blockchain Transactions Visualization. In Proceedings of the 2018 International Conference on Cloud Computing, Big Data and Blockchain (ICCBB), Fuzhou, China, 15–17 November 2018; pp. 1–6. [CrossRef]
11. Kim, Y.; Park, D.; Eom, H.; Ko, K.; Park, Y. Implementation of Blockchain Visualization with Prediction for DDoS. In Proceedings of the 2023 11th International Conference on Green and Human Information Technology (ICGHIT), Bangkok, Thailand, 31 January–2 February 2023.
12. Haslhofer, B.; Karl, R.; Filtz, E. O Bitcoin Where Art Thou? Insight into Large-Scale Transaction Graphs. In Proceedings of the International Conference on Semantic Systems, Leipzig, Germany, 12–15 September 2016.
13. Reid, F.; Harrigan, M. An Analysis of Anonymity in the Bitcoin System. In *Security and Privacy in Social Networks*; Altshuler, Y., Elovici, Y., Cremers, A.B., Aharony, N., Pentland, A., Eds.; Springer: New York, NY, USA, 2013; pp. 197–223. [CrossRef]
14. Zhao, C.; Guan, Y. A Graph-Based Investigation of Bitcoin Transactions. In Proceedings of the International Conference on Digital Forensics, Seoul, Republic of Korea, 6–8 October 2015.
15. Hu, T.; Liu, X.; Chen, T.; Zhang, X.; Huang, X.; Niu, W.; Lu, J.; Zhou, K.; Liu, Y. Transaction-based classification and detection approach for Ethereum smart contract. *Inf. Process. Manag.* **2021**, *58*, 102462. [CrossRef]
16. Patel, V.; Pan, L.; Rajasegarar, S. Graph Deep Learning Based Anomaly Detection in Ethereum Blockchain Network. In Proceedings of the International Conference on Network and System Security, Melbourne, Australia, 25–27 November 2020; pp. 132–148.

17. Delgado-Segura, S.; Bakshi, S.; Pérez-Solà, C.; Litton, J.; Pachulski, A.; Miller, A.; Bhattacharjee, B. TxProbe: Discovering Bitcoin's Network Topology Using Orphan Transactions. In *Financial Cryptography and Data Security*; Goldberg, I., Moore, T., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 550–566.
18. Johnson, B.; Laszka, A.; Grossklags, J.; Vasek, M.; Moore, T. Game-Theoretic Analysis of DDoS Attacks Against Bitcoin Mining Pools. In *Financial Cryptography and Data Security*; Böhme, R., Brenner, M., Moore, T., Smith, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; pp. 72–86.
19. Miller, A.K.; Litton, J.; Pachulski, A.; Gupta, N.; Levin, D.; Spring, N.; Bhattacharjee, B. *Discovering Bitcoin' s Public Topology and Influential Nodes*; University of Maryland: College Park, MD, USA, 2015. Available online: https://www.cs.umd.edu/projects/coinscope/coinscope.pdf (accessed on 18 October 2023).
20. Maeng, S.; Essaid, M.; Lee, C.; Park, S.; Ju, H. Visualization of Ethereum P2P network topology and peer properties. *Int. J. Netw. Manag.* **2021**, *31*, e2175. [CrossRef]
21. Dinh, T.T.A.; Wang, J.; Chen, G.; Liu, R.; Ooi, B.C.; Tan, K.L. BLOCKBENCH: A Framework for Analyzing Private Blockchains. In Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD '17), New York, NY, USA, 21–25 August 2017; pp. 1085–1100. [CrossRef]
22. Kalodner, H.; Möser, M.; Lee, K.; Goldfeder, S.; Plattner, M.; Chator, A.; Narayanan, A. BlockSci: Design and applications of a blockchain analysis platform. In Proceedings of the 29th USENIX Security Symposium (USENIX Security 20), Anaheim, CA, USA, 12–14 August 2020; pp. 2721–2738.
23. Weber, I.; Gramoli, V.; Ponomarev, A.; Staples, M.; Holz, R.; Tran, A.B.; Rimba, P. On Availability for Blockchain-Based Systems. In Proceedings of the 2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS), Hong Kong, 26–29 September 2017; pp. 64–73. [CrossRef]
24. Marino, B.; Juels, A. Setting Standards for Altering and Undoing Smart Contracts. In *Rule Technologies. Research, Tools, and Applications, Proceedings of the 10th International Symposium, RuleML 2016, Stony Brook, NY, USA, 6–9 July 2016*; Alferes, J.J., Bertossi, L., Governatori, G., Fodor, P., Roman, D., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 151–166.
25. Chen, T.; Li, X.; Luo, X.; Zhang, X. Under-optimized smart contracts devour your money. In Proceedings of the 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER), Klagenfurt, Austria, 20–24 February 2017; pp. 442–446. [CrossRef]
26. Salah, K.; Rehman, M.H.U.; Nizamuddin, N.; Al-Fuqaha, A. Blockchain for AI: Review and Open Research Challenges. *IEEE Access* **2019**, *7*, 10127–10149. [CrossRef]
27. Krichen, M.; Ammi, M.; Mihoub, A.; Almutiq, M. Blockchain for Modern Applications: A Survey. *Sensors* **2022**, *22*, 5274. [CrossRef] [PubMed]
28. Kouhizadeh, M.; Sarkis, J. Blockchain Practices, Potentials, and Perspectives in Greening Supply Chains. *Sustainability* **2018**, *10*, 3652. [CrossRef]
29. Bodziony, N.; Jemioło, P.; Kluza, K.; Ogiela, M.R. Blockchain-Based Address Alias System. *J. Theor. Appl. Electron. Commer. Res.* **2021**, *16*, 1280–1296. [CrossRef]
30. Available online: https://cryptopotato.com/solana-network-suffers-another-reported-ddos-attack/ (accessed on 10 October 2023).
31. Essaid, M.; Kim, D.; Maeng, S.H.; Park, S.; Ju, H.T. A Collaborative DDoS Mitigation Solution Based on Ethereum Smart Contract and RNN-LSTM. In Proceedings of the 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), Sejong, Republic of Korea, 18–20 September 2019; pp. 1–6. [CrossRef]
32. Abou El Houda, Z.; Hafid, A.S.; Khoukhi, L. Cochain-SC: An Intra- and Inter-Domain Ddos Mitigation Scheme Based on Blockchain Using SDN and Smart Contract. *IEEE Access* **2019**, *7*, 98893–98907. [CrossRef]
33. Zhang, L.; Kim, D. A Peer-to-Peer Smart Food Delivery Platform Based on Smart Contract. *Electronics* **2022**, *11*, 1806. [CrossRef]
34. Sayadi, S.; Ben Rejeb, S.; Choukair, Z. Anomaly Detection Model Over Blockchain Electronic Transactions. In Proceedings of the 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, Morocco, 24–28 June 2019; pp. 895–900. [CrossRef]
35. Bogner, A. Seeing is Understanding: Anomaly Detection in Blockchains with Visualized Features. In Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and of the 2017 ACM International Symposium on Wearable Computers (UbiComp '17), New York, NY, USA, 11–15 September 2017; pp. 5–8. [CrossRef]