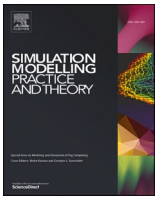
Simulation Modelling Practice and Theory 142 (2025) 103131

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat



Comparing blockchain and DAG technologies for smart agriculture traceability in terms of efficiency and latency

Antonio Villafranca [a], Igor Tasic [b], Victor Gallegos [c], Almudena Gimenez [c], Jesus Ochoa Rego [c], Juan Antonio Fernandez [c], Maria-Dolores Cano [a,*]

[a] Department of Information and Communication Technologies, Universidad Politécnica de Cartagena, 30202 Cartagena, Spain
[b] Faculty of Economics and Business, UCAM Universidad Católica San Antonio de Murcia, 30107 Murcia, Spain
[c] Department of Agricultural Engineering, Universidad Politécnica de Cartagena, 30202 Cartagena, Spain

ARTICLE INFO

Keywords:
Agritech
Bitcoin
Blockchain
DAG
Ethereum
IoT
Security
Traceability

ABSTRACT

Distributed Ledger Technologies (DLT), such as Bitcoin, Ethereum, and Directed Acyclic Graphs (DAG), are being positioned as a promising solution for smart agriculture by enabling secure, decentralized, and transparent traceability systems. However, these technologies face challenges related to scalability, latency, and efficiency in IoT environments. In this study, we conduct a comparative analysis of Bitcoin, Ethereum, and DAG technologies through extensive simulations, varying transaction generation rates and network latencies. A key methodological innovation of this research is the detailed codification of agricultural data transactions, encompassing parameters such as crop type, fertilization, harvesting, and transportation, enabling a structured and scalable approach to data representation. Our results reveal that Bitcoin's robustness is hindered by its high sensitivity to latency and network load, with inclusion times exceeding 700 s. Ethereum demonstrates better adaptability, with controlled inclusion times ranging from 12.91 to 35.76 s under varying conditions. DAG outperforms both, achieving significantly lower inclusion times between 4.27 and 22.25 s, highlighting its suitability for real-time applications. To the best of our knowledge, this is the first study to provide a direct comparison of these technologies in the context of agricultural traceability, showcasing the advantages and limitations of DAG-based systems for managing and scaling agricultural IoT networks.

1. Introduction

The Internet of Things (IoT) represents an important technological advance in various sectors by enabling autonomous interconnection and communication between devices. These devices, equipped with sensors and actuators, collect, process, and transmit data through networks, facilitating innovative applications in areas such as industrial automation, healthcare, and connected cities. Effective IoT implementation requires a robust infrastructure to ensure security, privacy, and scalability, given the volume of data generated and the heterogeneity of the devices involved.

Smart agriculture, in particular, can greatly benefit from enhanced traceability, which allows tracking and verifying each stage of the supply chain, from production to final consumption, thus ensuring the quality and safety of agricultural products. Distributed Ledger Technologies (DLT), such as Bitcoin [1] and Ethereum [2], as well as Directed Acyclic Graphs (DAG), have emerged as potential

* Corresponding author.
E-mail address: mdolores.cano@upct.es (M.-D. Cano).

https://doi.org/10.1016/j.simpat.2025.103131
Received 23 January 2025; Received in revised form 2 April 2025; Accepted 26 April 2025
Available online 5 May 2025

solutions to improve traceability in the IoT domain. These technologies provide a decentralized and secure platform for data storage and management, eliminating the need for intermediaries and reducing risks associated with single points of failure [3,4]. Nevertheless, traditional blockchains face significant limitations in terms of scalability and energy efficiency due to increasing transaction demand and the resource-intensive nature of Proof-of-Work (PoW).

In response to these challenges, DAGs have emerged as an interesting alternative to traditional blockchains. DAGs, exemplified by IOTA's Tangle [5], offer a data structure that allows the simultaneous validation of multiple transactions, improving scalability and reducing energy consumption in devices. In addition to IOTA's Tangle, there are various consensus algorithms for graph formation in DAGs, accepting specific adaptations to diverse needs. This technology has been specifically designed for the IoT environment, where efficiency and the ability to handle large volumes of data are essential.

The key contributions of this study, described below, establish a clear foundation for understanding the performance trade-offs among these distributed ledger technologies within the context of IoT-enabled smart agriculture:

- Comparative analysis: We present the first direct comparison of Bitcoin, Ethereum, and DAG-based technologies for smart agriculture traceability under identical simulation conditions. This approach enables a fair evaluation of their performance in terms of transaction inclusion times and network load handling.
- Simulation-based evaluation: We employ extensive computer simulations, varying parameters such as transaction generation rate, number of agents/miners, transaction size, and network latency, to systematically assess how each technology performs under different operating scenarios.
- Performance trade-offs: Our analysis highlights the trade-offs in terms of efficiency, where efficiency is defined primarily by transaction inclusion times and the impact of network load. We show that Bitcoin experiences significantly higher inclusion times under increased latency and load, while Ethereum demonstrates better resilience. Meanwhile, DAG-based solutions, using a weighted tip selection algorithm (especially with $\alpha = 0.8$), achieve the lowest inclusion times, making them attractive for applications that require real-time or near-real-time performance.
- Practical insights: We discuss the practical implications of our findings for smart agriculture traceability. Our study provides valuable insights into how each technology performs under varying conditions, thereby identifying which systems are more suitable for real-time data processing and highlighting areas for further optimization in future research.

The remainder of the document is as follows. Section II presents the related work and background on DLT technologies. In Section III, we describe the comparative analysis methodology we have followed. Section IV presents the simulation results, and Section V discusses these results along with the limitations of our approach. Finally, Section VI concludes the study by summarizing the main outcomes.

2. Related works

In this section, we show the current advances in the use of blockchain technologies applied to smart agriculture. To start with, Lv et al. [6] and Yele et al. [7] investigated the use of blockchain to improve traceability and security in agricultural and food supply chains. Both studies implemented blockchain solutions to increase transparency and consumer trust, achieving significant results in improving data quality and reducing the time required for traceability. In the former, the authors explored blockchain applications, including platforms like Bitcoin, Ethereum, and Hyperledger [8], highlighting how these can enhance transparency and trust in agricultural supply chains. In the latter, the authors focused on using Hyperledger along with IoT and RFID technologies to improve security and traceability in restaurants, achieving a 70 % reduction in traceability time.

How blockchain-based food traceability systems affect purchase intention was studied in Tao et al. [9]. Through signaling theory and a structural equation model with a sample of 415 participants, they demonstrated that the implementation of blockchain significantly increases purchase intention, improving product quality perception, consumer trust, and environmental information transparency. By providing greater transparency and trust in the supply chain, these systems not only improve food safety but also respond to consumers' growing demands for high-quality and sustainable products and reinforces consumer confidence. Patel et al. [10] developed AgriOnBlock, an Ethereum-based system that connects various actors in the agricultural sector through smart contracts, improving the accuracy of traceability data and reducing operational costs and processing times. To reduce processing times, they implemented smart contracts that automate the necessary transactions and verifications between the involved parties. This eliminated the need for intermediaries, reducing processing times by 30 %. Additionally, a 20 % reduction in operational costs was achieved by decreasing intermediary intervention and facilitating automated supply chain management. The use of blockchain also improved traceability data accuracy by 25 %, ensuring that each stage of the process is immutably and verifiably recorded.

On the other hand, Ren et al. [11] proposed a dual blockchain solution to secure data in agricultural IoT networks, using Ethereum and the Interplanetary File System (IPFS) [12]. They designed a consortium blockchain called Agricultural Sample Data Chain (ASDC), based on Ethereum. This approach allowed data to be stored in IPFS, and the hash values of this data were stored in ASDC. Finally, the hash of ASDC blocks was uploaded to the main Ethereum chain to maintain a secure public record in case of malicious attacks. The results showed an efficiency five times greater in data loading compared to blockchain and 21.6 % superior to cloud storage, being able to write >200 hash values per second. Limitations include data redundancy and fluctuations in Ethereum gas prices, complicating cost estimation and causing potential network delays.

Li et al. [13] and Zheng et al. [14] explored the use of blockchain to improve coordination and the freshness of agricultural products. Li et al. [13] investigated the dynamic adoption of blockchain to maintain the freshness of fresh products in the agricultural

supply chain. They used blockchain to improve product traceability and visibility, optimizing storage and transportation operations, and incorporated smart contracts to automate and secure transactions along the supply chain, ensuring that products remained fresh through continuous monitoring of storage and transportation conditions. This technology could reduce the average transportation time of fresh products from 4 hours to 3 hours and increase the monthly traffic of fresh products from 50 tons/month to 60 tons/month. A sustainable agricultural product logistics integration model that allowed visualization and traceability of information throughout the product life cycle, from production to consumption, addressing issues such as supply and demand desynchronization, quality management, and price fluctuations was presented in [14]. The model integrated three essential flows (information, resources, and data) to promote the sustainable development of the agricultural logistics sector in China. Comparative experiments show that blockchain implementation could increase additional revenue by at least 40 % and reduce product deterioration during storage and transportation.

Similarly, Rehman *et al.* [15] and Hasan et al. [16] used blockchain technology integrated with IoT sensors to enhance smart agriculture, focusing on continuous monitoring of key parameters and ensuring food safety. In the former, the authors implemented a private blockchain along with IoT sensors to improve data-driven decision-making in smart agriculture. Their model monitored parameters such as temperature, humidity, and light exposure in agricultural fields through sensors and processes this data through a Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol to optimize energy consumption and network stability. The results showed that the use of different fertilizers significantly affected the physical properties of the soil, such as temperature and humidity, and that blockchain implementation could reduce environmental degradation. In the latter, the authors used Ethereum blockchain integrated with IoT sensors to ensure sustainable and reliable food production. They use smart contracts to manage various aspects of the agricultural process, such as participant registration, crop management, anomaly identification and resolution, and product certification. IoT sensors installed on farms continuously monitor soil, air, and crop conditions, providing real-time data securely stored on blockchain and IPFS. This integration allowed consumers to verify the provenance and authenticity of agricultural products by scanning QR codes, ensuring that the products meet sustainable certifications such as organic, non-GMO, and Good Agricultural Practices (GAP). Thus, it addressed issues of misleading labels and fraud in the food industry, promoting trust and responsibility in food production and distribution.

Despite the advantages presented by these studies, several common limitations were identified. The scalability of blockchain systems is affected by the high demand for computational and energy resources. Interoperability between different blockchain systems remains a significant challenge. Additionally, there are barriers related to initial implementation costs and resistance to adoption by farmers due to the complexity and lack of familiarity with blockchain technology [1–10].

It is important to note that most smart agriculture solutions with blockchain employ the Ethereum model with smart contracts rather than Bitcoin-based models for several key reasons. Firstly, Ethereum offers a robust and flexible platform for implementing smart contracts, which automate processes in the agricultural supply chain, such as monitoring crop conditions and verifying sustainability certifications without the need for intermediaries. Moreover, Ethereum has an active development community and a broad ecosystem of decentralized applications (dApps). This facilitates the creation and integration of customized solutions for smart agriculture. Ethereum's ability to handle multiple types of transactions and its focus on flexibility and scalability make it ideal for the dynamic and varied needs of modern agriculture, where precise and transparent data and resource management are crucial.

In contrast, DAG technology has emerged as a viable alternative. Although initially primarily targeted at IIoT [17], there are some works that use DAG for traceability in smart agriculture, demonstrating high performance. Hasan et al. [18] used the public blockchain network Fantom with an aBFT consensus mechanism, showing greater speed and efficiency compared to Ethereum and Hyperledger Fabric. Fantom allowed transactions to be completed faster and more economically, outperforming other platforms in terms of transaction and verification time. Hasan et al. implemented a peer-to-peer blockchain architecture that integrated Self-Sovereign Identity (SSI) and a Decentralized Key Management System (DKMS), providing a reliable and secure traceability service for agricultural food products in supply networks. Li et al. [19] proposed an improved traceability model based on DAG, using an enhanced witness-based consensus mechanism that employs a Verifiable Random Function (VRF) and SM2 threshold signatures to improve security and efficiency in the traceability of agricultural products. The model proved capable of reducing centralization and shortening the time required to reach consensus, significantly improving the system's security and scalability. Li *z* managed to increase the system's throughput, demonstrating the ability to handle a high volume of transactions efficiently. Similarly, Watanabe et al. [20] combined DAG technology with the Ethereum blockchain to improve traceability in supply chain systems. They used DAG to structure tokens, allowing efficient tracking of token histories without needing to examine the entire blockchain. This combination facilitated complex operations such as token merges and splits, improving token history recovery and management through smart contracts on the Ethereum platform.

Even so, DAG technologies have also encountered limitations, such as challenges related to the scalability and cost of Blockchain networks, the need for a comprehensive regulatory framework to facilitate widespread adoption, interoperability challenges with legacy systems, and storage limitations in public networks. Despite these limitations, DAG technology has proven to be an efficient and reliable solution for traceability in smart agriculture.

2.1. Bitcoin

Bitcoin enables secure transactions without the need for a central authority. Proposed in 2008 by Satoshi Nakamoto and operational since 2009, it is based on a Peer-to-Peer (P2P) network where each node maintains a complete copy of the blockchain [21]. Nodes verify transactions and group them into blocks that are then added to the blockchain. The consensus mechanism used by Bitcoin is PoW, which involves solving complex mathematical problems that require a significant amount of computational resources. Miners

compete to solve these problems, and the first miner to find the valid solution has the right to add a new block to the blockchain and receive a reward in bitcoins. This process ensures that each block added to the blockchain is verified and approved by the majority of nodes, guaranteeing the integrity and security of the network.

Despite its innovations, Bitcoin has limitations in the IoT environment. Its processing capacity is limited to approximately 7 transactions per second due to the fixed size of blocks and the time required to mine each block [22]. The PoW process consumes a lot of energy, which is not sustainable for IoT devices with limited energy resources [23,24]. The transaction confirmation time can be several minutes, which is not suitable for IoT applications that require quick responses. Additionally, transaction fees can be high [23] during periods of high demand, making frequent microtransactions in IoT economically unfeasible.

2.2. Ethereum

Ethereum is a blockchain platform that extends the capabilities of Bitcoin's technology [25] by enabling the execution of smart contracts, which are computer programs designed to enforce a set of rules. Proposed in 2013 by Vitalik Buterin and launched in 2015, Ethereum is designed to be a programmable platform that allows developers to create and deploy decentralized applications. The Ethereum protocol also relies on a P2P network where each node maintains a complete copy of the blockchain. Transactions in Ethereum include not only value transfers but also the execution of smart contracts, which are programs that automatically execute when certain predefined conditions are met [23]. Like Bitcoin, Ethereum uses a consensus mechanism based on PoW. However, Ethereum is in the process of transitioning to a new consensus mechanism called Proof of Stake (PoS), known as Ethereum 2.0 [26]. This shift aims to improve scalability and reduce energy consumption. In PoS, validators are selected to create new blocks and validate transactions based on the amount of cryptocurrency they own and are willing to "stake" as collateral. Each block in the Ethereum blockchain contains a list of transactions, a hash of the previous block, a timestamp, and a nonce, like Bitcoin. However, Ethereum blocks can also contain executable code for smart contracts and related data.

Ethereum faces several limitations in the IoT environment. Its processing capacity is limited to approximately 15–20 transactions per second, which may be insufficient for IoT applications with a high volume of transactions. Although PoS promises to improve scalability, its implementation is still in progress. The energy consumption of PoW is considerable, although Ethereum 2.0 aims to reduce it with PoS, which requires less energy. The transaction confirmation time in Ethereum is faster than in Bitcoin, but it can still be too slow for IoT applications that require real-time responses [22]. Transaction fees, known as "gas," can be high during periods of congestion, making frequent microtransactions in IoT economically unfeasible. Additionally, storing smart contracts and their states on the blockchain can require a lot of storage, which is a challenge for IoT devices with limited resources [24].

2.3. DAG

DAG is a data structure that offers an alternative to traditional blockchains to improve scalability and reduce energy consumption. Unlike blockchains, where transactions are grouped into blocks added sequentially, in a DAG, transactions are processed individually and can be validated in parallel [23]. One of the most well-known implementations of DAG is IOTA's Tangle, specifically designed for the IoT environment. In the Tangle, each new transaction must approve at least two previous transactions, promoting a distributed validation network and eliminating the need for dedicated miners. This approach allows the network to grow more efficiently and scalably, as the number of approved transactions can increase exponentially as more devices join the network. In addition to IOTA's Tangle, there are various consensus algorithms for graph formation in DAGs, allowing specific adaptations to diverse needs [27,28].

The consensus protocol in a DAG, such as the Tangle, does not require the intensive energy use characteristic of PoW in Bitcoin or Ethereum. Instead, transactions are validated through a mutual approval mechanism, where each new transaction helps confirm previous transactions. This model significantly reduces energy consumption and allows IoT devices, which typically have limited resources, to actively participate in the network [29]. Each transaction in a DAG contains information about the transactions it approves, creating an interconnected network of transactions rather than a linear chain. This structure allows for greater concurrency and speed in transaction processing, which is crucial for IoT applications that require high processing capacity and low latency.

Despite their advantages, DAG technologies present some limitations. Searching for information in a graph can be more costly and complex compared to traditional blockchains due to the more distributed and less linear structure. They can also face security issues such as double-spending attacks due to parallel validation. Additionally, designing and maintaining a DAG is more complex, which can increase costs. Finally, the adoption of DAG technologies is limited, resulting in fewer tools and support resources available compared to traditional blockchains [28].

3. Simulation model

In this section, we will describe the applied methodology, indicating what we aim to simulate, the tools we have used for this purpose, and the configurations necessary to replicate the scenario. We have simulated different scenarios with the same parameters using three different data storage technologies in blockchains. For the simulated environment of Bitcoin and Ethereum, we used the BlockSim tool [30]. The simulation of the environment using DAG technology was performed with the DAGSim tool [31].

3.1. Blocksim

BlockSim is a discrete event simulation tool designed to analyze and evaluate blockchain systems. Implemented in Python,

BlockSim allows for the modeling of various aspects of blockchains through two main layers: the network layer and the consensus layer. This tool is highly extensible and can be used to simulate both Bitcoin and Ethereum, providing a flexible environment to investigate the behavior and performance of these technologies under different configurations.

It facilitates the simulation of complex scenarios in blockchain by modeling the propagation of blocks and transactions in the network as well as the execution of consensus algorithms. The network layer captures the blockchain nodes and the underlying P2P protocol for data exchange between nodes. The consensus layer includes the algorithms and rules adopted to reach an agreement on the current state of the blockchain ledger. The main configuration parameters of this tool are:

- Block Interval: It defines the average time required to generate a block. It is crucial for determining how frequently new blocks are added to the blockchain.
- Block Size: It specifies the block size, directly affecting the number of transactions that can be included in each block.
- Block Delay: It represents the delay in the propagation of blocks across the network. This parameter is important for modeling network latency.
- Transaction Size: It defines the size of each transaction, which affects the amount of data that must be processed and stored.
- Transaction Fee: It specifies the fee associated with each transaction, influencing the cost of conducting transactions on the network.
- Gas Limit: In Ethereum, this parameter sets the gas limit for transactions involving smart contracts, determining how much computational work can be performed in each transaction.
- Gas Price: In Ethereum, this parameter defines the gas price used to calculate transaction fees, directly impacting the cost of executing smart contracts and transactions.
- Number of Miners: It defines the number of active miners in the network. This parameter allows configuring the distribution of hash power, which is the total computational capacity that miners contribute to solving the consensus algorithms.
- Number of Transactions: Sets the number of transactions that will be generated during the simulation. This parameter is crucial for evaluating network performance under different workloads and determining how the network handles high transaction volumes.

3.2. DAGSim

DAGsim is an asynchronous, continuous-time, multi-agent simulation framework. It allows for the modeling of honest and semi-honest transactions, evaluating the robustness of DAG-based protocols against different agent strategies. Implemented in Python, it is extensible, allowing for the future inclusion of other protocols and malicious agents. The tool simulates scenarios where agents operate without the need for global synchronization, reflecting real decentralized networks. Each node acts as an independent agent that can issue transactions, providing a varied representation of strategies and behaviors in the network. It also allows for the modeling of networks with different topologies and degrees of connectivity, which are crucial for studying the impact on transaction confirmation and system robustness. It facilitates the inclusion of actors with different computational capacities and strategic behaviors, essential for developing and testing improvements in DAG protocols that resist adverse behaviors and enhance network security.

DAGsim incorporates multiple configurable parameters, such as the transaction arrival rate, the number of active nodes, and the tip selection algorithms, allowing adaptation to various scenarios. Additionally, it adjusts the communication latency between nodes, analyzing how proximity and latency affect transaction propagation and consensus. The main parameters are listed below:

- Number of Transactions: It sets the total number of transactions that will be processed during the simulation.
- Number of Agents: It determines the number of nodes in the network, responsible for validating and introducing transactions into the graph.
- Transaction Arrival Rate (λ): It defines the frequency at which transactions enter the system, modeled using a Poisson process to represent events occurring independently.
- Tip Selection Algorithm: It defines the method used to select transactions in the graph that will validate new transactions. The options are:
  - Random Selection: Tips are randomly selected from all previously unvalidated transactions in the graph.
  - Weighted Selection: Tips are chosen based on the cumulative weight of the transactions, influenced by factors such as the amount of work done to approve them or the number of times they have been referenced by other transactions.
  - Unweighted Selection: It combines elements from the previous methods, treating all transactions with equal importance regardless of their weight. Unlike random selection, it does not follow a random path but uses a systematic approach, traversing transactions from genesis in search of free tips to validate.
- Randomness in Tip Selection (α): When selecting the weighted tip selection algorithm, this parameter controls the degree of randomness. A higher value of α makes tip selection more deterministic, favoring transactions with greater cumulative weight.
- Distance: Specifies the proximity between agents in the network, affecting how and when nodes perceive each other's transactions. Shorter distances facilitate faster and more efficient information propagation.
- Latency: Modeled using a distance matrix, it determines how variations in distance affect the propagation time of transactions. This parameter can be modified to assess the impact of latencies in the network.

3.3. Simulated environment

This section details the simulated environment designed to evaluate the efficiency of different DLT technology approaches in managing and tracing agricultural data. For the purposes of this study, efficiency is defined solely by latency and network load, as these parameters are most critical in IoT-enabled smart agriculture. Other metrics, such as throughput, energy consumption, and storage overhead, are beyond the scope of this work and will be considered in future research. All necessary information has been gathered to understand the complete process of a crop, from the preparation phase to the transportation phase. Fig. 1 represents the simulated environment detailing the different stages of the crop.

As mentioned before, two main tools were used for the simulation framework, namely, BlockSim and DAGSim. BlockSim was employed to model Bitcoin and Ethereum-based systems, while DAGSim was used to simulate DAG-based systems. These tools enable realistic modeling of agricultural data management and traceability at different stages of cultivation. Each simulation was configured with 10,000 validator nodes for DAG and 10,000 miners for the Bitcoin and Ethereum models. This configuration aims to represent a large and distributed network, relevant for the simulated agricultural environment.

The simulation period spans 76 days, corresponding to the growth cycle of sea fennel, which was the real cultivated product, allowing for an evaluation throughout the entire process.

The frequency of transaction generation was modified in each simulation, as we consider this a crucial factor that will directly affect network load and information propagation speed. This parameter defines the generation of one transaction per day, the generation of 24 transactions per day, and the generation of 60 transactions per day, allowing us to see different configurations. For each transaction generation frequency configuration, different network latencies were introduced to analyze their effect on the efficiency and propagation of information. The evaluated latencies were 10 ms, 60 ms, and 120 ms. These latencies were selected to simulate varied network conditions, from optimal to challenging, reflecting real communication scenarios in distributed networks. During the cultivation period, various data can be captured at different stages of the agricultural process, including preparation, planting, harvesting, packaging, and transportation. These data could be collected using both IoT sensors and manual entries. In our study, the real data manually collected will be used as input data for the simulations.

Information entries are classified as public, private, or partially public, depending on the nature and sensitivity of the data. Public information would be accessible to all system participants, promoting transparency in the cultivation process. Private information would be restricted to authorized users, protecting sensitive data such as fertilization recipes or specific growth parameters. Partially public information would be accessible only to certain interest groups, balancing transparency and privacy according to needs.

The data logging frequency is crucial for the blockchain. IoT sensor readings, such as temperature and light levels, enable continuous monitoring and generate numerous transactions, requiring high processing and storage capacity. In contrast, manual entries, such as planting and harvesting dates, could be recorded only once. The frequency is set according to the criticality of the information and the system's capacity to avoid blockchain saturation and maintain optimal performance.

Table I describes the parameters recorded at each phase of the agricultural process, including the type of entry, the information captured, and the frequency of recording. Due to variations in data length and complexity depending on crop and environment, we fixed the transaction size at 512 bytes for our simulations. Since this limit may not allow encapsulating all details, we developed an encoding system to represent all necessary information in a compact, coded form within a single transaction. Each transaction includes a unique 64-byte ID, comprising a 32-byte hash for the crop cycle, a 16-byte timestamp, and a 16-byte field indicating the data collection phase (preparation, planting, harvesting, etc.), to differentiate between stages and crops. For instance, during the preparation phase, greenhouse details are encoded in 48 bytes (with 24 bytes for a unique greenhouse code and 24 bytes for GPS coordinates,

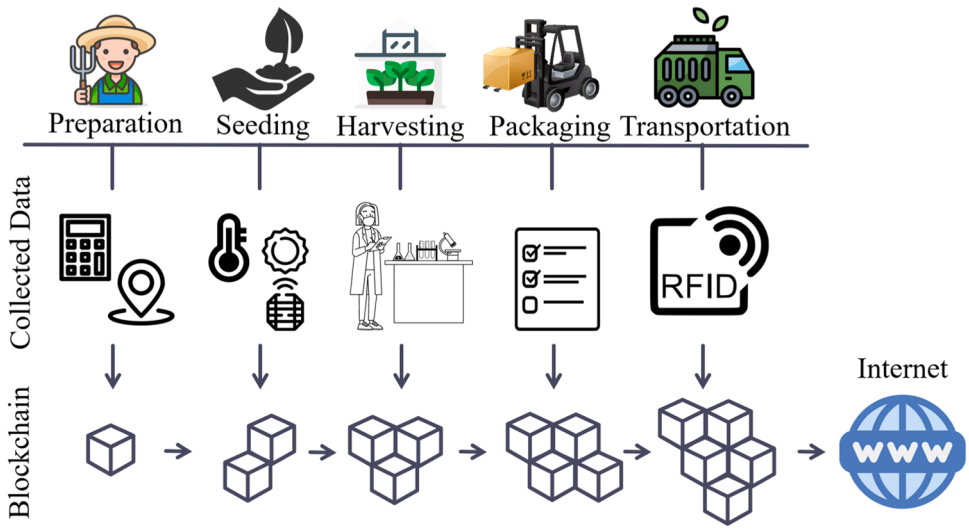


Fig. 1. Proposed environment for transaction generation in each phase of the cultivation process.

Table I
Collected data at each phase of the agricultural process.

Phase	Parameter	Type of Input	Examples	Type of Information	Frequency
Preparation	Greenhouse details	Manual	"Estación Experimental Agroalimentaria Tomás Ferro" (37.686376, −0.950268) of the "Universidad Politécnica de Cartagena" (Murcia, Spain)	Public	Once
	Crop type	Manual	Hydroponic (Floating system)	Public	Once
	Gutter (material and dimensions)	Manual	Styrofloat trays (0.6 $m \times 0.41$ m) floating beds (1.35 $\times$ 1.25 $\times$ 0.2 - $L \times W \times H$, respectively) dimensions	Private	Once
	Fertilisation, substrate, and growing media	Manual/ IoT	Growing media: 100 % peat leachate (pH 7.9, EC 3.6 dS m^{-1}) and 100 % compost leachate (pH 7.7, EC 3.1 dS m^{-1}) collected from wild rocket crop. NSC: 7.2 mM NO_3^-, 4.8 mM NH_4^+, 2 mM H_2PO_4-, 2.5 mM SO_42-, 6 mM $K+$, 1.9 mM Ca_2+, and 1.5 mM Mg^{2+}.	Partially public	Once
Seeding	Species	Manual	Sea Fennel (Crithmum maritimum L.)	Public	Once
	Irrigation system	IoT	Campbell CS547 sensors monitored the EC and temperature of the nutrient solution throughout the growing cycles. Amount of water applied: 200 L.	Partially public	Daily
	Fertigation	Manual/ IoT	Initial and weekly nutrient solution compostion were analysed and characterization by High-Performance Liquid Chromatography (HPLC)	Private	Weekly
	Lighting	IoT	PAR daily light integral (mol/m^2/d) and agrometeorological station (SIAM)	Private	Hourly
	Temperature	IoT	Goove Home measuring air temperature (min/max/avg) and humidity.	Private	Hourly
	Sowing date	Manual	4th of May 2022	Public	Once
	Plantation details	Manual/ IoT	256 plants/m^2	Private	Weekly
Harvesting	Number of harvests	Manual	1	Public	Once
	Harvesting date	Manual	18th of July 2022	Public	After each harvest
	Growth analysis	Manual/ IoT	Morphometric parameters: Leaves weight, leaves area, dry weight, root sytems, specific leaf area (SLA)	Private	After each harvest
	Pytochemical Analyses	Manual	Phenolic compounds, vitamin C, antioxidants, nitrate content, sensorial analysis, and head-space composition.	Partially public	After each harvest
Post harvesting processing	Disinfection	Manual	1 min in 150 ppm NaOCl solution (pH 6.5, 4 °C), then rinsed for 1 min in tap water (4 °C) to ensure final chlorine residue below 5 ppm.	Private	After each harvest
	Storage	Manual	7 days at 4 °C and 90 % RH	Partially public	Daily
Packaging	Material	Manual	Polypropylene, poly lactic acid	Public	Once
	Recyclability	Manual	Biodegradable	Public	Once
Transportation	Origin/ Destination	IoT	From "Estación Experimental Agroalimentaria Tomás Ferro" to restaurant or local market (GPS locations) or RFID reading	Public	Once
	Transport means	Manual	Electric vehicle	Public	Once
	Duration	IoT	Using RFID at origin and destination	Public	Once

Table II
Proposed encoding for crop types.

Crop Type	Code
Hydroponic	HYD001
Soil	SOI002
Aeroponic	AER003
Aquaponic	AQU004
Coco Coir	COC005
Earth	EAR006
Vermiculture	VER007
Bag culture	BAG008
Tower gardens	TOW009
Bioponic	BIO010
Rockwool	ROC011
Perlite	PER012
Vermiculite	VER013
Gravel culture	GRA014
Sand culture	SAN015
Mixed media culture	MIX016
Felt culture	FEL017

divided into latitude and longitude), while the crop type is captured in a 16-byte field using a unique alphanumeric code (as detailed in Table II). Additionally, material information is allocated 64 bytes (16 bytes for the material type and 48 bytes for dimensions), and a 320-byte package is reserved for fertilization, substrate, and cultivation medium data, divided evenly between fertilization/substrate details and the characteristics of the cultivation medium.

Table III details this encoding for the preparation phase, ensuring that all information fits within a 512-byte transaction. This approach allows for efficient and transparent management of agricultural data, ensuring that every step of the process is well-documented and verifiable.

For the planting phase, considering the previously mentioned ID, the specific fields for this phase will be encoded as follows. For the crop type, a 16-byte field has been selected to capture the taxonomic ID of the crop from the National Center for Biotechnology Information (NCBI) database [32]. This database provides a unique taxonomic identifier for each species, allowing for standardized and compact encoding. This taxonomic ID ensures that each crop type is identified accurately and efficiently. For the fertilization used, the composition of the nutrient solution (32 bytes) and the date and time of analysis (16 bytes) will be specified. This information is crucial for monitoring and adjusting the nutrients provided to the plants. For the light and temperature sensors, which measure these parameters, a 48-byte string will be encoded. Each string will include the type of sensor used (16 bytes), the date and time of the measurement (16 bytes), and the daily light level or temperature (16 bytes). For temperature, values will be recorded in a minimum-maximum-average format. For the planting date, 16 bytes are allocated to record the date and time of planting in the DDMMYYYY (8 bytes) and HHMMSS (8 bytes) format. This format ensures precise and compact encoding of the planting date and time. For planting details, 32 bytes are allocated. These bytes will capture the planting density (16 bytes), indicating the number of plants per square meter, and the planting dimensions (16 bytes). Table IV shows the proposed encoding for this stage.

The next phase is the harvest phase, which will be recorded with a single transaction once the planting stage is completed. This transaction will begin with the ID that identifies the crop it refers to. The number of harvests will be encoded in a 4-byte field. The growth analysis will be encoded in a 176-byte field. This field will include morphometric parameters such as leaf weight, leaf area, dry weight, root system, and Specific Leaf Area (SLA). Each of these parameters will be detailed with additional values to allow for a more complete analysis. For example, a leaf weight of 10 g will be encoded as "00,000,010″ a leaf area of 5 cm^2 as "00,000,005″. This field will include phenolic compounds, vitamin C, antioxidants, and nitrate content. Each of these subfields will have 65 bytes to allow for detailed encoding. For example, vitamin C content of 15 mg/g will be encoded as "VITC001500011000," allowing for the identification of the analyzed parameter and the measured amount. Table V shows an example of encoding for this stage.

For the packaging phase, the material field will occupy 64 bytes. Within this field, 32 bytes will be included for the type of material used in the packaging, such as "Polypropylene" encoded as "POLYPROPYLENE0000000000000000," and another 32 bytes for a unique material code, such as "MAT00100000000000000000000000." The recyclability field will have 16 bytes and will encode the recycling type of the material, such as "Biodegradable" encoded as "BIODEGRADABLE0000″ or "Recyclable" as "RECYCLABLE0000000."

For the transportation phase, the origin/destination field will occupy 64 bytes, using the same encoding as in the preparation phase. Twenty-four bytes will be allocated for GPS coordinates, with 12 bytes for latitude and 12 bytes for longitude, differentiating 1 byte for the sign and 11 bytes for the decimal value. This allows for a compact and precise representation of locations. The transportation medium field will have 32 bytes, where the type of vehicle used will be encoded. Instead of using long descriptions, codes will be assigned to different types of vehicles. For example, "Electric Vehicle" could be encoded as "EV001″ and "Truck" as "TRK002." Finally, the transportation duration field will have 32 bytes, encoding the duration in the format HHMMSSCC.

3.4. Simulation configuration

After explaining the simulated environment and the transaction sizes used, we will specify the parameters followed to configure both DAGSim and BlockSim for the different models evaluated. These configurations have been designed to realistically model the management and traceability of agricultural data, as detailed in the previous phases. The simulations were carried out on a computer equipped with Windows 11, 16 GB of RAM, a 13th generation Intel i7 processor, and a GTX 4060 graphics card with 6 GB. These specifications ensure the system's capability to handle the complexity and computational requirements of the simulated models. Table VI presents the parameters used in each tool for the simulation of each model.

4. Results

In this section, we present the results obtained from the various simulations conducted with the Bitcoin, Ethereum, and DAG

Table III
Proposed encoding for the preparation phase.

Collected data	Size	Encoding	Example
Greenhouse details	48 bytes	Greenhouse name	"Tomas Ferro" -> 0000,000,000,000,001
		GPS	1000,950,268 0376,863,760
Crop type	16 bytes	Type	HYD001
Gutter	64 bytes	Material	0000,001,928
		Dimensions (LengthxWidthxHeigh)	0135,000,000 0125,000,000 0020,000,000
Fertilisation, substrate, and growing media	320 bytes	Fertilisation and substrate	NSC: 7.2 mM “00,720,000″
		Growing media	Peat leachate: 100 % -> 0100,000,000

Table IV
Proposed encoding for the planting phase.

Collected data	Size	Encoding	Example
Species	16 bytes	NCBI ID	Crithmum maritimum -> "0000,000,040,916″
Irrigation system	32 bytes	Sensor Type	CSS54700000
		Amount of water applied	000,000,000,200
Fertigation	48 bytes	Composition	00,720,000
		Date	12,062,022
Lighting	48 bytes	Sensor Type	SIAM0000000000
		Date/Hour	120,520,220,900
		Values	010,020,015
Temperature	48 bytes	Sensor Type	GOOVE00000000
		Date/Hour	120,520,221,200
		Values	07,012,009
Sowing date	16 bytes	Date/Hour	07,052,022,095,544
Plantation details	32 bytes	Plants density/m^2	00,000,256
		Dimensiones	00,100,010

Table V
Proposed encoding for the harvest phase.

Collected data	Size	Encoding	Example
Number of harvests	4 bytes	Number	0001
Harvesting date	8 bytes	Date	28,072,022
Growth analysis	176 bytes	Leaf weight	00,000,010
		Leaf area	00,000,005
		Other measures	xxxxxxxx
Pytochemical Analyses	260 bytes	Phenolic compounds	OH00070000110000
		Vitamin C	VITC001500011000
		Other measures	xxxxxxxx

Table VI
Parameters used in each simulation.

Model	Parameter	Values
Bitcoin, Ethereum, DAG	Number of agents	10,000
	Transaction arrival rate	[1,24,60]
	# Total transactions	[92,2208,5500]
	Latencies	10, 60, 120ms
	Transaction size	512 bytes
DAG	Tip selection algorithm	Weighted, random
	Randomness in tip selection (α)	0.4, 0.8
Bitcoin	Block interval	600 s
	Block size	1 MB
Ethereum	Block interval	12.5 s
	Block size	1 MB

models. The results are analyzed based on several key parameters, including the number of transactions generated, the introduction of network latencies, and other relevant factors. This comparative analysis allows us to evaluate the efficiency and effectiveness of each technology in managing and tracing agricultural data, with a particular focus on transaction inclusion time and network load as measures of efficiency, providing a clear insight into how each model performs under different operating conditions.

The time taken for each transaction to be included in the blockchain or the graph has been measured. First, we will evaluate the inclusion times in each of the models, analyzing how this time varies depending on the number of transactions generated. Subsequently, we will assess how latency affects each of the models by introducing different levels of latency in the network to observe its impact on the performance and efficiency of transaction inclusion.

4.1. Bitcoin

Before analyzing the results obtained from the Bitcoin model simulations, it is important to establish the theoretical expectations to better understand the data.

On average, a block is mined every 10 min. Given that the size of a block is limited, and the processing capacity of the Bitcoin network is approximately 7 transactions per second, the average time for a transaction to be included in the blockchain can vary depending on network congestion. Theoretically, under optimal conditions and without congestion, a transaction could take about 10 min to be confirmed. However, during periods of high demand, this time can increase significantly due to the buildup of unconfirmed

transactions. With a minimum latency of 10 ms, the impact on the inclusion time of transactions in the blockchain will be almost negligible, remaining close to the expected average of 10 min, provided the network is not congested. With a moderate latency of 60 ms, block and transaction propagation times will increase slightly, causing minor delays in transaction inclusion, but generally not deviating significantly from the 10 min under normal conditions. With a high latency of 120 ms, the propagation time will be considerably greater, causing a notable increase in transaction inclusion time, especially if the network is under load, exceeding 10 min and becoming more evident with greater network congestion.

We present the results obtained for each transaction generation rate, evaluating the models with rates of 1, 24, and 60 transactions generated per day. Within each figure, there will be three subfigures representing the different latencies evaluated: 10 ms, 60 ms, and 120 ms. These subfigures will allow for a comparison of how the transaction inclusion time in the Bitcoin blockchain varies under different latency and network load conditions.

In Fig. 2a, we observe that most transactions are included in the blockchain in times close to 600 s, which is the expected theoretical time. However, there is a clear dispersion with transactions taking up to 3000 s to be included. This behavior may be due to the probabilistic nature of PoW, where some transactions may remain in the queue if not quickly selected by miners. Additionally, we observe transactions that are included very quickly, which could result from blocks being mined almost immediately after a

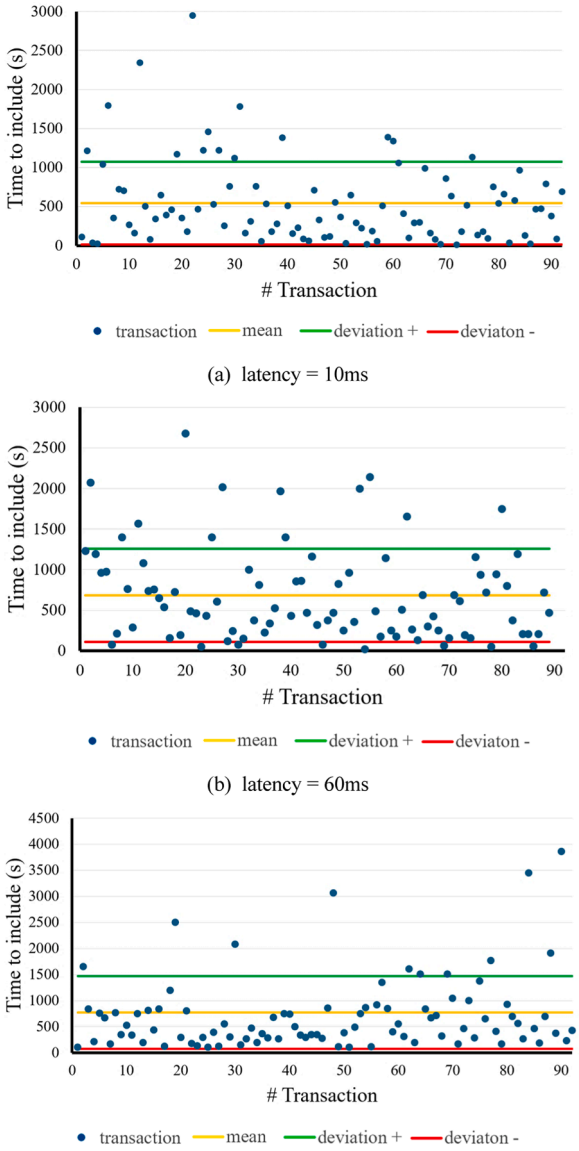
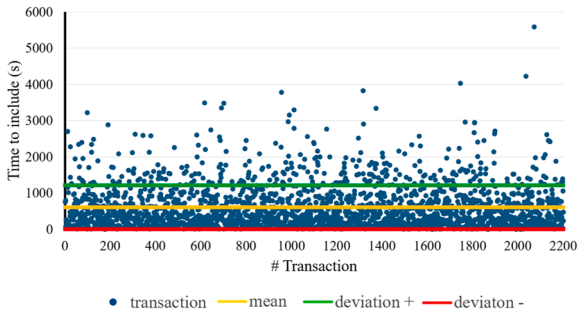


Fig. 2. Bitcoin model simulation generating one transaction per day during the cultivation process. Each subfigure represents latencies of 10 ms (a), 60 ms (b), and 120 ms (c).

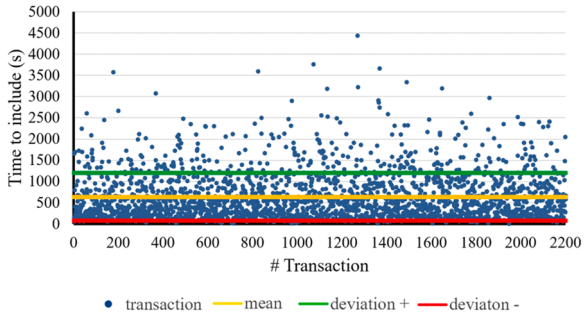
transaction's arrival. In Fig. 2b, an increase in the dispersion of transaction inclusion times is observed. The mean is slightly above 500 s, and again we see transactions taking up to 3000 s to be included. This wider dispersion and increased standard deviation reflect the moderate impact of latency on network performance. Transactions that take longer to be included could be related to the increased block propagation time, allowing more transactions to accumulate in the queue before being mined. In Fig. 2c, the impact of high latency is more evident. Although this figure shows the least dispersion of transaction inclusion times, we also observe transactions taking up to 4000 s. Notably, despite the higher latency, more transactions are included quickly compared to the 60 ms case. This behavior can be attributed to the random nature of the mining process and the fact that some transactions may benefit from blocks being mined almost immediately after their arrival. High latency makes inclusion times less predictable and more dependent on the random synchronization of transactions and block mining.

Considering the average time for a transaction to be included in the blockchain for each configuration, the results are as follows: with 10 ms latency, the average time is 592.19 s; with 60 ms, it is 652.83 s; and with 120 ms, it is 772.64 s.

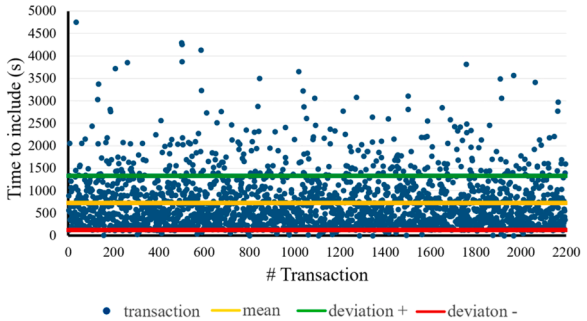
In Fig. 3a, we observe that most transactions are included in the blockchain in times close to 600 s, similar to the previous case with one transaction per day. However, the increase in the transaction generation rate has introduced slight additional dispersion, with some transactions taking longer to be included. The average inclusion time is 609.59 s, indicating a slight increase compared to the previous configuration. This increase is due to the higher network load, which increases competition among transactions to be included in the blocks. In Fig. 3b, with a latency of 60 ms, a behavior similar to that of 10 ms latency is observed, but with greater dispersion in transaction inclusion times. The mean is 707.64 s, reflecting a more pronounced impact of latency combined with the



(a) latency = 10ms



(b) latency = 60ms



(c) latency = 120ms

Fig. 3. Bitcoin model simulation generating 24 transaction per day during the cultivation process. Each subfigure represents latencies of 10 ms (a), 60 ms (b), and 120 ms (c).

higher transaction generation rate. As in the previous case, moderate latency increases block and transaction propagation times, resulting in a buildup of transactions in the mining queue. In Fig. 3c, with a latency of 120 ms, the impact of high latency and the higher transaction generation rate is even more evident. The average transaction inclusion time is 728.08 s, showing a significant increase compared to lower latencies. The dispersion of inclusion times is greater, with transactions taking much longer to be included due to the combination of high latency and increased network competition. This reinforces the observation that high latency introduces greater variability and delay in the mining and transaction propagation process.

Comparing these results with the previous case of one transaction per day, it is observed that increasing the transaction generation rate increases the average inclusion time and the dispersion in inclusion times, especially under higher latency conditions. This is due to the higher network load and competition among transactions to be included in the blocks. Latency has an amplified effect when combined with a high transaction generation rate, making inclusion times more unpredictable and prolonged.

In Fig. 4a, there is a notable increase in the dispersion of transaction inclusion times compared to the previous figures. The average inclusion time is 650.91 s, which is higher than in the previous cases with a lower transaction generation rate. This indicates that the increase in the number of daily transactions is overloading the network, even with minimal latency. In Fig. 4b, both the dispersion and the mean of inclusion times increase significantly. The average inclusion time is 760.68 s, showing an increase compared to the generation rates of 1 and 24 transactions per day. This result indicates that the combination of moderate latency and a high transaction generation rate affects network performance, resulting in much longer and more variable inclusion times. In Fig. 4c, with a latency of 120 ms, the impact of high latency is clearly evident, but with a dispersion in transaction inclusion times comparable to that of Fig. 4 (a). The average inclusion time is 827.65 s, which is significantly higher than the average inclusion times observed at 10 ms and 60 ms latencies. This behavior confirms that high latency introduces greater variability and delay in the mining and transaction propagation process.

Comparing these results with the previous cases of 1 and 24 transactions per day, it is observed that increasing the transaction generation rate to 60 per day significantly exacerbates average inclusion times and dispersion. The network shows greater stress under this load, with notable increases in average inclusion times, especially under conditions of moderate and high latency.

4.2. Ethereum

On average, a block on the Ethereum network is mined approximately every 12.5 - 15 s. Given that the size of an Ethereum block and its processing capacity allow for handling a greater number of transactions per second compared to Bitcoin, the average time for a

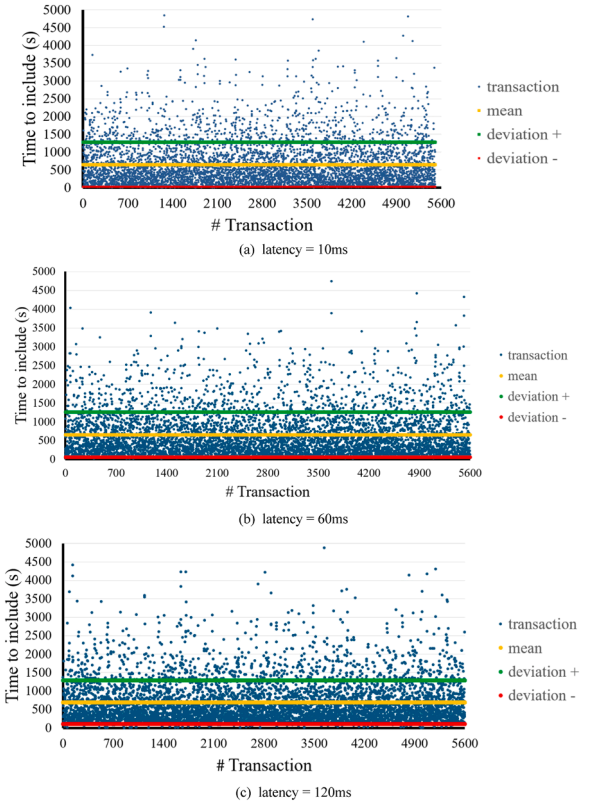
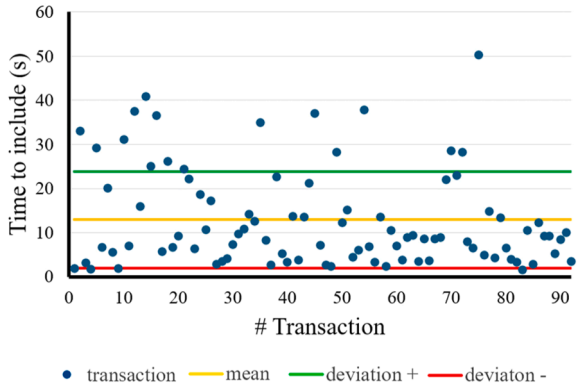


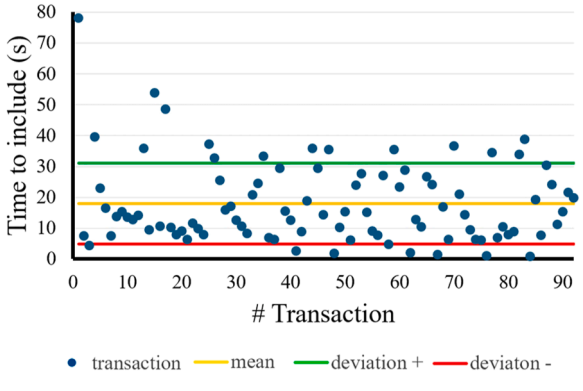
Fig. 4. Bitcoin model simulation generating 60 transaction per day during the cultivation process. Each subfigure represents latencies of 10 ms (a), 60 ms (b), and 120 ms (c).

transaction to be included is generally lower. However, network congestion can significantly influence these times. Theoretically, under optimal conditions and without congestion, a transaction on Ethereum could take around 15 s to be confirmed. However, during periods of high demand, this time can increase due to the buildup of unconfirmed transactions in the *mempool*, which is the waiting area for transactions before they are included in a block.

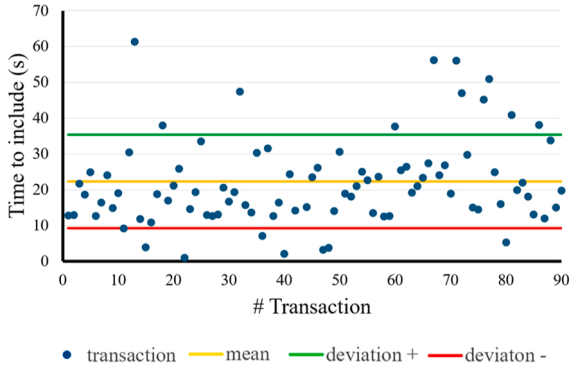
Generating one transaction per day, we observe in Fig. 5a, that most transactions are included in the blockchain in times close to 15 s, which is the expected theoretical time for Ethereum. We can see a slight dispersion in the time it takes for transactions to be included, making the average transaction inclusion time in this scenario 12.91 s. As expected, with a minimal latency of 10 ms, the impact on transaction inclusion time is almost negligible. In Fig. 5b, the average inclusion time is 17.89 s, indicating a notable increase due to



(a) latency = 10ms



(b) latency = 60ms

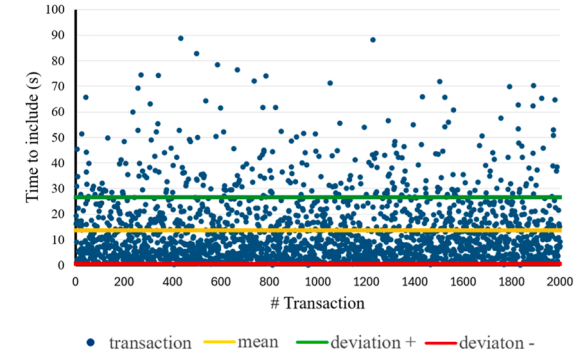


(c) latency = 120ms

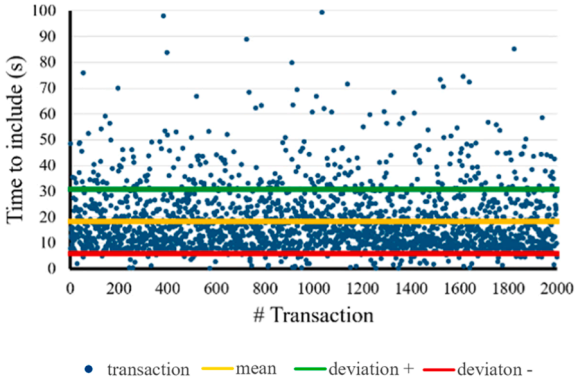
Fig. 5. Ethereum model simulation generating one transaction per day during the cultivation process. Each subfigure represents latencies of 10 ms (a), 60 ms (b), and 120 ms (c).

moderate latency. Most transactions are still included in a reasonable time close to 15 s, but more transactions take up to 70 s. Moderate latency increases block and transaction propagation times, resulting in a greater accumulation of transactions in the *mempool*. In Fig. 5c, the impact of high latency is more pronounced. Although most transactions are included in <30 s, there is greater dispersion, with inclusion times reaching up to 70 s. The average inclusion time is 22.25 s, reflecting a significant increase compared to lower latencies. Again, high latency introduces greater variability in the mining and transaction propagation process, making inclusion times more unpredictable.

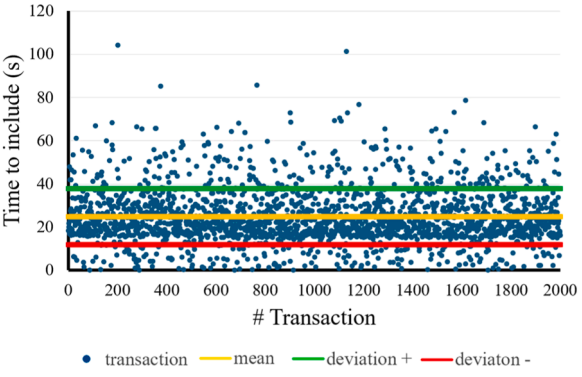
We observe in Fig. 6a that the increase in the transaction generation rate (up to 24 transaction per day during the cultivation process) introduces greater dispersion in inclusion times compared to the previous case of one transaction per day The average inclusion time is 13.58 s, indicating a slight increase due to the higher network load, which increases competition among transactions to be included in the blocks. Despite the higher load, minimal latency does not significantly impact performance. With a latency of 60 ms, an increase in transaction processing time is observed due to the combination of higher transaction generation and moderate latency



(a) latency = 10ms



(b) latency = 60ms



(c) latency = 120ms

Fig. 6. Ethereum model simulation generating 24 transaction per day during the cultivation process. Each subfigure represents latencies of 10 ms (a), 60 ms (b), and 120 ms (c).

(Fig. 6b). The average inclusion time is 18.39 s, showing a deviation from the expected theoretical time of 12.5–15 s. This increase reflects the impact of moderate latency on network efficiency under a higher load. Finally, the impact of high latency is more evident as shown in Fig. 6c. Although some transactions are still included within the expected theoretical time, most take longer than 12.5–15 s to be confirmed. The average inclusion time is 24.81 s, highlighting the significant effect of high latency on network performance. This greater variability and delay in inclusion times are direct consequences of high latency combined with a high transaction generation rate.

In Fig. 7a, with a latency of 10 ms, we observe that increasing the transaction generation rate to 60 per day causes greater dispersion in transaction inclusion times compared to the configurations of $h = 1$ and $h = 24$. The average inclusion time is 19.67 s, showing that although latency is low, the high transaction load significantly impacts performance, indicating that the network starts to show signs of congestion. With a latency of 60 ms, an even greater dispersion in inclusion times is observed in Fig. 7b. The average inclusion time is 27.45 s. This underscores how the combination of higher transaction generation and moderate latency more severely affects network performance, leading to longer and more variable inclusion times. In Fig. 7c, with a latency of 120 ms, the impact of high latency along with the high transaction generation rate becomes very evident. The average inclusion time is 35.76 s, significantly higher than in the previous configurations. The greater variability and prolonged inclusion times reflect the extreme pressure on the network, demonstrating that high latency combined with a high transaction load results in a notable degradation of performance.

4.3. DAG

In IOTA, unlike Bitcoin and Ethereum, transactions are not grouped into blocks but are added individually to the DAG. This approach theoretically allows for greater scalability and shorter inclusion times since transactions can be validated asynchronously and in parallel.

Two tip selection algorithms have been used to evaluate the performance of the model: the random selection algorithm and the weighted selection algorithm. In random selection, tips are chosen randomly from all previously unvalidated transactions in the graph.

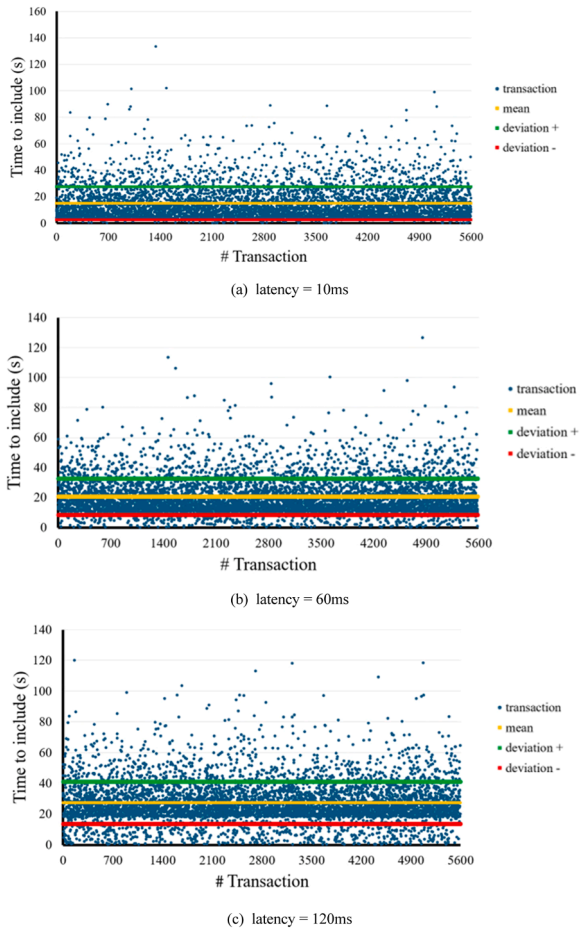


Fig. 7. Ethereum model simulation generating 60 transaction per day during the cultivation process. Each subfigure represents latencies of 10 ms (a), 60 ms (b), and 120 ms (c).

This method is simple and fast but may not be efficient in ensuring that all transactions are validated equitably. The unstructured nature of the random approach can result in variable inclusion times, as selecting a tip to validate may result in picking an already validated one, necessitating another search for a valid tip, thus increasing inclusion time.

The weighted selection algorithm chooses tips based on the cumulative weight of the transactions. The weight can be influenced by factors such as the amount of work done to approve them or the number of times they have been referenced by other transactions. This method should provide a more structured and faster validation of transactions, especially those with higher cumulative weight. A key parameter in weighted selection is the degree of randomness in the tip selection, controlled by α (alpha). The behavior of the model has been evaluated with two α values: 0.4 and 0.8. A lower α value (0.4) introduces more randomness into the selection, reducing efficiency but favoring decentralization. A higher α value (0.8) makes tip selection more deterministic, favoring transactions with higher cumulative weight and theoretically reducing inclusion times by focusing on transactions that already have greater support in the network.

For random selection, we expect to see more dispersed inclusion times due to the unstructured nature of the method. The variability in inclusion times will be higher, especially under conditions of high latency and transaction load.In the case of weighted selection with $\alpha = 0.4$, a balance between randomness and efficiency is anticipated, with moderately better inclusion times than the random approach but still with some dispersion. With $\alpha = 0.8$, weighted selection should show more consistent and faster inclusion times, as transactions with higher cumulative weight will be prioritized, reducing dispersion, and improving the overall efficiency of the transaction inclusion process in the Tangle.

We present the results obtained for each transaction generation rate, evaluating the models with rates of 1, 24, and 60 transactions generated per day. Within each figure, there will be subfigures representing the different algorithms and α values evaluated. These subfigures will allow for a comparison of how transaction inclusion times in the IOTA Tangle vary under different tip selection conditions and network load.

For one transaction per day, we observe in Fig. 8a, with $\alpha = 0.4$ and the weighted algorithm, that transaction inclusion times are relatively consistent, with minimal dispersion around the mean. The average inclusion time is low, indicating that the network handles this configuration well with low latency and a low transaction generation rate. Transactions are included efficiently, with few

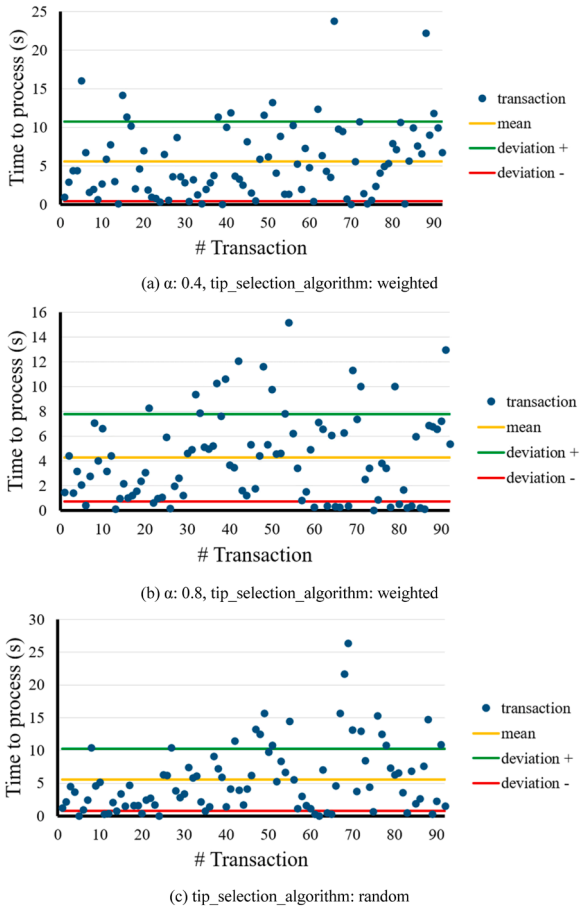
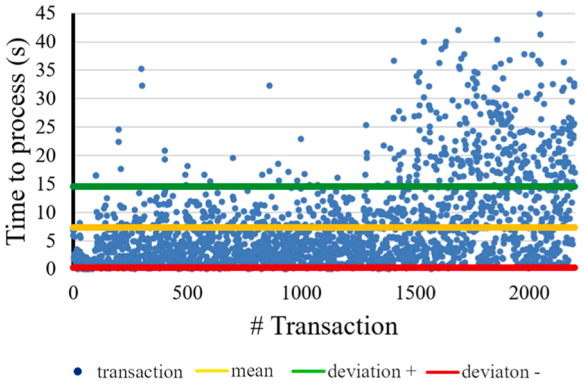


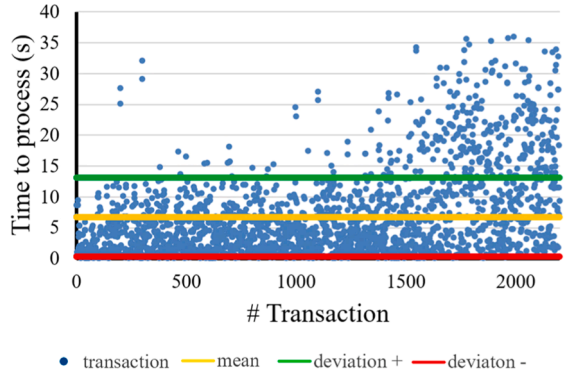
Fig. 8. DAG model simulation for one-day transaction generation and 10 ms latency. Subfigures (a) and (b) represent the simulation for the 'weighted' tip selection algorithm with α set to 0.4 (a) and 0.8 (b). Subfigure (c) represents the simulation for the 'random' tip selection algorithm.

deviations. In Fig. 8b, with $\alpha = 0.8$ and the weighted algorithm, inclusion times show a slight improvement compared to $\alpha = 0.4$. The more deterministic tip selection, favoring transactions with higher cumulative weight, results in even more consistent and faster inclusion times. The mean remains low and dispersion is minimal, confirming that a higher α value improves efficiency in this context. However, in Fig. 8c, using the random selection algorithm, we notice greater dispersion in transaction inclusion times. Although the average time is still low, variability is higher compared to the weighted configurations. This result was expected due to the unstructured nature of random tip selection, which introduces more variability in inclusion times.

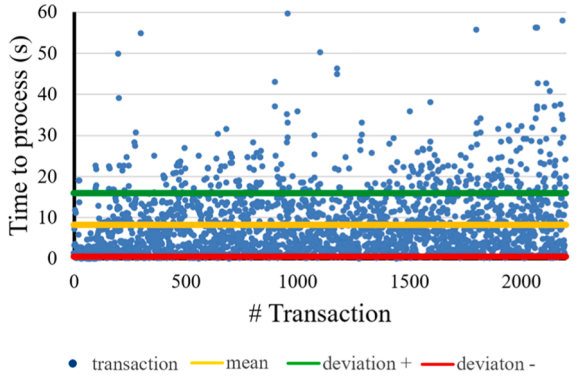
Comparing these subfigures, we can conclude that weighted selection with a higher α value (0.8) provides the best results in terms of consistency and speed in transaction inclusion. Random selection, though simple, shows greater dispersion and variability in inclusion times, which could be less efficient under conditions of higher load or latency. In Figs. 9 and Fig. 10, we can observe how the increase in the transaction generation rate affects the performance of the different tip selection algorithms.



(a) α: 0.4, tip_selection_algorithm: weighted

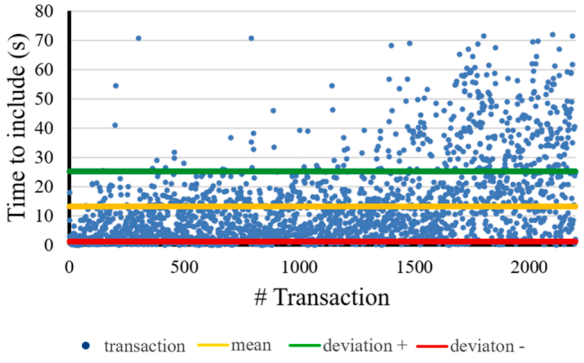


(b) α: 0.8, tip_selection_algorithm: weighted

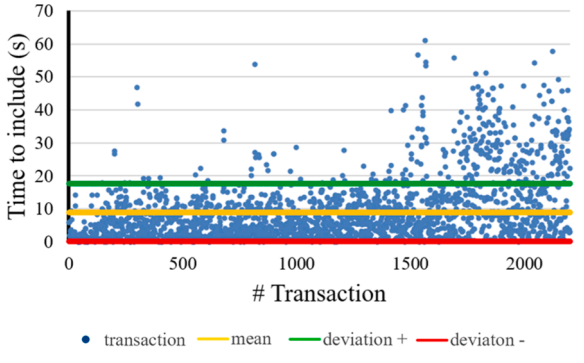


(c) tip_selection_algorithm: random

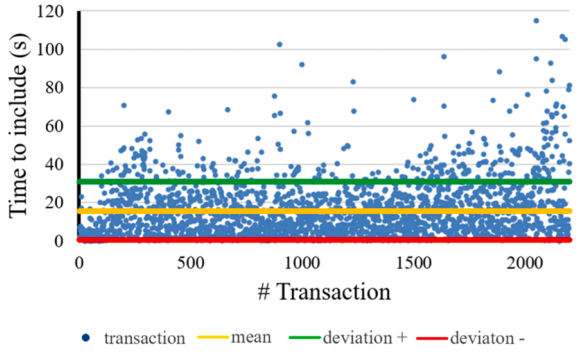
Fig. 9. DAG model simulation for generating 24 transactions per day and 10 ms latency. Subfigures (a) and (b) represent the simulation for the 'weighted' tip selection algorithm with α set to 0.4 (a) and 0.8 (b). Subfigure (c) represents the simulation for the 'random' tip selection algorithm.



(a) α: 0.4, tip_selection_algorithm: weighted



(b) α: 0.8, tip_selection_algorithm: weighted



(c) tip_selection_algorithm: random

Fig. 10. DAG model simulation for generating 24 transactions per day and 60 ms latency. Subfigures (a) and (b) represent the simulation for the 'weighted' tip selection algorithm with α set to 0.4 (a) and 0.8 (b). Subfigure (c) represents the simulation for the 'random' tip selection algorithm.

Generating 24 transactions per day, a greater dispersion in transaction inclusion times is observed compared to the configuration of 1 transaction per day as depicted in Fig. 9a, with $\alpha = 0.4$ and the weighted algorithm. The average inclusion time has increased, reflecting the higher network load. However, most transactions are included within an acceptable range, albeit with higher variability. Fig. 9b, with $\alpha = 0.8$ and the weighted algorithm, inclusion times show less dispersion compared to $\alpha = 0.4$, although variability is still higher than in the case of one transaction per day. The average inclusion time is more consistent and faster, confirming that a higher α value continues to provide better results under a higher transaction load. Then, in Fig. 9c, using the random selection algorithm, the dispersion in inclusion times is significantly greater than in the weighted configurations. Variability is high, and some transactions take considerably longer to be included. This behavior was expected due to the unstructured nature of random tip selection.

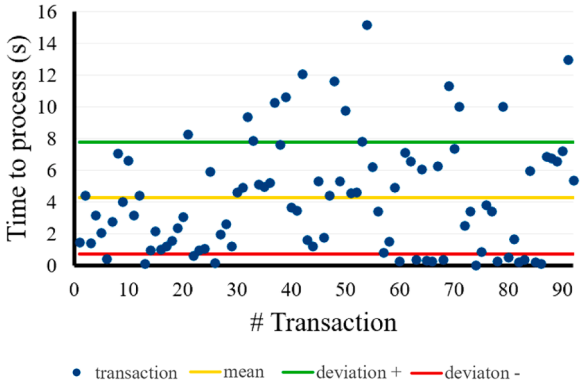
Now we will evaluate whether, by maintaining the generation rate of 24 transactions per day and increasing latencies, the weighted algorithm with $\alpha = 0.8$ continues to provide better results compared to $\alpha = 0.4$ and the random selection algorithm. This evaluation will determine whether the $\alpha = 0.8$ configuration maintains its efficiency under higher latency conditions and can effectively handle the transaction load.

In Fig. 10a, with $\alpha = 0.4$ and the weighted algorithm, a notable increase in the dispersion of transaction inclusion times is observed

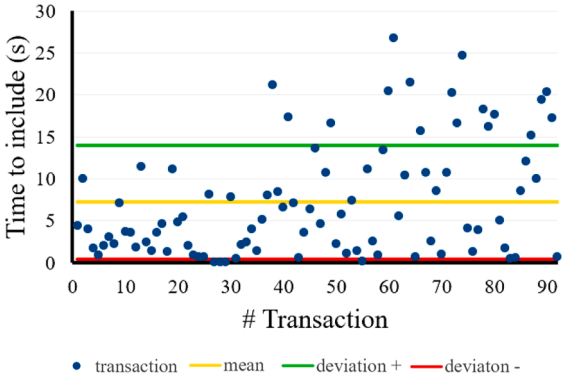
compared to previous configurations. The average inclusion time is higher, and there is significant variability in inclusion times, reflecting the impact of higher latency on the network. In Fig. 10b, with $\alpha = 0.8$ and the weighted algorithm, inclusion times show less dispersion compared to $\alpha = 0.4$. However, variability is still higher than in lower latency configurations. The average inclusion time remains more consistent and faster than with $\alpha = 0.4$, confirming that a higher α value provides better results even under higher latency conditions. Fig. 10c, using the random selection algorithm, shows that the dispersion in inclusion times is considerably greater. Variability is high, and some transactions take significantly longer to be included. This result was expected due to the unstructured nature of random tip selection, which introduces more variability in inclusion times under higher latency conditions.

Based on these results, we will focus the analysis on the remaining configurations using $\alpha = 0.8$ with the weighted algorithm, as it provides the best results. This will allow us to more effectively compare the IOTA, Bitcoin, and Ethereum models under different load and latency conditions.

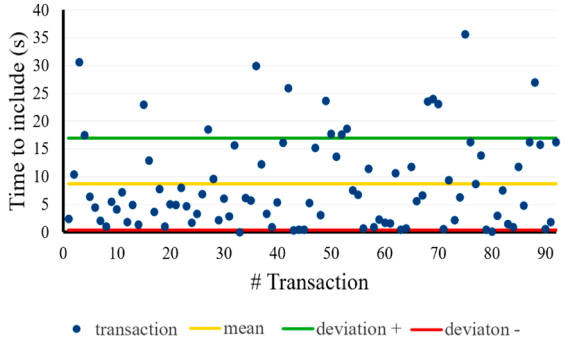
In Fig. 11a, with a latency of 10 ms, transaction inclusion times are consistent, and dispersion is minimal, with most transactions



(a) latency: 10ms



(b) latency: 60ms



(c) latency: 120ms

Fig. 11. DAG model simulation for generating 1 transaction per day and tip selection algorithm: weighted with α: 0.8. Subfigures (a), (b), and (c) show the simulation for different network latencies: 10 ms (a), 60 ms (b), and 120 ms (c).

included in <12 s. Increasing the latency to 60 ms, as shown in Fig. 11b, results in an increase in the dispersion of inclusion times, and the average time rises to approximately 18 s. This increase reflects the impact of higher latency on the network, though still within acceptable limits for an efficient DAG system. With a latency of 120 ms, represented in Fig. 11c, the average inclusion time increases to around 22 s, and the dispersion in inclusion times becomes significantly greater.

In Fig. 12a, with a latency of 10 ms, we observe greater dispersion in transaction inclusion times compared to scenarios with lower transaction generation rates. Although most transactions are included in <40 s, there is a notable increase in dispersion, indicating that the network is beginning to show signs of overload under this high transaction generation configuration. In Fig. 12b, with a latency of 60 ms, the dispersion in inclusion times increases even further. The average inclusion time rises, and variability is significantly greater, with some transactions taking up to 100 s to be included. This increase reflects the impact of moderate latency combined with a high transaction generation rate, testing the network's capacity to handle the load. Finally, in Fig. 12c, with a latency of 120 ms, the impact is even more pronounced. The dispersion of inclusion times is considerably high, with transactions taking up to 160 s to be included. This result underscores the difficulty for the network to maintain efficiency under conditions of high latency and transaction load.

The choice of tip selection algorithm in DAG also plays a crucial role. We quickly discarded the tip selection algorithm weighted with $\alpha = 0.4$ and the random algorithm due to their inferior results. With $\alpha = 0.4$, the dispersion of inclusion times is greater and less consistent compared to $\alpha = 0.8$. The random algorithm, while simple, shows greater variability in inclusion times due to its unstructured nature. Therefore, the weighted algorithm with $\alpha = 0.8$ stands out as the most efficient configuration for the DAG, providing the lowest and most consistent average inclusion times.

5. Discussion and limitations

Table VII summarizes the overall performance of Bitcoin, Ethereum, and DAG-based technologies under various transaction loads and network latencies. Bitcoin demonstrates high sensitivity to increased latency, with average inclusion times rising substantially as latency increases from 10 ms to 120 ms. In contrast, Ethereum maintains lower inclusion times across the same latency range, indicating greater resilience. Additionally, the DAG solution, particularly with a weighted tip selection algorithm ($\alpha = 0.8$), delivers the lowest inclusion times across all configurations, underscoring its potential for real-time or near-real-time applications.

Bitcoin provides robust security and immutability via its Proof-of-Work mechanism, which is advantageous for applications that require high trust. However, its scalability is limited, and performance degrades markedly under high transaction loads and elevated latencies. Although optimizations such as Segregated Witness and the Lightning Network can mitigate these issues, they introduce additional complexity, contributing to its limited adoption in smart agriculture, where slower transaction rates and potentially higher fees under congestion pose significant challenges.

Ethereum, with its capability to execute smart contracts, facilitates the automation of agricultural processes such as payments, insurance, and crop monitoring. While it shows moderate resilience to latency increases, its performance still declines under high transaction volumes. The ongoing transition to Proof-of-Stake (Ethereum 2.0), along with the implementation of sharding and layer-two scaling solutions, is expected to enhance throughput and reduce fees, thereby potentially expanding its applicability in IoT-driven agricultural environments.

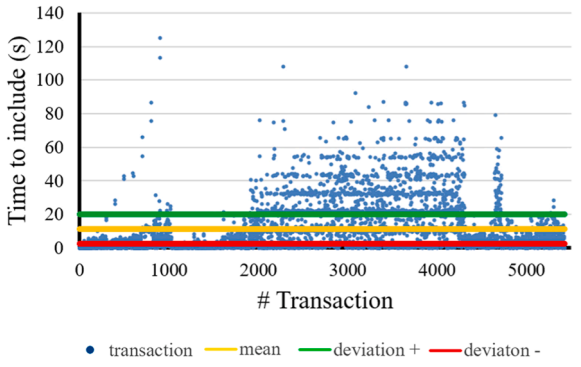
DAG-based technologies, employing an asynchronous and parallel validation structure, achieve significantly lower inclusion times, which is particularly beneficial for real-time data collection where continuous sensor data is generated. Nevertheless, the complexity of data retrieval and the absence of mature standards pose challenges for traceability and may impede widespread adoption. Further research is needed to address issues related to network topology design, consensus stability, and interoperability with existing blockchain systems.

Despite these results, several limitations and assumptions must be acknowledged. In terms of simulation scope and metrics, our analysis focuses primarily on transaction inclusion times, as measured by BlockSim for Bitcoin and Ethereum and DAGSim for DAG-based systems. This focus excludes other efficiency metrics such as energy consumption, storage overhead, and detailed security trade-offs. Moreover, the simulations are conducted under low-to-moderate load scenarios, specifically, 1, 24, and 60 transactions per day, which are significantly lower than the maximum theoretical throughput of public blockchains, thereby not fully exposing potential bottlenecks under very high load conditions. Throughput is inferred indirectly via inclusion times, rather than by simulating large-scale stress conditions where it would be the primary performance bottleneck.

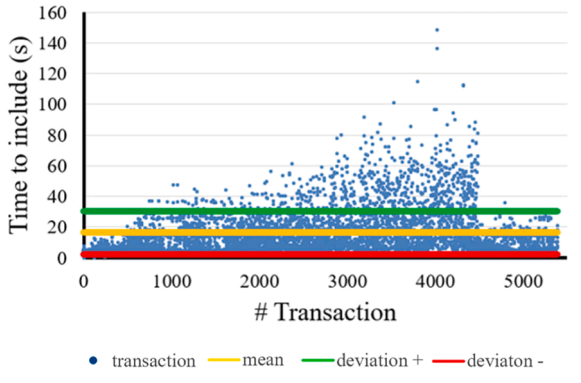
With regard to consensus and security, the simulations assume honest or semi-honest behavior among miners and validators, thereby not accounting for adversarial strategies like selfish mining or double-spending, which could significantly impact performance and security in real-world scenarios. Additionally, while the study examines Proof-of-Work for Bitcoin and Ethereum alongside DAG-based protocols, it does not incorporate newer consensus variants, such as Ethereum's Proof-of-Stake or other DAG variants, that might influence both performance and security assumptions.

Network and hardware constraints are also simplified in the simulations. A uniform or fixed latency model is employed (with delays of 10 ms, 60 ms, and 120 ms), which does not capture the variability in propagation times, network topologies, or bandwidth constraints observed in real-world networks. Furthermore, the simulators assume homogeneous hardware across all nodes, an assumption that does not reflect the wide variability in computational resources found in IoT devices used in agricultural settings, from low-power sensors to high-throughput data centers.

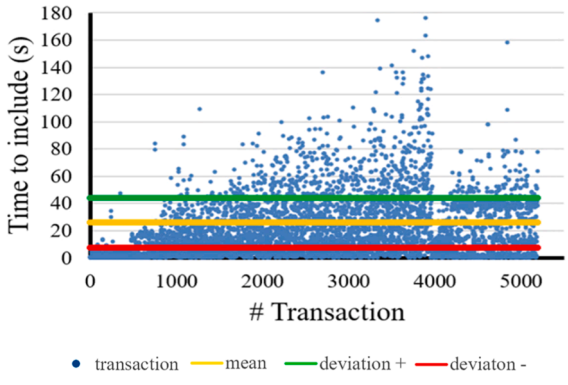
Data and traceability challenges arise from the use of input data based on a single 76-day crop cycle. This approach may not generalize to different crop types, varying environmental conditions, or continuous, year-round operations. Although DAG-based systems demonstrate superior parallel validation performance, the inherent complexity in searching and indexing data could adversely affect their utility in scenarios requiring rapid query responses for agricultural traceability.



(a) latency: 10ms



(a) latency: 60ms



(a) latency: 120ms

Fig. 12. DAG model simulation for generating 60 transaction per day and tip selection algorithm: weighted with α: 0.8. Subfigures (a), (b), and (c) show the simulation for different network latencies: 10 ms (a), 60 ms (b), and 120 ms (c).

Future research should address these limitations by evaluating energy efficiency and environmental impact, incorporating more comprehensive security models that simulate adversarial behaviors and potential collusion, and testing scalability under transaction loads more representative of real-world IoT scenarios. Additionally, integrating real hardware and protocol stacks, through pilot implementations or hardware-in-the-loop simulations, could yield more accurate insights into performance, cost implications, and adoption barriers.

Despite these constraints, each technology offers distinct advantages and faces specific challenges in the context of smart agriculture. Bitcoin technology provides great robustness and security, making it a solid option for applications where security and immutability are paramount. However, its limited processing capacity and sensitivity to latency and transaction load present significant challenges, which is reflected in the current state of the art where few solutions implement Bitcoin for smart agriculture. To enhance its performance under high latency and load conditions, Bitcoin's Proof-of-Work protocols could be optimized through

Table VII
Results summary.

Transactions per day	Latency	Bitcoin	Ethereum	DAG		
				$\alpha = 0.8$	$\alpha = 0.4$	random
1	10ms	542,23	12,91	4,27	5,58	5,96
	60ms	682,74	17,89	7,25	8,06	7,34
	120ms	702,64	22,25	8,67	10,98	12,90
24	10ms	609,59	13,58	6,67	7,32	8,21
	60ms	707,93	18,39	8,95	13,20	15,59
	120ms	728,08	24,81	11,32	17,53	20,43
60	10ms	650,91	19,67	11,32	12,23	18,84
	60ms	760,68	27,45	14,41	16,07	22,06
	120ms	827,65	35,76	18,97	25,91	26,05

improvements such as Segregated Witness (SegWit), which reduces transaction size, and the adoption of the Lightning Network for faster, more efficient transactions. Such modifications could leverage Bitcoin's inherent security benefits for applications like agricultural product traceability and origin certification.

Ethereum, with its better resilience to latency and more controlled inclusion times, offers a more adaptable solution for smart agriculture. Its smart contract functionality enables the automation of various agricultural processes, including payments, insurance, and crop monitoring, and has led to its widespread adoption in the field. Future advances, such as the transition to Proof-of-Stake with Ethereum 2.0, the implementation of sharding, and scaling techniques like Optimistic Rollups and zk-Rollups, are expected to further enhance Ethereum's scalability and transaction capacity, thereby improving its efficiency under higher loads.

On the other hand, DAG-based technology presents significant advantages in terms of high processing capacity and low inclusion times, owing to its asynchronous and parallel validation structure. These features are particularly beneficial for real-time data collection in environments where rapid processing is critical. Nevertheless, the DAG model encounters limitations that are not as pronounced in Bitcoin or Ethereum. For instance, data retrieval within a DAG is more complex than in linear blockchains, potentially impeding quick query responses necessary for effective traceability. The lack of clear standards further complicates its integration into smart agriculture applications.

6. Conclusion

This study has presented a comprehensive comparison of blockchain and DAG technologies for smart agriculture traceability, focusing on efficiency and latency under varying network conditions. In our study, efficiency is defined as the ability to process transactions quickly under different network loads (i.e., measured primarily by latency). We acknowledge that efficiency can also encompass metrics such as throughput, energy consumption, and storage overhead, which are not covered in this work and should be the subject of future investigations. Our results demonstrate that DAG, particularly when utilizing a weighted tip selection algorithm with a high α value, consistently outperforms traditional blockchain technologies in terms of transaction inclusion times and scalability. Ethereum offers a balanced compromise with faster inclusion times than Bitcoin, though both are significantly impacted by latency and transaction load. The findings underscore the potential of DAG as a viable solution for applications requiring high-frequency data validation and low-latency processing, such as IoT-enabled smart agriculture. The open-source implementation and emulated tests contribute to reproducibility and pave the way for further research. Future work should explore real-world deployments to validate these findings and address challenges related to searching data, interoperability, regulatory frameworks, and network adoption.

Acknowledgements

This work was supported by the grant PID2020–114410RB-I00 MCIN/AEI/10.13039/501100011033, formed part of the AGROALNEXT program, and was supported by MICIU with funding from European Union NextGenerationEU (PRTR-C17.I1) and by Fundación Séneca with funding from Comunidad Autónoma Región de Murcia (CARM).

Data availability

Data will be made available on request.

References

[1] "Bitcoin." [Online]. Available: https://bitcoin.org/es/. Accessed May 2, 2025.
[2] "Ethereum." [Online]. Available: https://ethereum.org/es/. Accessed May 2, 2025.
[3] I. Tasic, M.D. Cano, An orchestrated IoT-based blockchain system to foster innovation in agritech, IET Collabor. Intell. Manuf. 6 (2) (2024), https://doi.org/10.1049/cim2.12109.
[4] I. Tasic, R.R. Benaissa, P.A. Gómez, J.A. Fernandez, and M.-D. Cano, "Revisiting traceability of vegetable fresh-products in the EU: why blockchain does (not) work?", Int. Technol. Letters, 2025. in review.
[5] "IOTA." [Online]. Available: https://www.iota.org/. Accessed May 2, 2025.

[6] G. Lv, C. Song, P. Xu, Z. Qi, H. Song, and Y. Liu, "Blockchain-based traceability for agricultural products: a systematic literature review," 2023, Multidisciplinary Digital Publishing Institute (MDPI). doi: 10.3390/agriculture13091757.
[7] S. Yele, R. Litoriya, Blockchain-based secure dining: enhancing safety, transparency, and traceability in food consumption environment, Blockchain: Res. Appl. 5 (2) (2024), https://doi.org/10.1016/j.bcra.2023.100187.
[8] "Hyperledger F." [Online]. Available: https://www.hyperledger.org. Accessed May 2, 2025.
[9] Z. Tao, J. Chao, The impact of a blockchain-based food traceability system on the online purchase intention of organic agricultural products, in: Innovative Food Science and Emerging Technologies, 92, 2024, https://doi.org/10.1016/j.ifset.2024.103598.
[10] H. Patel, B. Shrimali, AgriOnBlock: secured data harvesting for agriculture sector using blockchain technology, ICT Express 9 (2) (2023) 150–159, https://doi.org/10.1016/j.icte.2021.07.003.
[11] W. Ren, X. Wan, P. Gan, A double-blockchain solution for agricultural sampled data security in Internet of Things network, Future Generat. Comput. Syst. 117 (2021) 453–461, https://doi.org/10.1016/j.future.2020.12.007.
[12] "IPFS." [Online]. Available: https://ipfs.tech. Accessed May 2, 2025.
[13] Y. Li, C. Tan, W.H. Ip, C.H. Wu, Dynamic blockchain adoption for freshness-keeping in the fresh agricultural product supply chain, Expert Syst. Appl. 217 (2023), https://doi.org/10.1016/j.eswa.2022.119494.
[14] F. Zheng, X. Zhou, Sustainable model of agricultural product logistics integration based on intelligent blockchain technology, in: Sustain. Energy Technol. Assessm., 57, 2023, https://doi.org/10.1016/j.seta.2023.103258.
[15] K.U. Rehman, S. Andleeb, M. Ashfaq, N. Akram, M.W. Akram, Blockchain-enabled smart agriculture: enhancing data-driven decision making and ensuring food security, J Clean Prod 427 (2023), https://doi.org/10.1016/j.jclepro.2023.138900.
[16] A.S.M.T. Hasan, S. Sabah, A. Daria, R.U. Haque, A peer-to-peer blockchain-based architecture for trusted and reliable agricultural product traceability, Decision Analyt. J. 9 (2023), https://doi.org/10.1016/j.dajour.2023.100363.
[17] L. Cui, S. Yang, Z. Chen, Y. Pan, M. Xu, K. Xu, An efficient and compacted DAG-based blockchain protocol for industrial internet of things, IEEE Trans. Ind. Inform 16 (6) (2020) 4134–4145, https://doi.org/10.1109/TII.2019.2931157.
[18] A.S.M.T. Hasan, S. Sabah, A. Daria, R.U. Haque, A peer-to-peer blockchain-based architecture for trusted and reliable agricultural product traceability, Decision Analyt. J. 9 (2023), https://doi.org/10.1016/j.dajour.2023.100363.
[19] L. Li, T. Li, Traceability model based on improved witness mechanism, CAAI Trans. Intell. Technol. 7 (3) (2022) 331–339, https://doi.org/10.1049/cit2.12124.
[20] H. Watanabe, et al., Enhancing blockchain traceability with DAG-based tokens, in: Proceedings - 2019 2nd IEEE International Conference on Blockchain, Institute of Electrical and Electronics Engineers Inc., 2019, pp. 220–227, https://doi.org/10.1109/Blockchain.2019.00036. Blockchain 2019.
[21] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, Bitcoin.org (2008). Available: https://bitcoin.org/bitcoin.pdf. Accessed May 2, 2025.
[22] A. Dorri, M. Steger, S.S. Kanhere, R. Jurdak, BlockChain: a distributed solution to automotive security and privacy, IEEE Commun. Mag. 55 (12) (2017) 119–125, https://doi.org/10.1109/MCOM.2017.1700879.
[23] X. Wang et al., "Survey on blockchain for Internet of Things," 2019, Elsevier B.V. doi: 10.1016/j.comcom.2019.01.006.
[24] H. Liu, D. Han, D. Li, Fabric-iot: a blockchain-based access control system in IoT, IEEE Access 8 (2020) 18207–18218, https://doi.org/10.1109/ACCESS.2020.2968492.
[25] G. Wood, Ethereum: a secure decentralised generalised transaction ledger, Ethereum Project (2014). Available: https://ethereum.github.io/yelowpaper/paper.pdf. Accessed May 2, 2025.
[26] V. Buterin et al., "Combining GHOST and Casper," 2020, [Online]. Available: http://arxiv.org/abs/2003.03052.
[27] W.F. Silvano, R. Marcelino, Iota Tangle: a cryptocurrency to communicate Internet-of-things data, Future Generat. Comput. Syst. 112 (2020) 307–319, https://doi.org/10.1016/j.future.2020.05.047.
[28] M. Conti, G. Kumar, P. Nerurkar, R. Saha, and L. Vigneri, "A survey on security challenges and solutions in the IOTA," 2022, Academic Press. doi: 10.1016/j.jnca.2022.103383.
[29] F.M. Benčić, I.P. Žarko, Distributed ledger technology: blockchain compared to directed acyclic graph, in: Proceedings - International Conference on Distributed Computing Systems, Institute of Electrical and Electronics Engineers Inc., 2018, pp. 1569–1570, https://doi.org/10.1109/ICDCS.2018.00171.
[30] M. Alharby, A. van Moorsel, BlockSim: an extensible simulation tool for blockchain systems, Front. Blockchain 3 (2020), https://doi.org/10.3389/fbloc.2020.00028.
[31] M. Zander, T. Waite, D. Harz, DAGsim: simulation of DAG-based distributed ledger protocols, ACM SIGMETRICS Perform. Eval. Rev. 46 (3) (2019) 118–121, https://doi.org/10.1145/3308897.3308951.
[32] "NCBI", [Online]. Available: https://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi. Accessed May 2, 2025.