Blockchain self-update smart contract for supply chain traceability with data validation

CRISTIAN VALENCIA-PAYAN[*], *Department of Telematics, Universidad del Cauca, Popayán, 190003, Colombia.*

DAVID GRIOL[**], *Department of Software Engineering, University of Granada, Granada, 18071, Spain.*

JUAN CARLOS CORRALES[†], *Department of Telematics, Universidad del Cauca, Popayán, 190003, Colombia.*

Abstract

A sustainable supply chain management strategy reduces risks and meets environmental, economic and social objectives by integrating environmental and financial practices. In an ever-changing environment, supply chains have become vulnerable at many levels. In a global supply chain, carefully tracing a product is of great importance to avoid future problems. This paper describes a self-updating smart contract, which includes data validation, for tracing global supply chains using blockchains. Our proposal uses a machine learning model to detect anomalies on traceable data, which helps supply chain operators detect anomalous behavior at any point in the chain in real time. Hyperledger Caliper has been used to evaluate our proposal, and obtained a combined average throughput of 184 transactions per second and an average latency of 0.41 seconds, ensuring that our proposal does not negatively impact supply chain processes while improving supply chain management through data anomaly detection.

Keywords: Blockchain, smart contracts, sustainable supply chain management, data validation, traceability, Industry 4.0.

1 Introduction

Sustainable supply chain management (SSCM) has been proposed to meet supply chain processes' environmental, economic and social goals [14, 28, 32]. However, there are still critical challenges in the sustainability of supply chains in a world becoming more of a global economy, such as waste management, employees state, child labor, etc. With the rise of Industry 4.0, smart manufacturing must also be considered a challenge for SSCM due to the supply chain further segmentation [5, 14]. In addition, the ever-changing environment has made the supply chains vulnerable at many levels, generating a more rigorous traceability process to avoid future problems.

[*] E-mail: chpayan@unicauca.edu.co

[**] E-mail: dgriol@ugr.es

[†] E-mail: jcorral@unicauca.edu.co

Vol. 00, No. 00, © The Author(s) 2024. Published by Oxford University Press.

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact reprints@oup.com for reprints and translation rights for reprints. All other permissions can be obtained through our RightsLink service via the Permissions link on the article page on our site-for further information please contact journals.permissions@oup.com.

https://doi.org/10.1093/jigpal/jzae047

Downloaded from https://academic.oup.com/jigpal/advance-article/doi/10.1093/jigpal/jzae047/7670719 by Universidad de Granada - Biblioteca user on 21 June 2024

Carefully tracing a product has become necessary in a global supply chain (GSC) [10, 30]. Businesses must be agile, with high resilience and risk mitigation, to survive in the current and complex GSC environment [5]. Blockchain technology improves traceability, quality control, safety and reliability [12, 21]. Traceability is the ability to identify and track the history, distribution, location and application of a final product's products, parts and materials to ensure reliability [18, 25]. Traceability is a crucial element in SSCM. A goal of SSCM is to improve efficiency by coordinating the efforts of the various entities in the supply chain. It can result in a company achieving a competitive advantage over its rivals and enhancing the quality of its products or services while reducing its environmental impact.

Blockchain technology also improves trust, accessibility, transparency and reliability of traceability processes using smart contracts deployed on the not-modified network [26]. To achieve these benefits, before its deployment, the parties involved must agree on the contract. In addition, Blockchain technology can present new threats to the traceability processes like phishing, routing and sibyl attacks, as well as data integrity and reliability [11]. For this reason, the Blockchain operators must address data integrity and reliability before the network block creation. It can also assist in processes such as those proposed by [8, 29], in which data from the Blockchain network is used as input into soft computing algorithms for decision making, user segmentation, etc. trust and information sharing, this could benefit the GSC performance [18] by improving trust on data and faster information transmission between chain members.

The main objective of this paper is to develop a self-updated Blockchain traceability smart contract with data validation. Smart contracts perform valid transactions when accomplishing a set of predefined conditions. Our proposal seeks to provide the smart contract with data analysis techniques to determine when the transaction data (such as storage temperature, ingredients, origin, etc.) does not correspond with the historical information available on the network. A novel use of smart contracts is to monitor data anomalies on a traceability scheme using Blockchain technology. The focus has not been on the data itself but on the smart contract and the Blockchain transactions. With this type of analysis, it is possible to identify problems as soon as they arise, improve decision-making processes and automate management recommendations and price penalties.

The remainder of the paper is as follows. Section 2 presents related work to data storage in Blockchain. Section 3 introduces the main concepts related to Blockchain-based GSC traceability. Sections 4 and 5 expose our proposal for defining smart contracts for GSC traceability, their implementation and the results of the evaluation using data from a pilot deployment of our smart contract on a Blockchain Traceability scheme. Finally, Section 6 presents the conclusions and future research lines.

2 Related work

A systematic mapping was performed, using scientific research databases based on the methodology proposed in [27]. With this we were able to find relevant contributions focused on our interest topics in the Blockchain. We selected the following research.

An ontology-based detection framework for anomalous transactions on the Blockchain in business processes is proposed in [24]. The proposed framework is evaluated on transaction logs of a simulated Blockchain network. However, the authors do not consider that manipulated data could be included in a typical transaction that can impact the trust of the Blockchain members. Tian *et al.* highlight the importance of information sharing in an SC and how the help of Blockchain technology solves the problems associated with information sharing [4]. However, the authors do

Downloaded from https://academic.oup.com/jigpal/advance-article/doi/10.1093/jigpal/jzae047/7670719 by Universidad de Granada - Biblioteca user on 21 June 2024

Downloaded from https://academic.oup.com/jigpal/advance-article/doi/10.1093/jigpal/jzae047/7670719 by Universidad de Granada - Biblioteca user on 21 June 2024

not consider the quality or reliability of the shared information, so any tampered data can be shared on the proposed sharing platform without being noticed.

A subgraph-based anomaly detection method able to run on GPUs is proposed in [23], The method was developed to run on GPUs trying to keep a consistently high number of transactions per second to reduce the impact on the Blockchain network. However, the proposed method only works with fraudulent transactions or stolen secret keys leaving aside tampered data that can be present at each transaction. Blockchain has also been presented to help detect anomalies in IoT devices and electricity consumption, as seen in [16, 22]. Also, an abnormal smart contract detection is proposed in [17] using Heterogeneous Graph Transformer Networks focused on financial fraud. Anomaly detection, however, is performed by tools outside the network, which leaves the door open to security problems due to the fact that the tools in charge of anomaly detection can be modified at any time without the operators noticing it on time.

Mezquita *et al.* propose a Blockchain architecture for logistic activities control using smart contracts to remove intermediaries in order to increase speed, and security, and to automate processes [20]. The possibility to apply penalties to parties that do not comply with the terms of the proposed platform is considered. Nevertheless, the authors do not consider data validation as part of smart contract development. Luo *et al.* review common scenarios where false information at any stage of the supply chain may represent a significant challenge to authorities [25]. They explain how Blockchain and smart contracts improve supply chain traceability transparency, security and data immutability. In [19] the authors explore Blockchain technology's applicability in logistics and supply chains using smart contracts in Ethereum, comparing their proposed solution with the ones currently available. Nevertheless, the authors do only consider data security by the intrinsic Blockchain characteristics, leaving aside problems that can occur with the data before being approved on a Blockchain transaction.

Although the previously related proposals have dealt with the different sources of information and how they are shared among supply chain members, all of them consider the assumption that the data associated with transactions do not present any problem. However, to have total reliability in supply chains supported by Blockchain technology, it is necessary to consider the possibility that the data sent in a transaction could be manipulated or incorrectly perceived, such as semantic or syntax errors. Our proposed smart contract has been developed to be able to detect these types of anomalies and to perform autonomous updates over time to be able to keep detecting new types of semantic anomalies that may arise in the future. This will help our developed smart contract to adapt to multiple traceability applications across all the supply chain processes on different types of products by using the initial agreed condition by the chain members at each stage.

3 Blockchain-based GSC traceability

A GSC is a supply chain that extends beyond a single national boundary [14]. Products and services are distributed to maximize profits and minimize waste for all active companies in the supply chain. Most of the members of this chain are transnational companies. Information flows depend on multiple factors. Identifying when a good or service has undergone a modification that could seriously affect the end consumer can be difficult and costly. In these cases, Blockchain technology can speed up verifying the conditions of a good or service. In the GSC, reputable companies are highly scrutinized for their environmental, social and economic sustainability performance. Figure 1 shows the general GSC. At the left, we got the GSC Suppliers ranging from raw materials to finished products. The GSC Corporations perform management and distribution of the materials and products. Finally, at the right, we got the Customers and the processes typically carried out by them.

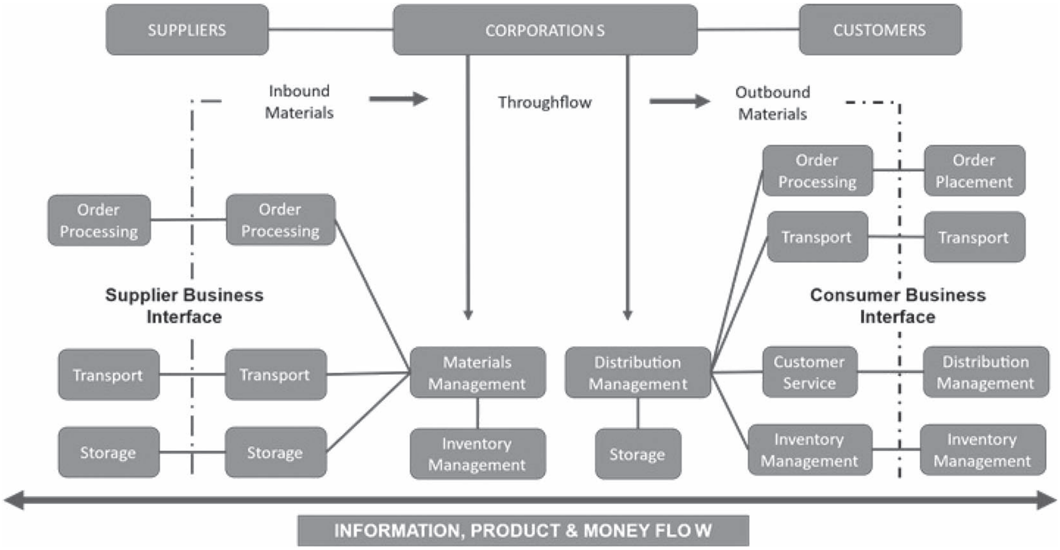


FIGURE 1. General Global Supply Chain representation based on [9].

In a traditional traceability scheme, information is stored on private servers with limited access for external members of the server organization. This design delays access to information, which inevitably affects decision-making processes on the production of goods and/or services. The Blockchain-based scheme configures each server on the chain as a node in the network. The most relevant information is available on all nodes simultaneously, facilitating access to information and allowing for streamlined decision-making processes. Similarly, the SSCM benefits from this scheme by improving the coordination of information flow in the SC. The Blockchain-based scheme uses smart contracts for managing agreements, pricing and improvements in the SC based on transaction data validated through these contracts.

4 Smart contract for GSC traceability anomaly detection

The smart contract deployed records all information generated by the GSC members, from raw materials to final products or services sent on each Blockchain transaction. Initially, the smart contract will issue notifications, and/or recommendations based on the terms initially agreed upon by the GSC members. This information and, subsequently, the data sent in each transaction is not susceptible to manipulation.

Once a transaction invokes the smart contract, it will automatically perform data validation looking for errors or anomalies. After this verification, it will generate recommendations based on the control variables, which will be available to all members of the GSC. To do this, we propose using a structure like MAPE-K [2], due to is decentralized nature that fits with the decentralized nature of the Blockchain technology.

The proposed knowledge component consists of all the information pieces generated by the different smart contracts and the transactions associated with them in the Blockchain network (i.e. the knowledge component consists of all the nodes that store the network information). The

Downloaded from https://academic.oup.com/jigpal/advance-article/doi/10.1093/jigpal/jzae047/7670719 by Universidad de Granada - Biblioteca user on 21 June 2024

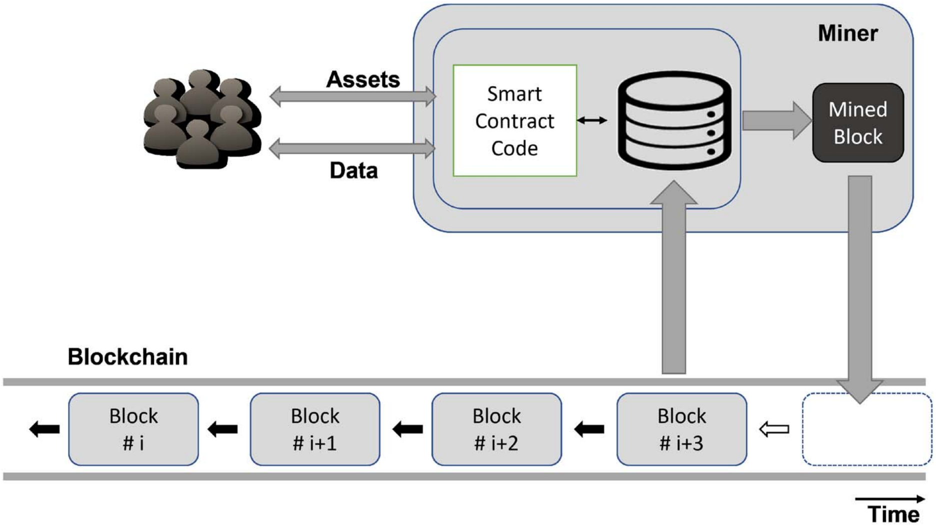


FIGURE 2. Smart Contract system based on [1].

monitoring module oversees each transaction made on the network, looking for syntax errors or semantic anomalies previously identified to notify those involved promptly about them.

The analysis module is responsible for finding errors or anomalies not previously identified, using the history of data stored in the network, comparing with the previous detections, and looking for ways to recall such errors or anomalies quickly and efficiently in the future. The planning module considers the information provided by the analysis module to update the smart contracts deployed in the network. Thus, it validates new transactions related to these contracts by looking for new errors or anomalies. Finally, the execution module sends the smart contract update transaction when required or a notification transaction to those involved in the traceability process so that they can make the appropriate decisions to prevent errors from recurring.

With the proposed scheme, it is possible to achieve greater control over the status of products sold, increase the information available for tracking the processes involved in the path of an altered product, available on a permanent and updated basis, and generate management recommendations and/or to set price penalties.

Figure 2 shows the smart contracts system with account information to assign assets to the user and recollect data, private storage and an executable code. As can be observed, a miner executes the smart contract and stores the transaction data on a new block on the Blockchain. Finally, smart contracts have access to the information generated on the Blockchain.

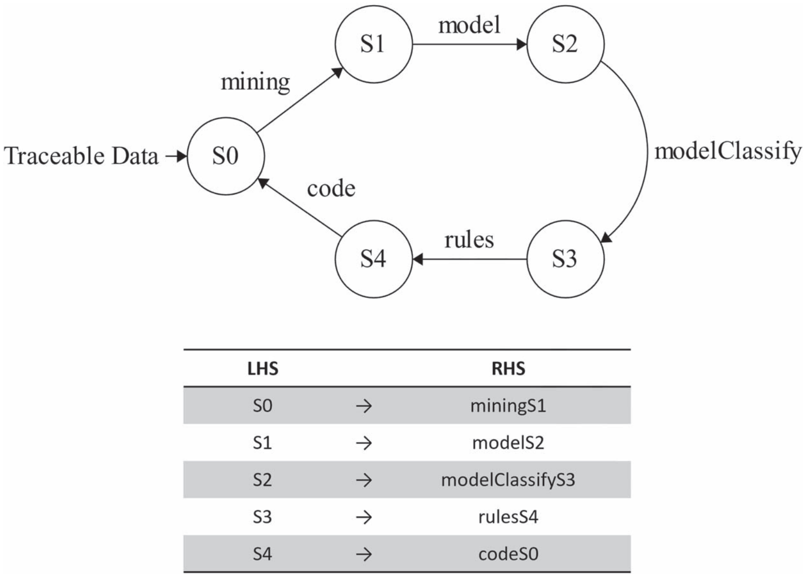
According to definition 12 on [13], the smart contract needs to satisfy the following equation:

$$C(S_i, Tx_i) = (S_j, R_j) \tag{1}$$

where C is the invoke smart contract, S represents all the possible states in the smart contract, Tx is a transaction on the blockchain network and R denotes all the possible responses to the current transaction.

Typical supply chain traceability problems on the main approaches based on their solutions are described in [3, 6, 15]. Missing data anomalies occur when no data value is stored for an

Downloaded from https://academic.oup.com/jigpal/advance-article/doi/10.1093/jigpal/jzae047/7670719 by Universidad de Granada - Biblioteca user on 21 June 2024



LHS		RHS
S0	→	miningS1
S1	→	modelS2
S2	→	modelClassifyS3
S3	→	rulesS4
S4	→	codeS0

FIGURE 3. Semantic Anomaly Detection Rules Generation Finite State Machine (FSM).

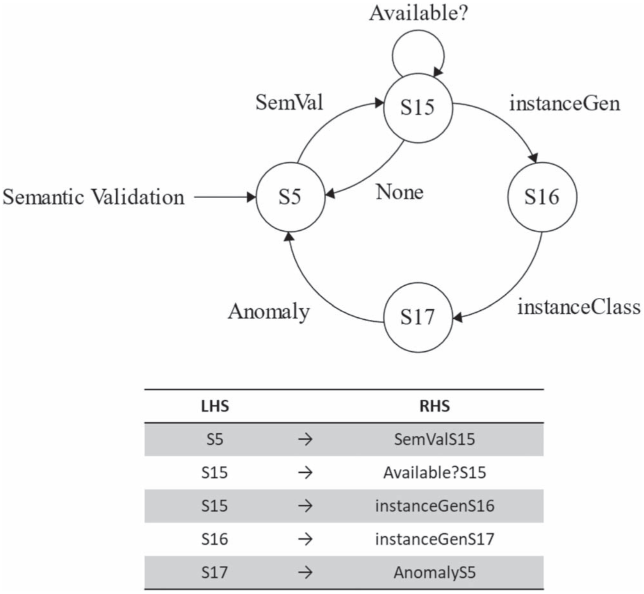
observed variable. Unknown data anomalies arise when the data stream contains additional data points unrelated to the observed variables. Duplicate data anomalies are originated when the data stream has identical values in all the observed variables. Syntaxis data anomalies appear when the information does not correspond with the observed variable's expected data type. Finally, semantic data anomalies occur when one or more instances do not follow the observed variables' usual pattern. This paper will focus on numeric syntaxis and semantic data anomalies only.

For semantic anomaly detection, we develop an algorithm that uses Random Forest or Decision Tree machine learning (ML) models, given the readability feature of these two ML models. From them, we generated rules translated into JavaScript language for the smart contract to detect semantic anomalies on the transaction sensed data. Although other ML models were tested only this two were used to generate semantic anomaly detection rules for the smart contract.

Figure 3 shows the general flow process of semantic anomaly detection rules generation, where Transaction data mining (S0), Model process (S1), Modeling classification (S2), Rules generation (S3), Code generation (S4) are the developed stages to generate a new smart contract version with better semantic anomaly detection capabilities. As previously mentioned, the Blockchain ledger processes and stores all the transaction data. The mining process adjusts the data to the model specifications. After that, the models are trained and evaluated to generate the rules used to give the smart contract the ability to detect the anomalies autonomously.

Figure 4 exposes the FSM for the semantic anomaly detection function on the smart contract. When a semantic validation is requested, the smart contract validates whether there are rules to perform this process (S15). If there are no rules, the value None is returned to the initial state (S5). If there are rules to perform semantic anomaly detection, the smart contract adjusts the data to a predefined format (S16) to create a dataset instance that will be classified (S17) as normal or abnormal using the current rules.

Downloaded from https://academic.oup.com/jigpal/advance-article/doi/10.1093/jigpal/jzae047/7670719 by Universidad de Granada - Biblioteca user on 21 June 2024



LHS		RHS
S5	→	SemValS15
S15	→	Available?S15
S15	→	instanceGenS16
S16	→	instanceGenS17
S17	→	AnomalyS5

FIGURE 4. Semantic Anomaly Detection FSM.

To give the smart contract the ability to detect syntax anomalies, we have developed an algorithm that looks through the sensed data in the current transaction data. For each transactional component, the algorithm checks if the transactions data received correspond with the expected data types. If one or several of these components do not fit the desired data types, then the smart contract stores them on the ledger as anomalies to be addressed by the network operators and keep the last correct transaction data value as the current data value.

Figure 5 shows the FSM for the syntax data validation, The initial stage (S5) at which a syntax check for the data is requested. Then the Syntax error check (S6) is called, and the data is checked according to the data type, composition and format, if the data is valid none will be returned, if not the transaction data will be marked as abnormal and the data with problems will be store for future analysis.

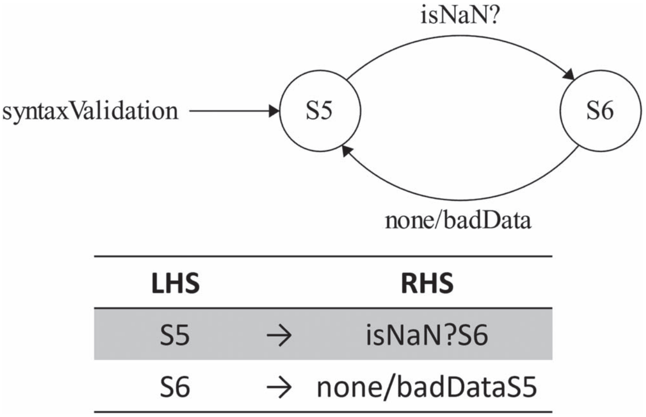
5 Smart contract self-update implementation and performance evaluation

We have implemented the Blockchain ecosystem for the traceability processes using Hyperledger Fabric[1] and JavaScript. To evaluate the smart contract, we use Hyperledger Caliper.[2] We decided to focus only on throughput, latency and successful and failed transactions based on the Blockchain challenges on [31] to ensure that the proposed smart contract will not threaten the network. Caliper was configured to simulate 6 clients sending transactions to the network, with a transaction load of 10 or 20 using the fixed load rate driver, a minimum duration of 60 seconds and a maximum period of 600 seconds. The driver used in the *TransactionLoad* parameter specifies a delay in the system

[1] https://www.hyperledger.org/

[2] https://www.hyperledger.org/use/caliper

Downloaded from https://academic.oup.com/jigpal/advance-article/doi/10.1093/jigpal/jzae047/7670719 by Universidad de Granada - Biblioteca user on 21 June 2024



LHS		RHS
S5	→	isNaN?S6
S6	→	none/badDataS5

FIGURE 5. Syntax Anomaly Detection FSM.

by modifying the transaction per second (TPS) driven parameter. These parameters were selected based on the pilot deployment used to test our proposal in a real environment using a production site. Nevertheless, other configuration parameters were tested, and results were consistent with the ones used for this research, In addition, we test a smart contract self-update process for semantic anomaly detection, using data collected from the production site used as a pilot test for our smart contract and the developed traceability platform using Blockchain technology.

Figure 6 shows the process defined for smart contract self-update. At the initial stage, the deployed smart contract receives a transaction after the correct composition of the transaction, and the smart contract performs syntax and semantic validation on the data. At this stage, two metrics for data reliability are estimated based on data correlation and the number of anomalies detected in the data. With all this information, the smart contract updates the states of all related variables.

Using the deployed smart contract, an external module sends a transaction to the network when enough transactions have been successful on the Blockchain network. As Figure 7 shows, a quick data mining process is applied to the downloaded data to adjust the data based on the requirements for the ML models. After the training process, the module generates rules based on the ramifications of the tree(s) on the selected model and creates a new smart contract version, translating the rules to Go and JavaScript languages. Finally, after the new rules have been added to the smart contract, a transaction request and a smart contract update are sent to the Blockchain network. If the updated transaction fails, an operator will be notified and must check the new smart contract code for error; if not, the recent transactions on the Blockchain network will use the updated smart contract.

Using the developed update module, we performed 19 Smart Contract autonomous updates, each update after 100 new successful transactions, the number was selected considering that each transaction on the productions sites is sent every 5 minutes, so every 8 hours a new smart contract version was installed on the Blockchain network. Nevertheless, these values can be adjusted according to the GSC environment or the traceability scheme specifications. The mean accuracy of anomaly detection was 92% when considering the cold start. A mean anomaly detection accuracy of 96% was achieved without a cold start. No update error was encountered during the test. The obtained results were good and prove that our proposed smart contract can perform data validations and can be updated without inconvenience in the selected test environment. To validate its behavior over time and/or with greater amounts of traceable data, a more extensive test is necessary. In this

Downloaded from https://academic.oup.com/jigpal/advance-article/doi/10.1093/jigpal/jzae047/7670719 by Universidad de Granada - Biblioteca user on 21 June 2024

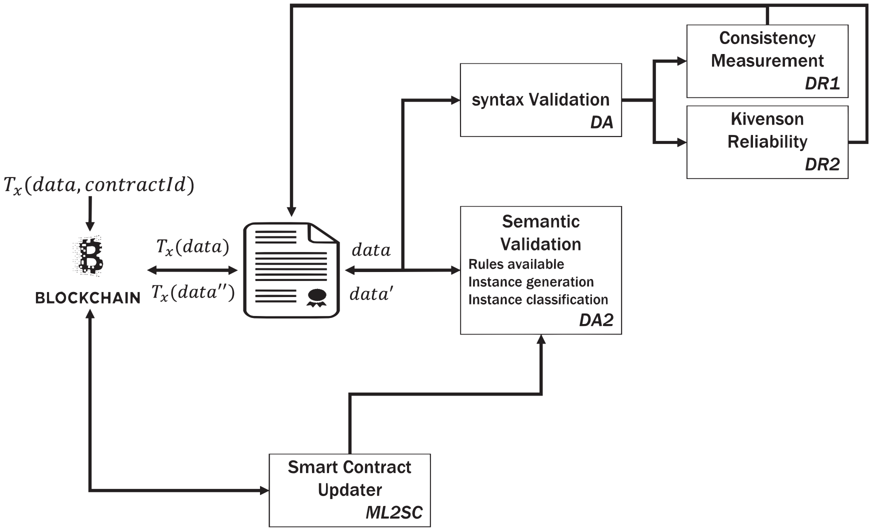


FIGURE 6. Smart contract self-update process.

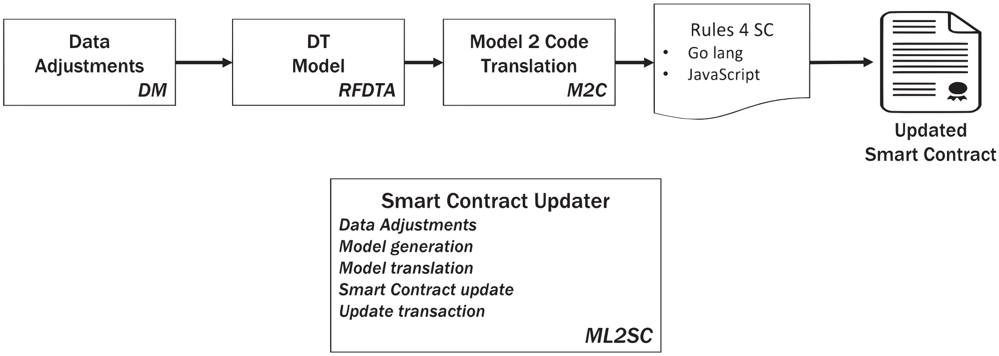


FIGURE 7. Smart contract update module.

case, the tool only uses the data stored in the Blockchain ledger, so anomaly detection could improve with each iteration. Even so, the accuracy of the smart contract detecting anomalies with each batch of data was high under these conditions. In addition, with each iteration, the number of rules used to detect anomalies will increase, making the semantic anomaly function more robust and trustworthy.

Table 1 shows the anomaly detection results for each version of the smart contract installed on the Blockchain network. The initial version 1.0 is the one considered as the cold start, on this contract, there is no semantic anomaly detection capacity at all. From version 2.0 and above the smart contract have the capacity to detect semantic anomalies using the data stored on the Blockchain network and the data marked as anomalies by previous smart contract versions are also stored on the Blockchain. Detecting anomalies requires a minimum number of successful transactions, so if the traceable processes are not correctly configured, a problem might arise. If the transaction data is

Downloaded from https://academic.oup.com/jigpal/advance-article/doi/10.1093/jigpal/jzae047/7670719 by Universidad de Granada - Biblioteca user on 21 June 2024

TABLE 1. Results for the self-update test simulation

Number of transactions	Confirmed Anomalies	Detected Anomalies	Accuracy (%)	Smart Contract Version
100	4	0	0	1.0
200	11	11	100	2.0
300	18	18	100	3.0
400	22	22	100	4.0
500	25	25	100	5.0
600	28	27	96	6.0
700	31	30	97	7.0
800	37	36	97	8.0
900	43	42	98	9.0
1000	47	43	91	10.0
1100	55	51	93	11.0
1200	62	58	94	12.0
1300	65	61	94	13.0
1400	74	72	97	14.0
1500	82	77	94	15.0
1600	86	83	97	16.0
1700	93	90	97	17.0
1800	98	93	95	18.0
1900	101	96	96	19.0
2000	106	101	96	20.0

filled with anomalies from the beginning, it will be classified as standard data, so good data may even be considered anomalous in this case.

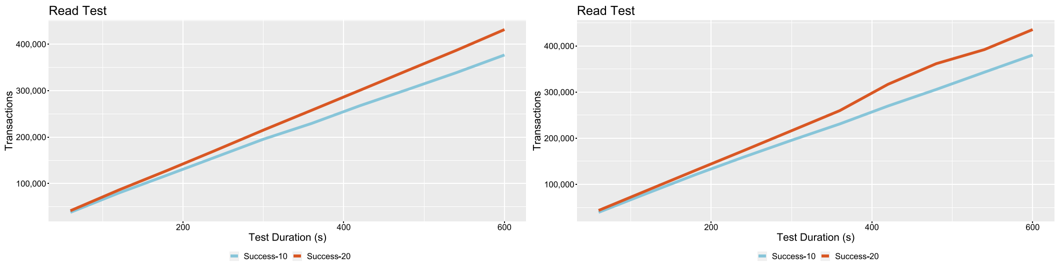
We have performed stress testing on the network in two stages. In the first stage, the Caliper sends workers to create objects on the Blockchain network. Subsequently, workers send read transactions for each created object none of the developed functions are used at this stage. The second stage is the update stage. The transaction is reviewed for errors or anomalies in the submitted data. We use the smart contract without the semantic anomaly detection capabilities and one update with rules to detect all the semantic anomalies found on all the collected data, around 1M instances or transactions.

Figure 8 shows the total number of transactions made correctly on the test network using six workers and transaction load values of 10 and 20, respectively. The results are similar because the asset creation and consult parts are the same. Other tasks running in the background on the host OS can cause slight differences.

Figure 9 shows the results for the update test on this case with the syntaxis anomaly detection only and the next one with both detections, syntaxis and semantics. As we can see, the results are like what we expected based on the previous Caliper results. There exists an increase in the number of failed transactions over time in both cases. Still, the results are similar, so adding the semantic detection function does not negatively impact the Blockchain network.

There is a clear difference between 10 and 20 as transaction loads in this case. During the reading test, the results were close between the two metrics. However, due to anomaly detection procedures,

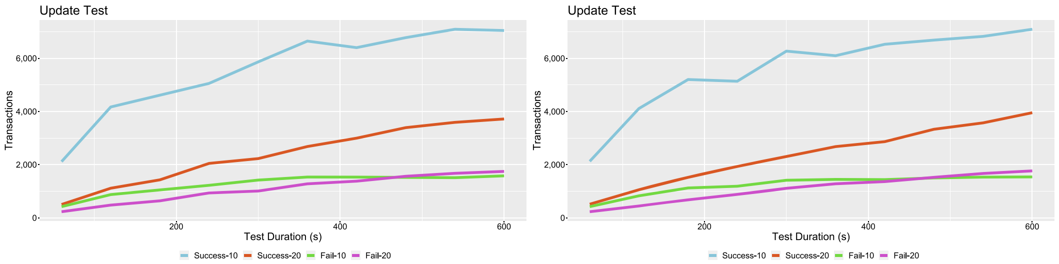
Downloaded from https://academic.oup.com/jigpal/advance-article/doi/10.1093/jigpal/jzae047/7670719 by Universidad de Granada - Biblioteca user on 21 June 2024



(a) Without the semantic anomaly detection

(b) With the semantic anomaly detection

FIGURE 8. Successful transactions in the Read test (without and with the semantic anomaly detection).



(a) Without the semantic anomaly detection

(b) With the semantic anomaly detection

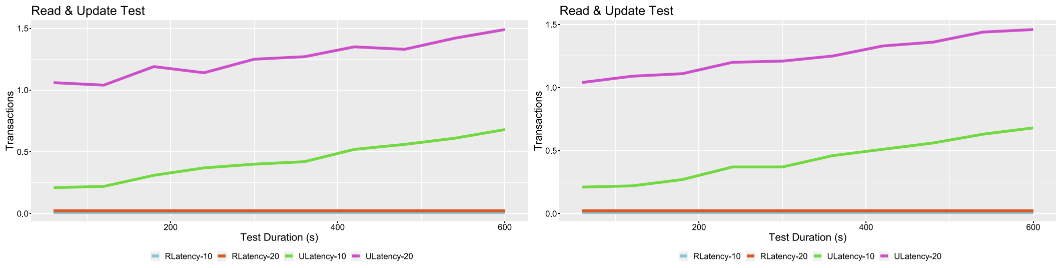
FIGURE 9. Successful and Failed transactions in the Update test (without and with the semantic anomaly detection).

the transaction process took longer during the update process, the smart contract has to perform additional processes with each transaction associated with the syntax and semantic data validations, and the fixed load rate driver reduced the number of transactions to maintain the selected transaction load on the Blockchain network. However, the behavior of both tests using the smart contract with and without the semantic anomaly detection function is remarkably similar, strengthening the conclusion that this function does not negatively impact the network.

Figures 10 and 11 show latency and throughput results for both smart contract versions. The results confirm what we see on the Read and Update test results. The read-throughput average for both cases is around 350 TPS with an average latency of 0.01 seconds, which mean that the information is available for all the chain members almost immediately. The updated throughput average in both cases is approximately 18 TPS with an average latency of 0.8 seconds. Although there is a significant difference in the average values in both cases is enough for the proposed traceability task. Most traceability processes can be carried out correctly with 18 transactions per second, and a latency of less than a second is considered real-time in most applications.

It is evident from the latency and throughput results that they have a similar pattern throughout all tests, increasing over time. It illustrates that the amount of stored data will consistently put more work into the semantic anomaly detection functions, the reliability estimation functions and the

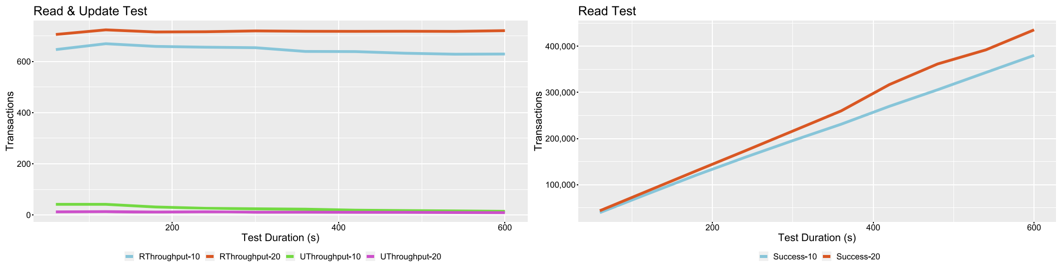
Downloaded from https://academic.oup.com/jigpal/advance-article/doi/10.1093/jigpal/jzae047/7670719 by Universidad de Granada - Biblioteca user on 21 June 2024



(a) Without the semantic anomaly detection

(b) With the semantic anomaly detection

FIGURE 10. Latency results in the Read and Update test.



(a) Without the semantic anomaly detection

(b) With the semantic anomaly detection

FIGURE 11. Throughput results in the Read and Update test.

recommendation functions built into the contract. Given that for most of this process, the smart contract must consult all the past data of variables. A limitation on how far back to check is needed to ensure that the load on the Blockchain network is on optimal values for long operation periods.

The developed smart contract will be able to keep up with new anomalies that may arise in the data stream in the future thanks to the self-update feature. In traceable processes, this is important due to the ever-changing environment and relationships between supply chain members. Supply chain members can consult all the information marked as anomalies at any time and adjust their processes to prevent these problems from occurring in the future. This information could also qualify supply chain members based on the number of anomalies in their data stream stored on the ledger.

ML algorithms will require more computational resources as the number of stored information increases, resulting in longer training times and longer execution processes for the semantic anomaly detection on the smart contract as rules become more complex. A more complex test would be needed to see the feasibility of this contract for all kinds of traceability applications. To update and deploy smart contracts autonomously, the update module must continuously run on all nodes with permission to deploy smart contracts on the Blockchain network introducing a new source of security problems that must be considered when deploying the Blockchain network and the permission assigned to each chain member.

Downloaded from https://academic.oup.com/jigpal/advance-article/doi/10.1093/jigpal/jzae047/7670719 by Universidad de Granada - Biblioteca user on 21 June 2024

6 Conclusions and future work

Sustainable supply chain management is a very complex and costly task to identify parameters that can affect the experience of end consumers. To facilitate the identification of this and other problems in an SCM, we have proposed the use of smart contracts with data validation in a traceability scheme in Blockchain.

With this development, the smart contract can detect syntax and semantic errors in the data and generate alerts about them so that supply chain operators can see the problems as soon as possible, correct the causes in time or prevent the products from reaching the end users. ML algorithms will require more computational resources as the number of stored information increases, resulting in longer training times. To update and deploy smart contracts autonomously and detect semantic anomalies, the developed tools must continuously run on all nodes with permission to deploy smart contracts on the Blockchain network.

We have developed a strategy to enable a smart contract to detect syntax and semantics anomalies in the data of Blockchain transactions in a traceability scheme. This strategy can translate Random Forest and Decision Tree rules into JavaScript code for smart contracts supported by Hyperledger Fabric. This strategy has proven useful in supply chain traceability systems, where most of the tracked data is generated autonomously. Also, the data gain a new level of trust, not only for being stored on the Blockchain ledger but also because the smart contract validates the data for each transaction.

As future work, additional tests on different configurations of the Blockchain network are required to achieve autonomous updating of the contracts deployed in the network, so that they can adapt to new types of anomalies that may arise and generate the appropriate recommendations or alerts for them. Allow the smart contract to reprocess past data to detect anomalies that were not detected by the previous version of the smart contract, which will provide the network operator with information about past errors.

Acknowledgements

We want to thank the Telematics Engineering Group (GIT) and the Research Group for Rural Development (Tull) of the University of Cauca and the Ministry of Science Technology and Innovation of Colombia (Minciencias) for the Ph.D. support granted to Cristian Valencia Payan. The research leading to these results has also received funding from ‘Estrategias para la valorización de los dulces tradicionales de Popayán’ (Code: 110380863995) founded by Minciencias–Colombia (Contract 127-2019), as well as Project ‘Incremento de la oferta de prototipos tecnológicos en estado pre-comercial derivados de resultados de I + D para el fortalecimiento del sector agropecuario en el departamento del Cauca’ funding by SGR (BPIN 2020000100098), and the Spanish R&D&i project GOMINOLA (PID2020-118112RB-C21 and PID2020-118112RB-C22) financed by MCIN/AEI/10.13039/501100011033. Funding for open access charge: Universidad de Granada / CBUA.

References

[1] M. Alharby and A. van Moorsel. Blockchain based smart contracts: a systematic mapping study. In *Arxiv*, pp. 125–140, 2017.

[2] P. Arcaini, E. Riccobene and P. Scandurra. Modeling and analyzing MAPE-K feedback loops

Downloaded from https://academic.oup.com/jigpal/advance-article/doi/10.1093/jigpal/jzae047/7670719 by Universidad de Granada - Biblioteca user on 21 June 2024

for self-adaptation. In *Proc. of 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS'15)*, pp. 13–23. Florence, Italy, 2015.

[3] P. Baumgartner and A. Krumpholz. Anomaly detection in a boxed beef supply chain. In *Proc. of the 13th Int. Conference on Computer Modeling and Simulation*. ACM, New York, NY, USA, 2021.

[4] D. Bechtsis, N. Tsolakis, E. Iakovou and D. Vlachos. Data-driven secure, resilient and sustainable supply chains: gaps, opportunities, and a new generalised data sharing and data monetisation framework. *International Journal of Production Research*, **207**, 1084–1098, 2021.

[5] M. Ben-Daya, E. Hassini and Z. Bahroun. Internet of things and supply chain management: a literature review. *International Journal of Production Research*, **57**, 4719–4742, 2019.

[6] A. Beteto, V. Melo, J. Lin, M. Alsultan, E. Mario Dias, E. Korte, D. A. Johnson, N. Moghadasi, T. L. Polmateer and J. H. Lambert. Anomaly and cyber fraud detection in pipelines and supply chains for liquid fuels. *Environment Systems and Decisions*, **42**, 306–324, 2022.

[7] C. Caballero-Gil, P. Caballero-Gil, J. Molina-Gil and N. García-Moreno. Ethereum-based decentralized car rental system. *Logic Journal of the IGPL*, **30**, 926–941, 2022.

[8] I. Erol, I. Murat Ar and I. Peker. Scrutinizing blockchain applicability in sustainable supply chains through an integrated fuzzy multi-criteria decision making framework. *Applied Soft Computing*, **116**, 108331, 2022.

[9] C. Fastbolt. *Program Management and Supplier of Quality Fasteners & Components*, 2021.

[10] W. Florkowski, R. Shewfelt, B. Brueckner and S. E. Prussia. Challenges in postharvest handling. In *Postharvest Handling*. Elsevier, 2009.

[11] J. Grover and S. Sharma. Security issues in wireless sensor network-a review. In *Proc. of 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO'16)*, pp. 397–404. Noida, India, 2016.

[12] I. H. Hong, F. Dang, Y. H. Tsai, C. S. Liu, W. T. Lee, M. L. Wang and P. C. Chen. An RFID application in the food supply chain: a case study of convenience stores in Taiwan. *Journal of Food Engineering*, **106**, 119–126, 2011.

[13] B. Hu, Z. Zhang, J. Liu, Y. Liu, J. Yin, R. Lu and X. Lin. A comprehensive survey on smart contract construction and execution: paradigms, tools, and systems. *Patterns*, **2**, 100179, 2021.

[14] E. Koberg and A. Longoni. A systematic review of sustainable supply chain management in global supply chains. *Journal of Cleaner Production*, **207**, 1084–1098, 2019.

[15] I. Konovalenko and A. Ludwig. Generating decision support for alarm processing in cold supply chains using a hybrid k-nn algorithm. *Expert Systems With Applications*, **190**, 116208, 2022.

[16] M. Li, K. Zhang, J. Liu, H. Gong and Z. Zhang. Blockchain-based anomaly detection of electricity consumption in smart grids. *Pattern Recognition Letters*, **138**, 476–482, 2020.

[17] L. Liu, W. T. Tsai, M. Z. A. Bhuiyan, H. Peng and M. Liu. Blockchain-enabled fraud discovery through abnormal smart contract detection on ethereum. *Future Generation Computer Systems*, **128**, 158–166, 2022.

[18] F. Longo, L. Nicoletti, A. Padovano, G. D'Atri and M. Forte. Blockchain-enabled supply chain: an experimental study. *Computers and Industrial Engineering*, **136**, 57–69, 2019.

[19] Q. Luo, R. Liao, J. Li, X. Ye and S. Chen. Blockchain enabled credibility applications: extant issues, frameworks and cases. *IEEE Access*, **10**, 45759–45771, 2022.

[20] Y. Mezquita, R. Casado-Vara, A. González-Briones, J. Prieto and J.M. Corchado. Blockchain-

Downloaded from https://academic.oup.com/jigpal/advance-article/doi/10.1093/jigpal/jzae047/7670719 by Universidad de Granada - Biblioteca user on 21 June 2024

based architecture for the control of logistics activities: pharmaceutical utilities case study. *Logic Journal of the IGPL*, **29**, 974–985, 2020.

[21] Y. Mezquita, J. Parra-Domínguez, M. E. Pérez-Pons, J. Prieto and J. M. Corchado. Blockchain-based land registry platforms: a survey on their implementation and potential challenges. *Logic Journal of the IGPL*, **30**, 1017–1027, 2022.

[22] Y. Mirsky, T. Golomb and Y. Elovici. Lightweight collaborative anomaly detection for the iot using blockchain. *Journal of Parallel and Distributed Computing*, **145**, 75–97, 2020.

[23] S. Morishima. Scalable anomaly detection in blockchain using graphics processing unit. *Computers & Electrical Engineering*, **92**, 107087, 2021.

[24] T. A. Musa and A. Bouras. Anomaly detection in blockchain-enabled supply chain: an ontological approach. In *Proc. of Int. Conference on Product Lifecycle Management (PLM'21)*, pp. 253–266. Curitiba, Brazil, 2022.

[25] United Nations. *A Guide to Traceability: A Practical Approach to Advance Sustainability in Global Supply Chains about the United Nations Global Compact*, 2014.

[26] Z. Ognjanović, A. Ilić Stepić and A. Perović. A probabilistic temporal epistemic logic: strong completeness. *Logic Journal of the IGPL*, **10**, 2022.

[27] K. Petersen, R. Feldt, S. Mujtaba and M. Mattsson. Systematic mapping studies in software engineering. In *Proc. of 12th Int. Conference on Evaluation and Assessment in Software Engineering (Ease'08)*, pp. 68–77. Bari, Italy, 2008.

[28] S. Seuring and M. Muller. From a literature review to a conceptual framework for sustainable supply chain management. *Journal of Cleaner Production*, **16**, 1699–1710, 2008.

[29] T. Thakur, A. Mehra, V. Hassija, V. Chamola, R. Srinivas, K. K. Gupta and A. P. Singh. Smart water conservation through a machine learning and blockchain-enabled decentralized edge computing network. *Applied Soft Computing*, **106**, 107274, 2021.

[30] S. Tian, F. Jiang and C. Huang. Global supply chain information compensation model based on free trade port blockchain information platform. *Lecture Notes on Data Engineering and Communications Technologies*, **107**, 288–300, 2022.

[31] H. Treiblmaier. Toward more rigorous blockchain research: recommendations for writing blockchain case studies. *Frontiers in Blockchain*, 0:3, **2**, 2019.

[32] N. Yakovleva and A. Flynn. Innovation and sustainability in the food system: a case of chicken production and consumption in the UK. *Journal of Environmental Policy and Planning*, **6**, 227–250, 2004.

Received 1 April 2023

Downloaded from https://academic.oup.com/jigpal/advance-article/doi/10.1093/jigpal/jzae047/7670719 by Universidad de Granada - Biblioteca user on 21 June 2024