A Blockchain-enabled and Transparent Evaluation of ML Models in the Decentralised Marketplace

Abstract— **In today's digital era, vast data in supply chain companies holds the potential for smarter products and services. Machine Learning (ML) and Artificial Intelligence (AI) unlock insights from this data, with many small ML developers owning suitable models. Yet, model buyers often face high costs or uncertainty in model quality. ML model owners hesitate to share or compete due to transparency and fairness concerns in evaluation. Decentralised model marketplaces aim to connect buyers and sellers, addressing cost, privacy, and fairness. However, gaps exist, hindering participation. This paper presents a decentralised evaluation platform for ML models, addressing marketplace challenges. Buyers evaluate models using automated ranking based on evaluation metrics, finding models meeting budget and quality expectations. Owners ensure transparent, fair, and immutable model ranking, enhancing motivation to share models. The designed workflow is implemented and analysed for efficiency, throughput, and costs.**

Keywords— Model Marketplace, Blockchain, Smart Contract, Machine Learning Model Evaluation, Supply Chain Data

I. INTRODUCTION

In the age of digital transformation, the widespread use of digital technology, for instance, enterprise resource planning (ERP), radio frequency identification (RFID), Internet of Things (IoT), etc [1, 2], in the supply chain and industries produce a large amount of data which has been stored internally or externally. On the other hand, Machine Learning (ML) and Artificial Intelligence (AI) have become indispensable parts of exploring insight from data and helping many industries and processes such as Energy, Manufacturing, Healthcare, and supply chain to perform smarter [3, 4, 5, 6, 7, 8]. The data owners need ML models to explore the hidden patterns from their stored data that give them valuable insight for effective decision-making and boost their products and services. Despite these intense needs, main industry actors such as producers or distributors are forced to buy high-cost models from a large corporation or centralised third party, which may not necessarily have a model with optimal performance or possess the desired features. An alternative approach for supplying affordable or robust models is turning to individual developers or small startups, but buyers suffer from the challenges of collecting well-trained ML models from such sellers. Despite a vastly growing number of ML engineers or small companies developed tremendous ML models, they are reluctant to share/sell their models as a result of a deficiency in incentive, payment fairness and transparency in evaluation and competition with other models.

In this paper, a secure and decentralised evaluation platform for ML models was developed. The proposed workflow enables ML model buyers (data owners or supply chain actors) to compare and select different ML models based on an automated ranking mechanism using several model evaluation metrics, besides, model owners (ML engineers) ensure model ranking and selecting are done in a transparent and immutable decentralised method. Our proposed design provides a decentralised method that helps buyers (like companies or supply chain actors) find a model from small ML model owners (for instance ML developers or small startups) who can provide lower price models and better performance and quality, concurrently, model owners are convinced with a transparent evaluation procedure and fair model-money swapping process. This transparency and fairness enhance the motivation of sellers to contribute more and share their best models.

In the implementation section, a supply chain company has a sample internal dataset of sales and looking for an optimised and affordable model to extract the best set of saleable products and utilise it for sales strategy. Simultaneously, several model owners have ML models and are willing to sell them. This paper assumes this demand and supply in a conceptual ML model marketplace and proposes a blockchain-enabled evaluation workflow that will play a vital role in creating fairness and transparency between model buyers and model owners inside our decentralised marketplace. For testing and validating, we execute our workflow with a smart contract developed via Solidity in the Truffle and Ganache as an Ethereum blockchain environment. The main contributions of our work are as follows:

- Proposing a decentralised and automated evaluation workflow that integrates blockchain technology, ensuring transparent, secure, and unbiased selection of ML models in the marketplace, thus enhancing buyer and seller motivation.
- Designing a mechanism which allows model buyers to specify their requirements transparently on the blockchain, attracting confident model owners and eliminating unexpected pricing discrepancies, thereby facilitating the interaction process.
- Implementing and validating the blockchain-enabled marketplace prototype through simulations on the Ethereum blockchain, demonstrating its robustness and stability even as participation scales, ensuring equitable and predictable cost distribution.

The paper reviews previous works in Section II, focusing on blockchain-based solutions for data and model marketplaces, highlighting gaps and challenges. The proposed design is implemented in Section III to address these challenges, and the results are analysed in Section IV.

II. RELATED WORKS

Ethereum blockchain is a popular platform for implementing smart contracts, chosen in this paper to ensure transparency and fairness in the evaluation process in decentralised marketplaces, as discussed in Section I. This section reviews existing solutions for blockchain-based marketplaces, identifies gaps and challenges, and explains the ML model of association rule mining and evaluation method.

A. Blockchain-Based Data/Model Marketplaces

A conceptual marketplace executes for swapping data, swapping models, and sometimes both. Besides, in a decentralised marketplace, there are several nodes like model owners and data owners. Each player can act as a buyer or seller of data and models depending on research and industry use cases. Fig. 1 shows the overall picture that properly categorises the different types of marketplaces, players, and objectives equipped by blockchain technology. In the following, the related works are considered and their research aims are revealed, then the categorization of these related works is shown in Table 1.

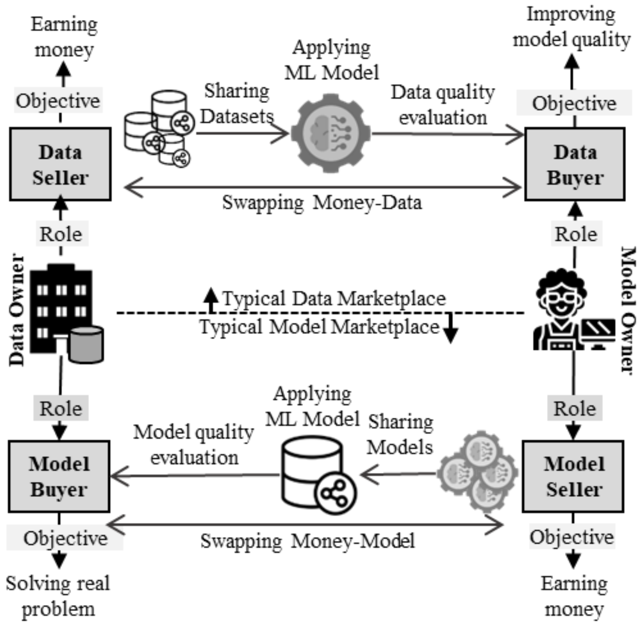


Fig. 1: What often happens in a typical Data/Model marketplace

Several blockchain-based solutions addressed the issue of creating a marketplace for machine learning models using blockchain technology. While Golden Grain [9] focuses on providing a secure and decentralised model marketplace, integrating benchmarking and model monetisation fairness, Dealer [10] emphasises the importance of differential privacy in the marketplace, considering Data Owners, Brokers, and Model Buyers. On the other hand, OmniLytics [11] aims at ensuring model and data security in decentralised ML, highlighting the role of smart contracts in guaranteeing payments and data integrity.

Some works focused on the data marketplace solutions that connect buyers to reliable data needed for the machine learning pipeline. Sterling [12] delves into the challenge of ensuring privacy when distributing and using data and introduces the concept of automatic contract verification. Meanwhile, Primal [13] presents a unique perspective on privacy, emphasising the need for preserving model and data privacy during cloud-based data evaluations. In contrast, Agora [14] focuses on enabling multiple parties concerned about privacy to exchange data without a trusted intermediary, achieving data privacy, verifiable outputs, and payment atomicity.

The advent of the Internet of Things (IoT) resulted in the production of a vast amount of data which can be sold and bought in the marketplace. Several solutions for blockchain-based marketplaces, deal with IoT and its intersection with blockchain-enabled data marketplaces and emphasise the need for secure, decentralised platforms for IoT data trade. While FASTDATA [15] concentrates on challenges like data fragmentation, privacy, and fairness in IoT data marketplaces, IDMoB [16] stresses the importance of a transparent trading platform and easy access to consensual IoT data.

Collaborative ML on the blockchain is a perspective that encompasses decentralised data marketplaces and collaborative learning platforms. Wibson [17] introduces the concept of a decentralised data marketplace, enabling individuals to securely sell personal information anonymously. Meanwhile, Buyukates et al. [18] explore the domain of collaborative ML on blockchain, emphasising the need for fair compensation for data owners and introducing mechanisms to prevent model owner evasion.

Reviewing the related works showed that there are data marketplaces focused on collecting high-quality data for training models via facilitating data exchange or data quality evaluation using blockchain (e.g. [12], [13]). On the other hand, few works were related to the model marketplace and more focused on motivating participants to share good-quality ML models (e.g. [9],[10]). Utilising a trusted execution environment (TEE for off-chain-on-chain integration) was one of the proposed designs for achieving fairness between model sellers and buyers in a typical ML model exchange. Among these works, Golden Grain [9] proposed a comprehensive design for addressing the challenges of benchmarking the correct model, protecting model privacy before selling (for model owners), and ensuring the authenticity of model performance before purchase (for model buyers).

After considering existing works, although comprehensive point of view in previous works addressed several issues, still there are challenges in the existing evaluation workflow for ML models in a typical ML model marketplace:

Challenge 1: Existing works involve model owners publishing their model characteristics report on the blockchain initially, followed by smart contract invocation for further processing. However, this can lead to unnecessary publication of many model reports on the blockchain, incurring costs for owners, especially if the evaluation process never commences. Furthermore, without knowledge of the benchmark dataset characteristics, model reports may not align with the specific needs of a typical buyer.

Challenge 2: In other related approaches, the model price remains unknown initially and is calculated for all participants at the end, allowing the buyer flexibility in their choice. However, this may leave model sellers unconvinced of the final price after a costly process. Additionally, buyers may select models based on undisclosed criteria or biases, causing dissatisfaction among sellers and hindering further marketplace contributions. Moreover, this approach imposes unnecessary costs for price calculation of unselected models.

Challenge 3: Previous methods ensure evaluation result authenticity by integrating blockchain with off-chain hardware environments, like TEE solutions. However, setting up such hardware is challenging for typical sellers, hindering their participation. Encouraging individual ML engineers or small startups to share models is vital for decentralised markets, but they may lack resources or expertise for complex setups, reducing their contributions.

Our design introduces a modified workflow for decentralised ML model evaluation using blockchain to tackle these challenges. As depicted in Fig. 2, buyers start the process by publishing model requirements, including the purchase budget upfront. Eventually, the selected model is automatically identified based on a pre-announced ranking formula. We'll analyse the impacts of our workflow compared to prior works in Section IV.

Table 1: Summary of existing solutions in the Data/Model marketplace

Existing Works	Goal	Methods/Techniques
Golden Grain [9]	A **model** marketplace facilitates monetising and sharing **data**	Off-chain-on-chain integration, trusted execution environments
Dealer [10]	A **model** marketplace focuses on differential privacy of three roles	Demonstration of a system with GUI
OmniLytics [11]	A marketplace enables secure **model** training, **data** sharing, and payment	Smart contracts for guaranteeing payment
Sterling [12]	A **data** marketplace enables data usage control and constraints	Secure enclaves (TEEs).
Primal [13]	A **data** evaluation framework for cloud-based data marketplaces	Protocols and methods for selecting data
Agora [14]	A **data-exchanging** platform enables multiple privacy-concerned parties	a round-based system for data exchange.
FAST DATA [15]	An IoT **data** marketplace ensures fair trading and novel data storage	Hyperledger Fabric, VerneMQ, Cloud
IDMoB [16]	An IoT **data** marketplace facilitates transparent trade and data accessing	Smart contract with Swarm
Wibson [17]	A **data** marketplace allows to sale of validated personal information	User control via smart contracts
Collaborative ML [18]	A marketplace for fair compensation and utilizes distributed **data** storage	Utilizes a distributed storage infrastructure

B. Machine Learning Models

Machine Learning (ML) encompasses statistical techniques for building models from data samples, categorized into supervised, unsupervised, and reinforced learning. Despite diverse applications, all ML techniques follow similar steps: training on a dataset and making inferences for new data [19]. Unsupervised learning techniques like Association Rule Mining (ARM) [20] and Frequent Episode Mining (FEM) [21] are increasingly important for pattern extraction, notably in real-world applications. ARM, a part of Knowledge Discovery in Databases (KDD), identifies associations between item sets, aiding decision-making. The Apriori algorithm [22] and the FP-Growth method [23] are key approaches in this field, although ARM has limitations like overlooking item variables such as price [24]. Extensions aim to enhance ARM's efficiency and address its binary focus [25]. Different ML algorithms vary in performance due to algorithms, parameters, and datasets, with ML as a Service (MLaaS) platforms facilitating comparison for optimal algorithm selection [26].

This paper assumes a company needs a Market Basket Analysis (MBA) model, typically executed via association rule models under unsupervised learning. Model sellers offer common MBA algorithms in our conceptual marketplace.

C. ML Models Evaluation Metrics

ML models produce various metrics indicating their quality or performance, such as Accuracy, Precision, Recall for classification, or MAPE and sMAPE for Time Series Forecasting. For the supply chain company's context, an MBA model requires association rules satisfying minimum Support and Confidence simultaneously, with Lift indicating rule confidence relative to expected confidence. Evaluation metrics like support, confidence, and lift are essential for assessing association rule models, as indicated by previous research [27, 28, 29]. Table 2 presents accepted metrics for evaluating and comparing association rule models, benchmarking against current standards.

Table 2: Common Evaluation Metrics for Association Rule ML Models

Evaluation Metric Name	Description
Support	It measures the frequency of an itemset. Support(X) = Transactions containing X / Total Transactions
Confidence	It measures the reliability of the inference made by a rule. Confidence (X→Y) = Transactions containing both X and Y
Lift	It measures how much more often a rule A->B occurs together than we expect Lift (X → Y) =Confidence (X → Y) / Support(Y)

III. Implementation

This paper proposes a 5-step automated workflow for collecting, evaluating, and comparing ML models while ensuring fairness and eliminating biases. The workflow utilizes blockchain for aggregating and comparing metrics, with a smart contract executed as a decentralised application (dApp) on Ethereum ensuring transparency and security. A prototype of the workflow is tested with a real-world dataset, employing Solidity for smart contract development and Truffle and Ganache for blockchain simulation. Figures 2 and 3 illustrate the execution logic, and Table 3 outlines the notations used.

A. Designing the evaluation workflow

In our conceptual model marketplace, supply chain companies seek optimized ML models tailored to their specific needs. As an example, a supply chain company aims to find the best ARM model for optimizing product sales. Multiple model owners believe their models can address this need. In a decentralised marketplace, parties can deploy a smart contract to execute business logic in sequential steps, as described below (details in Section III-B).

Step 1) Initiating smart contract with requirements: The supply chain company publishes its requirements on the ML marketplace via a smart contract, detailing the problem, desired model, evaluation metrics, scoring and ranking formulas, and benchmark dataset specifications. These requirements are stored on the blockchain network for transparency and immutability, fostering trust with model owners. A smart contract is initialized for the MBA problem, handling scoring, ranking, and transactions. The company selects a benchmark dataset of transactional sales, ensuring fairness in the model evaluation phase (Step 3) by providing the same dataset to all model owners.

Step 2) Getting model sellers' approval and granting access to the benchmark dataset: Model owners confirm their entry into the process upon reviewing the company's requirements on the blockchain. Upon confirmation, the smart contract progresses. Each model owner's approval triggers the transmission of a secret key for dataset decryption. Model owners receive the key, decrypt the dataset, and prepare their environment for model execution. The benchmark dataset is assumed to be stored by the buyer in a decryptable storage accessible with the provided secret key. The benchmark dataset, solely for model running, lacks real business data and sensitive information, eliminating the need for complex privacy measures. Symmetric cryptography decrypts the dataset, and the secret key is securely transferred via smart contract. Anyone with the key can decrypt the dataset, as it contains no private data, exclusively benefiting model owners.

Step 3) Aggregating returned evaluation metrics: model owners from Step 2 run their models on the provided benchmark dataset and submit the evaluation metrics to the smart contract (the set of evaluation metrics requested by the buyer in Step 1). These results are recorded on the blockchain, ensuring tamper-proof and verifiable data. Evaluation metrics are later reviewed by the buyer in Step 5 for authenticity which means that if the sellers have cheated and returned incorrect values of their model evaluation metrics, it will be revealed at the last step and they earn no money. All returned metrics are stored on the blockchain for transparency and immutability. This aggregated set of metrics is then processed in the next step, accessible to both buyers and sellers.

Based on previous works [9], buyers can enforce off-chain-on-chain mechanisms using TEE and Enclaves hardware like Intel SGX [30] in sellers' environments to ensure evaluation authenticity. Although enforcing such a TEE environment can have positive effects, it needs specific hardware and operating systems that make it difficult and costly to implement by sellers given these challenges, this paper opts to postpone evaluation examination, avoiding TEE enforcement for simplicity in participation, crucial for marketplace survival.

Table 3: Summary of existing solutions in the Data/Model marketplace

Notation	Description	Notation	Description
M_{od}	Model order description	B_{pk}	Buyer’s public key
B_{pb}	Buyer's purchase budget	B_p	Buyer’s private key
M_{sf}	Model scoring formula	M_{ad}	Model algorithm description
$EncD_{sk}$	Symmetric secret key for dataset	M_{em01}	Model evaluation metric number 01
$Addr_{bs}$	Storage address of dataset	M_{em02}	Model evaluation metric number 02
$Addr_m$	Storage address of model	M_{em03}	Model evaluation metric number 03
B	Buyer's address	E_s	Model evaluation score
S	Seller's address	E_r	Model evaluation rank
S_p	Seller's address participated	*Deal*	Deal associated data
S_e	Seller's address submitted metrics	M_a	Model authenticity approved state
S_s	Selected model seller	$AEncM_{sk}$	Asymmetric secret key
N_{sp}	Number of sellers participated	F_p	Stop participating flag
N_{se}	Number of sellers submitted metrics	F_e	Stop returning evaluation result flag
C	Contract’s address	AcB	Account balance
C_{st}	Contract’s state	$\$B_d$	Buyer’s deposit amount

Step 4) Ranking and Selecting Models: The smart contract calculates each model's rank based on evaluation results from Step 3, using scores defined in the model order description. Fairness is maintained as all models use the same benchmark dataset. Each model is assigned a rank based on their scores, with the top-ranked model flagged as the winner. Scores, ranks, and flags are recorded on the blockchain for transparency and verification. The smart contract autonomously executes scoring and ranking processes.

Step 5) Finalising stage, obtaining the selected model and releasing payment: After the scoring has been done in Step 4, the top-ranked model is identified. The trade confirmation request is then sent to both the model buyer and selected owners. Upon approval, the smart contract facilitates the swapping transaction, ensuring fairness and trust. This process, inspired by previous work [9], involves a fairness-swapping transaction where the model seller receives payment only if the buyer successfully decrypts the model using the secret key and confirms model authenticity. This secret key is provided by the selected seller after encrypting the model via an asymmetric method explained in Section III-B. Step 5 ensures that buyers cannot refuse payment upon receiving the decryption key. Conversely, sellers must reveal the key in time to receive payment, as the buyer's deposit can be refunded otherwise. All transaction details, including the selected model, amount, and timestamp, are permanently recorded on the blockchain. To maintain model privacy, a combination of symmetric and asymmetric cryptography schemes is employed, constituting a 3-step method.

In our design, we assumed the seller encrypted and stored his model in a desired storage system. Using a decentralised storage platform like IPFS or Filecoin [31, 32] may be an option for the selected seller. Then after transmitting the secret key to the buyer, he/she downloads and decrypts the model and should confirm what he obtained is authentic by comparing the evaluation results with the results sent by the seller in Step 3.

The 5-step workflow shown in Fig. 2. focuses on evaluation, scoring, ranking, and selection mechanisms. Implementing this model marketplace provides companies with a transparent and efficient platform for finding ML models that match their business problem, and have lower prices due to development by individual and small developers in a competitive and transparent approach.

B. *Deploying and executing the evaluation workflow*

In this section, we consider the company as the “model buyer” and model owners as the “model seller”, then the proposed evaluation workflow draws the steps and interactions between model buyer and model sellers using blockchain. According to Section III-A, the proposed evaluation workflow is a 5-steps interaction process. The smart contract is coded via Solidity to cover all these 5 steps. We showed the definition of smart contract logic in Fig. 3. To deploy and execute this smart contract on the blockchain, this paper uses Truffle and Ganache as a simulation environment of the Ethereum blockchain. In the following, we explained the technical implementation methods for the 5-step process and showed the results in Section III-C.

In Step 1, the model buyer (B) initiates the process by embedding MBA model requirements and trade conditions into the smart contract (C), deployed on the blockchain. These requirements and conditions (Mod, Msf, Bpb) are recorded on the blockchain for transparency and cannot be manipulated. Additionally, a benchmark dataset is generated synthetically from authentic historical sales data and encrypted using a symmetric algorithm. The encrypted dataset is stored, and the symmetric secret key (EncDsk) is shared with model sellers for Step 2. Desired model values and conditions are defined based on the expertise of the model buyer's experts, ensuring transparency for sellers before entering the process.

In Step 2, ML model owners (S) interested in participating sent approval (Sp) to the smart contract initiated by the model buyer. Upon approval, their information and public address were recorded on the blockchain, and the smart contract automatically sent a symmetric secret key (EncDsk) for dataset decryption. Model owners downloaded and decrypted the dataset (Addrbs) using the key, preparing their local environment for model execution. The smart contract internally stored the count of participation, allowing the buyer to track the number of approved model owners (Nsp) on the blockchain network.

In Step 3, owners executed their ML models off-chain and inputted evaluation metrics to the smart contract for on-chain storage and processing. Table 4 displays sample model types run on the benchmark dataset. Evaluation metrics (Mad, Mem01, Mem02, Mem03) were returned and aggregated by the smart contract. Each participating owner (Se) submitted a set of 4 values as requested in the requirement description (Mod) from Step 1. The aggregated set of all participants' metrics was stored on the blockchain for transparency and immutability. At this stage, the count of returned results was stored allowing the buyer to track the number of submitted evaluation results (Nse) on the blockchain network.

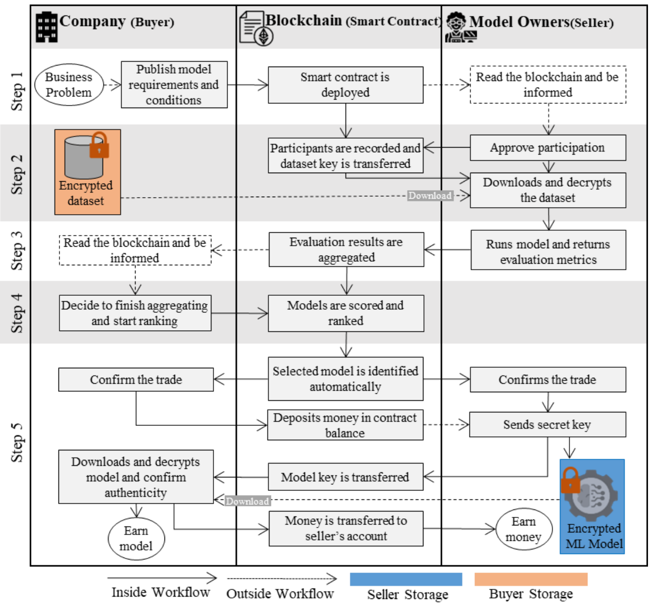

Company (Buyer)
Blockchain (Smart Contract)
Model Owners(Seller)
Step 1
Step 2
Step 3
Step 4
Step 5
Business Problem
Publish model requirements and conditions
Smart contract is deployed
Read the blockchain and be informed
Encrypted dataset
Participants are recorded and dataset key is transferred
Approve participation
Downloads and decrypts the dataset
Download
Read the blockchain and be informed
Evaluation results are aggregated
Runs model and returns evaluation metrics
Decide to finish aggregating and start ranking
Models are scored and ranked
Confirm the trade
Selected model is identified automatically
Confirms the trade
Deposits money in contract balance
Sends secret key
Downloads and decrypts model and confirm authenticity
Model key is transferred
Download
Encrypted ML Model
Earn model
Money is transferred to seller's account
Earn money
Inside Workflow
Outside Workflow
Seller Storage
Buyer Storage

Fig. 2: The 5-Step workflow for Secured and Transparent Evaluation of ML Models using Blockchain

In Step 4, the buyer retrieves blockchain data to determine the number of sellers who submitted evaluation metrics (Nse), using embedded variables and contract states in the smart contract. With this information, the buyer decides to stop accepting new metrics from sellers and proceeds to calculate model scores (Es). The smart contract automatically executes the scoring and ranking mechanism based on the initial contract definition. Each model's score is calculated using recorded evaluation metrics and the scoring formula (Msf) from Step 1. Models are ranked (Er) based on score sorting, with the top-ranked model flagged as the winner, and results are structured into a conceptual table, as shown in Table 4. All actors can transparently access comprehensive evaluation results for all participated models by executing specific functions embedded in the smart contract and reading recorded blockchain information.

In Step 5, the selected model (winner) is identified for swapping when the buyer obtains the model and pays the fixed budget (Deal) announced in Step 1. The model buyer (B) reads the blockchain and expresses interest in the deal, triggering a request for approval from the model owner (Ss). Upon confirmation, the buyer deposits the agreed amount ($Bd) into the smart contract (AcB [C]), equal to the budget (Bpb) specified in Step 1. The seller then sends the encrypted symmetric secret key (AEncMsk), allowing the buyer to download and decrypt the model (Addrm). After testing, the buyer confirms model authenticity (Ma) in the smart contract, completing the deal and triggering the automatic transfer of deposited funds to the seller's account (AcB [Ss]).

```
Smart contract definition {
Initialization is executed by buyer
    add ($B, M_{od}, M_{sf}, B_{pb}, EncD_{sk}, B_{pk}$) to ledger
  set $C_{st}$ = InitiatedByBuyer
SellerApproved is executed by seller
    read ($M_{od}, Addr_{bs}, EncD_{sk}, B_{pk}$) by seller
    send and add ($S_p$) to ledger
    read ($B, N_{sp}$)
  set $C_{st}$ = SellerApprovedParticipation
ModelMetricAggregation is executed by seller
    send and add ($S_e, M_{ad}, M_{em01}, M_{em02}, M_{em03}$) to ledger
    read ($B, N_{se}$)
  set $C_{st}$ = SellerMetricsAggregated
ModelSelection is executed by buyer
    send ($F_p, F_e$)
    add ($F_p, F_e, E_s, E_r, S_s$) to ledger
    read ($E_s, E_r$)
  set $C_{st}$ = SelectedModelIdentified
DealFinalisation is executed by buyer
    send and add ($B$, Deal → approved) to ledger
    send ($S_s$, Deal → approved)
    send ($B, \$B_d \rightarrow C$)
    $AcB[B] = AcB[B] - B_{pb}$
    $AcB[C] = AcB[C] + B_{pb}$
    send and add ($S_s, AEncM_{sk}$) to ledger
    send and add ($B, M_a \rightarrow$ approved)
    require ($\$B_{d} = B_{pb}$) $\rightarrow AcB[S_s] = AcB[S_s] + B_{pb}$
  set $C_{st}$ = DealCompleted }
```

Fig. 3: Definition of smart contract logic for evaluation workflow

In this Step, the final model is a sensitive asset protected for seller privacy. To achieve this, the selected model owner (Ss) performs a 3-step cryptography technique off-chain. Firstly, the buyer's public key (Bpk) is transferred to the seller. Secondly, the model is encrypted using a symmetric scheme, generating a symmetric secret key. Thirdly, this key is encrypted using an asymmetric scheme with the buyer's public key, creating an encrypted symmetric secret key (AEncMsk) usable only with the buyer's private key (Bp). The decrypted key is transferred to the buyer through the smart contract, ensuring only the buyer can decrypt the model.

Table 4: ML Models Evaluation Metrics Returned and Aggregated

ML Model Owner Code	ML Model Title	Support Metric Value	Confidence Metric Value	Lift Metric Value	Overall Score	Rank	Best Model
ML1	Apriori Algorithm	s1	c1	l1	E_{s1}	E_{r3}	0
ML2	FP-Growth Algorithm	s2	c2	l2	E_{s2}	E_{r1}	1
ML3	Apriori Algorithm	s3	c3	l3	E_{s3}	E_{r2}	0
ML4	FP-Growth Algorithm	s4	c4	l4	E_{s4}	E_{r4}	0

C. *Implementation Results*

Smart contracts contain methods and variables that define their state which are stored in blockchain. After its deployment, a smart contract acquires its own Ethereum address. Reading data from a smart contract is free while transacting with a contract, i.e., by creating it, or interacting with it via a method and changing its state (by submitting or altering data) requires gas and time [14].

Eventually, based on the evaluation workflow's steps, the proposed evaluation mechanism was implemented and validated in a conceptual model marketplace. In our marketplace, we assumed that a supply chain company (a buyer) acquired its desired ML model with the transparent and secure comparison of several ML models which have been provided by several model sellers. We applied the Truffle and Ganache as a simulated blockchain environment. For implementing the evaluation workflow prototype, we used

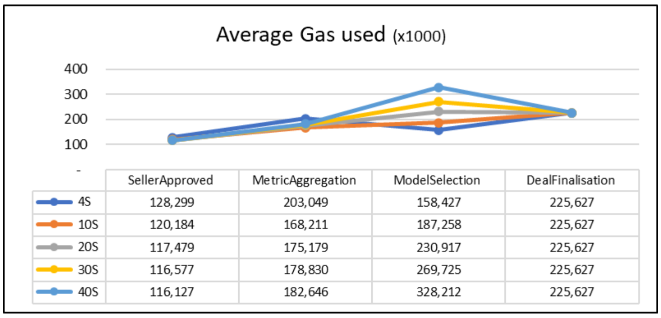
Solidity [33] for the smart contract. We measured the cost of deploying the contracts (Step 1) and executing transactions (Steps 2 to 5) in Gas, ETH and USD.

We considered the amount of gas used in different scenarios in terms of different numbers of sellers, from the small number of participants including 1 buyer and 4 sellers (named 4S scenario) which were raised gradually to 40 sellers (named 10S, 20S, 30S and 40S scenarios). We depicted the results of executing workflow via smart contract on the simulated blockchain in the charts and tables that are shown in Fig. 4, 5, 6 and 7. Overall for all scenarios, the outputs of executing around 150 functions and methods were recorded precisely, and then we calculated all values by the average of results from all scenarios. The average amount of gas used for one completed round of workflow (including smart contract deployment and transactions) was 5,265,770, equal to 0.017203525 Ether and 61,64 US dollars (according to the exchange rate of 26th March 2024: 1 ETH = 3,583 USD). We considered the common exchange rate of gas usage to ETH (1 ETH = 10^18 wei).

Smart contract deployment cost: In the designed workflow which was formulated in Fig. 3, a smart contract was deployed only by one party (buyer) and one time (Step 1) for each model's evaluation round. As expected, due to the size of the bytecode, deployment of smart contract had the most significant cost than the other four steps. We considered the deployment cost (Step 1) separately and eliminated its cost from charts to show a better comparison among transaction steps (Steps 2 to 5). The deployment of smart contract consumed 4,503,919, 0.015200727 and 54.46 in terms of average gas used, ETH and USD, respectively. Due to the logic of the proposed evaluation workflow, the costs of deploying smart contract are paid by the model buyer.

Transaction cost: We measured the gas used in executing steps and recorded it to consider the average costs (among different scenarios while the number of participants increased) for each step/role. According to Fig. 4, the comparison between the 4S to 40S scenario showed no significant growth in the average cost of steps 2, 3 and 5 while the number of participants (sellers) increased. These three steps have effects on sellers' costs, so it means that sellers will not pay the unfair extra cost because of others' participation. On the other hand, when the number of participants increased, the total cost for steps 2 and 3 raised at an average rate with negligible standard deviation, which means the amount of cost for the entire workflow execution will be predictable and grow linearly, not exponentially (Fig. 5). A significant rise in gas used (42,446 gas or 20% growth) has been seen in step 4 where the sellers' number grew, incrementally. This step is model selection and is executed only by the buyers who expect to pay more money to score and rank more models, moreover, no growth in cost is imposed on sellers due to the logic of workflow. We can see this expected cost growth (ETH) in Fig. 6 in the "Model Selection" column.

After finalising Step 5 in our conceptual marketplace, there are three types of parties including Buyer, Seller, and Selected seller. According to Fig. 7, the cost of the process was properly distributed among the parties who were engaged in each step without imposing the cost on those who did not contribute. For instance, only the seller whose model was flagged as a selected model in Step 5, paid the transaction cost for dealing with the buyer (which was 85,943 gas equal to 0.8 USD), while other participating sellers were not imposed any cost fairly, because their models were not selected. The selected seller was paid the full amount of money that was announced in Step 1. Due to the smart contract logic, the selected seller is allowed to release the model secret key only if the buyer deposited the correct ETH amount into the smart contract before ($Bd = Bpb).



	SellerApproved	MetricAggregation	ModelSelection	DealFinalisation
4S	128,299	203,049	158,427	225,627
10S	120,184	168,211	187,258	225,627
20S	117,479	175,179	230,917	225,627
30S	116,577	178,830	269,725	225,627
40S	116,127	182,646	328,212	225,627

Fig. 4: Average gas used for Steps in different scenarios

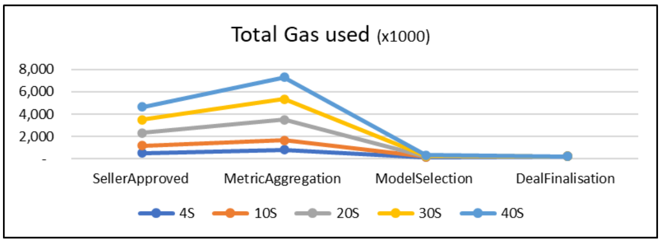
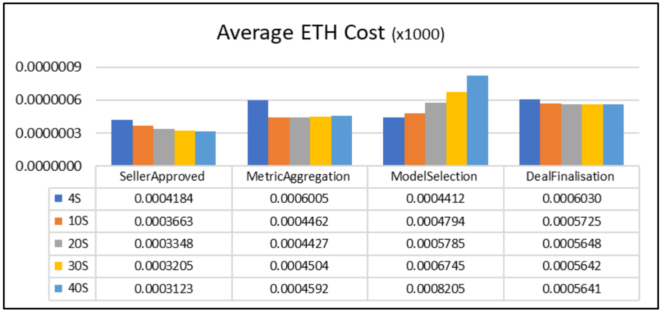


Fig. 5: Total gas used for Steps in different scenarios



	SellerApproved	MetricAggregation	ModelSelection	DealFinalisation
4S	0.0004184	0.0006005	0.0004412	0.0006030
10S	0.0003663	0.0004462	0.0004794	0.0005725
20S	0.0003348	0.0004427	0.0005785	0.0005648
30S	0.0003205	0.0004504	0.0006745	0.0005642
40S	0.0003123	0.0004592	0.0008205	0.0005641

Fig. 6: Average cost of Ether for Steps in different scenarios

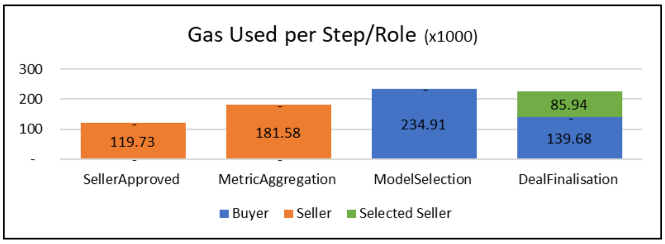


Fig. 7: Average gas used for Steps/Roles

IV. Analysis

A. *Proposing a decentralised evaluation workflow*

In this paper, a novel blockchain-enabled evaluation workflow for ML models was proposed. This decentralised evaluation focused on the evaluation, comparison and selection of the appropriate ML model among various models. Our designed method can be embedded in any type of model marketplace as the main decentralised method for aggregating models' evaluation metrics and ranking. This enables a transparent and automated model selecting process, without involving third parties, and all participants are ensured privacy, security and transparency during the evaluation process in line with the decentralised ledger technology. This transparency will have a positive impact on the motivation of both parties especially model owners in the marketplaces.

B. Addressing existing challenges

After reviewing related works, challenges were identified that could undermine actor motivation and threaten marketplace survival. To tackle these challenges elaborately explained in Section II, new contributions were integrated into the evaluation workflow and steps:

Firstly, model buyers initiate the process by publishing the requirements and conditions of the model that they are looking for on the blockchain. This approach results in the participation of more confident model owners who are completely aware of the buyers' specific requirements and the budget (if they win). Moreover, model owners only need a low-cost entrance confirmation to participate and no need to publish a costly model report first on the blockchain.

Secondly, the model buyers' budget for this purchase is known at the beginning of the process and the deal's worth has been revealed. Thus, there is no need to calculate any price. This results in the sellers who are agreed with the price entering the process, which means more confident model sellers enter the process, without any dissatisfaction about the final price that they may not expect and waste time. On the other hand, no cost is imposed on the buyer by calculating the price of several models, because our new workflow eliminates the need for a price-calculating stage for models.

Thirdly, the selected model is identified automatically based on the pre-announced formula and the buyer has only one choice: select the rank 1 model or not. This results in both the buyer and selected seller only having the choice to approve the deal or not. The buyer cannot select another model based on unknown bias, which means that model sellers are properly convinced that if they have provided the best model, they will be ranked 1 and only the rank 1 can be chosen by the buyer.

Fourthly, the evaluation workflow deters fraudulent sellers from entering by imposing fair costs and requiring confirmation of model authenticity by the buyer before payment. This discourages suspicious sellers from participating without imposing specific hardware or complicated off-chain engagement, positively impacting the willingness of individual model sellers to participate.

C. Addressing privacy and security issues

Model Privacy: in Step 3 of the evaluation workflow, the ML models were run on the model owners' environment and these models were kept locally. Besides, only evaluation metrics were transferred, and no other statistics of the model were shared in the smart contract or buyer before the deal. Moreover, in Step 5, when the deal is executed, although the encrypted symmetric secret key was transferred and may be obtained by others, the only one who can decrypt it is the buyer because this key was generated via the buyer's public key with the asymmetric algorithm, thus model's privacy preserved and if someone else would obtain the encrypted secret will be failed to decrypt the model. The asymmetric encryption algorithm used to encrypt is semantic secure [34].

Data Privacy: in step 2 of the evaluation workflow, the model buyer prepared, encrypted and stored a benchmark dataset which was a sample set that was just utilised for model running. It is not a real business dataset and does not contain an important business value, thus it is not necessary to impose costly complicated privacy or encryption techniques. The buyer used the symmetric cryptography scheme to decrypt the benchmark dataset and then transfer the symmetric secret key to the sellers that have confirmed participation (Step 2). The model owners got a secret key for decrypting the benchmark dataset. We assume that everyone who obtained the secret key can decrypt the sample dataset because it contains no private or sensitive data. Indeed, the authentic business data have not been contributed to the process and are kept private and secure in the buyer's local area.

D. Analysing implementation results

According to the implementation part and the outputs, the use of simulated Ethereum blockchain and smart contract underpins a robust design of evaluation workflow for ML models within a conceptual marketplace, demonstrating both stability and fairness essential for real-world applications. The implementation results in Section III-C ensure that the smart contract and workflow remain stable and predictable even as the number of participants increases.

The separate consideration of the deployment step and transaction steps showed that Step 1, which involves the smart contract's deployment by a buyer, incurred the highest cost, reflecting a deliberate design to centralize initial expenses to the initiating party. This design choice ensures that no extra costs are imposed on other participants not directly involved in this step, thereby maintaining fairness in cost allocation. Moreover, the average costs associated with Steps 2, 3, and 5 did not significantly increase despite a rise in the number of sellers (from scenario 4 to 40 sellers), highlighting the workflow's stability and cost predictability. Notably, Step 4, where the buyer ranks models, displayed a moderate increase in cost, which is justified as the buyer expected it since scoring and ranking more models demands more computational effort but does not place an additional cost on the sellers.

The cost distribution during evaluation was fair, with no additional costs imposed on sellers whose models weren't selected in Step 5, ensuring equitable treatment. Within the blockchain framework, each participant only bears costs directly tied to their involvement, fostering fairness. This equitable approach enhances economic efficiency in real-world applications, boosting motivation among individual/small developers to trust the evaluation workflow.

Conclusion

In this paper, we designed and demonstrated the effectiveness of a blockchain-enabled and transparent evaluation for machine learning (ML) models within a conceptual decentralised marketplace. The proposed evaluation workflow not only addresses the critical challenges of cost, quality, and sharing sensitive secrets faced by model buyers and sellers who interact for trading ML models, but it also fosters a fair and competitive dealing platform for both small and large actors. The transparency and security enabled by blockchain technology ensure that all participants can engage in the marketplace with confidence, knowing that the evaluation process is impartial and secure. This paper suggests that the principles of transparency, fairness and security are not only viable but also superior to traditional centralized models, so they need to be included in a decentralised marketplace precisely.

The implementation of the designed evaluation workflow was coded in a smart contract and executed as a decentralised application (dApp) on simulation Ethereum blockchain, providing evidence of its robustness, scalability, and ability to maintain stability even as the number of marketplace participants increases. The cost distribution among the participants has been managed equitably, which is crucial for the sustainability and attractiveness of the marketplace. This cost management ensures that no unnecessary financial burdens are placed on any party, particularly those not directly involved in transactions. Moreover, the privacy measures implemented for both model and data handling reassure participants that proprietary and sensitive information remains secure, addressing significant concerns in the adoption of a decentralised evaluation process. This paper's contributions to developing a decentralised, blockchain-enabled evaluation framework not only advance the field technically but also encourage greater participation by offering a more balanced, efficient, and transparent model trading environment.

This paper contributes to decentralised ML model marketplaces by introducing novel methods and addressing real-world challenges. Future research can expand on these results by integrating more complex workflows, such as combining model marketplaces with data marketplaces to accommodate different types of buyers and sellers. Additionally, the efficiency of decentralised evaluation can be improved by measuring block mining time on Testnet blockchains and incorporating other types of ML models, including supervised algorithms. Exploring the integration of this workflow with off-chain secure hardware environments could enhance model authenticity control for buyers while maintaining simplicity for sellers, which is crucial for marketplace sustainability.

REFERENCES

[1] McAfee, A., Brynjolfsson, E., Davenport, T.H., Patil, D.J. and Barton, D., “Big data: the management revolution,” Harvard Business Review, 90(10), pp.60-68, 2012.

[2] Fosso Wamba, S., Gunasekaran, A., Dubey, R. and Ngai, E.W., “Big data analytics in operations and supply chain management,” Annals of Operations Research, 270, pp.1-4, 2018.

[3] Younis, H., Sundarakani, B. and Alsharairi, M., “Applications of artificial intelligence and machine learning within supply chains: systematic review and future research directions,” Journal of Modelling in Management, 17(3), pp.916-940, 2022.

[4] Zamani, E.D., Smyth, C., Gupta, S. and Dennehy, D., “Artificial intelligence and big data analytics for supply chain resilience: a systematic literature review,” Annals of Operations Research, 327(2), pp.605-632, 2023.

[5] Awan, U., Kanwal, N., Alawi, S., Huiskonen, J. and Dahanayake, A., “Artificial intelligence for supply chain success in the era of data analytics,” The Fourth Industrial Revolution: Implementation of artificial intelligence for growing business success, pp.3-21, 2021.

[6] Mhlanga, D., “Artificial intelligence and machine learning for energy consumption and production in emerging markets: a review,” Energies, 16(2), p.745, 2023.

[7] Ge, Z., Song, Z., Ding, S.X. and Huang, B., “Data mining and analytics in the process industry: The role of machine learning,” IEEE Access, 5, pp.20590-20616, 2017.

[8] Rai, R., Tiwari, M.K., Ivanov, D. and Dolgui, A., “Machine learning in manufacturing and industry 4.0 applications,” International Journal of Production Research, 59(16), pp.4773-4778, 2021.

[9] J. Weng, J. Weng, C. Cai, H. Huang and C. Wang, "Golden Grain: Building a Secure and Decentralized Model Marketplace for MLaaS," in IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 5, pp. 3149-3167, 1 Sept.-Oct. 2022.

[10] Liu, J., Lou, J., Liu, J., Xiong, L., Pei, J. and Sun, J., “Dealer: an end-to-end model marketplace with differential privacy,” Proceedings of the VLDB Endowment, 14(6), 2021.

[11] Liang, J., Li, S., Cao, B., Jiang, W. and He, C., “Omnilytics: A blockchain-based secure data market for decentralized machine learning,” arXiv preprint arXiv:2107.05252, 2021.

[12] Hynes, N., Dao, D., Yan, D., Cheng, R. and Song, D., “A demonstration of sterling: a privacy-preserving data marketplace,” Proceedings of the VLDB Endowment, 11(12), pp.2086-2089, 2018.

[13] Song, Q., Cao, J., Sun, K., Li, Q. and Xu, K., “Try before you buy: Privacy-preserving data evaluation on cloud-based machine learning data marketplace,” In Annual Computer Security Applications Conference (pp. 260-272), 2021.

[14] Koutsos, V., Papadopoulos, D., Chatzopoulos, D., Tarkoma, S. and Hui, P., “Agora: A privacy-aware data marketplace,” IEEE Transactions on Dependable and Secure Computing, 19(6), pp.3728-3740, 2021.

[15] Dixit, A., Singh, A., Rahulamathavan, Y. and Rajarajan, M., “Fast data: A fair, secure and trusted decentralized IIoT data marketplace enabled by blockchain,” IEEE Internet of Things Journal, 2021.

[16] Özyilmaz, K.R., Doğan, M. and Yurdakul, A., “IDMoB: IoT data marketplace on blockchain,” Crypto Valley Conference on Blockchain Technology (CVCBT) (pp. 11-19). IEEE, 2018.

[17] Travizano, M., Sarraute, C., Dolata, M., French, A.M. and Treiblmaier, H., “Wibson: A case study of a decentralized, privacy-preserving data marketplace,” Blockchain and distributed ledger technology use cases: Applications and lessons learned, pp.149-170, 2020.

[18] Buyukates, B., He, C., Han, S., Fang, Z., Zhang, Y., Long, J., Farahanchi, A. and Avestimehr, S., “Proof-of-Contribution-Based Design for Collaborative Machine Learning on Blockchain,” arXiv preprint arXiv:2302.14031, 2023.

[19] Alpaydin, E., “Introduction to machine learning”. MIT Press, 2020.

[20] Agrawal, R. and Srikant, R., “Fast algorithms for mining association rules,” in Proc. 20th int. conf. very large databases, VLDB (Vol. 1215, pp. 487-499), 1994.

[21] Mannila, H., Toivonen, H. and Inkeri Verkamo, A., “Discovery of frequent episodes in event sequences,” Data mining and knowledge discovery, 1, pp.259-289, 1997.

[22] Agrawal, R., Imieliński, T. and Swami, A., “Mining association rules between sets of items in large databases,” In Proceedings of the ACM SIGMOD international conference on Management of data (pp. 207-216), 1993.

[23] Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: a frequent pattern tree approach. Data Min. Knowl. Discov. 8(1), 53–87, 2004.

[24] Ahmed, U., Srivastava, G. and Lin, J.C.W., “A federated learning approach to frequent itemset mining in cyber-physical systems,” Journal of Network and Systems Management, 29, pp.1-17, 2021.

[25] Savasere, A., Omiecinski, E. and Navathe, S., “An Efficient Algorithm for Mining Association Rules in Large Databases,” In Proceedings of the 21st International Conference on Very Large Databases (VLDB) (pp. 432-444), 1995.

[26] Ribeiro, M., Grolinger, K. and Capretz, M.A., “MLaaS: Machine learning as a service,” IEEE 14th International Conference on Machine Learning and Applications (ICMLA) (pp. 896-902), 2015.

[27] Kumbhare, T.A. and Chobe, S.V., “An overview of association rule mining algorithms,” International Journal of Computer Science and Information Technologies, 5(1), pp.927-930, 2014.

[28] Addi, A.M., Tarik, A. and Fatima, G., “Comparative survey of association rule mining algorithms based on multiple-criteria decision analysis approach,” 3rd International Conference on Control, Engineering & Information Technology (CEIT) (pp. 1-6). IEEE, 2015.

[29] Jaiswal, V. and Agarwal, J., “The evolution of the association rules. International Journal of Modelling and Optimization,” 2(6), p.726, 2012.

[30] F. McKeen et al., “Intel software guard extensions (intel sgx) support for dynamic memory management inside an enclave,” in Proc. ACM Hardware Archit. Support Secur. Privacy, pp.1–9, 2016.

[31] J. Benet, “Interplanetary file system,” [Online]. Available:https://ipfs.io/, 2014.

[32] Protocol Labs. “FileCoin: A Decentralized Storage Network,” Technical Report. [Online]. Available: https://filecoin.io/, 2018.

[33] “Solidity,” [Online]. Available: https://solidity.readthedocs.io, 2016.

[34] S. Goldwasser and S. Micali, “Probabilistic encryption,” J. Comput. Syst. Sci., vol. 28, no. 2, pp. 270–299, 1984.