Evaluating Data Trust in Blockchain-Based IoT Systems Using Machine Learning Techniques

Rashmi Ratnayake*, Madhusanka Liyanage†, Liam Murphy‡
*†‡School of Computer Science, University College Dublin, Ireland
Email: *rashmi.ratnayake@ucdconnect.ie, †madhusanka@ucd.ie, ‡liam.murphy@ucd.ie

***Abstract*—The convergence of blockchain and Internet of Things (IoT) has become increasingly prevalent recently, as it addresses challenges such as single point of failure and security concerns associated with IoT. Blockchain offers immutable data storage, availability, and transparency, but a significant drawback lies in its inability to verify the truthfulness of the data stored on it. State-of-the-art systems attempting to mitigate this concern often rely on conventional reputation-based approaches, which predominantly evaluate historical data from sensors and neglect the critical assessment of data in real-time. Furthermore, there is limited research on incorporating machine learning (ML) methods to enhance data trustworthiness in blockchain systems. This paper proposes a novel ML-based trust assessment approach that takes into account both historical reputation and real-time data trustworthiness. Our approach integrates multiple ML models within a blockchain framework using edge servers and validators, effectively functioning as a distributed ensemble to enhance classification accuracy, and contributing to more accurate reputation score calculations. Our results demonstrate significant accuracy gains in distinguishing trustworthy and untrustworthy IoT sensor data in blockchain networks.**

***Index Terms*—Blockchain, IoT, Smart Contracts, Machine Learning, Data Trust**

I. Introduction

Blockchain technology is increasingly being integrated into a diverse range of systems, with the Internet of Things (IoT) emerging as a key domain where its application is particularly significant. As IoT continues to expand, becoming a vital part of modern life and various industries, the vast network of interconnected devices it supports generates and exchanges massive amounts of data. Ensuring the security and trustworthiness of this data is crucial. Blockchain has emerged as a powerful alternative to traditional centralized cloud infrastructures, offering promising solutions for establishing trust in IoT applications. It has the potential to address key architectural challenges in IoT, particularly those related to centralization, data integrity, and availability. However, a critical challenge arises in ensuring the reliability of the data incorporated into the blockchain. While blockchain provides a secure and tamper-resistant environment, it does not guarantee that the data it stores is accurate or true (Fig. 1). This inherent gap gives rise to the pervasive issue known as the bad data problem in blockchain networks. This highlights the need for reliable IoT data trust evaluation in blockchain networks.

Recent research on IoT data trust assessments [1]–[3] have proposed the use of reputation systems focusing on device trust evaluation based on the device's external factors, inter-device relationships and historical contributions of data. While the historical perspective is valuable, it does not completely reflect the immediate trustworthiness of data presented to the system

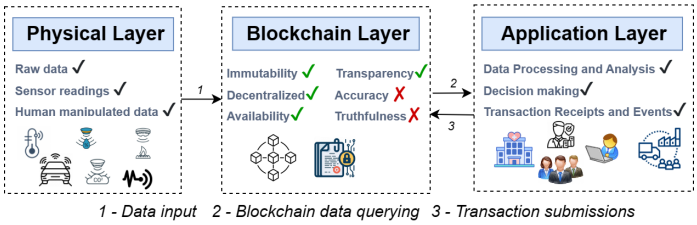

Fig. 1: Problem definition

in real-time. State-of-the-art studies [4], [5] have also explored using machine learning (ML) models for data trustworthiness classifications, but studies specifically focused on data trust assessments within blockchain networks remain limited. In this paper, we present a novel ML-based trust assessment technique considering both historical reputation and real-time data trust evaluation, and integrated into the blockchain system using edge servers and validators.

Our contribution: Our approach recognizes that assessing data exclusively on previous data contributions by the device may ignore key contextual indicators affecting current data quality. To counter this, we introduce a composite trust metric that uses not only the device's reputation over time, but also a real-time assessment method to evaluate the data quality at the time of submission. We employ multiple ML models at two different levels for calculating the composite trust score as well as the reputation score update. This improves trust evaluation accuracy and strengthens the system's resilience to dynamic data conditions, contributing to increased data quality and system reliability. In our proposed approach, blockchain not only ensures the immutable storage of data but also aids in establishing a robust reputation management system for devices through the implementation of smart contracts. A support vector machine (SVM) model is first employed at the edge server level for initial labeling of data by considering the composite score derived using SVM decision scores and reputation scores. We propose to execute the SVM model via blockchain smart contracts, mitigating the impact of potential misbehavior or malicious actions by edge servers. Utilizing a decentralized oracle platform for implementation, the smart contract can invoke the oracle API to conduct SVM model classifications on the data uploaded by the edge servers, ensuring security and reliability. Subsequently, at the blockchain level, validators implemented as oracle services apply various ML models specifically to data records identified as likely to be misclassified by the SVM model. This reduces the number of validations to be run by the validators. This comprehensive approach gives a more detailed

and responsive knowledge of data reliability, guaranteeing that the system can adapt to changing data dynamics and spot potential abnormalities or malicious inputs before they are added to the system. Our proposed approach could be especially important in use cases where the availability and dependability of data are crucial, such as healthcare, industrial applications, or the monitoring of sensor data in food supply chains.

The remainder of this paper is structured as follows: The related work is covered in Section II. Section III outlines our proposed solution, and Section IV analyzes the experimental results. Finally, Section V provides concluding remarks and directions for future work.

II. Related Work

Numerous studies in the literature have explored solutions to trust issues associated with IoT data. The use of blockchain in IoT for ensuring data provenance and immutability is a notable trend [6]–[8]. While these blockchain-based systems significantly enhance data security, they lack data trust evaluation at the point of origin. The authors in [9] have proposed a layered architecture that evaluates sensor observation trustworthiness and adapts block verification to increase end-to-end trust.

Trust modelling in IoT data has been studied in several recent studies. Data-centric and entity-centric are two predominant categories of trust attributes used in IoT data trust models. Conventional methods often focus on entity-centric trust evaluation; however, they might not provide an accurate representation of data trustworthiness. Therefore, recognizing the significance of data-centric evaluation is crucial, as emphasized by the authors in [10]. Another study [11] has presented a semi-centralized computational trust model by combining direct and indirect trust information between IoT devices engaging in data exchange.

ML techniques for bad data identification have also gained particular interest in recent research studies. ML methods have been used to assess IoT data trustworthiness in [4], [5]. These investigations highlight a common challenge: the scarcity of appropriately labeled IoT datasets for conducting ML-based analyses. To address this challenge, researchers have implemented various strategies to generate labels for training ML models. For instance, in [4], the authors employ clustering techniques to assign labels to data collected by sensors. However, this method is criticized in [5] for lacking a solid foundation that unsupervised methods like clustering can assign accurate trust labels to the data. Consequently, a novel data synthesis approach called Random Walk Infilling (RWI) has been introduced in [5] to generate untrustworthy data from trustworthy IoT sensor data, addressing the limitations of existing methodologies.

Reputation management in the context of IoT sensor networks has gained considerable attention in recent years. Several frameworks have been proposed to assess the trustworthiness of IoT sensors based on their historical behavior. In our previous study [12], we presented an adaptable framework intended to address the problem of IoT data reliability in blockchain systems. By employing a ML model to classify data and maintaining reputation scores for data sources, this approach provided a comprehensive solution to identify and reduce the amount of faulty sensor data that is entered into the blockchain system. In parallel, researchers have explored real-time data assessment techniques for ensuring IoT data quality and integrity as in [13].

In the broader context of reputation management, there are some hybrid approaches as in [10] that incorporate elements of both entity reputation and real-time data assessment. However, these approaches focus on social IoT networks and the reputation scores are generated based on trustor-trustee relationships and recommender systems rather than historical data contributions. In [12], we used evaluations of historical data contributions to update reputation scores. However, the approach used a single ML model which is executed by the edge servers as well as validators on the blockchain without any initial filtration at the edge server level.

Our research aims to bridge a gap in the existing literature by introducing a novel approach that employs ML models, edge servers and validators to identify untrustworthy IoT data on blockchains. This approach integrates historical reputation with real-time data assessment for comprehensive data trust evaluations. By integrating these dimensions, we aim to enhance the reliability of data trust evaluation, thereby reducing the risk of bad data infiltrating blockchain-based IoT systems. This approach has the potential to significantly improve the assessment of reliability in IoT data, contributing to more trustworthy and efficient IoT applications.

Table I provides a summary of our paper's contribution compared to related works.

TABLE I: Comparison of proposed solution with existing work

Characteristic	Ref. **[7]**	Ref. **[4]**	Ref. **[5]**	Ref. **[12]**	**Our Work**
IoT data trust assessment	–	✓	✓	✓	✓
Data source reputation	–	✓	–	✓	✓
Real-time data trust evaluation	–	✓	✓	✓	✓
ML models for data trust evaluations	–	✓	✓	✓	✓
Blockchain-based solution	✓	–	–	✓	✓
Combines reputation and current data trust	–	–	–	–	✓

III. Proposed Solution

In this section, we provide a comprehensive overview of our proposed approach. Fig. 2 shows a high-level illustration of how the suggested architecture would be deployed. Edge servers serve as the blockchain nodes tasked with gathering sensor data and transferring it to the blockchain. It is essential to deploy multiple edge servers to effectively gather data from sensors in distinct localities. The architecture is well-suited for applications involving IoT data sharing or storage, particularly in consortium-style blockchain deployments where multiple known entities collaborate within a shared network. The InterPlanetary File System (IPFS) is used for storing data off chain to improve storage scalability and reduce data redundancy in comparison to blockchain systems.

The solution incorporates a two-tiered data evaluation process. Initially, data undergoes assessment before its upload to the blockchain, followed by a secondary evaluation post-upload. It integrates historical data source reputation with real-time data evaluations performed by ML models, enabling comprehensive

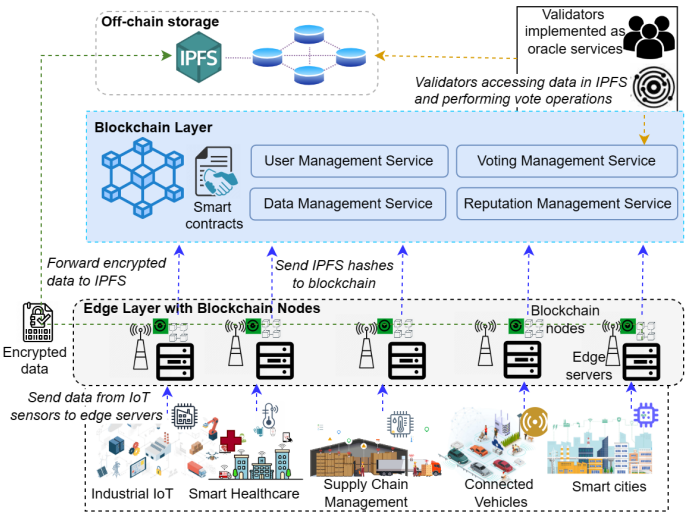

Fig. 2: Deployment of the proposed solution

assessments of data trust within blockchain systems. Moreover, it features a truly decentralized trust evaluation and reputation update mechanism on the blockchain. The process entails a series of steps illustrated in Fig. 3 and outlined below.

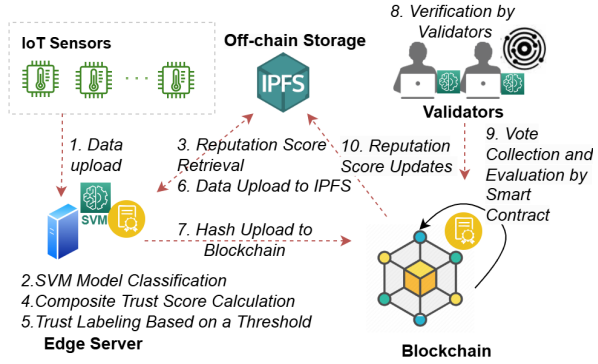

Fig. 3: Overview of the proposed solution

Step 1: Data Upload - The process initiates with IoT sensors situated within a specific neighborhood transmitting data collected in real-time to an edge server.

Step 2: SVM Model Classification - Upon data reception at the edge server, preprocessing activities encompass data cleaning, normalizing, and feature engineering to prepare the dataset for subsequent analysis. Utilizing an SVM model which is pre-trained, the data is classified into trustworthy and untrustworthy categories. To mitigate potential misconduct by edge servers, the SVM model execution can be integrated into smart contracts run on the blockchain, leveraging decentralized oracle networks for off-chain computations. Furthermore, the SVM model calculates decision scores for each data record, indicating their proximity to the decision hyperplane. These raw decision scores are then scaled using a sigmoid function to provide a more interpretable metric for trust assessment.

Step 3: Reputation Score Retrieval - The edge server obtains the previously calculated reputation scores (R_s) for corresponding sensors from the IPFS.

Algorithm 1 Calculating Composite Trust Score and Labeling Data at Edge Server

Require: D_s: Array of decision scores for test data, R_s: Reputation score for the sensor, T_h: Threshold for binary classification

Ensure: L_p: Predicted binary labels based on composite score

function WEIGHT_FUNCTION(D_s)
  $\lambda_{peak} \leftarrow 1$ ▷ Peak weight of the Gaussian curve
  $s \leftarrow 0.05$ ▷ Controls the spread of the curve
  return $\lambda \leftarrow \lambda_{peak} \cdot \exp\left(-\frac{(D_s-0.5)^2}{2 \cdot s^2}\right)$
end function
$T_h \leftarrow 0.5$
function CALCULATE_COMPOSITE_SCORE(D_s)
  $\tilde{D}_s \leftarrow \{\}$, $C_s \leftarrow \{\}$
  for d_i in D_s **do**
    $\tilde{D}_s \leftarrow \tilde{D}_s \cup \frac{1}{1+\exp(-d_i)}$ ▷ Scale decision scores
  end for
  for d_j **in** $\tilde{D}_s$ **do**
    $\lambda_j \leftarrow$ WEIGHT_FUNCTION(d_j)
    $c_j \leftarrow (1-\lambda_j) \cdot d_j + \lambda_j \cdot R_s$, $C_s \leftarrow C_s \cup c_j$
  end for
end function
function THRESHOLD_BASED_LABELING(C_s)
  $L_p \leftarrow \{\}$
  for c_j in C_s **do**
    if $c_j > T_h$ **then** $L_p \leftarrow L_p \cup 0$
    else $L_p \leftarrow L_p \cup 1$
    end if
  end for
  return L_p
end function

The main steps involved in step 4 and 5 are summarised in Algorithm 1.

Step 4: Composite Trust Score Calculation - To evaluate data trustworthiness at the edge server level, we introduce a Composite Trust Score (C_s). This score is derived by combining the scaled decision scores ($\tilde{D}_s$) obtained from the SVM classification with the reputation scores of the sensors, which are retrieved from the IPFS in the preceding step. Visualization of SVM model misclassifications reveals a concentration closer to the decision boundary, resulting in a higher occurrence of misclassifications near the mid-point (i.e., 0.5) of scaled decision scores. Thus, a weighted sum is employed, giving greater importance to reputation scores in such scenarios. This balance is achieved using a trust score weight (λ), which is modeled as a Gaussian distribution peaking at 1 for a scaled decision score of 0.5. The Composite Trust Score (C_s) is then calculated as in equation (1).

$$C_s = (1-\lambda) \cdot \tilde{D}_s + \lambda . R_s \tag{1}$$

Step 5: Trust Labeling Based on a Threshold - Data entries with composite trust scores surpassing a specified threshold (T_h) are categorized as trustworthy, while entries with lower

scores are deemed untrustworthy. Additionally, data points in near proximity to the decision hyperplane (e.g., with D_s values between 0.2 and 0.8) are identified as 'misclassification-prone,' acknowledging the inherent uncertainties near the hyperplane.

Step 6: Data Upload to IPFS - The labeled data entries and their corresponding trust evaluations are stored securely on the IPFS.

Step 7: Hash Upload to Blockchain - The corresponding IPFS hashes associated with the records are then stored on the blockchain, guaranteeing data integrity, transparency, and resistance to tampering.

Step 8: Verification by Validators - Within the blockchain, validators play a crucial role in evaluating the trustworthiness of data records identified as 'misclassification-prone.' By executing validation processes through decentralised oracle services, which leverage pretrained ML models for independent analysis, the system ensures robust security when integrating ML models into the blockchain, mitigating potential vulnerabilities associated with this integration.

Step 9: Vote Collection and Evaluation by Smart Contract - Validators' evaluations are then submitted as votes which are collected via a smart contract. The smart contract evaluates the collected votes and determines the majority vote. Depending on the majority consensus, the labels of data records are verified, and new entries are added with the correct labels in cases of discrepancies. Hence, the verification by validators acts as a distributed ensemble model, employing a majority voting mechanism for consensus. This process significantly enhances the refinement of data trustworthiness, leveraging the effectiveness of ensembles in addressing classification challenges when compared to individual ML models.

Step 10: Reputation Score Updates - In the final step, reputation scores for corresponding sensors are updated using Algorithm 2. This update incorporates combined predicted labels (L_c), which consist of SVM classifications for data records with $\tilde{D}_s$ values outside the range prone to misclassification, and the majority assessments provided by validators for the records identified as 'misclassification-prone.' The new reputation scores are securely stored on the IPFS, ensuring ongoing accessibility to edge servers for continuous data trust assessments and further enhancing the system reliability.

SVMs provide a unique measure of a sample's proximity to the decision boundary, offering us the advantage of obtaining immediate trust scores in the form of decision scores generated by the model. This approach is not only lightweight but also significantly faster than running multiple models. By leveraging SVM, we employ decision scores for an initial filtration step at the edge server, streamlining the process and alleviating the need for validators on the blockchain to validate every data record. This also facilitates quicker data availability in the system.

IV. Experiments

A. Experimental Configuration

Fig. 4 illustrates the prototype system implementation. The blockchain was developed with Hyperledger Fabric v2.4.4 using Java, employing Raft as the consensus algorithm. The prototype

Algorithm 2 Updating Reputation Scores on Blockchain

Require: L_c: Combined predicted labels from SVM and validators.
Require: N_T True predictions count up to $i-1^{th}$ iteration.
Require: N_A: All entries count up to $i-1^{th}$ iteration.
Ensure: R_n: New reputation score for the sensor.

```
function REPUTATION_SCORE_UPDATE(Lc, L_T, L_A)
    R_n ← {}
    N_A ← N_A + |L_c|        ▷ Add current entry count at i^th iteration
    L_1 = {x ∈ L_c | x = 1}, R_n ← (N_T + |L_1|) / N_A
    return R_n
end function
```

network consisted of one orderer node and two organizations, each with a peer node. ML models were implemented in Python. A client application in Java was created to simulate the interactions of the edge server with IPFS and the blockchain, as well as those of the validators with the blockchain's smart contract. The network was deployed on a server powered by an Intel Xeon CPU at 2.10 GHz, featuring 20 cores and 128 GB of RAM, running Ubuntu 20.04.

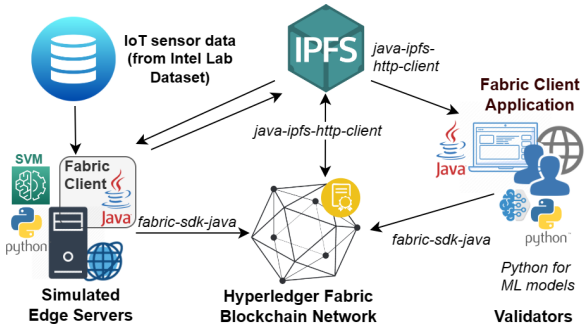

Fig. 4: Prototype implementation

In our experimentation, we used temperature data from the Intel Lab dataset [14], a comprehensive collection of real-world sensor data gathered from the Intel Berkeley Research lab. The dataset includes readings from a network of 54 sensors installed throughout the lab over the course of more than a month, capturing a variety of environmental measurements such as temperature, humidity, light, and voltage at regular intervals. In our research, this dataset was used to evaluate the performance of multiple ML models, offering a robust and realistic benchmark for model comparison. To model untrustworthy data, we applied the RWI algorithm as recommended in [5], to generate untrustworthy data corresponding to each original temperature reading in the dataset. The dataset was carefully labeled: outliers identified in the original data were marked as untrustworthy, while data without outlier characteristics were labeled as trustworthy. All data points produced by the RWI algorithm were labeled as untrustworthy. Data cleaning, normalizing, and feature engineering was done according to the guidelines in the same reference. Following feature engineering, the data was

used to train a linear SVM model with a regularization parameter of 1, which served as the model executed by the simulated edge servers.

For blockchain-based validation, three models were utilized by validators: K-Nearest Neighbors (KNN), Multilayer Perceptron (MLP), and Random Forest. The KNN model employed 3 neighbors with a distance-based weighting scheme. The MLP was configured with a single hidden layer of 30 neurons and optimized using the Adam solver, while the Random Forest used 300 trees with a maximum depth of 30. These parameters were carefully selected after experimenting with a range of values to optimize each model's performance on the dataset. Validators' votes were collected by a blockchain smart contract for each record of data being validated. The final evaluation of the data was then determined by a majority vote mechanism within the smart contract.

B. Results

The optimal model configurations were identified through systematic parameter tuning to achieve peak performance. Model behavior across varying levels of malicious activity or faults was analyzed by including different percentages of untrustworthy instances in the test set. The final accuracy values of using our method were determined considering the combined result of using SVM and threshold-based labeling using the composite trust score at the edge server level, in conjunction with the complementary models—KNN, MLP, and Random Forest—employed by blockchain validators for instances identified as 'misclassification-prone' at the edge server.

The resulting average accuracy outcomes are shown in Fig. 5, along with a comparative analysis with the study presented in [5], where we replicated their approach of using linear SVM and MLP with a single hidden layer in our experiments. These models were selected for comparison as they achieved the highest accuracy results in the referenced study. Our observations, as seen in Fig. 5, indicate that these individual models produce significantly lower accuracy values compared to the combined approach we introduce in this paper. Furthermore, [5] did not investigate the impact of varying percentages of untrustworthy data. In contrast, our results show that our proposed approach consistently achieves higher accuracy, even under conditions of increased untrustworthy data percentages.

To evaluate the blockchain-related performance characteristics of our approach under varying transaction loads, we recorded the time taken by the two distinct processes—data submission and voting—as the transaction count increased from 1,000 to 10,000 in increments of 1,000. We conducted multiple experiments varying block-time (1, 2, 3 seconds) and block-size (256 KB, 512 KB) configurations within the blockchain. The execution times for each transaction load across the two processes are illustrated in Figures 6 and 7, respectively. All blockchain configurations experience an increase in processing time as the volume of transactions increases. As clearly seen in Fig. 6, larger block sizes often result in longer completion times because a block is only generated when the block time expires or after the configured block size is reached. On the other hand,

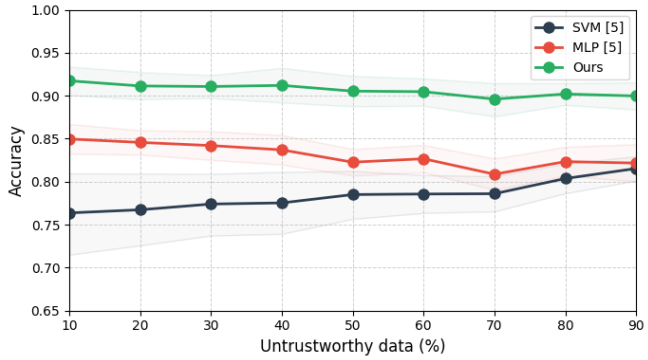

Fig. 5: Accuracy comparison with related work at different percentages of untrustworthy data

smaller block sizes facilitate quicker block creation, making them available on the blockchain sooner and thus reducing overall completion times. These figures offer valuable insights into the scalability and performance of our system under varying transaction loads, highlighting its efficiency and responsiveness in managing concurrent operations.

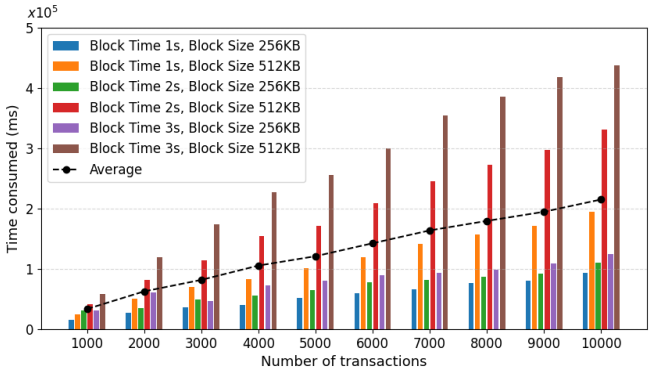

Fig. 6: Time taken for data submission process

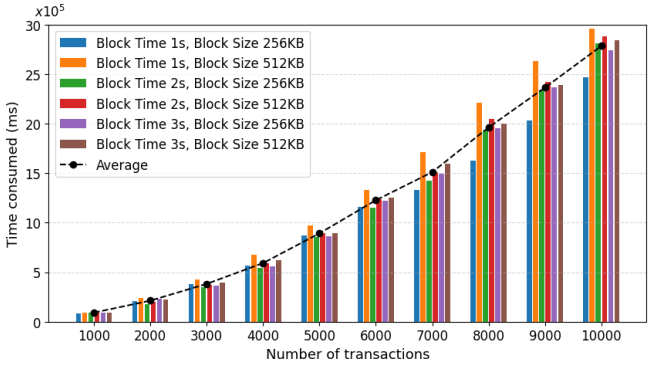

Fig. 7: Time taken for vote process

The average throughput of the system, in transactions per second (tps), for each configuration is presented in Table II. Notably, the configuration with a 1-second block time and a 256 KB block size achieved the highest throughput. Accordingly, the average end-to-end latency results shown in Table III were obtained using this optimal configuration.

This experiment evaluated the latency of the data trust evaluation process from an end user's perspective by simulating sce-

TABLE II: Average Throughput for Different Block Time and Block Size Configurations

Configuration	Throughput (tps)	
	Data Submission	Voting
1s, 256KB	95.47 ($\pm$4.74)	6.58 ($\pm$0.78)
1s, 512KB	47.67 ($\pm$1.47)	5.60 ($\pm$0.75)
2s, 256KB	74.42 ($\pm$6.27)	6.54 ($\pm$0.89)
2s, 512KB	27.77 ($\pm$0.71)	6.15 ($\pm$0.82)
3s, 256KB	63.49 ($\pm$5.88)	6.18 ($\pm$0.77)
3s, 512KB	19.37 ($\pm$0.64)	6.01 ($\pm$0.79)

TABLE III: End-to-End Latency for Different Numbers of Transactions

Transaction Count	System	End-to-End Latency (s)
10	No trust evaluation [15]	1.6190 ($\pm$0.0150)
10	With trust evaluation	4.2998 ($\pm$0.0231)
20	No trust evaluation [15]	1.7440 ($\pm$0.0122)
20	With trust evaluation	5.3767 ($\pm$0.0342)
30	No trust evaluation [15]	1.7990 ($\pm$0.0146)
30	With trust evaluation	6.4105 ($\pm$0.0516)

narios where the user submits 10, 20, and 30 transactions. Each blockchain transaction is assumed to encapsulate data from 300 readings. The validation process is conducted in batch mode, where readings in one transaction are assessed collectively rather than individually. This batching approach aligns with practical implementations where processing efficiency is enhanced by handling data in groups. The latency metrics include the time required for processing the data using the SVM model at the edge server, storing data on the blockchain, validation using ML models, and the voting process. To conduct a comparative study, a baseline implementation of the blockchain network was developed, utilizing blockchain with IPFS solely for sensor data storage, as described in [15]. The latency results for both implementations are presented in Table III.

As seen in Table III, the data trust evaluation process adds less than 5 seconds of overhead for 30 transactions. This overhead is acceptable when considering the overall time required for the final applications where the data is used. In real-world applications such as predictive maintenance in industrial IoT systems or traffic prediction in smart cities, training machine learning models typically require substantial time [16], [17]. For instance, work in [17] shows their model for highway traffic prediction has taken 1,000 to 10,000 seconds for training depending on dataset size. Therefore, the additional time incurred by the data trust evaluation process has no significant impact on these applications. Furthermore, this evaluation does not affect the availability of data on the blockchain, as the data is already stored and accessible.

V. Conclusions and Future Work

This paper introduces a novel method for evaluating data trust in blockchain-based IoT systems using ML methods. The proposed solution utilizes a composite trust metric that integrates blockchain-derived historical reputation data with real-time evaluations from SVM models, enabling more accurate trust evaluations at the time of submission. Additionally, the use of multiple machine learning models on the blockchain functions as a distributed ensemble, enhancing precision in reputation score calculations and improving overall system reliability. Our results demonstrate substantial accuracy gains in identifying trustworthy and untrustworthy records of IoT sensor data.

In future work, we plan to expand our research by exploring advanced ML techniques (e.g., reinforcement learning, federated learning) to improve the data trust evaluation accuracy and the system's responsiveness to dynamic IoT environments, as well as optimizing validation mechanisms to reduce delays and enhance overall system performance.

Acknowledgement

This work is partly supported by European Union in the CONFIDENTIAL-6G project (Grant ID. 101096435) and the Science Foundation Ireland under CONNECT phase 2 (Grant no. 13/RC/2077_P2) project.

References

[1] S. Asiri and A. Miri, "An IoT trust and reputation model based on recommender systems," in *14th Annual Conference on Privacy, Security and Trust (PST)*, 2016.

[2] H. Moudoud and S. Cherkaoui, "Empowering security and trust in 5G and beyond: A deep reinforcement learning approach," *IEEE Open Journal of the Communications Society*, vol. 4, pp. 2410–2420, 2023.

[3] H. Son, N. Kang, B. Gwak, and D. Lee, "An adaptive IoT trust estimation scheme combining interaction history and stereotypical reputation," in *14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2017.

[4] U. Jayasinghe, G. M. Lee, T.-W. Um, and Q. Shi, "Machine learning based trust computational model for IoT services," *IEEE Transactions on Sustainable Computing*, vol. 4, no. 1, pp. 39–52, 2019.

[5] T. Tadj, R. Arablouei, and V. Dedeoglu, "On evaluating IoT data trust via machine learning," *Future Internet*, vol. 15, no. 9, 2023.

[6] A. Lahbib, K. Toumi, A. Laouiti, A. Laube, and S. Martin, "Blockchain based trust management mechanism for IoT," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2019.

[7] B. Yu, J. Wright, S. Nepal, L. Zhu, J. Liu, and R. Ranjan, "IoTChain: Establishing trust in the internet of things ecosystem using blockchain," *IEEE Cloud Computing*, vol. 5, no. 4, pp. 12–23, 2018.

[8] R. Kumar and R. Sharma, "Leveraging blockchain for ensuring trust in IoT: A survey," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, Part A, pp. 8599–8622, 2022.

[9] V. Dedeoglu, R. Jurdak, G. D. Putra, A. Dorri, and S. S. Kanhere, "A trust architecture for blockchain in IoT," in *ACM Proceedings of the 16th EAI international conference on mobile and ubiquitous systems: computing, networking and services*, 2020.

[10] U. Jayasinghe, A. Otebolaku, T.-W. Um, and G. M. Lee, "Data centric trust evaluation and prediction framework for IoT," in *ITU Kaleidoscope: Challenges for a Data-Driven Society (ITU K)*, 2017.

[11] Y. Liu, C. Zhang, Y. Yan, X. Zhou, Z. Tian, and J. Zhang, "A semi-centralized trust management model based on blockchain for data exchange in IoT system," *IEEE Transactions on Services Computing*, 2023.

[12] R. Ratnayake, M. Liyanage, and L. Murphy, "Trust management and bad data reduction in internet of vehicles using blockchain and AI," in *IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, 2023.

[13] G. C. Karmakar, R. Das, and J. Kamruzzaman, "IoT sensor numerical data trust model using temporal correlation," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2573–2581, 2020.

[14] P. Bodik, W. Hong, C. Guestrin, S. Madden, M. Paskin, and R. Thibaux, "Intel lab data," 2004.

[15] M. S. Ali, K. Dolui, and F. Antonelli, "IoT data privacy via blockchains and IPFS," in *ACM Proceedings of the Seventh International Conference on the Internet of Things*, 2017.

[16] M. Züfle, J. Agne, J. Grohmann, I. Dörtoluk, and S. Kounev, "A predictive maintenance methodology: predicting the time-to-failure of machines in industry 4.0," in *2021 IEEE 19th International Conference on Industrial Informatics (INDIN)*, 2021.

[17] H. Yi and K.-H. N. Bui, "An automated hyperparameter search-based deep learning model for highway traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, 2021.