

Intellig_block: enhancing IoT security with blockchain-based adversarial machine learning protection

Walid Dhifallah^{1*}, Tarek Moulahi^{1,2}, Mounira Tarhouni¹ and Salah Zidi¹

Laboratoire Hatem Bettaher (IRESCOMATH), University of Gabes, Gabes, 6029, Tunisia¹

Department of Information Technology, College of Computer, Qassim University, Buraydah, KSA²

Received: 23-April-2023; Revised: 17-September-2023; Accepted: 20-September-2023

©2023 Walid Dhifallah et al. This is an open access article distributed under the Creative Commons Attribution (CC BY) License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

Internet of things (IoT) systems were becoming increasingly complex due to advancements in open innovation, especially in the realms of intelligent automation and artificial intelligence (AI). However, their effective deployment was impeded by security concerns and the need for enhanced threat detection capabilities. To address these challenges and bolster the security of IoT devices, an architecture called "intellig_block" was developed. This architecture seamlessly integrated blockchain (BC) and AI technology to mitigate vulnerabilities and enhance system efficiency. The goal was to harness the advantages of BC and AI to offer effective solutions for the security challenges confronting IoT systems. The primary focus centered on thwarting contamination and evasion attacks on intrusion detection systems (IDS) powered by machine learning (ML). At that time, many existing solutions relied on traditional statistical frameworks or ML techniques, resulting in increased deployment and runtime costs. In contrast, the "intellig_block" architecture hashed the template file and embedded it as a smart contract to implement the categorization algorithm. The results of the experiments conducted at that time were quite promising: the execution time was short with minimal gas overhead. A potential method was proposed at that time for effectively identifying cyber threats in ML models using the "intellig_block" architecture, which could significantly fortify IDS. Smart contracts (SC) have been introduced as a solution to safeguard IDS results against adversarial machine learning (AML) attacks within BC. In that context, IoT devices leveraged these SC to promptly detect AMLs in real-time data streams. Comprehensive performance analysis and experimental findings at that time substantiated the efficacy of the model in shielding IoT devices against unreliable services, all while maintaining cost-effectiveness within a reasonable time frame and at an affordable cost.

Keywords

Internet of things, Cyber threats, Decentralization, Blockchain, Machine learning (ML), Evasion attack.

1. Introduction

The widespread adoption of internet of things (IoT) systems in recent years has brought about a significant transformation in various industries. These systems provide ample opportunities for intelligent automation and artificial intelligence (AI) technologies. Machine learning (ML), a prominent AI approach, has shown great potential in improving the security of IoT devices and systems [1]. By efficiently analyzing large volumes of data, ML algorithms can detect anomalies, prevent fraud, and provide decision support in diverse fields such as healthcare and autonomous vehicles.

However, the growing reliance on ML for IoT security has exposed these systems to new threats posed by malicious actors.

During the process of training ML models, different types of attacks such as poisoning, and evasion can be injected to deceive the system's decision-making algorithms (as depicted in *Figure 1*). Adversaries may choose to strategically modify input samples to cause misclassifications (known as evasion attacks) [2] or contaminate the training dataset to skew the classifier model (known as poisoning attacks) [3] (as shown in *Figure 2*). These attacks can seriously compromise the overall performance and reliability of security applications based on IoT, leading to significant challenges in the successful deployment of AI-enabled smart systems.

*Author for correspondence

In the context of IoT security, ML-based technologies have emerged as a promising alternative to traditional rule-based approaches. ML offers a multifaceted AI approach that can outperform dynamic networks without requiring explicit programming. Its versatility allows for the development of sophisticated models capable of handling complex IoT data. For instance, ML can be

employed in health coverage, anomaly detection, fraud prevention, and various other applications where it learns to detect different types of assaults and deliver appropriate defensive strategies. Consequently, ML has become a valuable tool for monitoring, estimating, categorizing, and tracking future IoT activity.

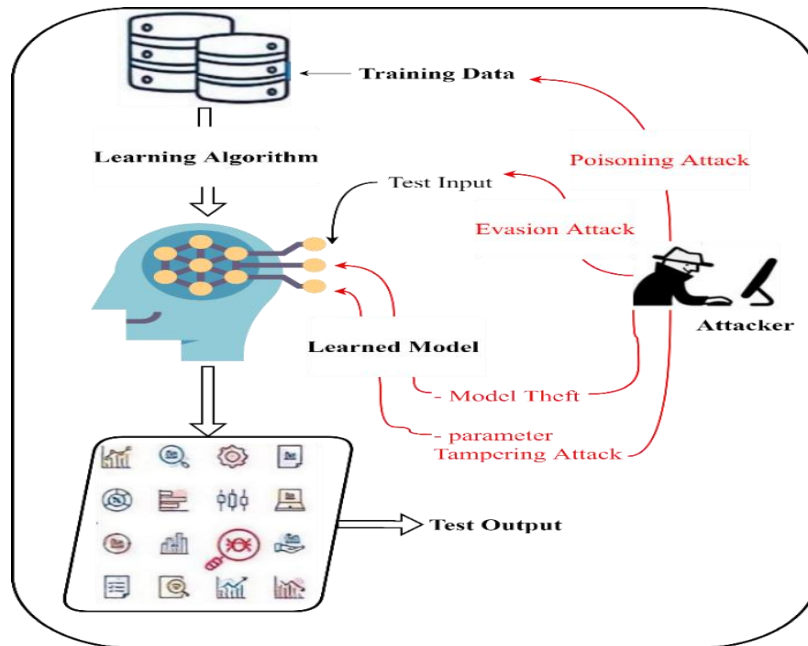


Figure 1 Attack zones in the implementation of AI

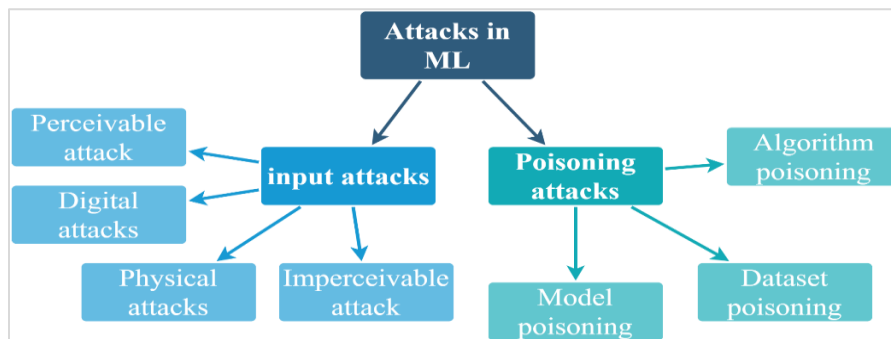


Figure 2 Taxonomy of attack properties on ML

The use of ML in securing IoT devices comes with its own set of difficulties. The procedure of ML model development, which includes data pre-processing, model training, and testing, can lead to potential security vulnerabilities (Figure 3). Adversaries may inject poisoning and evasion attacks during these stages, compromising the ML model's integrity and causing misclassification of IoT activities [4]. Current research on AI technologies for

IoT systems mainly focuses on ML and reinforcement learning (RL) [5]. While ML has demonstrated its potential in analyzing vast amounts of data and providing valuable insights, it has also become an attractive target for attackers who want to exploit these models for malicious purposes. Consequently, there is an urgent need to address these challenges to ensure that ML-based IoT security systems operate reliably and securely [6].

The suggested solution involves running the ML fitting procedure locally, off-chain. This is because ML fitting may be a time-consuming and expensive computing procedure that can also burden the blockchain (BC) network.

The model's parameters can be retrieved and kept on-chain after the ML model has been fitted off-chain. As a result, just the crucial data about the ML model must be stored on the BC, rather not the complete dataset or the model itself.

In the tests of our research, we use a variety of classifiers, including support vector machines (SVM), k-nearest neighbors (k-NN), random forest (RF), decision tree (DT), logistic regression (LR), naive bayes (NB), and multi-layer perceptron (MLP). In the field of ML, these classifiers are well known. They choose an ideal hyperplane using influence functions, enabling the pertinent separation of two categories within the data. As a member of the family of feedback neural networks, MLP stands out as a deep learning model. To enable advanced prediction conclusions, this method iteratively extracts characteristics and insights from the data using a stratified arrangement of nodes that are stacked on many levels. The convergence of SVM, k-NN, RF, DT, LR, NB, and MLP in this study provides an in-depth and versatile investigation, combining traditional approaches and deep learning methodologies for a notable improvement in predictive accuracy and the generalization of models. The suggested strategy is a promising one for applying ML to a BC. It can assist in enhancing the scalability and privacy of ML applications while concurrently decreasing the computational and networking costs associated with ML.

The two classifiers indicated in the previous sentence are further described in the following manner:

SVM: Useful for classification or regression applications, this kind of supervised learning technique. To best partition the data points into two or more classes, SVMs find a hyperplane. SVM is a supervised ML algorithm that can be used for both classification and regression tasks. It works by finding a hyperplane that best separates the data points into two or more classes. In this case, the SVM model achieved an accuracy of 98% on the test set, which means that it correctly classified of the data points.

RF: An ensemble of DT is used where each tree is trained on a random subset of the data, and their results are combined to improve performance and

robustness [2]. In your case, the RF model achieved an accuracy of 97% on the test set, which means that it correctly classified of the data points.

LR: A linear combination of features is used to estimate the likelihood of belonging to a class in the process of classification [3].

Equation 1 is used to calculate the probability using LR.

$$h_{\theta}(x) = \sigma(\theta^T X) \quad (1)$$

It works by fitting a logistic curve to the data to predict the probability of a data point belonging to a particular class. In this case, the LR model achieved an accuracy of 99.2% on the test set, which is the lowest accuracy of all the models.

k-NN: K-NN is a non-parametric ML algorithm that works by finding the k most similar data points to a new data point and then predicting the label of the new data point based on the labels of the k nearest neighbors [4]. The Euclidean distance to calculate an object's distance from its neighbors, as indicated in Equation 2, to classify unlabeled observations into the category of the most similar labeled cases.

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2)$$

In this case, the K-NN model achieved an accuracy of 97% on the test set, which is the same as the SVM model.

DT: DT are a type of supervised machine-learning algorithm that can be used for both classification and regression tasks. The classification technique builds a tree representing judgments based on characteristics from root to leaf [5].

In this case, the DT model achieved an accuracy of 99.8% on the test set, which is slightly lower than the SVM and K-NN models.

NB: A classification method based on Bayes theorem assumes independence of all characteristics and is effective in some circumstances [6].

It works by assuming that the features of the data are independent of each other. In this case, the Naive Bayes model achieved an accuracy of 99.4% on the test set, which is slightly lower than the other models.

MLP: MLP is a type of deep learning algorithm that can be utilized for classification or regression tasks. MLPs consist of multiple layers of nodes, where each layer is connected to the subsequent layer [7]. It works by learning the weights of the connections between the neurons in the network to minimize the error between the predicted and actual values. In this case, the MLP model achieved an accuracy of 99.6% on the test set, which is slightly lower than the SVM, K-NN, and DT models.

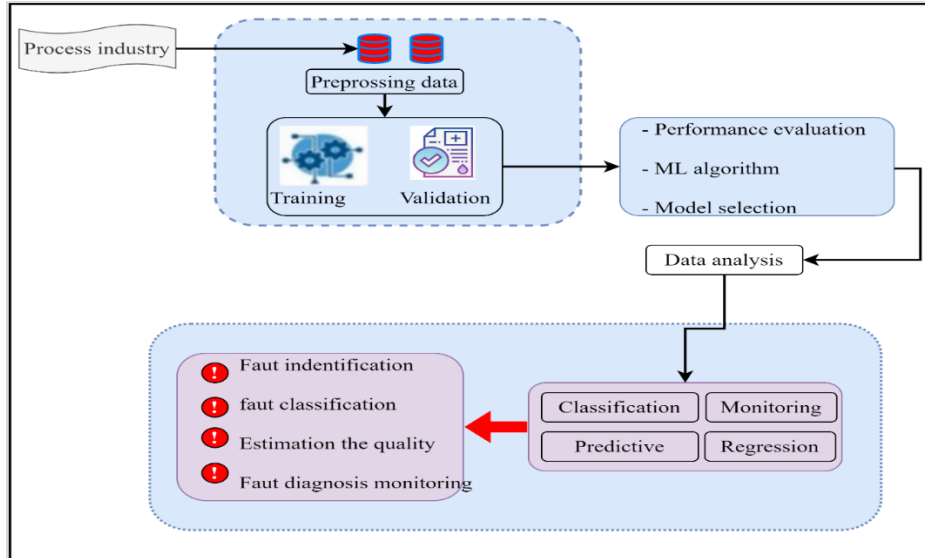


Figure 3 Process of detecting faults using ML

The challenges and vulnerabilities in ML-based IoT security systems are the focus of this research. The objective is to create a strong and secure framework that can protect ML and deep learning models from cyber threats, specifically poisoning and evasion attacks. The proposed framework, called “intellig_block,” aims to enhance security levels and confidence in detecting IoT system threats by utilizing the power of BC technology [8].

The contributions of this paper are as follows:

- Novel framework development: Introducing the intellig_block framework – a new and innovative approach that combines ML model hashing with smart contracts (SC) on the BC. This integration of ML and BC aims to decentralize the classification technique, reduce vulnerabilities, and enhance the overall security of IoT systems.
- Privacy and security enhancement: Our framework combines SC and access control mechanisms to enhance the security and privacy of ML models. Additionally, these benefits extend to intrusion detection systems (IDS)[9]. This comprehensive approach safeguards the ML-based IDS from potential attacks and intrusions.
- Empirical evaluation: Our experimental results showcase the effectiveness of the proposed intellig_block framework. They demonstrate the framework’s ability to achieve low execution time and minimal gas overhead. As a result, the framework is a viable and efficient solution for securing ML-based IoT systems.
- Performance comparison: We evaluated the intellig_block framework’s performance against existing approaches, examining accuracy,

deployment, and execution overhead. This comparison enabled us to assess the framework’s effectiveness and identify areas for improvement.

The paper was structured as follows: In section 2, a detailed review of the related work was presented, highlighting the importance of BC technology in addressing security challenges in IoT domains. This led to the introduction of our proposed system in section 3. Afterwards, section 4 provided the simulation results of the intellig_block framework, offering a comprehensive evaluation of its performance. Discussion has been elaborated in section 5. Finally, in section 6, the conclusion of this study was presented, summarizing the contributions and outlining potential avenues for future research.

2.Related work

The public study topics of the intelligent decentralized systems on IoT have been invested in and debated by several scholars. *Table 1* summarizes the several works presented in this section.

Banerjee et al. [10] investigated IoT security mechanisms, as well as the paucity of IoT datasets, used between academic and professional organizations. Wright et al. [11] this paper describes a streamlined resource management framework using Ethereum for smart edge BC networks, enabling secure and verified outsourcing of computations between devices for financial compensation. Swan in [12] explored the advantages of utilizing BC for cognitive progress and its architectural implications in the realm of intelligence advancement. Qian et al.

[13] demonstrated high BC-based safety control strategies for connected systems. Nevertheless, there are worries in this study about erroneous traffic control and proof of identity. Rathore et al. [14] suggested BC-driven private deep learning for trustworthy IoT data in line with intelligent decentralized device-level systems. Shinde et al. [15] in the context of securing AI and open innovations, the authors advocate BC's potential in ensuring data privacy, preventing data poisoning, and upholding AI model integrity, while acknowledging the technology's novelty and ongoing security concerns. Abdel-basset et al. [16] suggested a federated learning (FL) method for preserving privacy while learning from non-independent and identically distributed data in fog assisted IoT, employing secure aggregation from fog nodes, albeit potentially sacrificing accuracy compared to centralized approaches. Kumar et al. [17] provided an overview of BC's industrial IoT applications, outlining advantages and challenges, while noting the technology's novelty and the absence of standards and interoperability. Liu et al. [18] presented an exploration of BC-driven FL, spotlighting advantages, and challenges, while highlighting the technology's nascent stage and the absence of universal standards and interoperability. Rehman et al. [19] proposed a secure healthcare 5.0 system integration BC and FL for data storage and ML model training, while acknowledging potential scalability limitations in extensive healthcare setups. Sun et al. [20] suggested a BC-based audit method for encrypted data in FL, utilizing BC for data and audit result storage, while acknowledging scalability challenges in expansive FL setups. Chen et al. [21] presented an overview of security challenges in BC systems, covering attack types and defense mechanisms, while noting the potential outdated nature of some discussed attacks and defenses. Barbaria et al. [22] introduced an innovative BC-based architectural model for healthcare data integrity, leveraging BC for data storage and integrity assurance, yet potentially limited in managing substantial data volumes. Miao et al. [23] puts forth a data sharing plan for BC-driven IoT, utilizing BC for data storage and a privacy-preserving protocol for authorized sharing, despite potential computational costs associated with the privacy protocol. Yaacoub et al. [24] examined security hurdles in IoT-based FL, highlighting challenges like data privacy, poisoning, model theft, and Sybil attacks, while proposing solutions that might not scale efficiently for extensive IoT setups. Sáez-de-Cámara et al. [25] introduced a clustered FL architecture to enhance network anomaly detection in vast, diverse IoT

networks, aiming for accuracy and efficiency gains, albeit potentially involving intricate implementation and deployment. Mirdula and Roopa [26] suggested a manufacturer usage description enabled deep learning framework for smart building anomaly detection, leveraging multi-sensor data, yet potentially lacking in rare event anomaly detection. Habiba et al. [27] suggested an edge intelligence approach for IoT network intrusion prevention, utilizing edge devices for data analysis, but possibly falling short in countering advanced attacks on these edge devices. Taloba et al. [28] put forward a hybrid BC platform for IoT-healthcare multimedia processing, utilizing BC for data storage and security, alongside a hybrid processing approach, yet potentially constrained in scalability for expansive systems. Singh and Singh [29] delivered an assessment of varied IoT access management methods, weighing pros and cons, and introducing a BC-backed decentralized authentication approach, which might not be universally compatible with all current IoT devices. Alsuqaih et al. [30] introduced a privacy-focused control method for electronic health (e-health) apps, utilizing BC for patient records and employing homomorphic encryption for data privacy, though the encryption scheme could incur computational costs. Xi et al. [31]. The article examines BC's role in secure medical data sharing, exploring benefits and limitations, and highlighting future research challenges in the realm of privacy and security. Zhang et al. [32] the article introduces a BC-powered framework for secure IoT data sharing, employing both BC and encryption methods to ensure the privacy and security of data exchanged among IoT devices. Zhao et al. [33] proposed a BC-centric approach to protect privacy in FL, ensuring data security and integrity during cross-device model training without raw data sharing. Bhan et al. [34] suggested a BC-infused solution for healthcare data sharing security, employing BC and encryption to safeguard privacy and security during medical data exchange among providers. Lou et al. [35] introduced a BC-powered privacy-preserving framework for edge computing, leveraging BC to safeguard data privacy in shared edge device environments. Rafique et al. [36] presented a BC-enhanced security and privacy framework for IoT, utilizing both BC and encryption methods to safeguard data security and privacy among IoT devices. Kumar et al. [37] suggested a BC-centered method for secure and private ML, employing BC to safeguard training data privacy. Taloba et al. [38] suggested a BC-infused framework for secure healthcare data management, employing BC and encryption to ensure privacy and

security of medical data within healthcare organizations. Li et al. [39] suggested a BC-driven system for secure edge computing data sharing, leveraging BC and encryption to safeguard privacy during edge device data exchange. Karaszewski et al. [40] examined BC’s role in public sector data sharing, exploring its benefits, limitations, and future research challenges. Kamath et al. [41] in suggested a BC-driven framework for secure data sharing in supply chain management. They combined BC and encryption for privacy and security among partners. Wang et al. [42] suggested BC for private data sharing among social science researchers. In Chi et

al. [43] suggested a BC-powered solution for secure energy sector data sharing by combining BC and encryption to ensure privacy and security of data exchanged between providers and consumers. In Jiang et al. [44] suggested a BC-driven system for secure data sharing in finance, using BC and encryption for privacy among institutions. BC enhances security, privacy, and efficiency through tamper-proof ledgers, data encryption, and streamlined data sharing. Challenges include technical complexity, evolving regulations, and lack of universal standards.

Table 1 Summarized literature review

Article	Objective	BC			Performance		
		Type	Consensus algorithm	SC	Security	Privacy	Trust
[15]	Improve the protection of AI data	Private	Proof of stake (PoS)	●	-	-	●
[16]	Improve the privacy of ML	Federated	Proof of work (PoW)		●	●	
[17]	Improve the security and efficiency of the industrial IoT (IIoT)	Public	PoW		●	-	●
[18]	Improve the security and privacy of FL	Private	PoS	●	●	●	-
[19]	Improve the security and privacy of the 5.0 healthcare system	Private	PoS	●	-	●	●
[20]	Improve the security and privacy of encrypted data	Private	PoS	●	-	●	-
[21]	Improve the security of BC	Public	PoW			●	
[22]	Improve the integrity of healthcare data	Private	PoS	●	●	-	●
[23]	Improve the security and privacy of data sharing for BC-empowered IoT	Private	PoS	●	●	●	-
[24]	Improve the security and privacy of FL	Public	PoW	●	●	●	
[25]	Improve the efficiency of anomaly detection	Federated	PoW	-	●	-	●
[26]	Improve the accuracy of DL	MUD	PoW	-	●	-	-
[27]	Improve the intrusion detection	Edge intelligence	PoW		-	●	-
[28]	Improve the efficiency of multimedia data processing in IoT-Healthcare	Public	PoW	-	●	-	●
[29]	Improve the security of IoT device access management	Private	PoS	●	●	●	-
[30]	Improve the privacy of privacy-preserving control mechanisms for healthcare applications	Private	PoS	●	●	-	-
This approach	Securing IoT with Intellig_block: A BC-based Defense against adversarial machine learning (AML)	Prive/Public	PoW	●	●	●	●

3.Methods

The proposed concept is based on the use of BC technology and ML to enhance the security of AI-based systems. BC-based apps can be made smarter

by leveraging ML capabilities. The security of distributed ledgers can be improved by employing ML techniques. Additionally, the decentralized design of BC technology can help improve ML

models. We have provided an architecture for adopting ML in intelligent applications, as shown in *Figure 4*.

The proposed architecture aims to create a decentralized and intelligent system that can defend against various security threats. By integrating ML into BC-based applications, the system can leverage the benefits of both technologies. The architecture consists of four main levels, illustrated in *Figure 5*.

Data source: At this level, the system generates and collects data from various sources, such as IoT sensors, IoT applications, or cloud-based services. These sources can produce diverse types of data, ranging from structured data (e.g., numerical data stored in databases) to unstructured data (e.g., text files, images, or videos).

Model training: The collected data is then used to train the model. Model training is a critical stage where ML algorithms learn from the data to perform specific tasks, such as intrusion detection, data classification, or future behavior prediction. During model training, various techniques, such as supervised, unsupervised, and RL, may be employed based on the nature of the data and the system's objectives.

It is important to note that during training, models may be vulnerable to attacks, such as evasion attacks and poisoning attacks. These attacks aim to degrade the model's performance by manipulating the training data, which can compromise the security and reliability of the system.

At level 3 of the architecture, the Intelligent Secure Decentralized Model incorporates SC to ensure the security and decentralization of ML models. These contracts are self-executing programs that run automatically when triggered by specific events. In this context, they are used to embed the decision function of the ML algorithms.

The use of SC enhances the security of the model by governing access to sensitive information and the model's decision-making functions. This ensures that only authorized individuals with the appropriate access rights can interact with the model, thereby reducing the risk of malicious attacks. Furthermore, the decentralized structure of the BC enhances the

security of the system as there is no central point that can be compromised. The models are distributed across multiple nodes in the BC, making them more resilient and less susceptible to attacks.

After the model has been trained and secured through an intelligent contract, the results obtained by the model are displayed, and its performance is evaluated to assess its effectiveness and accuracy in detecting cyber threats.

To ensure data integrity and security, the model files are stored on the BC as encrypted pieces. When a user requests a model file, the pieces are encrypted, and subsequently defragmented to restore the original file.

To ensure that a file is authentic and trustworthy, the user can use the BC to compare the file's hash with the calculated hash of the original file. This process provides a genuine and trustworthy way of verifying the integrity of model files, allowing for transparent and accountable interactions. The proposed architecture is an innovative and promising approach to enhance the security of AI-based applications by utilizing BC technology and ML. By integrating SC into the architecture, access to the ML models is governed, minimizing the risk of unauthorized access and potential attacks. Furthermore, the encryption and verification process for model files stored on the BC enhances data integrity and ensures the reliability of the ML models.

The architecture concept seems to be well thought-out, but more detailed explanations are required to understand how it would be practically implemented and operated. Specifically, clarifications are needed on how the ML models are integrated into SC, the details of access control mechanisms, and how the hash comparison process is facilitated. This will help to improve the clarity of the concept.

Additionally, it would be helpful to discuss the potential limitations and challenges of the proposed architecture. For example, addressing issues related to scalability, gas costs on the BC, and the trade-offs between security and performance in decentralized systems would provide a more comprehensive evaluation of the architecture.

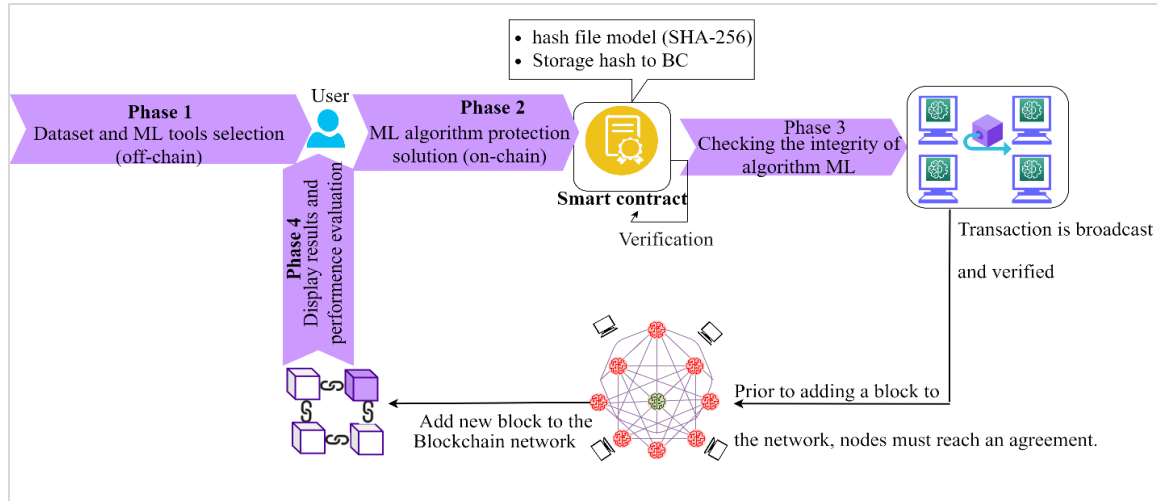


Figure 4 The design of IoT intelligent block architecture

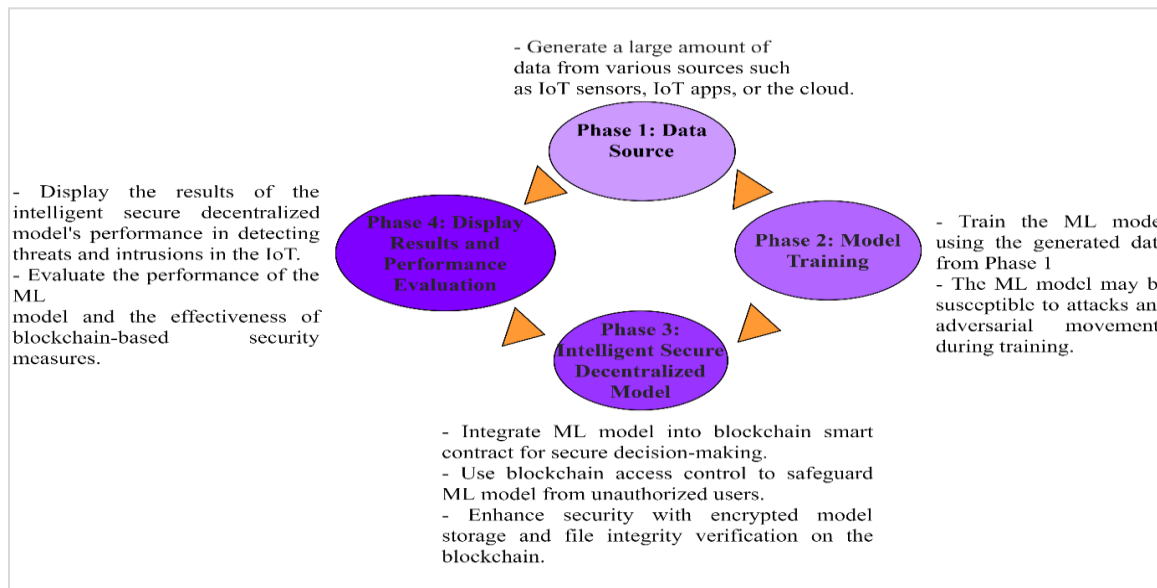


Figure 5 Proposed architecture mechanism flowchart for each phase

The ML model's working in the proposed approach for IoT intrusion detection involves the following steps:

Data collection: The ML model collects data from various sources, such as IoT sensors, applications, and cloud-based services. This data includes network traffic attributes, system logs, user behaviors, and relevant information to identify potential threats.

Data pre-processing: The data is pre-processed through cleaning, normalization, and feature extraction to prepare it for training and analysis.

Model training: The pre-processed data is used to train the ML model using various algorithms like SVM, RF, or K-NN. The model's parameters are

optimized during training to achieve accurate and dependable results.

Model evaluation: It is important to assess the performance and generalization capabilities of a ML model after it has been trained. This is done using a separate set of data that was not used during training, known as testing data. The model's ability to distinguish between normal and anomalous activities is measured during this evaluation using metrics such as precision, recall, accuracy, and F1-score. This evaluation is crucial in determining whether the model is suitable for its intended purpose.

Deployment and real-time monitoring: Once trained and evaluated, the ML model is deployed in

the intelligent secure decentralized model (Level 3). It continuously monitors real-time data from the IoT network. As new data arrives, the model predicts whether observed activity is normal or suspicious.

Smart contract integration: The ML model's decision function is integrated into SC, ensuring secure execution and governance. SC control access to the model, allowing only authorized entities to interact with it.

Intrusion detection and alert generation: When the ML model detects anomalous or malicious behavior, it alerts system administrators or security personnel, enabling prompt action to mitigate potential threats.

Advantages of the ML model and approach:

Real-time detection: The ML model allows for real-time threat detection and response, enabling quick action to prevent security breaches.

Adaptability: ML models are well-suited for dynamic and ever-changing IoT environments due to their ability to adapt to evolving cyber threats and new attack patterns.

Decentralization: Integrating the ML model into a decentralized BC network enhances system security and resilience to attacks.

Transparency and accountability: The use of BC technology enhances transparency and accountability as all activities related to the model are recorded on the BC, providing a transparent audit trail.

The integration of ML, BC technology, and SC provides the IDS with strong security features and effective threat detection capabilities in IoT systems. This makes it a dependable solution for safeguarding the security and authenticity of interconnected devices and applications.

BC algorithm

Algorithm 1: Smart contract algorithm

- **Input:** model weights, model name
- **Output:** model address
- 1. Create a smart contract with two properties:
 - **model weights**
 - **model name**
- 2. Define a constructor that takes the model weights and model name as input and stores the model weights and model name in the contract properties.
- 3. Define a function **getWeights()** that returns the model weights.
- 4. Define a function **getName()** that returns the model name.

4.Results

Intellig_block is a secure ML system that combines BC technology and ML. Its objective is to safeguard ML algorithms against IoT threats. To achieve this goal, intellig_block uses a collaborative ML paradigm that supports data collection and privacy breaches. In addition, it applies collaborative ML in a BC context to create a secure and reliable model against ML threats such as evasion and poisoning attacks. Finally, an intelligent block prototype model was created to test its effectiveness in real-world scenarios.

AML attacks aim to deceive a ML model, resulting in incorrect decisions. There are two main types of AML attacks: poison attacks and evasion attacks. Poisoning attacks refer to the act of adding malicious data to a ML model's training set. This harmful data can be intentionally designed to make the model biased against certain groups of individuals or result in incorrect decisions in specific situations.

Evasion attacks, on the other hand, involve modifying legitimate data to cause it to be misclassified by a ML model. These changes can be very subtle and challenging for humans to detect.

To test ML models against AML attacks, there are various approaches. One common method is to use artificial datasets that have been created to simulate AML attacks. Alternatively, real datasets that have been compromised by AML attacks can be employed.

It is also essential to configure ML models in a way that makes them more resistant to AML attacks. This may involve using techniques like data normalization, data strengthening, and attribute selection.

Test Scenario 1: A ML model can be manipulated by attackers who add malicious phishing emails to the model's training set, causing it to classify phishing emails as legitimate.

Test Scenario 2: A ML model is used to detect financial fraud. The attacker modifies legitimate transactions so that they are misclassified by the model. The model will then be more likely not to detect financial fraud.

Attack configuration 1: An ML model is utilized to categorize images of individuals. However, an attacker can manipulate the images in such a way that the model misclassifies them. This can lead to the model being more prone to misclassify people based on their race, gender, or age.

Attack configuration 2: A ML model is used to translate languages. The attacker modifies the translations so that they are mistranslated by the model. The model will then be more likely to produce incorrect translations.

The method used in the experiment relied on several underlying programs. We used Ganache and MetaMask tools to create a private Ethereum BC, which allowed us to test our Solidity contracts.

To develop an intelligent contract, we simulated the BC using the Ethereum-based Remix integrated development environment and Solidity. We tested our hypothesis using TON_IoT [45], which is a collection of datasets for testing the fidelity and efficiency of various AI-based cybersecurity solutions. These datasets include IoT and IIoT device monitoring datasets, Windows 7 and 10 OS datasets, Ubuntu 14 and 18 TLS datasets, and congestion datasets. Because of their diverse sources, these

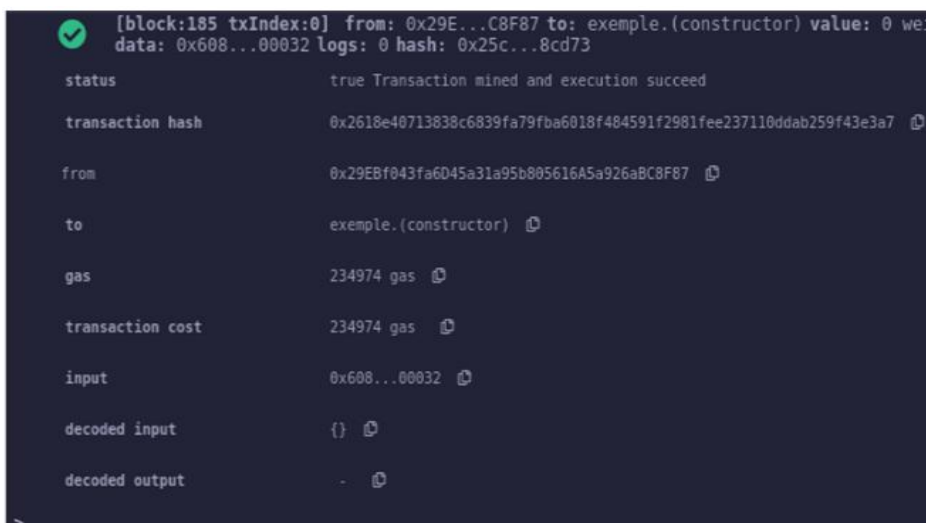
datasets are known as “ToN_IoT.” They are used for intrusion detection, malware detection, advanced analytics and privacy preservation models.

The data for the TON_IoT dataset was obtained from a realistic large-scale testing ground network that was created at the university of New South Wales Canberra cyber IoT lab. This network connected a range of virtualization and physical devices, hacking systems, cloud and fog platforms, and IoT devices, simulating the complexity and flexibility of IIoT and Industry 4.0 networks.

After the administrator calculates the hash of the ML methods, it is stored in a decentralized system. A classification approach was used to achieve the objectives of the study. *Figure 6* shows the examples of BC-based logs.

Our SC are implemented using a private BC called Ganache. This allows us to create Dapps and run tests. Ganache provides ten Ethereum accounts, each with a balance of 100 ether, along with a graphical user interface for tracking all network activity.

Figures 7 to 9 illustrate the process of compiling and migrating SC to the Ganache BC. The cost of one file for this transaction, as shown in the result of the SVM migration, is 0.00050368 ethereum (ETH) (equivalent to £0.65 as of January 27, 2023). After transferring our smart contract, we will build a local virtual server containing the client-side application using the Truffle framework. To join our BC network, we need to connect to our MetaMask wallet.



```
[block:185 txIndex:0] from: 0x29E...C8F87 to: exemple.(constructor) value: 0 wei
data: 0x608...00032 Logs: 0 hash: 0x25c...8cd73

status      true Transaction mined and execution succeed
transaction hash  0x2618e40713838c6839fa79fba6018f484591f2981fee237110ddb259f43e3a7
from        0x29EBf043fa6045a31a95b005616A5a926aBC8F87
to          exemple.(constructor)
gas         234974 gas
transaction cost 234974 gas
input       0x608...00032
decoded input  {}
decoded output -
```

Figure 6 Journal of the Blocks created in a decentralized intelligent system

The Ethereum BC evaluates the proposed platform based on the cost incurred by SC. These costs are measured in units of gas required to complete transactions and perform smart contract tasks, which include both execution and transaction costs. Transaction costs refer to the fees for adding smart contract code to the Ethereum BC, which are limited by the size of the smart contract. The size of the contract is determined by the fundamental actions it performs. Execution costs, on the other hand, refer to the cost of storing global variables and invoking smart contract methods and are influenced by the calculations performed during transaction execution. To determine the gas charge for each transaction in our system Equation 3 has been used.

$$\text{Transaction Fee} = \text{Gas Used} \times \text{Gas Price} \quad (3)$$

Gas price refers to the amount of Gwei required for a transaction, while gas used represents the amount of gas consumed based on the amount stored and processed for each transaction. To illustrate, let's take the add user method as an example, which allows for the assignment of roles and accounts to agents. In Figure 7, the gas used and gas price for this method are 84,275 and 20 Gwei, respectively. Therefore, Equation 4 can be formulated as:

$$\begin{aligned} \text{Transaction Fee} &= 84,275 \times 20 = 1,685,500 \\ \text{Gwei} &= 0,00000000163 \text{ ETH} \end{aligned} \quad (4)$$

According to the findings, the execution of k-NN and SVM transactions consumes a significant amount of gas, as shown in Figure 9. This indicates that these transactions involve a large amount of data and require a significant number of resources compared to other methods. Gas represents the amount of resources required to complete a transaction, and it incurs a cost. In Table 1, the total gas consumption was calculated by multiplying the gas used by the cost of gas.

The information gathered has been summarized in Table 2, which shows the amount of feedback received related to the ML model and time spent on each transaction. The average time is the duration required to complete a funds transfer. It is noticeable that each feedback takes approximately the same amount of time. The data is presented in a tabular format for better understanding. In Figure 10, it is shown that the MD5 hash function is faster at calculating the speed of an ML model as compared to other functions. This is because it is simpler and has a 128-bit size. However, it is also less secure when compared to SHA256 or SHA512. These two options are slower but more secure, making them suitable for high-security applications. The choice between them depends on the specific needs of the application.

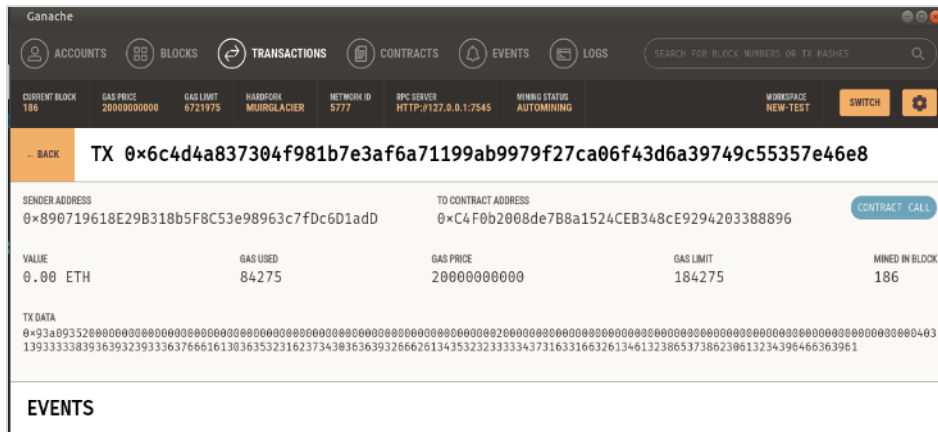


Figure 7 Registration smart contract deployed in Ganache

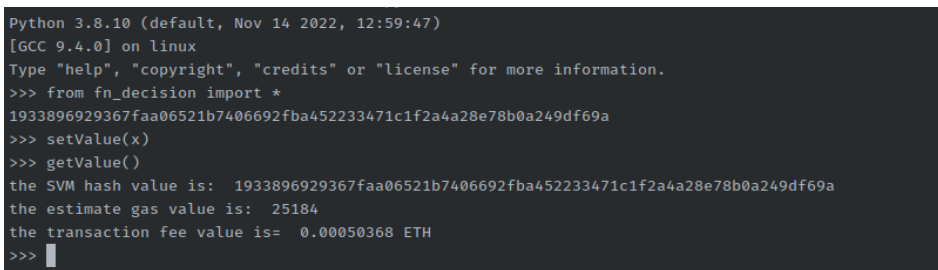


Figure 8 Smart contract deployment

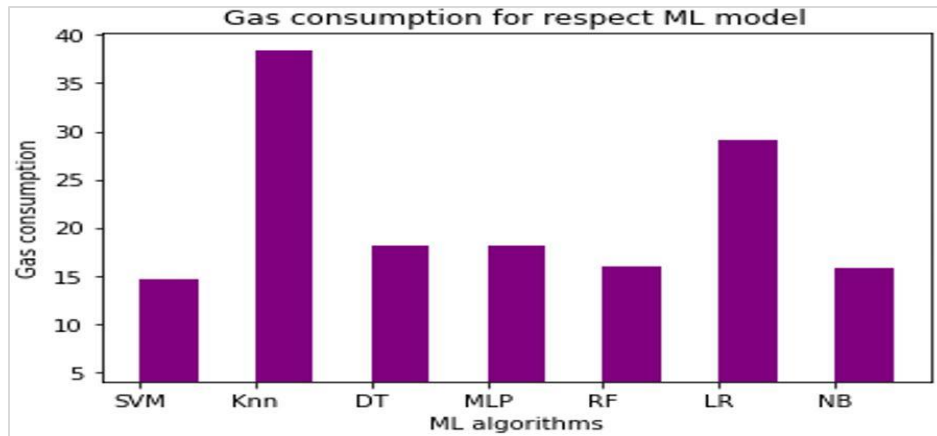


Figure 9 Gas consumption for respect ML model

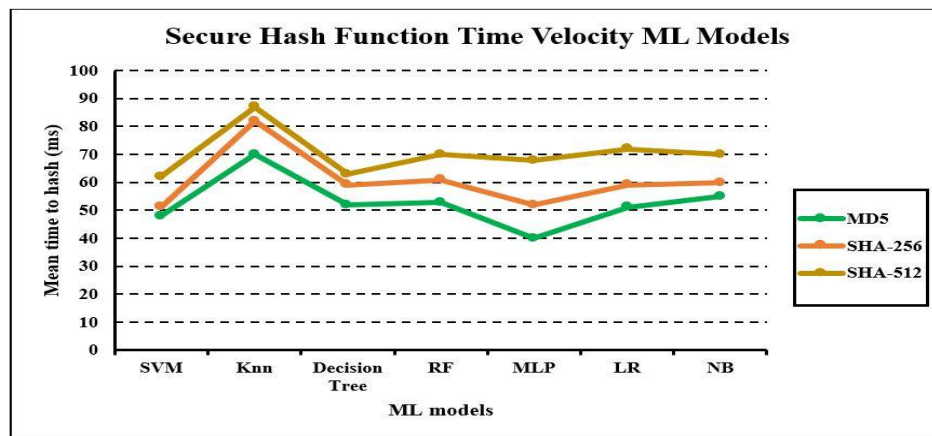


Figure 10 Secure hash function time velocity ML models

When accessing the same service, the IoT may rate the same cloud server multiple times. The Ganache interface calculates the average transaction time in Table 3. The findings show that K-NN takes longer compared to other types of input. Figure 11 displays how the bindings were analyzed.

Subsequently, we compared our framework’s results with those of a previous study using the same dataset. In intellig_block, MLP has an accuracy of 92.2%, while SVM has an accuracy of 98%. This comparison (Table 4) is based on some criteria deemed important such as BC-based, access control and security.

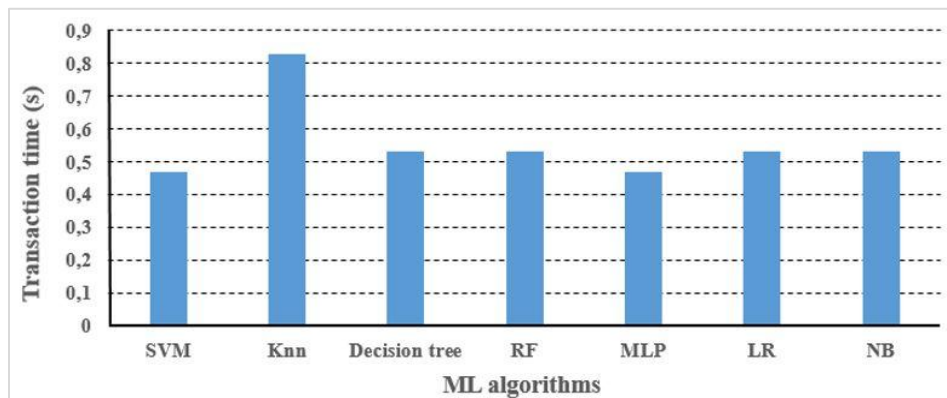


Figure 11 Average ML model of time transaction/seconds

Table 2 Consumption of gas against the ML classifier

ML classifier	Gas fee	Gas used	Gas consumption
SVM	0.00050	22308	10.616
k-NN	0.00083	41506	34.449
DT	0.00053	26506	14.048
RF	0.00053	26506	14.048
MLP	0.00047	25458	11.965
LR	0.00053	47210	25.021
NB	0.00053	22308	11.823

Table 3 Time transaction ML algorithm

ML classifier	Times(s)
SVM	0.47
K-NN	0.83
DT	0.53
RF	0.53
MLP	0.47
LR	0.53
NB	0.53

Table 4 Comparison the proposed system with related work

Parameter ID	[18]	[19]	[20]	[21]	[22]	Our system
Pr1	√	X	√	X	√	√
Pr2	X	X	√	√	√	√
Pr3	√	X	X	X	√	√
Pr4	√	X	√	X	X	√
Pr5	X	X	X	X	X	√

Pr 1: BC-based; Pr 2: Access Control-based;
 Pr 3: Security; Pr 4: Integrity; Pr 5: Multi-agent sys
 X: Not supported; √: supported.

5. Discussion

The proposed framework that merges BC with ML and deep learning has significant value in protecting IoT systems from cyber threats. By integrating BC with ML models, this framework can provide a higher level of security and trust in detecting system threats. This approach addresses the limitations of traditional security methods. The addition of a smart contract, which includes the hashing of ML models, further enhances the security of the classification technique. Decentralizing the classification technique in this way makes it less vulnerable to cyber threats. Additionally, by verifying the ML model through a smart contract before execution, this framework provides an additional layer of security.

Finally, the integration of SC and access control helps ensure the security and privacy of the ML model. By limiting access to only authorize parties, the privacy and security of the ML model can be

maintained. This approach can be particularly useful when implementing IDS. Validating the proposed approach, the fourth contribution presents experimental results that demonstrate the effectiveness of the proposed framework. The low execution time and gas consumed indicate that the framework is feasible for practical deployment. The experiments conducted to test the intellig_block framework revealed that it is highly efficient in detecting threats in IoT systems. The framework was able to effectively identify potential security breaches and intrusions with accuracy rates of 92.2% for MLP and 98% for SVM. By utilizing the power of ML algorithms and BC technology, intellig_block offers a robust solution to safeguard IoT systems against cyber threats.

Collaborative ML in a BC context: The combination of collaborative ML and BC presents a novel solution to address data collection and privacy challenges in the IoT domain. The decentralized nature of BC ensures that data is securely shared and validated across the network, which helps to mitigate the risks of privacy breaches. With collaborative ML, participants can collectively contribute to the model's training process without compromising the integrity of their data. This collaborative paradigm enhances trust among the participants and fosters a secure environment for sharing sensitive information.

Enhanced security through SC: The intellig_block framework's security is significantly enhanced by integrating ML model hashing as SC on the BC. SC are useful for verifying ML models before their execution, ensuring that only authorized and valid models are used for threat detection. The hashed representation of the ML models is stored on the BC, making it possible to detect any attempts at tampering or injecting malicious code into the models. As a result, the overall security and reliability of the framework is strengthened.

Access control for ML security and privacy: Intellig_block incorporates access control mechanisms that provide an additional layer of security and privacy for the ML models. With access control, only authorized entities can access and execute the models, preventing unauthorized access to sensitive information and safeguarding the confidentiality of the models. By managing access rights carefully, the framework ensures that the threat detection process's integrity and accuracy is maintained, and potential insider threats are minimized.

Experimentation and validation of the framework:

The `intellig_block` framework has been validated using the `TON_IoT` dataset, and experiments conducted on a private Ethereum network confirm its real-world efficacy. The framework has been subjected to practical scenarios, and the results demonstrate that it is feasible for deployment in real IoT environments. The low execution time and overhead, as well as the accuracy rates achieved, validate the practicality and efficiency of the `intellig_block` framework. These results position `intellig_block` as a promising solution to enhance the security and reliability of IoT-based IDS.

Additionally, the comparison of the proposed method with an existing one in terms of accuracy, deployment, and execution overhead is significant. This comparison helps evaluate the effectiveness of the proposed framework and understand how it can be improved. By identifying the strengths and weaknesses of the proposed method in comparison to existing methods, researchers can refine and optimize the proposed approach further. Overall, the contributions of this paper can advance the field of IoT security by providing a novel framework that leverages the strengths of BC and ML to protect against cyber threats. While there may be limitations and challenges to the proposed approach, the contributions of this paper are significant and can guide future research in this area. A complete list of abbreviations is shown in *Appendix I*.

Limitation

While the proposed framework has several strengths and contributions, there are also some limitations and potential challenges that should be considered. Some potential limitations of the proposed framework include:

- **Scalability:** The scalability of the proposed framework may be restricted due to the significant computational and storage demands of BC. As more participants and transactions join the network, the overhead costs of maintaining the BC network may become too expensive.
- **Security risks:** While the use of BC can certainly enhance the security of ML models, it is still not completely invulnerable to attacks. The distributed nature of the BC network makes it challenging to identify and counteract any malicious attacks. Additionally, the incorporation of additional layers into the system, such as SC and access control, can introduce new security vulnerabilities that need to be addressed.
- **Data privacy:** The proposed framework may not

provide sufficient data privacy guarantees, as sensitive data may be exposed to unauthorized parties during the execution of the SC. This issue may be particularly problematic in applications where data confidentiality is critical.

- **Integration complexity:** Integrating ML models with BC and SC can be a challenging task, which may require considerable development effort. This complexity can result in higher deployment and maintenance costs for the proposed framework, making it less appealing to organizations with limited resources.
- **Evaluation and benchmarking:** While the proposed framework is effective through experimental results, it's important to conduct a more comprehensive evaluation and benchmarking to validate the proposed approach's scalability and efficiency compared to existing methods.

6. Conclusion and future work

The purpose of this paper is to introduce a trustworthy service provisioning method based on BC to detect assaults on ML algorithms. The same principle can be used to satisfy an organization's demand for data security, where a private BC requires authentication from multiple senior authorities before reaching an agreement. Verifying the various general agreements for the ML model, considering processing speed and time, are all important considerations for the development and exploration of data blocks that might be a crucial analysis. Solidity program code was used to create SC and execute them on the Ethereum network. The results of our simulations show that our system model can help protect IoT devices from malicious ML assaults. In future work, we plan to conduct an `intellig_block` evaluation trial with a case study to detect an attack and prove its feasibility and compatibility in IoT.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

Author's contribution statement

Walid Dhifallah: Conceptualization, investigation, data curation, writing – original draft, writing – review and editing. **Tarek Moulahi:** Data collection, conceptualization, writing – original draft, supervision, analysis and interpretation of results. **Mounira Tarhouni and Salah Zidi:** Study conception, design, supervision, investigation on challenges and draft manuscript preparation.

References

- [1] Karie NM, Sahri NM, Yang W, Valli C, KEBANDE VR. A review of security standards and frameworks for IoT-based smart environments. *IEEE Access*. 2021; 9:121975-95.
- [2] Buczak AL, Guven E. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*. 2015; 18(2):1153-76.
- [3] Saleem TJ, Chishti MA. Exploring the applications of machine learning in healthcare. *International Journal of Sensors Wireless Communications and Control*. 2020; 10(4):458-72.
- [4] Hautamaki V, Karkkainen I, Franti P. Outlier detection using k-nearest neighbour graph. In *proceedings of the 17th international conference on pattern recognition, 2004*. 2004 (pp. 430-3). IEEE.
- [5] Dina AS, Manivannan D. Intrusion detection based on machine learning techniques in computer networks. *Internet of Things*. 2021; 16:100462.
- [6] Couderc N, Reichenbach C, Söderberg E. Performance analysis with Bayesian inference. In *IEEE/ACM 45th international conference on software engineering: new ideas and emerging results 2023* (pp. 112-6). IEEE.
- [7] Madhwaran M, Deepa SN. Comparative analysis on hidden neurons estimation in multi layer perceptron neural networks for wind speed forecasting. *Artificial Intelligence Review*. 2017; 48:449-71.
- [8] Alajlan R, Alhumam N, Frikha M. Cybersecurity for blockchain-based IoT systems: a review. *Applied Sciences*. 2023; 13(13):1-26.
- [9] Li W, Wang Y, Li J, Au MH. Toward a blockchain-based framework for challenge-based collaborative intrusion detection. *International Journal of Information Security*. 2021; 20:127-39.
- [10] Banerjee M, Lee J, Choo KK. A blockchain future for internet of things security: a position paper. *Digital Communications and Networks*. 2018; 4(3):149-60.
- [11] Wright KL, Martinez M, Chadha U, Krishnamachari B. SmartEdge: a smart contract for edge computing. In *IEEE international conference on internet of things (things) and IEEE green computing and communications (greencom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (smartdata) 2018* (pp. 1685-90). IEEE.
- [12] Swan M. Blockchain thinking: the brain as a decentralized autonomous corporation [commentary]. *IEEE Technology and Society Magazine*. 2015; 34(4):41-52.
- [13] Qian Y, Jiang Y, Chen J, Zhang Y, Song J, Zhou M, et al. Towards decentralized IoT security enhancement: a blockchain approach. *Computers & Electrical Engineering*. 2018; 72:266-73.
- [14] Rathore S, Pan Y, Park JH. BlockDeepNet: a blockchain-based secure deep learning for IoT network. *Sustainability*. 2019; 11(14):1-15.
- [15] Shinde R, Patil S, Kotecha K, Ruikar K. Blockchain for securing ai applications and open innovations. *Journal of Open Innovation: Technology, Market, and Complexity*. 2021; 7(3):1-37.
- [16] Abdel-basset M, Hawash H, Moustafa N, Razzak I, Abd EM. Privacy-preserved learning from non-iid data in fog-assisted IoT: a federated learning approach. *Digital Communications and Networks*. 2022;2-22.
- [17] Kumar RL, Khan F, Kadry S, Rho S. A survey on blockchain for industrial internet of things. *Alexandria Engineering Journal*. 2022; 61(8):6001-22.
- [18] Liu K, Yan Z, Liang X, Kantola R, Hu C. A survey on blockchain-enabled federated learning and its prospects with digital twin. *Digital Communications and Networks*. 2022:1-26.
- [19] Rehman A, Abbas S, Khan MA, Ghazal TM, Adnan KM, Mosavi A. A secure healthcare 5.0 system based on blockchain technology entangled with federated learning technique. *Computers in Biology and Medicine*. 2022; 150:106019.
- [20] Sun Z, Wan J, Yin L, Cao Z, Luo T, Wang B. A blockchain-based audit approach for encrypted data in federated learning. *Digital Communications and Networks*. 2022; 8(5):614-24.
- [21] Chen Y, Chen H, Zhang Y, Han M, Siddula M, Cai Z. A survey on blockchain systems: attacks, defenses, and privacy preservation. *High-Confidence Computing*. 2022; 2(2):100048.
- [22] Barbaria S, Mahjoubi H, Rahmouni HB. A novel blockchain-based architectural modal for healthcare data integrity: Covid19 screening laboratory use-case. *Procedia Computer Science*. 2023; 219:1436-43.
- [23] Miao Q, Lin H, Hu J, Wang X. An intelligent and privacy-enhanced data sharing strategy for blockchain-empowered internet of things. *Digital Communications and Networks*. 2022; 8(5):636-43.
- [24] Yaacoub JP, Noura HN, Salman O. Security of federated learning with IoT systems: Issues, limitations, challenges, and solutions. *Internet of Things and Cyber-Physical Systems*. 2023; 3:155-79.
- [25] Sáez-de-cámara X, Flores JL, Arellano C, Urbieta A, Zurutuza U. Clustered federated learning architecture for network anomaly detection in large scale heterogeneous IoT networks. *Computers & Security*. 2023; 131:103299.
- [26] Mirdula S, Roopa M. MUD enabled deep learning framework for anomaly detection in IoT integrated smart building. *e-Prime-Advances in Electrical Engineering, Electronics and Energy*. 2023; 5:100186.
- [27] Habiba M, Islam MR, Muyeen SM, Ali AS. Edge intelligence for network intrusion prevention in IoT ecosystem. *Computers and Electrical Engineering*. 2023; 108:108727.
- [28] Taloba AI, Elhadad A, Rayan A, Abd ERM, Salem M, Alzahrani AA, et al. A blockchain-based hybrid platform for multimedia data processing in IoT-Healthcare. *Alexandria Engineering Journal*. 2023; 65:263-74.
- [29] Singh I, Singh B. Access management of IoT devices using access control mechanism and decentralized

- authentication: a review. *Measurement: Sensors*. 2022; 100591.
- [30] Alsuaqih HN, Hamdan W, Elmessiry H, Abulkasim H. An efficient privacy-preserving control mechanism based on blockchain for E-health applications. *Alexandria Engineering Journal*. 2023; 73:159-72.
- [31] Xi P, Zhang X, Wang L, Liu W, Peng S. A review of blockchain-based secure sharing of healthcare data. *Applied Sciences*. 2022; 12(15):1-12.
- [32] Zhang J, Yang Y, Liu X, Ma J. An efficient blockchain-based hierarchical data sharing for healthcare internet of things. *IEEE Transactions on Industrial Informatics*. 2022; 18(10):7139-50.
- [33] Zhao Y, Zhao J, Jiang L, Tan R, Niyato D, Li Z, et al. Privacy-preserving blockchain-based federated learning for IoT devices. *IEEE Internet of Things Journal*. 2020; 8(3):1817-29.
- [34] Bhan R, Pamula R, Faruki P, Gajrani J. Blockchain-enabled secure and efficient data sharing scheme for trust management in healthcare smartphone network. *The Journal of Supercomputing*. 2023:1-42.
- [35] Lou JT, Bhat SA, Huang NF. Blockchain-based privacy-preserving data-sharing framework using proxy re-encryption scheme and interplanetary file system. *Peer-to-Peer Networking and Applications*. 2023:1-23.
- [36] Rafique W, Khan M, Khan S, Ally JS. SecureMed: a blockchain-based privacy-preserving framework for internet of medical things. *Wireless Communications and Mobile Computing*. 2023; 2023:1-14.
- [37] Kumar P, Kumar R, Srivastava G, Gupta GP, Tripathi R, Gadekallu TR, et al. PPSF: a privacy-preserving and secure framework using blockchain-based machine-learning for IoT-driven smart cities. *IEEE Transactions on Network Science and Engineering*. 2021; 8(3):2326-41.
- [38] Taloba AI, Rayan A, Elhadad A, Abozeid A, Shahin OR, Abd ERM. A framework for secure healthcare data management using blockchain technology. *International Journal of Advanced Computer Science and Applications*. 2021; 12(12):639-46.
- [39] Li T, Wang H, He D, Yu J. Blockchain-based privacy-preserving and rewarding private data sharing for IoT. *IEEE Internet of Things Journal*. 2022; 9(16):15138-49.
- [40] Karaszewski R, Modrzyński P, Modrzyńska J. The use of blockchain technology in public sector entities management: an example of security and energy efficiency in cloud computing data processing. *Energies*. 2021; 14(7):1-19.
- [41] Kamath V, Lahari Y, Mohanchandra K. Blockchain based framework for secure data sharing of medicine supply chain in health care system. *International Journal of Artificial Intelligence*. 2022; 9(1):32-8.
- [42] Wang Y, Che T, Zhao X, Zhou T, Zhang K, Hu X. A blockchain-based privacy information security sharing scheme in industrial internet of things. *Sensors*. 2022; 22(9):1-20.
- [43] Chi J, Li Y, Huang J, Liu J, Jin Y, Chen C, et al. A secure and efficient data sharing scheme based on blockchain in industrial Internet of Things. *Journal of Network and Computer Applications*. 2020; 167:102710.
- [44] Jiang S, Cao J, Wu H, Chen K, Liu X. Privacy-preserving and efficient data sharing for blockchain-based intelligent transportation systems. *Information Sciences*. 2023; 635:72-85.
- [45] Alsaedi A, Moustafa N, Tari Z, Mahmood A, Anwar A. TON_IoT telemetry dataset: a new generation dataset of IoT and IIoT for data-driven intrusion detection systems. *IEEE Access*. 2020; 8:165130-50.



Walid Dhifallah is pursuing a doctorate in electrical engineering with the University of Gabes, National School of Engineers, Tunisia. In 2018, she obtained the M.R. degree from the Higher Institute of Applied Sciences and Technologies of the University of Gafsa, Tunisia. His current research interests include IoT Systems, Artificial Intelligence, Machine Learning, Deep Learning, Blockchain, and Federated Learning.
Email: dhifallah.walid@gmail.com



Tarek Moulahi received the joint Ph.D. degree from the University of Franche-Comté, Besançon, France, in March 2015, and the Sfax National School of Engineering, Tunisia. He is currently an Assistant Professor with Mathematics and Computer Science Departments, Faculty of Science and Technology of Sidi Bouzid (FSTSB), University of Kairouan, Tunisia, and the Department of Information Technology, College of Computer, Qassim University, Saudi Arabia. His research interests include Wireless Sensor Networks, Vehicular Ad Hoc Networks (VANET) and the Internet of Things (IoT). He received the 2019 IEEE Sensors Council Sensors Journal Best Paper Runner-Up Award.
Email: t.moulahi@qu.edu.sa



Mounira Tarhouni received the engineer's degree in Electric from Engineering national engineering school-Tunisia in 2007 and the Ph.D. degree in electrical engineering from Faculty of sciences-Tunisia. She has been with the higher institute of informatics and multimedia, Gabes, Tunisia, since 2014. Her research interests include Artificial Intelligence, Machine Learning, Deep Learning and Blockchain.
Email: mounira.tarhouni@isimg.tn



Salah Zidi received the Ph.D. degree from the University of Lille, France, with a focus on regulation and reconfiguration of multimodal transportation systems, in July 2007, and the HDR degree from the University of Lille1, France, in 2017. He is currently an Assistant Professor with the MIS Department, College of Business and Economics, Qassim University, Saudi Arabia, and an Associate Professor with the University of Gabes, Tunisia. His research interests include Optimization, Artificial Intelligence, Machine Learning, Feature Extraction, and Data Analysis for Automation Systems and Complex Systems. He received the 2019 IEEE Sensors Council Sensors Journal Best Paper Runner-Up Award.
Email: salah_zidi@yahoo.fr

Appendix I

S. No.	Abbreviation	Description
1	AI	Artificial Intelligence
2	AML	Adversarial Machine Learning
3	BC	Blockchain
4	DPoS	Delegated Proof of Stake
5	DT	Decision Tree
6	E-health	Electronic Health
7	FL	Federated Learning
8	IDS	Intrusion Detection System
9	IoT	Internet of Things
10	IIoT	Industrial Internet of Things
11	k-NN	k-Nearest Neighbors
12	LR	Logistic Regression
13	ML	Machine Learning
14	MLP	Multi-layer Perceptron
15	N/A	Not Applicable
16	NB	Naive Bayes
17	NID	Network Intrusion Detection
18	PoA	Proof of Authority
19	PoS	Proof of Stake
20	PoW	Proof of Work
21	RF	Random Forest
22	RL	Reinforcement Learning
23	SC	Smart Contracts
24	SVM	Support Vector Machine