

Article

Situ-Oracle: A Learning-Based Situation Analysis Framework for Blockchain-Based IoT Systems [†]

Hongyi Bian , Wensheng Zhang * and Carl K. Chang 

The Department of Computer Science, Iowa State University, Ames, IA 50011, USA; hobian@iastate.edu (H.B.); chang@iastate.edu (C.K.C.)

* Correspondence: wzhang@iastate.edu

[†] This article is a revised and expanded version of a paper entitled: Hongyi Bian, et al. Situ-Oracle: A Learning-Based Situation Analysis Servicing Framework for BIoT Systems, which was presented at the 2023 IEEE International Conference on Software Services Engineering, Chicago, IL, USA, 2–8 July 2023.

Abstract: The decentralized nature of blockchain enables data traceability, transparency, and immutability as complementary security features to the existing Internet of Things (IoT) systems. These Blockchain-based IoT (BIoT) systems aim to mitigate security risks such as malicious control, data leakage, and dishonesty often found in traditional cloud-based, vendor-specific IoT networks. As we steadily advance into the era of situation-aware IoT, the use of machine learning (ML) techniques has become essential for synthesizing situations based on sensory contexts. However, the challenge to integrate learning-based situation awareness with BIoT systems restricts the full potential of such integration. This is primarily due to the conflicts between the deterministic nature of smart contracts and the non-deterministic nature of machine learning, as well as the high costs of conducting machine learning on blockchain. To address the challenge, we propose a framework named Situ-Oracle. With the framework, a computation oracle of the blockchain ecosystem is leveraged to provide situation analysis as a service, based on Recurrent Neural Network (RNN)-based learning models tailored for the Situ model, and specifically designed smart contracts are deployed as intermediary communication channels between the IoT devices and the computation oracle. We used smart homes as a case study to demonstrate the framework design. Subsequently, system-wide evaluations were conducted over a physically constructed BIoT system. The results indicate that the proposed framework achieves better situation analysis accuracy (above 95%) and improves gas consumption as well as network throughput and latency when compared to baseline systems (on-chain learning or off-chain model verification). Overall, the paper presents a promising approach for improving situation analysis for BIoT systems, with potential applications in various domains such as smart homes, healthcare, and industrial automation.

Keywords: IoT; context-aware; situation-aware; sensor network; blockchain; smart contract; smart home; smart city



Citation: Bian, H.; Zhang, W.; Chang, C.K. Situ-Oracle: A Learning-Based Situation Analysis Framework for Blockchain-Based IoT Systems. *Blockchains* **2024**, *2*, 173–194. <https://doi.org/10.3390/blockchains2020009>

Academic Editors: Keke Gai and Liehuang Zhu

Received: 20 April 2024

Revised: 12 May 2024

Accepted: 19 May 2024

Published: 22 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The fast-growing Internet of Things (IoT) has become more perception-diverse, resource-efficient, and intelligent [1]. This further enables situation awareness in decision-making by the edge IoT devices, such as Activities of Daily Living (ADL) pattern recognition in smart home applications [2]. It is worth noting that situation analysis is non-trivial and often requires complex synthesizing of the perceived values [3]. For example, in a smart home system, predicting that the inhabitant intends to go to bed would require a sequence of sensor inputs to be evaluated during the runtime. Given a sensor sequence of {living-room-light-off, bedroom-carpet-stepped-on, bedroom-light-off, bedroom-lamp-off, bed-pressure-sensed}, the system could infer that the inhabitant is going to sleep based on the empirical pattern of event memories and the underlying logic. Nevertheless, the sensor

sequence to infer a situation varies between different activities and inhabitants, e.g., an elderly inhabitant typically has quite different routines compared to a teenager, even for the same activity to be recognized.

On the other hand, traditional cloud-based, vendor-specific IoT devices provide means to achieve personalized services in a centralized fashion. However, if the central server or the devices are malicious or compromised, the system will be vulnerable to data leakage, malicious control, single-point failure of data storage, and other safety, security, and privacy-related risks [4]. Thus, assuring the reliability, validity, integrity, and traceability of control/data has become one of the most important but challenging aspects of IoT smart systems. With the emergence of blockchain technology, Blockchain-based IoT (BLoT) systems have been widely studied in recent years with the goal of implementing a reliable smart IoT system formed by trust-less devices in a decentralized manner. More specifically, smart contracts initiated by the Ethereum blockchain [5] have become the key driving force of the BLoT applications due to their programmability.

Although blockchain and smart contracts could potentially help enhance the security of IoT systems, how to integrate them is yet to be further studied. Currently, rule-based smart contracts deployed on a peer-to-peer (P2P) IoT network are often used for actuator triggering. A typical example is described in [6], where the smart contract takes as input temperature readings and executes air conditioning triggering statements with a set of hard-coded temperature thresholds. This approach takes advantage of smart contracts to address the potential risk concerns, but it has limitations in supporting situation awareness in a dynamic environment due to its naïve if-else-based decision-making logic. In general, existing BLoT approaches that implement rule-based decision-making with direct sensor reading as inputs often face the trade-off between functional (situation awareness) and non-functional (security, safety, privacy) expectations.

In this study, we propose “Situ-Oracle”, a framework that aims to provide situation analysis as a service to the BLoT systems. The framework enables more adaptive, fine-grained decision-making while preserving the advantages of the integration between blockchain and IoT technologies. Our key contributions can be summarized as follows:

1. Given the temporal nature of sensory sequences, we developed Recurrent Neural Network (RNN)-based learning models grounded in the theoretical framework of the Situ model [7–9]. This work is the first to offer a detailed formulation and design of three major RNN architectures—Vanilla RNN, Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRUs)—specifically within the context of the Situ framework. We then evaluated and compared these models against several non-learning-based models using an ADL dataset collected from human participants. Our aim is to demonstrate that learning-based approaches are both essential and necessary for achieving situation awareness.
2. We propose a novel framework, “Situ-Oracle”, that leverages a computation oracle component [10] in the blockchain ecosystem to provide situation analysis as a service to the BLoT infrastructure. Specifically, we detail the design of the computation oracle node and describe the process of training and deploying RNN-based models for situation prediction within the blockchain network.
3. Additionally, we highlight the design of two smart contracts, named *Context* and *Situ*. Unlike traditional contract designs, which often utilize threshold-based actuation logic or on-chain machine learning algorithms, these contracts are specifically designed to function as intermediary communication channels between IoT devices and the computational oracle node. This innovative approach aims to mitigate issues related to network consumption and costs.
4. We conduct an in-depth analysis of two pre-existing BLoT system architectures, which aim to integrate Artificial Intelligence (AI) and Machine Learning (ML) methods into BLoT systems. This paper details the design and implementation and utilizes them as baseline systems for assessing the feasibility and potential limitations inherent in integrating AI/ML with BLoT systems.

5. We evaluate our proposed framework with the baseline systems, examining them from various aspects, such as the accuracy of situation prediction, resource consumption, system throughput, and the latency encountered in service updates. The experimentation setup for our study is grounded in a private blockchain network, which operates using a Proof-of-Authority (PoA) consensus protocol on a collection of physical devices, i.e., Raspberry Pi (RPi) single-chip computers, to demonstrate the effectiveness and applicability of our framework.

In the remainder of the paper, Section 2 reviews the related works. Section 3 presents the system model and the baseline designs. Section 4 elaborates on the formulation of the learning-based situation analysis model and its integration methodology to a BIoT system. Experimentation and comparative evaluations are discussed in Section 5. The paper concludes in Section 6, where we summarize our findings and discuss potential paths for future research. A preliminary version of this work, which includes an initial study of the framework and preliminary results, is available at [11].

2. Related Works

This section reviews the related works on BIoT systems, the integration of AI/ML and BIoT systems, as well as situation awareness and the Situ framework.

2.1. BIoT: Integration of Blockchain and IoT Systems

When integrated with blockchain, an IoT system can enhance its reliability and security with a thoughtfully designed mechanism for device management and access regulation. For example, Hao et al. [12] proposed a blockchain-based architecture to facilitate intelligent and autonomous access control for IoT devices, where a consortium blockchain is used to securely store policies and provide robust authentication services for data access control. Additionally, it incorporates a trust evaluation mechanism for nodes requesting access based on a token accumulation strategy. Liu et al. [13] proposed a blockchain-based semi-centralized trust management system architecture, where a blockchain is employed to manage and calculate trust ratings for IoT devices. Yin et al. [14] proposed SmartDID for self-sovereign identities with enhanced privacy. It utilizes a dual-credential model using commitment schemes and zero-knowledge proofs to protect sensitive attributes and on-chain identity data, preventing unauthorized linkage of credentials. The anonymity of BIoT data sharing was studied by Wu et al. [15], who proposed BA-DS that utilizes an unconditional linkable ring signature and Signature of Knowledge (SoK) to achieve user data anonymity. Secured IoT resource sharing and identity management in a 6G-enabled environment were studied by Putra et al. [16], who used blockchain to regulate resource sharing between owners and users to preserve the users' privacy. A secure, intelligent fuzzy blockchain framework was proposed in [17], where the authors investigated BIoT attacks and tackled cyber threats to BIoT systems using fuzzy matching to validate transactions for fraud detection. Similarly, ref. [18] proposes a model using Generative Adversarial Networks (GAN) and RNN for cyber threat hunting in the Ethereum blockchain. This helps to mitigate potential Sybil attacks that often occur in BIoT systems.

The security features in BIoT systems have led to their application in a wide range of application scenarios. Various studies [19–29] have explored how BIoT systems enable self-regulated automation and facilitate decentralized peer-to-peer (P2P) trading marketplaces in smart homes, smart cities, and smart energy grids. In the context of the Internet of Vehicles (IoV) and vehicle-to-vehicle (V2V) communication and coordination, numerous studies [30–33] have explored how BIoT offers trusted communications and data exchanges, therefore contributing to safer and more efficient vehicular networks. Digital health is another sector where BIoT systems can provide significant benefits, particularly in ensuring privacy and creating tamper-proof digital health records [34–38]. These applications demonstrate the versatility and potential of BIoT systems across various industries and use cases, showcasing their adaptability and effectiveness in addressing security needs and

challenges. In general, these studies can be further enhanced with situation analysis to provide more fine-grained decision-making.

2.2. Integration of AI/ML and BIoT Systems

As interest in AI and ML increases, integrating intelligence into BIoT systems has also become a significant area of research. This mainly involves incorporating advanced AI and ML techniques into BIoT systems to enhance their functionality and efficiency. For example, Zhao et al. [39] proposed a learning-based consensus protocol named PoEM (Proof of Evolutionary Model), where ML models are trained iteratively using system synthesis logs, and the models are subsequently employed to select leader nodes in the blockchain consensus process. A similar approach for prioritized transaction consensus was proposed by Sanghami et al. [19], who considered the BIoT systems (e.g., used in smart cities) where emergent events should be handled quickly. An ML-based consensus scheme was designed to prioritize transaction packaging during runtime, achieving efficient data recording and fast event response. Furthermore, deep reinforcement learning (DRL) models have been employed to enhance overall performance by intelligently adjusting the consensus process in accordance with the insights gained from system attributes and parameters. The DRLPB scheme was proposed by Gadiraju et al. [40] to dynamically optimize the parameters of the Prism blockchain. Similarly, a DRL-based resource orchestration scheme for BIoT systems, particularly the edge servers, was proposed by Wang et al. [41], where dynamic resource allocation decisions are made based on changes in the network's state to improve the resource allocation efficiency and overall performance.

Most of the current approaches incorporate ML models into the blockchain consensus process by distributing off-chain model training and deployment at each trusted entity within the network. This method can result in high communication overhead due to the complex training and deployment processes. Additionally, it potentially increases the attack surface, as multiple nodes in the network are involved in handling and executing these ML models. As an alternative, Badruddoja et al. [42,43] proposed to conduct model training and prediction based on on-chain smart contracts. However, the efficiency and cost of this approach also pose significant limitations in practice. Instead, in this work, we study the potential method of creating a situation analysis oracle service that aims to mitigate the ML model handling fatigue while allowing the key operations to stay on-chain to ensure security.

2.3. Situation Awareness and Situ Framework

Situation awareness (SA) was introduced as a critical foundation for efficient and accurate decision-making in ubiquitous and pervasive computing systems [44]. SA-enabled smart systems emphasize three ascending levels, namely perception, comprehension, and projection, of the environment where they are deployed [45]. Many comprehensive modeling works for SA emerged to consider objective synthesized information such as weather, location, system synthesis logs, etc., for understanding the context of the deployed environment [46–52]. Nevertheless, among all the factors, human intentions play a key role in the evolvability and interactivity of smart systems. This means that these applications should not only respond to sensor data but also consider human inputs and decisions in real time to fully understand and hence react to different situations [53]. For instance, in the context of smart vehicles, driving behavior and the mental state of the driver should be a major part of the data considered. Past BIoT systems have primarily focused on environmental variables for tracking and decision-making. However, analyzing the driving behavior of different drivers could greatly enhance the intelligence of these BIoT systems, as this would allow for more precise and tailored decisions in service provision while preserving the data security benefits of blockchain technology.

It is challenging to identify the mental state of a human, which tends to change over time based on the given environmental feedback and human thinking. To address the human factors in an SA system, the Situ framework [7–9] proposed a human-centric SA

definition, suggesting that the situations perceived by humans depend on the human's internal mental state, and the actions performed can be regarded as an external reflection. Correspondingly, the value changes in the environmental context are side effects of human actions that are externally observable. Based on this assumption, a smart system shall be designed to observe environmental contexts, perceive human actions, and induce their internal goal. As such, a higher level of service evolution can be achieved for the uttermost human user's benefit. In a nutshell, the Situ framework is formulated as $situ_t = \langle M, B, E \rangle_t$, a triplet where M represents the user's mental state (intention) that is hidden, B represents the behavior context, E represents the environmental context, and t is the independent time variable to indicate the time of the situation. Under this framework, a smart system assisting the smart environment could first perceive the user's sequential behavioral context alongside the surrounding environmental context within the selected time frame; then, various analytic techniques can be exploited to predict the situation fitting the user's intention inferred at the given time to move towards the goal state. For example, non-learning-based approaches that utilized Conditional Random Fields (CRF) [54] and Locality-Sensitive Hashing (LSH) [55] were studied for user intention identification. In Section 4, we elaborate on the design of a learning-based situation analysis model that uses the Situ framework as the theoretical foundation to induce M based on the sensory-perceived context information B and E , aimed at guiding actuation decisions within a BIoT system. In the remainder of this writing, Situ-analysis refers to situation analysis based on the Situ framework.

3. Preliminaries

As discussed in Section 2, there exist research works on integrating blockchain, IoT systems, and AI/ML. However, there is a lack of research on integrating AI/ML with situation-aware (or situ-aware) BIoT systems, which would result in learning-based situ-aware BIoT systems. This paper aims to fill this gap. We start by presenting the system model and two baseline designs that borrow the ideas proposed in the research on integrating AI/ML and BIoT. Figure 1 provides a comprehensive overview of the baseline systems that will be elaborated upon in the following sections.

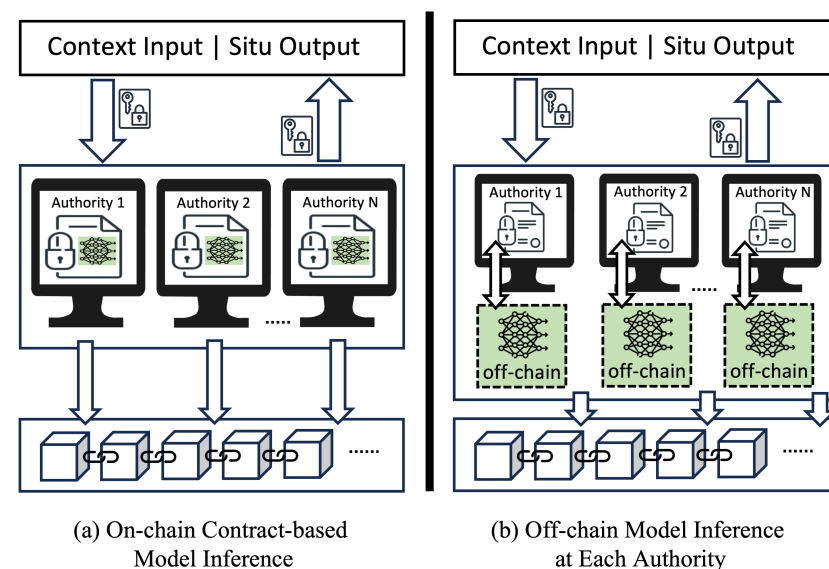


Figure 1. Baseline Designs: (a) the design is based on an on-chain approach, wherein learning-based Situ-analysis is directly implemented using smart contracts for both training and prediction processes; (b) the design is based on an off-chain approach, where analysis models are trained and deployed at each authority node. In this setup, the consensus protocol mandates that authorities provide both model inference results and model hashes for validation purposes.

3.1. System Model

As illustrated by the left side of Figure 2, the system model that we consider is a BIoT-based smart-living home environment. There are three physical device layers in such an environment: in-home monitoring sensors, in-home smart actuators (appliances), and controllers located on-premises, at the edge, or in the cloud. It is worth mentioning that the multi-node controller set in the figure depicts the concept of having not only one central hub in the space acting as the control unit but rather accepting a group of resource-sufficient devices to perform decision logic at different levels of use cases and locations within the smart-living setting. This also means that smart mobile devices (e.g., smartphones) connecting to the network are also considered controlling entities.

Having the controllers structured in such a way is essential to forming a peer-to-peer, interconnected blockchain network within the system. In addition, we consider these resource-sufficient controlling devices to be full nodes for the BIoT use case. This implies that smart contracts and transaction ledgers are kept at each controller for execution and storage purposes in a decentralized manner. Regarding the in-home monitoring sensors and actuators, as they tend to be resource-constrained, we consider them as the participating nodes in the blockchain network, which means they transmit transactions for updating context values but do not perform transaction validation, contract execution, or ledger storage.

3.2. Baseline-1: Situ-Aware BIoT Based on On-Chain Smart Contracts

To integrate AI/ML into a situ-aware BIoT system, certain learning-based Situ-analysis functions should be implemented inside the system. As shown in Figure 1a, this can be achieved through the direct implementation of learning-based smart contracts, wherein both training and prediction processes are executed on-chain. However, it is important to note that smart contract-enabled blockchains are deterministic and computationally constrained. The primary challenges in implementing a learning contract involve addressing the complexities of high-precision floating-point number computations and the associated training costs.

A previous study by Badruddoja et al. [42] depicts how to train Naive Bayes (NB) predictors on-chain and subsequently use them for classification tasks. The predictor training entails computing means, variances, and prior probabilities for each class, while the prediction involves Gaussian posterior probability calculations, i.e., $p(C|x)$, for all the potential classes. The final prediction is based on pair-wise posterior probability comparisons: $p(C|x) = \frac{p(x|C)p(C)}{p(x)}$.

More specifically, assuming a Gaussian Naive Bayes model with normal distributions denoted as $N(\mu_k, i, \sigma^2, k, i)$, the reformulated numerator is as follows (C_k denote the k th class, and x_i denote i th feature):

$$\begin{aligned} p(x|C_k)p(C_k) &= p(x_1|C_k) \cdots p(x_N|C_k)p(C_k) \\ &= \frac{p(C_k) \times 1}{\sqrt{(2\pi)^N \prod_i \sigma_{k,i}^2}} e^{-\frac{(x_1 - \mu_{k,1})^2}{2\sigma_{k,1}^2} - \cdots - \frac{(x_N - \mu_{k,N})^2}{2\sigma_{k,N}^2}} \end{aligned} \quad (1)$$

To address the constraints of floating-point arithmetic in smart contract execution, the study [42] adopted techniques such as squaring both sides of equations to eliminate the need for square root computations and leveraging Taylor series expansions for approximating exponential functions.

In our work, we adapt this existing approach and utilize it as *Baseline-1* for a comparative evaluation with our main proposal, i.e., the Situ-Oracle system elaborated in Section 4. The structures of the smart contracts for training and prediction are presented in Algorithms 1 and 2, respectively. The training time complexity of the Naive Bayes algorithm is $\mathcal{O}(n * d)$ and the prediction complexity is $\mathcal{O}(c * d)$, where n is the number of samples, d is the number of features, and c is the number of classes.

Algorithm 1: NB-based-Situ-Training Contract

StateVariable: *mean* ▷ mean of features for each situ
StateVariable: *var* ▷ variance of features for each situ

```

1 Function cal_mean(features, n)
2 | ▷ calculates the mean of the current set of features
3 Function cal_variance(features, m, n)
4 | ▷ calculates variance of the current set of features
5 Function get_mean()
6 | ▷ returns the mean of the current set of features
7 Function get_variance()
8 | ▷ returns the variance of the current set of features

```

Algorithm 2: NB-based-Situ-Prediction Contract

StateVariable: *Gaussian parameters*
StateVariable: *prior probabilities*
StateVariable: *posterior probabilities*

```

1 Function cal_parameter(features, var, mean)
2 | ▷ Compute Gaussian parameters for each class
3 Function cal_exponent(Gaussian parameters)
4 | ▷ Compute Taylor series-based exponential
5 Function cal_probability(prior probabilities)
6 | ▷ Computes Gaussian probability of all classes
7 Function predict(posterior probabilities)
8 | ▷ Compare and return the most likely class

```

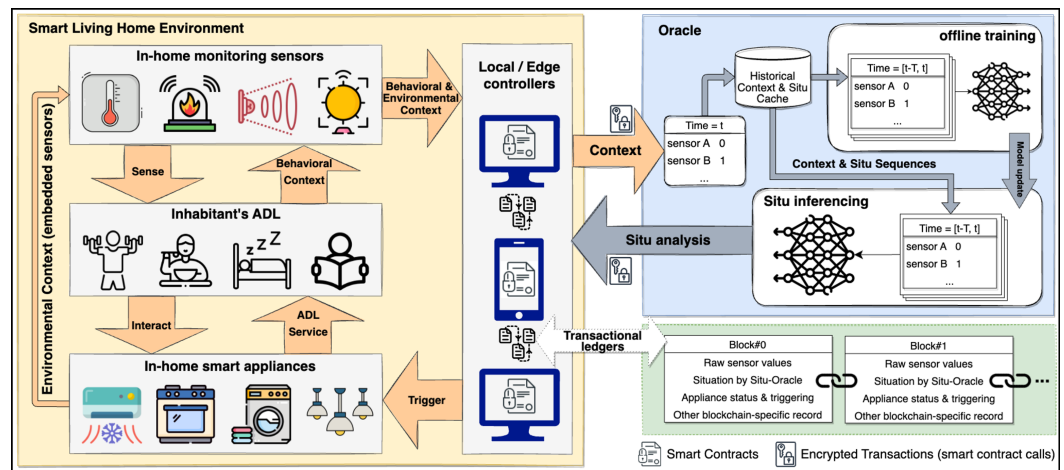


Figure 2. An Overview of the Proposed Situ-Oracle Framework (Using Smart-Living Home as an Example).

3.3. Baseline-2: Situ-Aware BIoT Based on Off-Chain Model at Each Authority

Figure 1b illustrates the alternative baseline design. This method operates under the premise that each authority node locally deploys a well-trained analysis model. When it is a node's turn for transaction validation, as per the Clique Proof-of-Authority (PoA) protocol, the node retrieves context input from on-chain contracts and performs model inference off-chain. Subsequently, it submits the Situ-analysis result and the model hash to the rest of the network for validation. This strategy is primarily designed to compensate for the computational limitations inherent in blockchain by bringing model training and inference off-chain.

Formally, suppose we have n pre-defined authorities in the Clique-based blockchain network, the public addresses of the authority nodes are denoted as $\lambda_a = \{\lambda_{a,1}, \lambda_{a,2}, \dots, \lambda_{a,n}\}$. Let $\mathcal{M}_{\lambda_{a,i}}$ be the local model at each authority node $\lambda_{a,i}$; $\mathcal{H}_{\mathcal{M}_{\lambda_{a,i}}} = \text{SHA256}(\mathcal{M}_{\lambda_{a,i}})$ be the hash of the model; $M_{\lambda_{a,i}}$ be the inference result at authority $\lambda_{a,i}$; $p: M \rightarrow A$ be the user intention to actuation mapping; and $q: \lambda \rightarrow M$ be the mapping of each authority $\lambda_{a,i}$ to their model inference result. The consensus process of an off-chain-based Situ-analysis service is based on the smart contract shown in Algorithm 3.

Algorithm 3: Off-chain-based Situ Contract

StateVariable: $M, \lambda_a, \mathcal{H}$
StateVariable: $p: M \rightarrow A, q: \lambda_a \rightarrow M$

```

1 Function constructor()
2   | Initialize  $\lambda_a, \mathcal{H}, p, q$ 
3 Function update_situ( $\hat{M}_{\lambda_{a,i}}, \mathcal{H}_{\mathcal{M}_{\lambda_{a,i}}}$ )
4   | require ( isAuthority(msg.sender) )
5   | require ( isValidModel( $\mathcal{H}_{\mathcal{M}_{\lambda_{a,i}}}$ ) )
6   |  $M[\lambda_{a,i}] \leftarrow \hat{M}_{\lambda_{a,i}}$ 
7   | if allDecisionsMade() then
8     |  $M \leftarrow \hat{M}_{\lambda_{a,i}}$ 
9     | emit_actuation_event (  $p[M]$  )
10    |  $q[\lambda_{a,i}] \rightarrow \phi, \forall \lambda_{a,i} \in \lambda_a$ 
11  | end
12 Function isAuthority( $\lambda^*$ )
13  | return bool  $\lambda^* \in \lambda_a$ 
14 Function isValidModel( $\mathcal{H}^*$ )
15  | return bool  $\mathcal{H}^* \equiv \mathcal{H}$ 
16 Function allDecisionsMade()
17  | return bool  $q[\lambda_{a,i}] \neq \phi, \forall \lambda_{a,i} \in \lambda_a$ 

```

Upon contract creation, a list of authority addresses λ_a , the SHA256 of the Situ-analysis model $\mathcal{H}$, the situ-based actuation mapping $p: M \rightarrow A$, and the initial authority inference results $q: \lambda_a \rightarrow M$ are initialized. The contract's core function, $update_situ()$, is designed to accept a new Situ-analysis result from a potential authority $\lambda_{a,i}$, and its corresponding model hash $\mathcal{H}_{\mathcal{M}_{\lambda_{a,i}}}$. This function updates the current user intention M once consensus is achieved among all authority nodes. Next, an actuation event is emitted whenever M is updated, therefore simulating service transition for the users. The complexity of the contract can be summarized as $\mathcal{O}(n)$, where n in this case denotes the number of authorities. In this paper, we adopt this design as *Baseline-2* for comparative evaluations with our Situ-Oracle system.

4. Proposed Framework: Situ-Oracle

In this section, we propose a new learning-based Situ-analysis servicing framework, based on the following assumptions.

Assumption 1. *The application context of the study is smart-living home environments, where IoT-based sensors, actuators, and controllers are present to meet the automation expectations.*

Assumption 2. *The smart-living home environment is a single-occupant space, i.e., only one person who can make rational decisions and actions resides in the given space.*

Assumption 3. *The oracle node that performs Situ-analysis and delivery is trustworthy, which implies that data generation and transmission at the oracle are trusted without potential tampering.*

This assumption is based on the following facts: different from IoT devices (i.e., sensors, actuators, and controllers), which are provided (and fully or partially managed) by a variety of potentially un-trusted vendors and are vulnerable to attacks (due to constrained resources and thus limited protection), an oracle node is deployed and fully managed by the smart home system administrator, and it can be run in a security-hardened environment due to its more isolated and dedicated working context and its resource-sufficiency.

4.1. Overview of the Framework

Figure 2 illustrates the proposed framework. In addition to the BIoT system on the left side, an oracle node is introduced to the system, which takes in behavioral and environmental contexts, performs off-chain model training, and constantly provides up-to-date Situ-analysis as a service to the BIoT network. With such a mechanism for Situ-analysis, smart contracts do not conduct sophisticated on-chain execution. The resulting distributed ledgers contain transaction information, including raw sensor values, the situation of the environment, actuator triggering traces, and other blockchain-specific records (nonce, gas usage, sealers, etc.). This enables the traceability of smart home automation and ensures its user data integrity.

4.2. Situ-Analysis Model Formulation

In this section, we explain the learning model design and rationale for Situ-analysis. Unlike other learning-based approaches, such as Bayesian classifiers, decision trees, or LSH, which process each input independently, RNN-based models can maintain a memory of previous inputs. This characteristic is particularly well-suited for processing time-series data, in our case, ADL sensory data, where the sequence and timing of activities are significant, and the context provided by previous sensor readings can greatly enhance the understanding and prediction of future situations. Therefore, we first define the system parameters and design a Vanilla RNN neural network for conceptual validation. Then, we further explore the design of utilizing LSTM and GRU-based neural networks to improve upon traditional RNNs by efficiently managing long-term dependencies, thus avoiding issues like vanishing or exploding gradients, making them more robust for long sequences typical in ADL monitoring.

Specifically, a sequence S of formatted sensory data is fed to the model, which, together with the surrounding environmental context E , is utilized to understand the sequential behavioral patterns B of the user. Then, the probabilities over M , the predetermined user intentions, are predicted by the model. The complete $situ_t$, including the predicted intention as well as the behavioral and environmental contexts with respect to time t , is finally provided to the on-chain smart contracts for decision-making. The main notations used in the remainder of the text are summarized in Table 1. The model for Situ-analysis is formulated as follows:

$$\mathcal{M} : \langle M_{t-T+1:t-1}, B_{t-T+1:t}, E_{t-T+1:t} \rangle \longrightarrow M_t, \quad (2)$$

where $M_{t-T+1:t-1}$ is the user's most recent intentions during $[t-T, t-1]$; $B_{t-T+1:t}$ is the sequence of the user's behavioral patterns during time interval $[t-T+1, t]$; and $E_{t-T+1:t}$ is the surrounding environmental context during $[t-T+1, t]$.

Figure 3a illustrates model $\mathcal{M}$. The input layer takes a batch of the two-dimensional feature vectors of size $T \times (|S| + 1)$, where $|S|$ denotes the total number of sensors in S . Then, an embedding layer normalizes the input feature vector, which is forwarded to the recurrent layer for final output calculation. The recurrent layer has various implementations, e.g., Vanilla RNN [56], Long-Short-Term-Memory (LSTM) [57], and Gated Recurrent Units (GRU) [58], for different use cases.

Table 1. A Summary of Key Notations.

Notation	Description
t	The current time instance.
T	The length of the time window for analysis.
n	The number of sensors.
m	The number of smart actuators.
k	The number of pre-defined user intentions.
$S_t = \langle s_{t,1}, \dots, s_{t,n} \rangle$	The set of sensory data produced at time t by all the n sensors, where $s_{t,i}$ for $i \in \{1, \dots, n\}$ denotes the sensory data by sensor i .
$A_t = \langle a_{t,1}, \dots, a_{t,m} \rangle$	The states of all the m smart actuators at time t , where $a_{t,i}$ for $i \in \{1, \dots, m\}$ denotes the state of smart actuator i .
B_t	The behavioral context of the user at time t .
E_t	The environmental context at time t .
$M_t \in \{d_1, \dots, d_k\}$	The user's intention at t , where d_i for $i \in \{1, \dots, k\}$ denotes a pre-defined user intention.

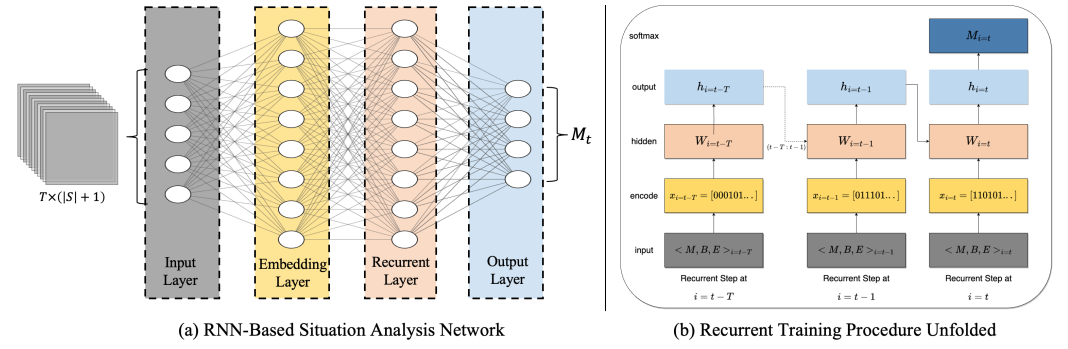


Figure 3. Recurrent Neural Network Learning Model for Situ-analysis. (a) depicts the overall neural network structure: $T \times (|S| + 1)$, where $|S| + 1$ denotes the total number of sensors in S alongside the previous user intention M . (b) illustrates the unfolded iterative learning process of the Vanilla RNN-based model. Note that the input vector will be padded with zeros for M at $i = t$ and for any missing sensory readings for the consistency of the input vector.

The iterative learning process of the Vanilla RNN-based model is unfolded in Figure 3b. Given input $\langle M, B, E \rangle_i$ of time frame $i \in [t - T + 1, t]$, the model iteratively feeds $\langle M, B, E \rangle_i$ for $\forall i \in [t - T + 1 : t]$ to the network for T times. For the normalized feature x_i of iteration $i \in [t - T + 1 : t]$ and $Tanh$ as activation function, the recurrent layer performs the following computation to obtain a temporary output h_i :

$$h_i = \tanh(W_{x_i} x_i + W_{h_i} h_{i-1}), \quad (3)$$

where W_{x_i} and W_{h_i} are neuron weight matrices of size $|M_i| \times (|S_i| + |M_i|)$ and $|M_i| \times |M_i|$, as the previous iteration output and the current iteration input, respectively (we use W_i in the figure to denote these matrices for simplicity).

Finally, given user intentions $\{d_1, d_2, \dots, d_k\}$, a *SoftMax* activation function σ is applied to vector $h_{i=t}$ to obtain the probabilities over $\{d_1, d_2, \dots, d_k\}$ at time t . Here, M_t is the highest probability among $\{d_1, d_2, \dots, d_k\}$, which is the final output of the Situ-analysis model $\mathcal{M}$:

$$M_t = \arg \max_{d \in \{d_1, d_2, \dots, d_k\}} (\sigma(h_{i=t})). \quad (4)$$

Dealing with activity sequence interruption (noise): An inhabitant is commonly interrupted by sub-tasks while attempting to achieve a major task, especially within a smart-living home environment. For example, while the inhabitant intends to “sleep”, some of the activities during this course may be interpreted as for another intention. To address this issue, we improve our analysis model by exploiting more sophisticated recurrent layers,

e.g., LSTM. In essence, the LSTM recurrent layer introduces “cell memory”. Instead of forwarding all information from the previous recurrent learning iteration (Vanilla RNN), the LSTM layer adds extra activation functions and masking steps within a neuron, referred to as “cell states”. This way, only the key patterns to the next iteration are fed forward, while the irrelevant information from the prior iteration is “forgotten” gradually. The LSTM layer computation is expressed as follows:

$$f_i = \text{sigmoid}(W_f \cdot [h_{i-1}, x_i] + b_f), \quad (5)$$

$$g_i = \text{sigmoid}(W_g \cdot [h_{i-1}, x_i] + b_g), \quad (6)$$

$$\tilde{C}_i = \tanh(W_c \cdot [h_{i-1}, x_i] + b_c), \quad (7)$$

and

$$C_i = f_i C_{i-1} + g_i \tilde{C}_i. \quad (8)$$

Here, $f_{i=t}$ is a *sigmoid* activation function that is used as a mask to ignore irrelevant cell states from the previous iteration; $g_{i=t}$ is another *sigmoid* function for masking out the irrelevant cell state from the input of the current iteration; $\tilde{C}_{i=t}$ is the current unmasked cell state; and $C_{i=t}$ will be the final calculated cell state after the current iteration.

Next, similar to the previously described Vanilla RNN-based approach, the temporary output at iteration $i \in [t - T + 1 : t]$ is computed as follows with the cell state added:

$$h_{i=t} = \tanh(W_{x_{i=t}} x_{i=t} + W_{h_{i=t}} h_{i=t-1}) \cdot \tanh(C_{i=t}). \quad (9)$$

Lastly, the user intention $M_{i=t}$ can be predicted using Equation (4). Intuitively, the extra cell states enable the model to highlight key user activities among all input sensory sequences. Therefore, the irrelevant sub-tasks (noises) during a user intentions identification process can be reduced.

4.3. Situ-Analysis as a Service to BIoT Systems

4.3.1. Blockchain Network and Smart Contracts

The actuators, sensors, controllers, and the oracle form key components of the blockchain network. Each node maintains a public-private key pair to participate in network activities, e.g., sending transactions. In the following text, we use λ to denote the public key (i.e., address or account) of a node in the network. The encrypted transactions initiated by a node should be signed with its private key correspondingly.

In our design, two smart contracts, named “Context” and “Situ”, are deployed for the Situ-analysis service. As presented in Algorithm 4, the “Context” contract maintains four state variables $\lambda_s, \lambda_o, B, E$, where λ_s and λ_o are the lists of sensors’ and oracle’s public key, respectively; B and E are the behavioral and environmental contexts updated with sensory readings, respectively. Upon contract creation, the pre-validated keys of the sensors and the oracle are written to the contract, while the context variables B and E are initialized. Two major functions are implemented in the “Context” contract, i.e., *update_context* and *get_context*. Function *update_context* takes as input sensory readings S then updates state variables B and E only if the initiator of the transaction is a valid sensor. Subsequently, the oracle node calls contract *get_context* to retrieve the context values and perform Situ-analysis.

Algorithm 5 shows the structure of the “Situ” contract. Specifically, it maintains three state variables $M, \lambda_o, p: M \rightarrow A$, where M is the current user intention; λ_o is the valid oracle’s address; and the mapping $p: M \rightarrow A$ maps the user intention to actuator states for service transition. The *update_situ* function inputs the new user intention $\hat{M}$ from a valid oracle node. Then, it updates the recorded user intention, followed by emitting an actuation transition event for the service updates. Additionally, the oracle’s address can be updated via function *update_oracle* by the system administrator. For any node, *get_situ* can be called to obtain the current user intention.

Algorithm 4: Context Contract

StateVariable: $\lambda_s, \lambda_o, B, E$

```

1 Function constructor()
2   | Initialize  $\lambda_s, \lambda_o, B, E$ 
3 Function update_context( $S$ )
4   | require (  $msg.sender \in \lambda_s$  )
5   |  $\langle B, E \rangle \leftarrow S$ 
6 Function get_context()
7   | require (  $msg.sender \equiv \lambda_o$  )
8   | return  $\langle B, E \rangle$ 

```

Algorithm 5: Situ Contract

StateVariable: $M, \lambda_o, p: M \rightarrow A$

```

1 Function constructor()
2   | Initialize  $\lambda_o$ 
3 Function update_situ( $\hat{M}$ )
4   | require (  $msg.sender \equiv \lambda_o$  )
5   |  $M \leftarrow \hat{M}$ 
6   | emit_actuation_event (  $p[M]$  )
7 Function switch_oracle( $\hat{\lambda}_o$ )
8   | require (  $msg.sender \equiv \lambda_o$  )
9   |  $\lambda_o \leftarrow \hat{\lambda}_o$ 
10 Function get_situ()
11 | return  $M$ 

```

4.3.2. The Oracle Node

The main tasks of the oracle node include: (1) raw context data caching; (2) online Situ-analysis (i.e., inference); and (3) offline Situ-analysis model training. The operation routine is presented in Algorithm 6. During the operation, up-to-date context values, i.e., $\langle B, E \rangle_{i=t}$, are acquired from “Context”, then stored in a local cache H of the oracle’s system. Next, the online learning model is used to perform Situ-analysis, using the obtained context values $\langle B, E \rangle_{i \in [t-T+1, t]}$ and previous user intention $M_{i \in [t-T+1, t-1]}$. Following that, it compares the new predicted user intention $M_{i=t}$ with the last-analyzed user intention $M_{i=t-1}$, and updates the state variable M in “Situ” if a new user intention is recognized, i.e., $M_{i=t} \neq M_{i=t-1}$. Lastly, $M_{i=t}$ is inserted into H for future Situ-analysis and offline model training.

Algorithm 6: Oracle Operation Routine

SystemVariable: $H, \mathcal{M}$

```

1 while TRUE do
2   |  $\langle B, E \rangle_{i=t} \leftarrow \text{Call\_Context\_Contract}$ 
3   | Append  $\langle B, E \rangle_{i=t}$  to  $H$ 
4   |  $M \leftarrow H[M_{i \in [t-T+1, t]}]$ 
5   |  $B \leftarrow H[B_{i \in [t-T+1, t]}]$ 
6   |  $E \leftarrow H[E_{i \in [t-T+1, t]}]$ 
7   |  $M_{i=t} = \mathcal{M}(M, B, E)$ 
8   | if  $M_{i=t} \neq M_{i=t-1}$  then
9     | | Call_Situ_Contract( $M_{i=t}$ )
10  | end
11  | Append  $M_{i=t}$  to  $H$ 
12 end

```

4.3.3. Overview of the System Workflow

Figure 4 illustrates the system’s operational process. At runtime, passive and wearable sensors capture the inhabitant’s ADL interactions. The perceived values, along with the status of actuators (measured by embedded sensors), are recorded in “Context”. The oracle then retrieves the context values from “Context” for Situ-analysis. The results of this analysis are updated in “Situ”. When a new *situ* is received, “Situ” triggers actuation decisions to enact suitable service adjustments for the inhabitant.

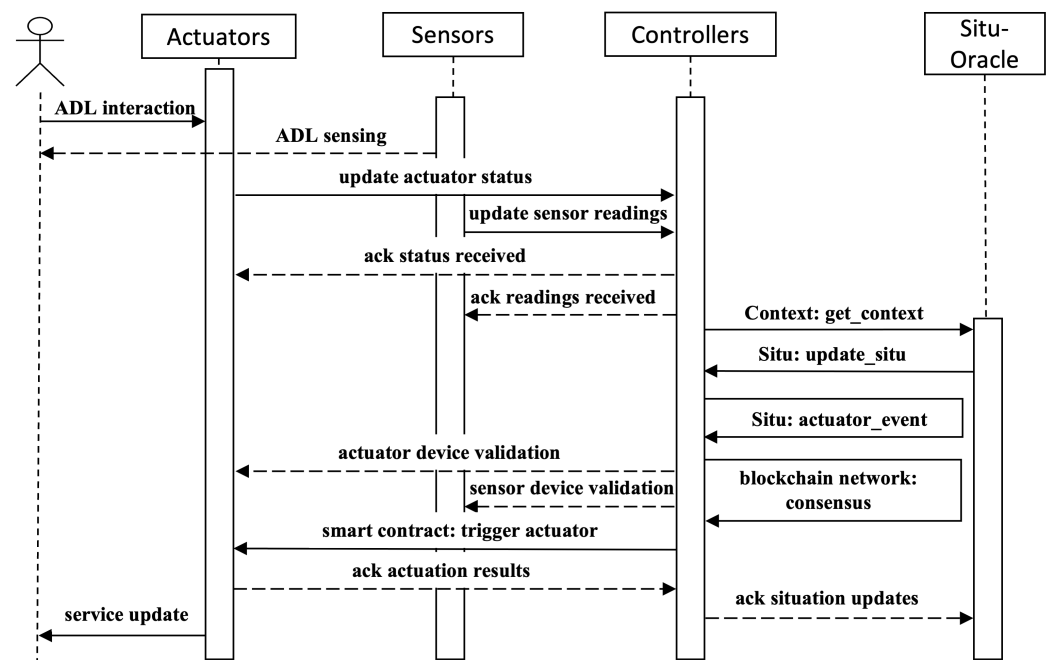


Figure 4. Sequence Diagram of the System Workflow.

5. Experiments and Evaluations

In this section, we first conduct a comparative evaluation of different Situ-analysis models. Following that, we conduct comparative evaluations of the system performance between the proposed Situ-Oracle framework with the two baseline systems. This section also provides an overview of the ADL dataset and the configuration of the testing blockchain systems. The detailed implementation and the codebase are publicly accessible on GitHub for reference (<https://github.com/okhobian/situ-oracle-paper>, accessed on 20 May 2024).

5.1. The SIMADL Dataset

OpenSHS [59], built upon open-source 3D modeling software Blender [60], is a widely utilized open-source tool for simulating smart home environments, primarily used to generate extensive datasets of time-sequenced ADL sensor activities in single-occupant scenarios. It allows for the placement and deployment of virtual sensors in various locations. Testers can interact with the pre-built environment through a first-person view, where they simulate the occupant’s daily routine. During data collection, the participant sets an ADL goal (e.g., sleep, eat, work), and the tool records the sequence of sensor activities triggered by the occupant’s actions. An example of the dataset generated by this process is presented in Table 2:

Table 2. A snapshot of SIMADL dataset for evaluation.

Timestamp	Bathroom Light	Carpet Pressure	...	Couch Pressure	Situ
t = 1	0	0	...	1	WatchTV
t = 2	0	0	...	1	WatchTV
t = 3	1	1	...	0	Personal
t = 4	1	1	...	0	Personal
t = 5	0	0	...	0	Sleep
t = 6	0	0	...	0	Sleep
⋮	⋮	⋮	⋮	⋮	⋮

The SIMADL [61] dataset was created with seven participants interacting with the OpenSHS’ virtual smart house (no interactions between participants), specifically:

- A smart house consists of one bedroom, one living room, one bathroom, one kitchen, and one in-home office.
- A total of 29 sensors were deployed. Each sensor has two states, on (1) and off (0), and two types, passive and active. Passive sensors operate without direct interaction from the participant and automatically respond to the participant’s movements and position. Active sensors require actions from the participant to alter their state, such as opening a door or switching on a light.
- The user intentions incorporated into this dataset include sleep, eating, personal, work, leisure, and others.
- The SIMADL aggregated dataset for each individual participant, each simulated over varying durations, such as 1 or 2 months, with time margins of 0, 5, or 10 min.

We utilized datasets representing a duration of 1 month with a 0 time margin for all participants. The sub-datasets for each participant consist of distinct record counts: 18,800; 37,332; 39,832; 40,232; 27,762; 81,859; and 49,282, respectively.

5.2. Testbed Configurations

BloT Network: The experimental private blockchain network comprises two groups of Raspberry Pis (RPi), including the 3b and 4b models interconnected using a network switch. This dual-version setup of RPis is specifically designed to investigate the effects of constrained IoT devices on performance outcomes, given that the RPis are configured with standard, commercially available specifications and are frequently used in IoT studies. Each RPi node operates the Ethereum client software, Go-Ethereum (Geth) of version linux-1.10.25, and the private Ethereum network is configured to run the officially supported Proof-of-Authority (PoA) consensus protocol, Clique [62]. All the RPi nodes were configured as sealers adhering to the specifications of Clique, in which they take turns validating transactions and packaging blocks. This network is configured with a gas limit set to 50 million units. An experiment with varying target block packaging times is conducted to analyze the resulting performance impacts.

Oracle Node: The oracle node, which includes the Situ-analysis model and a historical data cache, is implemented on a computing node with sufficient resources. This node is part of the same blockchain network as the IoT devices (RPis). The computing node’s core specifications are: a 6th generation Intel i5 processor, an Nvidia GTX 960M graphics card, 16 GB of 1600 MHz dual memory, and Ubuntu 20.04 operating system. The oracle node also operates Geth version 1.10.25, along with the web3 Python-SDK version 5.31.1, for interacting with smart contracts. For processing Situ-analysis models, the oracle node is equipped with Scikit-Learn version 1.3. and Keras-TensorFlow version 2.3.0.

5.3. Situ-Analysis Model Evaluations

The Situ-analysis model is the foremost component of the proposed Situ-Oracle framework. Previous research [6] has employed threshold-based methods for context-aware smart contracts. In our experimental setup, we initially examined the feasibility of a simple threshold-based approach for the Situ-analysis task. Given the complexity of the dataset, implementing an if-else structured algorithm is not straightforward. Hence, we constructed a basic Decision Tree (DT) model with a depth of two, serving as a proxy to emulate the performance of an if-else, threshold-based Situ-analysis algorithm.

Additionally, we investigated a non-learning approach as an intermediate method for Situ-analysis. With reference to the findings from [55], we considered Locality-sensitive hashing (LSH), a technique for effective approximation of nearest neighbor searches in high-dimensional spaces, for Situ-analysis tasks. In our experiment, an LSH bucket with default settings was implemented to serve as a benchmark of the prediction accuracy for evaluation. Furthermore, we implemented the baseline systems using Scikit-Learn in our experiment setup to facilitate a performance comparison.

Lastly, we developed three deep-learning models using the Keras-TensorFlow framework, adhering to our proposed RNN-based Situ-analysis service. Each learning model is based on a different recurrent neural network architecture: Vanilla RNN, LSTM, and GRU. The sampling rate of the sensors is assumed to be one reading per second, making the past time window T equivalent to T rows from the dataset. The training and testing were performed with a 70%–30% dataset split. Given that this is a multi-class, single-label classification problem in the context of machine learning, we evaluated and compared the learning approaches based on the exact match ratio:

$$\text{Accuracy} = \frac{\sum_{i=1}^N [\hat{y}_i = y_i]}{N} \quad (10)$$

where N is the total number of sequences; $\hat{y}_i$ and y_i are the predicted and the actual class label (situation) for the i -th sequence, respectively.

Our preliminary study [11] demonstrated that the RNN-based learning models can attain high prediction accuracy, in both non-interruptive and interruptive datasets. Moreover, under a similar simulation setting in the previous study, these models showed optimal performance with an input sequence length of $T = 15$. Hence, this study also adopts input sequence length $T = 15$ to conduct comparisons with alternative methods using more realistic datasets that were collected from experiments involving random human participants.

Figure 5 illustrates the Situ-analysis accuracy comparison results. The six models were trained and tested on all 1-month, 0-min-margin sub-datasets collected with seven participants. While we primarily used an input sequence length of $T = 15$, we also evaluated $T = 1$ for non-RNN-based approaches, as these methods typically do not account for temporal relationships in input sequences. Interestingly, the length of the input sequence did not result in notable differences in outcomes across these methods. Contrary to expectations, a longer sequence length, i.e., $T = 15$, actually led to improved accuracy in these methods despite their lack of focus on temporal dynamics. Conversely, all three RNN-based models not only achieved high accuracy as anticipated, but also exhibited stable prediction capabilities across the diverse living styles of all participants. In contrast, the performance of the non-RNN-based method fluctuated in this regard. Overall, as demonstrated in the Average chart, the RNN-based approaches show a consistently high level of performance in Situ-analysis.

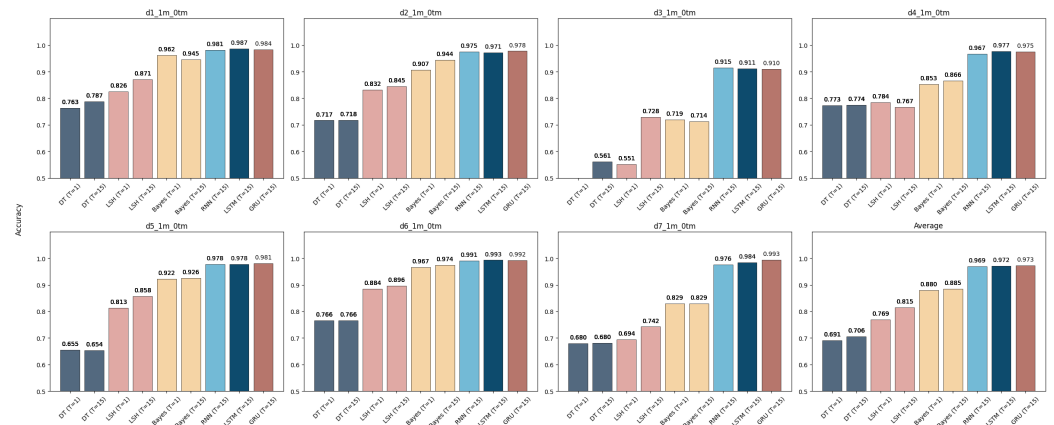


Figure 5. Accuracy Comparisons in Situ-analysis: The seven sub-datasets from the SIMADL dataset, each representing one of the seven participants with a duration of 1 month and a 0-min time margin, were trained and tested with four learning models individually: Decision Tree (DT), Locality-Sensitive Hashing (LSH), Naive Bayes (NB), and RNNs (Vallina RNN, LSTM, GRU). $T = 1$ and $T = 15$ were tested for DT, LSH, and NB; only $T = 15$ were tested for RNNs.

5.4. System Performance Evaluations

Gas Consumption: We first evaluated the gas consumption of the three frameworks. The experiment was conducted with the *d1_1m_0tm* sub-dataset from SIMADL. In the on-chain model prediction framework (baseline-1), a Naive Bayes model was employed, while both the off-chain model prediction approach (baseline-2) and the proposed Situ-Oracle framework utilized an LSTM model. It is important to note that by design, both the baseline-2 and Situ-Oracle frameworks are model-agnostic. Additionally, for these two frameworks, model training and inference occur off-chain. Therefore, no gas consumption during the training phase was incurred, and the gas consumption for prediction is independent of the model being used. Table 3 shows the average gas consumption across the specified settings for each of the three frameworks. Despite the baseline-1 system requiring only 141,833 units of gas to train a single epoch, the total gas consumption for training was not optimal, especially when considering the size of the dataset (18,800 rows) and the number of epochs repeated during the training process. On the other hand, the prediction process in the baseline-1 system involves executing all functions outlined in Algorithm 2, while the other two systems, baseline-2, and Situ-Oracle, simply call the *update_situ* function in their respective contracts as detailed in Algorithm 3. This results in a substantial difference in terms of the prediction cost among the frameworks. The Situ-Oracle framework demonstrated the lowest gas consumption for predictions, whereas the on-chain model approach proved to be completely impractical in terms of execution costs.

Table 3. The average gas units that were consumed by the three frameworks, for both the training and prediction phases (The Baseline-2 and the Situ-Oracle system do not incur gas consumption because of their off-chain model-agnostic design).

	Training (per Epoch)	Prediction (per Inference)
Baseline-1	141,833	35,965,460
Baseline-2	not applicable	175,774
Situ-Oracle	not applicable	31,821

Scalability: Next, we evaluated the scalability of the blockchain network. Specifically, we conducted experiments to examine the throughput of the blockchain network, measured in transactions per second (TPS). Let $\mathbf{B} = \{b_1, b_2, \dots, b_n\}$ be the set of blocks that were

appended to the blockchain, during which the system offers the Situ-analysis services. Then, we denote the actual block time of b_i as μ_{b_i} , which can be obtained by:

$$\mu_{b_i} = \text{timestamp}(b_{i+1}) - \text{timestamp}(b_i); 0 < i < n \quad (11)$$

In our study, we initially set the target block generation times per the Clique protocol requirement. However, it is important to note that the real block generation time may differ due to variations in computation and validation durations across individual sealer nodes within the network at runtime. Additionally, the number of transactions included in a block, denoted as η_{b_i} , for block b_i , can vary depending on the block gas limit and the gas consumption of each transaction. Consequently, the system's throughput during the period when the Situ-analysis service was operational can be calculated as follows:

$$TPS = \frac{1}{|B|} \times \sum_{i=1}^{|B|} \frac{\eta_{b_i}}{\mu_{b_i}} \quad (12)$$

Figure 6 illustrates the actual block sealing times, μ , observed for three Situ-analysis servicing frameworks, with three different target block generation times, i.e., 5, 10, and 15 s, executed on RPi3b and RPi4b sealer nodes. Among the three evaluated system configurations, the proposed Situ-Oracle framework generally exhibited the most efficient performance, recording the lowest block sealing times. In one exceptional case, the target block time was set to 10 s on the RPi4b sealers. Further analysis of the throughput revealed that, despite this deviation, the system succeeded in processing a significant number of transactions in each block, therefore sustaining a high level of throughput. In contrast, the on-chain model prediction system (baseline-1) often utilized the full duration of the allotted time. The experiment demonstrated that most of the sealer nodes can finalize block packaging within the predetermined generation time, indicating that the networks remained in a relatively healthy state throughout the tests.

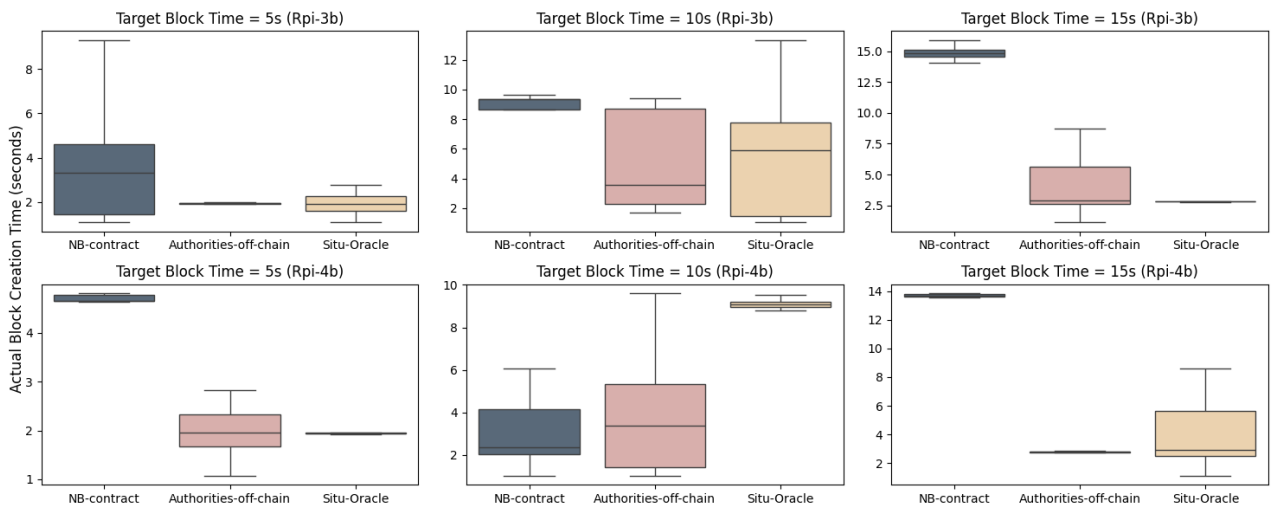


Figure 6. Actual block sealing times, μ , observed for three Situ-analysis servicing frameworks, with three different target block generation times, i.e., 5, 10, and 15 s, executed on RPi3b and RPi4b sealer nodes.

Figure 7 highlights the estimated network throughput, measured in transactions per second (TPS), during the Situ-analysis service for each system. In our experiment, a higher target block time often results in a higher network throughput, as expected for all three systems. Additionally, the experimental results demonstrate that the proposed Situ-Oracle framework consistently outperformed both baseline systems in terms of throughput. Specifically, the on-chain model prediction system (baseline-1) struggled to reach 1 TPS in

most scenarios, while the off-chain model prediction approach (baseline-2) maintained an average throughput close to 10 TPS. In contrast, the Situ-Oracle framework achieved an average of nearly 20 TPS across all tested settings.

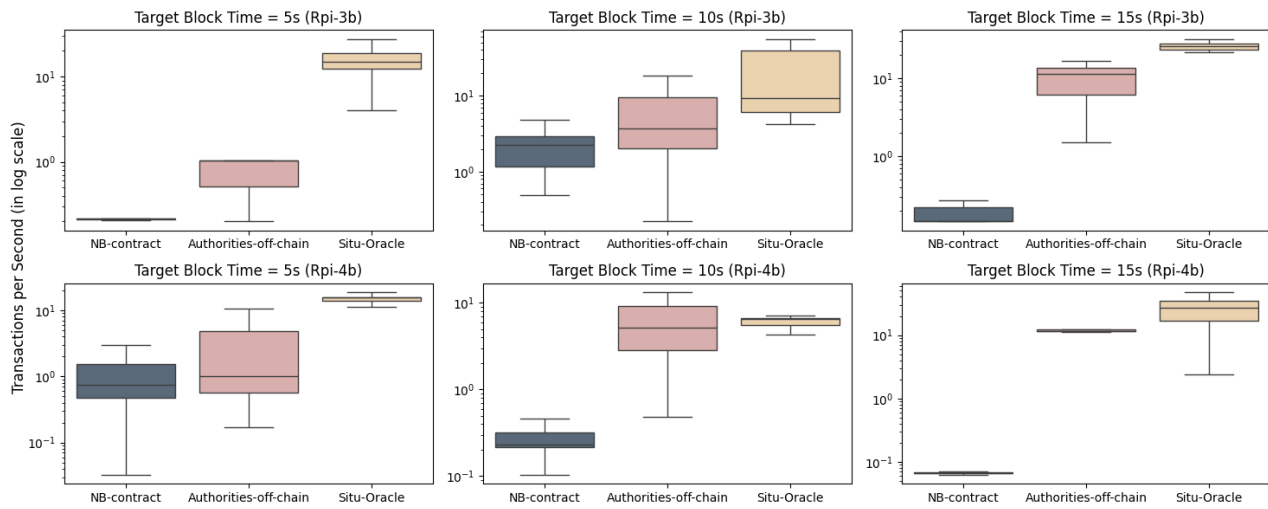


Figure 7. Estimated network throughput, measured in transactions per second (TPS), for three Situ-analysis servicing frameworks, with three different target block generation times, i.e., 5, 10, and 15 s, executed on RPi-3b and RPi-4b sealer nodes. Note: Due to the substantial discrepancies observed in the measured results, we adjusted the y-axis display to a logarithmic scale for better clarity and comparison of relations.

Latency: We evaluate the Situ-analysis service latency over the three approaches. During the experiment, we conducted 100 sensory updates for each system across the six setups mentioned earlier. Suppose for a given transaction tx_i ($1 \leq i \leq 100$), we define the time of transaction initiation as t_{tx_i} ; and $\hat{t}_{tx_i}$ as the time when tx_i is included into a block, signifying that the transaction has been validated by the sealers and is permanently recorded on the blockchain network. The latency for each of the 100 sensory updates was obtained by calculating the time difference between the time that the sensory update transaction $tx_{context,i}$ was initiated and the time when the Situ-analysis is concealed onto the blockchain through the corresponding situ contract update transaction $tx_{situ,i}$, i.e., $\hat{t}_{tx_{situ,i}} - t_{tx_{context,i}}$.

Figure 8 illustrates the probability distribution of latency for 100 sensory (context) updates conducted on each of the three frameworks. Based on the distribution illustration, the proposed Situ-Oracle framework achieves the best latency performance (around 7 s per update among all settings). In comparison, the baseline-2 framework exhibited an average latency of approximately 28.5 s per update across all configurations. However, there was a significant latency disparity between the Situ-Oracle framework and the baseline-1 framework, with the latter experiencing over 1000 s per update across all settings. The on-chain model prediction approach evidently leads to significant network congestion over time. As illustrated in Figure 9, both the Situ-Oracle framework and the off-chain model prediction method (baseline-2) maintained consistent and stable network traffic throughout the experiment. However, the baseline-1 system initially showed a manageable latency but was rapidly trapped by network congestion due to the excessive computational demand on each sealer node. For the baseline-1 system, the RPi-4b demonstrated marginally better performance than the RPi-3b in this stress test. Nonetheless, as the number of context-updating transactions accumulated, the latency became practically unfeasible in this approach.

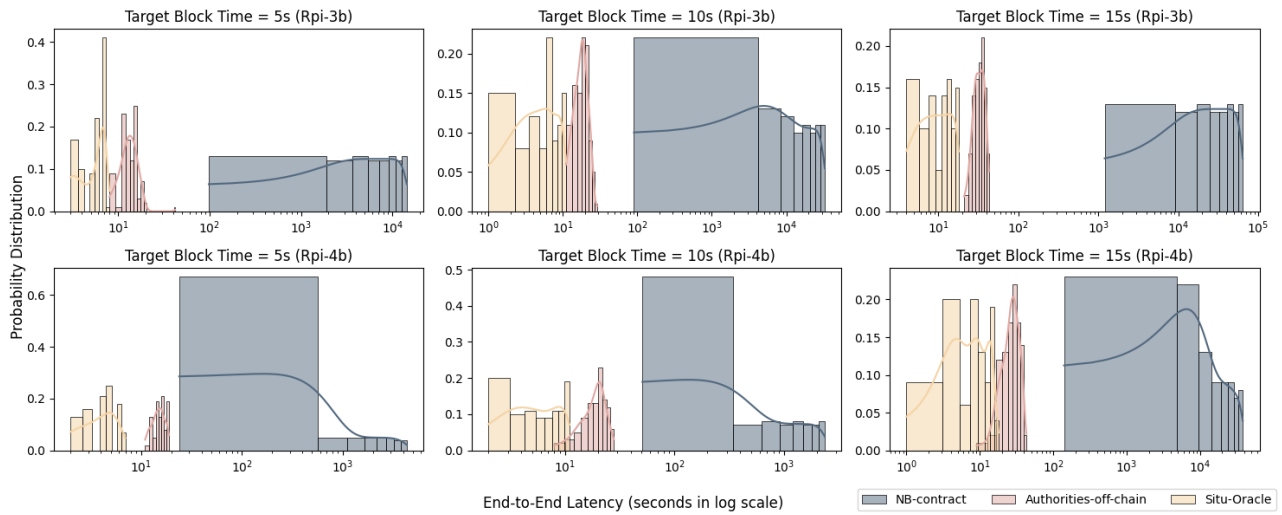


Figure 8. The probability distribution of Situ-analysis concealing latency for 100 sensory (context) updates, conducted on three Situ-analysis servicing frameworks, with three different target block generation times, i.e., 5, 10, and 15 s, executed on RPi3b and RPi4b sealer nodes. Note: due to the substantial discrepancies observed in the results, we adjusted the x-axis display to a logarithmic scale for better clarity and comparison of relations.

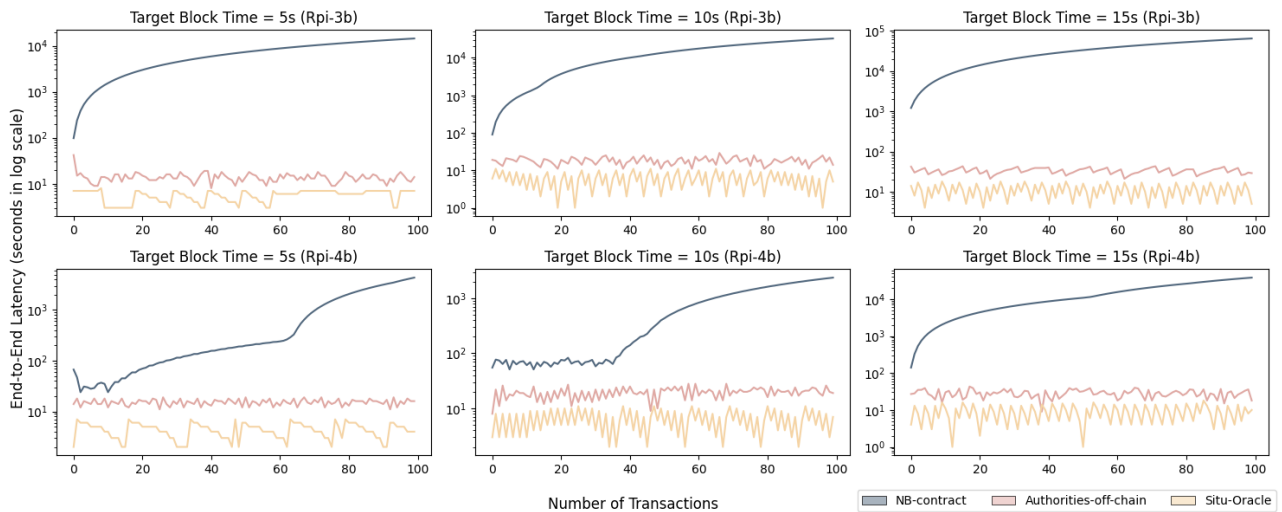


Figure 9. The Situ-analysis latency for each of the 100 sensory (context) updates. The Situ-Oracle framework and the off-chain model prediction method (baseline-2) maintained consistent and stable network traffic throughout the experiment. However, the baseline-1 system initially showed a manageable latency but was rapidly trapped by network congestion due to the excessive computational demand on each sealer node.

6. Conclusion and Future Works

This paper formulated and implemented a situation-aware, blockchain-based framework for smart IoT systems. We demonstrated our approach to embracing situation awareness in BIoT systems and used a smart home setting to exemplify the feasibility of the integration. The proposed Situ-Oracle design enables a fine-grained situ analysis as a service to IoT-centric smart contracts. The results indicate that the proposed framework achieves better situation analysis accuracy (above 95%) and improves gas consumption as well as network throughput and latency when compared to baseline systems (on-chain learning or off-chain model verification). Although we used the smart home as the illustration example, one can speculate the significance and extension of this research pertaining

to a much larger-scale smart environment, such as smart grids for situ-based P2P electricity exchange; smart cities, such as traffic congestion control, Air Quality Index (AQI) alarms, or disaster management in an earthquake-active zone, etc. The aforementioned use cases are predominantly human-centric applications, where the situation awareness provided by the Situ-Oracle framework can significantly enhance the “smartness” of these applications. For example, understanding the ADL patterns of an inhabitant could further aid in analyzing electricity usage, potentially broadening the scope for P2P electricity exchange in smart grid scenarios (common use case for BloT). Similarly, in the Internet of Vehicles (IoV), comprehending how drivers react in various situations could facilitate more effective smart traffic control measures. These scenarios all stand to benefit from the enhanced situational awareness that Situ-Oracle offers. Meanwhile, security concerns within the data and control planes can be effectively mitigated through the integration with the BloT infrastructure provided by the framework. Future work can be conducted to extend the system to atop a multi-user situ-analysis model, thus facilitating more realistic smart environments. Possible approaches include considering not only temporal but also spatial factors among the sensory sequences or considering attention-based mechanisms for user separation in terms of behavior and intention mapping distributions. Moreover, the Situ-Oracle system can be further explored in terms of security and privacy concerns. This can be achieved by incorporating a self-validating and adaptive Situ-Oracle system design with federated/distributed learning atop game-theory-based voting mechanisms.

Author Contributions: Writing—original draft, H.B., W.Z. and C.K.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hou, K.M.; Diao, X.; Shi, H.; Ding, H.; Zhou, H.; de Vaulx, C. Trends and Challenges in AIoT/IIoT/IoT Implementation. *Sensors* **2023**, *23*, 5074. [CrossRef]
2. Kim, D.; Bian, H.; Chang, C.K.; Dong, L.; Margrett, J. In-Home Monitoring Technology for Aging in Place: Scoping Review. *Interact. J. Med. Res.* **2022**, *11*, e39005. [CrossRef] [PubMed]
3. Munir, A.; Aved, A.; Blasch, E. Situational Awareness: Techniques, Challenges, and Prospects. *AI* **2022**, *3*, 55–77. [CrossRef]
4. Karie, N.M.; Sahri, N.M.; Yang, W.; Valli, C.; Kebande, V.R. A Review of Security Standards and Frameworks for IoT-Based Smart Environments. *IEEE Access* **2021**, *9*, 121975–121995. [CrossRef]
5. Buterin, V. A Next Generation Smart Contract & Decentralized Application Platform. Available online: https://finpedia.vn/wp-content/uploads/2022/02/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf (accessed on 18 May 2024).
6. Ngwira, L.; Merlec, M.M.; Lee, Y.K.; In, H.P. Towards Context-Aware Smart Contracts for Blockchain IoT Systems. In Proceedings of the 2021 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Republic of Korea, 20–22 October 2021; pp. 82–87. [CrossRef]
7. Chang, C.K.; Jiang, H.y.; Ming, H.; Oyama, K. Situ: A Situation-Theoretic Approach to Context-Aware Service Evolution. *IEEE Trans. Serv. Comput.* **2009**, *2*, 261–275. [CrossRef]
8. Chang, C.K. Situation Analytics: A Foundation for a New Software Engineering Paradigm. *Computer* **2016**, *49*, 24–33. [CrossRef]
9. Chang, C.K. Situation Analytics-at the Dawn of a New Software Engineering Paradigm. *Sci. China Inf. Sci.* **2018**, *61*, 050101. [CrossRef]
10. Antonopoulos, A.M.; Wood, G. *Mastering Ethereum: Building Smart Contracts and Dapps*; O’reilly Media: Sebastopol, CA, USA, 2018.
11. Bian, H.; Zhang, W.; Chang, C.K. Situ-Oracle: A Learning-Based Situation Analysis Servicing Framework for BloT Systems. In Proceedings of the 2023 IEEE International Conference on Software Services Engineering (SSE), Chicago, IL, USA, 2–8 July 2023; pp. 159–169. [CrossRef]
12. Hao, X.; Ren, W.; Fei, Y.; Zhu, T.; Choo, K.K.R. A Blockchain-Based Cross-Domain and Autonomous Access Control Scheme for Internet of Things. *IEEE Trans. Serv. Comput.* **2023**, *16*, 773–786. [CrossRef]
13. Liu, Y.; Zhang, C.; Yan, Y.; Zhou, X.; Tian, Z.; Zhang, J. A Semi-Centralized Trust Management Model Based on Blockchain for Data Exchange in IoT System. *IEEE Trans. Serv. Comput.* **2023**, *16*, 858–871. [CrossRef]
14. Yin, J.; Xiao, Y.; Pei, Q.; Ju, Y.; Liu, L.; Xiao, M.; Wu, C. SmartDID: A Novel Privacy-Preserving Identity Based on Blockchain for IoT. *IEEE Internet Things J.* **2023**, *10*, 6718–6732. [CrossRef]

15. Wu, T.; Wang, W.; Zhang, C.; Zhang, W.; Zhu, L.; Gai, K.; Wang, H. Blockchain-Based Anonymous Data Sharing with Accountability for Internet of Things. *IEEE Internet Things J.* **2023**, *10*, 5461–5475. [\[CrossRef\]](#)
16. Putra, G.D.; Dedeoglu, V.; Kanhere, S.S.; Jurdak, R. Privacy-Preserving Trust Management for Blockchain-based Resource Sharing in 6G-IoT. In Proceedings of the 2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Dubai, United Arab Emirates, 1–5 May 2023; pp. 1–9. [\[CrossRef\]](#)
17. Yazdinejad, A.; Dehghantanha, A.; Parizi, R.M.; Srivastava, G.; Karimipour, H. Secure Intelligent Fuzzy Blockchain Framework: Effective Threat Detection in IoT Networks. *Comput. Ind.* **2023**, *144*, 103801. [\[CrossRef\]](#)
18. Rabieinejad, E.; Yazdinejad, A.; Parizi, R.M.; Dehghantanha, A. Generative Adversarial Networks for Cyber Threat Hunting in Ethereum Blockchain. *Distrib. Ledger Technol. Res. Pract.* **2023**, *2*, 1–19. [\[CrossRef\]](#)
19. Sanghami, S.V.; Lee, J.J.; Hu, Q. Machine-Learning-Enhanced Blockchain Consensus with Transaction Prioritization for Smart Cities. *IEEE Internet Things J.* **2023**, *10*, 6661–6672. [\[CrossRef\]](#)
20. Lu, Y.; Tang, X.; Liu, L.; Yu, F.R.; Dustdar, S. Speeding at the Edge: An Efficient and Secure Redactable Blockchain for IoT-Based Smart Grid Systems. *IEEE Internet Things J.* **2023**, *10*, 12886–12897. [\[CrossRef\]](#)
21. Lin, X.; Wu, J.; Bashir, A.K.; Li, J.; Yang, W.; Piran, M.J. Blockchain-Based Incentive Energy-Knowledge Trading in IoT: Joint Power Transfer and AI Design. *IEEE Internet Things J.* **2022**, *9*, 14685–14698. [\[CrossRef\]](#)
22. Islam, M.J.; Rahman, A.; Kabir, S.; Karim, M.R.; Acharjee, U.K.; Nasir, M.K.; Band, S.S.; Sookhak, M.; Wu, S. Blockchain-SDN-Based Energy-Aware and Distributed Secure Architecture for IoT in Smart Cities. *IEEE Internet Things J.* **2022**, *9*, 3850–3864. [\[CrossRef\]](#)
23. Khan, A.U.R.; Ahmad, R.W. A Blockchain-Based IoT-Enabled E-Waste Tracking and Tracing System for Smart Cities. *IEEE Access* **2022**, *10*, 86256–86269. [\[CrossRef\]](#)
24. Dharma Putra, G.; Kang, C.; Kanhere, S.S.; Won-Ki Hong, J. DeTRM: Decentralised Trust and Reputation Management for Blockchain-based Supply Chains. In Proceedings of the 2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Shanghai, China, 2–5 May 2022; pp. 1–5. [\[CrossRef\]](#)
25. Yang, Q.; Wang, H. Privacy-Preserving Transactive Energy Management for IoT-Aided Smart Homes via Blockchain. *IEEE Internet Things J.* **2021**, *8*, 11463–11475. [\[CrossRef\]](#)
26. Qashlan, A.; Nanda, P.; He, X.; Mohanty, M. Privacy-Preserving Mechanism in Smart Home Using Blockchain. *IEEE Access* **2021**, *9*, 103651–103669. [\[CrossRef\]](#)
27. Rahman, A.; Nasir, M.K.; Rahman, Z.; Mosavi, A.S.S.; Minaei-Bidgoli, B. DistBlockBuilding: A Distributed Blockchain-Based SDN-IoT Network for Smart Building Management. *IEEE Access* **2020**, *8*, 140008–140018. [\[CrossRef\]](#)
28. Fan, M.; Zhang, X. Consortium Blockchain Based Data Aggregation and Regulation Mechanism for Smart Grid. *IEEE Access* **2019**, *7*, 35929–35940. [\[CrossRef\]](#)
29. Yazdinejad, A.; Dehghantanha, A.; Parizi, R.M.; Hammoudeh, M.; Karimipour, H.; Srivastava, G. Block Hunter: Federated Learning for Cyber Threat Hunting in Blockchain-Based IIoT Networks. *IEEE Trans. Ind. Inform.* **2022**, *18*, 8356–8366. [\[CrossRef\]](#)
30. Gai, K.; Tang, H.; Li, G.; Xie, T.; Wang, S.; Zhu, L.; Choo, K.K.R. Blockchain-Based Privacy-Preserving Positioning Data Sharing for IoT-Enabled Maritime Transportation Systems. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 2344–2358. [\[CrossRef\]](#)
31. Cui, J.; Ouyang, F.; Ying, Z.; Wei, L.; Zhong, H. Secure and Efficient Data Sharing among Vehicles Based on Consortium Blockchain. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 8857–8867. [\[CrossRef\]](#)
32. Ahmed, A.; Abdullah, S.; Iftikhar, S.; Ahmad, I.; Ajmal, S.; Hussain, Q. A Novel Blockchain Based Secured and QoS Aware IoT Vehicular Network in Edge Cloud Computing. *IEEE Access* **2022**, *10*, 77707–77722. [\[CrossRef\]](#)
33. Sharma, P.K.; Kumar, N.; Park, J.H. Blockchain-Based Distributed Framework for Automotive Industry in a Smart City. *IEEE Trans. Ind. Inform.* **2019**, *15*, 4197–4205. [\[CrossRef\]](#)
34. Ren, B.; Yang, L.T.; Zhang, Q.; Feng, J.; Nie, X. Blockchain-Powered Tensor Meta-Learning-Driven Intelligent Healthcare System With IoT Assistance. *IEEE Trans. Netw. Sci. Eng.* **2023**, *10*, 2503–2513. [\[CrossRef\]](#)
35. Li, J.; Li, D.; Zhang, X. A Secure Blockchain-Assisted Access Control Scheme for Smart Healthcare System in Fog Computing. *IEEE Internet Things J.* **2023**, *10*, 15980–15989. [\[CrossRef\]](#)
36. Wang, Y.; Zhang, A.; Zhang, P.; Qu, Y.; Yu, S. Security-Aware and Privacy-Preserving Personal Health Record Sharing Using Consortium Blockchain. *IEEE Internet Things J.* **2022**, *9*, 12014–12028. [\[CrossRef\]](#)
37. Bataineh, M.R.; Mardini, W.; Khamayseh, Y.M.; Yassein, M.M.B. Novel and Secure Blockchain Framework for Health Applications in IoT. *IEEE Access* **2022**, *10*, 14914–14926. [\[CrossRef\]](#)
38. Nguyen, D.C.; Pathirana, P.N.; Ding, M.; Seneviratne, A. A Cooperative Architecture of Data Offloading and Sharing for Smart Healthcare with Blockchain. In Proceedings of the 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Sydney, Australia, 3–6 May 2021; pp. 1–8. [\[CrossRef\]](#)
39. Zhao, Y.; Qu, Y.; Xiang, Y.; Zhang, Y.; Gao, L. A Lightweight Model-Based Evolutionary Consensus Protocol in Blockchain as a Service for IoT. *IEEE Trans. Serv. Comput.* **2023**, *16*, 2343–2358. [\[CrossRef\]](#)
40. Gadiraju, D.S.; Lalitha, V.; Aggarwal, V. An Optimization Framework Based on Deep Reinforcement Learning Approaches for Prism Blockchain. *IEEE Trans. Serv. Comput.* **2023**, *16*, 2451–2461. [\[CrossRef\]](#)
41. Wang, C.; Jiang, C.; Wang, J.; Shen, S.; Guo, S.; Zhang, P. Blockchain-Aided Network Resource Orchestration in Intelligent Internet of Things. *IEEE Internet Things J.* **2023**, *10*, 6151–6163. [\[CrossRef\]](#)

42. Badruddoja, S.; Dantu, R.; He, Y.; Upadhyay, K.; Thompson, M. Making Smart Contracts Smarter. In Proceedings of the 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Sydney, Australia, 3–6 May 2021; pp. 1–3. [\[CrossRef\]](#)
43. Badruddoja, S.; Dantu, R.; He, Y.; Thompson, M.; Salau, A.; Upadhyay, K. Making Smart Contracts Predict and Scale. In Proceedings of the 2022 Fourth International Conference on Blockchain Computing and Applications (BCCA), San Antonio, TX, USA, 5–7 September 2022; pp. 127–134. [\[CrossRef\]](#)
44. Barwise, J.; Perry, J. The Situation Underground. In *Situations and Speech Acts*; Barwise, J., Sag, I.A., Eds.; Stanford Working Papers in Semantics, Stanford, Cognitive Science Group, Part D; Taylor & Francis: Abingdon, UK, 2016.
45. Endsley, M.R. Design and Evaluation for Situation Awareness Enhancement. *Proc. Hum. Factors Soc. Annu. Meet.* **1988**, *32*, 97–101. [\[CrossRef\]](#)
46. Schilit, B.; Adams, N.; Want, R. Context-Aware Computing Applications. In Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, USA, 8–9 December 1994; pp. 85–90. [\[CrossRef\]](#)
47. Baldauf, M.; Dustdar, S.; Rosenberg, F. A Survey on Context-Aware Systems. *Int. J. Hoc Ubiquitous Comput.* **2007**, *2*, 263. [\[CrossRef\]](#)
48. Moore, P.; Hu, B.; Zhu, X.; Campbell, W.; Ratcliffe, M. A Survey of Context Modeling for Pervasive Cooperative Learning. In Proceedings of the 2007 First IEEE International Symposium on Information Technologies and Applications in Education, Kunming, China, 23–25 November 2007; pp. K5-1–K5-6. [\[CrossRef\]](#)
49. Ejigu, D.; Scuturici, M.; Brunie, L. An Ontology-Based Approach to Context Modeling and Reasoning in Pervasive Computing. In Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07), White Plains, NY, USA, 19–23 March 2007; pp. 14–19. [\[CrossRef\]](#)
50. Sezer, O.B.; Dogdu, E.; Ozbayoglu, A.M. Context-Aware Computing, Learning, and Big Data in Internet of Things: A Survey. *IEEE Internet Things J.* **2018**, *5*, 1–27. [\[CrossRef\]](#)
51. Chen, R.; Wang, X. Situation-Aware Orchestration of Resource Allocation and Task Scheduling for Collaborative Rendering in IoT Visualization. *IEEE Trans. Sustain. Comput.* **2022**, *7*, 935–949. [\[CrossRef\]](#)
52. Hamrouni, A.; Khanfor, A.; Ghazzai, H.; Massoud, Y. Context-Aware Service Discovery: Graph Techniques for IoT Network Learning and Socially Connected Objects. *IEEE Access* **2022**, *10*, 107330–107345. [\[CrossRef\]](#)
53. Chang, C.K.; Ceravolo, P.; Chang, R.N.; Helal, S.; Jin, Z.; Liu, X.; Ming, H. Software Services Engineering Manifesto-A Cross-Cutting Declaration. In Proceedings of the 2021 IEEE International Conference on Web Services (ICWS), Chicago, IL, USA, 5–10 September 2021; pp. 703–709. [\[CrossRef\]](#)
54. Xie, H.; Yang, J.; Chang, C.K.; Liu, L. A Statistical Analysis Approach to Predict User's Changing Requirements for Software Service Evolution. *J. Syst. Softw.* **2017**, *132*, 147–164. [\[CrossRef\]](#)
55. Gholami, H.; Chang, C.K.; Aduri, P.; Ma, A.; Rekabdar, B. A Data-Driven Situation-Aware Framework for Predictive Analysis in Smart Environments. *J. Intell. Inf. Syst.* **2022**, *59*, 679–704. [\[CrossRef\]](#)
56. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning Internal Representations by Error Propagation, Parallel Distributed Processing, Explorations in the Microstructure of Cognition. *Biometrika* **1986**, *71*, 599–607.
57. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [\[CrossRef\]](#) [\[PubMed\]](#)
58. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* **2014**, arXiv:1412.3555.
59. Alshammari, N.; Alshammari, T.; Sedky, M.; Champion, J.; Bauer, C. OpenSHS: Open Smart Home Simulator. *Sensors* **2017**, *17*, 1003. [\[CrossRef\]](#)
60. Blender-a 3D Modelling and Rendering Package. Available online: <https://www.blender.org/about/> (accessed on 18 May 2024).
61. Alshammari, T.; Alshammari, N.; Sedky, M.; Howard, C. SIMADL: Simulated Activities of Daily Living Dataset. *Data* **2018**, *3*, 11. [\[CrossRef\]](#)
62. Szilágyi, P. EIP-225: Clique Proof-of-Authority Consensus Protocol; Ethereum Improvement Proposals no. 225. [Online Serial]. Available online: <https://eips.ethereum.org/EIPS/eip-225> (accessed on 18 May 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.