BLOCKCHAIN-BASED FEDERATED LEARNING WITH SMART CONTRACT FOR FRUIT WASTE PREDICTION

LEE CHUN ON ALISDAIR

MPhil

The Hong Kong Polytechnic University

2025

The Hong Kong Polytechnic University
Department of Electrical and Electronic Engineering

Blockchain-based Federated Learning with Smart Contract for Fruit Waste Prediction

Lee Chun On Alisdair

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Philosophy

September 2024

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____

Lee Chun On Alisdair

_____

List of Publications

International journal paper:

1. C. A. Lee, K. M. Chow, H. A. Chan, and D. P. K. Lun, "Decentralized governance and artificial intelligence policy with blockchain-based voting in federated learning," *Frontiers in Research Metrics and Analytics*, vol. 8, 1035123, 2023.

Acknowledgments

I would like to thank my supervisors and supporters, Doctor Pak-Kong Lun Daniel, Doctor Y. L. Chan, Professor H. A. Chan, and Professor Wan-Chi Siu for their support throughout my study.

Abstract

Fruit losses in the supply chain owing to improper handling and a lack of proper control are common in the fruit logistics industry. As losses are caused by the inefficiency of the export method, selecting the appropriate export method is a possible solution. Several organizations employ only a single strategy, which is mainly based on a first-in-first-out approach. Such a policy is easy to manage but inefficient. Given that the batch of fruits may become overripe during transportation, frontline operators do not have the authority or immediate support to change the fruit dispatching strategy. This study aims to develop a dynamic strategy simulator to determine the sequence of delivery based on forecasting information projected from probabilistic data to reduce the amount of fruit loss.

The proposed method using asynchronous federated learning (FL) is based on blockchain technology and a serially interacting smart contract. In this method, each party in the chain updates its model parameters and uses a voting system to reach a consensus. This study employs blockchain technology with smart contracts to serially enable asynchronous FL, with each party in the chain updating its parameter model. A smart contract combines a global model with a voting system to reach a common consensus. Its artificial intelligence (AI) and Internet of Things engine further strengthen the support for implementing the Long Short-Term Memory (LSTM) forecasting model. Based on AI technology, a system was constructed using FL in a decentralized governance AI policy on a blockchain network platform. With mangoes being selected as the category of fruit in the study, the system improves the cost-effectiveness of the mango supply chain. In the proposed approach, the simulation results show fewer mangoes lost (from 6.91% to 0.035%) and operational costs reduced.

Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Motivation

In the past 5 years, the majority of cargoes and shipments have employed a first-in-first-out (FIFO) allocation strategy that does not involve using sensor devices or shelf-life knowledge. Consequently, the mango market experienced significant maturation losses. As a member of the category of perishable fruits, mango has a special organoleptic characteristic that is subject to continuous change throughout supply chain activities. The problem of overripe fruit spreads from upstream to downstream. Mango losses and quality problems are related to pre-harvest conditions and post-harvest management of the supply chain structure and operations [1]. Such losses and quality problems have been traced back to farm operations, such as picking, washing, and drying, and logistics operations, such as packing, storing, and exporting. The majority of losses occur at the distribution stage owing to frequent handling complications, unexpected events, transportation conditions, and operational practices. In particular, when mangoes are transported over long distances and for lengthy transit periods along global supply chains, the issues of fruit loss and quality become more severe [2]. Therefore, logistics and inventory management are considered the primary causes of low acceptance rates and fresh fruit losses during the retail and consumption stages [3]. For instance, Ridolfi *et al.* [4] reported that mango losses amount to 45.6% of the total production. Mango losses can be substantial, ranging from 19 to 46% of the crop. The losses incurred because of transport, storage, or over-ripening in different countries are listed as follows:

- Bangladesh: 8.2% out of 25.5% [5].
- Ghana: 29.6% out of 45.6% [4].
- Philippines: 3.5–4.9% out of 19.0–33.9% [6].
- India: 9.6% out of 34.5% [7].

• Ethiopia: 13.4–15.7% out of 40.7% [8] [9].

Moreover, the Queensland Government [10] found that the sources of mango losses start at the farm level and progress to the post-harvest level. Other sources of loss are in the wholesale and retail stages, primarily attributable to inadequate storage facilities and operations, low-tech packaging methods and materials used, a lack of cold chain containers and trucking facilities, and inadequate road maintenance and infrastructure networks [11] [12]. The shelf-life of mangoes can be reliably extended when stored over the supply chain cycle within the recommended temperature management ranges of the cool chamber: 10 to 12°C for storage, 12–16°C for transport, and 18–22°C for ripening [10]. However, farmers, logistics providers, warehouses, wholesalers, and retailers do not follow this practice.

1.1.1 Fruit logistics functionalities

The methodology proposed in this thesis collects status information about mangoes, such as export progress, maturity, temperature, and humidity. These are used to construct a scenario that enables better logistics management. The sensors detect the fruit maturity loss [13] using the “Export Progress” information, and the export progress loss is then estimated. Hence, the industry should stimulate advanced management ideas on strategy simulation and recoup a part of the sensor cost while increasing fruit quality [14]. Using the “maturity” information, a mango ripeness data feed is essential to train a self-supervised deep learning model to predict shelf-life [15]. It provides deeper insight into autonomous fruit supply chain management and decision-making. Using the “Temperature and Humidity” information, the maturity datasets are categorized into different relative humidity and temperature ranges [16]. When real-time temperature and humidity are within the range of the datasets, there is a strong correlation between actual and predicted ripeness.

1.1.2 Logistics prediction through federated learning

Developing a decisive model for destination selection requires a large amount of training data to produce an effective model. If all of the data are collected from a single source, the size of the dataset will be insufficient for the training process. However, if data are collected from different suppliers, there are security problems, update difficulties, and the risk of node failure. Therefore, federated learning (FL) was used in this study. FL aims to achieve the best model without exchanging data with individuals [17]. To eliminate the problem of requiring a permission server for FL [18], a serverless approach was used in the FL implementation. This machine-learning model derives from decentralized governance. Many parties can then use this machine learning model to develop scalable AI policy simulations.

1.1.3 Decentralized Governance and AI Policy (DGAP)

As mentioned above, traditional FL depends on a central server for model aggregation. The current method poses risks related to single points of failure and limited scalability. Prediction models not backed by blockchain are vulnerable to data tampering. Once the current model changes, other models cannot absorb the knowledge from the previous scenario's model. The non-smart-contract-based predictive models would not obtain the optimized overall accuracy due to a lack of scenario variety. The centralized server may store different scenario prediction models; however, the trustworthiness of predictions may be questioned without a decentralized and transparent framework. In this thesis, we propose DGAP to solve the existing federated learning problems for the supply chain. Establishing transparency provided by smart contracts with an immutable voting system is important. Smart contracts are programs stored on the blockchain that execute when predetermined conditions are met. The smart contract can then act as a predictive-driven serverless approach. The serverless FL allows for greater scalability. The smart

contract accommodates a larger number of participants without performance degradation to extend the usage possibility of each local model. The more predictive scenarios are, the higher the overall accuracy is, and the longer the agent's sustainability is. The fruit waste can, therefore, be reduced.

DGAP is a double helix framework to integrate federated learning with multiple participants using blockchain-based voting mechanisms. Decentralized Governance allows multiple parties to collaboratively train AI models without sharing their raw data. Each party updates its local model through a smart contract. Blockchain stores critical parameters, making the models tamper-proof and traceable. Multiple AI policies in agent decision making system ensure that one of the policies is beneficial for each supply chain stakeholder. The system allows stakeholders to vote for an optimal prediction model highly compatible with agent strategic planning. The decentralized system automates the execution of decisions and ensures that all parties adhere to the agreed-upon rules. The voting mechanism is achieved within smart contracts about the fusion of model aggregation. These smart contracts are pivotal in facilitating model selection and enhancing fault tolerance through consensus.

1.1.4 Benefits of federated learning, decentralized governance, and AI policy

In this study, federated learning as well as decentralized governance and AI policy (DGAP) benefit the supply chain in five ways:

- **Ensuring the responsible and ethical use of AI:** AI systems are used in a fair, transparent, and respectful way. This can help build public trust in AI and mitigate the potential negative consequences of its use.
- **Promoting economic growth and development:** AI policy supports fruit saving, in addition to the development and deployment of AI technologies that drive economic growth

and create new opportunities for workers and businesses.

- **Protecting privacy and security:** federated learning can help safeguard the privacy and security of individuals by establishing standards for the protection of personal data.
- **Facilitating innovation and research:** supply chain policy managers can encourage the development of innovative AI technologies and support fruit waste research to help advance the field.
- **Enhancing public awareness and understanding:** the prediction model informs the public about the capabilities and limitations of the current supply chain system and its potential impacts on logistics.

1.2 Aim of this work

In traditional logistic management, fruit loss information is collected at the beginning and end of the transportation route, and information on fruit ripeness throughout the journey is unknown. However, with a system simulating the mango allocation strategy, decision-makers can obtain recommendations and receive guidance generated with decentralized governance by artificial intelligence (AI) policy [19]. The simulation promotes an impact that influences peers and forms a consensus when making policy decisions. Without export simulation, it is difficult to determine the amount of fruit wasted in the company. It is also difficult for storeroom workers to execute a single strategy to address the supply chain issue. The proposed system collects data, shares experiences, learns from quick trials, transforms wisdom into a global AI model, and broadcasts it to a permanent public network. This study aims to (1) propose scene simulation, (2) launch a permanent intelligence broadcaster, and (3) improve the cost-effectiveness of the fruit supply chain, as presented in the analysis of Indonesia's most important fruit commodities [20]. Because the agricultural supply chain industry has to handle perishable production, planting, growing, and harvesting processes that depend on climate and

season, and yields of varying shapes and sizes, such management is more complex than that for non-agricultural supply chains.

1.3 Organization of the Thesis

This thesis is organized as follows: Chapter 2 provides the background of federated learning and fruit waste detection, focusing on the concept of the FIFO mechanism, permanent intelligence broadcaster, and blockchain-based federated learning. Chapter 3 describes the three-process flow-integration design and blockchain-based AI models, addressing the lack of digital information using reproduction technologies for individual scenes, such as stochastic simulations of fruit loss. Chapter 4 concludes the study and discusses real-world industrial applications that could limit the interpretability of simulation outcomes.

Chapter 2 Related Work

2.1 The development of federated learning

Historically, the widespread adoption of FL services grants considerable leverage to centralized companies. Initially dominated by technology enterprises, the market for FL methods garnered success with substantial support from mobile phone users during its initial phase. As shown in Figure 2.1, the FL process involves four steps: 1) the central server selects a statistical model for training, 2) transmits the initial model to nodes, 3) nodes locally train the model with their data, and 4) the central server aggregates results to generate a global model without accessing any raw data.

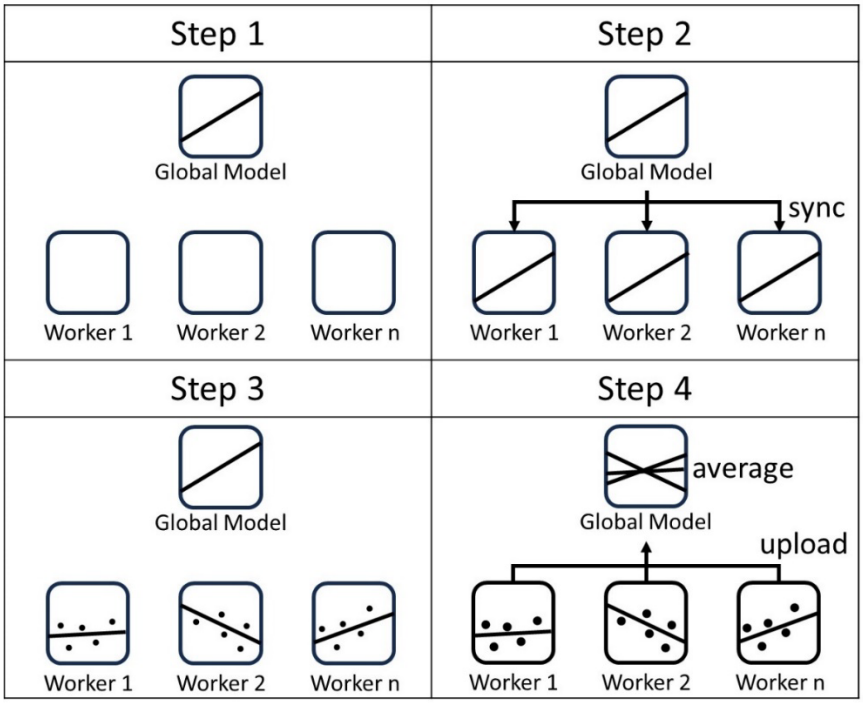

Figure 2.1 Traditional FL Process

The concept of FL was introduced by Google in a pivotal paper titled "Communication-Efficient Learning of Deep Networks from Decentralized Data," published in 2016 by Google AI researchers [21]. That seminal work aimed to address the critical issue of preserving raw data privacy while harnessing the collective power of decentralized information for machine

learning. The paper outlined a visionary approach wherein a large-scale distributed neural network model training framework was proposed. This framework enabled smartphone users to enjoy the benefits of advanced machine learning services without compromising the privacy of their local data. The key innovation is in the strategy of uploading model weights rather than raw data. By transmitting only the model's learned parameters, users could contribute to the collective intelligence without exposing sensitive information, thus striking a balance between personalized service experiences and data privacy.

FL scientists, subsequently, recognized the omission of transfer learning in their method, leading to the development of on-device FL [22]. This version allows algorithms to be trained across distributed and isolated data sets with interactive subspaces. In 2017, Google focused on on-device FL to enhance energy efficiency during training, leading to the development of Horizontal Federated Learning (HFL) [23]. FL on the device involves individual mobile phones making personalized enhancements to the model based on user usage. A comprehensive plan for modifying the overall model is then formulated and applied to shared models, with the process repeating in a continuous loop. As shown in Figure 2.2, HFL involves the transmission of encrypted gradients, secure aggregation, sending back model updates, and updating models. This approach is particularly applicable to scenarios where diverse characteristics, such as mobile phone models, usage time, battery levels, and people's locations, share similar features but utilize different samples. These varying characteristics are considered, contributing to a collaborative and privacy-preserving model improvement process in HFL.

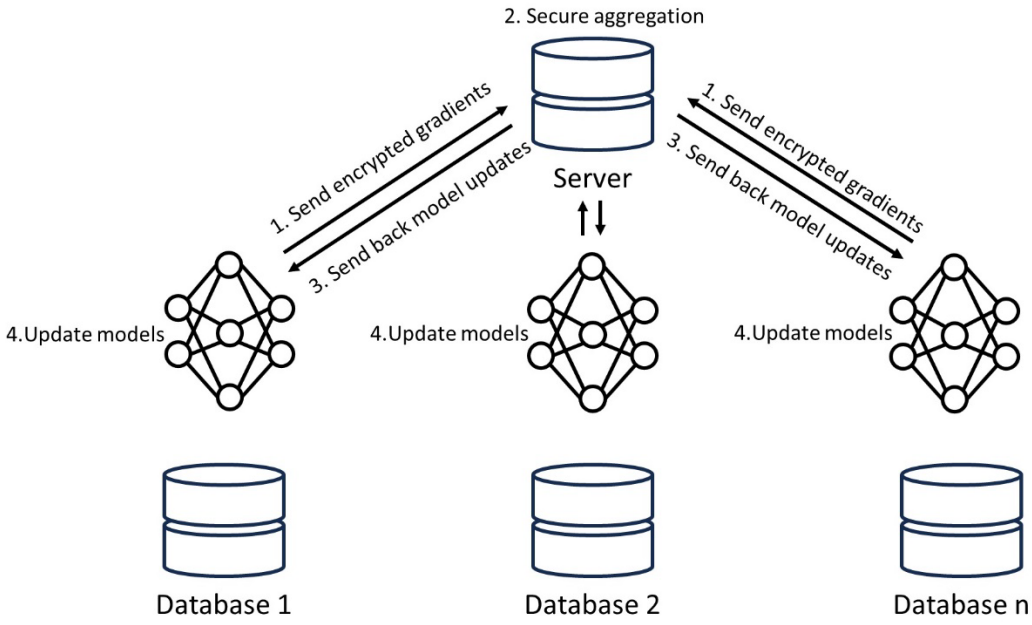

Figure 2.2 Architecture for a HFL System

However, as of 2018, the European Union introduced the General Data Protection Regulation (GDPR), which has significant implications for data privacy [24]. Under GDPR, data usage must align with the user's agreement, preventing companies from utilizing data for purposes outside the agreed scope. While user data analysis can be employed to enhance product experiences, using it to train dialogue systems without explicit consent violates GDPR provisions. The regulation also grants users the right to be forgotten, allowing them to request the removal of their data from models. Such stringent data protection requirements are not exclusive to Europe but are also enforced in California and the USA.

In 2019, researchers from the Hong Kong University of Science and Technology proposed on-device FL targeted at business applications (To B), leading to the development of Vertical Federated Learning (VFL) as shown in Figure 2.3. That approach addresses concerns related to GDPR. Unlike Google's HFL solution (To C), where the focus is on using the same data features for building identical models through the federated average algorithm, VFL is tailored for scenarios where different enterprises may have distinct data characteristics, even when

dealing with the same user scenario [25]. VFL employs encrypted entity alignment, ensuring that no data is directly exchanged between corporations A and B. Instead, only the encrypted models from entities A and B undergo joint training through a process involving the exchange of public keys, intermediate results, computation of gradients and loss, and model updates. This framework ensures collaborative model improvement without compromising sensitive data privacy.

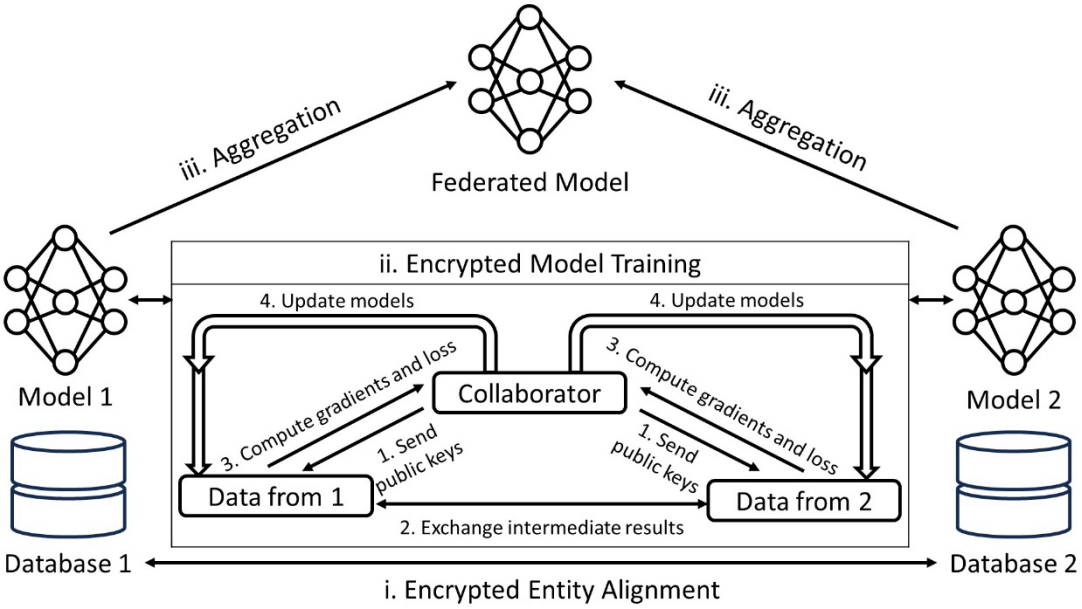

Figure 2.3 Architecture for a VFL System

In the following research, federated transfer learning (FTL) was introduced specifically focusing on transferring knowledge from a source domain to a target domain in healthcare [26]. The subspaces within this framework exhibit interactivity, and the interactions occurring among these subspaces serve as a basis for effective transfer learning. Despite the absence of direct features and user overlap between the source and target domains, FTL manages to identify commonalities that can be leveraged for transfer learning. This approach highlights the ability to extract shared insights and knowledge across distinct domains, contributing to a more versatile and adaptable FL model.

Later, the integration of FL with blockchain technology was suggested, resulting in

blockchain-based FL. This iteration incorporates adaptive differential privacy (DP) algorithms to safeguard data privacy and employs a consensus protocol based on gradient verification. Edge nodes uphold the blockchain to mitigate the risk of single points of failure, while smart devices operate FL to gather a more widespread set of clinical data. The following are the details of the FL development. In 2021, researchers directed their efforts toward implementing blockchain technology to allocate benefits within FL and data alliances [2]. That research specifically focuses on enhancing data security through integrating Blockchain and FL, emphasizing establishing trust and collaboration in sharing smart city data. As shown in Figure 2.4, the emerging field of Blockchain-based Federated Learning (BFL) incorporates an adaptive DP algorithm designed to safeguard data privacy. Firstly, the host initiates publishing a task on the blockchain, specifically within an edge node embedded in the blockchain network. Subsequently, local model training occurs, wherein IoT devices contribute data to the local training process, generating local gradients. DP noise is added to the model, and adjustments are made to the model service tailored for each IoT device. Following the local training, the gradients are uploaded from the individual devices to the blockchain. Upon reaching the blockchain, the gradient undergoes verification by a committee on the edge node. Once successfully verified, the gradient message progresses to the beacon. Finally, a new block is generated and broadcast across the blockchain network, marking the completion of this decentralized and privacy-conscious FL process.

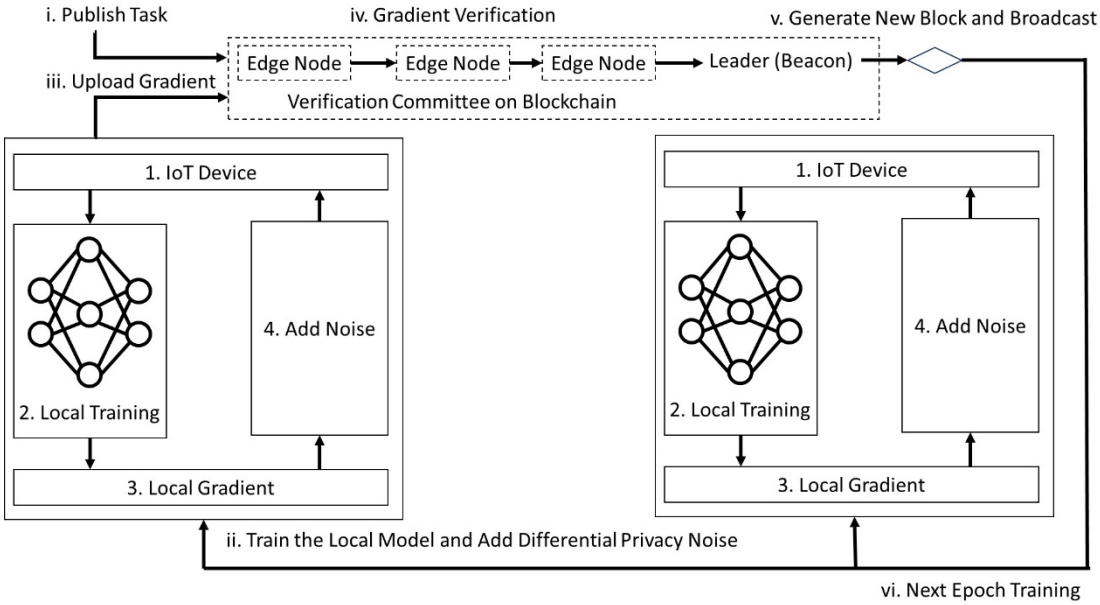

Figure 2.4 Architecture for a BFL System

2.2 The Background of Fruit Waste Detection

We examined the Indonesian mango industry as a case study to illustrate the current operational issues. Ali *et al.* designed a mini-experiment to measure the amount of wasted fruit in 2019. After an investigation of the mango industry, results revealed total postharvest losses of mangoes at different stages between harvesting and consumption (25.51%). The majority loss dominance percentage, was 6.91% [8] due to the lack of shelf-life information. The exploration of the primary industry led to the finding of one mango supplier (Anto Wijaya Fruit).

The purchase of IoT devices could not solely rely on a single source of funds because locals usually paid with cash after selling mangoes in the supply chain exchange [3]. Increasing the budget for purchasing sensing devices was critical for transferring the sustainable balance generated by preserving mangoes before they perished. We selected the TGS 2,600 and DHT11 sensors because of their high reliability as IoT components [27]. TGS 2,600 will reach the end

of its service life after 10–13 years [28]. Moreover, the DHT11 sensor provides precise temperature and humidity readings and ensures high reliability and long-term stability [30].

Apart from the cost, another key issue was the lack of sensors to detect the shelf-life of mangoes. Thus, the retailers had only been using the FIFO strategy to reduce mango loss. However, the degree of ripeness may not correlate with the arrival order. This strategy, therefore, risked reducing the shelf-life of some fruits. Ali investigated the market dominance of mango loss when using the FIFO strategy [5]. Mango failure datasets were collected between March and April 2017 and 2019 to examine the quality of mango cultivation in five districts: Rajshahi, Chapainawabgonj, Cuadanga, Meherpur, and Satkhira. It was assumed that all postharvest mangoes had approximately 4 weeks of shelf-life between collection and consumption by collectors, merchants, wholesale agents, retailers, and finally, end consumers. The retailer had to separate the fruits into different sizes and sell them within 2 to 3 days.

The FIFO strategy [29] works as follows: when the quantity on sale (X) is less than the quantity for the 1st week ($Q1$), the system displays the total profit, which is the target amount multiplied by the profit per unit for the 1st week ($X * L1$). Otherwise, it moves on to the next week when the overall goal is to subtract a smaller amount from the 1st week's amount than the number in the 2nd week ($X - Q1 < Q2$) [30]. The following are the symbol meanings:

X: the quantity on sale

$L1$: the profit per unit for the 1st week

$Q1$: the target quantity on 1st week

$Q2$: the target quantity on 2nd week

A recent survey reported mango losses in Indonesia. When 100% of the mangoes were subjected to the FIFO strategy, total mango loss due to grading and maturity issues was 6.91% [8], as follows:

- Farm level: 1% (sorting and grading), 0.95% (overripening or shriveling).
- Wholesale level: 1.2% (overripening or shriveling), 1% (immature or unmarketable size).
- Retailer level: 1.5% (overripening or shriveling), 0.26% (immature or unmarketable size).
- Storage level: 1% (overripening or shriveling).

Mango loss information was collected locally at the beginning and end stages. The timeline from unripe to ripe or overripe fruit during mango logistics is a missing piece of the puzzle, as there is neither a sensor nor a prediction throughout the logistics process [3]. It should be noted that even if the sensors are only deployed to diagnose the present mango status (early ripe, ripe, or overripe), scenario planning and high-dimensional simulation cannot be conducted without a strategy-making agent, which is enabled by a global forecasting model.

The First-in-first-out (FIFO) strategy is a widely used inventory management technique that ensures the oldest inventory items are sold first. This strategy is particularly effective in industries where products have a limited shelf life, such as food and pharmaceuticals. It simplifies inventory tracking and reduces the complexity of inventory management systems. Dynamic supply chain strategies are flexible approaches that adapt to changing conditions in real time. These strategies leverage advanced technologies such as IoT, big data analytics, and AI to optimize supply chain operations dynamically.

Table 2.1 Comparison of Perishable Inventory Management Methods.

Method	Limitation	Model	Factors			
			Estimating Waste	Saving Cost	Multiple Strategies	Collaboration
2021 [31]	$0 \leq t \leq t_{F_2}$	Differential	✓			
2020 [32]	$\theta_o < \theta_r$, $f_o < f_r$	Exponential Decay	✓	✓		

2024 [33]	Absence of Fine-Tuning	Advantage Actor-Critic	✓	✓		
Proposed	Temperature and Humidity Range	Aligned Prediction-to-decision	✓	✓	✓	✓

As shown in Table 2.1, the current FIFO mechanism in perishable supply chain management limits its flexibility for minimizing waste and saving costs, utilizing FIFO simulation and equations.

In [31], FIFO is assumed to be governed by the differential equation below to estimate inventory level vanishes in overflow warehouse due to deterioration.

$$W_{OW} = \frac{d\prod_{OW}^{FIFO}(t)}{dt} = \left[(1-\beta)\prod_{OW}^{FIFO}(t)\right] \qquad 0 \le t \le t_{F_2} \tag{2.1}$$

$$\text{Waste Percentage} = \left(\frac{W_{OW}}{I_{OW}}\right) \times 100 \tag{2.2}$$

where

OW is the overflow warehouse (warehouse with waste)

$\prod_{OW}^{FIFO}(t)$ is inventory level at time t in which the product has shortages

W_{OW} is waste in the overflow warehouse under the FIFO dispatching policy

I_{OW} is inventory in the overflow warehouse under the FIFO dispatching policy

$(1-\beta)$ is the cost of deterioration in OW under the FIFO dispatching policy

t_{F_2} is the time at which the inventory level reaches zero in OW and shortages begin

$0 \le t \le t_{F_2}$ is time interval $(0, t_{F_2})$

In [32], the following exponential decay formula is used to define the percentage of deterioration:

$$N(t) = N_0 e^{-\theta t} \tag{2.3}$$

where $N(t)$ is the remaining quantity at time t, N_0 is the initial quantity, and t is the number of periods. The hyperparameters of the equation are selected as follows:

Deterioration rate for the owner (θ_o): 0.08

Deterioration rate for the retailer (θ_r): 0.09

Holding cost for the owner (f_o): 0.4

Holding cost for the retailer (f_r): 0.5

Owner's Waste Percentage: $1 - e^{-\theta_o t} = 1 - e^{-0.08\times 1} \approx 7.69\%$

Retailer's Waste Percentage: $1 - e^{-\theta_o t} = 1 - e^{-0.09\times 1} \approx 8.61\%$

Combined Waste Percentage:

$$\frac{\text{Owner's Waste Percentage} + \text{Retailer's Waste Percentage}}{2} = \frac{7.69\% + 8.61\%}{2}$$

$$\approx 8.15\%$$

In [33], the FIFO mechanism in the perishable supply chain management operates by prioritizing the oldest inventory for order fulfillment. The following steps outline the FIFO process.

1. Demand Fulfillment: Orders are fulfilled based on the FIFO principle, where the oldest inventory is used first. Any demand that cannot be satisfied from the available inventory results in a cost, which represents the cost of unsatisfied demands.
2. Inventory Shelf-Life Update: The remaining inventory in the retailers' warehouses is revised to account for product shelf-life. Items that have expired are removed from inventory.
3. New Stock Arrival: New products arrive at the supplier's warehouse, prompting an update in inventory levels. The shelf-life countdown for these products begins immediately.
4. Cost of Expired Products: The supplier incurs a cost for any products that expire before they can be sold.

5. Optimal Inventory Allocation: The supplier allocates inventory to each retailer to minimize the total cost associated with wastages and shortages across the supply chain.

The simulation uses the Advantage Actor-Critic algorithm, which takes into the FIFO mechanism to optimize inventory allocation and minimize wastage. Equations used to model the inventory dynamics with FIFO include:

1. Objective function

$$\max_{\pi\in\Pi} E\left[-C_s \sum_{t=0}^{T} \sum_{j\in(HS1,HS2)} \max\left(0, D_t^j - I_t^{j\pi}\right) - C_W \sum_{t=0}^{T} \sum_{i\in(DC,HS1,HS2)} \left(B_{ik+t}^{\pi}\right) \right] \tag{2.4}$$

where C_S is the shortage cost per unit and C_W is the wastage cost per unit. This function seeks to maximize the negative costs, effectively minimizing them by optimizing the policy π.

2. Inventory constraint

$$\sum_{j\in(HS1,HS2)} a_t^{j\pi} \leq X_t^{\pi}, \forall t \in \{1, \dots, T\} \tag{2.5}$$

This constraint ensures that the total allocated inventory at any time t does not exceed the available inventory.

3. Fulfillment probability

$$p\left(\left|\left|\min\left(0, I_t^{j\pi} - D_t^j\right)\right| - D_t^j\right| = D_t^j\right) \geq \varepsilon, \forall j \in \{HS1, HS2\}, t \in \{1, \dots, T\} \tag{2.6}$$

This ensures that the probability of fulfilling the demand meets or exceeds a predefined threshold ϵ.

4. Inventory update equations:

$$X_{t+1}^{\pi} = X_t^{\pi} + L_t - \sum_{j\in(HS1,HS2)} a_t^{j\pi}, \forall t \in \{1, \dots, T\} \tag{2.7}$$

$$I_{t+1}^{j\ \pi} = I_t^{j\pi} + a_t^{j\pi} - D_t^j\ , \forall j \in \{HS1, HS2\}, t \in \{1, \dots, T\} \quad (2.8)$$

The equations update the inventory levels for the supplier and hospitals at each time step.

5. Non-negativity constraint:

$$\varepsilon, a_t^{j\pi} \geq 0, \forall j \in \{HS1, HS2\}, t \in \{1, \dots, T\} \quad (2.9)$$

This ensures that allocations and the probability threshold are non-negative.

For the above equations, the variables

DC is the supplier;

HS is the retailer (hospital);

i is the index of all sites $\in \{DC, HS1, HS2\}$;

j is the index of retailers $\in \{HS1, HS2\}$;

k are the age classes of the product $\in \{1, \dots, K\}$;

t is the index of the timestep $\in \{1, \dots, T\}$;

π is the distribution policy $\in \Pi$;

a_t^j are the actions (number of blood bags) proposed by the algorithm for site j at timestep t;

$a_t^{j\pi}$ are the actions (number of blood bags) proposed by the algorithm for site j at timestep t under distribution policy π;

D_t^j is the stochastic demand of site j at timestep t;

L_t is the number of inflow blood bags at timestep t;

I_t^j is the inventory level of site j at timestep t;

$I_t^{j\pi}$ is the inventory level of site j at timestep t under distribution policy π;

X_t^π is the inventory level of the supplier at timestep t under distribution policy π; and

B_{ik+t}^π is the number of blood bags of age more than k at site i at timestep t under distribution policy π.

Chapter 3 Minimization of Fruit Waste through Blockchain-based Federated Learning

3.1 Introduction

The proposed system contributes to a potential Supply Chain 4.0 for addressing the issues discussed in Chapter 1. The interpretive structural modeling (ISM) method suggested in [34] [35] transforms the operation flow of the "Anto Wijaya Fruit" Company [3] into our design for process flows covering product, finance, and risk, as outlined in the following sections.

3.1.1 Process flow of mango production

Figure 3.1 illustrates the double-helix architecture for the process flow of mango production. When farmers pick mangoes, they place them into a deployed IoT device box. The box contains the mangoes harvested at the same time. Sensing devices can thereafter collect data on temperature, humidity, and odor. These discrete data are the training materials for a local forecasting model, the Long Short-term Memory (LSTM) model. This AI model can then predict mango maturity trends to update the mango information. Its parameters can be called or uploaded through a public smart contract, combined as a global model using decentralized governance techniques, and produce probabilistic patterns. While the governance is decentralized, mango information is transferred to insight information in the AI policy. In this regard, a discussion-making agent acts based on the future relational pattern. The collector can allocate mangoes to different retailers and dealers via AI agent discussion. The choices of the policy function of Least-shelf-life-first-out (LSFO) or First-in-first-out (FIFO) in a stochastic simulation are identified by the trained evaluation net and the target net within the Deep Q-learning model. The intelligent agent assists the dealer and retailer in choosing which box to select and export to consumers. This can help keep mangoes fresh and improve the source of

insightful information.

3.1.2 Financial flow of sensing devices

This insight can potentially increase cost efficiency, reduce mango waste, and generate profit. Table 3.1 lists the Indonesian mango export value (EV) sourced from tridge.com. It should be considered that the cost of US$ 200 per sensor device includes all operating and maintenance costs for a sensor device over 10 years. Equation (3.2) evaluates the net cost efficiency ($NetCE$), which is the profit due to the reduction of mango loss, and the risk of decision-making caused by the AI agent to determine the yield and purchasing power of the IoT devices.

$$CE = L\% - R\% \tag{3.1}$$

$$NetCE = CE - \frac{N(200 + x)}{EV} \tag{3.2}$$

where L is the mango loss, R denotes risk percentage as a result of the less-than-ideal decision to reduce mango loss, N is the number of sensor devices, and x denotes the additional cost per sensing device.

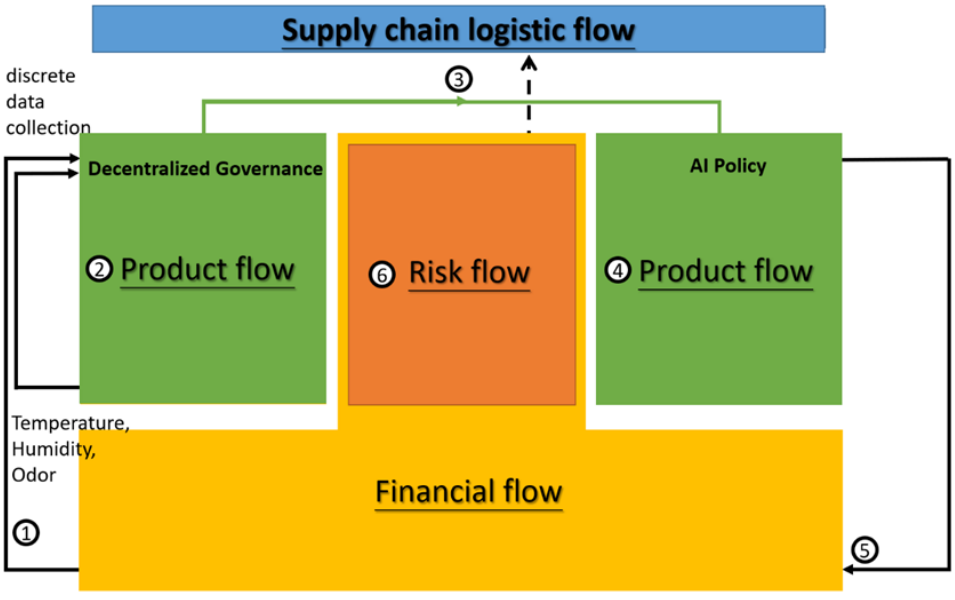

Figure 3.1 Double-helix architecture based on Anto Wijaya Fruit's (AWF) structure.

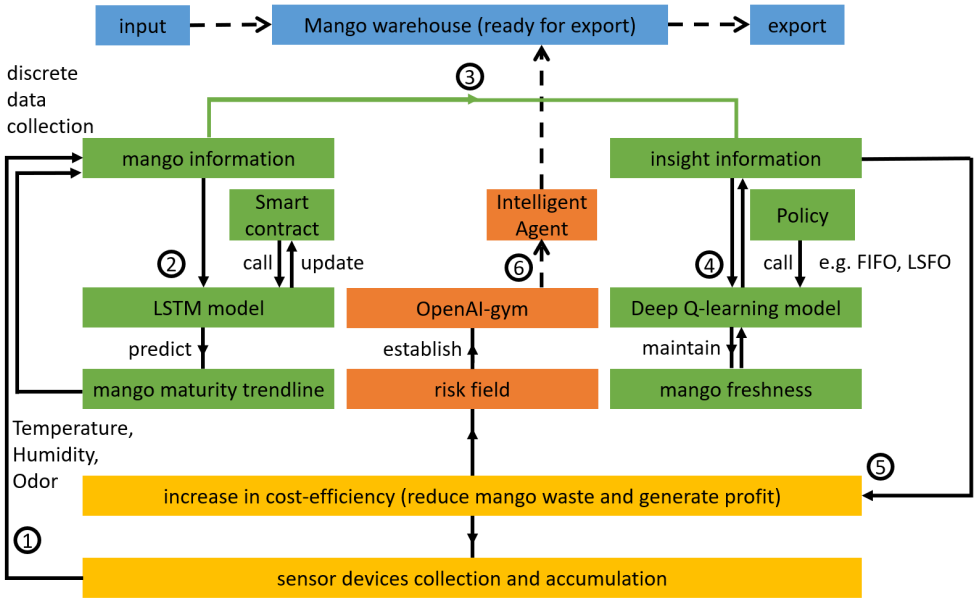

Figure 3.2 Double-helix architecture based on AWF's structure. (detail)

Table 3.1 Indonesian mango EV sourced at tridge.com and the maximum number of sensing devices (2013–2020).

Year	EV (USD)	$EV * (L - R\%)$	Budget for sensing devices (N sets)
2013	7330000	503937.5	2290
2014	8530000	586437.5	2665
2015	19260000	1324125	6018
2016	21150000	1454063	6609
2017	5310000	365062.5	1659
2018	34700000	2385625	10843
2019	44530000	3061438	13915
2020	82490000	5671188	25777

The additional cost per sensing device x in (3.2) can be obtained as,

$$x = sensor\ storage\ cost + implementation\ cost + installation\ cost + broadband\ transmission\ cost + visualization\ equipment\ cost$$

The details are as follows:

1. Sensor storage costs, S: it is postulated that the storage size of the device is only one percent of the storage size of the mango. The logistics company may absorb the sensor storage cost as a 1% increase in storage costs.

2. Implementation costs, I: the fruit supplier can pay a third-party logistics provider to implement sensor logistics.

3. Installation costs: sensors are mounted on the surface of the fruit box plate to minimize labor costs associated with inserting and removing the sensors.

4. Broadband transmission costs, T: as less detailed, data-intensive flows are required to be transmitted over the IoT network in the ocean, data transmission can use the original ship channel and the satellite communication fee can be minimized. On land, the inventory's data transmission method could use LoRaWAN and 5G on a decentralized wireless IoT network, the Helium blockchain network. Regular broadband transmission expenses are relatively cheap, costing US$ 0.00001 per every 24 bytes sent in a packet. Each device transmits the signal every minute, and their cost is approximately US$ 5, that is, six times * 24 h * 365 days * US$ 0.00001 per transmission = US$ 0.5256 per year.

5. Visualization equipment costs, V: sensor readings can be visualized using a web-based dashboard. They can also be accessed on a personal smartphone instead of using a dedicated display device.

Assuming a total additional cost per sensing device (x), A, is US$ 20 for the sensor's life span (10 years), the total upfront cost of purchasing and operating each sensor is US$ 200 + US$ 20 = US$ 220. The profit recovered by this AI system is $EV * (L\% - R\%)$, where $L =$

6.91% and $R = 0.035\%$. This could be used as the budget for paying the initial sensor cost. The maximum number of sensors that could be purchased using this profit is listed in Table 3.1.

In set theory, the total target cost C is composed of various distinct cost components, expressed as C=$S \cup I \cup T \cup V \cup A$. For example, each subset corresponds to a specific category of costs involved. The first subset, S, represents sensor storage costs, which account for 1% of mango storage costs. This is an opportunity cost incurred by replacing mango storage space with sensor storage, estimated at $120. The second subset, I, pertains to implementation and installation costs, such as mounting sensors on box plates, with an estimated cost of $50. The third subset, T, refers to broadband transmission costs, which involve data transmission and are calculated at $0.5256. The fourth subset, V, covers visualization costs, which include web dashboard equipment used for monitoring instead of dedicated displays, estimated at $30. Finally, A accounts for additional operational and maintenance costs, such as unexpected expenses, with an example cost of $20.

In blockchain-based AI systems, these subsets can sometimes overlap due to shared resources. For instance, $S \cap I$ represents costs associated with sensor placement that also impact storage, reflecting the dual purpose of sensors in optimizing both placement logistics and storage utilization. Similarly, $T \cap V$ denotes costs related to both data transfer and its visualization in dashboards, where resources are shared for transmitting data and presenting it visually through monitoring systems. By recognizing and addressing both individual cost subsets and their intersections, this approach ensures a comprehensive and efficient estimation of the total target cost C, enabling better resource allocation and cost optimization.

3.1.3 Risk flow associated with the use of AI agents

The risk flow in Figure 3.1 represents the effectiveness of launching AI to select policies

in an environment. Using an OpenAI-gym library, the system simulates mango allocation during the export period in a customized risk field. This library is a toolkit for reinforcement learning. It includes several benchmark problems that expose standard interfaces and compare algorithm performance [36]. The simulation system with this library can construct a scene of fruit logistics, such as the fruit loss process (loss at the farm level, loss due to transportation, loss at the wholesale level, loss during storage, loss at the retail level, loss at the consumer level, and loss during processing). In the simulation, the intelligent agent drives the reaction process. Only a small probability of mango loss may occur using the FIFO and LSFO policies, as shown in [4].

3.2 Blockchain-based AI models

3.2.1 LSTM AI forecasting model

The LSTM model is an artificial recurrent neural network capable of learning long-term order dependencies in data [37]. This LSTM model undergoes four main stages, in this order: preprocessing, training, testing, and evaluation. The following paragraphs explain these stages in detail: the LSTM unit comprises a cell, an input gate, an output gate, and a forget gate [38]. The cell stores values over arbitrary time intervals, while the three gates regulate the flow of information into and out of the cell. The process moves from the forget gate to the input gate and then to the output gate.

The **forget gate** uses the previously mentioned hidden state and the latest input data to determine the essential information. The previous hidden state and the latest input data are fed into a neural network that uses sigmoid activation to generate a vector in which each element is between 0 and 1. The network is trained to consider irrelevant information as 0, whereas relevant information is 1. Subsequently, the values are multiplied by the previous cell state. This process ensures that irrelevant information is multiplied by 0 and has less influence later.

Next, the **input gate** determines the unique information that is implemented in the cell state after taking into account the previous hidden state and the latest input data. The tanh-activated memory neural network generates the latest vector within a range of −1 to 1 because the neural network has already learned to combine the previous hidden state and the latest input data. After incorporating the most recent data, the generated vector indicates the extent to which each component of the cell state of the network should be updated. However, because the tanh-activated neural network does not check if it is recalling recent data, the input gate, a sigmoid-activated network with an output vector ranging from 0 to 1, is used as a filter to identify the components of the vector. Then, the output of the tanh-activated memory neural network is obtained by performing a point-wise multiplication with the input gate, and this vector is added to the cell state.

Finally, the **output gate** establishes the latest hidden state based on the newly updated cell state, the previous hidden state, and the most recent input data. The output gate uses a sigmoid-activated neural network, the previous hidden state, and the latest data to output a value from 0 to 1. This procedure ensures that only essential details are stored in the latest hidden state. However, before this process, the cell state is passed through a tanh function to output a value between −1 and 1. The tanh function output is then multiplied by the output gate to receive the current hidden state.

3.2.2 Immutable broadcasting blockchain framework

The proposed smart contract broadcasts LSTM AI model parameters. Discrete mango data owners can upload local model parameters globally, while other people can read them. To design an intelligent technology policy, the developer needs scale-free reinforcement learning to compute the execution timing, whether launching a FIFO or LSFO method when processing a mango export strategy [39]. In addition, the forecasting model and the LSTM model, which

predicts the mango maturity trend [40], supports AI decision-making by sharing its features on the blockchain.

The value of a blockchain lies in its ability to store intelligence. The smart contract aggregation feature allows for sharing AI models without needing middlemen. Spreading machine knowledge through the hive mind platform is likely to solve one of the enormous supply chain normalization and fruit maturity consensus problems. Just as the internet allows websites to spread information, smart contracts allow the broadcasting of model parameters. After data collection by sensors, the AI model has been trained using these datasets. Subsequently, the model predicts the ripeness of the 20% trend in future projections. The prediction uses the trained intelligence to generate a meaningful equation-free model [41] to measure the relative ripeness momentum. These processes allow authorized users to vote on forecasting model proposals on the blockchain, choosing whether to merge the old and new forecasting models. However, it should be noted that there is a fee for running a smart contract. Every time a smart contract is executed, a fee must be paid to the Ethereum Virtual Machine (EVM) for execution. This fee is paid to the nodes that help store, compute, execute, and validate smart contracts. EVM is known as the core of Ethereum, demonstrating its importance to the Fantom network [42] [43] (layer 1).

Layer 1 is a blockchain architectural term that refers to a network that provides infrastructure or consensus on projects, such as an event-based coffee supply chain [44]. A virtual machine (VM) is a computer system with complete hardware functions simulated by software and running in a completely isolated environment. By generating a new virtual image of the existing operating system, the VM performs the same functions as the Windows system; however, it runs independently from a Windows system. As the name suggests, the EVM is Ethereum's VM. Notably, there are no VMs in the Bitcoin blockchain [45]. The primary

function of Bitcoin is to store data in a distributed manner, and we can record, verify, store, and replicate transaction data in this network. Ethereum is a “decentralized real-world computer,” and developers can also build DApps on it, implying that Ethereum not only needs to be able to distribute data storage but also needs to run code and conduct consensus communication [46] [47]. If an account wants to execute a smart contract, the transfer will be completed according to the smart contract, and the relevant execution rules will be recorded in the data to guide the contract's operation. The network nodes execute the smart contract code every time the described transaction occurs through the EVM.

3.2.3 Deep Q-learning AI decision model

An AI agent selects the best action for the batches of mangoes based on the estimated shelf-life or the first import's mango. The program starts by defining the parameters of Deep Q-learning, and thereafter defines three classes and a function. These three classes define the environment, neural network, and Deep Q-learning, respectively, while a function runs on the main program. The parameters for Deep Q-learning are as follows: 0.9 for Epsilon, 0.9 for Gamma, 0.01 is the learning rate in an Adam optimizer, memory capacity is 3,000, Q-Network iterations are 100, batch size is 32, and episodes are 1,000. The environment class selects a random integer between 0 and 1,200 for the shelf-life state, and 480 array shapes for the shelf-life future projection. Then, it creates a store state from 1 to 480 to determine the reward. After 480 steps, the environment is reset to its original parameters and returns an array from the shelf-life state, future projection, and storage state.

3.3 Simulation process

This section explains how to operate simulated frameworks to predict and make decisions based on complete process flows.

The state of physical entities on an information platform relies on digital twin technology.

Figure 3.3 shows two underripe mangoes inside an A4 paper box equipped with TGS 2,600 and DHT11 sensors. A Raspberry Pi collects data from these two sensors. The box remained closed for 6 days until the odor reading rose from 3 (underripe) to 7.5 (overripe).

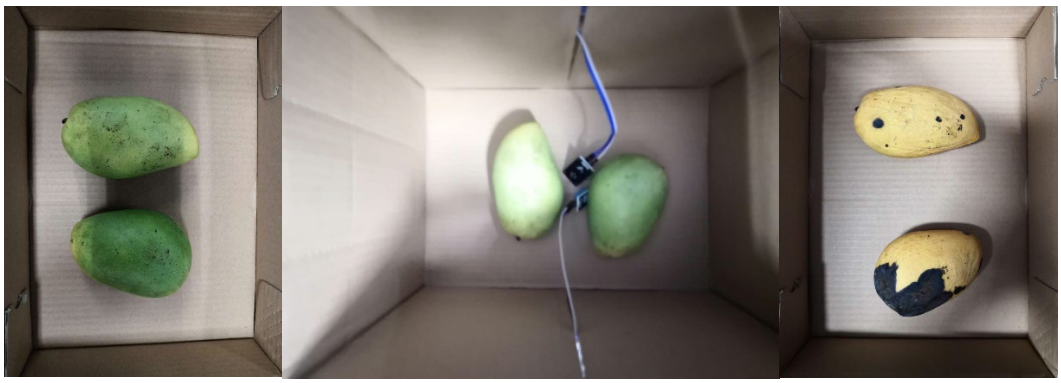

Figure 3.3 Underripe mangoes (left), and overripe mangoes (right).

The collected data are discrete and extensive. The machine first uses the iloc function to rescale the original dataset into 2,400 sampling points and process a regression method. The shelf-life of mangoes (from underripe to overripe) is irreversible. Because the reading includes the fluctuation property, it leads to an irregular shifting up and down in the shelf-life level. Therefore, a maximum function is added to address this issue by comparing the back-and-forth difference between the two frames each time.

In the regression stage, a data frame is created for odor data. It is defined using a Gaussian process model [48] [49]. The likelihood and model are initialized, and the optimal model hyperparameter is determined. The Adam optimizer uses the gaussian likelihood parameters [50] and runs 40 training iterations. Following that, the model and likelihood are evaluated. The predictions can then be made by feeding the model through likelihood. For instance, a data frame and graph are created using data after regression. In the feature selection stage, 2,400 sample points are taken from the data frame and converted into a new one.

3.3.1. Forecasting process

The proposed LSTM neural network has a feature size of one hidden unit, one output, and

one layer of LSTM to stack. A small odor value is made into a graph that represents the early ripeness of the fruit. The odor dataset is resized to 2,400 frames. This dataset is used to produce another graph that plots a portion of the original dataset. Subsequently, 80% of the data from the dataset is used for training, whereas 20% is used for testing. The training data are divided into five batches. The dataset is used in the LSTM model, which has one feature size, 16 hidden units, and a maximum epoch of 10,000. After the training stage, the testing stage begins by switching the LSTM model to the testing model. The prediction on the test dataset is made by setting the batch and feature sizes to 1. When the testing section is finished, it is plotted on a graph, and a root-mean-square deviation (RSME) is provided. Finally, the local LSTM model parameters broadcast their application to a public network in four steps:

1. Upload to an EVM-compatible smart contract on the Fantom network.
2. Interact with a global model.
3. Group into a global model.
4. Generate a shelf-life future projection for deep Q-Learning.

3.3.2 Immutable AI model broadcasting process

Several local models can be grouped into global ones through voting. In terms of the feature aggregator (FL), the voting system presents an EVM-compatible smart contract program for FL. Voting is a program that runs on the blockchain. This allows authorized voters to vote for proposals when they fulfill the program conditions. A solidity voting contract has two structures: a constructor and six functions. It also includes some lines of code that store the chairperson's address, the voter's address, and proposals. The following paragraphs explain voting contracts in detail.

The proposed smart contract, named “model parameter,” is a collection of functions and LSTM model information (its state). A deployed contract resides at a specific address on the

Fantom network, which is an EVM-compatible blockchain. This smart contract is divided into two parts: the constructor and the function. Similar to several class-based object-oriented languages, a constructor is a special function that is executed only when a contract is created and is used to initialize the contract's data. A public function accepts a string argument and updates the "parameters" storage variable. State variables, or "parameters," are variables whose values are stored permanently in the contract storage. The keyword "public" makes the variable available outside of the contract and creates a function that other contracts or clients can use to access the LSTM model parameters for simulation initiation.

The contract has two structures: "voter" and "proposal." In the "voter" structure, the voter can change their choice. To ballot the proposal, each voter has votes attributed to their unique address. The "proposal" structure stores the name of the proposal and the number of accumulated votes. These structures help store vital information about voters and proposals. The program has a constructor that runs once a ballot is launched. The constructor assigns voting rights to the chairperson. Following that, it takes the address name of a proposal in bytes-32 form and initializes the proposals with zero votes. Then, it can add proposals to an array in a voting contract.

Forecasting and broadcasting processes facilitate the development of a general simulation structure. OpenAI-gym is a simulation structure package that includes three simulation initiation functions: (1) definition, (2) step, and (3) reset. The program begins by defining the parameters of a Deep Q-learning model before launching three classes and a function. The first step of the main program defined three classes (environment, neural network, and Deep Q-learning). The following paragraphs explain the program in detail, as shown in Figures 3.4-3.9. In the internal state s_t, the shelf-life is determined by the odor of the fruits, which reflects the ripeness of the batch of samples. The Q-values from Q-Networks can then be used to select an

action to execute and observe the outcome.

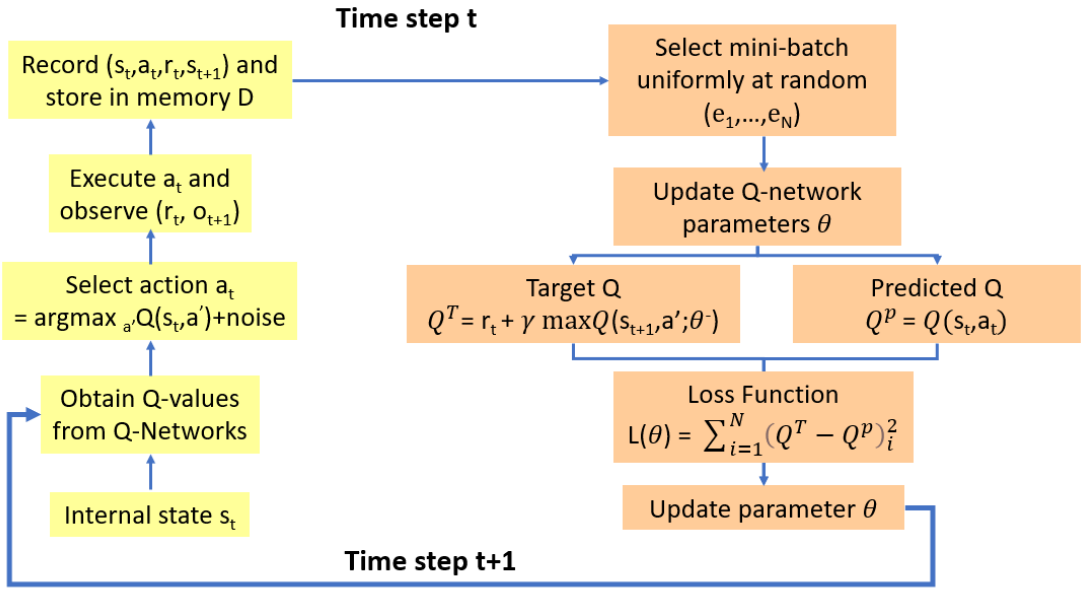

Figure 3.4 Deep Q-learning model.

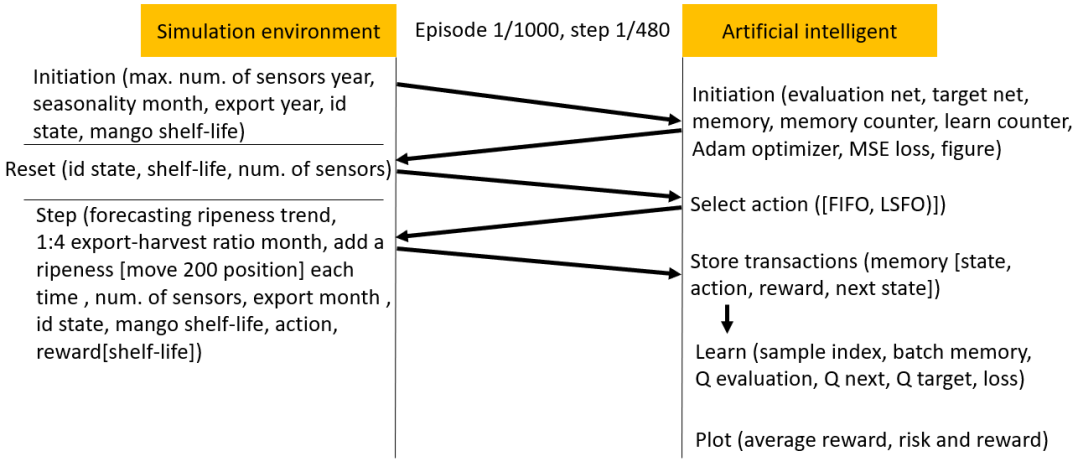

Figure 3.5 Simulation flow in episode 1, step 1.

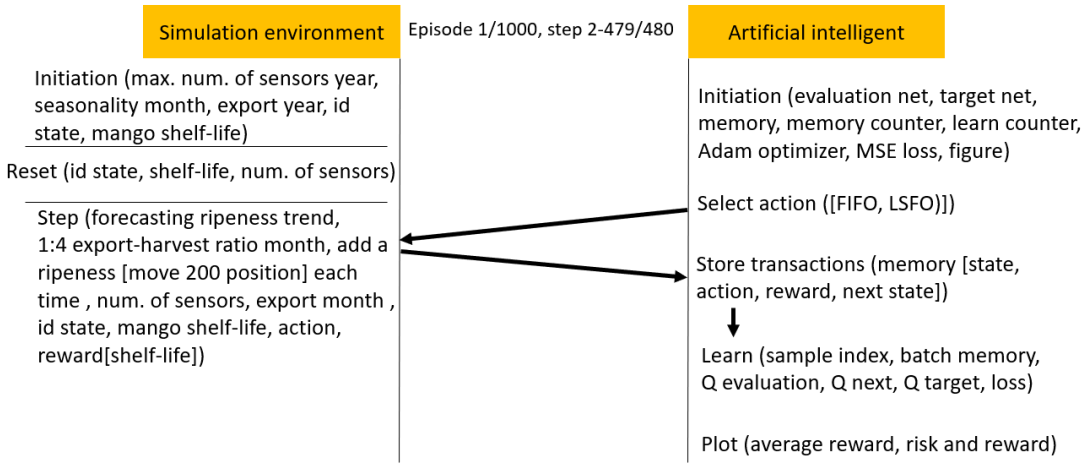

Figure 3.6 Simulation flow in episode 1, steps 2–479.

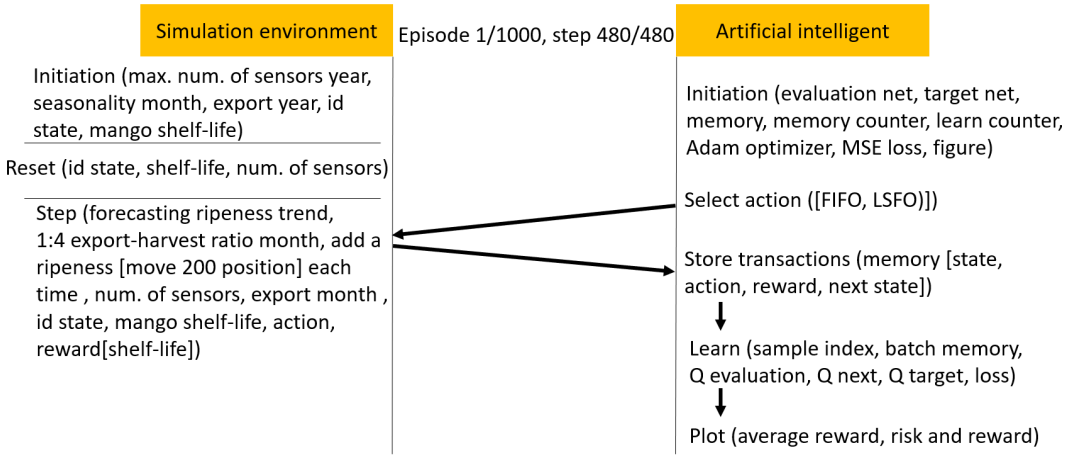

Figure 3.7 Simulation flow in episode 1, step 480.

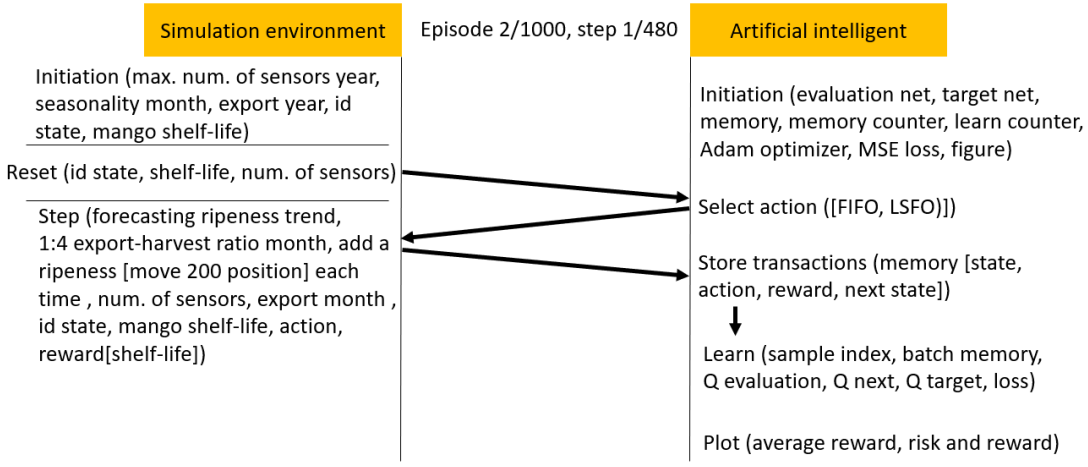

Figure 3.8 Simulation flow in episode 2, step 1.

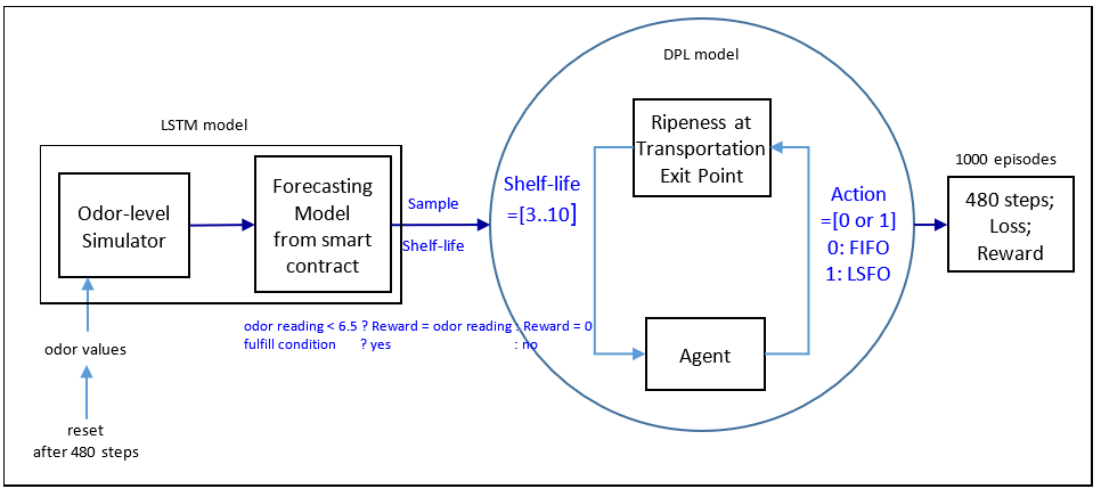

Figure 3.9 Deep Q-learning with simulated forecast scenarios.

Because the study collected 8 years of EV data, the machine used those and calculated the total steps for each episode, that is, (4 harvesting times + 1 delivery) × 12 months × 8 years = 480 steps. The deep Q-learning class starts by defining a "memory counter" as zero and a "learn counter" as zero. The deep Q-learning class has a function known as "store trans," which counts and stores a state s_t, an action a_t, a reward r_t, and a next state s_{t+1} in memory D. It also has a function known as "choose action," which determines whether to pick the new or old model. Another function in the class is the "plot" function, which charts a graph of total reward against episode.

In addition to the memory counter and the choosing action, this class has another function known as "learn." This function is called 1000 times before updating the target network. After 100 episodes, the memory can then provide an experience to help select an action in the remaining 900 episodes of the simulation. Memory has four sections: state, action, reward, and next state. Within the memory capacity, the memory counter follows a step counter to select an action based on finding the largest action value in the observation space (shelf-life future projection and storage state). When the agent selects the action value in the shelf-life future projection, it chooses to execute the LSFO policy. Alternatively, there is an option that uses

the FIFO strategy after receiving an action index. The AI agent experiences each learning process to determine an optimal strategy.

In every learning cycle, the learn counter follows the step counter to obtain batch memory. In each episode, a new data frame is created by passing the odor data frame through the reset function of Deep Q-learning. When the step counter is zero, the "main" function also creates memory slots to concatenate the data from the array created by the Deep Q-learning class. The memory slot is then passed to the "choose action" function for digit-field construction.

Specifically, the calculation of the Target Q (Q^T) and Predicted Q (Q^P) values on Figure 3.4 involves several sequential steps. At each time step (t), the AI agent observes the current state and selects an action using an ε-greedy policy: $a_t = \begin{cases} random\ action, with\ \varepsilon \\ arg\ max_{a'} Q(s_t, a'), with\ 1-\varepsilon \end{cases}$. The Target Q ($Q^T$) value is calculated by combining the immediate reward (r_t) and the discounted future reward for the next state (s_{t+1}). The discounted future reward is computed by evaluating the Q values for all possible actions (a') in the next state using the Target Q Network with weights (θ^-). The maximum Q value among these predictions ($\max_{a'} Q(s_{t+1}, a'; \theta^-)$) is selected and scaled by the discount factor (γ). The formula for Q^T is $Q^T = r_t + \gamma \cdot \max_{a'} Q(s_{t+1}, a'; \theta^-)$. The Predicted Q ($Q^P$) value is the Q value predicted by the Main Q Network with weights (θ) for the current state-action pair (s_t, a_t). Using the current network's weights, the predicted Q value is calculated as $Q^P = Q(s_t, a_t)$ in simplified notation. The difference between the Target Q and Predicted Q values drives the training of the Main Q Network. The loss function quantifies this difference using the Mean Squared Error (MSE): $L(\theta) = \frac{1}{N}\sum_{i=1}^{N}(Q^T - Q^P)_i^2$, where N is the batch size. Using gradient descent, the weights of the Main Q Network are updated to minimize the loss: $\theta \leftarrow \theta - \alpha \cdot \nabla_\theta L(\theta)$, where α is the learning rate. This process iteratively improves the Main Q Network, ensuring its

predictions for Q^P closely approximate Q^T.

The digit-field construction method is dependent on the prepared model parameters and logic. Figures 3.5 to 3.8 illustrate the AI agent being trained inside the digit field after having processed 1,000 episodes. In the first episode and first step, there are six segments in the simulation flow: initiation, reset, select action, step, store transactions, and learn. In the simulation environment, the initiation program sets up the maximum number of sensors, the seasonality month, the export year, the boxes' ID state, and the mango shelf-life for mango information. The evaluation net, target net, memory, memory counter, learn counter, Adam optimizer, mean-square error loss, and output figures are all machine learning settings that are part of the AI initiation program. In the simulation environment, the reset program resets the box ID state, fruit shelf-life, and number of sensors. The AI agent can then obtain the information necessary to select an action (FIFO or LSFO) and execute it in the simulation step program. The export-harvest ratio per month was assumed to be 1:4 for mango box allocation in the warehouse. Before making a decision, the ripeness sampling position of the received boxes was increased by 200 steps to simulate the ripeness speed while maintaining the testing sampling position for the forecasting model to predict mango shelf-life. The program eliminates the box ID to simulate export scenarios. The AI agent can then memorize rewards based on the mango shelf-life of exported boxes and store transactions in its memory to learn from the experience gained in this step.

The AI agent only processes two segments between two and 479 steps in the first episode: (1) select an action to respond to the step function, and (2) store transactions from the step function. At the first episode and 480th step, the simulation process is the same as it was at the first episode and first step, and the average reward, risk, and reward are plotted. In the second episode and first step, the simulation environment needs to be reset before the AI agent's

learning process can continue.

Figure 3.9 describes the Deep Q model, which integrates deep learning and Q-learning [51], and selects a mini-batch uniformly at random to update the Q-network parameter θ. The model can thereafter operate two neural networks to map the input state to action and Q-value pairs, using the same architecture but different weights. The two networks are the Q and the target networks. The target network is identical to the Q network, whereas the Q network is trained to produce the optimal state-action value. In an arbitrary number of steps, the network function replicates the weights from the Q-network to the target network. In addition to the two neural networks, it also has a component known as "Experience Replay."

Experience Replay interacts with the environment, which gathers a training sample saved as training data. The function is performed by selecting an ε-greedy action from the current state and executing it in the environment that receives a reward and the next state. It can store the observations as a sample of the training data.

In the following step, the Q and target networks are used to predict a projected Q value and a target Q value, respectively. The function is executed by taking a random group of samples from the training set and inputting them into the target and Q networks. The Q network predicts the Q value for the action by combining the current state and action of each data sample to obtain the predicted Q value, whereas the target network predicts the target Q value by taking the next state from each data sample to compute the best Q value of all possible actions that could have been taken from the state. The target Q value then becomes the target network output plus the reward.

After defining the predicted Q value and the target Q value, the difference between the two is used to determine the mean squared error loss. In the loss function, gradient descent can then be used to back-propagate the loss and update the parameter θ of the Q network. The gradient

descent concludes the processing for time step t. However, no loss or back-propagation is computed for the target network because it is not trained. In the next time step, t + 1, the processing is repeated, which allows the Q network to learn to predict more accurate Q values, while the target Q values are temporarily maintained. After an arbitrary number of steps, the weights of the Q network are copied to the target network, assisting the target network in receiving the improved weights to predict more accurate Q values. The processing can then resume as before.

3.4 Simulation outcomes

The fundamental idea behind the fruit supply chain simulation is to estimate the fidelity of mango maturity loss. Because of environmental constraints, such as temperature and humidity, the outcome of the data-driven experiment can be limited. The odor pattern is valid only when the temperature and humidity are within the recorded data boundary. Temperature data were subjected to 40 iterations. Throughout these, the temperature data experienced a decrease in loss from 2.484 to 1.721, an increase in the length scale from 0.693 to 1.384, and an increase in noise from 0.693 to 1.541, as shown in Figure 3.10. Temperature data also showed maximum and minimum values of 30.1385° and 23.2405°, respectively. The humidity data were also iterated 40 times while experiencing a decrease in loss from 11.106 to 2.766, a decrease in length scale from 0.693 to 0.152, and an increase in noise from 0.693 to 1.939, with maximum and minimum values of 84.6541 and 54.6122%, respectively, as shown in Figure 3.10.

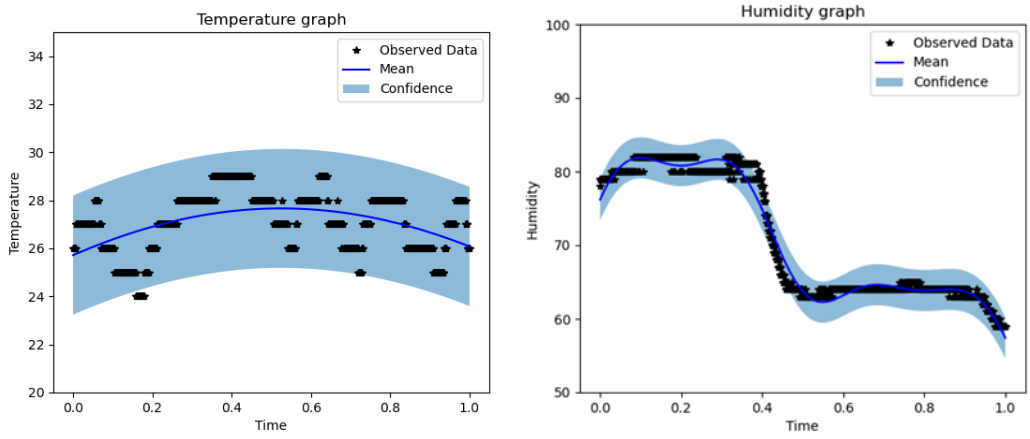

Figure 3.10 Temperature (left) and humidity (right) boundaries for the forecasting model.

In Figures 3.11 to 3.22, the shelf-life future projection uses 12 different sample dataset sizes ranging from 200 to 2,400 frames, with an increase of 200 frames each time. The shelf-life future projection maintained a loss of 1e-5. All projections contained 80% of the data for training and 20% for testing. In Table 3.2, the readings for the epoch have maximum, minimum, and average values of 7,698, 3,528, and 5,284.667, respectively. The test scores of the root mean square equation (RMSE) have maximum, minimum, and average values of 0.35, 0.15, and 0.2367, respectively. In the 200, 400, and 800 datasets, the graph reached a plateau at approximately four readings. In the 1,000 datasets, the graph reached a plateau at approximately 4.3 readings. The graph reaches a plateau at approximately five readings for the 1,200, 1,400, and 1,600 datasets. In the 1,800, 2,000, and 2,200 datasets, the graph reached a plateau at approximately six readings. In the 2,400 datasets, the graph reached a plateau at approximately seven readings.

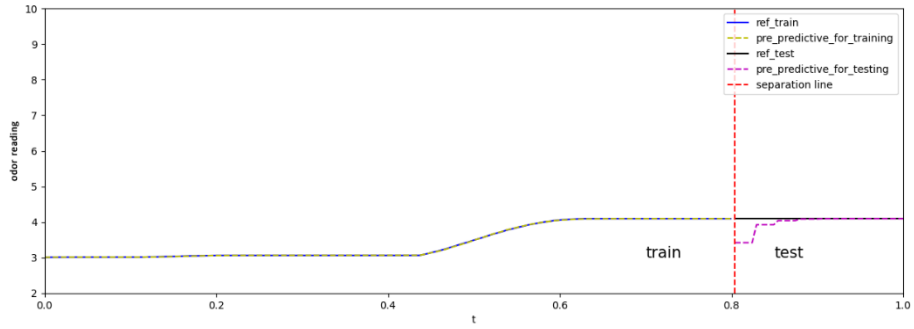

Figure 3.11 Total of 200 odor-sampling points in the forecasting model.

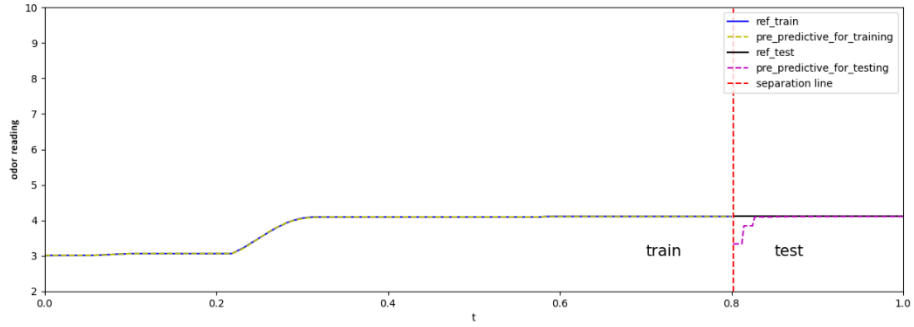

Figure 3.12 Total of 400 odor-sampling points in the forecasting model.

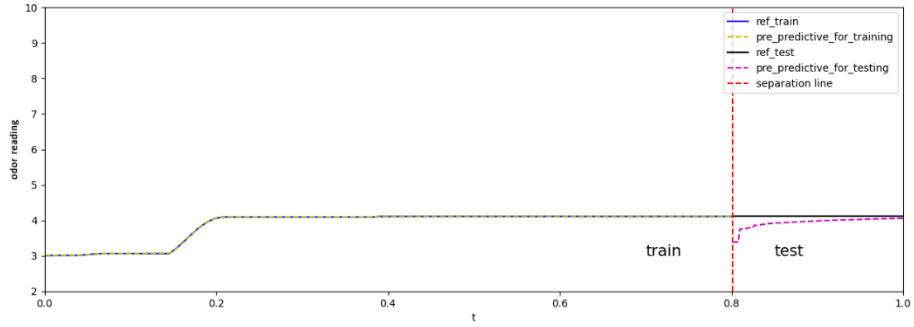

Figure 3.13 Total of 600 odor-sampling points in the forecasting model.

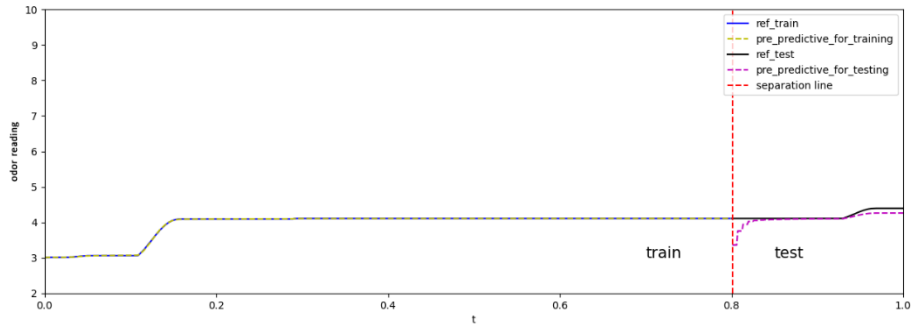

Figure 3.14 Total of 800 odor-sampling points in the forecasting model.

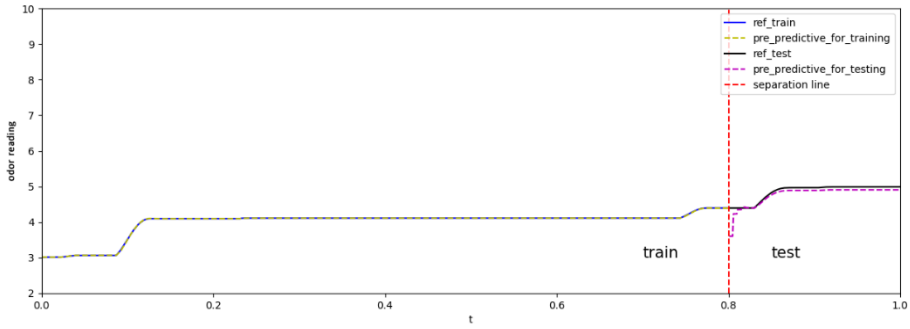

Figure 3.15 Total of 1,000 odor-sampling points in the forecasting model.

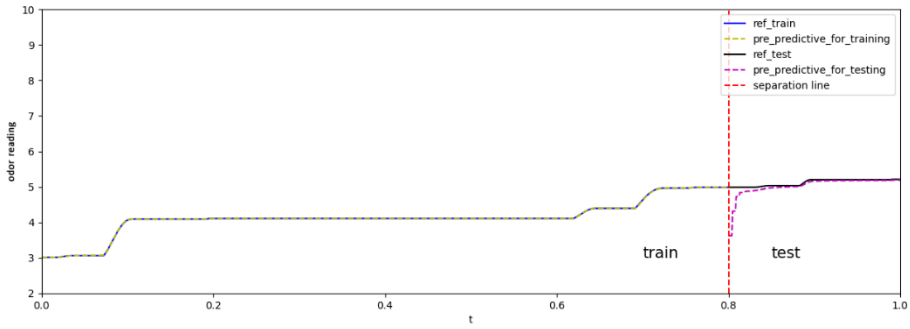

Figure 3.16 Total of 1,200 odor-sampling points in the forecasting model.

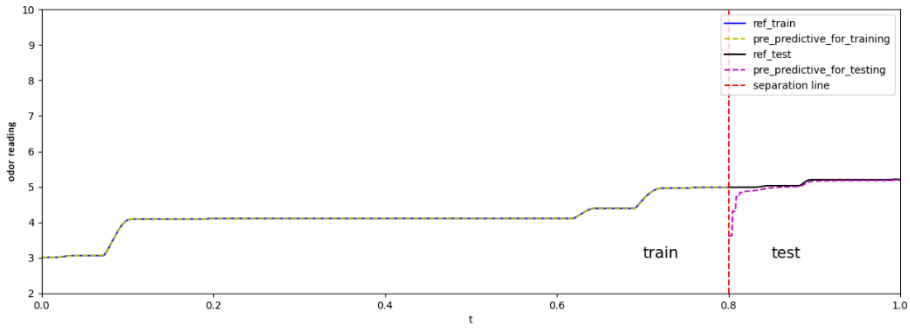

Figure 3.17 Total of 1,400 odor-sampling points in the forecasting model.

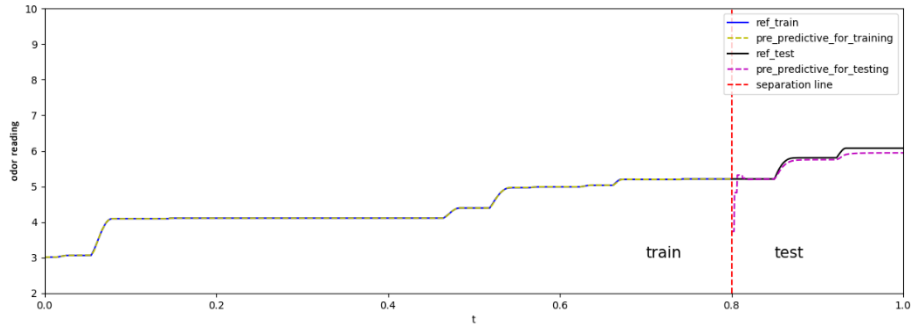

Figure 3.18 Total of 1,600 odor-sampling points in the forecasting model.

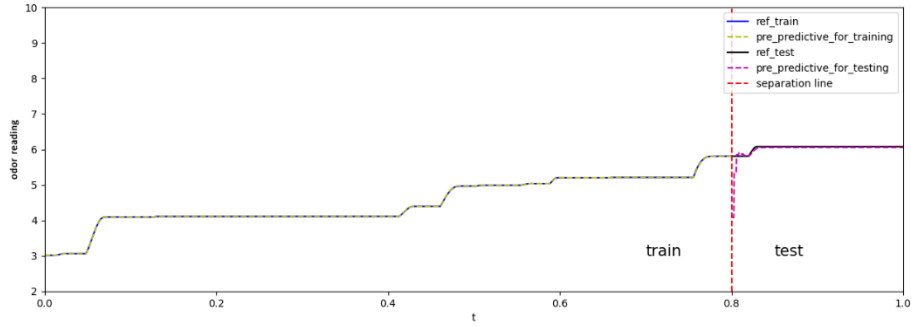

Figure 3.19 Total of 1,800 odor-sampling points in the forecasting model.

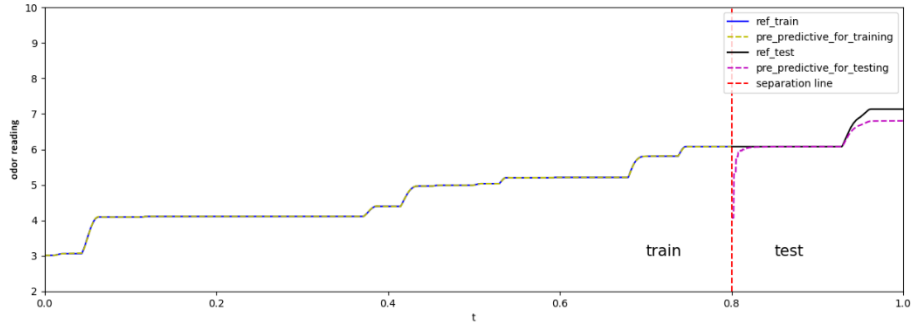

Figure 3.20 Total of 2,000 odor sampling points in the forecasting model.

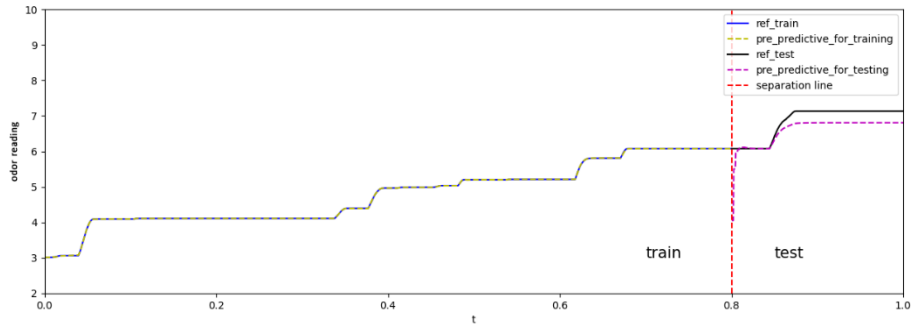

Figure 3.21 Total of 2,200 odor sampling points in the forecasting model.

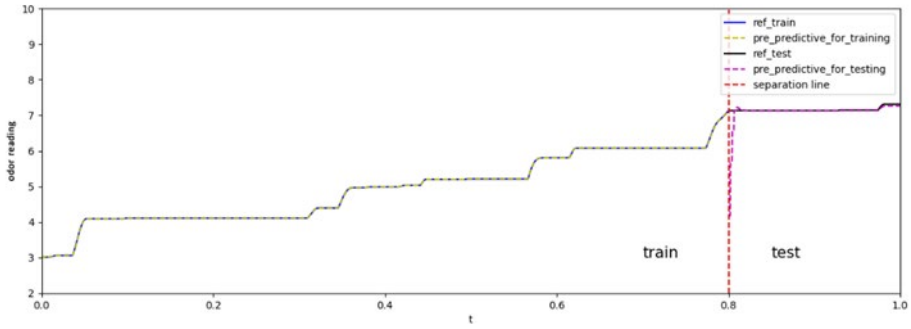

Figure 3.22 Total of 2,400 odor-sampling points in the forecasting model.

Table 3.2 Test score RMSE and epoch for forecasting.

The data points from the beginning to point N maintains loss at 1e-5		
N	Epoch	Test RMSE scores
200	4153	0.25
400	6368	0.2
600	6863	0.22
800	6155	0.17
1000	4885	0.15
1200	4250	0.23
1400	7698	0.21
1600	5619	0.21
1800	3775	0.21
2000	5310	0.3
2200	4812	0.35
2400	3528	0.34

The cost-efficiency of purchasing IoT devices is based on data from the past 8 years (2013–2020). Risk data were collected after processing 1,000 episodes. When the forecasted odor was greater than 6.5, implying that the mango inside the box was overripe, the reward was counted as 0. The ultimate reinforcement model reward was then generated.

The Deep Q-learning model program aims to output the distribution of selecting FIFO and LSFO policies, such that mango ripeness can maintain freshness starting from seven episodes. Figure 3.23 shows the average rewards for readings ranging from 3 to 6.5. This method can ideally reduce the loss from 6.91 to 0%. The reward, on the other hand, is sampled in the last step of each episode, resulting in five overripe signals within a 1,000-time window. The simulated output concluded that the simulated loss improvement was adjusted from 6.91% (loss percentage, $L\%$) to 0.035% (risk percentage, $R\%$), as expressed in equation 2:6.91% × 5/1000 = 0.035% [5].

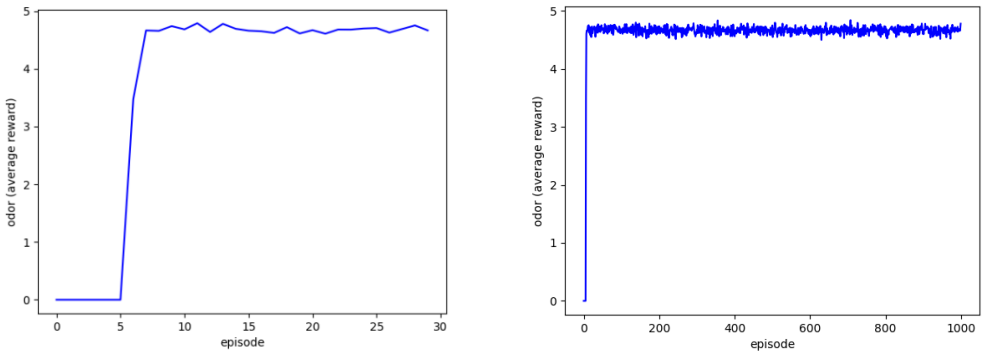

Figure 3.23 Agent's average reward at 30 (left) and 1,000 episodes (right).

The reference-based mango market has 93.09% market dominance, 100% FIFO strategy, and 6.91% of mangoes lost owing to an unknown shelf-life. However, a simulated policy coverage is proposed in which 85.21% of the market uses the FIFO strategy while 14.79% uses the LSFO strategy.

These strategies had a tolerance of approximately 8% in terms of scene construction

fidelity. This LSFO policy would also require 100% of the sensors to be used for the LSFO strategy, whereas no sensors would be required for the FIFO strategy. A total of 100% of the sensors indicate the proper number of sensors, which is equal to the number of harvesting times according to fractal theory.

3.5 Summary

An innovative feature of the proposed FL method is that it allows scene communication and generates standards. The predictive model with an FL network adds weight to public ledgers, which have more frequently voted for updates to the mathematical model. These highly connected ledgers, which belong to the data contributors, provide a solid consensus on the ability to predict fruit lifespan trends. This method assigns an internal reinforcement learning model to each ledger, and the percentage of adopted strategies can cover the existing market share of the problem to reflect the positive correlation of jointly simulating the decision-making environment under the same cornerstone. This interaction generates a highly influential model using blockchain technology.

The proposed prototype is based on blockchain technology to accommodate the features of decentralized governance and immutability and to align with the general trend of supply chain management in driving toward deploying distributed databases. The advantage of such an event-driven system architecture is that it involves more stakeholders at different stages of the process. With such decentralized governance, there is no unified authority in the system. Individual players cannot dominate or manipulate operations. In addition, scene applications provide a more flexible and scalable method for users to establish forecasting norms without middlemen. Using blockchain technology, critical parameters are stored in a public ledger. As the updated parameters are timestamped, they cannot be tempered by hackers. This implies that every user can read the data but cannot alter or reverse it. These records are permanently stored

on the computers of the individual nodes. One critical feature in the development of AI policy is that the scene feature library is closely related to the scene consensus applied to the backbone technology of the double helix system. Finally, it produces scenarios that facilitate risk minimization and cost reduction of IoT devices used in the supply chain to derive a global optimal policy with the support of AI insight.

There are several potential directions for future development in blockchain-based federated learning. Some possible areas of focus include the following:

• Improve the efficiency and scalability of federated learning systems so that the systems can handle larger and more complex datasets.

• Investigate blockchain technology to enable new forms of decentralized AI model sharing and collaboration, such as creating “AI model marketplaces” where organizations can buy and sell access to anonymize the AI model via smart contracts.

• Research ways to incentivize participation in federated learning systems, such as using digital currency-based rewards or other financial incentives.

Chapter 4 Conclusion

In this thesis, a novel system for fruit waste prediction based on blockchain-based Federated Learning (FL) with Smart Contract is proposed. The proposed system can contribute to the fruit logistics industry through more accurate estimation of shelf-life and better recommendations on the delivery sequence of product batches to their destinations. Mango delivery was chosen in this thesis to demonstrate the application of the method. Mangoes are highly sensitive to various environmental conditions. If the mangoes are delivered to the retailers within the right period of time, it will help increase the sales of the retailer and, thus, the returns of the growers. The proposed system obtains mangoes' ripeness information and makes logistics decisions using AI agents so that a maximum number of fruits are delivered to the destinations within their expected shelf-life.

Developing a prediction AI model for destination selection requires a large amount of training data. Rather than from a single source, it is better to collect the data from different suppliers but it then gives rise to security problems, update difficulties, and the risk of node failure. Therefore, federated learning (FL) is used in this study. Besides, to eliminate the problem of requiring a centralized permission server for FL, a serverless approach, which is dubbed the DGAP framework, is proposed in this study. The proposed DGAP framework for fruit waste reduction is based on blockchain-based Federated Learning (FL) with Smart Contracts. Smart contracts are located at a specific address on an EVM-compatible blockchain. They consist of two main components: the constructor and the function. The constructor is a unique function that executes a single time during the contract's deployment, serving to initialize its data. Within the contract, a public function accepts a string argument and modifies the "parameters" storage variable. These state variables are persistently retained within the contract's storage. By using the "public" keyword, the variable is made accessible externally,

enabling other contracts or clients to access and retrieve the prediction model parameters. The smart-contract-based FL algorithm used in this thesis involves local model training, asynchronous aggregation, and global model updates. Each farmer trains a local model on the local machines. The prediction model for farmers is a suitable neural network for forecasting fruit ripeness and shelf life. After local training, the model updates are sent to an aggregator. Aggregation combines these updates to form a global model. The aggregated model is redistributed to all nodes for further training. This iterative process continues until the model converges. Convergence is assessed depending on predefined metrics such as accuracy.

As mentioned above, the prediction model for mangoes' shelf-life estimation leverages AI agent-driven decision-making algorithms to improve overall accuracy. Mangoes can be effectively stored for 2-3 weeks at 12-13°C through refrigeration [55]; however, the absence of shelf-life categorization in the training model leads to a reduced storage period. Therefore, overall prediction-based reinforcement learning is suitable for solving chain effects from shorter storage duration by training with the simultaneous ripening of different fruit boxes when exporting fruit. The study gathered 8 years of expert value data, enabling the machine to calculate the total steps for each episode: (4 harvesting times + 1 delivery) × 12 months × 8 years = 480 steps. By utilizing expert value data from Tridge.com, stakeholders can perform detailed trend analyses, revealing shifts in global demand and supply chains. For instance, such data can highlight seasonal pricing fluctuations or regional production changes. The historical scope of the data enables predictive modeling, helping users anticipate future market behaviors based on past trends. Following 100 episodes, the memory becomes capable of offering experiences that guide action selection for the next 900 episodes of the simulation. Memory is structured into four key components: state, action, reward, and next state. Within its limits, the memory counter synchronizes with the step counter to choose actions by pinpointing the

highest action value within the observation space, encompassing shelf-life projections and storage conditions. When the agent identifies an action value associated with shelf-life projections, it decides to apply the LSFO policy. Alternatively, the agent may choose the FIFO strategy based on the received action index. As the AI agent progresses through each learning cycle, it continually hones its approach to discover the most effective strategy. As a result, the proposed system achieves a significant reduction in mango loss, decreasing from 6.91% to 0.035%, as indicated in the simulation outcome. It demonstrates the effectiveness of the proposed system.

While the proposed DGAP framework is effective, it is constrained to adjusting future mango shelf-life only within the temperature range of 23.24°C to 30.13°C and humidity levels between 54.61% and 84.65%. Further research is needed to extend the system to more flexible settings. Besides, research into ways to incentivize participation in FL systems is ongoing, with potential solutions including digital currency-based rewards and other financial incentives. This will encourage broader adoption and contribution to the federated learning framework. Future development of blockchain-based federated learning includes exploring decentralized AI model marketplaces and devising financial incentives to encourage participation. Integrating these advancements will contribute to the evolution of a sustainable collaborative AI ecosystem.

Future work will focus on enhancing the forecasting-driven simulation environment to model various aspects of supply chain operations, enabling AI agents to navigate complex networks. The hybrid agents' system will optimize the allocation and logistics of real-world assets for maximum efficiency and cost reduction. The supply chain protocol will be integrated to model interactions between stakeholders such as farmers, investors, and institutional actors within the context of tokenized agricultural commodities. AI-to-AI interactions will allow

agents to collaborate on optimizing supply chain operations and executing transactions, including crop production, storage, transportation, and commodity trading. Agents will aim to maximize yield, minimize waste, and mitigate risks through interactions with simulated wallets and tokenized assets. Smart contract functionalities will automate and enforce agreements and transactions, with AI agents verifying compliance and facilitating secure transactions. AI agents will also explore investment opportunities in the tokenized agricultural commodities market, using insights from the supply chain protocol to assess risk-return profiles, diversify portfolios, and hedge against market volatility. Performance metrics such as return on investment, portfolio diversification, and risk management will be used to evaluate the effectiveness of these AI-driven financial strategies.

Another possible future work is to develop a Proactive Settlement Layer (PSL). This layer will enhance the capability of deep learning models to identify the relationship between prediction and value within a settlement layer. This new layer addresses a gap in the Arweave AO ecosystem, which focuses on on-chain storage and compute scaling but lacks features for autonomous deep learning. An "Imaginary Layer" will abstract value from data by tokenizing factors like food ripeness and systematic costs, with outputs including sustainable value estimation. The "Internet of Value" block will innovate by having sensors function as nodes in a value network through smart contracts, with sensors' values collectively correlated with their sensitivity and forming liquidity pools to transform prediction utility into value signals. AI agents will interact with this system, moving towards greater automation and efficiency in supply chain services, representing a shift from human-centric to machine-driven processes.

References

[1] Joas, J., Desvignes, C., Vulcain, E., Lechaudel, M., "Understanding links between preharvest conditions and postharvest management in production chains is one of the keys to ensuring fruit quality in commercial networks," *Acta Hortic.,* vol. 880, p. 207–215, 2010.

[2] Canali, M., Amani, P., Aramyan, L., Gheoldus, M., Moates, G., Östergren, K., et al., "Food waste drivers in Europe, from identification to possible interventions," *Sustainability,* vol. 9, p. 37, 2017.

[3] Gustavsson, J., Cederberg, C., Sonesson, U., et al., *Global Food Losses and Food Waste: Extent, Causes and Prevention,* Rome, Italy: Food and Agriculture Organization of the United Nations, 2011.

[4] Ridolfi, C., Hoffmann, V., Baral, S., "Postharvest Losses: Global Scale, Solution, and Relevance to Ghana," *Ruiz: International Food Policy Research Institute,* 2018.

[5] Ali, S. Y., Hossain, et al., "Postharvest losses of mangoes at different stages from harvesting to consumption," *Int. J. Bus. Soc. Sci. Res.,* vol. 7, p. 21–26, 2019.

[6] Galvan, S., "Postharvest losses of mango (Mangifera indica, L.) in Iloilo and Guimaras, Philippines," *Int. J. Innov. Creativ.,* vol. 13, p. 1228–1239, 2020.

[7] Sab, M., Ashok, M. B., Sudhakara, S. N., "Estimation of post-harvest losses of mangoes at different stages from harvesting to consumption," *Int. J. Curr. Microbiol. Appl. Sci.,* vol. 6, p. 310–318, 2017.

[8] Yeshiwas, Y., Tadele, E., "An investigation into major causes for postharvest losses of horticultural crops and their handling practice in debre markos, north-western Ethiopia," *Adv. Agric.,* p. 1–10, 2021.

[9] Tarekegn, K., Kelem, F., "Assessment of mango post-harvest losses along value chain in the gamo zone, Southern Ethiopia," *Int. J. Fruit Sci.,* vol. 22, p. 170–182, 2022.

[10] Government, Queensland, "Mango Handling Guide. Department of Primary Industries and Fisheries," 2017.

[11] Albert, R., Barabási, A. L., "Statistical mechanics of complex networks," *Rev. Mod. Phys.,* vol. 74, p. 47–97, 2002.

[12] Ruiz-Garcia, L., Lunadei, L., "Monitoring cold chain logistics by means of RFID," *Sust. Radio Freq. Identific. Solut.,* vol. 2, p. 37–50, 2010.

[13] Baietto, M., Wilson, A. D., "Electronic-nose applications for fruit identification, ripeness, and quality grading," *Sensors,* vol. 15, p. 899–931, 2015.

[14] Tort, Ö. Ö., Vayvay, Ö., Çobanoglu, E. A., "Systematic review of sustainable fresh fruit and vegetable supply chains," *Sustainability,* vol. 14, p. 1573, 2022.

[15] Nordey, T., Davrieux, F., Léchaudel, M., "Predictions of fruit shelf life and quality after ripening: Are quality traits measured at harvest reliable indicators?," *Postharvest. Biol. Technol.,* vol. 153, p. 52–60, 2019.

[16] Dutta, J., Deshpande, P., Rai, B., "AI-based soft-sensor for shelf life prediction of'Kesar'mango," *SN Appl. Sci.,* vol. 3, p. 1–9, 2021.

[17] Kuo, T. T., Ohno-Machado, L., "Modelchain: Decentralized privacy-preserving healthcare predictive modeling framework on private blockchain networks," 2018.

[18] Mammen, P. M., "Federated learning: Opportunities and challenges," 2021.

[19] Chen, Y., Richter, J. I., Patel, P. C., "Decentralized governance of digital platforms," *J. Manag.,* vol. 47, p. 1305–1337, 2021.

[20] Mahendra, M. S., Rai, I. N., Janes, J., "Current postharvest handling practices of salak and mango fruits in Indonesia," *IV International Symposium on Tropical and Subtropical Fruits,* vol. 975, p. 479–486, 2008.

[21] McMahan, Brendan, et al., "Communication-efficient learning of deep networks from decentralized data," *Artificial intelligence and statistics, PMLR,* pp. 1273-1282, 2017.

[22] Yang, Q., Liu, Y., Chen, T., Tong, Y., "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST),* vol. 10, no. 2, pp. 1-19, 2019.

[23] Q. Yang, et al., "Horizontal Federated Learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning,* p. 49–67, 2020.

[24] Regulation, G. D. P., "General data protection regulation (GDPR)," Accessed in October, 24(1), 2018.

[25] Q. Yang, et al., "Vertical Federated Learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning,* p. 69–81, 2020.

[26] Chen, Y., Qin, X., Wang, et al., "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intelligent Systems,* vol. 35, no. 4, pp. 83-93, 2020.

[27] Song, Y., Yu, F. R., Zhou, L., Yang, X., He, Z., "Applications of the Internet of Things (IoT) in smart logistics: A comprehensive survey," *IEEE Internet Things J.,* vol. 8, p. 4250–4274, 2020.

[28] Eugster, W., Laundre, J., Eugster, J., Kling, G. W., "Long-term reliability of the Figaro

TGS 2600 solid-state methane sensor under low-Arctic conditions at Toolik Lake, Alaska.," *Atmos. Meas. Tech. ,* vol. 13, p. 2681–2695, 2020.

[29] Srinivasan, R., Lilien, G. L., Rangaswamy, A., "First in, first out? The effects of network externalities on pioneer survival," *J. Mark.,* vol. 68, p. 41–58, 2004.

[30] endes, A., Cruz, J., Saraiva, T., Lima, T. M., Gaspar, P. D., "Logistics strategy (FIFO, FEFO or LSFO) decision support system for perishable food products," *International Conference on Decision Aid Sciences and Application (DASA) (IEEE Publications),* vol. 173, no. 8, 2020.

[31] Yadav, A. S., Sharma, V., Agarwal, P., Swami, A., Yadav, P. K., "Pharmaceutical drug two-warehouse inventory model under FIFO dispatching policy using ant colony optimization for travelling salesman problem," *Linguistics and Culture Review,* vol. 5, no. S2, pp. 1148-1171, 2021.

[32] Yousef Ghiami, Patrick Beullens, "The continuous resupply policy for deteriorating items with stock-dependent observable demand in a two-warehouse and two-echelon supply chain," *Applied Mathematical Modelling,* vol. 82, pp. 271-292, 2020.

[33] Navid Mohamadi, Seyed Taghi Akhavan Niaki, Mahdi Taher, Ali Shavandi, "An application of deep reinforcement learning and vendor-managed inventory in perishable supply chain management," *Engineering Applications of Artificial Intelligence,* vol. 127, 2024.

[34] Pfohl, H. C., Gallus, P., Thomas, D., "Interpretive structural modeling of supply chain risks.," *Int. J. Phys. Distrib. Logist. Manag. ,* vol. 41, p. 839–859, 2011.

[35] Astuti, R., Marimin, M., Machfud, M., Arkeman, Y, "Risks and risks mitigations in the supply chain of mangosteen: a case study," *Operat. Supply Chain Manag.,* vol. 6, p. 11–25, 2014.

[36] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., et al., "Openai Gym," 2016.

[37] Hochreiter, S., Schmidhuber, J. , "Long short-term memory.," *Neural Comput.,* vol. 9, 1997.

[38] Yu, Y., Si, X., Hu, C., Zhang, J. A., "review of recurrent neural networks: LSTM cells and network architectures.," *Neural Comput. ,* vol. 31, pp. 1235-1270, 2019.

[39] Kaelbling, L. P., Littman, M. L., Moore, A. W. , "Reinforcement learning: A survey.," *J. Artif. Intell. Res.,* vol. 4, pp. 237-285, 1996.

[40] Bruckner, S., Albrecht, A., Petersen, B., Kreyenschmidt, J. A., "predictive shelf life model as a tool for the improvement of quality management in pork and poultry

chains.," *Food Control,* vol. 29, pp. 451-460, 2013.

[41] Kevrekidis, Y., and Samaey, G., "Equation-free modeling.," *Scholarpedia,* vol. 5, p. 4847, 2010.

[42] Choi, S. M., Park, J., Nguyen, Q., and Cronje, A., "Fantom: A scalable framework for asynchronous distributed systems.," 2018.

[43] Kaur, G., Gandhi, C., "Scalability in blockchain: Challenges and solutions," *Handbook of Research on Blockchain Technology ,* pp. 373-406, 2020.

[44] Bager, S. L., Düdder, B., Henglein, F., Hébert, J. M., Wu, H. , "Event-based supply chain network modeling: Blockchain for good coffee.," *Front. Blockchain,* vol. 5, p. 846783, 2022.

[45] Nakamoto, S. , "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review ,* p. 21260, 2008.

[46] Tikhomirov, S., "Ethereum: State of knowledge and research perspectives," *International Symposium on Foundations and Practice of Security,* pp. 206-21, 2017.

[47] Hildenbrandt, E., Saxena, M., Rodrigues, N., Zhu, X., Daian, P., Guth, D., et al., "Kevm: A complete formal semantics of the Ethereum virtual machine," *31st Comput Sec Found Symposium (CSF) (IEEE Publications) ,* pp. 204-17, 2018.

[48] Kocijan, J., Murray-Smith, R., Rasmussen, C. E., Girard, A., "Gaussian process model based predictive control," *Proceedings of the 2004 American Control Conference (IEEE Publications),* 2004.

[49] Azman, K., Kocijan, J., "An application of Gaussian process models for control design," *International Control Conference ICC,* vol. 4, 2006.

[50] Rafi, U., Leibe, B., Gall, J., Kostrikov, I., "An efficient convolutional network for human pose estimation," *BMVC,* 2016.

[51] Fan, J., Wang, Z., Xie, Y., Yang, Z., "A theoretical analysis of Deep Q-learning," *Learning for Dynamics and Control (PMLR),* p. 486–489, 2020.

[52] Frederico, G. F., "Project Management for Supply Chains 4.0: A conceptual framework proposal based on PMBOK methodology.," *Oper. Mana. Res.,* vol. 14, p. 434–450, 2021.

[53] Barbieri, P., Boffelli, A., Elia, S., Fratocchi, L., Kalchschmidt, M., Samson, D., et al., "Oper. Manag. Res.," *What can we learn about reshoring after Covid-19? ,* vol. 13, pp. 131-136, 2020.

[54] Kumar, R., Sindhwani, R., and Singh, P. L. , "IIoT implementation challenges: Analysis and mitigation by blockchain," *J. Glob. Oper. Strateg. Sour. ,* vol. 15, pp.

363-379, 2022.

[55] Singh, Z., Zaharah, S. S., "Controlled atmosphere storage of mango fruit: Challenges and thrusts and its implications in international mango trade," *Acta Hortic Global Conference on Augmenting Production and Utilization Of Mango: Biotic and Abiotic,* pp. 179-191, 2015.

[56] Liukai Wang, Fu Jia, Lujie Chen, Qifa Xu, "Forecasting SMEs' credit risk in supply chain finance with a sampling strategy based on machine learning techniques," *Annals of Operations Research,* vol. 331, pp. 1-33, 2023.