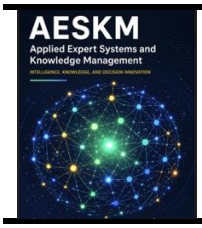
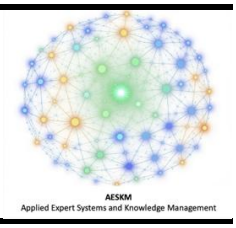


Applied Expert Systems and Knowledge Management

www.aeskm.org
eISSN: 3153–3825



Integrating Reinforcement Learning, Response Surface Methodology, and Neural Networks into Agent-Based Modeling for Dynamic Investment Decisions

Muhammad Khurram Ali[1,*], Haider Ali[1], Hafiz Mohammad[1]

[1] Department of Industrial Engineering, University of Engineering and Technology, Taxila, Pakistan

ARTICLE INFO

Article history:
Received 22 July 2026
Received in revised form 1 September 2026
Accepted 9 September 2026
Available online 11 September 2026

Keywords:
Multiagent Reinforcement Learning (MARL); Complexity Economics; Complex Adaptive Systems (CAS); Design of Experiments (DoE); Decision Support System (DSS); Computational Intelligence

ABSTRACT

The recent growth in data- and simulation-driven decision-making has prompted researchers to develop intelligent, more adaptive agent-based models. However, smarter policy modeling in complex economic systems remains challenging, with significant room for improvement. This paper develops an Artificial Neural Network (ANN) based Decision Support System (DSS) using Multiagent Reinforcement Learning (MARL) for dynamic investment decision analytics within a complex economic system. It embeds Reinforcement Learning (RL) into an Agent-Based Model (ABM) of business investments, enabling dynamic and adaptive decision-making. RL algorithms, including Q-Learning, Deep Q-Networks, and Proximal Policy Optimization (PPO), are integrated into the working behavior of investor agents and investment alternatives to reinforce decision intelligence. A comparative evaluation of the results from the four models demonstrates that the PPO-ABM yields the best performance, with higher investor wealth and lower annual failure risk. A formal factorial experiment with three levels of input factors is then designed in the PPO-ABM, generating a large amount of experimental data. Response Surface Methodology (RSM) is used to develop predictive mathematical models and response surfaces for the eight output variables of the economic system. An ANN model is trained on the generated big data, and a recommender system is accordingly designed to facilitate intelligent decision-making and policy modeling. The developed framework can be utilized by researchers, policymakers, investors, and strategic decision-makers to conveniently test different economic scenarios and generate system-level results of the adopted policies.

1. Introduction

In modern economics, studying the dynamics of investment agents and their hidden connections with one another and with the environment is an exciting research area for both industry and academic researchers. The ability to predict complex market indices using adaptive simulation models leads to informed, scientifically grounded policies and decisions. Traditional models generally rely on representative agents and equilibrium assumptions, but these abstractions cannot address

* *Corresponding author.*
E-mail address: khurram.ali@uettaxila.edu.pk
https://doi.org/10.59543/deh08h23

real-world phenomena such as collective behavior and volatility clustering. Conventional financial market models typically leave significant room for the incorporation of adaptable paradigms that account for market complexity, investor heterogeneity, and learning [1].

Agent-based modeling (ABM) ensures dynamism, agent interactions, real-world stochasticity, adaptability, and system-level emergence by incorporating all stakeholders and the overall environment [2]. In financial markets, ABM simulates the interactions among boundedly rational agents to reveal emergent phenomena such as bubbles, crashes, and herding [3]. Such agent-based models enable interactions among stakeholders, including buyers and sellers, to study how individual decision rules combine to shape overall market patterns [4]. Several agent-based models have been created to replicate the decentralized decision-making procedures of diverse participants in financial markets. These models offer essential insights into the internal structure of markets and individuals. A few instances of such ABM models from the reported literature are presented here.

Within a behavioral ABM framework, Takahashi and Terano [5] modeled two types of traders: fundamentalists and trend followers. They demonstrate how selection mechanisms and heterogeneity lead to price deviations and excess returns. Following up, Souissi and Bensaid [6] offered a multi-agent stock market model that simulates trading volume patterns across different behavior profiles, using zero-intelligence, fundamentalist, and history-based traders.

Mastroeni and Naldi [7] employ a game-theoretic, agent-based decision model between investors and financial professionals to replicate investment decisions [8]. The model by Assenza and Gatti [9] incorporates trading causes and price-limit rules into ABMs to uncover emergent investment dynamics influenced by institutional rules rather than by agent learning. A behavioral agent-based model of two interacting national markets was developed by M. Ezzat [10]. The model analyzes the impact of transaction taxes on investment decisions and simulates bubbles, crashes, and clustered instability. Another model-based study by Stefan and Atman [11] uses imitation and anti-imitation strategies in a small-world trading network to explain psychological characteristics that affect asymmetric return rates and wealth distribution in investment outcomes. The agents in the study proposed by Sueyoshi and Tadiparthi [12] evaluate different market policies and make decisions accordingly.

It is obvious from these studies that although ABMs are gradually getting popular among financial market researchers, they generally follow basic behavioral rules rather than complex intelligence. There is a need to make the finance ABMs intelligent enough to replicate real-world financial market characteristics, including volatility clustering, extreme losses or returns, and the functioning of buy/sell orders. Some further examples of these traditional ABMs support this argument. The model by Mizuta [13] and Trimborn [14] represents the financial dynamics of artificial markets, but simple heuristic rules influence the agents. A variety of agent-based financial market models are also reviewed in general surveys, such as Todd and Beling [15], which highlight how complex yet simplified models of real-world price formation and investment are produced by simple rules of behavior at the agent level. Farmer and Patelli [16] have explicitly named their model the zero-intelligence framework (ZI). Also, Kim, Lee, et al. propose a model in which heterogeneous agents compare their wealth to a reference group and adjust their behavior based on whether they are outperforming or falling behind.

One approach the researchers have adopted to make the finance ABMs intelligent is to develop a data-driven ABM by providing large amounts of real-world data. One such example is the study by Souissi and Bensaid [6], which demonstrates that value-based traders, trend-based traders, and random decision traders may all compensate for their shortcomings by using real-world data that

reveal clear patterns and aid in forecasting market behavior. Monti e al. [17] make a similar case for data-driven ABM, where agents are first given the microdata.

Building on the fixed rule-based heuristics, recent studies have focused on enhancing the reactive and adaptable behaviors of agent-based models. Vitali and Battiston [18] presented a network-based ABM in which markets adjust by learning from the environment to avoid full collapse. A study by Kroujiline and Gusev [19] examines how different investor timeframes influence price changes and shows how short- and long-term behaviors shape market trends and the ability to predict returns. Tubbenhauer and Fieberg [20] created a framework for multi-agent value-at-risk forecasting in which agents' behavior was learned from experience in risk-prediction contexts by iteratively updating their strategies based on past simulation errors. The agent-based financial market model by Franke and Westerhoff [21] states that traders' changing behavior can alter market uncertainty, but in a patterned way. It identifies an important behavioral mechanism of herding that replicates real market patterns through simulated moment matching.

Another form of intelligent agent behavior is demonstrated by equation-free control and data-based calibration methods, which improve the consistency of model development through experimental setup without directly changing the agent's rules. Patsatzis and Russo [22], for instance, presented a variable-free machine learning framework that uses a diffusion map (a mathematical technique for identifying hidden patterns) to identify a few essential patterns that can explain the complexity of agent behavior. They direct the financial market simulation to aim for a constant balanced state. The impact of managers' overconfidence on initial public offerings (IPO) decisions and financial stability is examined using an agent-based model calibrated for Poland (Rzeszutek, Godin et al). Some agent-based models have been developed to explain how market dynamics are affected when individuals' behavioral characteristics are incorporated [23]. Wei and Chen [24] built an artificial market to generate synthetic trading data. Similarly, in order to investigate how investors with varying risk profiles select P2P loans when accounting for platform bankruptcy risk, Liu & Dong et al. (2022) created a multi-agent simulation; they demonstrated that the popularity of borrower credit grades, average returns, and investment volumes are all significantly impacted by platform risk and fee structure. These developments imply that ABM agents are being made intelligent through learning from social interactions, market reactions, and empirical calibration. However, most of these models lack sophisticated machine learning and reinforcement learning techniques for designing intelligent agent behavior. Overall, the reported ABM modeling practice shows that most current agents cannot independently understand and utilize advanced learning strategies. Instead, they mainly rely on fixed rules or on choices within a limited set of strategies. Although advanced Bayesian estimates and other data-driven techniques have improved statistical realism, current agents are mostly 'reactive' rather than 'trained/learnt'.

One solution to achieve agent intelligence is the model-fitting approach. According to recent discussions on the calibration and validity of ABMs, they have generally been shown to be "intelligent" compared to traditional agent adaptation modeling [25]. A comparative study of calibration methods revealed that researchers mainly rely on modeling sudden market movements, such as crashes, rather than simulating how agents make decisions based on their knowledge over time [26].

An agent-based model proposed by Lamperti and Roventini [27] uses the random forest machine learning algorithm to reduce computational time. Agent behavior in this model is still designed using traditional modeling approaches, and machine learning is not embedded into it. Dehkordi and Lechner [28] conducted a detailed literature review of environments in which machine learning

algorithms are used to design agent decision-making rules. This study proposes practical guidelines and extensions for embedding ML algorithms in a variety of agent behaviors.

Studies by [29] and [8] integrated Fuzzy Analytic Hierarchy Process with agent-based modeling to explore global investment decisions. A model by Ali [30] used Fuzzy Analytic Hierarchy, integrated with an ABM, to identify key factors and rank countries for plant location; this approach proved more practical and flexible than traditional methods.

Reinforcement Learning (RL) has emerged as a powerful tool for making investment decisions in environments with uncertainty [31]. It enables agents to learn optimal policies from experience rather than from predefined rules [32] [3]. Many studies have been reported in the literature that deploy RL in dynamic decision-making; however, integrated ABM-RL models for investment decisions are very rare. Related work by Olmez, Heppenstall, Ge, Elsenbroich, Birks et al. (2024) developed an agent-based model of the UK housing market and incorporated a central bank agent that learns via reinforcement learning to make real-time adjustments to mortgage rates. They demonstrate through shock simulations that the RL agent outperforms fixed-rule policy approaches in stabilizing prices and reducing housing-market volatility. One such study of RL-ABM in financial markets is presented by Maeda and DeGraw [33], which demonstrates how reinforcement learning can better adjust the agent's behavior in response to emergent market structures, considering real-world patterns and behaviors. The model does not utilize RL algorithms, such as PPO, and it does not include risk factors. The multi-agent deep RL system by Shavandi and Khedmati [34] demonstrated that hierarchical deep Q-Networks (DQN) can outperform single-agent models across different timeframes. This model is limited to a DQN application only.

Another instance of the integration of an RL trading agent into an event-driven ABM is presented by Dicks, Paskaramoorthy [35], which highlights the behavior of RL agents in various market situations. In this model, only a single RL agent learns and adapts while the rest remain static and fixed rule-based. Moreover, the model is limited to Q-Learning only. Similarly, He and Lin [36] investigated RL equilibrium dynamics in order-book simulations, providing a methodological basis, but it doesn’t present a complete investment decision framework capable of inducing smart decisions under risk.

Another case of Q-Learning-based ABM is the work by Zhou and Lin [37], which uses RL in limit-order market models to achieve equilibrium. [38] constructed a multi-agent RL market simulator in which agents learn trading behavior by trial/error and rewards received from the market. The model is limited to a specific background and does not develop a recommender system for future convenience. These are some instances of the reported RL-ABM studies. Overall, they are limited to a single RL algorithm, and no formal scientific experiments are conducted. Moreover, they lack the development of machine learning-based recommender systems.

It is evident from the above literature review that although the use of agent-based models (ABMs) to study the dynamics and investment behavior of financial markets is growing, very few studies integrate deep reinforcement learning with ABMs to achieve intelligent decision analytics for such markets. Most reported ABMs rely on predefined heuristic or rule-based judgments, which limit their flexibility in the intricate complexities of ever-changing real-world markets. RL algorithms, including DQN and PPO, therefore, need to be embedded in a variety of investment-related ABMs.

Moreover, most studies develop models with limited applicability, primarily among researchers and academics. There are hardly any studies that formulate predictive mathematical models, generate response surfaces, and build recommender systems to establish input/output relationships from RL-ABM simulations, thereby providing convenience and flexibility for a broader range of users.

Considering these clearly explored research gaps, this paper develops RL-integrated agent-based models for investment decisions. It is accompanied by thorough experimentation on the generated data using RSM. A Recommender System, backed by an ANN model, is also trained on a large dataset generated by a second scientifically designed experiment within the RL-ABM model.

The following are the main objectives of this study:

1. To develop intelligent agent behavior using various RL algorithms and then select the one that yields the best results for further experimentation.
2. To conduct scientifically designed experiments in the selected RL-ABM to gain deeper insight into the data created.
3. To develop predictive mathematical models, response surfaces, and an ANN-trained recommender system that can be utilized by a wide range of readers of this study to facilitate convenient experimentation.

In the next section of the paper, the methods applied are presented. It includes mathematical models, working logic, pseudocode, and schematic diagrams of the RL-ABM models. All these models are implemented in the Results and Discussion section, which also includes the development of input/output relationships, response surfaces, and the ANN-based recommender system. The paper concludes in Section 4.

2. Methodology

This section presents the overall methodology for developing RL-ABM models. It is structured into three phases as shown in Figure 1. A phase-wise description of the adopted methods is elaborated here. The third phase consists of two parts

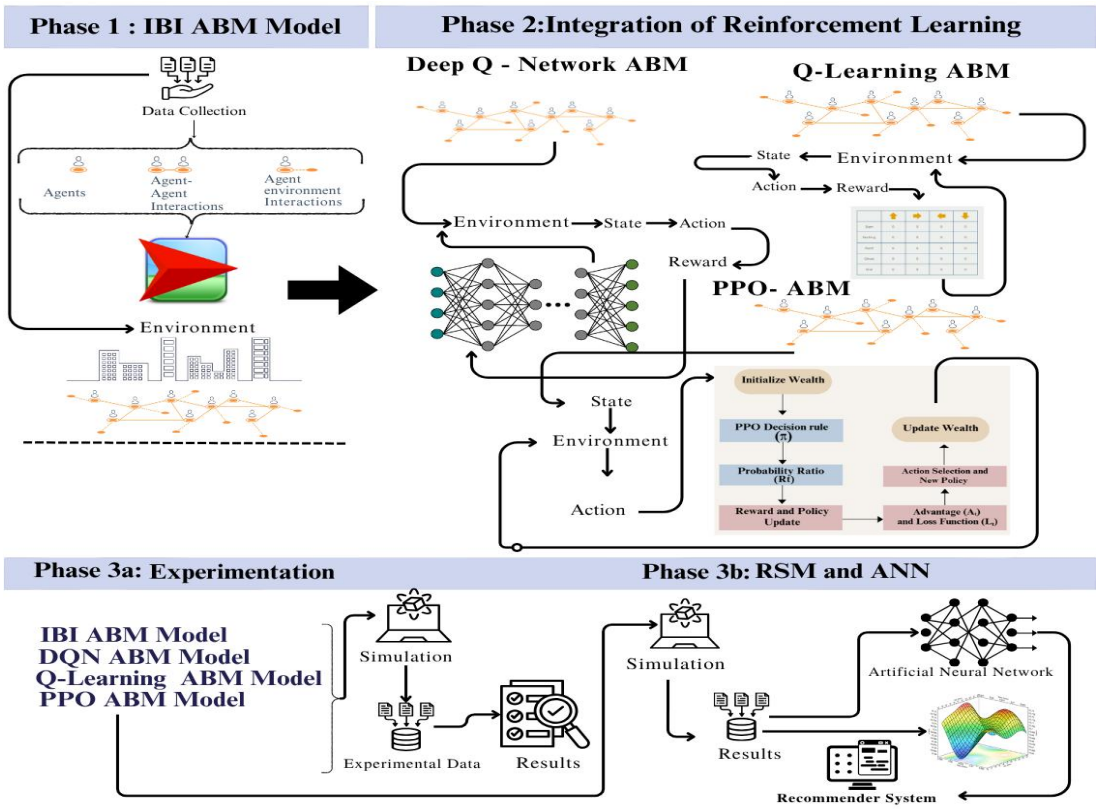


Fig.1. Overall Methodology of the Study

2.1. Phase 1: Initial Business Investment ABM Model (IBI-ABM)

In the first phase, the initial business investment model (IBI-ABM) proposed by Railsback and Grimm [39] is developed in line with the details provided in the book. The IBI-ABM represents how investors allocate resources among competing businesses. In its basic form, investor agents evaluate each business agent by considering profit and risk, and then invest where the utility, typically expressed as profit relative to risk, is highest. While this captures the mechanics of decision-making, it remains reactive, as investors rely solely on current values and do not anticipate future outcomes. A Bayesian extension addresses this limitation by allowing investors to hold prior beliefs about profitability and update them as new evidence emerges. This adjustment makes the model more adaptive and forward-looking, offering a closer reflection of real investment behavior under uncertainty.

The main computations performed by the investor agents and the business opportunities agents are summarized here. At the start of the simulation, the global parameters are defined as per the model proposed by *Railsback and Grimm [39]*.

The initial mean risk 'R' and initial variance of risk 'V' associated with the businesses/investments are calculated using the maximum and minimum investment agent risks, r_{max} and r_{min} , respectively.

$R = (r_{max} + r_{min})/2$ (1)

$V= (r_{max} – r_{min})^2/12$

(2) $\alpha_o = (R^2 - R^3 - (R.V) / V$

(3) $\beta_o = (R/V) (1 - R)^2 + (R - 1)$

(4)

Investor agents are initially placed on randomly selected unoccupied business agents, with initial wealth Wi(0) = 0. αo and βo are the prior parameters representing the initial probabilities of imaginary failure and success, respectively. The Initialization, updating utility(Bayesian), decision rule, and accounting are calculated by the following equations ;

Each investment/business agent is defined with a profit P_j and failure risk Fj , which is given by

$P_j \sim$ Exponential (mean = μ)

(5)

$Fj \sim$ Uniform (r_{min} , r_{max}) (6)

Each business agent (the investment opportunity) also records the number of failures $f_j(0) = 0$ and the number of years occupied $y_j(0) = 0$

where P_j= profit of investment agent, μ_π = global mean business opportunity profit, F_j= annual risk investment agent, U(0,x) = uniform random draws on [0,x]

At the start of each year t , investors consider their current opportunity and neighboring unoccupied opportunities as potential destinations

$\alpha_j = \alpha + f_j$ (7)

$\beta_j = \beta + (y_j – f_j)$ (8)

$F_j(t) = \alpha_j / (\beta_j + \alpha_j)$ (9)

Utility is then predicted as expected end-of-horizon wealth:

$U_{ij}(t) = (W_i(t) + T \cdot P_j) \cdot (1–F_{oj}(t))^T$. (10)

Here α_j and β_j are the updated failure and success count parameters after the real failures and successes are observed

Each investor selects the business with the highest utility and relocates itself there:

$a^* = \arg_j \max U_{ij}$ (11)

After the movement of investors, their wealth is updated based on the investment agent's profit and stochastic risk of failure.

$$W_i(t^+) = W_i(t) + P_j \quad (12)$$

Then the investor's wealth at time t+1 is;

$$w_i(t+1) = w_i(t) + P_j \quad (13)$$

If investment fails:

$$U(0,1) < F_j \Rightarrow w_i(t+1) = 0 \quad (14)$$

If failure occurs;

$$F \leftarrow F + 1 \quad (15)$$

$$f_j \leftarrow f_j + 1 \quad (16)$$

$$y_j \leftarrow y_j + 1 \quad (17)$$

where F = total investor failures in the model, f_j = failures of the investment agent, y_j = years the investment agent has been occupied.

Investment agent counters (f_j ,y_j) feed into the Bayesian updating process using Eqs. (7), (8), (9). As a result, the estimated risk $F_{oj}(t)$ changes over time, influencing future utility calculations in Eq. (10). This is how the agents adapt their perception of risk based on experience. The pseudo-code of the IBI-ABM is presented in Table 1, and it is described in Figure 2.

Table 1
Pseudo Code for Initial Business Model

NO	**Explanation**
1:	Initialize business opportunities with profit P and risk F
2:	Initialize investors with wealth = 0
3:	compute prior (α_0, $\beta 0$)
4:	GO (each year):
5:	for each investor:
6:	for each candidate patch:
7:	estimated risk = ($\alpha 0$ + failures) / ($\alpha 0$ + $\beta 0$ + years)
8:	U = (wealth + $P_j H$) × (1 – estimated risk)^H
9:	move to business opportunity with max U
10:	for each investor:
11:	wealth + = P_j
12:	if random < r → wealth = 0 (failure)
13:	update counters

Fig. 2. Schematic Diagram of the Initial Business Investment (IBI) ABM Model

2.2. Phase 2: Reinforcement Learning in Agent-based Modeling

In this phase, three reinforcement learning algorithms, including Q-Learning, Deep Q-Networks, and Proximal Policy Optimization (PPO), are integrated into the IBI-ABM presented in Section 2.1.

2.2.1.Q-Learning ABM (QL-ABM)

In IBI-ABM, the model selects investments that maximize expected utility via utility maximization with Bayesian risk updating. In QL-ABM, the investor agents choose the business using ε-greedy Q-values and update the Q-table using the learning rule.

IBI to QL Conversion Logic:

The IBI-ABM model is converted into a Q-Learning model by replacing the utility calculation function with a learned action-value estimate based on the epsilon-greedy policy's exploration/exploitation trade-off, and then updating the Q-values in the table. The main logic of converting the IBI-ABM model to Q-Learning is to replace rule-based decision-making with adaptive learning.

New global parameters are introduced that are standard RL parameters, which enable agents to learn from experience rather than a fixed Bayesian updated rule [40]. The profit and patch risk variables are kept the same. Investment agent counters start at $f_j(0) = 0$, $y_j(0) = 0$ and $W_i(0) = 0$.

New reinforcement learning parameters are introduced, including the learning rate α, the discount factor γ, and the exploration probability **ϵ**. Each investor maintains the following:

- a state $s_i(t)$, defined as current wealth and business opportunity conditions.
- an action $a_i(t)$, representing the business opportunity chosen.
- A Q-table Q(s,a) is initialized with all entries set to zero.

Decision (Action Selection): In each simulation year, the investor selects a business opportunity from the set of available opportunities [40].

($N_i(t)$) to move to, this process is followed by **ϵ-**Greedy policy;

$$a_i(t) = \begin{cases} \text{random Business opportunity from } N_i(t), & \text{with probability } \epsilon, \\ \text{argmax}_{a \in Ni(t)}\, Q(s_i(t),a), & \text{with probability } 1-\epsilon, \end{cases} \quad (18)$$

$s_i(t)$ is the current state of investor i (wealth, profit, risk).

Instead of maximizing utility directly, investors now explore or exploit new business opportunities based on what they have already learned.

Reward: After choosing the business opportunity, the investor agent computes a reward [40, 41];

$$r_i(t) = P_j - c \cdot F_j, \quad (19)$$

where c is a scaling factor (set to 1000 in the implementation of this study).

Higher profits yield higher rewards, while higher risks reduce rewards. The state of the business opportunity is represented as:

$$s_t^i = (w_i(t), P_j, F_j) \quad (20)$$

$$\text{Q-Value Lookup} = Q(s,a) \quad (21)$$

$$\text{Max Next Q (Look Ahead)} = \max_{a'} Q(s',a') \quad (22)$$

$$\text{Target Calculation} = y = r + \gamma \max_{a'} Q(s',a') \quad (23)$$

Q- Learning update: In Q-learning, the utility is replaced by Q-table values; i.e., decisions are made by maximizing expected discounted reward [40, 42].

$$Q(s,a) \leftarrow Q(s,a)+\alpha[(r+\gamma_{a'}\max Q(s',a')-Q(s,a)] \quad (24)$$

where Q(s,a) = Q-value of state–action pair, s (state) = investor's current state, a(action) = chosen business opportunity, α = learning rate, γ = discount factor, r (reward) = immediate reward, Max $_{a'}$ Q(s',a') = best possible Q-value in next state

Wealth Accounting: Once the investors have moved to a business opportunity based on a greedy policy, the wealth will be updated as;

$$W_i(t^+) = W_i(t) + P_j \quad (25)$$

$$W_i(t+1) = \begin{cases} 0 & \text{if } U(0,1) < F_j, \\ W_i(t+), & \text{otherwise.} \end{cases} \quad (26)$$

$$f_j(t+1) = f_j(t) + 1_{\{\text{failure occurred}\}}, \quad y_j(t+1) = y_j(t) + 1 \quad (27)$$

State Transition: The investors update their state for the next decision:

$$s_i(t+1) = (W_i(t+1), P_j, F_j) \quad (28)$$

This new state is used in equation (18) to make decisions about finding new business opportunities. The pseudo-code of the QL-ABM is presented in Table 2, and it is described in Figure 3 as well.

Table 2
Pseudo Code for Q learning Model

NO	Explanation
1:	initialize Business opportunities with profit P and risk F
2:	initialize investors with wealth = 0
3:	initialize Q-table, α, γ, ε
4:	GO (each year):
5:	for each investor:
6:	s = discretize (wealth, P, F)
7:	with prob ε: choose a random action (business opportunity)
8:	else: choose action = argmax_a Q(s,a)
9:	move to chosen Business opportunity
10:	apply profit/loss, observe reward r
11:	s' = new state
12:	Q(s,a) = Q(s,a) + α × (r + γ × max_a' Q(s',a') - Q(s,a))
13:	update wealth and business opportunity counter

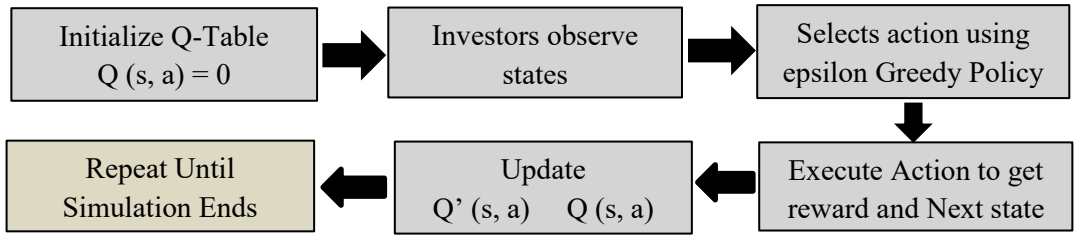


Fig. 3. Schematic Diagram of the Q-Learning ABM Model

2.2.3 Deep Q Network

While Q-Learning uses a Q table to update rewards, the DQN uses a neural network Q(s,a;θ) to approximate the Q values, making it useful in large state spaces or in high-dimensional inputs. It stores transitions (s,a,r,s') in memory and samples them randomly to break the correlation between them. A second network updates less frequently to provide stable target values [43]. DQN uses a function approximator to estimate Q-values from weights. It first uses the same epsilon-greedy policy as Q-Learning, but instead of storing them in a table, the Q-values are approximated by a function Q(s,a;θ), which is updated through temporal difference.

While transforming the IBI-ABM into DQN-ABM, the static wealth and profit equations remain the same, but the decision-making equations change. The utility function is now replaced by the DQN target equation, the Loss function, and the gradient update. This conversion of the model introduces great adaptability. Instead of passively following fixed equations, agents now actively learn from experience, adjust their strategy over time, and handle fluctuating environments through function approximation and weight updates [40, 43]. The profit and patch risk variables are kept the same

Investor setup: Each investor starts with zero wealth: $W_i(0) = 0$. Instead of initializing a Q-table or utility function, we introduce weights.

Weights vector for Q-function: w (0) = (0,0,0).

Decision: Each investor decides where to move. Using the probability ϵ, the agents explore and choose a random business opportunity. With probability 1−ϵ, they exploit and choose the business opportunity with the highest estimated Q-value) [40].

$$a_i(t) = \begin{cases} \text{random Business opportunity from } N_i(t), & \text{with probability } \epsilon, \\ \operatorname{argmax}_{a \in Ni(t)} Q(s_i(t),a, w), & \text{with probability } 1-\epsilon, \end{cases} \quad (29)$$

Action selection follows the standard ϵ-greedy policy, where the best action is chosen based on the estimated Q-value $Q(st,a;\theta_t)$ [43]. In this implementation, θ_t is simplified to a linear weight vector $w_t = (w_1,w_2,w_3)$ corresponding to wealth, profit, and risk features. So the Q values are no longer looked up in a table; instead, they are approximated with weights.

Reward: The reward is introduced for investors to get feedback on the decision they made [39];

$$r_i(t) = P_j - c \cdot F_j, \quad c = \text{RiskPenalty} \quad (30)$$

The Q value is updated as:

$$Q(s,a;w) = w_1 \times W_i(t) + w_2 \times P_j + w_3 \times F_j \quad (31)$$

This function replaces the Q table and utility function implemented in QL-ABM and the IBI-ABM, respectively [41].

The weights are updated using the temporal-difference (TD) error

$$\text{Target Value} = y_i(t) = r_i(t) + \gamma_{a'} \max Q(s_i (t+1), a'; w) \quad (32)$$

The TD error is computed using the following relation;

$$\delta_i(t) = y_i(t) - Q(s_i(t), a_i(t); w) \quad (33)$$

Weights are updated using the relation,

$$w \leftarrow w + \alpha\delta_i(t)\, \phi(s_i(t), a_i(t)), \quad (34)$$

where $\phi(s,a) = (W_i(t),P_j,F_j)$ is the feature vector

Wealth Accounting: Once an investor has moved to a business opportunity based on Deep Q network, the wealth will be updated as;

$$W_i(t^+) = W_i(t) + P_j \quad (35)$$

$$W_i(t+1) = \begin{cases} 0 & \text{if } U(0,1) < F_j, \\ W_i(t+), & \text{otherwise.} \end{cases} \quad (36)$$

$$f_j(t+1) = f_j(t) + 1_{\{\text{failure occurred}\}}, \quad y_j(t+1) = y_j(t) + 1 \quad (37)$$

The investor updates their state for the next decision using the following relation;

$$s_i(t+1) = (W_i(t+1), P_j, F_j) \quad (38)$$

In the DQN extension, the Q-function is approximated using a linear function of investor wealth, patch profit, and patch risk. The Q-value is updated using the temporal-difference (TD) error, consistent with the principle introduced in the Deep Q-Network framework developed by Mnih and Kavukcuoglu [43].

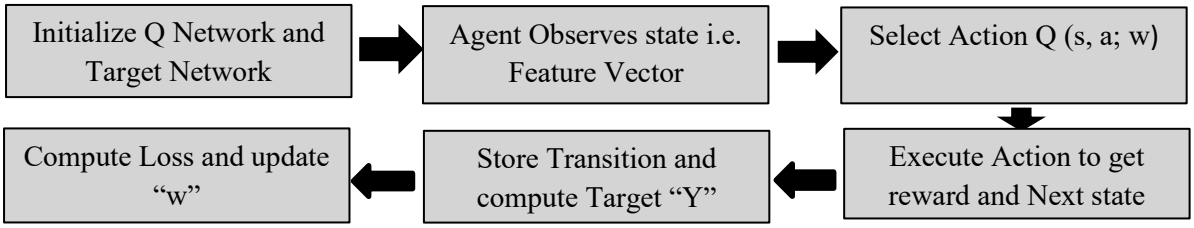


Fig.4. Schematic Diagram of the DQN ABM Model

2.2.3. Proximal Policy Optimization in Agent-based Modeling (PPO-ABM)

Proximal Policy Optimization (PPO) belongs to the actor-critic family of policy gradient algorithms. In contrast to value-based algorithms like Q-Learning and DQN, PPO optimizes the policy directly ($\pi\theta(a|s)$). It uses clipping policy updates to prevent the new policy from deviating too far from the previous one, thereby increasing training stability. The clipping policy restricts how much the new policy can differ from the old one by bounding the probability ratio $r_t(\theta)$. One of the most important components of PPO is the probability ratio, which quantifies the difference between the old and new policies for the same action-state pair.

The wealth factor is converted by reducing the profit contribution to a percentage. A fraction penalty is introduced if the risk factor occurs (gradual wealth reduction). Instead of the utility calculations adopted in the IBI-ABM, the PPO-ABM uses a probability ratio, which is then used by the 'clipped loss' calculation (to prevent large policy updates). It is followed by the gradient using the value of the 'clipped loss' based on the probability ratio. Unlike IBI-ABM, where the utility is used for investment selection, in PPO the utility serves as a numerical reward signal. The initial profits and risks assigned to the business opportunities will remain the same as in the IBI model, but each investor will start with positive initial wealth ($W_i(0) = n$) .

In this case, the investor agents do not store Q values or weights; instead, they interact with the learned policy. Each investor records their state (wealth, profit of current business opportunity, risk of current business opportunity) using;

$$s_i(t) = (W_i(t), P_j, F_j) \quad (39)$$

The investor applies a PPO-based decision mechanism. Equations 40-46 represent the PPO mechanism [44, 45]**;**

$\pi_{new} = 1 / 1+e^{-(Pt - Ft.100)}$ (40)

Probability ratio = $R_t = \pi_{new} / \pi_{old} + \epsilon$, $\epsilon = z$ (41)

Reward = $r_t = W_t + (P_j \cdot T \cdot 0.8) \cdot (1 - F_j / 3)^T$ (42)

The policy update is based on reward function as previous and new reward as inspired by Schulman, Wolski [44], [45].

Advantage $= A_t = r_t - r_{t-1}$ (43)

Clipped surrogate objective: $L_t = \min (r_t A_t,\ \text{clip}\ (r_t, 0.8, 1.2)\ A_t)$ (44)

Action selection:

if $L_t > 0$ { choose argmax(P − (F×100)) , If empty business exists
argmax(P − (F×100)) / n +1 , Otherwise

if $L_t \leq 0$ { Choose a random business such that $n \leq 1$, if such patches exist
Choose a random patch from all patches; otherwise (45)

n : number of investors on patch.

Policy Update occurs after each step by:

$\pi_{old} \leftarrow \pi_{new}$ (46)

The updated wealth after the policy upgrade is calculated as;

$W_i (t^+) = W_i(t) + 0.8 \cdot P_j$ (47)

$W_i(t+1) =$ $0.7 \times W_i(t+)$, if $U(0,1) < F_j$,.
$W_i(t+)$, otherwise (48)

$f_j (t+1) = f_j(t) + 1_{\{\text{failure occurred}\}}$, $y_j (t+1) = y_j(t) + 1$ (49)

The investor's next state is updated:

$s_i (t+1) = (W_i (t+1), P_j, F_j)$ (50)

Pseudo code for PPO Inspired model is presented in Table 3.

Table 3
Pseudo code for PPO Inspired model

NO	**Explanation**
1:	Initialize investor wealth Wi(0) = n; $\pi_{old} = 0.5$, $\pi_{new} = 0.5$
2:	Initialize patches with profit Pj, risk Fj, occupancy yj, failures fj
3:	For t = 1 to years-simulated:
4:	For each investor i:
5:	Observe state $si_{(t)} = (Wi(t), P_j, F_j)$
6:	Compute policy: $\pi_{new} = 1 / (1 + \exp(-(Pj - 100 \times j)))$
7:	Compute probability ratio: $r_t = \pi new / (\pi old + \epsilon)$, ϵ = small constant
8:	Compute reward: $R_t = Wi_{(t)} + (P_j \times T \times 0.8) \times (1 - Fj/3)^{\wedge}T$
9:	Compute advantage: At = rt - rt-1
10:	Compute clipped objective: $Lt = \min(r_t \times At, \text{clip}(rt, 0.8, 1.2) \times At)$
11:	Action selection:
12:	Wealth update: $Wi(t+) = Wi_{(t)} + 0.8 \times P_j$
13:	Risk penalty: Wi(t+1) = 0.7× Wi(t+) if U(0,1) < Fj; else Wi(t+)
14:	Update patch stats: $fj(t+1) = fj(t) + 1_{\{failure\}}$; yj(t+1) = yj(t)+1
15:	Update policy: πold ← πnew; next state $si_{(t+1)} = (Wi_{(t+1)}, P_j, F_j)$

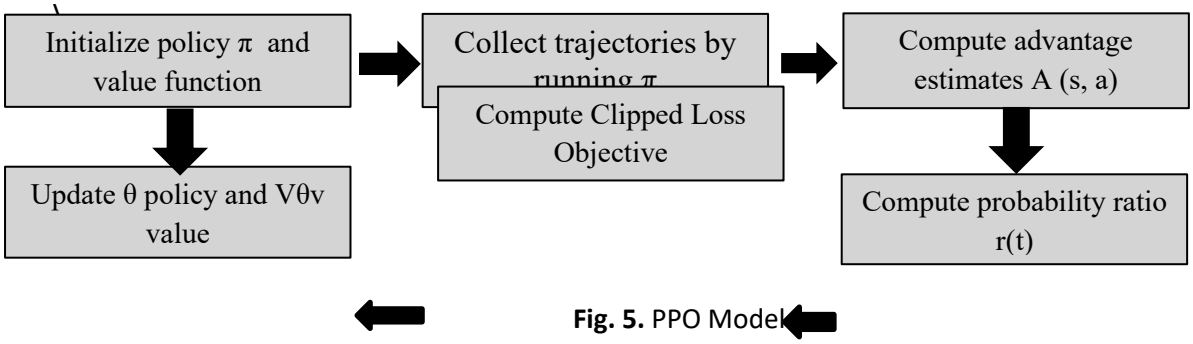


Fig. 5. PPO Model

3. Results and Discussions

This section presents detailed results from simulations of the developed Multi-Agent Deep Reinforcement Learning (MADRL) models for dynamic investment decisions. Key performance metrics, including investment returns, risk assessment, and agent behavior, are evaluated in all four models: the initial business investor ABM (IBI-ABM), Q-Learning-based ABM (QL-ABM), Deep Q-Network ABM (DQN-ABM), and Proximal Policy Optimization ABM (PPO-ABM). Results from the four models are then evaluated comparatively. Formal factorial experiments are conducted in the PPO-ABM, which generated a massive amount of experimental data. Response surface methodology is applied during this experimentation phase. In the final subsection, an ANN model is trained on the large experimental dataset, thereby developing a Recommender System.

3.1. Evaluation of the Built Models

In this subsection, the four models are constructed and evaluated, and the quantitative results from the experiment are briefly elaborated.

3.1.1. Initial Business Investor ABM Model (IBI-ABM)

The IBI-ABM proposed by *Railsback and Grimm [39]* in the twelfth chapter of their book is built and presented here. The model is developed in the open-source platform of NetLogo®. In this model, the investor agents sense information about investment opportunities, represented as patches. Each investor chooses the business alternative with the highest profit value. Investors maintain beliefs (in the form of probability distributions) about each business's expected returns. These beliefs are updated using Bayesian updating after each investment period, based on actual observed returns. The agents make dynamic decisions about relocation based on the financial health of the investment opportunities. Readers are encouraged to read the details of this model provided in chapters 10 and 12 of the book [39]. The overall patterns that emerged from the model analysis are briefly discussed here.

The IBI-ABM model interface is shown in Figure 6a. The overall performance of this complex economic system is evaluated by five agent attributes, including the mean wealth of investors, standard deviation (SD) in the wealth of investors, utility of the businesses for the investors, mean profit from the business alternatives, and the mean risk associated with the business

alternatives/opportunities represented by the patches. Wealth, its standard deviation, and utility are investor-specific variables, whereas profit and risk are associated with the business alternatives (patches).

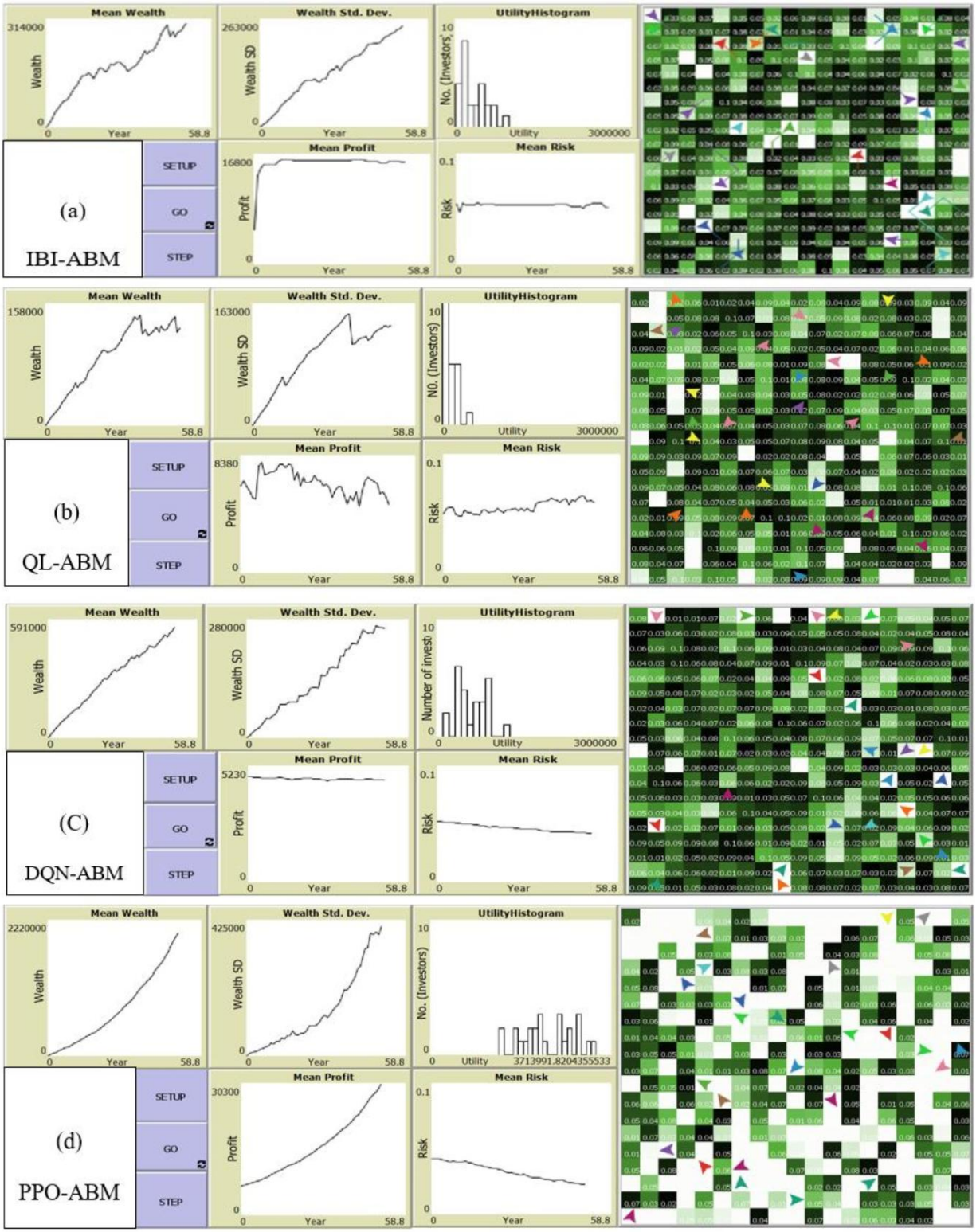


Fig. 6. Main output windows of the four ABM models

Simulation results for these five variables are presented in the interfaces (Figure 6). The mean wealth of the investors, along with its standard deviation (SD), is increasing as the simulation runs. Investor agents compute their updated wealth annually. Despite a few spikes in the graph, the overall trend for both variables is nearly linear. The utility histogram is left-skewed, indicating that investors have received very low utility values. It can be inferred from this histogram that this economic system is providing low utility to investors.

The mean profit and mean risk associated with the investment alternatives represented by the patches remain stable throughout the simulation. The patches window is called the 'world' as per NetLogo® terminology. Black patches represent business alternatives that go bankrupt due to their low profits. White patches have the highest profit values. Green patches have different profit values associated with their shades of green. In the case of IBI-ABM, many businesses have gone bankrupt, and there are very few white patches.

3.1.2 Q-Learning ABM Model (QL-ABM)

The IBI-ABM is updated to incorporate the Q-Learning theme, as detailed in Section 2. The results from the Q-Learning model presented in Figure 6(b) show that it exhibits volatile patterns across all attributes. Although the mean wealth and its SD are increasing, there are some sudden drops. During the simulation, it exhibited increasingly random behavior across iterations. The utility histograms are squeezed to the left with very low values. The mean profit of businesses is showing instability, following a downward trend. The mean risk shows a slight upward trend, with many small fluctuations. The world is a mix of black, white, and green patches. Just like the IBI-ABM, white patches in the QL-ABM are fewer. Discussion on the relocations of investor agents is presented in section 3.2.

3.1.3 Deep Q-Network ABM Model (DQN-ABM)

In this section, the IBI-ABM is embedded with the reinforcement learning (RL) technique of Deep Q-Networks. The ABM model is rebuilt, and the results of an iteration are shown in Fig. 6 (c). The inclusion of this DQN has a significant impact on the overall emerging pattern of the investment system. The indices are stable, with a gradual increase in the mean wealth. The mean profit of the patches is almost linear and stable. The risk is decreasing slightly as the simulation years pass. The utility histogram spread is broad, indicating that investors have different utilities associated with the business opportunities. The mean utility has increased significantly. Once again, there are a large number of black patches with the lowest profit values, and very few white patches (the highest possible profits).

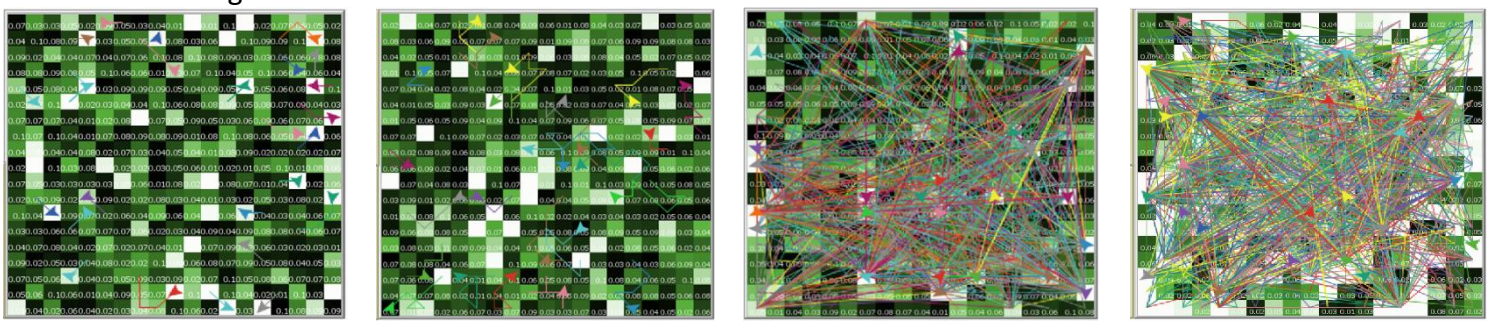
3.1.4 Proximal Policy Optimization ABM Model (PPO-ABM)

In the fourth case, the IBI-ABM is embedded with proximal policy optimization. The model is publicly available for download from the CoMSES Computational Model Library [46] . The results of the PPO-ABM are quite interesting and exciting when compared with all previous models. As agents are exposed to the economic system, they continue to learn using PPO and thus make wiser decisions. The mean wealth of investors is therefore increasing with a smooth nonlinear trend. The same is true for investors' mean profit. The risk associated with business opportunities decreases and is the lowest among all four models.

The utility histogram is very close to a bell-shaped normal distribution curve. This means that the utility values are distributed among the investors in a very natural way. The utility values are numerous, too. Another exciting finding from the PPO-ABM is the emergence of many patches with the highest profit values. This means that these business opportunities have flourished due to the healthy investment environment surrounding them. The number of bankrupt (black) patches is very small in this case.

3.2. Comparative Evaluation of the Four Models

In this section, the results from all four models are evaluated comparatively. As a first step of this evaluation, the overall movement of the investors (turtles) is monitored using the 'pen-down' feature of NetLogo. The investor agents switch from one business opportunity to another whenever they perceive higher profits in the surrounding business environment. This transforms the economic system into one that is dynamic, adaptive, and complex. Whenever the investor agents relocate themselves to invest in better opportunities, they leave a line in the patch window of Net Logo. After performing the separate simulations of the four models, the patch windows with 'pen-down' are shown in Figure 7.



(a) Initial Business Investor ABM Model (IBI-ABM) (b) Q-Learning ABM Model (QL-ABM) (c) Deep Q-Network ABM Model (DQN-ABM) (d) Proximal Policy Optimization ABM Model (PPO-ABM)

Fig. 7. A Comparative Analysis of Investor Relocation Dynamics in Four ABMs

The movement patterns that emerged in the simulations indicate the least agent movement in the IBI-ABM. The amount of this movement increases in the QL-ABM, DQN-ABM, and PPO-ABM, respectively. However, these relocation movements are very small in IBI-ABM and QL-ABM, but very large in DQN-ABM and PPO-ABM. The most complex and intensive movement pattern is observed in the PPO-ABM system.

A sample of the quantitative results, rounded to the nearest hundred, obtained from simulating these four models is presented in Table 4. The mean wealth of investors is lowest in QL-ABM, and it is almost double that of IBI-ABM in the DQN-ABM. However, this wealth is massively increased in the PPO-ABM models. The standard deviation is also higher in PPO-ABM, indicating greater stability and uniformity in the wealth distribution among community members in this economic system. The PPO-ABM has also outperformed in accumulating investors' aggregate utility.

Table 4
Sample Quantitative Results from the Four ABM Model Simulations

Agent Attributes	IBI-ABM	QL-ABM	DQN-ABM	PPO-ABM
Mean Wealth of Investors	300000	155000	618700	2000015
Wealth Stand. Dev.	267000	162000	401000	359527
Utility for Investors	226500	80700	566600	2480482
Mean Business Profits	16000	5300	5400	27944
Patches Mean Failure Risk	0.054	0.056	0.043	0.024292
No. of Investors with wealth < 100000	8	19	3	0
Total Wealth of Investors	7241900	1628600	1.2 E7	5.051E7
No. of patches with profit <500 (bankrupt)	40	35	38	5
No. of patches with profit >15000	15	22	27	181

Mean business profits are found to be highly volatile in QL-ABM. In the case presented in Table 4, it has the lowest value. This attribute has also achieved the highest score in the case of PPO-ABM. However, DQN-ABM is very close to QL-ABM for this attribute. The mean annual failure risk associated with the business opportunities is also lowest in the case of PPO-ABM, and it is highest for the QL-ABM.

To make the comparison more comprehensive and interesting, this study investigates four additional attributes. The first attribute is the number of investors who are left with the least wealth over the simulation period. These investors can be considered bankrupt or failures. The threshold for this category is set below 100,000 units of wealth. From Table 4, the highest number of such failures is observed in QL-ABM, whereas no investor falls into this category in PPO-ABM. The DQN-ABM model has shown only three such cases.The second additional attribute observed in this study is the cumulative wealth received by all investors in each economic system. In both the DQN-ABM and PPO-ABM models, the investors have received a huge amount of wealth as compared to IBI-ABM and QL-ABM. This value is highest in the PPO-ABM, and it can be considered a symbol of economic strength associated with the PPO-ABM system.

Another interesting aspect of the comparative evaluation of these four economic systems is the number of business alternatives that could not survive in the aggregate dynamic environment. The threshold for this attribute is set to 500 units of patch profit. Such businesses can be considered bankrupt companies. The Number of such failed businesses is 40 in the IBI-ABM model, the highest among the four scenarios. Only five patches from the PPO-ABM model have faced this bad scenario.

The final attribute under this additional investigation is the number of high-performing business alternatives. All patches that have generated more than 15,000 units are placed in this cluster. From Table 4, it is clear that PPO-ABM has outperformed all other models by a wide margin.

Overall, it can be concluded from this detailed comparative evaluation that the PPO-ABM-based economic system has outperformed the other three systems due to the cumulative intelligent decision-making of the agents, which is reinforced by the PPO procedure of Reinforcement Learning (RL). In some indices, it is very close to DQN-ABM, but its overall impact is remarkable.

Due to the strong economic patterns that emerged in the PPO-ABM, it is selected for further in-depth investigation through formal experiments.

3.3 In-Depth Designed Experimentation in the PPO-ABM Model

A formal full factorial experiment is designed and conducted in the PPO-ABM model, using the Behavior Space feature of NetLogo. The selected input parameters and their levels for this experiment are shown in Table 5

Table 5
Input Parameters with their Experimental Levels

Factors	**Initial Levels**		
	Low	Medium	High
No. of Investors	15	30	45
Mean Profit of Business Alternatives	2000	3000	4000
Decision Time Horizon (DTH)	3	5	7

The output variables remain the same as presented in Table 1, with the exclusion of the wealth standard deviation. The inclusion of eight output variables makes this problem a multi-objective one. The effects of input variables on the output variables are analyzed using the RSM with the face-centered central composite design (FC-CCD). This design requires three levels of each input variable.

A total of nineteen experimental runs were conducted in the PPO-ABM model, as shown in Table 5. The experiment is repeated 100 times per run to account for stochastic variation, and the mean values of the response variables are computed accordingly for each input level. The results obtained for all eight output/response variables are compiled in Table 6.

Table 6
Experimental Design Matrix of the PPO-ABM

Input Variables				Response Variables						
Run	No. of Investors	Profit	DTH	Investor Mean Wealth	Investor Total Wealth	Investor Utility	Mean Business Profit	Business Failure Risk	Failed Businesses	Strong Businesses
1	15	2000	7	729688	1.09453E+07	930549	9593.06	0.0298595	20.4	69.05
2	30	3000	5	922674	2.76802E+07	1.09976E+06	14334.4	0.0212629	13.44	111.97
3	45	4000	7	1.10981E+06	4.99413E+07	1.40251E+06	19280.9	0.0165905	10.37	146.54
4	15	4000	7	1.48005E+06	2.22008E+07	1.88156E+06	19303.3	0.0297481	10.11	145.03
5	30	3000	5	934341	2.80302E+07	1.11222E+06	14366.2	0.021231	14.06	111.58
6	30	3000	7	937837	2.81351E+07	1.18506E+06	14469.4	0.0215043	13.5	112.17
7	30	4000	5	1.24698E+06	3.74095E+07	1.48585E+06	19260.2	0.0212926	10.33	145.33
8	30	3000	5	916853	2.75056E+07	1.09304E+06	14373.1	0.0213326	13.29	112.28
9	30	2000	5	633013	1.89904E+07	756079	9744.59	0.0214077	20.75	70.26
10	45	2000	3	556282	2.50327E+07	620523	9590.04	0.0164792	20.17	69.45
11	30	3000	5	933877	2.80163E+07	1.1155E+06	14410.2	0.0212643	13.42	111.07
12	30	3000	5	933877	2.80163E+07	1.1155E+06	14410.2	0.0212643	13.42	111.07
13	15	2000	3	745938	1.11891E+07	833979	9610.33	0.029633	20.73	69.49
14	45	3000	5	840275	3.78124E+07	1.00111E+06	14542	0.0164246	13.66	112.28
15	15	3000	5	1.10762E+06	1.66143E+07	1.32131E+06	14511.3	0.0300508	13.81	112.3
16	15	4000	3	1.49117E+06	2.23676E+07	1.66367E+06	19394	0.0297803	10.33	146.32
17	30	3000	3	949840	2.84952E+07	1.05956E+06	14613.7	0.0211507	14	113.86
18	45	4000	3	1.11638E+06	5.02371E+07	1.24519E+06	19307.5	0.0164936	10.55	145.16
19	45	2000	7	555392	2.49926E+07	701199	9644.78	0.0164274	20.28	70.39

Regression analysis is conducted on all eight response variables using the respective commercially available statistical software. The appropriateness of the developed models has been tested using analysis of variance (ANOVA). Detailed ANOVA tables are compiled in the Appendices A1 through A8. The final empirical models for predicting eight output variables are presented in the equations.

Investor Mean Wealth = 162,962 +(-9,453.23 ×No. of Investors)+(410.259 ×Profit)+(-17,490.6 ×DTH) + (-3.17572 ×No.of Investors × Profit) + (82.9379 ×No. of Investors × DTH) + (0.0343438×Profit×DTH)+(156.503×No.ofInvestors^2)+(0.00126539×Profit^2)+(1276.38×DTH^2) (51)

Investor Total Wealth = -2.32232e+06 + (263,793 ×No. of Investors) + (2,164.34 ×Profit) + (-345,141 ×DTH) +(230.992×No.ofInvestors×Profit)+(311.236×No.ofInvestors×DTH)+(-11.1682×Profit×DTH)+(4,338.88×No.ofInvestors^2)+(0.0103911×Profit^2)+(31,398.8×DTH^2) (52)

Investor Utility = 139,314 +(-9,523.92 ×No.of Investors) +(420.093 ×Profit) + (-3,430.65×DTH) + (-3.78941 ×No. of Investors × Profit) + (-318.594 ×No. of Investors × DTH) + (12.3731×Profit×DTH)+(190.248×No.ofInvestors^2)+(0.00256165×Profit^2)+(976.754×DTH$^{2.}$ (53)

Mean Business Profit = 65.5946 + (-4.67617 ×No. of Investors) + (4.9521 ×Profit) + (-83.7056 ×DTH) + (-0.0011691 ×No.of Investors× Profit) + (0.567114 ×No. of Investors × DTH) + (-0.00966938 ×Profit×DTH) + (0.0839428 × No. of Investors^2) + (-5.39401e-06 ×Profit^2+(8.44951×DTH^2) (54)

Business Failure Risk =0.0415433 + (-0.000944825 ×No. of Investors) + (8.36948e-08 ×Profit) + (0.000156822 ×DTH) + (1.17928e-09 ×No. of Investors × Profit) + (-6.21486e-07 ×No. of Investors × DTH) + (-6.87737e-09 ×Profit ×DTH) + (8.33364e-06 × No. of Investors^2)+(-1.24799e-11×Profit^2)+(8.78948e-06×DTH^2) (55)

Failed Businesses = 46.2052 + (-0.0371409 ×No. of Investors)+(-0.0160334 ×Profit)+(-0.0804459 ×DTH)+ (9.66667e-06 ×No. of Investors × Profit) + (0.002 ×No. of Investors × DTH) + (-1.125e-05 ×Profit×DTH) +(-6.9874e-05 ×No.of Investors^2) + (1.78928e-06×Profit^2)+(-0.000180412×DTH^2) (56)

Strong Businesses = -39.0571 + (-0.0143357 ×No. of Investors) + (0.0661066 ×Profit) + (-1.96479 ×DTH) + (-7.91667e-06 ×No. of Investors × Profit) + (0.016875 ×No. of Investors × DTH) + (-2.5625e-05×Profit×DTH)+(-0.000590378×No.ofInvestors^2)+(-4.62784e-06×Profit^2)+(0.148041×DTH^2) (57)

The response surface plots for all eight output variables are presented in Fig. 8 through Fig. 15. The statistical significance of the input/output relationships can also be inferred from the ANOVA tables compiled in Appendices A1 through A7.

Figures 8 (a), 8 (b), and 8 (c) depict the effects of three input variables taken two at a time on the Investor Mean Wealth. From the ANOVA table and RSM figure, it can be inferred that the interaction effect of factors A (No. of Investors) and B (Initial Profit of Patches), denoted AB, is statistically significant for the response variable (Investor Mean Wealth), with a p-value of 0.0001. Moreover, the interactive effect of AC (No. of Investors and Decision Time Horizon (DTH)) is highly insignificant. As far as stand-alone input variables A, B, and C are concerned, A and B have proved to have a significant effect on the Investor Mean Profit.

Figures 9 (a, b, c) are the response surfaces of Total Investor Wealth with respect to the interactive effect of three input variables taken two at a time. The combined impact of only A and B is statistically proven to be significant ($p<0.0001$), while AC and BC are not significant. The effect of C (DTH) is once again insignificant, just like in the case of Investor Mean Wealth.

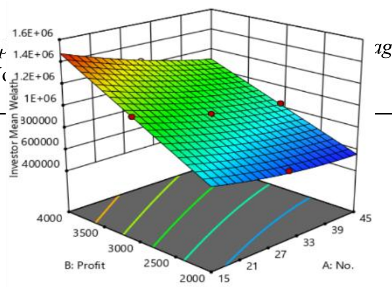


Fig 8 (a): Effects of Profit and No. of Investors on Investor Mean Wealth

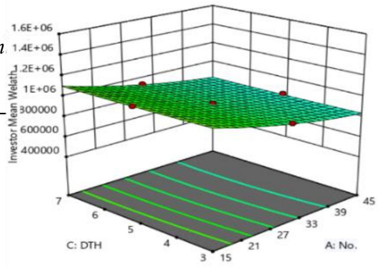


Fig 8 (b): Effects of DTH and No. of Investors on Investor Mean Wealth

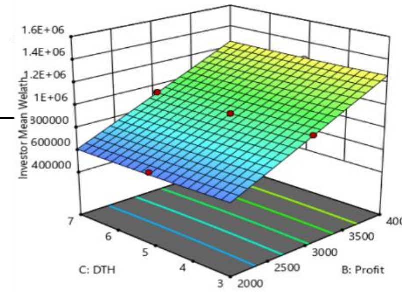


Fig 8 (c): Effects of DTH and Profit on Investor Mean Wealth

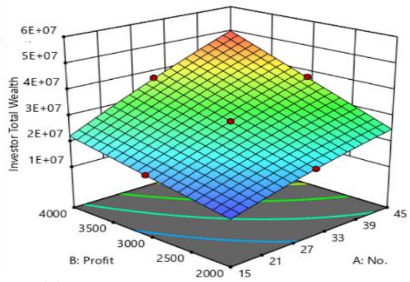


Fig 9 (a): Effects of Profit and No. of Investors on Investor Total Wealth

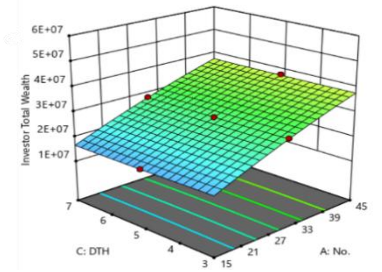


Fig 9 (b): Effects of DTH and No. of Investors on Total Investor Wealth

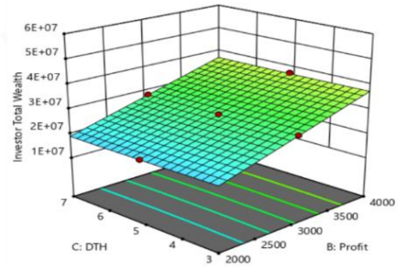


Fig 9 (c): Effects of DTH and Profit on Investor Total Wealth

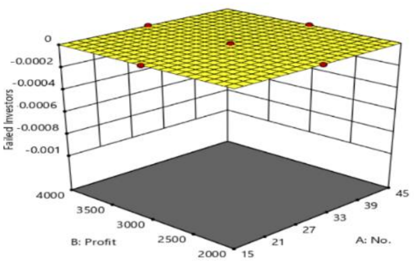


Fig 10 (a): Effects of Profit and No. of Investors on Failed Investors

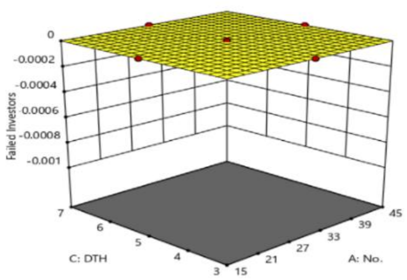


Fig 10 (b): Effects of DTH and No. of Investors on Failed Investors

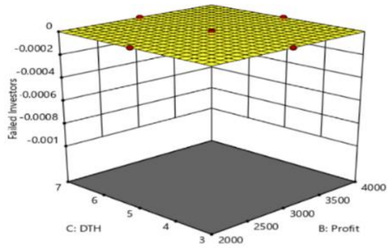


Fig 10 (c): Effects of DTH and Profit on Failed Investors

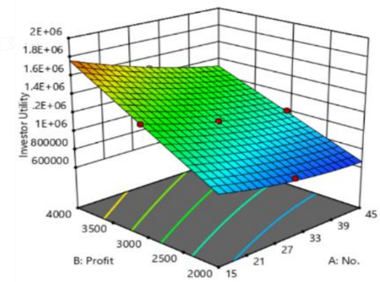


Fig 11 (a): Effects of Profit and No. of Investors on Investor Utility

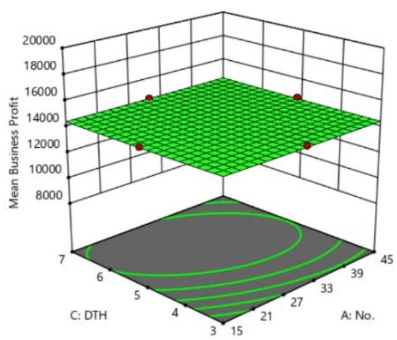


Fig 11 (b): Effects of DTH and No. of Investors on Investor Utility

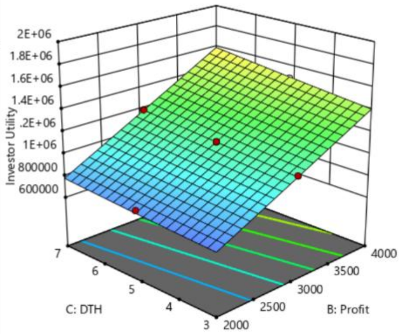


Fig 11 (c): Effects of DTH and Profit on Investor Utility

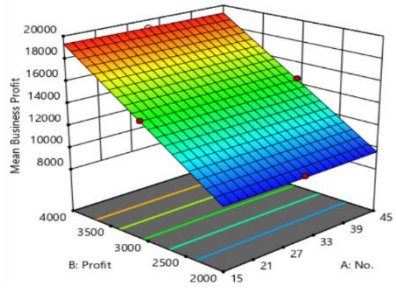


Fig 12 (a): Effects of Profit and No. of Investors on Mean Business Profit

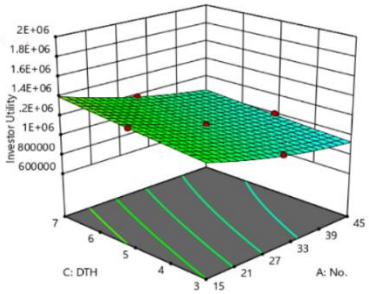


Fig 12 (b): Effects of DTH and No. of Investors on Mean Business Profit

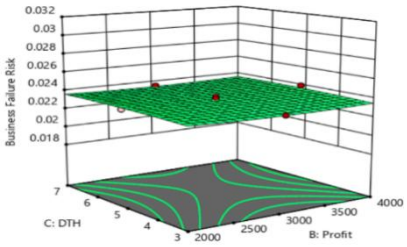


Fig 12 (c): Effects of DTH and Profit on Mean Business Profit

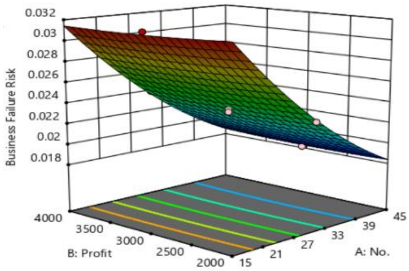


Fig 13 (a): Effects of Profit and No. of Investors on Business Failure Risk

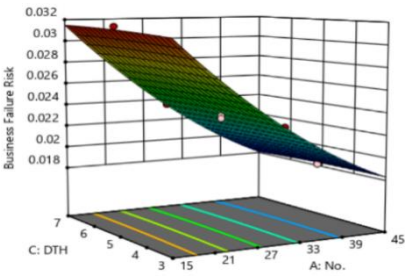


Fig 13 (b): Effects of DTH and No. of Investors on Mean Business Failure Risk

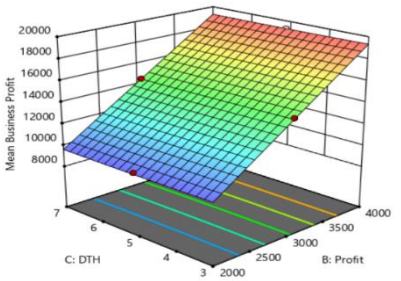


Fig 13 (c): Effects of DTH and Profit on Business Failure Risk

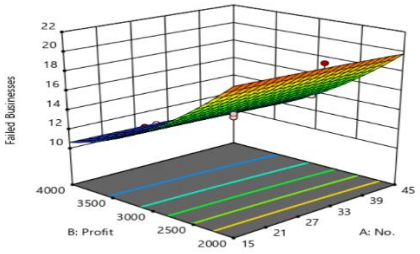


Fig 14 (a): Effects of Profit and No. of Investors on Failed Business

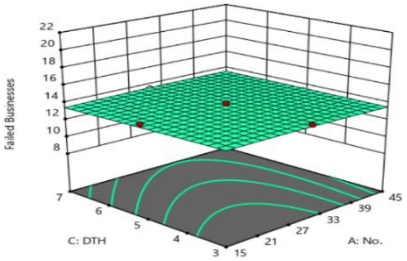


Fig 14 (b): Effects of DTH and No. of Investors on Failed Business

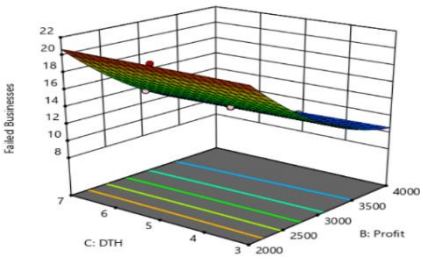


Fig 14 (c): Effects of DTH and Profit on Failed Business

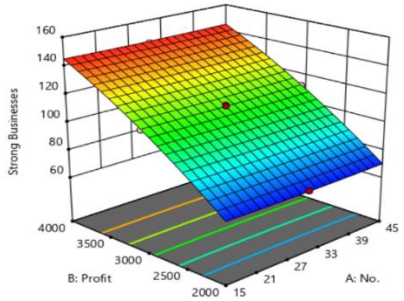


Fig 15 (a): Effects of Profit and No. of Investors on Strong Business

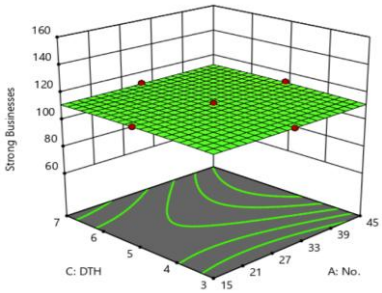


Fig 15 (b): Effects of DTH and No. of Investors on Strong Business

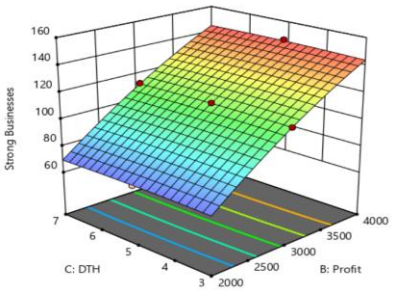


Fig 15 (c): Effects of DTH and Profit on Strong Business

In Fig 10 (a, b, c), the response surfaces of the no. of Failed Investors with respect to the interactive effects of input variables are shown. Since there are no failed investors in any of the experimental iterations of the PPO-ABM model, these graphs do not add value to the discussion. The ANOVA table is also redundant and is excluded.

The next RSM plots show the output variable, Investor Utility (Fig. 11). The effects of the stand-alone input factors are significant for all three variables. The combined effect of AC is not significant, whereas the effects of AB and BC are significant.

For 'Mean Business Profit', only the Initial Profit of the business has emerged as significant with a p-value less than 0.0001 (Fig. 12). None of the interactive effects are statistically significant for this response variable. The patterns of remaining response variables can be inferred from the respective ANOVA tables and response surfaces plotted in Figures 13, 14, and 15. One common phenomenon observed in these three cases is the insignificance of the interaction effects of the input variables when considered two at a time.

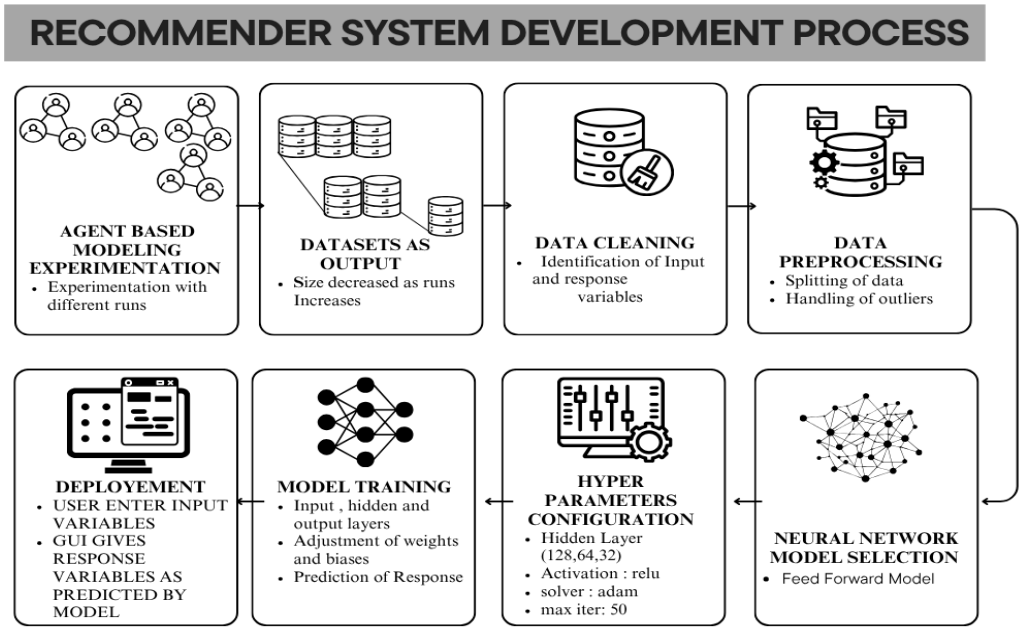


Fig. 16. ANN training and development of the Recommender System

3.4.Development of the ANN-trained Recommender System

In the second phase of experimentation, a new set of experiments is designed and conducted in the PPO-ABM. Six input variables and eight output variables are included in this exhaustive experimentation. Each input variable is assigned three levels, including low, medium, and high, thus making a total of 729 experimental scenarios. Each scenario is run 10 times to account for experimental stochasticity. The results are recorded at the end of each simulation run. A random sample of some of the obtained results is shown in Appendix B. An Artificial Neural Network (ANN) is trained on this data, followed by the development of a Recommender System (RS). The steps followed in this RS development are summarized in Fig. 16. The multilayer perceptron regressor (MPR) employed in this study is a type of feed-forward neural network trained using backpropagation. The chosen architecture consists of three hidden layers with 128, 64, and 32 neurons, respectively. The Rectified Linear Unit (ReLU) is used as an activation function. The algorithm used for updating the network weights during training is the Adam optimizer with an initial

learning rate of 0.001. The model is implemented in Python using Visual Studio Code as the development environment. A screenshot of the developed Recommender System is shown in Fig. 17. It includes the recommended ranges for the input variables and a sample of output values. The developed recommender system can be utilized for further convenient experimentation and policy modeling of the complex economic system under consideration. It enables non-technical users to run what-if scenarios and implement AI-driven decision-making in investment policy modeling

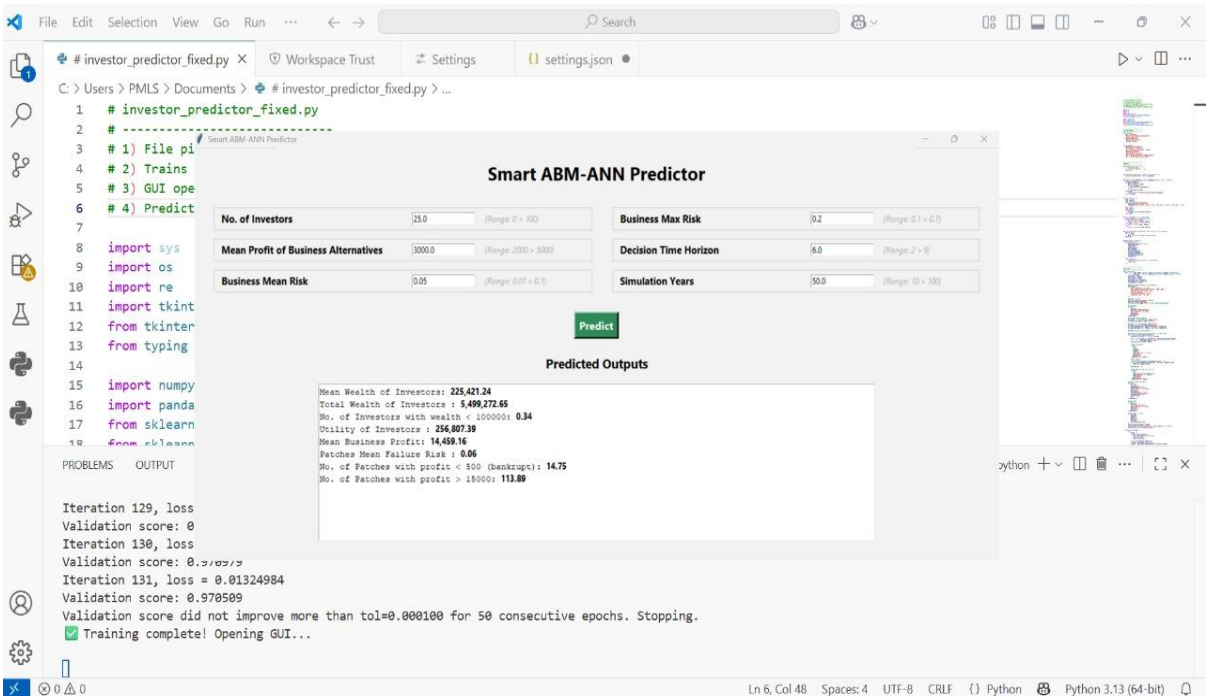


Fig. 17. The ANN-trained Recommender System

4. Conclusion

In line with the aims of this study, it introduced reinforcement learning (RL)-based intelligent behavior in the agent's decision-making process for the business investment agent-based model (ABM). Among the three RL-ABM models, Proximal Policy Optimization (PPO-ABM) proved to be the most effective for the response variables and achieved system-level targets. A two-phase, thorough experimental study was conducted in the PPO-ABM, resulting in the development of novel predictive empirical models, the generation of response surfaces, and the development of an artificial neural network-based recommender system.

The comparative evaluation of four RL-ABM models has provided in-depth and novel insights into the intelligent behavior of the complex economic environment under consideration. The investor agents and business alternatives showed stronger overall financial standing due to their intelligent decisions and adaptive behavior in the PPO-ABM. The developed predictive mathematical models and response surfaces represent the formal input/output relationship that emerged from the detailed experimentation conducted in the PPO-ABM. The recommender system enables non-technical users to perform a wide range of experimental analyses and evaluate alternative policy

scenarios with ease. In doing so, the paper addresses the critical gap between data-driven modeling and real-world usability.

However, the study presented has some limitations. It primarily focuses on a specific business investment problem, which involves several initial assumptions. Moreover, it is limited to the application of only three specific RL themes. The results obtained may vary in a given simulation due to stochasticity. The developed ABM models can be modified to represent more realistic investment environments by considering additional factors. Many other RL algorithms can be tested by integrating them into investment models.

Author Contributions

All authors contributed equally in preparing and writing the manuscript.

Funding

This research received no external funding.

Data Availability Statement

The PPO-ABM model is publicly available for download from the CoMSES Computational Model Library [46] . During simulations of the presented models, a large volume of data is generated and used to train an artificial neural network (ANN) and develop predictive mathematical models. A sample of the generated data is included in the manuscript. The developed agent-based model, recommender system, and associated source code are also publicly available on GitHub at: https://github.com/muhkh/rl-abm-investment-model. The remaining data are available from the authors upon reasonable request.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Hutahayan, B., Fadli, M., Amimakmur, S. A., & Dewantara, R. (2024). Investment decision, legal certainty and its determinant factors: Evidence from the Indonesia Stock Exchange. Cogent Business & Management, 11(1), Article 2332950. https://doi.org/10.1080/23311975.2024.2332950.

[2] Wilensky, U., & Rand, W. (2015). An introduction to agent-based modeling: Modeling natural, social, and engineered complex systems with NetLogo. The MIT Press.

[3] Rekik, Y. M., Hachicha, W., & Boujelbene, Y. (2014). Agent-based modeling and investors' behavior explanation of asset price dynamics on artificial financial markets. Procedia Economics and Finance, 13, 30-46. https://doi.org/10.1016/S2212-5671(14)00428-6.

[4] Avci, E., Bunn, D., Ketter, W., & van Heck, E. (2019). Agent-level determinants of price expectation formation in online double-sided auctions. Decision Support Systems, 124, Article 113068. https://doi.org/10.1016/j.dss.2019.05.008.

[5] Takahashi, H., & Terano, T. (2003). Agent-based approach to investors' behavior and asset price fluctuation in financial markets. Journal of Artificial Societies and Social Simulation, 6(3).

[6] Souissi, M. A., Bensaid, K., & Ellaia, R. (2018). Multi-agent modeling and simulation of a stock market. Investment Management and Financial Innovations, 15(4), 123-134. https://doi.org/10.21511/imfi.15(4).2018.10.

[7] Mastroeni, L., Naldi, M., & Vellucci, P. (2023). Personal finance decisions with untruthful advisors: An agent-based model. Computational Economics, 61(4), 1477-1522. https://doi.org/10.1007/s10614-022-10256-4.

[8] Ali, M., Akhtar, K., & Jahanzaib, M. (2019). An MCDM-based dynamic patch sensing cycle for agent-based modeling and simulation of global facility location [Conference presentation]. 25th International Conference on Multiple Criteria Decision Making, Istanbul Technical University, Istanbul, Turkey.

[9] Assenza, T., Delli Gatti, D., & Grazzini, J. (2015). Emergent dynamics of a macroeconomic agent-based model with capital and credit. Journal of Economic Dynamics and Control, 50, 5-28. https://doi.org/10.1016/j.jedc.2014.07.001.

[10] Ezzat, H. M. (2020). Behavioral agent-based framework for interacting financial markets. Review of Economics and Political Science, 5(2), 94-115. https://doi.org/10.1108/REPS-03-2019-0037.

[11] Stefan, F. M., & Atman, A. P. F. (2017). Asymmetric return rates and wealth distribution influenced by the introduction of technical analysis into a behavioral agent-based model [Preprint]. arXiv https://arxiv.org/abs/1711.08282.

[12] Sueyoshi, T., & Tadiparthi, G. R. (2008). An agent-based decision support system for wholesale electricity market. Decision Support Systems, 44(2), 425-446. https://doi.org/10.1016/j.dss.2007.05.007.

[13] Mizuta, T. (2020). An agent-based model for designing a financial market that works well. In 2020 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE. https://doi.org/10.1109/SSCI47803.2020.9308376.

[14] Trimborn, T. (2019). A macroscopic portfolio model: From rational agents to bounded rationality. Mathematics and Financial Economics, 13(3), 491-518. https://doi.org/10.1007/s11579-019-00235-z.

[15] Todd, A., Beling, P., Scherer, W., & Yang, S. Y. (2016). Agent-based financial markets: A review of the methodology and domain. In 2016 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE. https://doi.org/10.1109/SSCI.2016.7850016.

[16] Farmer, J. D., Patelli, P., & Zovko, I. I. (2005). The predictive power of zero intelligence in financial markets. Proceedings of the National Academy of Sciences of the United States of America, 102(6), 2254-2259. https://doi.org/10.1073/pnas.0409157102.

[17] Monti, C., Pangallo, M., De Francisci Morales, G., & Bonchi, F. (2023). On learning agent-based models from data. Scientific Reports, 13, Article 9268.
https://doi.org/10.1038/s41598-023-35536-3.

[18] Vitali, S., Battiston, S., & Gallegati, M. (2016). Financial fragility and distress propagation in a network of regions. Journal of Economic Dynamics and Control, 62, 56-75. https://doi.org/10.1016/j.jedc.2015.10.003.

[19] Kroujiline, D., Gusev, M., Ushanov, D., Sharov, S. V., & Govorkov, B. (2016). Forecasting stock market returns over multiple time horizons. Quantitative Finance, 16(11), 1695-1712. https://doi.org/10.1080/14697688.2016.1176241.

[20] Tubbenhauer, T., Fieberg, C., & Poddig, T. (2021). Multi-agent-based VaR forecasting. Journal of Economic Dynamics and Control, 131, Article 104231. https://doi.org/10.1016/j.jedc.2021.104231.

[21] Franke, R., & Westerhoff, F. (2012). Structural stochastic volatility in asset pricing dynamics: Estimation and model contest. Journal of Economic Dynamics and Control, 36(8), 1193-1211. https://doi.org/10.1016/j.jedc.2011.10.004.

[22] Patsatzis, D. G., Russo, L., Kevrekidis, I. G., & Siettos, C. (2023). Data-driven control of agent-based models: An equation/variable-free machine learning approach. Journal of Computational Physics, 478, Article 111953. https://doi.org/10.1016/j.jcp.2023.111953.

[23] Pruna, R. T., Polukarov, M., & Jennings, N. R. (2020). Loss aversion in an agent-based asset pricing model. Quantitative Finance, 20(2), 275-290. https://doi.org/10.1080/14697688.2019.1655784.

[24] Wei, L., Chen, S., Lin, J., & Shi, L. (2025). Enhancing return forecasting using LSTM with agent-based synthetic data. Decision Support Systems, 193, Article 114452. https://doi.org/10.1016/j.dss.2025.114452.

[25] Grazzini, J., Richiardi, M. G., & Tsionas, M. (2017). Bayesian estimation of agent-based models. Journal of Economic Dynamics and Control, 77, 26-47. https://doi.org/10.1016/j.jedc.2017.01.014.

[26] Cramer, S., & Trimborn, T. (2019). Stylized facts and agent-based modeling [Preprint]. arXiv. https://arxiv.org/abs/1912.02684.

[27] Lamperti, F., Roventini, A., & Sani, A. (2018). Agent-based model calibration using machine learning surrogates. Journal of Economic Dynamics and Control, 90, 366-389. https://doi.org/10.1016/j.jedc.2018.03.011.

[28] Ale Ebrahim Dehkordi, M., Lechner, J. M., Ghorbani, A., Nikolic, I., Chappin, E. J. L., & Herder, P. M. (2023). Using machine learning for agent specifications in agent-based models and simulations: A critical review and guidelines. Journal of Artificial Societies and Social Simulation, 26(1), Article 9. https://doi.org/10.18564/jasss.5016.

[29] Ali, H. M. K., Akhtar, K., Farooq, S., & Jahanzaib, M. (2018). A dynamic hybrid modelling approach for global facility location. Proceedings of the International Conference on Computers and Industrial Engineering, CIE, 2018-December.

[30] Ali, H. M. K. (2015). Global facility location: A hybrid modeling approach [Doctoral dissertation, University of Engineering & Technology Taxila].

[31] Padur, K., Borrion, H., & Hailes, S. (2025). Using agent-based modelling and reinforcement learning to study hybrid threats. Journal of Artificial Societies and Social Simulation, 28(1), Article 1. https://doi.org/10.18564/jasss.5539.

[32] Maringer, D., & Ramtohul, T. (2012). Regime-switching recurrent reinforcement learning for investment decision making. Computational Management Science, 9(1), 89-107. https://doi.org/10.1007/s10287-011-0131-1.

[33] Maeda, I., deGraw, D., Kitano, M., Matsushima, H., Sakaji, H., Izumi, K., & Kato, A. (2020). Deep reinforcement learning in agent-based financial market simulation. Journal of Risk and Financial Management, 13(4), Article 71. https://doi.org/10.3390/jrfm13040071.

[34] Shavandi, A., & Khedmati, M. (2022). A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets. Expert Systems with Applications, 208, Article 118124. https://doi.org/10.1016/j.eswa.2022.118124.

[35] Dicks, M., Paskaramoorthy, A., & Gebbie, T. (2024). A simple learning agent interacting with an agent-based market model. Physica A: Statistical Mechanics and Its Applications, 633, Article 129363. https://doi.org/10.1016/j.physa.2023.129363.

[36] He, X.-Z., & Lin, S. (2022). Reinforcement learning equilibrium in limit order markets. Journal of Economic Dynamics and Control, 144, Article 104497. https://doi.org/10.1016/j.jedc.2022.104497.

[37] Zhou, X., Lin, S., & He, X.-Z. (2025). Reinforcement learning and rational expectations equilibrium in limit order markets. Journal of Economic Dynamics and Control, 172, Article 104991. https://doi.org/10.1016/j.jedc.2024.104991.

[38] Lussange, J., Lazarevich, I., Bourgeois-Gironde, S., Palminteri, S., & Gutkin, B. (2021). Modelling stock markets by multi-agent reinforcement learning. Computational Economics, 57(1), 113-147. https://doi.org/10.1007/s10614-020-10038-w.

[39] Railsback, S. F., & Grimm, V. (2019). Agent-based and individual-based modeling: A practical introduction (2nd ed.). Princeton University Press. https://doi.org/10.2307/jj.28274141.

[40] Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). The MIT Press.

[41] Srivastava, U., Aryan, S., & Singh, S. (2025). A risk-aware reinforcement learning reward for financial trading [Preprint]. arXiv. https://arxiv.org/abs/2506.04358.

[42] Wunder, M., Littman, M. L., & Babes, M. (2010). Classes of multiagent Q-learning dynamics with ε-greedy exploration. In Proceedings of the 27th International Conference on Machine Learning. Omnipress.

[43] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533. https://doi.org/10.1038/nature14236.

[44] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms [Preprint]. arXiv. https://arxiv.org/abs/1707.06347.

[45] Kobayashi, T. (2023). Proximal policy optimization with adaptive threshold for symmetric relative density ratio. Results in Control and Optimization, 10, Article 100192. https://doi.org/10.1016/j.rico.2022.100192.

[46] Ali, M. K., Ali, H., & Mohammad, H. (2026, May 5). *Integrating reinforcement learning in agent-based modeling for dynamic investment decisions* (Version 1.0.0) [Computer model]. CoMSES Computational Model Library. https://doi.org/10.25937/644j-cv09

Appendix A1: ANOVA for Investor Mean Wealth

Source	Sum of Squares	df	Mean Square	F-value	p-value	
Model	1.255E+12	9	1.394E+11	1012.93	< 0.0001	significant
A-No.	1.894E+11	1	1.894E+11	1376.47	< 0.0001	
B-Profit	1.039E+12	1	1.039E+12	7553.21	< 0.0001	
C-DTH	2.193E+08	1	2.193E+08	1.59	0.2385	
AB	1.815E+10	1	1.815E+10	131.91	< 0.0001	
AC	4.953E+07	1	4.953E+07	0.3599	0.5634	
BC	37743.95	1	37743.95	0.0003	0.9871	
A^2	3.388E+09	1	3.388E+09	24.62	0.0008	
B^2	4.375E+06	1	4.375E+06	0.0318	0.8624	
C^2	7.122E+07	1	7.122E+07	0.5175	0.4902	
Residual	1.239E+09	9	1.376E+08			
Lack of Fit	9.772E+08	5	1.954E+08	2.99	0.1554	not significant
Pure Error	2.614E+08	4	6.535E+07			
Cor Total	1.256E+12	18				

Appendix A2: ANOVA for Investor Total Wealth

Source	Sum of Squares	df	Mean Square	F-value	p-value	
Model	2.024E+15	9	2.249E+14	2984.94	< 0.0001	significant
A-No.	1.096E+15	1	1.096E+15	14546.50	< 0.0001	
B-Profit	8.282E+14	1	8.282E+14	10990.43	< 0.0001	
C-DTH	1.224E+11	1	1.224E+11	1.62	0.2344	
AB	9.604E+13	1	9.604E+13	1274.50	< 0.0001	
AC	6.974E+08	1	6.974E+08	0.0093	0.9255	
BC	3.991E+09	1	3.991E+09	0.0530	0.8231	
A^2	2.604E+12	1	2.604E+12	34.56	0.0002	
B^2	2.950E+08	1	2.950E+08	0.0039	0.9515	
C^2	4.310E+10	1	4.310E+10	0.5720	0.4688	
Residual	6.782E+11	9	7.536E+10			
Lack of Fit	4.430E+11	5	8.859E+10	1.51	0.3563	not significant
Pure Error	2.353E+11	4	5.881E+10			
Cor Total	2.025E+15	18				

Appendix A3: ANOVA for Investor Utility

Source	Sum of Squares	df	Mean Square	F-value	p-value	
Model	1.835E+12	9	2.039E+11	883.98	< 0.0001	significant
A-No.	2.757E+11	1	2.757E+11	1195.24	< 0.0001	
B-Profit	1.472E+12	1	1.472E+12	6379.98	< 0.0001	
C-DTH	4.596E+10	1	4.596E+10	199.24	< 0.0001	
AB	2.585E+10	1	2.585E+10	112.04	< 0.0001	
AC	7.308E+08	1	7.308E+08	3.17	0.1088	
BC	4.899E+09	1	4.899E+09	21.24	0.0013	
A^2	5.007E+09	1	5.007E+09	21.70	0.0012	
B^2	1.793E+07	1	1.793E+07	0.0777	0.7867	
C^2	4.171E+07	1	4.171E+07	0.1808	0.6807	
Residual	2.076E+09	9	2.307E+08			
Lack of Fit	1.657E+09	5	3.315E+08	3.16	0.1436	not significant
Pure Error	4.189E+08	4	1.047E+08			
Cor Total	1.837E+12	18				

Appendix A4: ANOVA for Mean Business Profit

Source	Sum of Squares	df	Mean Square	F-value	p-value	
Model	2.339E+08	9	2.599E+07	3055.63	< 0.0001	significant
A-No.	218.11	1	218.11	0.0256	0.8763	
B-Profit	2.339E+08	1	2.339E+08	27498.06	< 0.0001	
C-DTH	5022.28	1	5022.28	0.5904	0.4619	
AB	2460.25	1	2460.25	0.2892	0.6038	
AC	2315.65	1	2315.65	0.2722	0.6144	
BC	2991.90	1	2991.90	0.3517	0.5677	
A^2	974.71	1	974.71	0.1146	0.7427	
B^2	79.50	1	79.50	0.0093	0.9251	
C^2	3121.24	1	3121.24	0.3669	0.5596	
Residual	76554.20	9	8506.02			
Lack of Fit	72424.26	5	14484.85	14.03	0.0121	Not significant
Pure Error	4129.94	4	1032.48			
Cor Total	2.340E+08	18				

Appendix A5: ANOVA for Business Failure Risk

Source	Sum of Squares	df	Mean Square	F-value	p-value	
Model	0.0005	9	0.0001	3036.64	< 0.0001	significant
A-No.	0.0004	1	0.0004	26367.34	< 0.0001	
B-Profit	9.617E-10	1	9.617E-10	0.0571	0.8165	
C-DTH	3.517E-08	1	3.517E-08	2.09	0.1825	
AB	2.503E-09	1	2.503E-09	0.1486	0.7089	
AC	2.781E-09	1	2.781E-09	0.1650	0.6941	
BC	1.514E-09	1	1.514E-09	0.0898	0.7712	
A^2	9.607E-06	1	9.607E-06	570.11	< 0.0001	
B^2	4.256E-10	1	4.256E-10	0.0253	0.8772	
C^2	3.377E-09	1	3.377E-09	0.2004	0.6650	
Residual	1.517E-07	9	1.685E-08			
Lack of Fit	1.461E-07	5	2.922E-08	21.06	0.0056	Not significant
Pure Error	5.549E-09	4	1.387E-09			
Cor Total	0.0005	18				

Appendix A6: Failed Businesses

Source	Sum of Squares	df	Mean Square	F-value	p-value	
Model	271.78	9	30.20	420.19	< 0.0001	significant
A-No.	0.0122	1	0.0122	0.1705	0.6894	
B-Profit	256.44	1	256.44	3568.25	< 0.0001	
C-DTH	0.1254	1	0.1254	1.75	0.2191	
AB	0.1682	1	0.1682	2.34	0.1604	
AC	0.0288	1	0.0288	0.4007	0.5425	
BC	0.0041	1	0.0041	0.0564	0.8177	
A^2	0.0007	1	0.0007	0.0094	0.9249	
B^2	8.75	1	8.75	121.72	< 0.0001	
C^2	1.423E-06	1	1.423E-06	0.0000	0.9965	
Residual	0.6468	9	0.0719			
Lack of Fit	0.2761	5	0.0552	0.5958	0.7104	not significant
Pure Error	0.3707	4	0.0927			
Cor Total	272.43	18				

Appendix A7: Strong Businesses

Source	Sum of Squares	df	Mean Square	F-value	p-value	
Model	14513.94	9	1612.66	2721.44	< 0.0001	significant
A-No.	0.2657	1	0.2657	0.4484	0.5199	
B-Profit	14420.25	1	14420.25	24334.86	< 0.0001	
C-DTH	0.1210	1	0.1210	0.2042	0.6621	
AB	0.1128	1	0.1128	0.1904	0.6729	
AC	2.05	1	2.05	3.46	0.0958	
BC	0.0210	1	0.0210	0.0355	0.8548	
A^2	0.0482	1	0.0482	0.0814	0.7819	
B^2	58.52	1	58.52	98.75	< 0.0001	
C^2	0.9581	1	0.9581	1.62	0.2354	
Residual	5.33	9	0.5926			
Lack of Fit	4.17	5	0.8344	2.87	0.1642	not significant
Pure Error	1.16	4	0.2903			
Cor Total	14519.28	18				

Appendix B: A Random Sample from the Large Experimental ABM Dataset used for Neural Network Model Training

	Input Variables					Response Variables						
Run	No. of Investors	Business Mean Profit	Business Mean Risk	Business Max Risk	DTH	Mean Wealth of Turtles	Total Wealth of Turtles	Utility for Investors	Business Mean Profit	Patches Mean Annual Risk	No. of investors profit <500	No. of Investors with profit > 15000
1	15	2000	0.01	0.1	3	292071.1	4381067	287354.8	4498.167	0.047567	49	18
2	15	2000	0.01	0.1	3	312852.8	4692792	367957	4834.325	0.045487	38	21
3	15	2000	0.01	0.1	3	291563.6	4373455	283015.4	4302.957	0.049077	43	13
1346	15	3000	0.05	0.3	8	2731053	4.10E+07	4049719	27921.12	0.116508	4	172
1347	15	3000	0.05	0.3	8	2940272	4.41E+07	4473143	26181.46	0.122879	3	183
1348	15	3000	0.05	0.3	8	1866737	2.80E+07	1660650	19615.54	0.113925	11	145
1349	15	3000	0.05	0.3	8	2397800	3.60E+07	1973420	28303.29	0.118672	7	171
2186	15	5000	0.08	0.1	3	5832577	8.75E+07	5398277	39526.48	0.084935	2	222
2187	15	5000	0.08	0.1	3	9331311	1.40E+08	1.09E+07	41523.93	0.085482	3	210
2188	15	5000	0.08	0.1	3	5421785	8.13E+07	5762352	45289.97	0.085861	3	227
2189	15	5000	0.08	0.1	3	4224387	6.34E+07	4417511	44732.37	0.085464	5	228
3126	25	2000	0.08	0.2	8	279620.9	6990522	227974	4494.584	0.128631	48	17
3127	25	2000	0.08	0.2	8	266281.3	6657033	206828.1	4340.846	0.130779	40	11
3128	25	2000	0.08	0.2	8	224740.8	5618519	324971.5	4868.043	0.131993	44	18
3280	25	3000	0.01	0.1	6	505966.8	1.26E+07	656441.5	6432.864	0.045871	40	41
3281	25	3000	0.01	0.1	6	1514186	3.79E+07	1887568	14149.21	0.03237	6	100
3282	25	3000	0.01	0.1	6	1815162	4.54E+07	2412680	15742.8	0.028899	12	122
3385	25	3000	0.01	0.2	6	5506849	1.38E+08	7577048	26122.49	0.048529	6	162
3386	25	3000	0.01	0.2	6	4310842	1.08E+08	5202457	24726.34	0.043606	5	178
3387	25	3000	0.01	0.2	6	2884290	7.21E+07	3458224	23990.89	0.03793	4	187
3388	25	3000	0.01	0.2	6	3162555	7.91E+07	4169864	26827.99	0.051189	5	184
5012	40	2000	0.01	0.2	8	307103.6	1.23E+07	286110.4	4735.767	0.087988	44	19
5013	40	2000	0.01	0.2	8	364181.2	1.46E+07	649093.6	4369.445	0.087886	47	12
5014	40	2000	0.01	0.2	8	328727.1	1.31E+07	469648.8	4515.4	0.080675	41	14
5234	40	2000	0.05	0.2	3	1333940	5.34E+07	1275604	9754.867	0.070286	15	77
5235	40	2000	0.05	0.2	3	851143.4	3.40E+07	982645.7	9921.551	0.075929	15	71
5236	40	2000	0.05	0.2	3	956036.1	3.82E+07	923099.3	8630.832	0.07455	23	57