MOGO-AFNNet: A Deep Learning and Multi-Objective Genetic Algorithm Framework for Intelligent logistics Warehouse Layout and Inventory Control

Kai Wang [1*], Yameng Bai [2]
[1]College of Innovation and Entrepreneurship, jiaozuo university, Jiaozuo,Henan ,454000, China
[2]College of Information Engineering, jiaozuo university, Jiaozuo,Henan , 454000, China
E-mail: wangkai198980@hotmail.com
[*]Corresponding author

Keywords: logistics, warehouse, inventory regulation, multi-objective genetic algorithm-driven adaptive fuzzy neuro network (MOGO-AFNNet), stockouts

Received: August 1, 2025

In modern logistics, warehouse layout and inventory control face challenges such as demand variability, inefficient space utilization, and frequent stockouts, where traditional analytical systems often fail to adapt in real time. To address these issues, this study proposes a Multi-Objective Genetic Algorithm-driven Adaptive Fuzzy Neuro Network (MOGO-AFNNet), which integrates an Adaptive Fuzzy Neuro Network (AFNNet) for demand forecasting with the NSGA-II genetic algorithm for multi-objective optimization. The Smart Logistics Supply Chain Dataset, comprising real-time IoT-based records of shipments, delays, stock levels, and operational costs, was employed. Data preprocessing was performed using Z-score normalization to standardize features, followed by Principal Component Analysis (PCA) to extract key variables such as reorder frequency, lead time variability, and item popularity. The AFNNet component enabled adaptive inventory regulation under uncertainty, while NSGA-II optimized warehouse layout and inventory strategies across conflicting KPIs. Experimental evaluation showed that the proposed framework achieved 97.5% accuracy, 98% precision, 96.8% recall, and a 97.9% F1-score, significantly outperforming baseline models. ANOVA confirmed significant performance differences among models (F = 21.47, p = 0.0004). These results demonstrate that MOGO-AFNNet offers a scalable and robust solution for intelligent logistics, reducing stockouts and enhancing warehouse efficiency in dynamic operational environments.

Povzetek: Članek predlaga MOGO-AFNNet, ki na logističnih podatkih združi napovedovanje povpraševanja z adaptivno mehko-nevronsko mrežo ter večciljno optimizacijo za sočasno optimiranje razporeditve skladišča in zalog, kar zmanjšuje primanjkljaje in dviga učinkovitost.

1 Introduction

Smart logistics warehouse design/inventory management is a revolutionary way of handling contemporary supply chains [1]. In the current dynamic and very competitive market, organizations are under a lot of pressure to maximize effectiveness, minimize operational expenses, and be quick in addressing their needs [2]. To achieve these expectations, the logistics businesses must move beyond traditional operations and adopt intelligent systems incorporating automation, real-time data, and smart technologies [3]. It is the design of a logistics warehouse, which defines the overall result of material management, storage, and distribution [4]. Considering a warehouse layout has been optimized, reducing travel time and maximizing use of space and free flow of items [5]. Adopting the intelligent design principles can dynamically reconfigure storage areas, enhance workflow, and enhance access to items that are frequently moved [6]. Figure 1 shows the logistics warehouse layout and inventory control.

Inventory control on its part guarantees the provision of the right quantity of goods at the right time, avoiding overstocking and under-stocking [7]. Smart inventory solutions include IoT sensors, Radio Frequency Identification (RFID) tracking, and Artificial Intelligence (AI) based demand prediction to ensure real-time accuracy and visibility [8]. The systems enable stock record automation, automatic triggering of replenishments, and elimination of human factors [9].

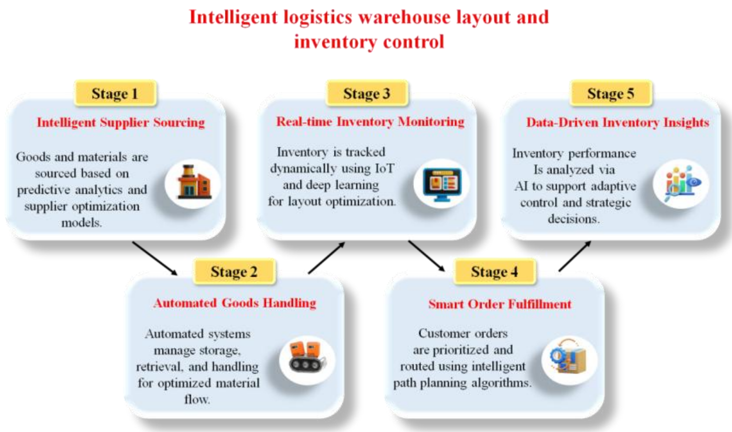

Figure 1: Five-stage intelligent logistics framework optimizing sourcing, handling, and inventory.

Smart warehouse design and inventory management, in combination with each other, create the core of the agile and resilient logistics networks. They make business processes simpler, as well as boost service levels and respond rapidly to market changes [10]. With logistics transforming into a digital operation, intelligent systems cannot be considered a luxury anymore but an essential element of the organizational processes that give more companies and a competitive advantage to keep pace with the dynamic customer demands. Conventional practices in logistics were dependent upon planning manually and rule-driven systems. Machine Learning (ML) and Deep Learning (DL) have made optimization of warehouse layout, warehouse demand forecasting, and inventory control in real-time more accurate and efficient, with methods including Support Vector Machine (SVM), K-Means Clustering, Artificial Neural Networks (ANN), and Long Short-Term Memory (LSTM) networks.

Although ML and DL approaches to logistics have various advantages, they have limitations that include data dependence, complicated model-building, interpretability, and inability to handle sudden changes. Also, its installation involves implementation of huge computational units, experienced human resources, and interconnection with established systems; it is costly and time-consuming for most organizations. The aim was to develop an intelligent logistics framework using a MOGO-AFNNet model that integrates adaptive fuzzy neural networks with a multi-objective genetic algorithm to optimize warehouse layout, improve goods flow, and enhance inventory control while minimizing costs and stockouts.

1.1 Key contribution

➢ Data collection: The real-time logistics data for optimizing supply chains, enabling predictive modeling, anomaly detection, and efficient warehouse and inventory management.

➢ Data preprocessing: Applied Z-Score normalization to standardize input features, ensuring consistent data scaling and improving model performance and convergence stability.

➢ Feature extraction: Used PCA to extract key variables, reduce dimensionality, and retain critical variance in logistics datasets.

➢ Proposed method: Integrated MOGO and AFNN to optimize warehouse layout and inventory control by balancing KPIs and adapting to demand uncertainty.

The structural framework of the research is listed as follows: The significant background of the research is provided in Section 1. A list of literature reviews was provided in Section 2. The method is explained in Section 3. Results and discussion section contained Section 4. Section 5 provides the conclusion.

1.2 Research questions

1. How can the integration of deep learning–based demand forecasting with multi-objective genetic algorithms improves both inventory management and warehouse layout optimization in dynamic logistics environments?
2. To what extent does the MOGO-AFNNet framework enhance operational efficiency, reduce stockouts, and maintain scalability compared to existing domain-specific intelligent warehousing methods?

2 Related works

Table 1 presents the reviewed studies employ AI, IoT, and optimization techniques to improve inventory control, path planning, warehouse layout, and resource scheduling. They show gains in efficiency, accuracy, cost reduction, and adaptability. However, most are domain-specific or limited in scalability, highlighting the need for integrated, real-world applicable frameworks.

Table 1: Summary of related works

Reference	Objective Function(s)	Method Used	Performance	Limitations
[11]	Minimize task execution time, cost, and maximize execution efficiency	Improved hybrid-parameter Ant Colony Algorithm with reinforcement learning	Achieved 16052 s packing time, 29865-yuan cost, 95.65% execution rate, 9.54% cost reduction	Validation limited to logistics scheduling tasks; generalizability to broader logistics contexts not addressed
[12]	Overcome control challenges in intelligent warehousing	Neuro-Fuzzy Dynamic Inventory Regulation Model (NFDIRM) combining Radial Basis Function Neural Network (RBFNN) + fuzzy logic	Higher inventory turnover, fewer stockouts, reduced costs vs. Economic Order Quantity (EOQ) & AutoRegressive Integrated Moving Average (ARIMA)	Complexity of implementation; real-world scalability not discussed
[13]	Minimize path length, completion time; maximize punctuality and dispatch success	Deep Reinforcement Learning (DRL) combined with CNN	Achieved 92–95% punctuality/dispatch success, reduced path length and time	Tested only; broader real-world scalability and dynamic uncertainties not fully validated
[14]	Optimize warehouse location & minimize distribution costs	Internet of Things (IoT) + Blockchain	Higher positioning accuracy, reduced computational load & costs	Applicability to large-scale logistics not fully validated
[15]	Intelligent crane scheduling in steel production	IoT + AI + Light Detection and Ranging (LiDAR) + Multi-Agent Reinforcement Learning (MARL) + optimized anti-swing control	Achieved unmanned operation & improved production	High technology & infrastructure cost; domain-specific
[16]	Onion buffer stock monitoring	IoT + Radio Frequency Identification (RFID) + cloud integration	Improved traceability, spoilage prevention, proactive environmental control	Focused on single product type (onions), generalization needed
[17]	Human & goods tracking in warehouses	You Only Look Once version 5 (YOLOv5) + Deep Simple Online and Realtime Tracking (DeepSORT)	Accurate, robust tracking within 30 ms	Limited to specific warehouse environments
[18]	Optimize e-commerce warehouse layout	Genetic Algorithm (GA)	Reduced handling costs; 39.25% improvement, outperformed Particle Swarm Optimization (PSO) & Simulated Annealing (SA)	May require high computational resources
[19]	Improve Warehouse Management (WM) efficiency & reduce costs	Intelligent decision system using GA, SA, and Ant Colony Optimization (ACO)	Significant efficiency & cost-effectiveness gains	Scalability to complex warehouses untested
[20]	Categorization of warehouse goods	Artificial Neural Network (ANN) enhanced by Black Hole Algorithm (BHA)	High accuracy with minimal input parameters	Focused on categorization only, not full warehouse management (WM)
[21]	Improve order picking efficiency	Smart lighting systems in warehouses	Promising efficiency improvements	Underexplored, needs more industrial applications

[22]	Enhance warehousing efficiency & safety	Programmable Logic Controller (PLC) + Human-Computer Interaction (HCI) integration	Better speed, space utilization, data efficiency, and reliability	Technology adaptation costs
[23]	Improve manual order picking	Smart Scale (Design Science Research, DSR; weight sensors)	Better accuracy, less human error, efficiency gains	Limited to lightweight items
[24]	Enable intelligent warehouse management & optimized distribution	IoT + AI integration	Real-time monitoring, optimized supply chain	Broad scope; practical case studies limited
[25]	Optimize path selection, task scheduling, resource allocation under dynamic demand	DRL with Multi-Task Learning (MTL) and Q-learning	Achieved 15% cost reduction, 85% resource utilization, strong adaptability in dynamic scenarios	Validation limited to simulations; real-world deployment and scalability not demonstrated

2.1 Research gaps

Prior research advanced intelligent warehousing through neuro-fuzzy inventory models [12], IoT-driven crane automation [15], and genetic algorithm–based layout optimization [18]. However, these solutions remained domain-specific, addressing isolated issues like inventory turnover, crane scheduling, or layout efficiency, with limited integration of predictive intelligence and multi-objective optimization. To overcome this, the proposed MOGO-AFNNet framework integrates deep learning–based demand forecasting with multi-objective genetic algorithms for layout and inventory optimization. This unified approach enhances adaptability, aligns with diverse KPIs, consolidates spatial and stock management, and delivers greater accuracy, fewer stockouts, and improved operational efficiency across logistics environments.

3 Methodology

The dataset provides real-time IoT-based logistics data for analyzing shipments, delays, asset tracking, and optimizing supply chain operations. Data preprocessing used Z-score normalization to standardize input features. PCA was applied for feature extraction, reducing dimensionality while retaining key variance. The proposed MOGO-AFNNet model integrates an Adaptive Fuzzy Neural Network for demand forecasting with a Multi-Objective Genetic Algorithm to optimize warehouse layout and inventory control strategies across multiple KPIs. Figure 2 illustrates the overall flow of the methodology.

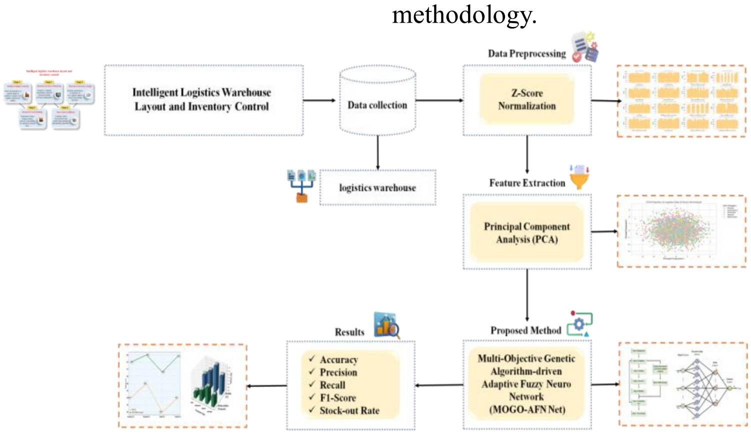

Figure 2: Methodology flow for intelligent logistics warehouse layout

3.1 Data collection

The Smart Logistics Supply Chain Dataset delivers real-time logistics and warehouse operations data. It includes time-stamped records of asset tracking, shipments, delays, package movements, and related categorical and numerical details captured over an extended period. Ideal for optimization, predictive modeling, supply chain simulations, and anomaly detection, it allows users to analyze operational workflow, delivery performance, and inventory throughput for smarter logistics decision-making. It includes stock levels, demand variability, lead time, reorder frequency, item popularity, and operational costs. These attributes support predictive modeling, inventory optimization, and warehouse layout planning for intelligent logistics systems. The Smart Logistics Supply Chain Dataset was split 80% for training and 20% for testing to evaluate model performance

Source:https://www.kaggle.com/datasets/ziya07/logistics-warehouse-dataset/data.

3.2 Data preprocessing using Z-Score normalization

Z-Score normalization standardizes input features by transforming them to have zero mean and unit variance, enhancing the performance of deep learning models in intelligent warehouse layout and inventory control. This technique adjusted the scales of datasets to a standard scale where the mean was equal to zero and the standard deviation was one. The data had undergone preliminary processing to make it ready for primary processing and analysis. Z-score normalization, also known as zero normalization, involves dividing each feature's mean and standard deviation in a training dataset by several variables. The general formula in equation (1) quantifies the transformation.

$$d' = \frac{(d-\mu)}{\sigma} \quad (1)$$

In this equation, d represents the original data, μ is the mean, σ is the standard deviation, and d' is the normalized value. The z-score technique was used to normalize all of the dataset's features. To use the standard deviation and mean for each feature as weights in the system design, it was essential to preserve those values once a set of training data had been calculated. Normalization adjusts the scale of behavioral and categorical data to make them suitable for modelling. Figure 3 shows the Z-score distribution in logistics variables.

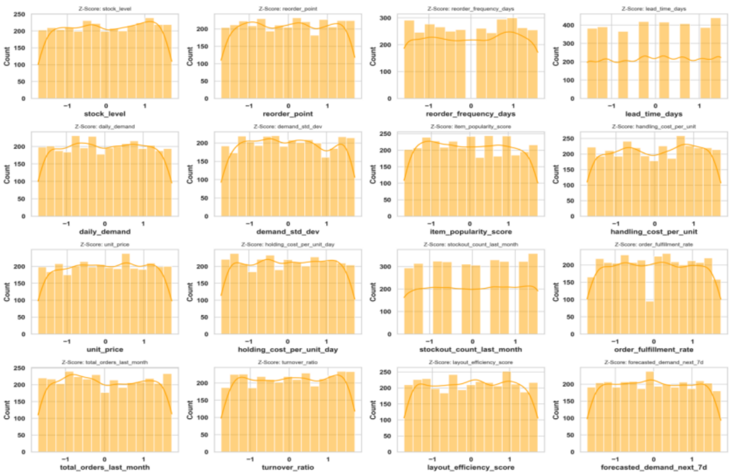

Figure 3: Z-score normalized logistics features distribution for warehouse optimization modeling.

3.3 Feature extraction using PCA

PCA reducing dimensionality, identifying key operational patterns, improving data-driven decision-making, and enabling efficient optimization of space, inventory flow, and resource allocation strategies. The fundamental idea of PCA is to maximize the range of the data by linearly transforming it into a low-dimensional subspace. In mathematical terms, every sample in a data set with l samples is m-dimensional when the class label is ignored. Consider that$w_1, w_{2,} \dots\dots\dots w_l \ {}_{\epsilon\, \Re^m}$. The subsequent steps in the PCA computation. The l-dimensional mean vector μ can be calculated in equation (2).

$$\mu = \frac{1}{l}\sum_{j=1}^{l} w_j \quad (2)$$

Determine the anticipated covariance matrix T for the collected data by Equation (3).

$$T = \frac{1}{L}\sum_{j=1}^{l} (w_j - \mu)(w_j - \mu)^s$$

(3)

Determine the associated eigenvectors and eigenvalues of T, where $\lambda_1 \geq \lambda_{2,} \geq \cdots . \geq \lambda_{l,} \geq 0$.

Determine the l principal components from the l initial variables by Equations (4).

$$z_1 = b_{11}w_1 + b_{12}w_2 + \cdots + a_{1l}w_l$$
$$z_2 = b_{21}w_1 + b_{22}w_2 + \cdots + a_{2l}w_l$$
$$\dots\dots$$
$$z_l = b_{l1}w_1 + b_{l2}w_2 + \cdots + a_{ll}w_l$$

(4)

There is no correlation between z_l. The majority of the initial variance in the data set can be explained by z_1, the majority of the remaining variance can be explained by z_2, etc., which is shown in Equation (5).

$$\gamma l = \frac{\lambda_1 + \lambda_{2,} + \cdots \dots \lambda_n}{\lambda_1 + \lambda_{2,} + \cdots \dots + \lambda_n + \cdots + \lambda_l} \geq 80\%$$

(5)

Where,γl is the proportion that was maintained in the data representation. When using PCA for feature extraction, the main components that can explain at least 80% of the variation in the overall data should be retained.

The principal components of demand variability, supplier lead time risk, product popularity, and cost-efficiency trade-offs were mapped as input features to AFNNet, guiding the forecasting model.

Figure 4 displays the PCA visualization for intelligent logistics control.

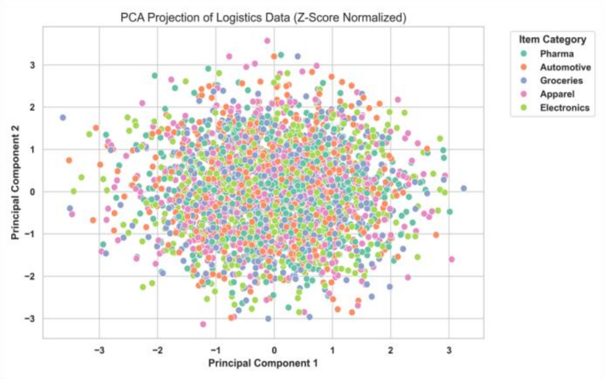

Figure 4: PCA projection showing category clusters in normalized logistics dataset.

3.4 MOGO-AFNNet

The hybrid MOGO-AFNNet framework is developedmodel synergistically combines the adaptive learning capabilities of fuzzy neural networks with the global search efficiency of multi-objective genetic algorithms. The AFNNet dynamically models nonlinear and uncertain logistics environments, enabling precise control over inventory levels, retrieval strategies, and spatial configurations. Simultaneously, the MOGO explores optimal solutions across conflicting objectives such as maximizing space utilization, minimizing stockouts, and reducing inventory holding costs. The hybrid system iteratively evolves optimal warehouse layouts and inventory rules by evaluating fitness across multiple performance metrics. This enables real-time adaptation to demand fluctuations, operational constraints, and logistical complexities. MOGO-AFNNet enhances decision-making accuracy, operational agility, and overall warehouse efficiency in intelligent logistics, outperforming traditional static and heuristic-based methods in dynamic and data-intensive supply chain environments. Algorithm 1 shows the pseudo code of MOGO-AFNNet.

Algorithm 1: MOGO-AFNNet

```
Initialize:
  Population size N = 30
  Max generations G = 50
  Mutation rate μ = 0.1
  Crossover rate = 0.9
  Number of fuzzy inputs = 4
  Number of fuzzy MFs per input = 3
  Neural network layers = [4, 8, 1]
  For i = 1 to N:
    Randomly generate chromosome[i]:
      - Fuzzy rule base
      - Membership function parameters
      - Neural net weights
    Evaluate chromosome[i] using multi-objective fitness:
      - Accuracy
      - Interpretability
      - Complexity
Set generation = 0
While generation < 50:
  For i = 1 to N:
    Decode chromosome[i]:
      → Extract fuzzy rules, MFs, and NN weights
```

Train Adaptive Fuzzy Neuro-Network (AFNNet) on training data
Compute objectives:
→ Obj1: MSE = 0.0225
→ Obj2: Interpretability = 0.1 (inverse of number of rules)
→ Obj3: Complexity = 40 (total number of parameters)
Apply NSGA-II sorting:
- Rank chromosomes into Pareto fronts (F1, F2, ...)
- Compute crowding distance within each front
Crossover (probability = 0.9):
For j = 1 to N/2:
- Select parents P1, P2
- Apply uniform crossover:
→ Swap fuzzy rules and NN weight segments
→ Child1, Child2 created
Mutation (probability μ = 0.1):
For each child chromosome:
- Randomly mutate:
→ Membership function parameter
→ Add/remove fuzzy rule
→ Perturb NN weight
Create a combined population (N = 60)
Select top 30 using elitism and crowding distance
generation = generation + 1
Return:
- Final Pareto front of optimized AFNNet models
- Best trade-offs: e.g.,

3.4.1 AFNNet

The AFNNet component uses fuzzy logic and neural learning to model warehouse operations, incorporating features from PCA. The fuzzy layer generates rules based on continuous features, while the neural layers learn membership and weight parameters. The outputs are domain-relevant inventory control signals for operators.

- The input language parameters are sent by the input layer. $w_j (j = 1, \dots, n)$ to the layers below werew_j stands for the AFNNet control scheme's input elements. The sliding surface in this warehouse layout is denoted by w_1, and its differentiation is denoted by w_2.
- The input values are represented by the membership layer using the Gaussian membership functions listed in equation (6).

$$\mu_j^i(w_j) = \exp\left[\frac{-\left(w_j - n_j^i\right)^2}{\left(D_j^i\right)^2}\right] \qquad (6)$$

Where n_j^i and $d_j^i (j = 1, \dots, m; i = 1, \dots, M_{oj})$ are the Gaussian function's standard deviation and mean in the ith term corresponding variable of the jth input,w_j is the node of the layers,and $exp(.)$ was the exponential operator. The symbol M_{oj} is used to indicate the distinct number of functions.

To make notation easier, define parameter vectors n and d as follows: $m = [n_1^1 \dots n_1{}^{M_{O1}} n_2^1 \dots n_m{}^{M_{Om}}]^S \epsilon\ Q^{M_q \times 1}$ $and \quad d = [d_1^1 \dots d_2^{M_{o2}} \dots d_m^1 \dots d_m^{M_{om}}]^S \epsilon\ Q^{M_q \times 1}$, where $M_q = \sum_{j=1}^m M_{oj}$ indicates the total number of functions for membership. Figure 5 contains Ob1 - Inventory Accuracy, Ob2 - Operational Efficiency, and Ob3 - System Complexity. Figure 5 shows the Structure Diagram of AFNNet.

- The fuzzy inference process is implemented via the rule layer, where each node multiplies its input signals and outputs the product. This layer's output is provided in equation (7).

$$k_l = \prod_{j=1}^m X_{ij}^l \mu_j^i (w_j) \qquad (7)$$

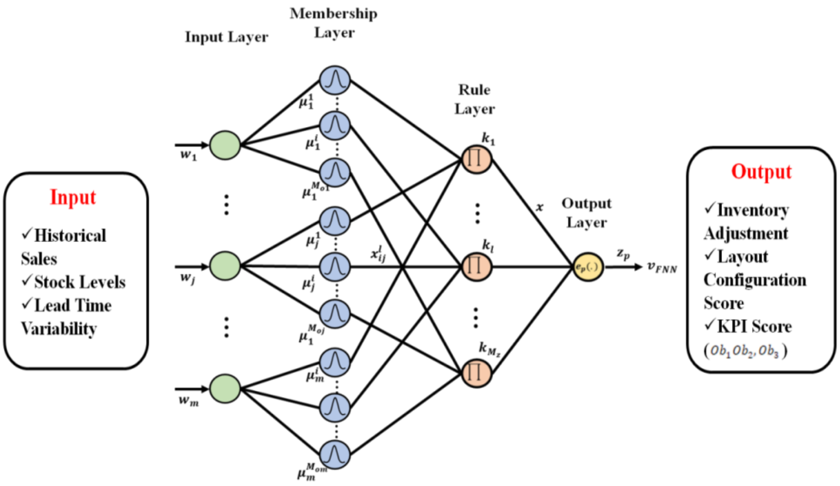

Figure 5: Structure of AFNNet.

Where the vector of parameters $k_l(l = 1, \ldots, M_z)$can collect all values; X_{ij}^l are the weights allocated to the members of the rule layer, which are considered to be equal to one, and M_z is the number of rules. Where$k = [l_1 \;\; l_2 \;\; \ldots l_{M_z}]^S$ is the kth outcome of the rule layers.

The outcome layers have four nodes that represent the output language variables. The output is calculated as the sum of every input signal of the following type by the output node $\mathcal{Z}_p$in equation (8).

$$\mathcal{Z}_p = e_{\mathrm{p}}\left(\textstyle\sum_{l=1}^{M_z} X_l \, k_l\right) \qquad (8)$$

The activation function is $e_p(.)$, and the following matrix can be used to collect the changeable weights between the output layer and the rule layers, indicated by X_l in equation (9).

$$x = [x_1 \, , x_2 \, , \ldots x_{Mz}] \qquad (9)$$

Additionally, the FNNet output can be rewritten as follows in equation (10).

$$\mathcal{Z}_p = v_{FNN}\left(x_j \, , x, n, d\right) = xk \qquad (10)$$

The tuning techniques for the major FNNet control and related network parameters are part of the AFNNet system. To preserve the strong control performance without the need for auxiliary compensated control, the FNNet control is made to resemble the law of the total sliding-mode control (TSMC). There is an ideal FNNet control v_{FNN}^* to learn the TSMC law v_{TSMC}, as per the strong approximation ability in equation (11).

$$v_{TSMC} = v_{FNN}^*\left(w_j, x^*, n^*, d^*\right) + \varepsilon \qquad (11)$$

Where x^*, n^*, and d^* are the ideal values for x, n, and d in the FNNet, and ε is the smallest reconstructed error.

The AFNNet output is the inventory adjustment signal used to update inventory rules and warehouse actions in equation (12).

$$\bar{v} = \hat{v}_{FNN}(w_j, \hat{x}, \hat{n}, \hat{d}) = \hat{x}\hat{k} \qquad (12)$$

Where $\hat{x}, \hat{n}$, and $\hat{d}$contain the estimates of x^*, n^*, and d^*, respectively, as supplied by the tuning procedure that will be presented later, and $\hat{k}$ represents the estimate of k^*, where k^* is the ideal value of k. The approximate error $\tilde{v}$ is defined as equations (12) from (13).

$$\tilde{v} = v_{TSMC} - \hat{v}_{FNN} = v_{FNN}^* + \varepsilon - \hat{v}_{FNN} \; (13)$$

Equation (14) is obtained by expanding $\tilde{v}$ in the Taylor sequence after the activation functions are transformed into a partially linear structure using the linearization technique.

$$\tilde{v} = \frac{\partial \hat{v}FNN}{\partial \hat{x}}\bigg|\hat{x} = x^*\left(x^{*S} - \hat{x}^S\right) + g_x + \frac{\partial \hat{v}FNN}{\partial \hat{k}}\bigg|\hat{k} = k^{*(k^* - \hat{k})} + g_k + \varepsilon$$
$$= \frac{\partial \hat{v}FNN}{\partial \hat{x}}\bigg|\hat{x} = x^{*\tilde{x}^S} + \frac{\partial \hat{v}FNN}{\partial \hat{k}}\bigg|\hat{k} = k^{*\hat{k}} g_x + g_k + \varepsilon = z_x{}^{\tilde{x}^S} + v_k\tilde{k} + g_x + g_k + \varepsilon \qquad (14)$$

Where $\tilde{x} = x^* - \hat{x} \epsilon Q^{1 \times M_z}$; $\tilde{k} = k^* - \hat{k} \in Q^{M_z \times 1}$; and $\hat{x} = x^*$; $v_k = \left(\partial \hat{v}FNN \big/ _{\partial \hat{k}}\right) | \; \hat{k} = k^*$; $z_x = (\partial \hat{v}FNN \big/ _{\partial \hat{x})|}$; and $g_x, g_k \in Q$are more complex terms.

Furthermore, the linearization technique is used once again to convert the membership functions into an incomplete linear form, enabling the expansion of $\tilde{k}$ in the Taylor sequence to yield equation (15).

$$\tilde{k} = \begin{bmatrix} \tilde{k}_1 \\ \tilde{k}_2 \\ \vdots \\ \tilde{k}_{M_z} \end{bmatrix} = \begin{bmatrix} \frac{\partial \hat{k}_1}{\partial \hat{n}^S} \\ \frac{\partial \hat{k}_2}{\partial \hat{n}^S} \\ \vdots \\ \frac{\partial \hat{k}_{M_z}}{\partial \hat{n}^S} \end{bmatrix} |\hat{n} = n^*(n^* - \hat{n}) + \begin{bmatrix} \frac{\partial \hat{k}_1}{\partial \hat{d}^S} \\ \frac{\partial \hat{k}_2}{\partial \hat{d}^S} \\ \vdots \\ \frac{\partial \hat{k}_{M_z}}{\partial \hat{d}^S} \end{bmatrix} |\hat{d} = d^*\left(d^* - \hat{d}\right) + g_{nd} \equiv k_n\tilde{n} + k_d\tilde{d} + g_{nd} (15)$$

The vector of higher-order terms is represented. The approximation error $\bar{v}$ can be rewritten as follows to replace (15) with (14).

$$\bar{v} = z_x\tilde{x}^S + v_k k_n\tilde{n} + v_k k_d\tilde{d} + z' \qquad (16)$$

Where $z' = g_x + g_k + v_k g_{nd} + \varepsilon$.

The AFNNet's inventory dynamics and parameter-adaptation behavior in warehouse operational contexts are improved through adaptation laws, ensuring stable forecasting during online updates. The open-source model includes formal adaptation equations for projection-based constraint handling, enhancing reasoning and reproducibility as shown in the equations (17-19), have positive learning rates denoted by $b_1, b_2, and\ b_3$, and positive parameter bounds denoted by a_x, a_n , and a_d.The vectors of

$z_x = \frac{\partial \hat{v}FNN}{\partial \hat{x}}$, $v_k = {\partial \hat{v}FNN}/{\partial \hat{k}}$, $k_n = [({\partial \hat{k}_1}/{\partial \hat{n}})\ [({\partial \hat{k}_2}/{\partial \hat{n}}) \cdots ({\partial \hat{k}_{M_z}}/{\partial \hat{n}})]^S, and$ $k_d = [({\partial \hat{k}_1}/{\partial d})\ ({\partial \hat{k}_2}/{\partial d})\left({\partial \hat{k}_{M_z}}/{\partial d}\right)]^S$ when the FNNet parameters' adaption laws in (17)–(19) are applied. s is simply computed using equations (6)–(8) and (12). The approximation error (16) is represented by the activation and membership functions' linearization around ideal parameters in (14) and (15), which is practical for linear for the parameters and the stability analysis. In actuality, the suggested AFNNet scheme can be implemented practically without the ideal parameters x^*, n^*, and d^*.

$$\dot{\hat{x}} = \begin{cases} -b_1\left(t\frac{\partial_y}{\partial f^S}az_x\right) & if\left(\left||\hat{x}|\right| < a_x\right) \\ -b_1\left(t\frac{\partial_y}{\partial f^S}az_x\right) & or\ \left(\left||\hat{x}|\right| = a_x and\ t\frac{\partial_y}{\partial f^S}az_x\hat{x}^S \geq 0\right) \\ +b_1[T\frac{\partial_y}{\partial f^S}az_x)(\hat{x}^S\hat{x}/|\hat{x}^2|)] & if\ \left(\left||\hat{x}|\right| < a_x and\ t\frac{\partial_y}{\partial f^S}az_x\hat{x}^S < 0\right) \end{cases} \qquad (17)$$

$$\dot{\hat{x}} = \begin{cases} -b_2(t\frac{\partial_y}{\partial f^S}av_k k_n)^S & if\ \left(\left||\hat{n}|\right| < a_n\right) \\ -b_2(t\frac{\partial_y}{\partial f^S}av_k k_n)^S & or\ \left(\left||\hat{x}|\right| = a_n and\ t\frac{\partial_y}{\partial f^S}av_k k_n\hat{n} \geq 0\right) \\ +b_2[t\frac{\partial_y}{\partial f^S}av_k k_n)(\hat{x}\hat{x}^S/|\hat{x}^2|)]^S & if\ (\left||\hat{n}|\right| = a_n and\ t\frac{\partial_y}{\partial f^S}av_k k_n\hat{n} < 0 \end{cases} \qquad (18)$$

$$\dot{\hat{d}} = \begin{cases} -b_3(t\frac{\partial_y}{\partial f^S}av_k k_n)^S & if\ \left(\left||\hat{d}|\right| < a_d\right) \\ -b_3(t\frac{\partial_y}{\partial f^S}av_k k_d)^S & or\ \left(\left||\hat{d}|\right| = a_d and\ t\frac{\partial_y}{\partial f^S}av_k k_d\hat{d} \geq 0\right) \\ +b_3[t\frac{\partial_y}{\partial f^S}av_k k_n)(\hat{x}\hat{x}^S/|\hat{x}^2|)]^S & if\ (\left||\hat{d}|\right| = a_d and\ t\frac{\partial_y}{\partial f^S}av_k k_d\hat{d} < 0 \end{cases} \qquad (19)$$

The following is a brief explanation of the fundamental concept of the projection technique in (17)–(19). If equations (17), (18), and (19), the If the parameter vector lies on the edge of the constraint set but is moving inward or within the set, the basic modifying law of the Lyapunov synthesis approach is applied. However, if the parameter vector is on the boundary of the constraint set and tends to move outside of it, the associated gradient vector should be projected onto the underlying hyper plane to ensure the constraints are not violated. As a result, the projection procedure in the FNN parameter adaptation laws will keep the estimates ($\hat{x}$, $\hat{n}$, and $\hat{d}$) of the optimum parameters inside the constraint sets. Figure 6 illustrates the AFNNet layer output for the intelligent logistics warehouse. Table 2 shows the hyperparameters for AFNNet.

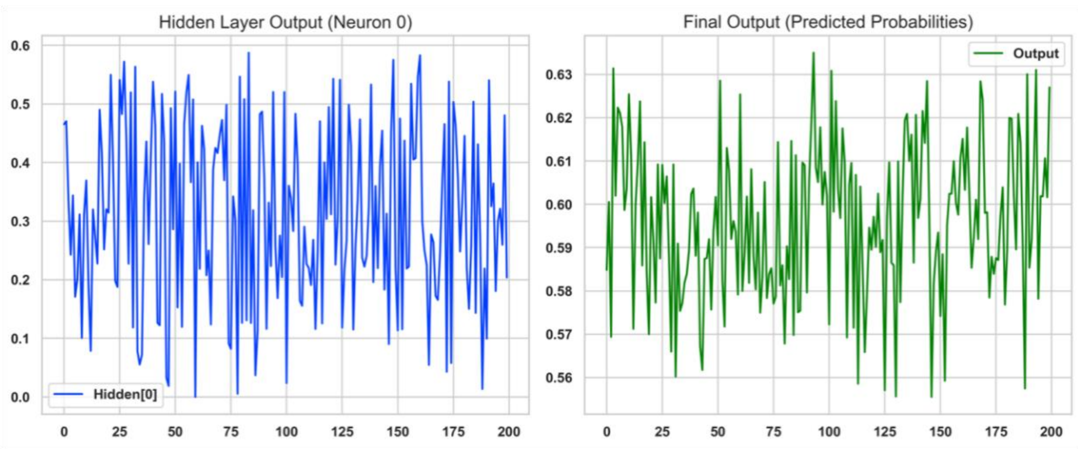

Figure 6: AFNNet hidden and output layers showing inventory prediction behavior.

Table 2: Hyperparameters for AFNNet

Parameter	Value
Fuzzy inputs	4
Membership functions	3 (Gaussian)
NN layers	[4, 8, 1]
Rule base	Dynamic
Learning method	TSMC-based
MSE	0.0225
Interpretability	0.1
Complexity	40

3.4.2 MOGO

The MOGO balance conflicting objectives such as space utilization, retrieval efficiency, and cost minimization, Facilitating flexible, data-driven choices for improved operational efficiency in intricate logistics settings.

- Selection Procedure: The following weighted sum method is a straightforward method of combining several desired functions into a fitness scalar function (Where all the objective functions are expected to be maximized) in equation (20).
- $e(w) = x_1 . e_1(w) + \cdots + x_j . e_j(w) + \cdots + x_m . e_m(w)$ (20)
- Where m is the total number of objective functions, x_m is a constant weight for $e_j(w)$, $e(w)$ is an integrated fitness function, $e_1(w)$ is the jth objective function, and w is a string. Genetic algorithms use constant weights and have a fixed search orientationx_j' in equation (21). Random weights are used in the selection process to find Pareto optimum solutions using a variety of search orientations. Utilize the procedure described in equation (21) to allocate a random real integer to each weight when two strings are chosen for a crossover operation.
- $x_j = \frac{random_j(.)}{\sum_{i=1}^{m} random_{i(.)}}, \quad j = 1,2, \ldots, n,$ (21)
- When a non-negative randomized number is denoted by $random_{i(.)}$.Equation (21) indicates that x_j is a real number within the [0, 11] closed interval. The following pair of strings is chosen using freshly provided weight values and so forth.
- Elite Preserve Strategy: A provisional collection of Pareto optimum solutions has been saved and modernized at every generation while the MOGO is running. At each generation, a specific number, $(say,\ M_{elite})$ of people are chosen at random from the collection. MOGO, those solutions are employed by outstanding persons. The diversity of each group in MOGO is maintained as a result of the elite preservation technique.

Initialization: M_{pop} is the quantity of strings there are in each starting population; therefore, create an initial population with M_{pop} strings.

Evaluation: Determine the objective functions' values at the formed strings. The preliminary collection of Pareto optimum (PO) solutions.

Selection: Utilizing a randomly selected weight x_j' in equation (21), determine each string's fitness value. Using the following selection probability, choose two strings from the current population. In a population Ψ, the selection probabilities $O(w)$ of a string w are given in equations (22) and (23).

$$P(w) = \frac{e(w) - e_{\min}(\Psi)}{\sum_{w \epsilon \Psi} \{e(w) - e_{\min}(\Psi)\}} \quad (22)$$

Where

$$e_{\min}(\Psi) = \min\{e(w) | e \epsilon \Psi\} \quad (23)$$

${M_{pop}}/{2}$ pairs of threads were chosen from the existing populations using this process again.

Crossover: Use a creating two new strings using a crossover operation for every pair that is chosen. The crossover process generates M_{pop}.

Mutations: With a predetermined mutations probability, does a mutation process on every bit values in the strings of data produced by the intersect operation.

Elitists' strategy: Replace the M_{elite}strings that were randomly taken out of the set of M_{pop}strings produced by the earlier procedures with M_{elite} strings.

Termination test: Proceed to Step 1 contain the preset stopping condition is not met.

User selection: MOGO presents the final list of PO choices to make a decision. Following that, according to the decision-makers' inclinations. The specific characteristics of the issue determine the selection of genetic operations, such as mutation and crossover. Table 3 displays the hyperparameters of MOGO.

Table 3: Hyper parameters of MOGO

Parameter	Value
Population size (N)	30
Generations (G)	50
Crossover prob.	0.9
Mutation prob.	0.1
Elite size	User-defined
Selection	Random weight
Sorting	NSGA-II
Objectives	Accuracy, Interpretability, Complexity

4 Results and discussion

The experiment setup was performed using Python to implement the suggested method. The experimental results are explained in detail in this section. The following metrics were used to evaluate the suggested approach and ascertain its efficacy: Accuracy, Recall, Precision, F1-Score, and Stock-out Rate. Additionally, a comparative analysis was conducted with other current methods, such as the NFDIRM [12] and Deep Q-Networks (DQN) [26].

The existing methods of NFDIRM [12] and DQN [26], were comprehensively assessed based on the Smart Logistics Supply Chain Dataset. The results indicated that these conventional methods achieved results that were mediocre in terms of dealing with demand forecasting and inventory optimization. Conversely, the MOGO-AFNNet framework consistently outperformed

The 3D surface plot shows the variability of key performance indicators (KPIs) with the variation in the lead time and demand variability change. Such analysis assists in situating areas of sensitivity in performance (about intelligent logistics) by delays or by variable demands that affect the efficiency of operations. With this understanding incorporated into inventory control systems, the warehouse can adapt by using buffer stock optimization and supplier coordination as risk attenuating strategies. It improves planning of layout and responsiveness to create a robust and driven data framework for warehouse management. Figure 7 shows the lead time impact on intelligent logistics.

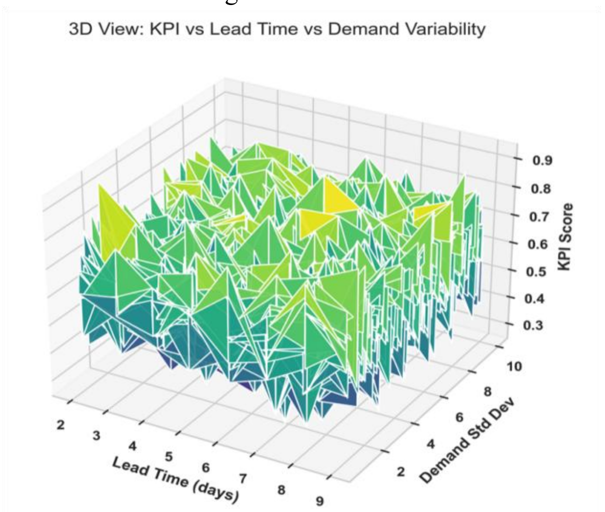

Figure 7: 3D surface plot showing the effect of lead time variability (x-axis) and demand fluctuations (y-axis) on KPIs (z-axis). Data from the Smart Logistics Supply Chain Dataset highlight how delays impact warehouse efficiency.

The bubble chart allows intelligent logistics as the chart exposes the association between stock and daily demand, where the size of the bubble shows the predicted demand and the color represents the KPI numbers. Multidimensional visualization can easily solve how to determine the right stock according to the demand patterns. When this wisdom is embedded in warehouse control systems, then stocks can be adjusted in real time, without excessive stocking of a product or running out of stock, and there is better accuracy in fulfilment. It also supports the smooth working of aligning inventory plans with predictive demand modeling to deal efficiently with the layout of the warehouse and inventory. Figure 8 displays the stock and demand relationship analysis.

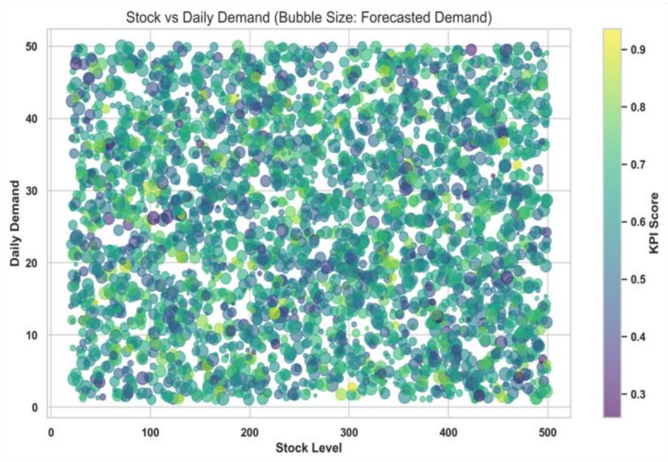

Figure 8: Bubble chart of stock levels (x-axis) vs. daily demand (y-axis), where bubble size represents forecasted demand and color encodes KPI values. Based on the Smart Logistics dataset, it reveals stock-demand alignment effects on fulfillment accuracy.

The turnover ratio distribution chart gives a picture of the performance of product categories in various areas of the warehouse. This trend is useful in intelligent logistics since it can determine where there is high and low turnover, allowing them to perform strategic placement decisions. By analyzing category-specific movement, warehouses can optimize inventory placement, reduce retrieval time, and enhance space utilization. Such data-driven insights, when combined with adaptive control systems, enable dynamic inventory adjustments, improve order fulfillment efficiency, and support real-time decision-making, key components of framework. Figure 9 shows the turnover distribution for intelligent logistics.

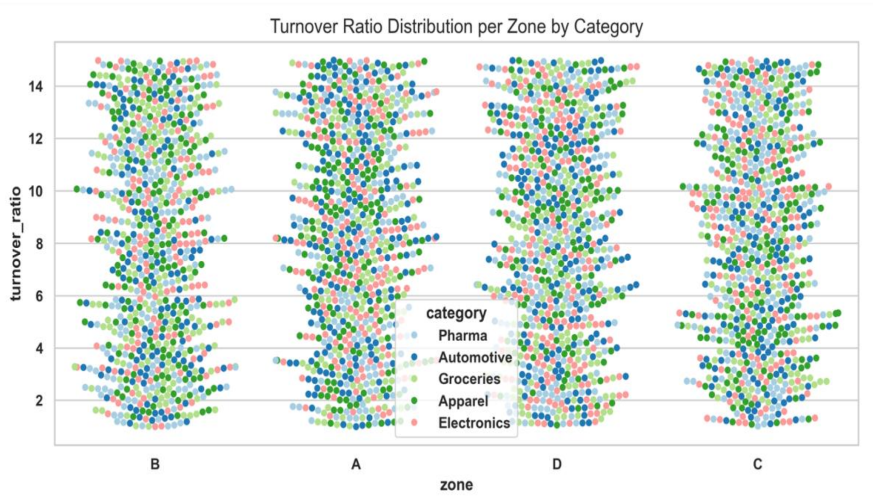

Figure 9: Turnover ratio distribution across warehouse sections (x-axis) and product categories (y-axis). Results from the Smart Logistics dataset expose spatial inefficiencies and imbalances in stock movement.

4.1 Performance metrics

Accuracy: The accuracy of intelligent logistics warehouse layout and inventory management is a measure of how well the system tracks the inventory needs and uses spaces most effectively and efficiently. The increased level of accuracy guarantees the minimization of stockouts, effective routes, and enhanced decisions within dynamic logistics settings.

Precision: Precision in intelligent logistics warehouse design and inventory control are indicators that gauge how well the system can recognize the right inventory activities, when restocking or retrieving inventory. High precision leads to minimum false operations, waste, and an improvement in efficiency in the management of space and the pace of orders.

Recall: Recall in smart logistics warehouse design and inventory management implies that a system can properly track all the associated inventory requirements and logistical tasks. Better recall will also mean that there are less errors of missing restocking or allocation instances, and the stocks will be available, and the operations will also be maintained.

F1-Score: F1-Score is the harmonic mean of precision and recall, striking the right balance between making true positive predictions (precision) and making comprehensive inventory events (recall). The F1-score is very important since it ensures that the operations of the warehouse and the inventory management system are efficient, reliable, and well optimized.

Stock-out Rate: Stock-out rate measures how often inventory levels fail to meet demand. Lower stock-out rates indicate effective forecasting, timely replenishment, and efficient inventory strategies, ensuring product availability and minimizing disruptions in supply chains.

The performance comparison between DQN and the proposed MOGO-AFNNet model highlights the superiority of the latter. The proposed MOGO-AFNNet achieved an accuracy of 97.5%, outperforming DQN's 95%. In terms of precision, MOGO-AFNNet reached 98%, while DQN attained 96%. The recall value improved from 94% (DQN) to 96.8% with MOGO-AFNNet. Similarly, the F1-score increased from 95% in DQN to 97.9% in the proposed method. These improvements indicate MOGO-AFNNet's enhanced ability to accurately predict, optimize, and control warehouse logistics processes, ensuring higher efficiency and reliability across various KPI. Table 4 and Figure 10 illustrate the performance comparison of MOGO-AFNNet for an intelligent logistics warehouse.

Table 4: Performance comparison of MOGO-AFNNet for intelligent logistics.

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
DQN [26]	95%	96%	94%	95%
MOGO-AFNNet [Proposed]	97.5%	98%	96.8%	97.9%

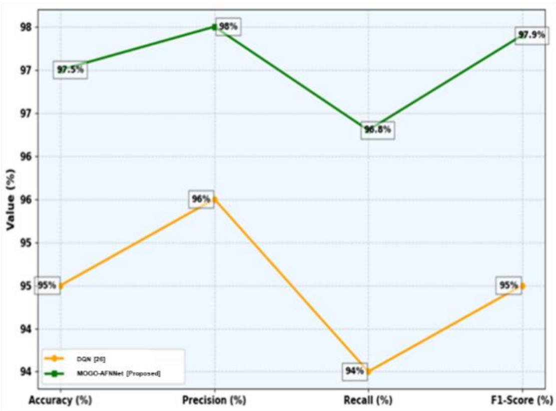

Figure 10: Performance comparison of DQN and MOGO-AFNNet across accuracy, precision, recall, and F1-score. Using the Smart Logistics dataset, results confirm superior predictive power of MOGO-AFNNet.

The comparative analysis of stock-out rates under various promotion types reveals the clear advantage of the proposed MOGO-AFNNet model, which outperformed NFDIRM. Under mild promotion, MOGO-AFNNet achieves a stock-out rate of 0.85%, significantly lower than NFDIRM's 1.80%. For moderate promotion, the proposed model records 1.42%, compared to 2.60% with NFDIRM. In the case of severe promotion, MOGO-AFNNet maintains a rate of 2.33%, outperforming NFDIRM's 3.90%. Overall, the comprehensive average stock-out rate is reduced from 2.77% (NFDIRM) to 1.57% with MOGO-AFNNet, highlighting its superior adaptability and efficiency in dynamic promotional scenarios. Table 5 and Figure 11 display the stock-out rate comparison for intelligent logistics control.

Table 5: Stock-out rate comparison for intelligent logistics control

Promotion Type	**Stock-out Rate (%)**	
	NFDIRM [12]	MOGO-AFNNet [Proposed]
Mild Promotion	1.80%	0.85%
Moderate Promotion	2.60%	1.42%
Severe Promotion	3.90%	2.33%
Comprehensive Average	2.77%	1.57%

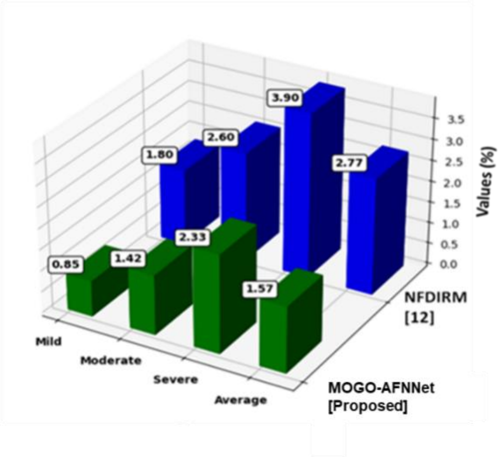

Figure 11: Stock-out rate under mild, moderate, and severe promotions (x-axis). Y-axis shows stock-out percentage. Comparison of NFDIRM and MOGO-AFNNet highlights MOGO-AFNNet's stronger demand-stock balance.

The comparison between NFDIRM and the MOGO-AFNNet model Inventory turnover rate across different commodity value tiers and seasonal intensities shows that MOGO-AFNNet exhibits slightly higher stock-out rates in this scenario. For low-priced goods, stock-out rates increase from 23.1% to 26.8% under low strength, 22.4% to 25.8% under medium intensity, and 21.7% to 25.4% under high strength. For mid-priced items, rates shift from 22.8% to 26.3%, 22.1% to 26.0%, and 21.4% to 25.6% respectively. Similarly, for high-ticket items, stock-outs rise from 22.5% to 25.9%, 21.8% to 24.8%, and 21.1% to 24.6%. Although the proposed model underperforms in terms of raw stock-out percentages here, these values may reflect trade-offs made to optimize other objectives such as cost efficiency, delivery speed, or inventory turnover in a multi-objective optimization setting. Table 6 and Figure 12 show the impact of seasonality on intelligent logistics control.

Table 6: Impact of seasonality on intelligent logistics control

Commodity Value Tier	Seasonality	NFDIRM [12] (%)	MOGO-AFNNet [Proposed] (%)
Low-Priced Goods	Low Strength	23.1%	26.8%
	Medium Intensity	22.4%	25.8%
	High Strength	21.7%	25.4%
Mid-Priced Items	Low Strength	22.8%	26.3%
	Medium Intensity	22.1%	26%
	High Strength	21.4%	25.6%
High-Ticket Items	Low Strength	22.5%	25.9%
	Medium Intensity	21.8%	24.8%
	High Strength	21.1%	24.6%

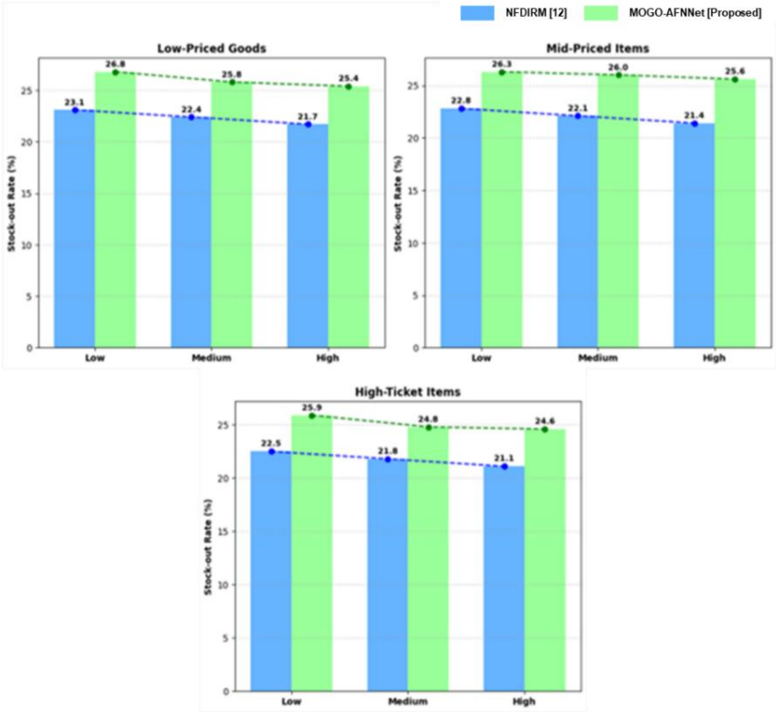

Figure 12: Seasonal intensity (x-axis) vs. stock-out percentage (y-axis) across low, mid, and high-value goods. Smart Logistics dataset results demonstrate how MOGO-AFNNet manages turnover efficiency under demand volatility.

The observed increase in stock-out rates for MOGO-AFNNet under specific scenarios results from inherent trade-offs in multi-objective optimization. The framework simultaneously balances cost minimization, retrieval speed, and turnover ratio. During extreme variability and seasonality, stock-out percentages rise slightly as resources are reallocated to maintain lower holding costs and maximize throughput. This reflects the essence of MOGO-AFNNet, where gains in one objective, such as space utilization or retrieval efficiency can compromise another, such as stock-out reduction. These outcomes highlight strategic trade-offs intrinsic to MOGO-AFNNet and emphasize its alignment with real-world logistics dynamics.

ANOVA (Analysis of variance)

Table 7 shows the ANOVA results for model comparison. The between-model variance (SS = 18.52) is significantly higher than within-model variance (SS = 4.33), yielding F = 21.47 and p = 0.0004. This indicates a statistically significant difference in performance among the models.

Table 7: Values of ANOVA

Source of Variation	SS	df	MS	F	p-value
Between Models	18.52	2	9.26	21.47	0.0004
Within Models	4.33	12	0.36		
Total	22.85	14			

Cross validation

Table 8 presents the cross-validation results for the proposed model across five folds. The RMSE values range from 4.10 to 4.35, MSE from 16.81 to 18.92, and MAE from 3.38 to 3.52. The average metrics indicate consistent predictive performance, with RMSE = 4.22, MSE = 17.80, and MAE = 3.45.

Table 8: Values of cross validation

Fold	RMSE	MSE	MAE
Fold 1	4.21	17.73	3.45
Fold 2	4.10	16.81	3.38
Fold 3	4.35	18.92	3.52
Fold 4	4.28	18.32	3.49
Fold 5	4.15	17.22	3.41
Average	4.22	17.80	3.45

Ablation study

Table 9 presents the ablation study results, showing the contribution of each component to model performance. The proposed MOGO-AFNNet outperforms individual components, achieving the highest specificity (0.902), AUC (0.947), and R^2 (0.925). This demonstrates the effectiveness of combining Z-score normalization, PCA, MOGO, and AFNNet for enhanced predictive accuracy.

Table 9: Values of the ablation

Model	Specificity	AUC	R^2
Z-Score Normalization	0.798	0.872	0.812
PCA	0.825	0.889	0.848
MOGO	0.842	0.903	0.871
AFNNet	0.861	0.915	0.884
MOGO-AFNNet (Proposed)	0.902	0.947	0.925

4.2 Discussion

NFDIRM [12] Concentrates on inventory management, enhancing turnover while decreasing stockouts. It is complicated to execute, and it lacks validity for large-scale dynamic warehouse situations. Ultimately, limited or nonexistent integration with other logistics functions limits adaptability. IoT-enabled crane automation is Typically not applied across logistics. Facilitates unmanned crane operations but; it has high technology and infrastructure costs [15]. The systems tend to be very specific for the steel warehouse context, and therefore, it is difficult to generalize across warehouse types or logistics issues. GA-based warehouse layout optimization Decreases handling costs and provides greater efficiency with warehouse layouts [18]. Requires significant computational resources, and does not analyze inventory management concurrently. Poor adaptability to dynamic multi-objective logistics problems. DQN [26] Provides flexibility for decision making for logistics tasks and adapting related decision patterns, facilitating dynamic path planning. It is limited when simultaneously having to adapt multiple objectives, e.g., inventory and layout optimization. The system has not been verified or validated for scale within large-scale logistics environments in the real world.

By combining deep learning-based demand forecasting with multi-objective genetic algorithm optimization, the MOGO-AFNNet framework proposed in this manuscript mitigates these shortfalls. In particular, the proposed framework accounts for inventory management, warehouse design, and productivity at the same time, minimizing stock outs and handling costs. The interconnectedness of these three elements results in improved adaptability, scalability, and applicability to real-world scenarios in diverse logistics environments.

The research's generalizability is limited by its single dataset, necessitating future work to utilize multiple datasets from real-world and simulated examples for greater validity and stronger reliability claims. Table 5 reveals higher stock-out rates under seasonality stress tests, indicating trade-offs in multi-objective optimization. MOGO-AFNNet prioritizes cost efficiency, turnover ratio, and retrieval speed, potentially increasing stock-out levels compared to NFDIRM.

5 Conclusion

The aim of this research was to develop an adaptive reinforcement learning framework for real-time robot parameter tuning to optimize energy efficiency, execution time, and motion smoothness under dynamic manufacturing conditions. Data were collected from onboard and external sensors, recording joint positions, velocities, accelerations, torques, energy consumption, and environmental contexts, ensuring comprehensive coverage of operational variations. Features such as motion characteristics, forces, energy usage, and contextual task information were extracted and prepared for trajectory optimization. Fifth-order B-spline interpolation was employed for smooth joint-space trajectory generation, while the Efficient Prairie Dog Optimized Dynamic Soft Actor-Critic (EPD-DSAC) model was proposed for real-time optimization. In this setup, Efficient Prairie Dog Optimization provided high-quality initial trajectory parameters, and DSAC reinforcement learning continuously refined them using sensor feedback to ensure stability, adaptability, and robustness. The assessment modeled normal and promotional demands, with reported seasonal variations ranging from 24.6% to 26.8%, which correspond to stock-out rates, providing context for performance interpretation. Future work will extend the model to stress-test performance under extreme demand shifts and other unforeseen operational conditions.

References

[1] Zhang, S., Han, Q., Zhu, H., Wang, H., Li, H., & Wang, K. (2025). Real-time task planning for order picking in intelligent logistics warehousing. *Scientific Reports, 15*(1), 7331.https://doi.org/10.1038/s41598-025-88305-9.

[2] Wu, Y. (2022). Warehouse layout optimization based on EIQ model. *International Core Journal of Engineering, 8*(1), 153–159.https://doi.org/10.6919/ICJE.202201_8(1).0022.

[3] Zhang, R., Zhou, X., Jin, Y., & Li, J. (2022). Research on intelligent warehousing and logistics management system of electronic market based on machine learning. *Computational Intelligence and Neuroscience, 2022*(1), 2076591. https://doi.org/10.1155/2022/2076591.

[4] Du, C. (2021). Logistics and warehousing intelligent management and optimization based on radio frequency identification technology. *Journal of Sensors, 2021*(1), 2225465. https://doi.org/10.1155/2021/2225465.

[5] He, D. (2022). Teaching practices of a warehousing management curriculum based on virtual reality simulation technology. *International Journal of Emerging Technologies in Learning (iJET), 17*(9), 96–109.https://doi.org/10.3991/ijet.v17i09.30939.

[6] Zhang, Y., & Pan, F. (2022). Design and implementation of a new intelligent warehouse management system based on MySQL database technology. *Informatica, 46*(3).https://doi.org/10.31449/inf.v46i3.3968.

[7] Odeyinka, O. F., & Omoegun, O. G. (2023). Warehouse operations: An examination of traditional and automated approaches in supply chain management. In *Operations Management—Recent Advances and New Perspectives*. IntechOpen.https://doi.org/10.5772/intechopen.113147.

[8] Zhai, Y., & Song, M. (2024). Design of intelligent logistics management system based on machine learning. *CAD and Applications, 21*(S27), 159–173.https://doi.org/10.14733/cadaps.2024.S27.159-173.

[9] Liu, X. (2025). Warehouse location selection and transportation cost control in performance network optimization. *Economics and Management Innovation, 2*(2), 100–107.https://doi.org/10.71222/zbjrx788.

[10] Khetani, N., Sharma, A., & Kulkarni, A. (2023, November). Real-time logistics and warehouse integration via networks of the future and dynamic digital twins. In *2023 IEEE Engineering Informatics Conference* (pp. 1–10). IEEE.https://doi.org/10.1109/IEEECONF58110.2023.10520525.

[11] Li, W., & Wang, L. (2025). Intelligent logistics resource scheduling based on hybrid parameter ant colony algorithm and reinforcement learning. *Informatica, 49*(13). https://doi.org/10.31449/inf.v49i13.6565.

[12] Xie, C., & Xie, C. (2025). Inventory control strategy based on neural network and fuzzy algorithm in intelligent warehousing system. *Discover Artificial Intelligence, 5*(1), 159.https://doi.org/10.1007/s44163-025-00423-5.

[13] Yang, Y., & Wang, K. (2025). Efficient logistics path optimization and scheduling using deep reinforcement learning and convolutional neural networks. *Informatica, 49*(16). https://doi.org/10.31449/inf.v49i16.7839.

[14] Chen, J., Xu, S., Liu, K., Yao, S., Luo, X., & Wu, H. (2022). Intelligent transportation logistics optimal warehouse location method based on Internet of Things and blockchain technology. *Sensors, 22*(4), 1544. https://doi.org/10.3390/s22041544.

[15] He, H. N., Wang, X. C., Peng, G. Z., Xu, D., Liu, Y., Jiang, M., Wu, Z. D., Zhang, D., & Yan, H. (2021). Intelligent logistics system of steel bar warehouse based on ubiquitous information. *International Journal of Minerals, Metallurgy and Materials, 28*(8), 1367–1377.https://doi.org/10.1007/s12613-021-2325-z.

[16] Fang, X., & Chen, H. C. (2022). Using vendor management inventory system for goods inventory management in IoT manufacturing. *Enterprise Information Systems*, *16*(7), 1885743. https://doi.org/10.1080/17517575.2021.1885743

[17] Xie, T., & Yao, X. (2023). Smart logistics warehouse moving-object tracking based on YOLOv5 and DeepSORT. *Applied Sciences, 13*(17), 9895. https://doi.org/10.3390/app13179895.

[18] Hu, X., & Chuang, Y. F. (2023). E-commerce warehouse layout optimization: Systematic layout planning using a genetic algorithm. *Electronic Commerce Research, 23*(1), 97–114. https://doi.org/10.1007/s10660-021-09521-9.

[19] Zhang, F., Yang, X., Dang, Y., Wu, C., & Chen, B. (2025). Construction and practice of intelligent warehouse management decision support system based on artificial intelligence. *Procedia Computer Science, 259*, 711–720.https://doi.org/10.1016/j.procs.2025.04.022.

[20] Veres, P. (2023). Increasing the efficiency of warehouse analysis using artificial intelligence. *Acta Logistica, 10*(3). https://doi.org/10.22306/al.v10i3.415.

[21] Füchtenhans, M., Grosse, E. H., & Glock, C. H. (2021). Smart lighting systems: State-of-the-art and potential applications in warehouse order picking. *International Journal of Production Research, 59*(12), 3817–3839.https://doi.org/10.1080/00207543.2021.1897177.

[22] Zhou, H. (2024). Intelligent warehouse control system based on PLC and human-computer interaction interactive control. *Procedia Computer Science, 243*, 440–449.https://doi.org/10.1016/j.procs.2024.09.054.

[23] Chou, T. H., You-Sheng, C., Chung-Wei, P., & Chang, H. H. (2024). A smart scale for efficient inventory management based on design science research principles. *Contemporary Economics, 18*(2), 153.https://doi.org/10.5709/ce.1897-9254.531%0A.

[24] Liu, T., & Ju, H. (2024). Research on the application of IoT and AI in modern logistics and warehousing. *Journal of Industrial Engineering and Applied Science, 2*, 1–4.https://doi.org/10.5281/zenodo.10755279.

[25] Zhu, X. (2025). Multi-task deep reinforcement learning for intelligent logistics path planning and scheduling optimization. *Informatica, 49*(20). https://doi.org/10.31449/inf.v49i20.7996.

[26] Pillella, S. (2025). Reinforcement learning for autonomous warehouse orchestration in SAP logistics execution: Redefining supply chain agility. *arXiv preprint arXiv:2506.06523*.https://doi.org/10.48550/arXiv.2506.06523.