Journal of Umm Al-Qura University for Engineering and Architecture (2025) 16:1864–1880
https://doi.org/10.1007/s43995-025-00213-1

ORIGINAL ARTICLE

Smart shelves: transforming retail stocking with internet of things and machine learning

Ali Abdullah S. AlQahtani[1] · Aadel A. Darrat[2] · Lonnie Turpin[3] · Thamraa Alshayeb[4]

Received: 3 June 2025 / Accepted: 25 August 2025 / Published online: 26 September 2025
© The Author(s) 2025

Abstract

Retail environments are increasingly leveraging technology to streamline operations and improve efficiency. This paper presents 'Smart Shelves', an innovative application that utilizes the Internet of Things (IoT) and machine learning (ML) to facilitate shelf replenishment. The application is designed to assist retail stock clerks in managing inventory more efficiently, reducing the potential for human error, and enhancing overall retail performance. The system's effectiveness is highlighted by its 99.35% accuracy in distinguishing between products, even when dealing with near-identical product features that require restocking. The system's operational efficiency is demonstrated by its 8.66-second average response time to issue notifications. A significant contribution of the proposed work is the development of a synthetic dataset that closely mirrors real-world retail conditions as obtaining a real-world dataset presented significant challenges. The findings demonstrate the transformative potential of IoT and ML technologies in the retail industry, particularly in the realm of shelf replenishment. Finally, this paper opens the door for future explorations that integrate the proposed system with existing inventory management software.

Keywords Shelf replenishment · On-shelf availability · Retail stock clerks · Internet of things · Machine learning · Retail stocking · IoT · ML

1 Introduction

The global retail industry, at the threshold of a new era, is undergoing a significant transformation. In 2022, the industry generated over 27 trillion U.S. dollars in sales, a figure that is projected to exceed 30 trillion U.S. dollars by 2024 [1]. This growth is not only a testament to the industry's economic significance but also a reflection of the challenges it faces, particularly in the realm of inventory management [2].

Traditional inventory tracking methods, such as manual counts and barcode scanning, have long been the industry standard. However, these methods are labor-intensive, prone to errors, and increasingly inadequate in the face of the industry's rapid expansion and the growing complexity of supply chains [3].

Amidst these challenges, the emergence of IoT and ML technologies presents valuable opportunities for progress. At the forefront of the ongoing digital transformation, these technologies are revolutionizing various sectors, including the retail industry [4–6].

✉ Ali Abdullah S. AlQahtani
AlQahtani.aasa@gmail.com

Aadel A. Darrat
Aadel.Darrat@lsus.edu

Lonnie Turpin
lturpin@mcneese.edu

Thamraa Alshayeb
Alshayeb.t@gmail.com

[1] Department of Software Engineering (Cybersecurity Track), College of Computer & Information Science, Prince Sultan University, Riyadh 12435, Saudi Arabia

[2] Department of Management & Marketing, College of Business, Louisiana State University Shreveport, Shreveport 71115, LA, USA

[3] Department of Management, Marketing & Business Administration, College of Business, McNeese State University, Lake Charles 70605, LA, USA

[4] Department of Physics and Astronomy, College of Science, George Mason University, Fairfax 22030, VA, USA

IoT with its ability to connect devices and systems [7], offers immense potential for automating and optimizing retail operations [8]. Simultaneously, ML, a subset of artificial intelligence (AI), has the capability to identify patterns and make predictions based on datasets, making it an ideal tool for inventory management [9].

McKinsey Global Institute reports that IoT technology will continue to improve and, by 2025, will have a direct economic impact of $2.7 trillion to $6.2 trillion per year across ten major sectors [10].

The traditional retail setting often struggles with labor-intensive and time-consuming shelf replenishment processes, leading to frequent out-of-stock scenarios and customer dissatisfaction [11]. This research addresses these challenges by leveraging IoT and ML to streamline the replenishment process. The proposed system employs smart scales, sensors, and ML algorithms to monitor product availability on shelves and provide real-time predictions. It also sends notifications to retail stock clerks for timely restocking. This innovative approach enhances efficiency and significantly reduces the potential for human error.

The broader objective of this study is to devise a system that supports often overlooked roles, such as those of retail stock clerks. Such efforts to enhance retail worker conditions with more efficient processes may improve performance for the business as a whole by increasing its sales [12].

The reserach questions we answer in this paper are:

1. How can IoT and ML technologies be leveraged to improve shelf replenishment processes in retail environments?
2. To what extent can the proposed smart shelving system accurately differentiate products requiring restocking, particularly when product features are nearly identical?
3. What is the optimal approach for determining restocking thresholds using mathematical modeling and optimization techniques?
4. What are the key features required in an ideal shelf replenishment system, and how does the proposed system compare to existing solutions in terms of autonomy, continuity, and adaptability?
5. What are the broader implications of adopting the proposed smart shelving system for retailers, retail stock clerks, and customers?

The proposed system's main contributions are summarized as follows:

- Introducing a novel system that utilizes IoT and ML, this approach enhances retail operations by semi-automating the shelf replenishment process. This not only reduces human error but also significantly improves operational efficiency, ultimately leading to increased customer satisfaction.
- Presenting a feature that tracks the expiration dates of products, ensuring that expired products are promptly identified and removed from the shelves.
- The restocking threshold is formulated as a mathematical optimization problem, incorporating an M/G/1 queueing model. This model accounts for time-dependent demand and multi-item settings, enhancing the system's adaptability to a range of retail scenarios.
- Offering an optimized model with exceptional accuracy in identifying products that need restocking (success rate of 99.35%), even when handling products with near-identical features.
- Generating a synthetic dataset that closely mirrors real-world retail conditions to overcome the lack of real-world data, published on *IEEE Dataport* [13], serving as a valuable resource for future research.

The manuscript is structured as follows: Sect. 2 surveys various technologies and methods, including RFID and IoT, for addressing product availability and replenishment. Section 3 presents the proposed system in detail, outlining its IoT network model, workflow, key features, and comparisons with related works. Section 4 describes the generation of the synthetic dataset and applied optimization techniques. This section also offers an in-depth analysis of the system's performance metrics, response time, and feature importance. Section 5 offers a quantitative comparison of the proposed system with related works. It discusses its benefits for retailers, stockers, and customers, and outlines the limitations and potential avenues for future research. Finally, Sect. 6 concludes the paper.

2 Literature review

Retailers have long-implemented tag reading systems to help track the movement and location of products within stores and to automatically update inventory in real-time. RFID technology, a worthy successor of the barcode scanning system, is commonly suggested to facilitate retail shelf replenishment, as demonstrated in various studies [14–18]. In [19], researchers address the challenges faced by supermarkets and large retail stores in the handling and placement of products on shelves by designing a system that utilizes barcode recognition and location detection techniques that accommodate for products of different sizes, shapes, and weights. Another paper [20] offers an interesting comparative study on replenishment in a supply chain system that considers misplacement discrepancies. The model is demonstrated through two examples, one without the use of

RFID and blockchain, where the manager incurs losses due to inventory discrepancies, and one with the use of RFID and blockchain for product visibility. The authors conclude that not using RFID technology is only successful in situations where demand is high enough to offset the impact of product misplacement. Importantly, the integration of RFID and IoT technology has indubious potential to revolutionize retail operations. See [21] and [22] for a useful review of retail management applications and challenges faced.

Other research explores novel RFID-based applications beyond stock management [23]. By leveraging RFID technology and analyzing stocktake data, retailers can gain insights into the spatial distribution of goods within their stores. The accuracy and efficacy of these approaches opens doors to valuable applications, empowering retailers to optimize operations and make data-informed decisions. One investigation into complex event processing within a RFID-enabled retail store operation offers a promising avenue for addressing the prevalent problem of out-of-stock occurrences [24]. By defining and utilizing event queries to analyze RFID data streams, retailers can detect and respond to movement at different levels, enabling efficient and timely shelf replenishment. The outcomes of this research can inform and inspire the adoption of automated inventory management systems in the retail industry, fostering enhanced customer experiences and streamlined operations.

Electronic shelf labels (ESL) are digital displays that provide real-time information about the products on store shelves, including the price, promotions, availability, and other relevant details. The use of ESLs in retail environments has been the subject of research and discussion in several studies [25–28]. In a case study of a grocery retailer, [29] report that the installation of ESLs has a sizeable effect on gross margins. [30] show that inventory record inaccuracy (IRI), or the lack of accurate and complete information about a retail store's inventory, is a more significant factor in explaining price rigidity in perishable grocery products than physical menu costs (such as labor and material costs associated with price adjustment). In [31], researchers present a comprehensive examination of the ESL system, focusing on design and real-world deployment experiences. Their work highlights the technical challenges of achieving low-power wireless communication in scatter-rich indoor environments with high node density. To overcome these challenges, they proposed multi-radio and multi-channel diversity techniques that significantly improve system performance with minimal modifications to gateways. The evaluation, based on a convenience store deployment of 550 tags, demonstrates a network connectivity rate of 98.51%, 590 ms average latency for price updates, and over five years of e-tag lifetime. Other research [32] discusses an ESL system that provides accurate price information in stores. Manual price tag updates are time-consuming and error-prone, leading to high labor costs. The ESL systems offer dynamic price updating and product evaluation display features. To improve wireless communication success, the paper proposes an ESL image compression mechanism based on chain coding, reducing data transmission and improving performance. Results show smaller compressed images and faster decompression. Overall, the proposed mechanism enhances ESL system efficiency in dense indoor environments, benefiting store management.

ML techniques have been widely used to facilitate shelf replenishment in various systems [33–35]. These algorithms can analyze historical sales data to forecast future demand for products on the shelf, enabling retailers to proactively restock items that are likely to run out. They can also be used to classify or predict which products are most likely to be purchased together, allowing retailers to optimize shelf layout and placement to increase sales. [36] use ML algorithms to develop a new method for monitoring on-shelf availability (OSA) in grocery stores using computer vision techniques. The method offers a solution to improve accuracy and efficiency in monitoring OSA, while alleviating the traditional manual-labeling which can be time-consuming. Moreover, ML models provide an effective methodology to infer demand [37] and to improve customer shopping experience [38]. For example, [39] introduce a smart replenishment system that relies on point-of-consumption (POC) information for product delivery decisions. The system uses the classic Vendor-Managed Inventory (VMI) concept and shows that replenishment decisions can be significantly improved with POC information. Although a VMI approach provides the major advantage of cost reduction [40], one drawback is that end user VMI does not aid stockers in their shelf replenishment duties within the retail store. A primary motivating factor for the work is to address this deficiency.

Research in [41] addresses the challenges associated with weakly supervised instance segmentation in the retail industry. By leveraging the region normalization mechanism and an adaptive region division strategy, they effectively tackle the limitations imposed by the specific characteristics of this industry. This approach outperforms existing methods on the MVTec D2S benchmark dataset, showcasing its practical value in real-world retail applications. Other work presents an innovative electricity retail pricing strategy that incorporates the optimal operation of an ESS using a ML algorithm [42]. Specifically, an artificial neural network (ANN) is leveraged to construct a practical model for DR scheduling of an ESS. By training this model on historical data encompassing electricity prices and the corresponding optimal demand derived from the building energy management system, the proposed approach achieves both accuracy and effectiveness. Furthermore, the derived ANN-based DR model is seamlessly integrated into the constraints of the retail pricing optimization problem within the distribution management system. By doing so, it provides a single-level structure that encompasses the decision-making process of both the DISCO and the building operator. The results of the retail pricing analysis reveal that the proposed strategy accurately determines balancing points while simultaneously reducing peak load, showcasing its significant potential for real-world deployment.

The retail sector has witnessed extensive research aimed at maintaining on-shelf availability. Approaches range from manual inventory counts to technologically advanced solutions, such as barcode scanning, RFID technology, ESLs, and ML. Despite these advancements, we believe that there is a compelling need for the scientific community to propel the field further. The existing methodologies appear to share common drawbacks. Perhaps the most prominent flaw is the necessity for human intervention that inherently introduces the potential for error. The proposed research aims to address.

3 The proposed system

This section provides a detailed exposition of the proposed system, focusing on the network model and operational workflow, and optimization of the restocking threshold. Moreover, this section introduces the features of the proposed system and compares them with related works previously discussed.

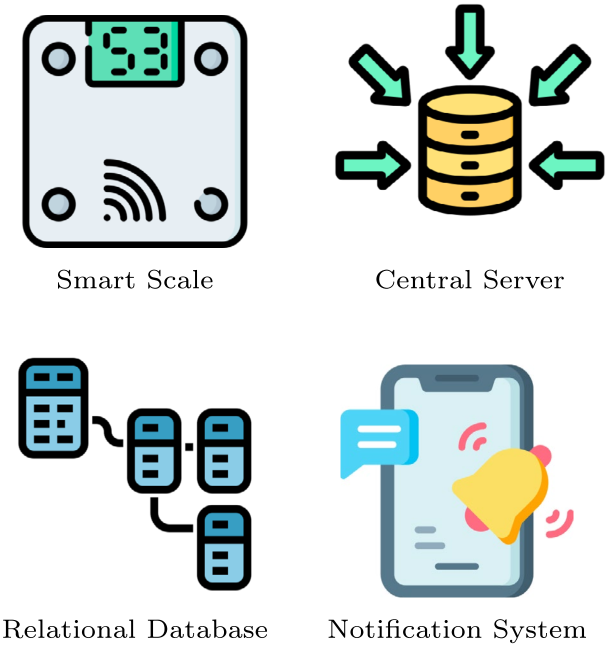

Fig. 1 Network model; Icons From [43]

3.1 Network model

The network model capitalizes on the transformative potential of IoT in retail, integrating four key components: smart scales, a central server, a relational database, and a notification system; see Fig. 1.

Smart scales, integral to the network, are equipped with sensors and are strategically positioned on shelves or under individual items. Connected via Wi-Fi, these nodes monitor product weight, translating weight changes into real-time inventory updates, a key feature of IoT applications.

The central server, functioning as the control hub of the network, receives data from the smart scales. It stores this information in a relational database and processes it to identify when restocking is required. This automation of complex processes showcases the power of IoT in creating efficient network models.

The relational database, serving as a crucial node in the IoT network, stores essential data on product weight, expiration dates, and inventory levels. Continually updated via the central server, it ensures the network operates with the most current information, a key aspect of real-time IoT systems.

Our relational database is organized into four principal tables-Products, ScaleReadings, RestockEvents, and InventoryThresholds-to capture all facets of shelf monitoring and replenishment. The Products table (ProductID PK, Name, UnitWeight, ExpirationDate, AisleID, ShelfID) records each item's identity, weight parameters, and storage location. ScaleReadings (ReadingID PK, ProductID FK, Timestamp, MeasuredWeight) logs real-time weight measurements, while RestockEvents (EventID PK, ProductID FK, Timestamp, QuantityAdded) tracks every restocking action. InventoryThresholds (ProductID PK/FK, ThresholdWeight) defines the weight-based trigger for notifications. A separate Locations table (AisleID PK, ShelfID PK, Description) normalizes physical layout data, with Products. AisleID/ShelfID referencing Locations.AisleID/ShelfID. Each foreign key relationship enforces referential integrity and supports efficient joins to compute current inventory levels and expiration alerts. This schema minimizes redundancy, facilitates rapid queries for both real-time weight analytics and historical restock statistics, and underpins the system's ability to deliver accurate, timely notifications.

The relational database, serving as a crucial node in the IoT. It sends comprehensive alerts, including product name, shelf location, and nearest expiration date, to specific individuals or devices. This immediate attention to restocking needs underlines the efficiency of IoT in retail operations.

3.2 Workflow

The proposed system operates by configuring key product parameters, including weight, expiration date (updated with each restock), in-store location, and inventory threshold. The threshold, defined as three times the weight of a single product, can be adjusted to meet retailer specifications. The system persistently monitors in-store product levels, triggering a restocking alert when inventory falls below the set threshold. The notification system then conveys comprehensive product details, such as name, location, and nearest expiration date, to the stocker.

A novel feature of the proposed system is its ability to track product expiration dates. This capability utilizes data from the last restocking event. Each product's expiration date is stored in a set E, with individual expiration dates represented by the variable e. The system continuously checks the current date against each e in the set E. If the current date surpasses any e, an expiration notification is sent, and the expired item is removed from the inventory.

The system's operation can be encapsulated in a mathematical function. The current date is denoted as $d_{current}$ and the set of expiration dates as $E = e_1, e_2, ..., e_n$, where each e_i corresponds to the expiration date of the i-th product on the shelf. The system's operation can be represented by Eq. (1):

$$f(d_{current}, E) = e_i \in E | d_{current} > e_i \tag{1}$$

This function takes the current date and the set of expiration dates as inputs, returning the set of expiration dates surpassed by the current date. For each such date, an expiration notification is dispatched, and the corresponding item is removed from the inventory.

The data transmitted from the smart scales to the server and subsequently stored in the relational database, is used to determine whether or not to send a notification to the stocker. The core process of the system, illustrated in Algorithm 1, computes the total quantity of products on a shelf by dividing the overall weight by the weight of a single item. With the threshold set at three times the weight of one product, the algorithm runs in a perpetual checking loop. That is, a restocking alert is sent if the product quantity falls below the threshold; otherwise, the checking process continues, ensuring real-time monitoring and timely restocking notifications.

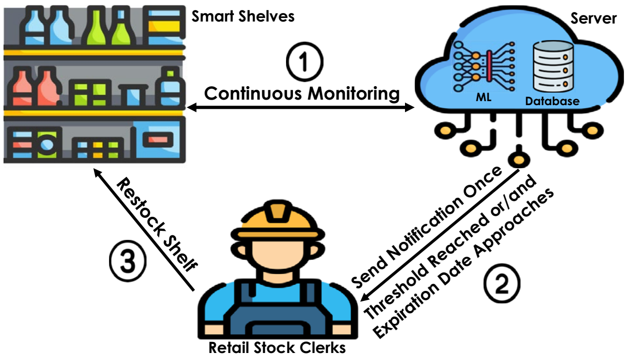

Fig. 2 System overview; Icons From [43]

To illustrate this, consider a retail shelf with a variety of products, each with different expiration dates. The system continuously monitors the weight of the products on the shelf, and if the total weight of the products falls below the threshold, a restocking alert is sent. Simultaneously, the system checks the current date against the expiration dates of the products. If any product's expiration date has passed, an expiration notification is sent, and the expired product is removed from the inventory.

This dual monitoring of inventory levels and product expiration dates ensures that the stocker is always informed about the current state of the shelf, facilitating efficient restocking and minimizing the presence of expired products on the shelf. Figure 2 depicts an overview of the proposed system.

```
Total Weight of Products on shelf → $W$
Number of Products on shelf → $N$
Single Product Weight → $P$
Threshold → $T$
Current Date → $D$
Expiration Dates of Products on shelf → $E$
Individual Expiration Date from $E$ → $e$

$N = W \div P$
$T = 3 \times P$
while (true) do
    if $N < T$ then
        Send Restock Notification;
    for each $e$ in $E$ do
        if $D > e$ then
            Send Expiration Notification;
            Remove Expired Item from Inventory;
```

Algorithm 1 The proposed system's mechanism with expiration date tracking

To guarantee seamless interaction among smart scales, central server, and relational database, the Smart Shelves system employs standard IoT communication protocols (e.g., MQTT for lightweight telemetry and HTTP/REST for control messages), with all sensor readings published as JSON payloads against a rigorously versioned schema. Each incoming message is first validated against this schema-checking field presence, data types, value ranges (for example, weight $\in [0\text{ g}, 50\text{ kg}]$), and timestamp ordering-to prevent malformed or stale records. At the server, transactions are managed via ACID-compliant database operations, ensuring consistency under concurrent updates. Anomaly detection routines flag outliers (e.g., sudden $\pm 10\text{ kg}$ jumps) and route suspect data through a cleansing pipeline that can interpolate or discard corrupt entries. Periodic sensor calibration routines (e.g., zero-offset checks every 24 h) and automated health-check messages maintain sensor accuracy and detect drift.

3.3 Optimization of restocking threshold

In the proposed system, the restocking threshold is a pivotal parameter. It is crucial to set this threshold judiciously as a high value may trigger unnecessary restocking alerts, while a low value may lead to stockouts before restocking can occur. To address this, the optimization of the restocking threshold is formulated as a mathematical problem

An important aspect of this optimization involves correlating the weight-based threshold with the actual count of inventory items. For each product type, the system defines a standard unit weight, W_u, which is used to convert weight measurements into estimated counts. The relationship between the total weight on the shelf, W_t, and the estimated inventory count, N, is given Eq. (2):

$$N = \frac{W_t}{W_u} \tag{2}$$

This equation allows the system to dynamically translate the weight data from smart scales into an estimated count of products, ensuring that the restocking threshold in terms of weight accurately reflects the actual inventory levels in terms of count. Additionally, to accommodate variations in product weight or packaging changes, the system includes a calibration mechanism, which periodically updates W_u based on real-time measurements and historical data. This adaptive approach ensures the precision of the weight-to-count conversion over time, maintaining the accuracy of inventory estimates and the efficacy of the restocking process.

The objective is to minimize the total cost, which is the sum of the restocking cost (C_r) and the stockout cost (C_s). The stockout cost is dependent on the probability of a stockout (p_s), which in turn depends on the restocking threshold (T). The optimization problem can be expressed by Eq. (3):

$$\min_{T}\ C_r + p_s(T)C_s \tag{3}$$

To accurately capture the dynamics of customer arrivals and service times in the retail environment, an M/G/1 queueing model is incorporated. This model assumes that customer arrivals follow a Poisson process with rate λ, and service times follow a general distribution with mean $1/\mu$.

Given the variable nature of demand in retail settings, the model has been extended to accommodate time-dependent demand. The restocking threshold is dynamically adjusted based on the time of day. Let the restocking threshold at time t be denoted as $T(t)$, and the arrival and service rates at time t be represented as $\lambda(t)$ and $\mu(t)$ respectively. The optimization problem is then expressed by Eq. (4).

$$\min_{T(t)} \int (C_r + p_s(T(t), \lambda(t), \mu(t))C_s)\, dt \tag{4}$$

In real-world retail environments, replenishment operations typically involve multiple products. Consequently, the model has been expanded to a multi-item setting. Let the restocking threshold for item i be denoted as T_i, and the restocking and stockout costs for item i be represented as C_{r_i} and C_{s_i} respectively. Under these definitions, the optimization problem is expanded to Eq. (5):

$$\min_{T_i} \sum (C_{r_i} + p_s(T_i, \lambda_i, \mu_i)C_{s_i}) \tag{5}$$

Equation (6) then represents the gradient of the cost function:

$$\nabla_{T_i}(C_{r_i} + p_s(T_i, \lambda_i, \mu_i)C_{s_i}) = C_{s_i}\nabla_{T_i} p_s(T_i, \lambda_i, \mu_i) \tag{6}$$

By solving these optimization problems, the optimal restocking thresholds that minimize the total cost can be determined. This leads to more efficient and effective restocking strategies, enhancing the performance of the proposed system in various retail scenarios.

3.4 The proposed system features

This section outlines the features of the system and their potential benefits for retailers, stockers, and customers, which are discussed in detail in Sect. 5.

3.4.1 Noninteractive

In the proposed system, the feature of being $'Noninteractive'$ stands as a testament to its autonomous functionality. This characteristic is deeply ingrained in the system's architecture, enabling it to operate independently, free from the constraints of human supervision.

At the heart of this autonomous operation are the smart scales, equipped with sensor technology and IoT capabilities. These scales, strategically placed on the shelves or beneath individual items, serve a crucial role. They continuously monitor the weight of the products, generating real-time data about the inventory changes.

This data, once generated, is transmitted via IoT protocols to a central server. The server processes the incoming data. It discerns when the quantity of items on the shelves dips below a predetermined threshold, indicating the need for a restocking alert.

Upon identifying a need for restocking, the system autonomously triggers the notification system. This notification system generates a comprehensive alert, detailing the product name, its location on the shelf, and other pertinent information.

This entire process, from the initial monitoring of inventory levels to the final triggering of restocking alerts, is carried out autonomously, encapsulating the essence of the $'Noninteractive'$ feature. The IoT integration enhances the system's ability to function independently, making it a truly smart and interconnected system.

3.4.2 Continuous

The $'Continuous'$ feature, a cornerstone of the proposed system, enables real-time monitoring of product availability. This is achieved through the integration of IoT-enabled smart scales into store shelves. These scales, embedded with sensor technology, continuously monitor the weight of products, providing a live feed of inventory status.

This live data stream, facilitated by IoT connectivity, is directed towards a central server, the system's control hub. The server processes the incoming data, maintaining an up-to-the-minute understanding of product availability across the store. The real-time nature of this data allows the system to stay in a constant state of readiness, prepared to identify and respond to inventory changes instantaneously. In the event of a potential product shortage, indicated by a product's weight falling below a predetermined threshold, the server activates the notification system. This system generates an alert, signaling the need for restocking.

Table 1 Comparison with the discussed works, where ✓: feature supported, and ✗: not supported

Study	Noninteractive	Continuous	Adjustable
[14–24]	✗	✗	✗
[25–32]	✗	✗	✗
[33–42]	✓	✗	✗
Smart shelves	✓	✓	✓

3.4.3 Adjustable

The dynamic nature of the proposed system is demonstrated through its $'Adjustable'$ feature. This feature allows the system to adapt to varying retail scenarios by flexibly setting and modifying restocking thresholds. These thresholds are not static values; they can be adjusted based on a variety of factors, making the system responsive to changes in the retail environment.

One of the key factors influencing the adjustment of restocking thresholds is store traffic. By integrating with store traffic data, the system can anticipate periods of high demand and adjust the restocking thresholds accordingly. This ensures that the system remains responsive to changes in customer traffic and can adapt its restocking strategies to meet changing demand.

Another influential factor is historical sales data. The system can analyze patterns and trends in past sales to identify future demand for different products. This allows the system to adjust restocking thresholds based on anticipated demand, ensuring that it remains proactive in its restocking strategies.

Incorporating these factors into the optimization problem allows for the expression of the restocking threshold as outlined in Eq. (4), as described previously in Sect. 3.3. Solving this equation offers an optimal restocking threshold that minimizes the total cost. This dynamic adjustment of restocking thresholds allows the proposed system to respond effectively to real-time changes in the retail environment, thereby ensuring optimal inventory management.

3.5 Comparison with the discussed works

The characteristics of an ideal system, as inferred from the study, should comprise three key attributes: non-interactive, continuous, and adjustable. $'Noninteractive'$ signifies the ability of the system to operate autonomously, eliminating the need for constant human monitoring. The $'Continuous'$ aspect refers to the system's capability to perform real-time monitoring of product availability, ensuring a constant state of readiness to respond. Lastly, being $'Adjustable'$ embodies the system's ability to flexibly set and modify the restocking thresholds based on varying factors, such as store traffic or historical sales data. This dynamic nature ensures a more efficient restocking process, mitigating the risk of inventory shortages or overstocks.

The comparative analysis, summarized in Table 1, delineates how existing solutions compare against these criteria. The results underscore the distinct aspects of the proposed method, which encompasses all three critical features.

4 Experiment and results

This section describes the dataset and experimental setup utilized to evaluate the proposed system.

4.1 Data acquisition

As obtaining a real-world dataset presented significant challenges, a synthetic dataset was generated using Python that closely mirrors real-life conditions. This dataset is published on *IEEE Dataport* [13]. This approach enabled the construction of a balanced representation of diverse scenarios, providing a strong basis for thorough testing and evaluation of the ML models.

A total of 39,500 unique product identities were assumed, given this is the average number of products carried in a retail store [44], with a desired total of 70,000 samples distributed across these products. The number of samples corresponding to each product was determined through the chi-square distribution, ensuring at least one sample for each product. This approach led to the actual sample total being close to the desired number.

Each unique product was then assigned attributes, including a weight (assumed to be normally distributed around 10 with a standard deviation of 2), an aisle number, a shelf number, and a restocking threshold (modeled as three times the single product weight plus some noise to account for real-world variability). Product location was then defined using a dictionary that mapped each product ID to its respective aisle and shelf number.

Next, the time since the last restocking was generated. The code creates random restock dates for each product within a specified time range from January 1, 2022, to June 7, 2023 (adjustable). This information is then used to calculate the number of days since restocking. The product data, alongside these calculated metrics, were incorporated into a DataFrame for further processing.

With this dataset, the code calculates the total weight on the shelf for each product and identifies whether restocking is necessary based on whether the total weight on the shelf drops below a product-specific threshold.

4.2 Preprocessing

The preprocessing stage involved feature engineering and data balancing, which were critical in preparing the data for the ML models and improving their performance.

4.2.1 Feature engineering

Feature engineering was performed to create new features from the existing ones. This process involved various transformations and calculations, which helped to improve the model's performance.

The first feature created was a binary indicator that signifies whether the weight of a product on the shelf is below a certain threshold. This feature directly relates to the target variable of whether a product needs to be restocked, making it a valuable addition to the dataset.

Another feature created represents the time elapsed since the last restock. This feature was calculated by recording the time at which each restock occurred and then calculating the difference in time between each restock. This new feature provides information about the frequency of restocking, which could be a valuable predictor for when restocking is needed.

These new features were then included in the data that was fed into the ML models, along with the original features. The inclusion of these engineered features helped to improve the models' classification performance, as they provided additional, valuable information that the models could learn from.

4.2.2 Data balancing

The raw dataset was found to be imbalanced concerning the $'need\_restock'$ label. To address this, the dataset was balanced by undersampling the majority class to match the minority class count using sklearn resample utility. This resulted in a resampled majority class which was then combined with the minority class to form a balanced dataset. Both classes contained 35,040 samples, bringing the total to 70,080 samples. The balanced dataset was then shuffled to ensure randomness, thus making it ready for evaluation.

4.3 Optimization

The proposed system uses four optimized different ML models, i.e., Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), and Support Vector Machine (SVM). The use of multiple ML models in the proposed system is a strategic decision aimed at harnessing the unique strengths of each model to achieve optimal performance. The pursuit of robust and efficient classification systems necessitates the optimization of ML models. The selection of an appropriate optimization algorithm is pivotal, as it can significantly influence the model's performance. It is essential to tailor the optimization strategy to the specificities of the data and the problem at hand.

4.3.1 LR model

The research has achieved significant strides in optimizing Logistic Regression (LR) models by employing Stochastic Gradient Descent (SGD). SGD is an iterative method that refines the model parameters to minimize a suitable objective function, thereby enhancing the model's predictive accuracy. Its computational efficiency and scalability make it particularly effective for large datasets, a common scenario in IoT applications.

The objective function, or cost function, in LR is given by Eq. (7):

$$J(\theta) = -\frac{1}{m}\sum[y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))] \tag{7}$$

In Eq. (7):

- m represents the number of training examples,
- $y^{(i)}$ and $x^{(i)}$ denote the actual label and the *i*-th training example, respectively.
- The predicted output for the *i*-th training example, $h_\theta(x^{(i)})$, is computed using the logistic function $1/(1+e^{-\theta^T x})$.

The core of SGD lies in its update rule, which adjusts the model parameters θ, Eq. (8), iteratively for each training example:

$$\theta_j := \theta_j - \alpha(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} \tag{8}$$

In Eq. (8):

- α is the learning rate,
- θ_j is the *j*-th parameter,
- $x_j^{(i)}$ is the *j*-th feature of the *i*-th training example. This rule is derived from the gradient of the cost function with respect to θ_j.

The study contributes by effectively using SGD to optimize LR models. By performing updates more frequently, SGD allows for faster and more efficient optimization, leading to improved model performance.

4.3.2 SVMmodel

The approach utilizes the power of Adam (Adaptive Moment Estimation), a potent tool in the realm of machine learning optimization algorithms. Unlike the classical stochastic gradient descent method, which uses a single learning rate for all weight updates, Adam maintains individual learning rates for each weight in the model. This unique feature allows Adam to adaptively adjust these rates as the optimization progresses, leading to more efficient and effective model training.

The core of the Adam optimizer lies in its mathematical update rules. These rules, as detailed below, govern how the optimizer adjusts the model's weights at each step of the training process:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)g_t \tag{9}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)g_t^2 \tag{10}$$

$$\hat{m_t} = \frac{m_t}{1-\beta_1^t} \tag{11}$$

$$\hat{v_t} = \frac{v_t}{1-\beta_2^t} \tag{12}$$

$$\theta = \theta - \frac{\alpha\hat{m_t}}{\sqrt{\hat{v_t}}+\epsilon} \tag{13}$$

In Eqs. (9), (10), (11), (12), (13):

- m_t and v_t are estimates of the first moment (the mean) and the second moment (the uncentered variance) of the gradients, respectively.
- β_1 and β_2 are the forgetting factors for these moment estimates. Typically, they are initialized as $\beta_1 = 0.9$ and $\beta_2 = 0.999$.
- g_t represents the gradient at time step *t*.
- α is the learning rate.
- ϵ is a small constant, usually on the order of $1e-8$, introduced to prevent division by zero.

Table 2 Evaluation results

Model	Accuracy	Precision	Recall	F1 Score
LR	0.9930	0.9930	0.9930	0.9930
DT	0.9928	0.9928	0.9928	0.9928
RF	0.9886	0.9886	0.9886	0.9886
SVM	0.9935	0.9935	0.9935	0.9935

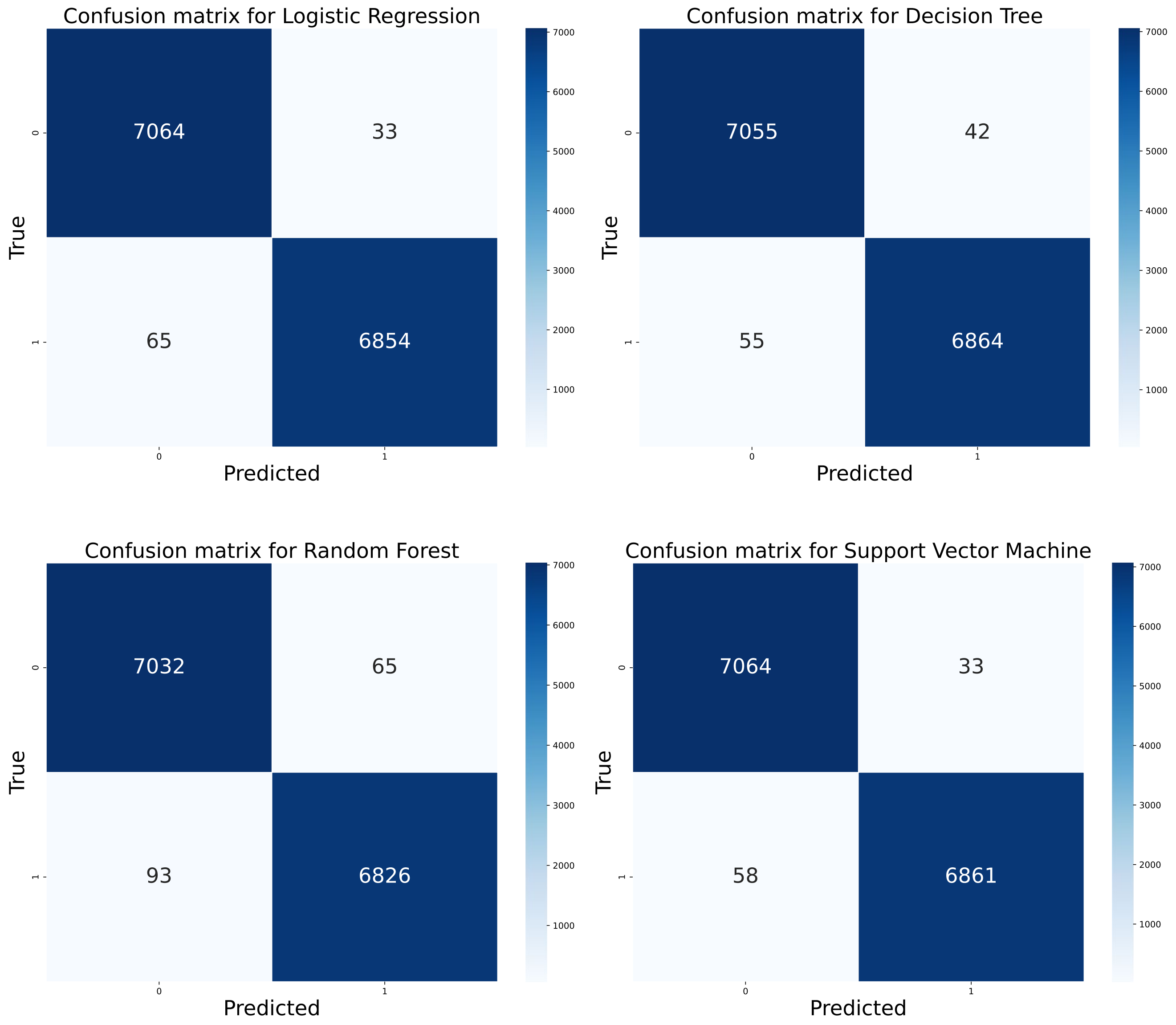

Fig. 3 Confusion matrices for the used classifiers

By leveraging Adam's adaptive learning rates, the performance of the SVM model is enhanced, resulting in increased robustness and accuracy.

4.3.3 RF and DT models

The intricacies of DT and RF models are explored, with a particular focus on their splitting criteria. The quality of a split, a fundamental aspect of these models, can be quantified using measures such as Gini impurity.

The Gini impurity for a node, a measure of misclassification where a lower value indicates a better split, is calculated as:

$$Gini(p) = 1 - \sum (p_i)^2 \tag{14}$$

In Eq. (14):

- p_i represents the probability of an item with label i being chosen.

4.4 Evaluation

After optimization, the four models were evaluated using four key metrics: accuracy, precision, recall, and F1 score. A train-test ratio of 80:20 was set, with 80% of the data allocated for training and the remaining 20% for testing. The resulting performance evaluations are presented in Table 2, and the respective confusion matrices for each classifier are depicted in Fig. 3.

As shown in Table 2, SVM model emerges as the best performing model with the highest scores across all four

Table 3 Summary time duration (seconds)

Min	Max	Ave	Standard deviation
5.99	16.18	8.66	1.95

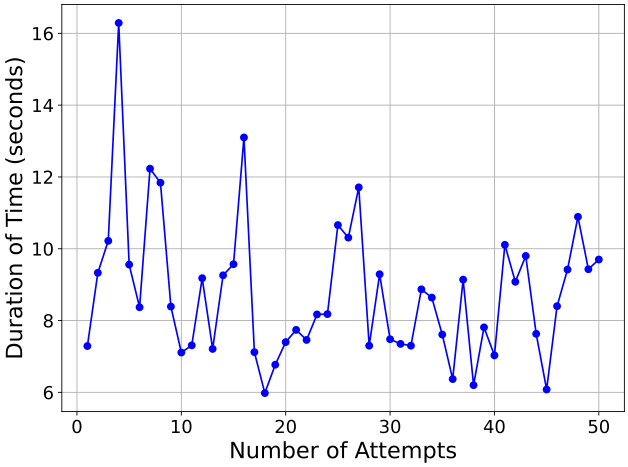

Fig. 4 Total time of sending a notification (seconds)

metrics (0.9935). Meanwhile, RF model shows the lowest scores (0.9886) across all indicators. The LR and DT models display almost equivalent performances, with their scores at 0.9930 and 0.9928, respectively. As can be noticed from Table 2, the differences among the models' performances are relatively small, indicating that all models tested have a high degree of classification performance.

4.5 Time consumption

In this experiment, the aim was to assess the real-time performance of the proposed inventory management system. The key metric was the time taken from the initiation of a restocking notification to its receipt by the stocker, an essential measure of the system's responsiveness to inventory changes. The experimental setup is designed to mimic a real-world retail environment. A Wireless Access Point (WAP) was utilized to ensure seamless connectivity among Wi-Fi-enabled devices within the test environment. Two Raspberry Pi devices simulated an IoT system in a retail store: one acted as a store shelf scale, and the other as a stocker's mobile device. Data processing was conducted by a server equipped with an Intel Xeon processor, 16 GB of RAM, and running Ubuntu 16.04 LTS (64-bit). The server was set up with Apache Server and phpMyAdmin, enabling the creation and management of a database for storing data from the Wi-Fi-enabled devices.

The experiment is comprised of 50 trials. In each trial, an object was placed on the scale, simulating a product reaching its restocking threshold. This action triggered the system to measure the time from the notification dispatch to its display on the stocker's Raspberry Pi LCD. Upon completion of the trials, the data were analyzed to determine the minimum, maximum, and average times. As shown in Table 3, the proposed system, on average, took 8.66 seconds to send a notification to the stocker, with a standard deviation of 1.95 seconds. The fastest recorded time was 5.99 seconds, and the longest was 16.18 seconds. Figure 4 illustrates the entire trial record.

Several factors contributed to the observed response times. First, network latency plays a significant role; any delay in data transmission between devices and the server directly affects how quickly notifications are dispatched and received. Additionally, the processing load on the server, particularly when handling multiple simultaneous requests, can introduce variability in response times. The efficiency of the communication protocol between the devices can be designed to ensure reliable and swift data exchange, which also can impact overall performance.

In a real-time retail environment, the system should be designed to prioritize notifications based on urgency. For example, products with higher sales velocity or those approaching stock depletion thresholds are given precedence. The system also can employ a dynamic queuing mechanism that evaluates the criticality of each notification, ensuring that the most pressing restocking tasks are addressed first. This approach not only optimizes response time but also enhances the overall efficiency of inventory management by preventing stockouts of high-demand items and maintaining smooth operational flow.

4.6 Feature importance analysis

Assessing feature importance can significantly optimize the system's performance and contribute to the store's operational efficiency and profitability. In this experiment, the importance of each feature for the classification task was calculated for each model. This analysis facilitates an understanding of the most influential features in determining the decision for restocking needs. The $feature\_importances\_$ attribute in Python was employed to obtain the importance

Table 4 Feature importance analysis results

Model	*product_id*	*single_product_weight*	*aisle*	*shelf*	*last_restock_unix*	*time_since_restock*	*threshold*	*total_weight*
LR	0.000294	0.200558	0.000122	0.000288	0.000039	0.000039	0.312083	0.486578
DT	0.000891	0.007125	0.000577	0.000059	0.000262	0.000575	0.157037	0.833474
RF	0.017376	0.043237	0.007843	0.005193	0.013935	0.013708	0.078029	0.820679
SVM	0.000515	0.192878	0.000258	0.000394	0.000080	0.000080	0.307013	0.498782

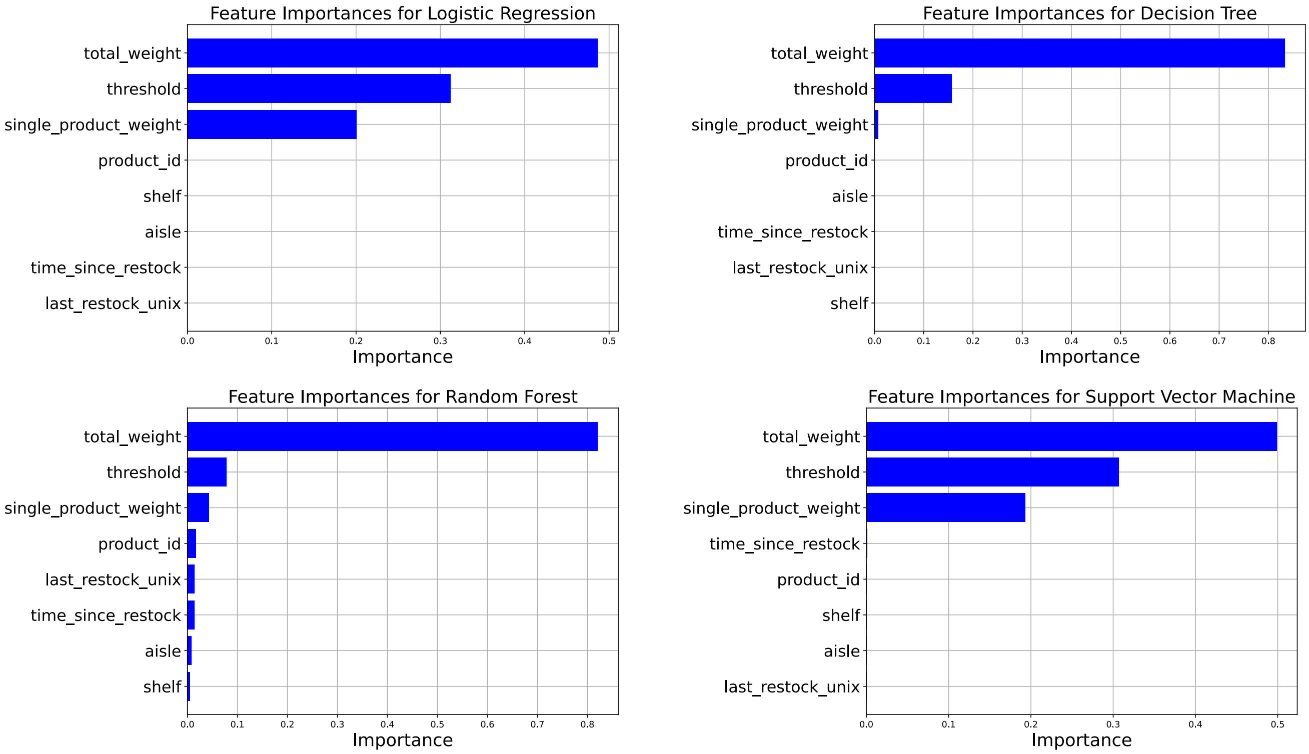

Fig. 5 Feature importance for the used classifiers

of each feature for DT and RF models. For SVM and LR, feature importance was ascertained by using the absolute value of the coefficients, accessed via the $coef\_$ attribute. The results are displayed in Table 4 and plotted in Fig. 5.

The $total\_weight$ feature stands out as the most important across all models. This means that the total weight of products on the shelf is the most influential factor in deciding when restocking is needed. For retail stores, this insight is valuable since understanding the importance of each feature in the classification task model is crucial prior to system deployment. It allows managers to focus on the key factors that influence restocking needs. This knowledge enhances the classification accuracy of the system, ensuring reliable restocking classifications. Furthermore, it provides strategic insights that can inform inventory planning and sales strategies, and aids in the effective scaling of the system as the store expands or diversifies its product range.

5 Discussion

This section presents a quantitative comparison of the proposed system with previously discussed methodologies, along with a detailed description of its potential benefits for retailers, stockers, and customers.

Table 5 Comparison with the discussed works

Study	Accuracy	Time consumption	Number of feature
[14–24]	N/M	N/M	N/A
[25–32]	N/M	N/M	N/A
[33–42]	89.264–89.6%	N/M	N/M
Smart Shelves	99.33%	8.66s (Ave)	7

Where N/M: Not Mentioned, and N/A: Not Applicable

5.1 Quantitative comparison

The retail technology landscape has seen a myriad of methodologies designed to enhance inventory management and user experience. The quantitative comparative analysis with The discussed works aims to highlight the innovations introduced by the ’Smart Shelves’ approach in this field.

5.1.1 RFID-based approaches

The Studies [14–24] utilize the tried-and-true RFID technology. While the exact accuracy and time consumption metrics remain unspecified in these studies, a notable gap is the non-applicability of decision-driving features, emphasizing a potential limitation in granularity and adaptability of their systems.

5.1.2 Electronic shelf labels

The works represented by [25–32] employ ESL. As with the RFID-based studies, these papers do not specify accuracy or time consumption metrics. Moreover, their non-applicability in feature-based decision-making further underscores a potential shortcoming in their methodologies.

5.1.3 Machine learning approaches

A shift towards computational intelligence is evident in the studies [33–42]. These papers, harnessing ML, reported accuracy ranging from 89.264% to 89.6%. However, specific time consumption metrics and the number of features utilized for decision-making were not mentioned, hinting at potential inefficiencies or lack of transparency.

5.1.4 Smart shelves

Contrasting the aforementioned works, The proposed approach, dubbed 'Smart Shelves,' manifests a compelling edge. The 'Smart Shelves' approach achieved an accuracy of 99.33%, substantially outperforming the ML-based approaches. The proposed system's average time consumption stands at 8.66 seconds, showcasing its efficiency. Furthermore, The proposed method employs seven distinct features for decision-making, indicating a comprehensive and nuanced approach to inventory management. Table 5 summarizes the results.

5.2 For the retailer

The proposed system is scalable and adaptable to different retail stores due to its use of IoT technology and ML. For instance, if a large store with many shelves and products requires more monitoring, the system can be scaled up by raising the restocking notification threshold. On the other hand, if a small store with fewer shelves and products requires less monitoring, the system can be scaled down by lowering the restocking notification threshold. This versatility makes the proposed system a practical solution for retailers of any size, as it can be customized to fit each store's specific needs.

Convenient service innovations have a positive impact on shareholder value [45]. The proposed replenishment system provides a feature for inventory management, which accurately tracks product levels on shelves and alerts the stocker when replenishment is necessary. This helps reduce waste and increase profits by avoiding stocking products close to expiration. The system also tracks the nearest expiration date of products on the shelves, making inventory management more efficient. This is especially true for grocery stores looking to prevent losses from perishable products.

Importantly, the proposed application is a cost-effective solution for retailers. The cost of implementing this application is relatively low compared to the benefits it provides. The smart scales are easy to install and integrate with existing systems, eliminating the need for major renovations or investments in new infrastructure. The use of IoT technology and ML also helps to minimize maintenance costs, as the system can be remotely monitored and updated as needed. In addition, the savings generated from improved inventory management, reduced waste, and increased sales can offset the initial cost of the system investment. By providing a comprehensive solution to shelf monitoring and inventory management, the proposed replenishment system represents a cost-effective solution for retailers of all sizes looking to improve their operations and benefit their bottom line in the process.

5.3 For the stocker

The proposed system has a key benefit of requiring no effort nor interaction from stockers. By automating the shelf-monitoring process, stockers are able to complete tasks more efficiently. In addition, freeing stockers from the time-consuming task of manually checking shelves allows them to focus on other job duties. Proactive monitoring also eliminates the possibility of human error and ensures accurate and timely shelf replenishment, which in turn maximizes on-shelf product availability. Furthermore, the elimination of manual monitoring reduces the stress and workload of stockers, making their jobs easier and less demanding. Taken together, adopting an automated shelf-replenishment system would lead to increased efficiency, productivity, and improved satisfaction of these important, yet commonly neglected, retail employees.

5.4 For the customer

Retail stores should employ simple, smart technologies that provide value by improving customer shopping efficiency [46]. The proposed system has the potential to enhance customer satisfaction and store loyalty by maximizing on-shelf availability. Empty shelves can lead to frustration and inconvenience for customer and can ultimately affect their perceived value of the store. On the other hand, a store that consistently has the products its customers want in stock is more likely to encourage customer loyalty and repeat business. The automated replenishment system proposed in this research fosters optimal on-shelf availability which signals store reliability and customer care through quality shopping experiences. Ensuring on-shelf availability also helps stores stay competitive and the meet ever-changing needs of their customers.

5.5 User training and adoption

To ensure effective implementation and user adoption of the Smart Shelves system, a comprehensive training program and adoption strategy are essential. The training process could be structured to accommodate various levels of technical expertise among retail staff.

5.5.1 Training requirements

Retail stock clerks could receive training through a combination of in-person workshops and online tutorials. The training will cover system operations, including how to interpret notifications, manage inventory through the interface, and troubleshoot common issues. Training materials will be designed to be intuitive, incorporating hands-on demonstrations and interactive modules to facilitate understanding.

5.5.2 Adoption strategies

To encourage smooth adoption, the user could implement a phased rollout of the system, starting with a pilot program in select stores. This approach will allow staff to become familiar with the system in a controlled environment, gather feedback, and make necessary adjustments before a full-scale deployment. Continuous support will be provided through a dedicated helpdesk, offering assistance and addressing any challenges that arise. Additionally, user feedback will be actively sought and used to refine the system and training materials, ensuring that the system meets the practical needs of the retail staff.

5.6 Role of ML

The proposed system leverages optimized ML algorithms to achieve a superior accuracy of 99.35% in distinguishing between products, even those with near-identical features. This high level of precision is accomplished by training the models on a synthetic dataset that mirrors real-world retail conditions, incorporating a diverse array of product attributes such as size, weight, and packaging variations.

The ML models are finely tuned to recognize subtle differences in these attributes, significantly reducing the likelihood of misidentification. This precision is crucial in retail environments where similar products are often stocked together, helping to minimize errors like incorrect restocking and misplaced items. Additionally, the system's ability to adapt to new products and packaging changes ensures sustained accuracy over time, reducing the need for frequent manual updates and further mitigating the potential for human error.

5.7 System reliability and maintenance

While high accuracy and automation underpin system value, sustained performance depends on proactive maintenance and software updates. IoT sensors and edge devices require regular calibration to correct drift in weight and environmental measurements, preserving data integrity. Concurrently, periodic firmware upgrades and model retraining accommodate changes in product appearance and store lighting, preventing algorithmic degradation. We recommend a maintenance schedule of hardware inspections and semiannual algorithm-performance reviews, supported by automated health checks and remote diagnostics. Over-the-air updates can then be deployed seamlessly, minimizing downtime and ensuring continuous, reliable operation.

5.8 Data privacy and security

To address data privacy and security concerns, the "Smart Shelves" system can employ several safeguards to protect the information collected through IoT and ML technologies. First, all data transmitted between devices and servers can encrypted using advanced encryption standards, ensuring that sensitive information remains secure. Additionally, data stored on servers can be protected through encryption and access controls, limiting data visibility to authorized personnel only.

The system can be designed to comply with major data protection regulations, including the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA). It can include features for data anonymization, enabling the system to handle data in a way that preserves user privacy. Users can be given control over their data, with options to opt-out or request the deletion of their information as needed.

Regular security audits and vulnerability assessments can be conducted to identify and address potential risks, ensuring that the system remains resilient against data breaches and other security threats. These measures collectively contribute to a robust framework for protecting data privacy and ensuring compliance with relevant regulations.

5.9 Limitations and future research

While the proposed system shows promising potential, it is important to acknowledge several areas requiring further exploration and refinement. First, the system's end-to-end performance in a live retail setting remains untested, and its adaptability to diversified product assortments and heterogeneous store layouts is unknown. Moreover, the requirement for initial capital outlay-encompassing the procurement and installation of smart scales and environmental sensors, integration with existing network infrastructure, and comprehensive staff training-constitutes a non-trivial barrier for retailers with constrained budgets. Future investigations should evaluate phased deployment strategies, such as targeting high-turnover aisles first, as well as alternative financing models (e.g., equipment leasing or sensor-as-a-service arrangements) to mitigate upfront investment and de-risk widespread adoption.

Integration with enterprise software ecosystems-including inventory management platforms, customer relationship management systems, and loyalty-program databases-could yield a more seamless view of stock levels and restocking needs while enhancing demand forecasting accuracy through richer behavioral data. The practical feasibility and effectiveness of such integrations remain to be validated empirically. Additionally, current system capabilities do not extend to perishable-item management, such as automated tracking of expiration dates or shelf removal of expired products; incorporating these functionalities could further streamline replenishment workflows and reduce waste.

Beyond retail, industries such as healthcare and manufacturing may benefit from analogous IoT-enabled shelf-monitoring solutions; however, these applications remain theoretical until rigorously tested in domain-specific settings. Extending the system to accommodate diverse product forms and environmental conditions will be critical for broad applicability. Finally, augmenting the platform with advanced machine learning modalities-such as natural language processing for voice-driven stocker commands-and predictive perishable-item analytics represents a promising avenue for evolving the system's capabilities and maximizing operational efficiency.

6 Conclusion

This paper introduces and evaluates an innovative system utilizing IoT and ML technologies to streamline inventory management in the retail sector. By employing smart scales and sophisticated ML models, the system autonomously monitors product availability on store shelves and issues timely restocking notifications once predefined inventory thresholds are met. Rigorous evaluations across four ML algorithms-LR, DT, RF, and SVM-were performed, with optimization techniques applied to each. The SVM model demonstrated superior performance, achieving an impressive accuracy of 99.35%, thus validating the efficacy of our optimization strategies.

Our system's practical effectiveness is further underscored by its operational performance, specifically its average notification response time of only 8.66 seconds. This swift and highly accurate response capability significantly enhances real-time inventory management, ensuring product availability and minimizing stockouts.

The proposed solution delivers clear benefits across stakeholders: retailers benefit from optimized inventory management, reduced waste, and improved profitability; retail stock clerks experience decreased manual workload due to automated shelf monitoring; and customers enjoy enhanced shopping experiences through consistent product availability.

Looking ahead, future research should explore testing the system in varied and complex retail environments to better understand and enhance its adaptability. Integrating the proposed system with existing inventory management platforms presents another valuable research direction, potentially facilitating smoother adoption and broader functionality. Additionally, investigating the use of edge computing could provide further performance improvements by enabling faster data processing and reduced response times.

Overall, this research significantly advances the integration of IoT and ML in retail, presenting a scalable, effective, and economically viable inventory management solution poised to shape future retail operations.

Acknowledgements The authors would like to acknowledge and thank Prince Sultan University for their continued support.

Funding This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data availability The datasets generated and/or analysed during the current study are available in the IEEE Data Port repository; see reference number 13, https://doi.org/10.21227/85qk-q280.

Declarations

Competing interests The authors declare no competing interests; in addition, Ali Abdullah S. AlQahtani (as an editorial board member) did not participate in the review or acceptance process.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Sabanoglu T (2023) Topic: Retail market worldwide. Statista. https://www.statista.com/topics/5922/retail-market-worldwide/#editorsPicks
2. Kumar MS, Raut RD, Narwane VS, Narkhede BE (2020) Applications of industry 4.0 to overcome the covid-19 operational challenges. Diabetes & Metabolic Syndrome: Clinical Research & Reviews 14(5):1283–1289
3. Mou S, Robb DJ, DeHoratius N (2018) Retail store operations: literature review and research directions. Eur J Oper Res 265(2):399–422
4. Shankar V, Kalyanam K, Setia P, Golmohammadi A, Tirunillai S, Douglass T, Hennessey J, Bull J, Waddoups R (2021) How technology is changing retail. J Retail 97(1):13–27
5. Ratasich D, Khalid F, Geissler F, Grosu R, Shafique M, Bartocci E (2019) A roadmap toward the resilient internet of things for cyber-physical systems. IEEE Access 7:13260–13283. https://doi.org/10.1109/ACCESS.2019.2891969
6. Chu Y, Feng D, Liu Z, Zhao Z, Wang Z, Xia X-G, Quek TQS (2022) Hybrid-learning-based operational visual quality inspection for edge-computing-enabled iot system. IEEE Internet Things J 9(7):4958–4972. https://doi.org/10.1109/JIOT.2021.3107902
7. Zhu B, Bedeer E, Nguyen HH, Barton R, Henry J (2022) Joint cluster head selection and trajectory planning in UAV-aided iot networks by reinforcement learning with sequential model. IEEE Internet Things J 9(14):12071–12084. https://doi.org/10.1109/JIOT.2021.3133278
8. Aubrey C, Judge D (2012) Re-imagine retail: Why store innovation is key to a brand’s growth in the ‘new normal’, digitally-connected and transparent world. J Brand Strateg 1(1):31–39
9. Haleem A, Javaid M, Qadri MA, Singh RP, Suman R (2022) Artificial intelligence (AI) applications for marketing: a literature-based study. Int J Intell Netw. https://doi.org/10.1016/j.ijin.2022.08.005
10. Manyika J, Chui M, Bughin J, Dobbs R, Bisson P, Marrs A (2013) Disruptive technologies: advances that will transform life, business, and the global economy. McKinsey Global Institute. https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/disruptive-technologies
11. Kaateri M (2022) The process of implementing microsoft dynamics 365 supply chain management in a new warehouse, Metropolia University of Applied Sciences, Bachelor’s Thesis. https://www.theseus.fi/bitstream/handle/10024/756365/Kaateri_Miikka.pdf?sequence=2&isAllowed=y
12. Meyer J-H, González EM, Lopez-Lomelí MA (2022) Better support for supportive jobs. How to improve brand performance through better compensation and training for in-store merchandisers. J Retail Consum Serv 64:102750
13. AlQahtani AAS, Darrat A, Alshayeb T (2023) Synthetic retail dataset machine learning evaluation https://doi.org/10.21227/85qk-q280
14. Bernardini F, Motroni A, Buffi A, Nepa P, Tripicchio P, D’Avella S, Del Col L, Salvatore A (2022) Retail robots with uhf-rfid moving antennas enabling 3d localization. In: 2022 IEEE 12th International Conference on RFID Technology and Applications (RFID-TA), IEEE, pp. 1–4
15. Reyes PM, Visich JK, Jaska P, Roethlein C (2021) Inventory replenishment policies for a grocery supply chain using RFID to improve the performance frontier. Journal of Supply Chain and Operations Management 19(1):52
16. Jebbor S, Chiheb R, El Afia A, Gallab M (2021) Designing a fully automated and integrated inventory and replenishment system for hospitals. International Journal of Systems Science: Operations and Logistics, 1–24
17. Amirapu M, Eshwari R, Nischitha H, Prajwal N, Lakshmi K (2021) Rfid-based smart inventory system for all businesses. International Journal of Inventory Research 6(1):1–13
18. Cui L, Wang L, Deng J (2014) Rfid technology investment evaluation model for the stochastic joint replenishment and delivery problem. Expert Syst Appl 41(4):1792–1805
19. Zia Ur Rahman M, Usman M, Farea A, Ahmad N, Mahmood I, Imran M (2022) Vision-based mobile manipulator for handling and transportation of supermarket products. Math Probl Eng 2022:1–10
20. Saxena N, Sarkar B (2023) How does the retailing industry decide the best replenishment strategy by utilizing technological support through blockchain? J Retail Consum Serv 71:103151
21. Tan WC, Sidhu MS (2022) Review of RFID and IoT integration in supply chain management. Oper Res Perspect 9:100229
22. Kamble SS, Gunasekaran A, Parekh H, Joshi S (2019) Modeling the internet of things adoption barriers in food retail supply chains. J Retail Consum Serv 48:154–168
23. Wölbitsch M, Hasler T, Helic D, Walk S (2020) Show me the money: Rfid-based article-to-fixture predictions for fashion retail stores. In: 2020 IEEE International Conference on RFID (RFID), pp. 1–8. https://doi.org/10.1109/RFID49298.2020.9244903
24. Peng S, He J, Yu H, Cang S (2017) Complex event processing for rfid-enabled retail store. In: 2017 23rd International Conference on Automation and Computing (ICAC), pp. 1–5. https://doi.org/10.23919/IConAC.2017.8081977
25. Lalitha V, Srikanth S, Subramanian S et al (2022) Smart shelf using internet of things. Elementary Education Online 19(2):1983–1990
26. Garaus M, Wolfsteiner E, Wagner U (2016) Shoppers’ acceptance and perceptions of electronic shelf labels. J Bus Res 69(9):3687–3692
27. Soutjis B, Cochoy F, Hagberg J (2017) An ethnography of electronic shelf labels: the resisted digitalization of prices in contemporary supermarkets. J Retail Consum Serv 39:296–304
28. Solomon PJ, Deeter-Schmelz DR (2015) Electronic shelf labeling: An empirical investigation of consumers’ attitudes toward a new technology in retailing. In: Proceedings of the 1993 Academy of Marketing Science (AMS) Annual Conference. Springer, pp. 256–260
29. Stamatopoulos I, Bassamboo A, Moreno A (2021) The effects of menu costs on retail performance: evidence from adoption of the electronic shelf label technology. Manage Sci 67(1):242–256
30. Chehrazi N, Sanders R.E, Stamatopoulos I (2021) Inventory record inaccuracy explains price rigidity in perishable groceries. Available at SSRN 3808358
31. Ock J, Kim H, Kim H-S, Paek J, Bahk S (2019) Low-power wireless with denseness: the case of an electronic shelf labeling system-design and experience. IEEE Access 7:163887–163897. https://doi.org/10.1109/ACCESS.2019.2950886
32. Kim Y, Kim K-H, Cho W-D (2021) Image compression using chain coding for electronic shelf labels (esl) systems. IEEE Access 9:8497–8511. https://doi.org/10.1109/ACCESS.2021.3049868
33. García-Arca J, Prado-Prado JC, Garrido ATG-P (2020) On-shelf availability and logistics rationalization. A participative methodology for supply chain improvement. J Retail Consum Serv 52:101889
34. Milella A, Petitti A, Marani R, Cicirelli G, D’orazio T (2020) Towards intelligent retail: automated on-shelf availability estimation using a depth camera. IEEE Access 8:19353–19363
35. Higa K, Iwamoto K (2019) Robust shelf monitoring using supervised learning for improving on-shelf availability in retail stores. Sensors 19(12):2722

36. Yilmazer R, Birant D (2021) Shelf auditing based on image classification using semi-supervised deep learning to increase on-shelf availability in grocery stores. Sensors 21(2):327
37. Kim G, Moon I (2021) Integrated planning for product selection, shelf-space allocation, and replenishment decision with elasticity and positioning effects. J Retail Consum Serv 58:102274
38. Wang XS, Ryoo JHJ, Bendle N, Kopalle PK (2021) The role of machine learning analytics and metrics in retailing research. J Retail 97(4):658–675
39. Weißhuhn S, Hoberg K (2021) Designing smart replenishment systems: Internet-of-things technology for vendor-managed inventory at end consumers. Eur J Oper Res 295(3):949–964
40. Taleizadeh AA, Shokr I, Konstantaras I, VafaeiNejad M (2020) Stock replenishment policies for a vendor-managed inventory in a retailing system. J Retail Consum Serv 55:102137
41. Bi X, Wang L (2021) Performing weakly supervised retail instance segmentation via region normalization. IEEE Access 9:67761–67775. https://doi.org/10.1109/ACCESS.2021.3077031
42. Hwang H-K, Yoon A-Y, Kang H-K, Moon S-I (2021) Retail electricity pricing strategy via an artificial neural network-based demand response model of an energy storage system. IEEE Access 9:13440–13450. https://doi.org/10.1109/ACCESS.2020.3048048
43. Flaticon: Flaticon: Free vector icons (2023). https://www.flaticon.com
44. FMI TFIA (2021) Supermarket facts. The Food Industry Association: FMI. https://www.fmi.org/our-research/supermarket-facts
45. Lamey L, Breugelmans E, Vuegen M, Braak A (2021) Retail service innovations and their impact on retailer shareholder value: evidence from an event study. J Acad Mark Sci 49:811–833
46. Roy SK, Balaji M, Quazi A, Quaddus M (2018) Predictors of customer acceptance of and resistance to smart technologies in the retail sector. J Retail Consum Serv 42:147–160

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.