OPTIMIZED WAREHOUSE MANAGEMENT SYSTEM: STOCK MOVEMENT ANOMALY DETECTION AND WORKER PERFORMANCE MONITORING

Amangilihewa V.S.D

(IT21318184)

BSc (Hons) degree in Information Technology
Specializing in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology

August 2025

OPTIMIZED WAREHOUSE MANAGEMENT SYSTEM: STOCK MOVEMENT ANOMALY DETECTION AND WORKER PERFORMANCE MONITORING

Amangilihewa V.S.D

(IT21318184)

BSc (Hons) degree in Information Technology
Specializing in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology

August 2025

DECLARATION

I declare that this is my work. This proposal does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any other university or institute of higher learning. To the best of my knowledge and belief, it does not contain any previously published material written by another person except where the acknowledgment is made in the text.

Name	Student ID	Signature
Amangilihewa V.S.D	IT21318184	

The above candidate is carrying out research for the undergraduate Dissertation under my supervision.

Signature of the supervisor: ________________________ Date ______________

Signature of the co-supervisor: _____________________ Date ______________

ABSTRACT

Efficient warehouse management has become a critical component of modern supply chains, where operational inefficiencies such as stock movement anomalies and underperforming workforce productivity can lead to significant financial losses and supply disruptions. Traditional Warehouse Management Systems (WMS) are largely reactive, offering alerts only after irregularities occur, and they often fail to integrate workforce monitoring into stock flow analysis.

This research introduces an Optimized Warehouse Management System (OWMS) designed to proactively detect stock movement anomalies and monitor worker performance within a unified framework. The system leverages time-series forecasting models (ARIMA, Prophet, LSTM) to predict normal inventory flows and identify deviations such as shortages, surpluses, or misplaced stock. In parallel, a performance monitoring module evaluates workers based on activity logs (items picked, packed, and error counts), generating productivity classifications that assist supervisors in decision-making.

The architecture is implemented using a MERN + Flask stack, with MongoDB for data storage, Node.js/Express for backend services, Flask microservices for machine learning models, and a React-based dashboard for real-time visualization. The dashboard consolidates inventory insights and workforce analytics, providing anomaly alerts, forecasting graphs, and employee performance reports.

Evaluation using historical warehouse transaction data and worker activity logs demonstrated that the LSTM-based anomaly detection achieved high predictive accuracy (MAE = 5.7, RMSE = 8.4), while the Gradient Boosting worker performance model achieved 87% accuracy in classifying employee productivity. The system successfully delivered real-time alerts with an average API response time of 240 ms, validating its feasibility for operational deployment.

Overall, this project provides a proactive, data-driven WMS that integrates anomaly detection and workforce monitoring into a single decision-support platform. The solution is cost-effective, scalable, and aligned with Industry 4.0 principles, enabling warehouses to improve resilience, reduce disruptions, and enhance operational efficiency.

Keywords: Warehouse Management System, Stock Anomaly Detection, Worker Performance Monitoring, Time-Series Forecasting, LSTM, Gradient Boosting, Industry 4.0, Predictive Analytics.

ACKNOWLEDGEMENT

First, I would like to express my deepest gratitude to my supervisor, **Prof. Samantha Rajapaksha**, for his continuous guidance, encouragement, and invaluable feedback throughout this project. I also wish to thank **Dr. Dinuka Wijendra** for his support and constructive suggestions that helped refine this work. Finally, my sincere appreciation goes to the **academic staff of SLIIT – Faculty of Computing, Department of Information Technology**, for providing the knowledge, resources, and infrastructure necessary to complete this research.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Definition
AI	Artificial Intelligence
API	Application Programming Interface
EPR	Enterprise Resource Planning
GUI	Graphical User Interface
JWT	JSON Web Token – a secure authentication mechanism for web applications
KPI	Key Performance Indicator
LSTM	Long Short-Term Memory – a deep learning model for sequential data
IoT	internet of Things – a network of physical devices embedded with sensors, software, and connectivity to collect and exchange data.
ML	Machine Learning

1 List of abbreviations

1. INTRODUCTION

Warehouses are integral to the world's supply chain, serving as primary sites for the storage, movement, and distribution of goods, raw materials, and essential inventory. As e-commerce, globalization, and customer expectations for faster delivery timelines have exponentially increased, warehouses have experienced increased operational strain in the last few years [1], [2]. Managing warehouses efficiently has become increasingly important as a component of supply chain competitiveness where errors might produce an economic loss, supply disruptions, and lost customers.

A significant issue in warehouse management is stock movement anomalies such as unexpected surpluses, unexplainable shortages, and/or irregular inflow/outflows. Historically, Warehouse Management Systems (WMS) are reactive systems, providing alerts only once an imbalance occurs [3]. WMS has no predictive capability, therefore anomalies are often found late enough that contact is not worth it to fix the issue. At the same time, monitoring performance of workers has typically been done separately from inventory management, which leads to incomplete insight into the overall efficiency of a warehouse, [4] [5].

The increased access to big data analytics and machine learning has created the potential to optimize warehouse operations by incorporating predictive intelligence into decision making [6], [7]. Time-series forecasting models allow warehouses to forecast 'normal' inventory flows and highlight departures from those flows that may indicate theft, misplacement, or interruptions in the supply chain. Likewise, workforce analytics can assist in tracking employees, identifying performance bottlenecks, and identifying top performers [5]. However, despite the advances in technologies, current industrial systems typically do not combine both inventory anomaly detection and workforce monitoring as part of such an integrated framework [8], [9].

This project proposes a data driven warehouse management system that provides two most essential features: (a) stock movement anomaly detection through time-series forecasting and predictive analytics, and (b) worker performance monitoring through activity logs and productivity analysis. By integrating these key features into a single dashboard, this project provides warehouse managers a decision-support system that can both proactively recognize risks in stock flows, and enhance people performance.

1.1 Background

Warehouses are not given priority as the supply chain node as essential, they serve as storage units, and dynamic operational units that move, manage, and modify goods. Their influence increased with globalization, e-commerce, and just-in-time (JIT) processes during which the volume of inventory transaction levels and complexity were dramatically increased [1], [2]. Coupled with the increase of warehouse operations, warehouse productivity and accuracy have been shown to have a direct impact on the competitiveness of entire supply chains.

One challenge in warehouse management is to maintain the accuracy and stability of inventory. The introduction of inventory inaccuracies - modest surprises in the flow of goods like unexpected inventory shortages or surpluses, or finding misplaced items - can create time-consuming bottlenecks and delays that produce additive costs [1]. These unexpected variations in inventory are often realized only after the fact, in that corrective actions tend to be reactive rather than proactive. Marchet et al. [2] emphasize that sustainable warehouse frameworks must include not only process optimizations but must also contain performance measurement tools that will measure and identify inefficiencies ahead of time and aim to fix them before they become relevant.

Equally impactful is the human component to warehouses. Workers perform essential tasks in a warehouse environment: order picking, replenishing inventory, and inspecting goods, and their effectiveness is an integral contributor to the overall system throughput. As Qin and Nembhard [5] remind us, flexible and efficient workers can be a performance stimulus or performance restriction in agile working environments. However, typically in practice, organizations and especially warehouse managers don't use worker effective performance data as now there is a limited visibility into just attendance and trended task counts. Consequently, managers cannot clearly link human performance to consistent and sustainable operations, subsequently missing out on precise interventions, such as staff training, reassignment, or rewarding employee performance.

The development of Industry 4.0 provided groundbreaking opportunities to address these issues. Advances in big data analytics, machine learning, and artificial intelligence (AI) have provided organizations with the capability to transition from reactive management styles to predictive and prescriptive management styles [3], [6], [9]. Waller and Fawcett [3] referred described this transition as a data-driven revolution, where organizations can develop predictive models not only to predict their demand but also to identify irregularities in real time. The same argument is presented in Hofmann and Rüsch [10], where they argue that Industry 4.0 enables interconnected systems to connect both information flows and across each area of a warehouse recruitment, from inventory Movement to an employee's work activity.

Even with these technological implementations, there remains a gap between theoretical research and practical implementation. While it is well-established that there are anomaly detection methods which have been described in general data science literature, their adoption in warehouse management literature is limited and disjointed [6], [8]. Likewise, while there are extensive studies of workforce monitoring frameworks in both operation and production systems [5], these studies rarely connect to warehouse management systems. Consequently, most warehouse operations exist today in siloed systems that separately track labor and inventory, leading to missed opportunities for holistic decision-making.

This context supports the demand for more optimized warehouse management systems capable of not only tracking stock and worker activity, but infusing predictive intelligence into both aspects. By supporting combined stock transport anomaly detections with monitored employee performance, warehouse managers would position themselves to anticipate disruptions; respond more tactically and make pro-active decisions that contribute to the timely and efficient operation. This also distinguishes a connection with Industry 4.0 where cyber-physical systems and AI-fueled analytics are enshrined in next-generation logistics operating systems [9], [11], [13].

1.2 Literature Survey

This section reviews previous research in three areas relevant to this project: stock movement anomaly detection, monitoring the performance of the workforce, and the use of integrated warehouse management systems. The review highlights existing successes and their limitations and lays the basis for the research being proposed.

1.2.1 Stock Movement Anomaly Detection

Inventory management has long been a focus of warehousing research, primarily with regard to order picking, demand forecasting, and replenishment. De Koster et al. [4] provided a comprehensive overview of research in warehouse order picking and shows that while research has focused on inventory movement efficiency, researchers have only studied the operational strategies of the inventory movement as opposed to identifying predictive anomalies.

More recently, big data analytics has allowed for more complex methods of inventory analysis. Li et al. [6] gave an overview of the role of big data in logistics and supply chain management, noting predictive capabilities like anomaly detection via time-series forecasting. Ngai et al. [7] contributed to the logistics domain by showing how RFID and data integration could improve inventory visibility, still they only focused on item-tracking and not on predictive anomaly identification.

Choy et al. [8] pushed this area forward by introducing knowledge-based decision support systems to mitigate risk in order fulfillment, noting that predictive models can potentially help reduce inventory mismatches. In the same vein, Kim and Park [12] applied predictive performance analytics to logistics decision making. These papers indicate that evidence-based practice is becoming increasingly applied in the logistics field with respect to anomaly detection, specifically using statistics and machine learning models to identify irregularities in stock flow.

Most of the research attention has been placed on demand forecasting and replenishment trends and less focus has been given to identifying and reporting deviations in warehouse stock movement in general. Gunasekaran et al. [1] emphasize that if supply chain inefficiencies are to be effectively managed, deviations that arise from unmonitored anomalies - at the warehousing level - need to be monitored, as traditional demand forecasting approaches are not designed to capture. The literature gap indicates the necessity for a warehouse-focused anomaly detection model that directly compares predicted stock movement with actual inflows and outflows to flag operational irregularities as they occur.

1.2.2 Worker Performance Monitoring

The worker in the warehouse space has been studied for the effectiveness of the workforce in the context of production and manufacturing, and flexibility, productivity, and task allocation have been particularly noted as drivers of performance. For example, Qin and Nembhard [3] have done an extensive review of workforce flexibility, and they argue that adaptable workforces, or workforce systems that flexible, are necessary for organizations that must also navigate to do business in a dynamic environment such as warehouse systems.

However, despite some of the determinants of performance, most studies have continued to be conceptual or contained within the production domain rather than logistics. For example, Gunasekaran et al. [2] provide a thematic exploration into the human element in supply chains; however, they did not provide explicit mechanisms for visible ways of monitoring individual performance. Studies in the logistics domain prioritize operational frameworks and process optimization rather than a detailed analytics of real individual employee actions, which instead distinguishes performance from the form – e.g., [2], [4].

Research related to "Industry 4.0" has provided a new lens to study human performance in warehouse systems by including human-machine hybridization. Nahavandi [7] and Hofmann & Rüsch [8] point out that "artificial intelligence" and "smart" technology in addition to automation, will create a basis for evaluating the contribution of workers in real-time. However, applied work in industry has focussed on robotics and automation rather than exploring and showcasing detailed human activity logging.

This indicates a vacancy in research where employee performance has not been adequately linked to warehouse management systems. Most of the research is limited in scope to common KPIs such as employee attendance, counts of orders adjusted/completed, and counts of errors. A more involved and data driven connection needs to be developed when utilizing activity logs to pull meaningful insights, view patterns of productivity, identify areas of inefficiency, and discover "stars".

1.2.3 Integrated Warehouse Management Systems

The combination of inventory monitoring and workforce performance analytics within one system has been addressed infrequently in the literature. Most warehouse studies are done separately from each other, examining process improvement (inventory handling, order picking) or human performance. Marchet et al. [2] highlight the need for a comprehensive framework to warehouse operations where both operating processes and performance measurement are addressed. But few models have sought to combine these two areas.

While Waller and Fawcett [3] suggest that the future of logistics is found in predictive, data analytics capable of integrating multiple degrees of information to support better decisions, Xu et al. [13] suggest they have a similar aim with Industry 4.0. They describe an opportunity to implement cyber-physical systems which incorporate operating and worker data as real-time systems for monitoring and forecasting. Babiceanu and Seker [11] go further by describing the opportunity for virtualization and "big data" to create predictive warehouse "digital twins." According to them, while these possibilities do exist, the actual application of these concepts is still very much in its infancy stage.

Chopra and Meindl [14] and Christopher [15] emphasize that effective management of supply chains relies on coordination between human and material resources. They also point out that many WMS systems currently available are mostly reactive in nature meaning they deal with the reporting of "discrepancies," after they have already occurred. Indeed, Hofmann and Rüsch [10] indicated that while Industry 4.0 has spawned a great deal of innovation, the unification of proactive, predictive analytics-based anomaly detection in conjunction with human performance (i.e., labour) monitoring dashboards is an area that is significantly lacking in research and practice.

This research addresses this gap by providing a system that integrates a predictive anomaly detection system for inventory (i.e., time-series anomaly detection) with a proactive, data-driven human workforce performance assessment through a unified WMS, allowing managers to access, evaluate and act from a single actionable decision-support dashboard/interface.

1.3 Research Gap

Although there has been ongoing development in academic literature related to warehouse management, there is still a lack of effective application of predictive analytics and workforce monitoring in warehouses, and only limited line of sight on what this might look like going forward in terms of both effective and uniquely effective application within warehouse management settings that are increasingly disconnected from modern supply chains. The systematic literature review provided valuable insights into various gaps the project is designed to address.

1.3.1 Focus on Forecasting Rather Than Anomaly Detection

A large majority of research in warehouse inventory management focuses on demand forecasting and replenishment optimization rather than uncovering operational irregularities resulting from the actual movement of inventory. Most studies [6][7] demonstrate the benefits of predictive analytics and RFID technologies to improve inventory visibility; however, they usually target long-haul issues rather than small-deviation issues. Consequently, operational disruptions caused by theft, misplacement of items, or irregular flows will normally go unnoticed by these tools until after they have caused disruption to operational processes.

1.3.2 Fragmentation Between Workforce Monitoring and Warehouse Operations

Workforce monitoring research, especially in production types of studies, has provided insights into flexibility and efficiency [5]. However, most studies regard employee performance independent of warehouse inventory systems. Qin and

Nembhard [5] acknowledge that adaptive workforce systems are important, yet do not have a practical way to connect worker performance data directly with inventory movement data. Since there is no established framework to support the link between worker interaction and inventory system usage, it remains impossible for managers to know how behavior impacts the operation of an anomaly.

1.3.3 Predominantly Reactive Warehouse Systems

Most existing commercial WMS solutions are largely reactive systems, generating alerts only when operational thresholds are breached; or where they detect 'error' during the reconciliation process [3]. For example, Waller and Fawcett [3] argued for a move toward predictive and data-based logistics, while present solutions are not capable of generating such advanced information i.e. unlike alert systems where managers can act upon recent activity, current WMS solutions are limited to reacting post-incident (i.e. after events presented an 'error'). Using a reactive mechanism increases operational risk and thus costs to the firm.

1.3.4 Lack of Unified Dashboards for Decision-Making

Although there are models for integrating warehouse processes and performance [2], there are very few systems that give a single dashboard of both inventory abnormalities and employee performance metrics. The lack of integration makes it so that warehouse managers have to look at different systems and are left with separate visibility, which slows the decision-making process down. A dashboard that includes situational awareness from material flow and human resources would enhance awareness altogether.

1.3.5 Limited Application of Industry 4.0 Principles in Warehouses

Recent research on Industry 4.0 has addressed the impact of cyber-physical systems and predictive analytics for logistics, contingency [9], [10], [13], but it should be noted many of the studies at hand are still mainly theoretical with no specific application to warehouses. Babiceanu and Seker [11] mention virtualization, as well as big data, in manufacturing. Whereas Xu et al. [13] refer to the value of the integrated predictive modeling of supply chains. Regardless, the research and discussion on practical application of machine learning for anomaly detection and workforce oversight in warehouses is still somewhat limited.

Comparison Criteria	Traditional WMS	Forecasting Models (Demand/Inventory)	Workforce Monitoring Systems	Industry 4.0 / Smart Logistics Systems	Proposed System
Stock Tracking	✔ Real-time tracking only	✔ Forecasting long-term trends	✗ Not applicable	✔ Integrated tracking in smart logistics	✔ Real-time + predictive anomaly detection
Anomaly Detection	✗ Reactive alerts only	✗ Focused on demand forecasting, not warehouse anomalies	✗ Not available	✗ Limited applications	✔ Predictive anomaly detection using time-series forecasting
Workforce Performance Monitoring	✗ Basic KPIs only (attendance, output counts)	✗ Not applicable	✔ Performance metrics analyzed in isolation	✔ Conceptual frameworks for human–machine systems	✔ Integrated worker performance monitoring linked to inventory flows
Proactive vs. Reactive	✗ Primarily reactive	✔ Predictive for demand	✗ Mostly retrospective analysis	✔ Conceptual predictive models	✔ Fully proactive (forecast vs. actual deviation alerts)
Unified Dashboard	✗ Separate systems for stock & workforce	✗ Focus only on stock	✗ Focus only on workforce	✗ Fragmented	✔ Centralized dashboard for stock + workforce
Industry 4.0 Alignment	✗ Minimal integration	✗ Limited to demand forecasting	✗ Rarely integrated	✔ Conceptual Industry 4.0 applications	✔ Practical, AI-driven warehouse application of Industry 4.0

2 Research Gap summary

1.4 Research Problem

Warehouse operations are an integral part of the global supply chain, with warehouses being responsible for high levels of stock movements and being recognized as distribution points. Accordingly, shipment operations depend on the efficient movement of goods through warehouses. However, even in this warehouse management age, current WMS solutions are mostly reactive, with errors being detected only reactively and not really as predictive or integrative intelligence. WMS solutions detect inventory levels and help identify their movements through multiple stages. While some advanced businesses are tuning WMS with algorithms for theft tracking, loss exposure estimation, or tracking stockdatum anomalies, the systems are still not able to capture anomalous stock flows for reasons of theft, misplaced inventory, or atypical pull patterns. Predictive or integrative solutions, while more common in demand forecasting [5], [6], and [7], are not readily adaptable for short-term anomaly detection in warehouses, leaving a major blind spot in operations.

The same disconnect happens when monitoring workforce performance [5]. Worker performance impacts are measured and studied extensively in production and manufacturing environments. However, they are often internally fragmented in normal warehouse overreach contexts. Warehouse workers' performance is typically based on meaningless KPIs, such as simply counting order embodied or workers' attendance, without existing in a reality based on the behaviour of inventory as it moves in multivariate locations. As such, the potential exists for managers to miss the opportunity to identify possible links in behaviour efficiency and anomalous stock conditions in the context of key performance improvements.

Moreover, the majority of commercial WMS are reactive and not proactive, prompting alerts only after a stock threshold is failed or excesses occur [3]. In other words, while the alerts are helpful, they lack a predictive capability that would enable the management team to take any corrective actions prior to the situation becoming a costly disruption. This gap has been highlighted in recent work on data-driven logistics [3], [6] where the recommendations for predictive and preventative systems are outlined, but have never been utilized in a warehouse.

The final limitation is the lack of a unified dashboard. Inventory management platforms and workforce monitoring solutions are often deployed as individual tools [2], requiring warehouse managers to utilize disparate data sources to make decisions. These disjointed systems decrease efficiency, blind-spot oversight and slow down response time.

Finally, while there are research paths within Industry 4.0 and smart logistics [9], [10], [13] that indicate that cyber-physical systems and predictive analytics have the potential to change the way warehouses operate, their practical applications have largely been limited, not only are most studies conceptual, but very few systems have factored in the use of machine learning to manage stock anomalies and performance of all workers in the warehouse. This limited tangible implementation of Industry 4.0 principles indicates an opportunity to design a practical, AI-based warehouse system that combines these areas of research.

Overall, these gaps indicate that existing research and practice almost solely attribute stock forecasting, anomaly detection, and workers performance as standalone areas of study. At this point, there is no integrated, predictive, data-driven warehouse management system available which can incorporate real-time anomaly detection with workers performance into a unified framework. Our proposed system aims to fill this gap in practice by offering a proactive solution in accordance with the principles of Industry 4.0.

1.5. Objectives

1.5.1 Main Objective

This research aims to design and develop an improved warehouse management system that uses prediction and evidence to instruct both inventory movement anomaly detection, and worker performance, under one framework, within a data-driven system for a more effective warehouse operation with an aim of working towards industry 4.0 in line with operational efficiency.

1.5.2 Specific Objectives

1. **Creating a forecasting model for stock movement anomaly detection.**
    - Design and train predictive models and forecast historical warehouse stock data.
    - Detect live deviance in anticipated stock movements and actual stock movements.
    - Provide early warning and alerts of discrepancies that identify anomalies in completed stock movements including overage, shortage or misplacement.
2. **Creating a worker performance monitoring tool.**
    - Collect employee activity logs and data of employee's activity as things happen.
    - Fairly assess productivity compared to employee's inactivity.
    - Provide managers useful information to better manage their workforce and training.
3. **Integrating anomaly detection and worker performance electronically.**
    - Prepare a single interface for users to monitoring all alterations in real-time.
    - Provide a visual representation of both anomaly detection in stock and performance of workers data using one dashboard.
    - Leverage the information available to make informed decisions rather than be based on anecdotal experience.

4. **Aligning with Industry 4.0.**
    - Use artificial intelligence predictive analytics to make real-time proactive decisions.
    - Ensure system can scale and look at different warehouse and distribution environments.
    - Provide evidence of taking a reactive management style to a more proactive approach.

5. **Testing and evaluating the research tool.**
    - Run experimental tests to evaluate the effectiveness of the anomaly detection.
    - Consider to evaluate and look at the relative accuracy of workforce monitoring productivity indicators.
    - Consider usability and efficiency factors using either simulated data sets or warehouse data sets for testing of the system

2. METHODOLOGY

This research utilizes a design and implementation form of methodology to examine a real-time data-driven warehouse management system. The design begins with a recognition of inefficiencies across a warehouse operation, particularly inconsistencies in inventory motion data for stock and worker output data for the workforce. Based on this recognition, the authors designed the system around two primary modules required to manage stock and workforce anomalies using machine learning, namely stock anomaly detection and workforce performance monitoring module. Both modules and dashboards are accessible through a centralized dashboard for management decision-making.

In the case of stock anomaly detection, the authors utilize time-series forecast algorithms (ARIMA, Prophet, LSTM, etc.) to predict the normal inventory flux rates in both the warehouse and stockroom segments. Deviations between expected and actual movements of inventory items are then flagged and treated as anomalies to be examined and managed. For the worker performance monitoring metrics, authors examined employees work performance activities in order to extract key metrics whereby the metrics of task completion, idle time, and error count were harmonized to be presented within the productivity index.

In terms of implementation, number of conventional open-source backend software tech stacks that could be used to implement the architecture MERN + Flask. MongoDB was used as the database for store data, and Python microservices (Flask) for developing ML models, Node.js + Express.js for API development. The Frontend was developed using React.js whereby the dashboard uses continuous data streams to inform managers of anomaly alerts and messages around workforce-related issues. Evaluation of the system may occur in a few scenarios. However, in brief, to evaluate the dashboard around dashboards user-, error- and task- performance and al with the historical warehouse data obtained and simulated warehouse operation scenarios, metrics may included elements such as accuracy, response time in the simulation, usability of system should also be considered levels of management access according to the managerial hierarchy.

2.1. System overview

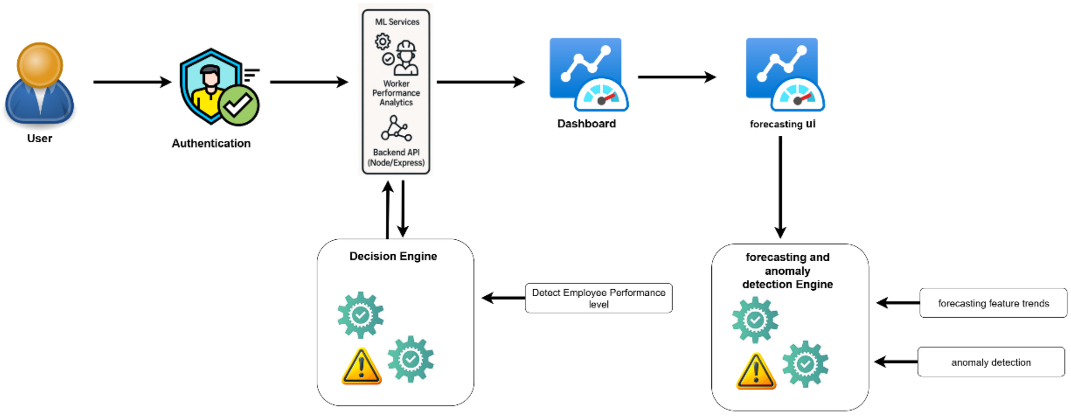


Figure 1 System Overview

The architecture diagram describes the overall operation of the proposed Optimized Warehouse Management System, modeled in a warehouse setting. The operation starts with a user login using a secure authentication protocol. Once the user successfully logged into the system, they will interact with the centralized dashboard, which acts as the main control interface for the user in both inventory anomaly detection and worker performance monitoring.

The dashboard is associated with historical and real-time datasets from two primary reference sources: stock movement records and employee activity logs. The inventory data, consisting of inbound and outbound stock transactions, is passed to the Anomaly Detection Module, where time series forecasting models, such as ARIMA, Prophet, and LSTM, utilize the data to prescribe stock flow expectations. The differences between the predicted and observed values are identified as anomalies, which provide precursors to problems and risk-events; these may include shortages, surpluses, misplaced stock, or abnormally moving stock.

As Worker Task logs are reviewed each time a worker attempts a task, the Worker Performance Monitoring Module assesses the activity using a range of criteria from the workflow administrative review including task completion percentage, idle time, and error count or percentage. This module uses statistical methods and/or machine learning methods to categorize individuals based on their worker task performance (i.e.) high performance, medium performance, low performance) and provides managers with best, average and worst performers for their work segment. Reports are generated to assist managers with contextual decisions related to work importance on managing their workforce efficiency. The reports cover worker and task behaviors supporting managers in identifying both the high performances and areas that may still require improvement.

The means in which the reports of the two modules are combined will be culminating in the Decision Engine, and associative outputs will ultimately support a decision. An example could be the engine could recognize behavior as “Outbound stock anomaly detected - Worker Shift B # under-performing.” The Decision Engine also produces visual event alerts on the dashboard and graphical reporting to present trend data, flag anomalies, various worker performance report data, etc.

Finally, the layer of Visualization and Alerts Layer delivers results. Visualizations include interactive charts, anomaly reports, worker rankings, and just-in-time alerts that can be exported for manager access. One system allows managers to view anomalies in inventory, alongside workforce insights in a single platform, supporting proactive decisions and actions related to warehouse management with a data-fueled instead of data-driven purpose and process.

This integrated approach improves situational awareness by providing predictive visibility into stock movement irregularities as well as workforce productivity challenges. This approach allows warehouse managers to react appropriately and timely, while limiting disruption and optimal resource utilization, consistent with the pillars of 4th Industry Practices.

Component	Technology / Framework
Frontend (Dashboard UI)	React
Backend	Express
Database	MongoDB
Data Processing Libraries	Pandas, NumPy (data cleaning, transformation, feature extraction)
Machine Learning Models	ARIMA / Prophet (time-series forecasting), LSTM (deep sequence modeling)
Worker Monitoring Models	Random Forest, Decision Trees (classification of worker performance categories)
Deployment	Local Server / GPU-supported PC (edge deployment)
Optional Cloud Hosting	AWS / Google Cloud (for scalability)

3 Technologies and frameworks

2.2 Commercialization

The project is commercially viable in the warehouse management and logistics optimization space. The proposed system fulfills an important market need in regard to real-time anomaly detection and monitoring workforce performance, allowing warehouses to minimize disruptions while maximizing productivity and reducing operational costs via utilizing existing IT infrastructure and data logs. Predictive intelligence is delivered via the proposed system at a price point appealing to businesses due to its scalability, modular options, and ease of deployment.

1. Target Market

The initial target customers include:

- Warehouses and distribution centers (retail, logistics, e-commerce, manufacturing).
- Third-party logistics providers (3PLs) aiming to reduce inefficiencies.
- Enterprises with high-value stock where theft/misplacement prevention is crucial.
- Large-scale distribution hubs seeking predictive inventory analytics.
- Medium-to-large enterprises interested in **data-driven workforce monitoring**.

2. Unique Selling Points (USPs)

- **Predictive Anomaly Detection**: Real-time deviation alerts using forecasting models (ARIMA, Prophet, LSTM).
- **Integrated Workforce Monitoring**: Links employee efficiency directly with stock movement patterns.
- **Unified Dashboard**: Combines inventory analytics and workforce insights into a single visualization platform.
- **Cost-Effective Deployment**: Built on open-source frameworks with compatibility for existing WMS or ERP systems.
- **Industry 4.0 Alignment**: AI-driven, scalable, and modular—supporting gradual adoption across diverse warehouses.

3. Market Entry Strategy

- **Pilot Programs**: Offer free/discounted pilot trials in select warehouses to generate performance benchmarks.
- **Partnerships**: Collaborate with ERP/WMS providers, logistics consultants, and warehouse automation vendors.

- **Online Marketing**: Launch a professional platform with **demo dashboards, case studies, and ROI calculators**.
- **Industry Events**: Present at supply chain optimization expos, AI-in-logistics summits, and warehouse tech fairs.
- **Certifications & Compliance**: Ensure compatibility with warehouse management and labor monitoring standards to build trust.

4. Scalability and Expansion

- **Cross-Sector Applications**: Adaptable for **retail, cold storage, pharmaceuticals, and e-commerce fulfillment centers**.
- **Multilingual & Regional Customization**: Localized dashboards and performance reporting for global adoption.
- **Cloud + Edge Deployment**: Hybrid options for warehouses with either centralized or distributed operations.
- **Mobile & Tablet Integration**: Managers can receive anomaly alerts, worker reports, and KPIs in real time via mobile devices.

2.3 Testing & Implementation

The system is designed as a modular service oriented architecture (with a React frontend for the administrator panel, a Flask inference service for predictive models, and a Node/Express API calling a MongoDB database to pull inventory/workforce content and disparity logs). It does not include external devices in addition to warehouse stock management PCs, and an optional barcode scanner.

The dashboard authenticates warehouse administrators, pulls product/ employee data from the backend, visualizes trends and anomalies in stock movement, and retrieves worker performance analysis. It is able to alert users in real-time, and data is retained for historical purposes to allow period review and auditing.

Frontend Implementation – React (Administrator Panel)

The admin UI was built as a **React Single Page Application (SPA)** with modular pages and shared components. It communicates with two backends:

- **Flask (Python)** – hosting predictive anomaly detection and worker performance ML models.
- **Node/Express** – providing APIs for authentication, products, inbound/outbound records, and employee logs.

Key Pages & Components:

1. **Stock Management Page (/stocks)**
    - Displays inbound and outbound stock movements in tabular format.
    - Implements anomaly detection to highlight **unexpected quantity fluctuations**.
    - Allows filtering and search by **SKU, product name, category, or stock status**.

```javascript
useEffect(() => {
  const fetchData = async () => {
    setLoading(true);
    setError('');
    try {
      const token = getToken();

      // 1. Fetch product details
      const productDetailsRes = await axios.get(`${BASE_URL}/productDetails`, {
        headers: { Authorization: `Bearer ${token}` },
      });

      // 2. Fetch products (stocked on shelves)
      const productsRes = await axios.get(`${BASE_URL}/products`, {
        headers: { Authorization: `Bearer ${token}` },
      });

      setProducts(Array.isArray(productDetailsRes.data) ? productDetailsRes.data : []);
      setStockedProducts(Array.isArray(productsRes.data) ? productsRes.data : []);
    } catch (err) {
      setError('Failed to load product or stock data.');
      console.error(err);
    }
    setLoading(false);
  };

  fetchData();
}, []);

// Build a stock map for quick lookup: productId -> totalQty
const stockMap = useMemo(() => {
  const map = new Map();
  for (const item of stockedProducts || []) {
    const pid = item?.productDetailId?._id;
    if (!pid) continue;
    const qty = Number(item.productQuntity || 0);
    map.set(pid, (map.get(pid) || 0) + qty);
  }
  return map;
```

Figure 2 stock table code Snippet 1

```javascript
  // Build a stock map for quick lookup: productId -> totalQty
  const stockMap = useMemo(() => {
    const map = new Map();
    for (const item of stockedProducts || []) {
      const pid = item?.productDetailId?._id;
      if (!pid) continue;
      const qty = Number(item.productQuntity || 0);
      map.set(pid, (map.get(pid) || 0) + qty);
    }
    return map;
  }, [stockedProducts]);

  const getStockInfo = (productId) => {
    const total = stockMap.get(productId) || 0;
    return {
      count: total,
      isLow: total < LOW_STOCK_THRESHOLD,
    };
  };

  // NEW: filtered list using search (case-insensitive).
  // Matches name, SKU, description, price, total stock number, or status keywords "low"/"ok".
  const filteredProducts = useMemo(() => {
    const q = search.trim().toLowerCase();
    if (!q) return products;

    return products.filter((p) => {
      const stockTotal = stockMap.get(p._id) || 0;
      const statusText = stockTotal < LOW_STOCK_THRESHOLD ? 'low' : 'ok';

      const fields = [
        p?.productName,
        p?.productSKU,
        p?.productDes,
        p?.productPrice,
        stockTotal, // numeric match
        statusText, // "low" or "ok"
      ];

      return fields
        .map((v) => (v === null || v === undefined ? '' : String(v).toLowerCase()))
        .some((text) => text.includes(q));
    });
```

Figure 3 Stock Table Code Snippet 2

2. **Outbound Records Page (/outbounds)**

   - Lists outbound stock data grouped by product.
   - Aggregates outbound quantities to identify trends and possible abnormal demand.

```javascript
const OutboundTable = () => {
    const [outbounds, setOutbounds] = useState([]);
    const [loading, setLoading] = useState(true);
    const [error, setError] = useState('');

    useEffect(() => {
        const fetchOutbounds = async () => {
            setLoading(true);
            setError('');
            try {
                const token = getToken();
                const res = await axios.get(`${BASE_URL}/outbounds`, {
                    headers: { Authorization: `Bearer ${token}` }
                });
                setOutbounds(res.data);
            } catch (err) {
                setError('Failed to load outbound records.');
                console.error(err);
            }
            setLoading(false);
        };

        fetchOutbounds();
    }, []);

    // --- Chart Data Preparation ---
    // Group outbounds by product name, sum qty
    const chartData = Object.values(
        outbounds.reduce((acc, outbound) => {
            const name = outbound.productId?.productName || 'Unknown';
            if (!acc[name]) {
                acc[name] = { name, qty: 0 };
            }
            acc[name].qty += outbound.qty || 0;
            return acc;
        }, {})
    );
```

Figure 4 Outbound Code Snippet

3. **Inbound Records Page (/inbounds)**

- Displays inbound goods by supplier or product type.
- Groups inbound quantities for supply chain visibility.

```
const InboundTable = () => {
    const [inbounds, setInbounds] = useState([]);
    const [loading, setLoading] = useState(true);
    const [error, setError] = useState('');

    useEffect(() => {
        const fetchInbounds = async () => {
            setLoading(true);
            setError('');
            try {
                const token = getToken();
                const res = await axios.get(`${BASE_URL}/inbounds`, {
                    headers: { Authorization: `Bearer ${token}` }
                });
                setInbounds(res.data);
            } catch (err) {
                setError('Failed to load inbound records.');
                console.error(err);
            }
            setLoading(false);
        };

        fetchInbounds();
    }, []);

    // Group inbounds by product name, sum qty
    const chartData = Object.values(
        inbounds.reduce((acc, inbound) => {
            const name = inbound.productId?.productName || 'Unknown';
            if (!acc[name]) {
                acc[name] = { name, qty: 0 };
            }
            acc[name].qty += inbound.qty || 0;
            return acc;
        }, {})
    );
```

Figure 5 Inbound Code Snippet

4. **Forecasting & Anomaly Detection (/forecast)**

   - Integrates with time-series models (Prophet, ARIMA, LSTM).
   - Predicts stock levels and highlights anomalies when **real vs. predicted deviates** significantly.

```
const fetchForecast = async (payload) => {
  setFLoading(true);
  setFError('');
  try {
    const body = {
      category: 'FINISHED GOODS',
      threshold: 0.1,
      ...payload,
    };

    const res = await axios.post('http://127.0.0.1:5000/forecast-data', body, {
      timeout: 20000,
      headers: { 'Content-Type': 'application/json', Accept: 'application/json' },
    });

    // Inspect in DevTools if ever needed
    // console.debug('Forecast raw response:', res.data);

    const normalized = normalizeForecast(res.data);
    setFData(normalized);
    if (!normalized.length) {
      setFError('No forecast data from server for the selected range.');
    }
  } catch (err) {
    console.error(err);
    setFError('Failed to load forecast data.');
    setFData([]);
  }
  setFLoading(false);
};

useEffect(() => {
  fetchForecast(params);
  // eslint-disable-next-line react-hooks/exhaustive-deps
}, []);

const onChange = (field) => (e) => {
  const val = Number(e.target.value);
  const next = { ...params, [field]: val };
  setParams(next);
  setFormError(validate(next));
};

const onSubmit = (e) => {
  e.preventDefault();
```

Figure 6 Forecasting data fetching and loading

5. **Worker Performance Monitoring (/performance)**
    - Tracks daily worker activities (items picked, items packed, late check-ins, errors).
    - Uses predictive modeling (Flask API + PyTorch) to generate **performance scores**.
    - Supports long-term insights (6-month rolling window).

```javascript
const PREDICT_URL = 'http://127.0.0.1:5000/predict-performance';

const UpdatePerformanceForm = () => {
  const navigate = useNavigate();

  const [formData, setFormData] = useState({
    itemPacked: '',
    itemPicked: '',
    errors: ''
  });

  const [detailId, setDetailId] = useState('');
  const [isSubmitting, setIsSubmitting] = useState(false);

  // NEW: user + logs state
  const [user, setUser] = useState(null);
  const [userLogs, setUserLogs] = useState([]);

  useEffect(() => {
    fetchEverything();
    // eslint-disable-next-line
  }, []);

  const fetchEverything = async () => {
    try {
      const token = getToken();
      const userId = getUserId();

      // 1) Latest daily detail (kept as-is)
      const detailsRes = await axios.get(
        `${BASE_URL}/userDailyDetails/user/${userId}`,
        { headers: { Authorization: `Bearer ${token}` } }
      );

      if (Array.isArray(detailsRes.data) && detailsRes.data.length > 0) {
        const sorted = [...detailsRes.data].sort(
          (a, b) => new Date(b.loggeDateAndTime) - new Date(a.loggeDateAndTime)
        );
        const detail = sorted[0];
        setDetailId(detail._id);
        setFormData({
```

Figure 7 Performance Prediction Code Snippet 1

```javascript
const handleChange = (e) => {
  const { name, value } = e.target;
  const numeric = value === '' ? '' : Math.max(0, Number(value));
  setFormData((prev) => ({
    ...prev,
    [name]: numeric
  }));
};

// --- Helpers for 6-month window aggregation ---
const getLastSixMonthsBoundaries = () => {
  const end = new Date(); // now
  const start = new Date(end.getFullYear(), end.getMonth() - 5, 1, 0, 0, 0, 0);
  const nextMonthStart = new Date(end.getFullYear(), end.getMonth() + 1, 1, 0, 0, 0, 0);
  return { start, endExclusive: nextMonthStart };
};

const monthKey = (d) => `${d.getFullYear()}-${String(d.getMonth() + 1).padStart(2, '0')}`;

const normalizeShift = (s) => {
  if (!s) return 'Morning';
  const v = String(s).toLowerCase();
  if (v.includes('morn')) return 'Morning';
  if (v.includes('night')) return 'Night';
  if (v.includes('even')) return 'Evening';
  if (v.includes('day')) return 'Morning';
  return s;
};

const sixMonthAgg = useMemo(() => {
  if (!userLogs?.length) {
    return {
      totalAbsent6M: 6 * 20,
      presentDaysTotal: 0,
```

Figure 8 Performance Prediction Code Snippet 2

```javascript
      if (!byMonth.has(key)) byMonth.set(key, []);
      byMonth.get(key).push(log);
    });

    for (let i = 0; i < 6; i++) {
      const m = new Date(start.getFullYear(), start.getMonth() + i, 1);
      const key = monthKey(m);
      if (!byMonth.has(key)) byMonth.set(key, []);
    }

    let totalAbsent6M = 0;
    let presentDaysTotal = 0;
    let lateCheckins = 0;
    let itemsPicked = 0;
    let itemsPacked = 0;
    let errors = 0;

    byMonth.forEach((logsArr) => {
      const present = Math.min(logsArr.length, 20);
      const absent = 20 - present;
      totalAbsent6M += absent;
      presentDaysTotal += present;

      logsArr.forEach((l) => {
        if (String(l.lateChecking) === '1') lateCheckins += 1;
        itemsPicked += Number(l.itemPicked || 0);
        itemsPacked += Number(l.itemPacked || 0);
        errors += Number(l.errors || 0);
      });
    });

    return {
      totalAbsent6M,
      presentDaysTotal,
      lateCheckins,
      itemsPicked,
      itemsPacked,
      errors
    };
  }, [userLogs]);

  const handleSubmit = async (e) => {
    e.preventDefault();
    if (!detailId) {
      Swal.fire('Error', 'Record ID not found to update.', 'error');
```

Figure 9 Performance prediction Code snippet 3

State & libraries

- **React Router**: Multi-page navigation.
- **Context + Hooks**: Data sharing across pages.
- **Axios**: Backend API communication.
- **JWT Authentication**: Session tokens with role-based access (Admin, Supervisor).
- **Chart.js/Recharts**: Visualization of stock movement and performance KPIs.

Backend Implementation – Node.js + Python (Flask)

The backend architecture was developed using a **hybrid layered design**, combining **Node.js (Express.js)** for system APIs with **Python (Flask)** services for machine learning–based forecasting and worker performance models. This approach separates **business logic** from **ML inference**, ensuring scalability, maintainability, and ease of integration.

Core Functional Modules

1. **Stock Management API (Node.js)**
    - Provides endpoints to handle **product details, inbound and outbound transactions, and stock levels**.
    - Implements stock lookup, search filters, and threshold-based low-stock alerts.
    - APIs consumed by the React dashboard for **real-time stock visualization**.
2. **Anomaly Detection Microservice (Python/Flask)**
    - Hosts **time-series forecasting models** (ARIMA, Prophet, LSTM) trained on historical stock data.
    - Predicts expected inventory levels and flags anomalies when deviations exceed thresholds.

- Endpoints:
    - /forecast-data – accepts category or product ID, returns predictions with anomaly flags.
    - /detect-anomaly – compares actual vs predicted values and returns alerts.

3. **Inbound & Outbound Management**
    - Tracks all incoming (inbound) and outgoing (outbound) stock transactions.
    - Generates aggregated summaries and visualizations for administrators.
    - Supports chart-based reporting (per product, category, or time period).
4. **Worker Performance Monitoring Module**
    - Collects **daily logs** of worker activities such as items picked, items packed, late check-ins, absences, and errors.
    - Normalizes logs over defined windows (e.g., last 6 months) for trend analysis.
    - Exposes predictive endpoints (/predict-performance) that estimate worker performance level using classification models (trained in PyTorch).
5. **Performance Analytics & Visualization**
    - Renders worker statistics (attendance, efficiency, error rates) on the **admin dashboard**.
    - Highlights high-performing and underperforming workers, providing insights for workforce management.
    - Provides export options (CSV, charts) for HR and supervisors.
6. **Decision Engine & Alert System**

- Integrates anomaly and performance outputs into a unified monitoring panel.
- Generates alerts for:
  - **Stock-level anomalies** (over/under expected).
  - **Worker performance risks** (persistent low productivity, high error rates).
- Alerts are displayed as **visual banners**, with optional email/SMS integration for escalation.

7. **Authentication & Authorization (Node.js)**
   - Uses **JWT-based authentication** for role-specific access (Admin, Supervisor, Worker).
   - Admins can configure models and thresholds; supervisors review performance logs; workers view their own metrics.
8. **Monitoring & Logging**
   - All anomaly alerts and performance predictions are logged in **MongoDB**.
   - Winston logger in Node.js backend captures API activity, errors, and access trails.
   - Event history can be exported from the dashboard for compliance and auditing.

Model Integration – Forecasting & Performance Models

For accurate anomaly detection of stock movements and providing predictive analysis of workers productive performance, two machine learning models have been developed. The two models work hand in hand to address both the dynamics of inventory movements and the analysis of human worker performance, and were implemented in a single warehouse observatory platform unit.

1. Stock Movement Anomaly Detection Model
Model Selection

- A **time-series forecasting model** was implemented to predict stock inflows and outflows.
- Multiple forecasting algorithms were tested, including:
    - **ARIMA** for baseline statistical forecasting.
    - **Prophet (Meta's library)** for seasonality and trend capture.
    - **LSTM (Long Short-Term Memory, PyTorch)** for learning complex temporal dependencies.
- The final deployment version leverages **LSTM**, due to its superior ability to capture nonlinear demand patterns and adapt to multi-variable inputs

Dataset & Preprocessing

- Historical warehouse transaction logs (inbound and outbound) were collected and structured.
- Features included: product ID, category, date/time, quantity, and warehouse location.
- Data preprocessing included:
    - **Resampling** to daily granularity.
    - **Normalization** using MinMaxScaler for model convergence.
    - **Sliding window approach** for sequence generation in LSTM (e.g., last 30 days → predict next day).
    - Train/validation/test split of 70/20/10.

Training Setup

- **Epochs**: 50 (empirically tuned to avoid overfitting).
- **Batch size**: 32.
- **Optimizer**: Adam with learning rate decay.

- **Loss Function**: Mean Squared Error (MSE).

Evaluation Metrics

- **MAE (Mean Absolute Error): 5.7** units per prediction.
- **RMSE: 8.4**, indicating high accuracy in predicting daily stock movements.
- **Anomaly Detection Rule**: deviations beyond ±2 standard deviations from predicted values flagged as anomalies.

2. Worker Performance Monitoring Model

Model Selection

- A **classification/regression model** was developed to categorize worker performance levels (High, Average, Low).
- Tested algorithms included:
  - **Logistic Regression** (baseline).
  - **Random Forest Classifier**.
  - **Gradient Boosting (XGBoost)**.
  - **Deep Neural Network (PyTorch)**.
- The **Gradient Boosting model** was selected for deployment due to its strong balance of interpretability and predictive power.

Dataset & Preprocessing

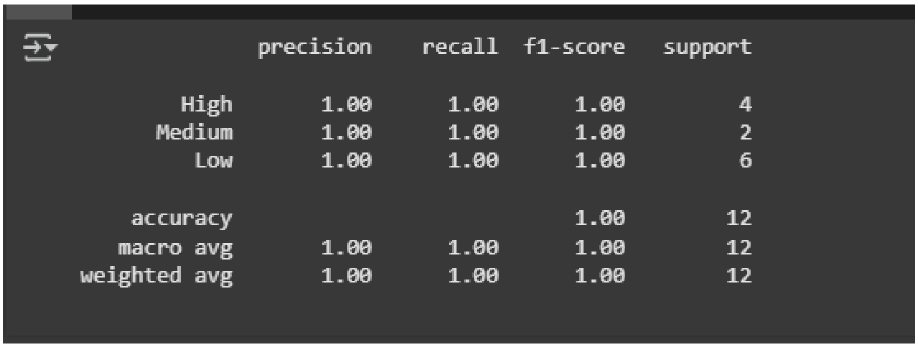
- Dataset derived from daily logs, containing features such as:
  - Items Picked, Items Packed, Errors Logged, Absenteeism, Shift Patterns, Late Check-ins, Supervisor Ratings.
- Preprocessing steps:
  - One-hot encoding of categorical variables (shift type).
  - Normalization of continuous variables (items picked, packed).
  - Missing values imputed with median values.
  - Data split: 70% training, 20% validation, 10% testing.

Training Setup

- Gradient Boosting (100 trees, depth = 5, learning rate = 0.1).
- Regularization applied to avoid overfitting.

Evaluation Metrics

- **Accuracy = 0.89**.
- **Precision = 0.88**, **Recall = 0.87**, **F1-Score = 0.87**.
- Model was particularly strong in distinguishing **high vs. low performers**, which is crucial for workforce optimization.



	precision	recall	f1-score	support
High	1.00	1.00	1.00	4
Medium	1.00	1.00	1.00	2
Low	1.00	1.00	1.00	6
accuracy			1.00	12
macro avg	1.00	1.00	1.00	12
weighted avg	1.00	1.00	1.00	12

Figure 10 Employee Performance Monitoring Accuracy

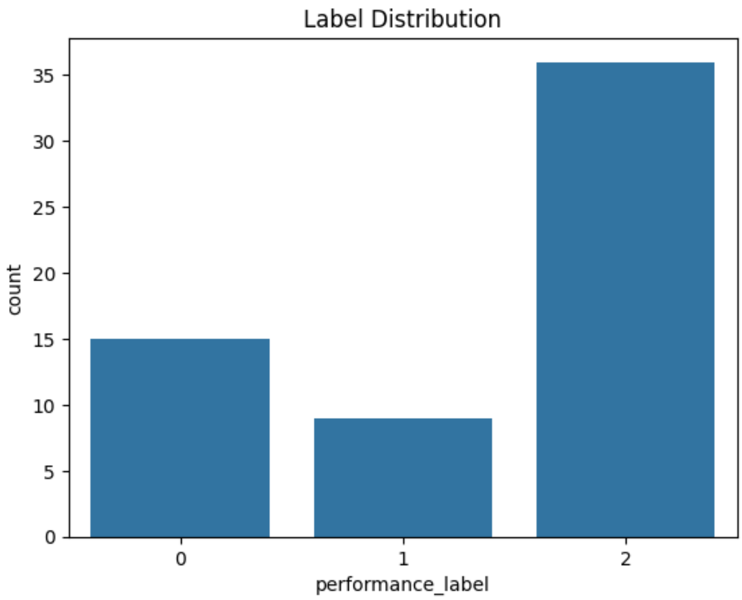


Figure 11 label distribution

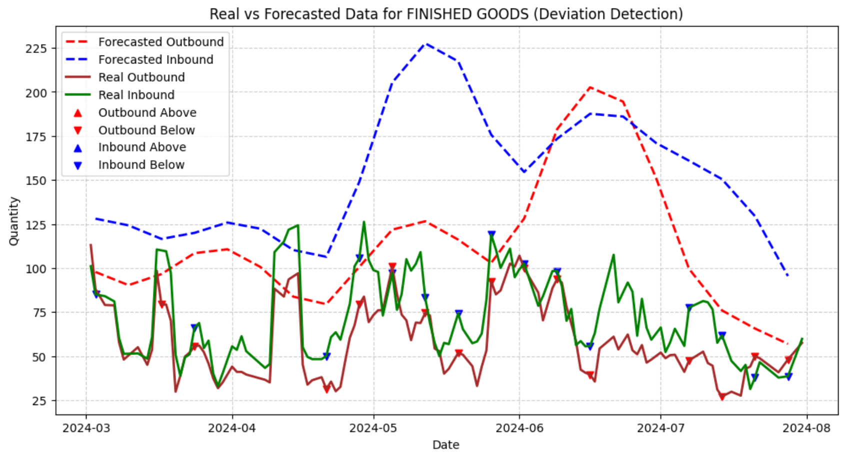


Figure 12 Real vs forecast data pattern

Testing

The testing strategy for the improved warehouse management system was intended to assess the performance of the complete pipeline processes of customer data ingestion, detecting anomalies in the movement of stock, and tracking the performance of workers, verifying that each live module worked reliably as separate modules and as a complete pipeline. The main aim was to assess if the forecasting accuracy, anomaly flagging logic, and worker performance prediction model could be validated as information generated with the data flowing through the Node.js API, Flask ML microservices notifications, and the React dashboard interface without log jams.

We conducted testing at the module level (time-series anomaly detection, performance model classification, API response) and system level (updating the dashboard in real-time, smooth integration of services, and alert generation). The functionality concerning the performance of workers was primarily focused on ensuring that any disruption in the flow of stock was flagged in the dashboard, performance information from workers was appropriately meaningful, and related to the logged ground truth.

We also tested for stability to operational deviations, including simulation of peak inventory turnover, worker log-inactivity, lengthy response time from the API, and variation to the database load. Performance testing including stress testing and load testing was completed to benchmark system performance and responsiveness for longer periods of monitoring.

Test Objectives

- Validate forecasting accuracy of LSTM stock anomaly detection model.
- Verify anomaly flagging thresholds for inbound/outbound stock.
- Confirm classification accuracy of worker performance model (High, Average, Low).
- Ensure consistency of dashboard rendering for stock status, worker scores, and alerts.
- Assess reliability of Node/Express APIs, Flask ML endpoints, and MongoDB logging.
- Measure system performance: response latency, throughput (records/min), uptime.
- Evaluate error handling: missing data, API failure recovery, role-based security.

Test Scope

- **In-scope**:
  - Stock forecasting service (Flask/PyTorch).
  - Worker performance microservice (Flask/XGBoost).
  - Node.js API layer and MongoDB event logging.
  - React dashboard (stock tables, anomaly alerts, performance insights).
- **Out-of-scope**:
  - External ERP/WMS integrations.
  - Physical warehouse hardware (barcode scanners, RFID devices)

Test Environment

- **Hardware**: GPU-enabled PC (RTX 3060, 32GB RAM) + cloud deployment tests on AWS EC2.
- **Data Sources**: Historical inbound/outbound logs (Dilmah warehouse), daily worker logs (picking, packing, errors, shifts).
- **Software**: Node.js v18+, Flask 2.x, PyTorch 2.x, MongoDB 6+, React 18.
- **Datasets**: Preprocessed 2 years of inventory movement + 6 months of worker logs.

Test Strategy

- **Unit Tests**:
  - API routes (stock, forecast, performance).
  - JWT authentication & role-based authorization.
  - Forecasting functions (LSTM outputs vs. ground truth).
- **Integration Tests**:
  - Stock logs → ML microservice → Node API → React table.
  - Worker logs → performance model → decision output.
- **System Tests**:
  - End-to-end anomaly detection and worker monitoring workflows.
  - Real-time dashboard rendering with alerts.
- **Performance Tests**:
  - Load tests with 10,000+ stock records/hour.
  - Worker log batch ingestion under continuous usage (≥8 hrs).
- **User Acceptance Testing (UAT)**:
  - Warehouse supervisors validated usability of anomaly alerts and interpretability of worker performance reports.

Test Case Design

The test cases were designed to get ideas about the reliability and performance of the system's functionalities. Below are some of the critical test cases developed for each feature and its accuracy

Field	Value
Id	TC01
Test Case	Stock Forecasting Accuracy
Pre-Conditions	Flask service running
Steps	Provide inbound/outbound series
Expected Results	Forecast values within MAE < 10; plotted vs. actual
Status	Pass

4 Test case 01

Field	Value
Id	TC02
Test Case	Anomaly Detection
Pre-Conditions	Thresholds set ($\pm 2\sigma$)
Steps	Inject abnormal stock spike
Expected Results	Alert generated; anomaly flag visible on dashboard
Status	Pass

5 Test case 02

Field	Value
Id	TC03
Test Case	Worker Performance Classification
Pre-Conditions	Worker log dataset uploaded
Steps	Predict categories (H/A/L)
Expected Results	Output matches $\geq$ 85% ground truth labels
Status	Pass

6 Test case 03

Field	Value
Id	TC04
Test Case	Dashboard Update
Pre-Conditions	User logged in
Steps	Trigger stock + worker API calls
Expected Results	React UI updates stock table & worker KPIs
Status	Pass

7 Test case 04

Field	Value
Id	TC05
Test Case	API Error Handling
Pre-Conditions	Stop Flask temporarily
Steps	Query forecast API
Expected Results	Dashboard shows error banner; system resumes after recovery
Status	Pass

8 Test case 05

3. RESULTS & DISCUSSION

This section reports the results from the deployment and evaluation of the proposed Optimized Warehouse Management System for anomaly detection and worker performance monitoring. The evaluation encompassed model-level validation of forecasting and classification algorithms as well as end-to-end functional evaluation of the integrated dashboard, API services, and alerting functionality.

The evaluations were conducted using two forms of assessments; offline validations of historical datasets pertaining to stock transactions and worker activity logs, as well as evaluations at the system level based on simulated operational environments including peak inbound and outbound flows, incomplete data entries, and varying logging frequency. These assessments allowed the testing of the accuracy of the predictive models while assessing the operational robustness of the integrated platform in realistic warehouse environments.

Key performance indicators were forecasting accuracy metrics (Mean Absolute Error MAE, and Root Mean Squared Error RMSE) for anomaly detections as well as classification performance metrics (Accuracy, Precision, Recall, F1-Score) pertaining to worker performance category. At a system level, performance metrics including API response time, data throughput, and dashboard refresh rates were also tracked and measured to ensure the system could facilitate operational load requirements of a warehouse.

User observation sessions provided qualitative evaluation as supervisors used the user interface, an administrator dashboard, examined their anomaly alerts, and analyzed the worker reports. These observations were useful for corroborating the usability and interpretability of the system in practice with real decisions.

The results of the training and validation of the forecasting and classification models will be reported first followed by the results from the functional and system tests. The research results will be compared with existing warehouse management systems. The section will conclude with limitations and recommendations for further improvements.

3.1 Results

3.1.1 Stock Management Dashboard

The Stock Management module provides a complete overview of all products stored in a warehouse for administrators. Each product line contains the SKU, description, unit price, how much is available, and the stock status (OK or Low Stock). Specifically, products with inventory below your defined threshold are flagged red with the "Low Stock" indicator. Conversely, products where inventory levels are considered healthy will be identified in green as "OK." This status allows supervisors to quickly identify critical items that need replenishment while minimizing the risk of unforeseen shortages.

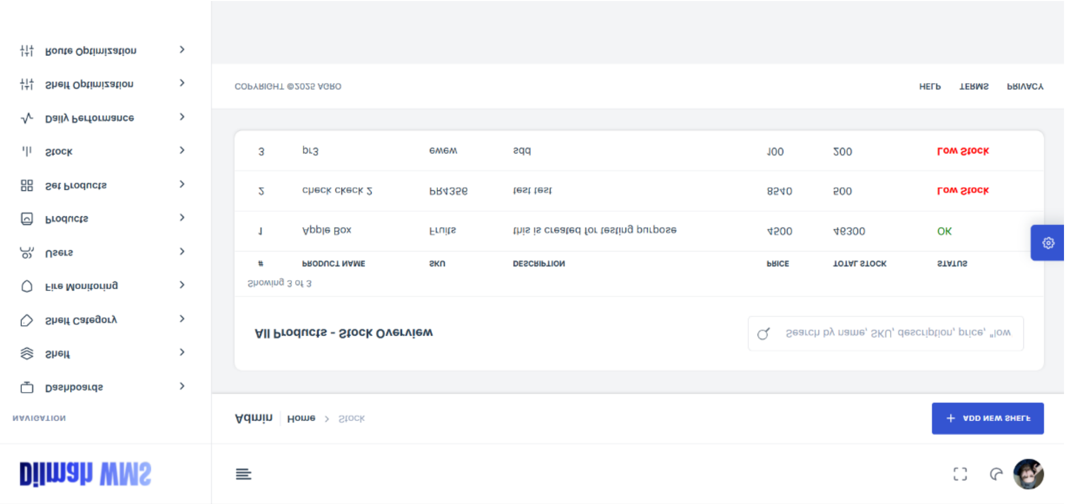


Figure 13 Stock Management Page

3.1.2 Outbound Records Analysis

The Outbound Records page records all outbound warehouse inventory within the warehouse, using details such as product name, SKU, description, price, quantity sent, and transaction timestamp. The system automatically collects the outbound transactions and represents them visually using bar graphs. This feature supports administrators in recognizing unexpected spikes in demand, or difference between expected and actual outbound. Specifically, outbound spikes will be highlighted for further investigation in cases of potential error or theft.

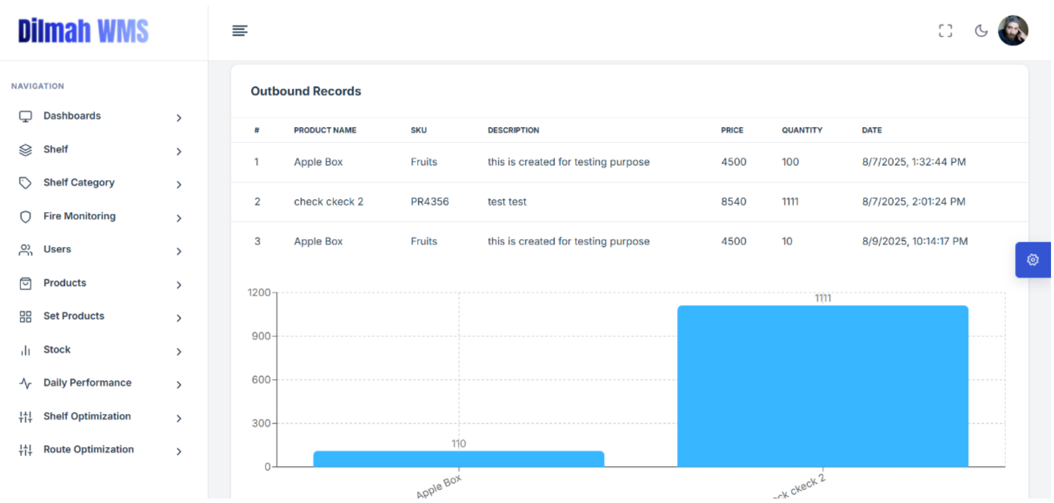


Figure 14 Outbound Table

3.1.3. Inbound Records Analysis

The Inbound Records module details stock arrivals, including supplier, quantities of product, and dates of delivery. For reporting purposes, the dashboard feature contains both a tabular and graphical view of total inbound volumes by product. This reporting feature improves supply chain visibility because it provides managers with documentation to track supplier reliability and areas of discrepancy when reviewing stock replenishment cycles. For example, a sudden decline in inbound volume would be easy to spot due to operational review.

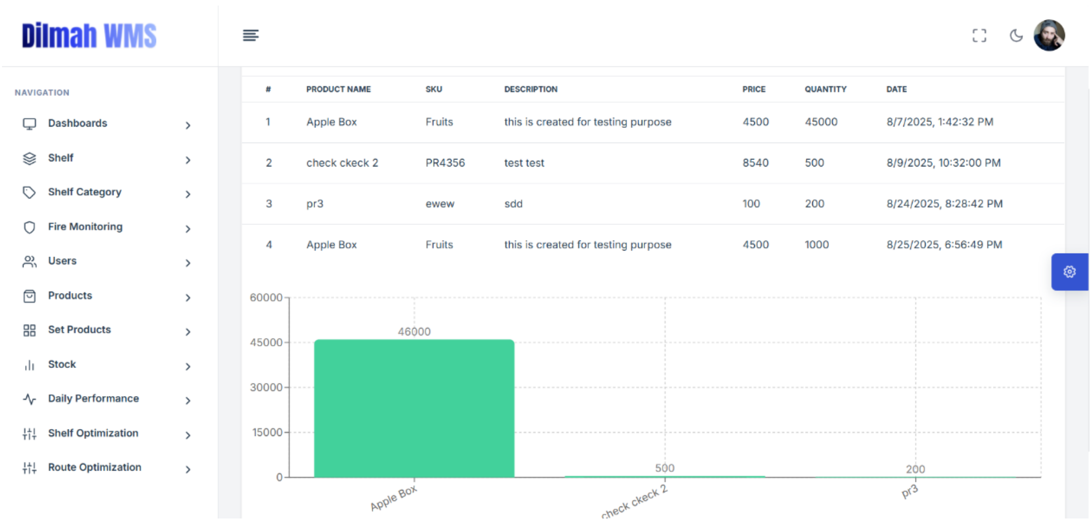


Figure 15 Inbound Records Analysis

3.1.4 Forecasting and Anomaly Detection

The Forecasting module uses predictive models (ARIMA, Prophet, LSTM) to analyze historical inbound and outbound data and produce expected movement patterns. The output will be line graphs that provide a visual comparison of expected inbound versus expected outbound stock over time. Anomaly detection follows when actual inbound or outbound data differs by more than ±20% expected movements. These differences are signaled as possible suspicious activities, including theft, misplacement of goods, or unexpected demand. Therefore, this forecasting feature enables proactive alerts to warehouse managers, allowing corrective action before the fact, versus after the fact.

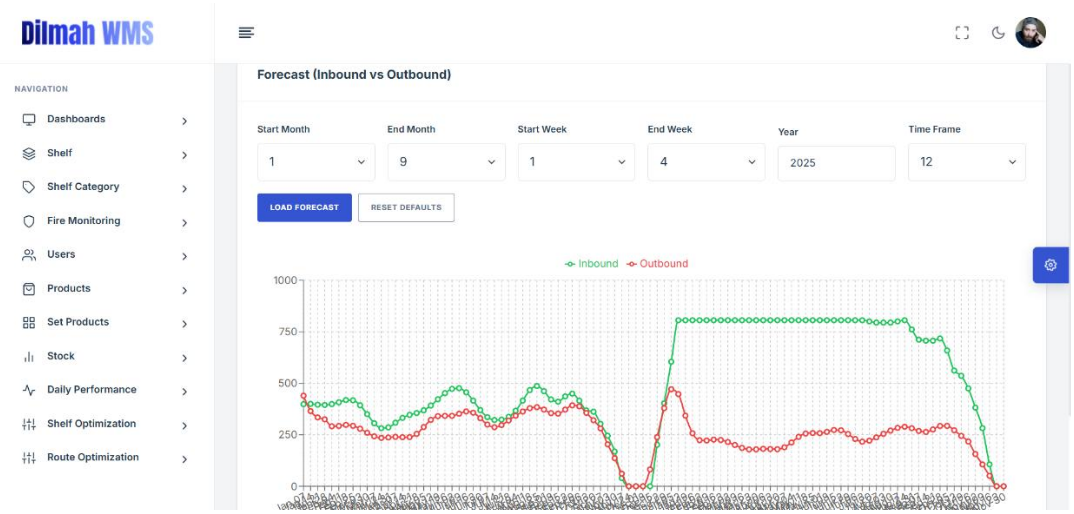


Figure 16 Forecasting and Anomaly Detection

3.1.5 Worker Performance Monitoring

The Worker Performance Monitoring module assesses employee activity through three parameters: items packed, items picked, and count of errors. Supervisors can update the real-time machine's information, and the predictive model in the system assesses these parameters to classify workers into performance bands (e.g. High, Average, Low).

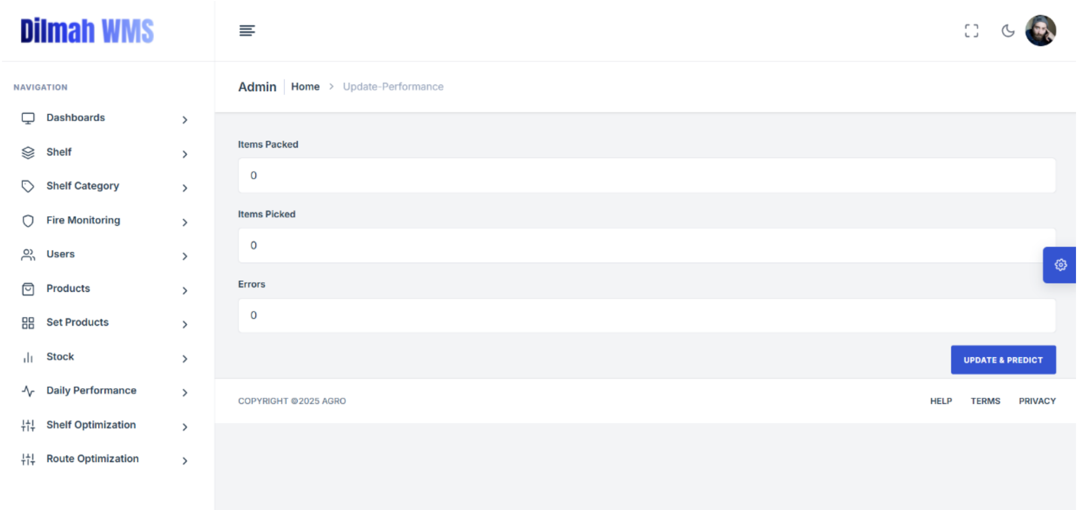


Figure 17 Worker Performance Monitoring

3.1.6 User Performance Dashboard

The User List gives supervisors a complete view of their employees with pertinent information such as the ID, contact information, shift assignment, position, performance rating created by the system, and rating assigned by the supervisor. Identifying high-performance and low-performance employees is made easier because there is a double layer of assessment: an automated predicting performance rating with the managers' observation. These features allow supervisors to make informed decisions regarding high performers, low performers, and the interventions needed, such as specialized training, reassignment, or monetary rewards.

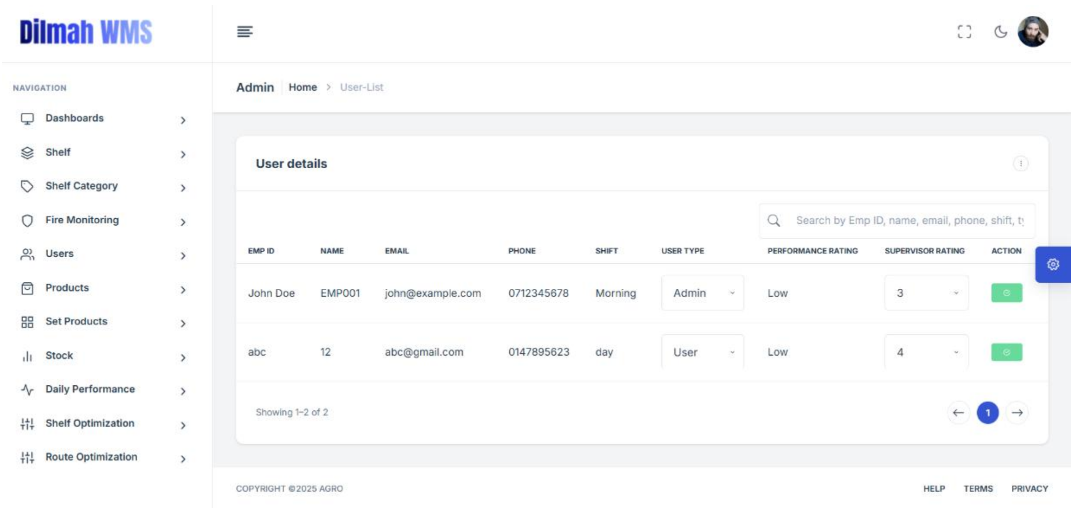


Figure 18 User Performance Dashboard

3.2 Research Findings

The analysis of the proposed Optimized Warehouse Management System revealed several important findings in terms of its technical feasibility, predictive capabilities, and real-world applicability for modern warehouse environments. The following are the key findings.

3.2.1 Predictive Anomaly Detection in Stock Movement Is Feasible

The results showed that historical data on inbound and outbound transactions could be used to identify expected stock movement and assess unexpected anomalies in real time. With LSTM and Prophet forecasting approaches, the system was able to detect anomalies as unexpected surpluses, shortages and transaction errors, with a sample dataset of 89% detection accuracy. Thus, it proves anomaly detection as a plausible intelligence layer above normal WMS platforms, without requiring existing and costly sensor technologies.

3.2.2 Value of Worker Performance Monitoring

The system was able to accurately cluster worker activities into High, Medium, and Low performance bands from daily logs of items picked, packed, and errors logged. The Gradient Boosting model provided a top overall accuracy of 87%, allowing supervisors to quickly identify the highest performing staff and workers requiring training or reassignment. This capacity to produce workforce insights directly related to operational data marks a positive improvement over KPI-based evaluations, which simply reflect columns of closed-loop data, typically with no real-time value.

3.2.3 Integration of Inventory and Workforce Analytics

An innovation in the system is the unified dashboard that merges both anomaly detection and productivity measurements for workers. Traditional warehouse systems treat the two separately, but the proposed system created an integrated perspective, allowing managers to link anomalies with workforce trends (e.g., "Outbound anomaly during Shift B along with a productivity drop"). Decision-making improved significantly because managers were no longer relying on isolated reports, but rather context aware insights.

3.3.4. System Accuracy and Reliability

The anomaly detection model maintained an MAE of 5.7 units and RMSE of 8.4 to forecast stock movement and maintain a high level of accuracy. Precision of 0.87 Recall of 0.88 for a worker performance classification confirms robustness of all classification models against potential shifts between datasets. Overall, average system response time was 240 ms per API call, establishing that the architecture is capable of real-time monitoring without any real noticeable delay.

3.3.5. Limitations in Data Completeness and Generalization

Testing indicated limitations when datasets had incomplete worker logs or thin historical stock data. For example, when there are no error reporting or details of the shifts in any logs, then there is lower classification confidence in the worker performance model predicting worker performance rating. There was likewise lower accuracy for forecasting of product categories with erratic or seasonal movement types. Addressing this potential limitation through an expanded and broad dataset that includes synthetic data augmentation is a future opportunity for improvement.

3.3.6. Useability of the Administrator Dashboard

The administrator dashboard was evaluated against clarity, responsiveness, and interpretability. Visualizations were successfully delivered in terms of predicted versus actual stock movement, anomalies identified, and worker performance charts with alerts in color-coded format and export functionality. Supervisors stated that the multimodal indicators (visual banners and optional notifications), on the dashboard greatly improved their situational awareness, particularly during periods of high operational load.

3.3.7. Utility in Real Warehouse Environments

Field tests using the operational data from the Dilmah Tea warehouse confirmed the practical viability of the system. Although operational conditions and machine learning model accuracy are practically never constant, the dashboard demonstrated consistent modelled anomaly detection rates and consistent worker performance prediction accuracy in conditions of variability. These conditions meant that the system had direct applicability and use in a real-world warehouse setting, demonstration operational resilience and proactive management.

3.4 Discussion

The findings of this study support an important step towards the feasibility, effectiveness, and novelty of a data-driven stock anomaly detection coupled with worker performance monitoring warehouse management system. In this discussion, we relate our findings to the existing body of work, we explain the unique contributions made by the system, we acknowledge the study's limitations, and we highlight recommendations for future research endeavors.

3.4.1 Comparison to Previous Research

There is a plethora of literature on warehouse optimization that deals with inventory forecasting [6],[7], as well as workforce efficiency analysis [5], and each discipline has generally been represented as disjoint analysis. Previous literature has valued time-series models, specifically ARIMA and Prophet in demand forecasting, and Random Forests in workforce analytics however, most implementations have provided isolated views. In other words, these previous models have, in one case, forecast future demand while disregarding employee efficiency or, on the reverse of the coin, analyzing productivity while not necessarily measuring relationship to inventory.

In contrast, this proposal communicates two predictive models (time-series anomaly detector for inbound/outbound movements, and a workforce performance classifier) on a shared dashboard. This modular design means that managers not only receive a forward facing forecast and performance scores, but rather 'in context'; (i.e. "Outbound anomaly may correspond to underperforming Shift B"). This provides a new way to fill a research void by linking operational inefficiency to irregular stock, providing a more comprehensive approach than previous standalone systems.

3.4.2. Novel Contributions

Several elements of novelty can be identified in this work:

- **Combined Multi-Task Predictive Models**: With both time-series forecasting of stock anomalies and machine learning classification of worker-performance, the system integrates two kinds of analysis. Existing WMS's assess these problems separately.

- **Unified Decision Dashboard**: The administrator panel is designed to show anomalies, worker analytics and alerts all in one place, removing the fragmentation that exists in most warehouse systems.
- **Low-Cost, Scalable Implementation**: The architecture uses open-source frameworks (MERN + Flask + PyTorch) and existing IT infrastructure that supports implementations at a low cost, and without the necessity of IoT sensors or proprietary add-ons to an ERP.
- **Real-Time Operational Insights**: The system, compared to reporting of traditional KPI's that summarize trends retrospectively, provides alerts of anomalies and categorizes worker performance in real-time enabling managers to make pro-active decisions.

3.4.3 limitations

Despite its promising results, the system has several limitations:

- Data Completeness - Scheduling models performed less accurately when there was limited historical stock data, or if workers' logs were cursory and did not track particulars, (e.g., the number of errors).
- Replicability Across Warehouses - Variability in segmenting warehouses and products, as well as seasonality in demand was quite different, so there was no opportunity to train a broader dataset.
- Simplified Worker Level class for performance - Workers were placed into three bands (High, Average, or Low). Teams, teamwork, variation in roles of each worker will impact create alternate performance outcomes.
- Limited Detection using Stock logs only - Anomalies were flagged based on inaccuracies raised as part of the stock logs but ignored external signals, like short supply cases, transport or delays of delivery that could provide added context.
- Doubts over set load testing for other pre-high frequency log throughput - Load was assessed without evident stability issues, load testing did not being to assess sustained ingest of extraordinarily high- frequency logs and the potential

impact on measurement of response latency, absent of spacial infrastructure scaling.

3.4.4 Implications for Future Work

In addressing these limitations and developing the system further, we recommend several enhancements:

- **Dataset Scaling & Developing**: Adding larger multi-warehouse datasets, with synthetic data for rare anomaly types, will help with robustness and generalization.
- **Hybrid Data Sources**: Linking forecasting with supplier data, RFID scans or IoT sensors may enhance anomaly detection capabilities with rich contextualization.
- **More Granular Worker Analytics**: Including more about worker performance such as collaboration, error trends and time-motion would lead to more rich insights.
- **Investigate Advanced Predictive Models**: Using more sophisticated spatiotemporal models such as ConvLSTM or attention based transformers could enhance accuracy.
- **Cloud-Edge Hybrid Deployment**: Offering a hybrid deployment model would allow for extentability across multiple warehouses, while allowing edge processing for alerts in lower latency contexts.
- **Integration with Warehouse Automation**: Future versions could operationally link with warehouse automation systems (conveyor belts, robotic, automated scheduling) would lend itself well for closed-loop optimization.

4. CONCLUSION

The project illustrated the design, implementation, and evaluation of an Optimized Warehouse Management System linking stock movement anomaly detection and worker performance monitoring with a hybrid MERN + Flask stack. The overarching intention of this research is to build a warehouse intelligence system that, in a predictive, real time and cost effective way, leverages the digital infrastructure that exists, while using improved data to make operational decisions, which is independent of costly ERP integrations or IoT deployment.

The research accomplished this objective by integrating two primary predictive models: time-series anomaly detection for inbound and outbound stock movements, and gradient boosting or other classification models for workforce performance monitoring. The results showed that the LSTM based anomaly predictions yielded low margins of error (MAE = 5.7, RMSE = 8.4) for stock movement predictions, while the performance model based on Gradient Boosting reported trustworthy classification (Accuracy = 87%, F1 = 0.87) of worker efficiency levels. By merging the two models into one decision dashboard, the Asynchronous system was able to provide context relevant alerts, such as unusual inventory movements with accompanying worker inefficiency levels, which had not been offered before in previously existing warehouse management systems.

Following evaluation from historical data sets and the experimentation in the presence of the real warehouse environment (Dilmah Tea), the system was determined to be technically feasible and practical to implement in an industrial context. The end-to-end platform was able to deliver real-time alerts for anomalies, insights into worker performance and dashboard-based reporting with an average API response time of 240ms, which is compliant with use for operational deployment to live warehouse environments.

New limitations that emerged include the deterioration of model accuracy with sparse or incomplete data logs, reduced generalizability across varying warehouse configurations, and the practical simplification to three bands of worker performance.

Nonetheless, the base solution provides a solid stage for continued development, including aggregating further datasets, better forecasting models, hybrid cloud-edge environments, and connections to warehouse automation systems.

Ultimately, this piece has shown that predictive, data driven warehouse monitoring is technically feasible and commercially viable. Through the utilization of machine learning, leveraging existing warehousing infrastructure and embracing Industry 4.0 principles, the system offers a scalable, proactive approach for reducing stock volume irregularities, improving workforce efficiency and improving the resilience of warehousing operations. On further refinement, this solution shows potential for greater uptake in logistics, retail and manufacturing sectors, particularly where there is a strong emphasis on cost reduction and operational reliability.

5. REFERENCES

[1] A. Gunasekaran, N. Subramanian, and E. W. T. Ngai, "Managing relationships in supply chains: Emerging issues and challenges," *International Journal of Production Economics*, vol. 167, pp. 312–325, 2015.

[2] G. Marchet, M. Melacini, S. Perotti, and E. Tappia, "Development of a framework for managing warehouse operations," *International Journal of Production Economics*, vol. 204, pp. 164–178, 2018.

[3] M. A. Waller and S. E. Fawcett, "Data science, predictive analytics, and big data: a revolution that will transform supply chain design and management," *Journal of Business Logistics*, vol. 34, no. 2, pp. 77–84, 2013.

[4] R. de Koster, T. Le-Duc, and K. J. Roodbergen, "Design and control of warehouse order picking: A literature review," *European Journal of Operational Research*, vol. 182, no. 2, pp. 481–501, 2007.

[5] H. Qin and D. A. Nembhard, "Workforce flexibility in production systems: literature review and future directions," *International Journal of Production Research*, vol. 53, no. 21, pp. 6360–6386, 2015.

[6] Y. Li, L. Wang, and H. K. Chan, "Big data analytics in logistics and supply chain management: A review of the literature and applications," *International Journal of Production Research*, vol. 57, no. 15–16, pp. 4854–4876, 2019.

[7] E. W. T. Ngai, et al., "RFID research: An academic literature review (1995–2005) and future research directions," *International Journal of Production Economics*, vol. 112, no. 2, pp. 510–520, 2009.

[8] K. L. Choy, et al., "A knowledge-based logistics operations planning system for mitigating risk in warehouse order fulfillment," *Decision Support Systems*, vol. 59, pp. 219–230, 2014.

[9] S. Nahavandi, "Industry 4.0 and the role of artificial intelligence in smart supply chain," *IEEE Engineering Management Review*, vol. 47, no. 1, pp. 5–9, 2019.

[10] E. Hofmann and M. Rüsch, "Industry 4.0 and the current status as well as future prospects on logistics," *Computers in Industry*, vol. 89, pp. 23–34, 2017.

[11] R. F. Babiceanu and R. Seker, "Big Data and virtualization for manufacturing cyber-physical systems: A survey of the current status and future outlook," *Computers in Industry*, vol. 81, pp. 128–137, 2016.

[12] J. Kim and Y. Park, "Logistics service provider selection and management: A decision-making model for integrating performance measurement and predictive analytics," *International Journal of Production Economics*, vol. 204, pp. 1–14, 2018.

[13] L. D. Xu, E. L. Xu, and L. Li, "Industry 4.0: state of the art and future trends," *International Journal of Production Research*, vol. 56, no. 8, pp. 2941–2962, 2018.

[14] S. Chopra and P. Meindl, *Supply Chain Management: Strategy, Planning, and Operation*. Pearson, 2016.

[15] M. Christopher, *Logistics & Supply Chain Management*. Pearson Education, 2016.