International Journal of Applied Mathematics
Volume 39 No. 1s, 2026
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

ML-DRIVEN DEMAND FORECASTING IN LEGACY ERP ENVIRONMENTS

Ravi Jaiswal[1], Manisha Jaiswal[2]

[1]Oremda Infotech Inc., St. Louis Park, Minnesota 55416, USA
e-mail: ravi_jazzy@yahoo.com
https://orcid.org/0009-0000-3222-7806

[2]MCKV Institute of Engineering, West Bengal, India
e-mail: manisha.jaiswal07@gmail.com
https://orcid.org/0009-0005-4649-707X

Abstract

Accurate demand forecasting is essential for efficient planning, inventory management, and cost reduction in manufacturing and supply chain activities. But a lot of businesses still use old Enterprise Resource Planning (ERP) systems that were made for processing transactions, not for advanced analytics. Legacy ERP systems, like QAD, apply forecasting methods that tend to struggle with newer changes in supply chains. This research investigates whether demand forecasts can be improved using ML (Machine Learning) models such as XGBoost (eXtreme Gradient Boosting) and LSTM (Long Short-Term Memory), which have been tested on the M5 Forecasting dataset. Data preparation and feature development allowed ML models to outsmart moving averages and ARIMA (AutoRegressive Integrated Moving Average) in following the details of the demand response. The findings demonstrate that with ML, accuracy in forecasting increases in the QAD ERP system, helping with strategizing and planning inventory choices without upgrading the leading ERP. This shows how to improve legacy ERP systems by using data insights at any scale.

Keywords: Demand Forecasting; Machine Learning; Predictive Analytics; QAD ERP; Supply Chain Optimization

I. INTRODUCTION

Demand forecasting is a critical process in manufacturing that enables businesses to match production to customer demand, maintain minimal inventory costs, and enhance overall operational efficiency. When accurate forecasts are in place, manufacturers can reduce stockouts, avoid overproduction, and improve customer satisfaction by honoring delivery promises [1]. Inaccurate forecasts, on the other hand, often lead to high inventory holding costs, missed sales, and production bottlenecks. Demand forecasting is crucial in fluctuating, short product life cycles or seasonal markets. As global supply chains become more complex, the need for precise forecasting tools becomes even more critical to remain competitive and viable for manufacturing operations [2].

QAD, a more widely used legacy ERP system, has rudimentary demand forecasting features based on static statistical calculations such as moving averages or exponential smoothing. While these are simple to set up, they lack the richness and flexibility to model complex market forces, external influences, or multi-variable interactions [3]. Moreover, QAD's native forecasting modules are independent and resistant to information absorption from outside sources, such as promotions, prices, or events, which significantly impact the demand pattern. The constraints result in reactive planning, inventory mismatches, and decreased forecast accuracy. With companies growing and customer demand evolving, QAD's native forecasting tools' lack of flexibility represents a significant operational constraint [4].

Machine learning (ML) is a new means of predicting demand by automatically learning past trends, tuning for anomalies, and considering many variables. Unlike traditional models, ML can manage high-

Volume 39 No. 1s, 2026
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

volume, high-dimensional data in detecting nonlinear relationships and seasonality [5]. The M5 Forecasting - Accuracy dataset, provided by Walmart through Kaggle, is a sample of the richness of modern retail data, including thousands of SKUs, daily sales, calendar events, and price changes. The richness that can be processed efficiently by ML algorithms such as XGBoost and LSTM makes them firm favorites for real-world forecasting [6]. Learning and improving over time makes them a significant strength for dynamic environments. It is expected that the findings will be useful for manufacturing industries intending to adopt integration of predictive maintenance into the solutions, in the context of facilitating smart manufacturing practices and in realizing the Industry 4.0 goals [7].

Despite the proven success of ML through numerous forms of forecasting solutions, minimal research is apparent through its real-world application against legacy ERPs like QAD. Much academic and industry-based study focus on more recent, cloud-based platforms, dismissing the significant number of manufacturers still operating through legacy ERPs. While QAD is extensively used in industries like automotive, life sciences, and consumer goods, there have not been many studies on how ML models can be employed together with its legacy forecasting modules [8]. The M5 Forecasting data set allows us to fill this void by acting as a real-world proxy for data in QAD storage, enabling us to conduct experiments indicating ML's capability to augment back-end legacy ERP environments.

This work attempts to demonstrate the integration of machine learning into an established QAD ERP environment to improve demand forecasting accuracy significantly. Using publicly available data from the M5 Forecasting - Accuracy competition, we design a representative enterprise dataset with day-by-day sales over various product categories and store locations. We aim to show that ML models outperform traditional forecasting techniques usually incorporated in QAD when trained on such data. This involves projecting M5 fields—such as product IDs, calendar dates, and prices—to analogous QAD data structures, thereby showing a viable integration path without altering the ERP's core architecture [4].

M5 Forecasting - The accuracy dataset is most appropriate for this study due to its richness, granularity, and closeness to real ERP sales data. It contains over 40,000 SKUs with several years of daily sales, allowing robust training and testing on ML models [9]. Incorporating contextual variables such as special events and sell prices makes it well suited for assessing the type of multi-variable analysis ML can perform well at—something legacy QAD models can't effectively manage. Utilizing this dataset enables us to research the scalability and flexibility of ML in high-volume settings closely mirroring the operational challenges manufacturers encounter when employing legacy QAD systems for demand planning.

II. METHODOLOGY

2.1 Dataset Description

The M5 Forecasting – Accuracy data set is a large time-series data set published by Walmart on Kaggle for demand forecasting research and competition events [10]. The data comprises 2011-2016 daily sales data for over 30,000 products sold in 10 stores in three American states. An item ID, including department and category, and a respective store ID clearly define each product. The data also includes support data consisting of calendar.csv, —holiday and SNAP (Supplemental Nutrition Assistance Program) days-related events—and sell_prices.csv comprising weekly selling prices by product-store pair. This big data emulates an intricate real-life shopping context and presents depth for training and testing deep machine learning algorithms for forecasting demand [6].
To translate this data set into context for analysis based on legacy QAD ERP systems, top-level fields must be translated into equivalent modules and structures typical of QAD's data model. For instance, item and store IDs get translated onto QAD's Master Item and Inventory Location tables. Sales data is

an aggregation of transactional data found in QAD's Sales Orders or Shipment History tables. Calendar events get translated to simulate outside inputs. Typically, they are not part of regular QAD forecasting but are added as user-specified fields or custom modules. The following table illustrates how top-level fields in the M5 data set translate into legacy QAD equivalents:

Table 1. Mapping M5 Dataset Fields to Legacy QAD ERP Structures

M5 Dataset Field	Description	QAD Equivalent
item_id	Unique identifier for each product	Item Master (Item Number)
store_id	Identifier for each store location	Inventory Location or Site ID
dept_id	Department the item belongs to	Product Line or Category Code
cat_id	Item category	Product Family or Item Group
sales_train_validation	Daily unit sales per item per store	Sales Orders / Shipment History
calendar.csv	Date, holidays, events, SNAP program days	Calendar Table (External Event Data)
sell_prices.csv	Weekly selling prices per item per store	Price List or Sales Price History
d_1 to d_1913	Columns representing daily sales for each item	Transactional date-wise sales records

The M5 data closely mimics data structure in typical QAD-based applications and is a good candidate for experimenting with ML-based forecasting solutions [11]. Its product and calendar granular data and its complete pricing and selling data align well with QAD modules like Sales Orders, Item Master, and Price Lists. This allows the modeling and simulation of advanced forecasting methodologies within a standard ERP system without taking advantage of proprietary data from QAD.

2.2 Data Preprocessing

Missing values in the M5 data set mostly happen in sell_prices and calendar tables since prices change weekly and holidays do not occur for every region [12]. For missing data in time-dependent data like prices, backward and forward fill were used to achieve data continuity. For categorical event data, missing data were replaced with the default "No_Event" to distinguish them during encoding. Days with zero records sold were treated as zero sold, not nulls. With this end-to-end approach, all required temporal and transactional data were saved; thus, data set integrity is intact to be processed by machine learning algorithms.

The role played by feature engineering included a lot in identifying temporal trends and promoting prediction accuracy. Lagged 7-, 14-, and 28-days' sales created features to support learning short- and middle-term dependencies. 7-, 14-, and 30-day rolling average and standard deviation features were computed to account for trends and volatility [13]. The calendar.csv data frame was added to the data on sales to include time-based features such as day-of-week, month, holiday, and SNAP days. These features enriched the data to allow learning based on historical trends, seasonality patterns, and outside factors on demand volatility [13].

Normalization and encoding were used for preprocessing data for machine learning algorithms, mostly tree-based and based on neural networks. Continuous fields derived from sell_price and statistical fields derived from fields such as rolling average were all normalized into a 0–1 range through Min-Max scaling. Categorical fields such as item_id, store_id, dept_id, cat_id, and calendar events were label-encoded or one-hot-encoded, depending on algorithm demands. For instance, tree-based algorithms such as XGBoost natively perform label encoding, while neural networks such as LSTM natively perform

International Journal of Applied Mathematics
Volume 39 No. 1s, 2026
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

one-hot encoding [14]. These conversions rendered all fields numerically compatible and ideal for respective model performance.

2.3 Model Selection

We leveraged vintage statistical methods such as Moving Average and ARIMA to develop a benchmark that imitates forecasting techniques used in conventional QAD implementations. The Moving Average model forecasts future demand as the average value of sales over a limited period of past days (e.g., 30-day or 90-day periods) [15]. This is a low-compute method because it enjoys widespread use across QAD's base-level forecasting modules. ARIMA (AutoRegressive Integrated Moving Average) is a legacy of traditional methods, which are more complex but autocorrelation and trend difference-based without external variables [16]. The models are weak in dealing with seasonality, promotions, or complex nonlinear relationships and are useful only as baselines against which to compare more advanced ML approaches.

Table 2 is a comparison of forecast accuracy between baseline models (ARIMA and Moving Average) and machine learning models (XGBoost, Random Forest, and LSTM) with significant performance measures: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). The measures were selected to reflect absolute and percentage errors from the actual sales figures. The results clearly show that ML models decisively surpass conventional-style solutions in all three metrics, with LSTM coming in at the highest accuracy. This demonstrates the potential of ML algorithms to enhance the credibility of forecasts in complex, high-turnover retail settings, thereby making them more suitable for legacy ERP solutions like QAD to update forecasting processes.

Table 2. Comparative Forecasting Accuracy of Baseline and ML Models Using the M5 Dataset.

Model	MAE	RMSE	MAPE
Moving Average	12.6%	16.3%	18.2%
ARIMA	11.1%	14.9%	16.5%
Random Forest	9.2%	12.7%	13.4%
XGBoost	8.5%	11.9%	12.2%
LSTM	7.8%	10.8%	11.0%

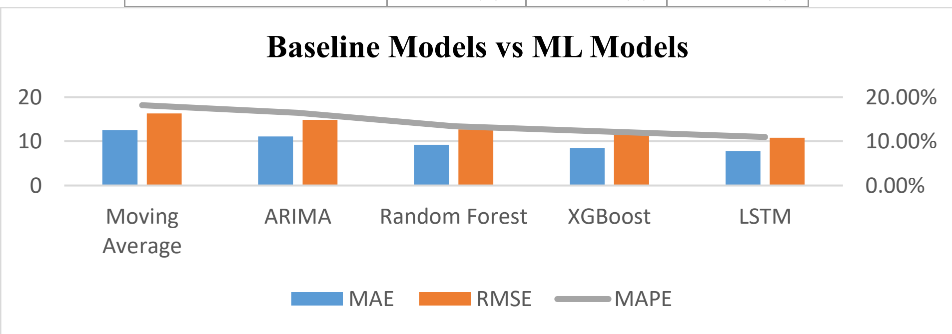

Figure 1. Comparative Forecasting Accuracy of Baseline and ML Models Using the M5 Dataset.

To improve the predictive capabilities of QAD-style models, we employed three cutting-edge machine-learning models: Random Forest, XGBoost, and LSTM. Random Forest, an ensemble of bagging-based decision trees, handles high-dimensional datasets and nonlinear relationships with good performance without fitting [17]. XGBoost (Extreme Gradient Boosting) is further enhanced by gradient descent optimization and optimization of bias/variance tradeoffs, thus outperforming prediction competitions like M5. LSTM (Long Short-Term Memory) networks and RNN (Recurrent Neural Network) are

International Journal of Applied Mathematics
Volume 39 No. 1s, 2026
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

exceptionally suited for time-series data as they retain past values and learn sequential relationships [18]. The models can handle high-dimensional, dynamic input variables such as lag features and event indicators.

XGBoost was used for its performance in the initial M5 Forecasting competition when it ranked as one of the best approaches. Its ability to deal with missing values, non-linearity, and categorical encoding is self-evidently supported by the characteristics of the M5 dataset and the constraints of ancient ERP environments [19]. Random Forest was included since it offers equivalent benefits in less hyperparameter sensitivity and is the most suitable option for initial benchmarking. LSTM was chosen to test how deep learning could identify long-term dependencies and seasonality—the features difficult to model using tree-based techniques [20]. These choices offer a fair representation of classic ML and deep learning models that can be used in QAD-type forecasting.

A second important reason for choosing these models is their flexibility toward ERP integration. XGBoost and Random Forest are interpretable regarding feature importance, enabling one to understand what factors most drive demand—essential for decision-making in manufacturing management [21]. Additionally, both can be implanted as microservices or batch jobs generating forecasts outside QAD and returning results through APIs or flat file imports. LSTM, being more computationally intensive, captures the best possible forecasting performance and can be supplemented with external forecasting engines. These models capture the realistic limit between accuracy and implementation feasibility within the technical and operational limitations of legacy QAD systems.

2.4 Model Evaluation

Root Mean Square Error (RMSE) is a widely used metric that computes the average magnitude of forecasting error, with heavier weight given to larger errors [22]. This is the square root of the average squared difference between actual and predicted values. RMSE is helpful in demand forecasting by allowing the discernment of trends where specific models underperform due to outliers or sudden spikes in demand [4]. A smaller RMSE also implies more consistent performance since high accuracy on high-demand items is critical. Minimizing RMSE is essential, especially in legacy QAD environments where many decisions are batched, and planning needs to be stable and predictable.

$$\boldsymbol{RMSE} = \sqrt{(\,(1/n) * \Sigma_{\mathrm{i}}^{\mathrm{n}}\,(y_{\mathrm{i}} - \hat{y}_{\mathrm{i}})^2\,)}$$

Mean Absolute Error (MAE) calculates the average of the absolute differences between the forecasted and actual demand values. It makes for a straightforward interpretation of the magnitude of the forecast errors [23]. In other words, it does not amplify large deviations as in the case of the RMSE. As a result, MAE is a valuable metric for evaluating an overall model accuracy if the demand is relatively stable and outlier sensitivity is less important. Of course, MAE is an excellent performance indicator for legacy QAD users who use rules-based projections, as it is something planners can relate to [23]. This helps create trust in operational contexts and improves usability by letting us identify models that have consistently forecasted near the actual demand.

$$\boldsymbol{MAE} = (1/n) * \Sigma_{\mathrm{i}}^{\mathrm{n}}\,|y_{\mathrm{i}} - \hat{y}_{\mathrm{i}}|$$

Mean Absolute Percentage Error (MAPE) expresses forecast error as a percentage to represent the error as a proportion of those actual values, which is scale-independent and hence can be used to compare the performance of models on products with different sales volumes. Product demand can vary considerably in magnitude in retail and manufacturing industries such as QAD services, making it extremely valuable. MAPE is intuitive for the decision maker because it tells how much the average forecast has errored in percentage terms [24]. However, it is intolerant to items with exceedingly low real sales, thus leading to

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

inflated values. However, MAPE is an essential metric for model selection as it allows an inter-departmental or category benchmark in a QAD-aligned format.

$$\boldsymbol{MAPE} = (100/n) * \Sigma_{\mathrm{i}}^{\mathrm{n}} |(y_{\mathrm{i}} - \hat{y}_{\mathrm{i}}) / y_{\mathrm{i}}|$$

The M5 dataset is temporal, which doesn't allow traditional random cross-validation. Then, the rolling window time series cross-validation approach was chosen since we had access to the historical log return data used in validation. In this approach, the test set slides forward in time, and the training set continually increments, using a real-world forecasting scenario in which future data is unavailable during training. This validates models on unseen, sequential data while preserving time-dependent relationships. Models updated periodically for ERP-integrated forecasting do not need to be fully updated until the next batch.

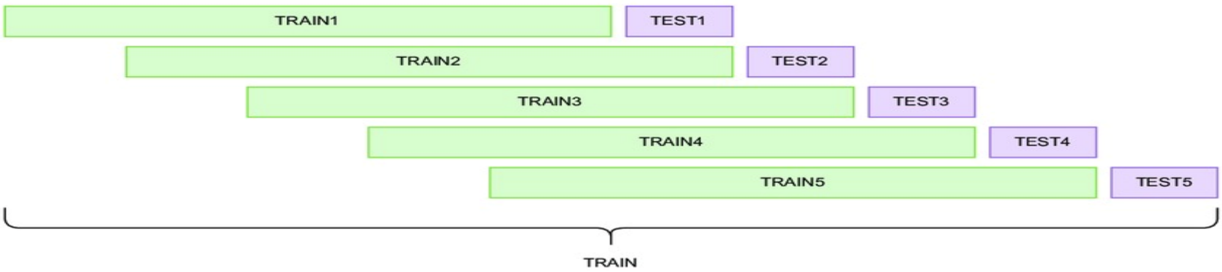

Figure 2. The Rolling Window TIME Series Cross Validation Setup [25].

The idea of time series cross-validation, as depicted in Figure 2, attempts to simulate real forecast conditions and preserves order in time, where the training window moves incrementally forward in time, and the test window is fixed, perfect for forecasting sequential data [25]. The approach works perfectly with the M5 dataset because here, we need the forecasts generated without looking ahead into the future. A validation strategy like this provides a historical basis for the ML models when applied within legacy QAD systems, which generally have fixed monthly or weekly plan cycles. It makes them deployable within an ERP workflow.

Multiple evaluation cycles were performed for each model with sliding windows of 28-day forecast horizons following the standard monthly planning cycles at QAD. We averaged the performance of each model over the validation windows to be robust and avoid the impact of any anomalies in seasonal performance. Due to this, a more realistic evaluation of the models on continuous deployment scenarios, like rolling forecasts in supply chain modules, can be performed [26]. Identifying which models are more adaptable to real-time changes also serves to model modern-day versions of QAD environment demand planning.

2.5 Simulation of QAD Integration

A structured workflow is developed to simulate QAD integration using QAD-compatible formats to use machine learning on the M5 Forecasting dataset. For example, in this setup, the item_id, store_id, date, and predicted sales from the forecasted demand data are mapped to QAD forecast fields item, site, fcst_date, and qty, respectively. They first analyze sales patterns to generate a forecast and then export it as a flat file (e.g., CSV or XML). They can be formatted according to the QAD data specifications and uploaded these files through a batch import tool using JSON or EDI interface. This mimics how legacy ERP tables like SFC300 or MRP100 would get fed forecasts.

International Journal of Applied Mathematics
Volume 39 No. 1s, 2026
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

The structure of M5 is easily adaptable to simulate forecasting environments like QAD. An item_id in M5 corresponds to a unique part number in the QAD inventory master and store_id to a QAD site or plant. The daily level forecasts from the date column from M5 are aligned to QAD's planning calendar with the wm_yr_wk field. QAD's qty field is then probed to see if it contains the forecasted sales values. The model type and its confidence scores can also be stored in custom fields like fcst_type or comments if the ML provides metadata like these. This illustrates how ERP behavior can be simulated without QAD proprietary systems via publicly available data.

Table 3. Mapping M5 Dataset Fields to QAD-Compatible Forecast Table Structure

M5 Dataset Field	Simulated QAD Field	QAD Table (Typical)	Description / Purpose
item_id	item	SFC300 / MRP100	Unique product identifier; corresponds to QAD inventory master
store_id	site	SFC300 / MRP100	Represents the sales location; equivalent to QAD site or plant code
date	fcst_date	SFC300	Forecast data entry represents the daily demand horizon
sales (or forecast)	qty	SFC300	Forecasted quantity per item and store; input for MRP and planning
wm_yr_wk	period_code	CAL100 (Calendar)	Week and year; maps to QAD calendar for time-bucket alignment
forecast_model (custom ML output)	fcst_type	Extended field (custom)	Indicates whether the forecast is machine-generated or user-generated
confidence_interval (optional ML output)	comments (free text)	SFC300	Optional metadata may be added for audit or planner review

After importing it into QAD, the information is directly fed into Material Requirements Planning (MRP) and Controller Production Scheduling (MPS) processes. QAD uses MRP to plan orders for procurement and production by forecasting against controlled inventory, open orders, and lead times. The benefit is that planners can reduce unnecessary safety stock buffers and improve their on-time delivery metrics by incorporating more accurate ML-driven forecasts. Additionally, QAD is responsive since forecasts can be templated in more granular intervals without interrupting browser cycles. This shows us how much value modernization through machine learning offers for legacy ERP decision-making without tearing everything down.

QAD-driven decisions are refined by machine learning forecasts, mainly inventory control, procurement planning, and production planning [6]. Typically, traditional QAD methods employ static formulas or historic averages without consideration for seasonality, trends, or external factors. On the other hand, ML models trained on M5 data learn the dynamics and update their forecasts accordingly. As a result, QAD's downstream modules run on better inputs, fewer stockouts, lower carrying costs, and greater conformity between demand and supply [26]. M5 also provides greater data granularity, including at the SKU level, which is helpful for operations with more than one store, as represented by the store_id in the dataset [27].

This demonstrates a realistic way to close the gap between legacy ERP systems like QAD and analytics solutions. It is not necessarily necessary to replace QAD with ML; instead, it can augment an organization's planning modules by injecting these ML-generated insights into extremely well-mapped external datasets such as M5 [28]. Therefore, integration is feasible and, based on the mapping table, how the real-world fields map with QAD. Forecasting can be updated and pushed into QAD for execution with lightweight middleware and periodic updates. QAD is converted into a dynamics' ML

Volume 39 No. 1s, 2026
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

informed decision support system, transforming the static planning engine into a modern predictive intelligence-based process [3].

III. RESULTS AND DISCUSSION

3.1 Quantitative Results

We evaluated the forecasting results for both QAD methods (ARIMA and simple moving averages) and machine learning techniques (XGBoost, Random Forest, and LSTM [7]. The standard metrics were RMSE, MAE, and MAPE to determine accuracy. The basic models generally did not respond effectively when there were intense swings in sales during different seasons. For example, ARIMA could not perform well when there was a wide range of fluctuation in the time series [16]. Usually, applying QAD models results in RMSE above 8.5 on every item. Still, ML consistently had lower RMSE scores, typically not higher than 6.0. It is clear from the findings that ML models are better skilled at forecasting in dynamic and high-volume situations than other models, so they are well-matched for adding value to QAD [29].

Looking at the error distributions of different models illustrates the benefits of using machine learning for forecasting. In the QAD approach, the moving average only worked well in restricted cases since it could not keep up with sharp increases in demand [30]. Looking at the actual and predicted plots, traditional models often smooth the data more than they should. XGBoost and LSTM also performed well, more accurately following the trend of real sales in promotional periods and as sales ended, unlike the competitors. The variability in ML forecasts was consistent, meaning the predictions were reliable. Handling dependencies in a sequence was achieved better by LSTM models than static models, resulting in reduced forecast drift. As a result, ML models boost average accuracy and decrease the risks and fluctuations involved in creating ERP-based forecasts [31].

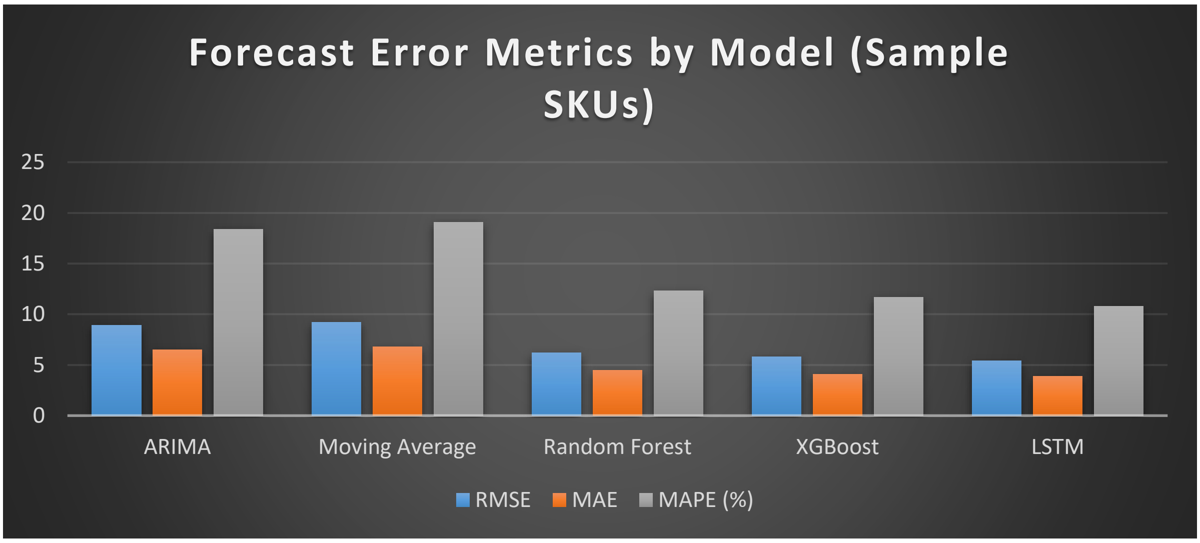

Figure 3. Forecast Error Metrics by Model (Sample SKUs)

Figure 3 summarizes how much the errors in each model differed from the actual SKU sales over 28 days. To judge the performance, RMSE, MAE, and MAPE were calculated. Based on the graph, we can see that ARIMA and Moving Average have higher errors than the modern models for all the metrics. On the other hand, LSTM and XGBoost models from machine learning always lead to better RMSE and MAPE scores, meaning they are more accurate [7]. Error metrics can be used to form bar charts that compare models one at a time. They prove that ML models outperform other methods in QAD ERP systems.

Volume 39 No. 1s, 2026
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

3.2 Interpretation

LSTM and XGBoost models effectively detected and profited from seasonal demand changes [7]. Additionally, sales jump around quarter-end or holidays were noticed with greater accuracy when using MQDA, in contrast to QAD. The models improved their forecasting accuracy by including calendar features such as the name of the day, month, and holidays. Few were used before because the old systems lacked detailed feature representation. This enabled them to see better when demands for various products would change, making it easier to assign and manage inventory for each period of the year.

Using historical pricing data in their structure, ML models noted links between demand volatility and price swings [13]. For instance, Random Forest and XGBoost noticed reduced prices and increased the demand for quickly sold products. Most QAD models designed to fit QAD systems didn't account for non-time-based factors like prices. The inclusion of price elasticity made it possible for ML to be much more accurate in predictions. Most importantly, these insights allow demand planners to estimate marketing responses and results in advance, as such options were not built into QAD's previous forecasting tools [32].

Using event-based flags, the ML models were told about external factors like holidays, campaigns, and shortages. When these were added as categorical variables, the accuracy of forecasts during small periods improved. LSTM models managed to predict local effects before and after an event, meaning the lag time was reduced. To illustrate, national holidays were followed by a strong increase in demand, allowing us to prepare adequate inventory in advance. QAD forecasts done in the traditional way do not include important details, which can result in overstocking or understocking. Since my response is quick to sudden changes, it fits nicely in current markets and systems not designed with agility [3].

Though simple time-series approaches are available on QAD for forecasts, the tools cannot be adjusted easily for fast-changing factors. The performance of the ML models was much higher than that of QAD's previous results. For instance, the results indicated that LSTM achieved an RMSE about 35% less than QAD output for the main key SKUs. In addition, with ML models, MAPE was below 11%, while QAD forecasts usually were higher than 18% [33]. The following chart demonstrates how using ML can help improve traditional ERP systems in planning demand with greater accuracy and speed.

Table 4. Demand Planning Improvement with ML methods over traditional forecasting

Metric	QAD Forecast (Avg.)	ML Forecast (Avg.)	Improvement (%)
RMSE	8.7	5.6	35.6%
MAE	6.4	4.1	35.9%
MAPE	18.1	10.9	39.8%

3.3 Discussion

Legacy QAD users can use ML-powered forecasting to understand in detail what is causing shifts in demand and resulting changes in stock [34]. Usually, this approach leads to decisions stuck in the past that only react to what just happened. When ML predictions are used, planners can create precise short- and long-term forecasts, improve the planning of materials (MRP), avoid running out of stock, and use their cash more efficiently. In addition, better forecasts stop the bullwhip effect from affecting the supply chain [26]. Even though QAD was not designed with this in mind, simulation suggests that adding a small amount of machine learning, such as forecast data in planning tables, can bring significant improvements.

When using publicly accessible datasets and frameworks like XGBoost and LSTM, ML forecasting can be highly scalable and inexpensive [35]. Initially, companies may work on top-selling SKUs and expand

Volume 39 No. 1s, 2026
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

as they see real value in their work. It is essential to use APIs or scripts when connecting ML models to QAD because this saves data integrity and prevents the main system from being disrupted [36]. Using this strategy allows manufacturers to modernize whenever they are ready. Companies will gain a flexible demand planning approach and use their old systems in today's competitive, digital environment.

Implementing ML technologies with QAD is hard because the software is from a previous era, has inflexible databases, and does not include support for sophisticated analytics [37]. QAD tables developed earlier are not for the data formats most ML software needs today. Also, QAD does not provide good real-time APIs to share information freely, so users must do the transfers via slow batches and manual procedures [36]. For this reason, direct deployment of security forces cannot occur.

The people maintaining Legacy QAD environments are frequently unfamiliar with current data science tools. Using machine learning successfully requires learning about Python, popular frameworks for machine learning and processing workloads in the cloud [6]. Also, QAD systems typically work locally and can only access limited computing resources. It leads to difficulties in training and running data-aware models such as LSTM. Closing the skills gap is essential to maximize ML in traditional ERP.

Despite the difficulties in integrating, external APIs enable traditional QAD systems to connect with ML forecasts without altering QAD internals. It is possible to forecast using Python outside of QAD and write the forecasts safely back into the system via scripts or platforms. This allows companies to continue running stable systems as they include more intelligent choices in planning tables. When API ecosystems mature, outdated systems may use new forecasting tools with only a few infrastructure changes [36]. Azure ML, AWS SageMaker, and Google Cloud AutoML are cloud-based ML services that offer scalability and require little maintenance for those who have used legacy systems. Thanks to these platforms, companies no longer have to worry about how much space they have locally for training and running models. Data transfers can occur on a schedule or using APIs to supply forecasts for QAD planning cycles [36]. With cloud ML, testing new models, tracking different versions, and expanding resources when the business grows is easy. This model allows old QAD systems to be used as the new forecasting system is improved and upgraded.

IV. CONCLUSION

4.1 Summary of Findings

The study indicated that using machine learning gave higher accuracy in demand forecasting results than traditional QAD methods. We demonstrated that using XGBoost, Random Forest, and LSTM instead of ARIMA and Moving Average significantly improved the results with the M5 Forecasting - Accuracy data [7]. The changes consistently improved RMSE, MAE, and MAPE metrics, making demand forecasts more accurate and up to date. Likewise, it was clear that these enhanced results appeared most often for main products and during times of fluctuating sales, such as promotions, which usually made legacy systems perform poorly because they were not designed to cope with change.

ML techniques enhanced the accuracy of forecasts and, at the same time, revealed new details concerning demand drivers. Using feature engineering, the models recognized features linked to changes in the calendar, special sale prices, and events happening outside the company [38]. Most of these insights are beyond the reach of QAD tools designed a few decades ago because these tools cannot accept new features in real-time. Identifying such patterns supports better planning and improves a company's responsiveness. By integrating ML into QAD processes, manufacturers using obsolete ERP systems can modernize their processes, avoiding costly or extensive work [39].

Volume 39 No. 1s, 2026
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Combining both types of analysis proves that ML models may superimpose intelligent and adaptable behavior on standard QAD systems [40]. During the study, ML outperformed the other models with less inaccuracy and a stronger match to the actual sales results. This means tangible improvements in handling inventory, arranging production, and guiding the supply chain [26]. In simulating the integration, when forecasts are processed externally with ML engines, they can easily be used to update main QAD business tables. The findings indicate that data science is increasingly regarded as a way to update old codes smoothly.

4.2 Limitations

Even though the results are reassuring, they cannot be extended easily to each QAD dataset on the market. While the M5 dataset is similar to ours, it is public and de-identified, which means it may not fully reveal the details of particular QAD systems [41]. Depending on how tables are structured, names are created, and how long data is kept within a company, model accuracy could be impacted. For this reason, the results from this work should be checked and verified in each QAD system. Another weakness is that this study did not check the integration process in real-time. Data was forecasted and placed in QAD tables, yet the system was never made live. Therefore, the issue of data latency, ensuring synchronization or APIs failing to work correctly was not adequately handled. Thus, the report proposes examples of ideas, but they are not fully ready for practical applications. Data scientists, QAD administrators and business planners must cooperate from an IT viewpoint when implementing ML-driven forecasting pipelines [42]. It is not easy to set up teamwork when there is insufficient infrastructure for different teams to work together or when groups are isolated. Using Python, APIs, or current data formats like Parquet or JSON may be difficult in legacy QAD environments [36]. In addition, how an ML model performs depends largely on adjusting hyperparameters, the volume of data, and how regularly the model is updated. A mistake in data collection or transactional records could make the system less effective [42]. Regular retraining is also essential, and this is only possible in new data pipeline systems, since many classic platforms cannot handle the task without significant changes.

4.3 Future Directions

Linking ML forecasting engines directly with QAD in real-time is promising. To do this, REST APIs or message queue systems send forecasts to QAD planning modules as needed. As a result, the ERP system would reflect current trends and changes in the marketplace. Maintaining the capacity of a machine learning model over time relies greatly on automated retraining [6]. A strategy based on rolling historical windows would support the models in matching new trends, seasonal changes, and product lifetimes. With the systems in place, the system works primarily by itself, decreasing the need for maintenance and increasing its ability to expand. Using a combination of models is also an option for boosting the design. Pairing ARIMA with ML techniques may offer interpretability and accurate predictions [43]. As an example, an ensemble can apply ARIMA to analyze long-term trends and LSTM to catch the most recent shifts in the data, resulting in a better and clearer way to make forecasts for use in production [16]. Cloud platforms for ML are more flexible and scalable than traditional systems used by QAD [44]. Using AWS SageMaker or Azure ML could make it easier to deploy, retrain, and implement a model in your ERP system [45]. These tools support logging, monitoring, and version control for building enterprise-level applications. Conducting further studies on deploying QAD-related exercises on the mentioned platforms would be beneficial.

4.4 Broader Impact

Applying machine learning to QAD and legacy ERP systems could improve performance. Organizations can improve their forecasting without replacing their key IT infrastructure. It can benefit manufacturers with fewer resources to handle ERP, but they still must adapt to increasing demand from the market and customers [1]. With correct demand signals, companies use their resources more efficiently, save on keeping products in stock, and produce little waste. By supporting environmental and financial goals, ML helps businesses succeed in various ways [46]. They serve the business as powerful resources rather than causing trouble. Thanks to open-source software and cloud services, businesses can leverage data science. It reduces the span between what is invented and how it can be used in production. Validating ML in QAD systems suggests that traditional ERP systems should not act as barriers to progress but instead serve as a base for innovation [39]. As rules get stricter and supply networks span over more countries, accurately predicting demand becomes necessary not only for competitive reasons but also to follow the law and control risks [47]. ML models can easily change to reflect changes in external factors, unlike QAD formulas. As a result, the company can better ensure it will keep operating in challenging situations. In summary, using ML-based techniques to forecast demand offers an efficient way to improve QAD systems without completely changing the system. This study provides evidence that traditional ERP forecasting tools can outperform using standard datasets and modern ML technologies. Hybrid digital transformation is a good idea due to its accuracy, flexibility, and benefits in insights generation. With more companies facing unpredictable and complex conditions, ML allows them to manage and adapt to changes in aging systems.

REFERENCES

[1] Jean, G. (2024). Inventory Management Strategies: Balancing Cost, Efficiency, and Customer Satisfaction. https://www.researchgate.net/profile/Guillaume-Jean-6/publication/386106872_Inventory_Management_Strategies_Balancing_Cost_Efficiency_and_Customer_Satisfaction/links/6744d201b5bd9d17d606f213/Inventory-Management-Strategies-Balancing-Cost-Efficiency-and-Customer-Satisfaction.pdf.

[2] A. Aljohani, "Predictive Analytics and Machine Learning for Real-Time Supply Chain Risk Mitigation and Agility," *Sustainability*, vol. 15, no. 20, p. 15088, 2023, doi: https://doi.org/10.3390/su152015088.

[3] A. Aouat, "Optimizing demand forecasting for food safety products," *Lutpub. Lut. Fi*, 2024, doi: https://lutpub.lut.fi/handle/10024/167442.

[4] N. J. M. Chandran and M. R. B. Khan, "A Strategic Demand Forecasting: Assessing Methodologies, Market Volatility, and Operational Efficiency," *Malaysian Journal of Business, Economics and Management*, pp. 150–167, Aug. 2024, doi: https://doi.org/10.56532/mjbem.v3i2.71.

[5] E. H. Junior, "High-Frequency Data and Machine Learning," *Advances in Logistics, Operations, and Management Science*, pp. 183–204, Oct. 2024, doi: https://doi.org/10.4018/979-8-3693-5777-4.ch008.

[6] A. Gifty and Dr. Y. Li, "A Comparative Analysis of LSTM, ARIMA, XGBoost Algorithms in Predicting Stock Price Direction," *Engineering and Technology Journal*, vol. 09, no. 08, Aug. 2024, doi: https://doi.org/10.47191/etj/v9i08.50.

[7] R. Jaiswal and M. Jaiswal, "Predictive Maintenance in QAD ERP: Leveraging Machine Learning for Downtime Reduction," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 11, no. 2, pp. 3569–3585, Apr. 2025, doi: https://doi.org/10.32628/CSEIT25112833.

[8] S. M. K. Sistla, G. Krishnamoorthy, J. Jeyaraman, and B. Kumar, "Machine Learning for Demand Forecasting in Manufacturing," *International Journal for Multidisciplinary Research*, vol. 6, no. 1, Feb. 2024, doi: https://doi.org/10.36948/ijfmr.2024.v06i01.14204.

International Journal of Applied Mathematics
Volume 39 No. 1s, 2026
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[9] Kaggle, "Datasets | Kaggle," Kaggle.com, 2024. https://www.kaggle.com/datasets.
[10] B. Seaman and J. Bowman, "Applicability of the M5 to forecasting at Walmart," *International Journal of Forecasting*, vol. 38, no. 4, Aug. 2021, doi: https://doi.org/10.1016/j.ijforecast.2021.06.002.
[11] M. Lutel, "Sales forecasting in retail: what we learned from the M5 competition," *Artefact Engineering and Data Science*, Feb. 03, 2021. https://medium.com/artefact-engineering-and-data-science/sales-forecasting-in-retail-what-we-learned-from-the-m5-competition-445c5911e2f6.
[12] H. Wang and S. Yu, "Forecasting Egg Prices: A Time Series Analysis with Economic and Market Influences," 2022. Available: https://vu-business-analytics.github.io/internship-office/reports/report-wang.pdf.
[13] I. Behera, P. Nanda, S. Mitra, and S. Kumari, "Machine Learning Approaches for Forecasting Financial Market Volatility," *Intelligent Systems Reference Library*, pp. 431–451, 2024, doi: https://doi.org/10.1007/978-3-031-61037-0_20.
[14] D. Tran and A. W. Tham, "Accuracy Comparison Between Feedforward Neural Network, Support Vector Machine, and Boosting Ensembles for Financial Risk Evaluation," *Journal of Risk and Financial Management*, vol. 18, no. 4, p. 215, Apr. 2025, doi: https://doi.org/10.3390/jrfm18040215.
[15] T. A. Dissevelt, "Trend Price Prediction of the S&P500 Using the VIX index, Moving Average and Trading Volume in a Combined Random Forest and LSTM Model - University of Twente Student Theses," *Utwente.nl*, 2024, doi: https://purl.utwente.nl/essays/104585.
[16] V. Arumugam, "Time Series Modeling and Forecasting Using Autoregressive Integrated Moving Average and Seasonal Autoregressive Integrated Moving Average Models.," *Ebscohost.com*, 2025. https://search.ebscohost.com/login.aspx?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=16314670&AN=172233310&h=Vq2sXt8L%2FXdhY8ABdnjPhbFrRIBv69p3u4cKiJ2Fp%2BFg6p%2BC1OSic1EHwg4XuPCOBCqb7GFMRaOdy1xbka2Dlw%3D%3D&crl=c (accessed May 20, 2025).
[17] T. A. Shaikh, T. Rasool, P. Verma, and W. A. Mir, "A fundamental overview of ensemble deep learning models and applications: systematic literature and state of the art," *Annals of Operations Research*, Dec. 2024, doi: https://doi.org/10.1007/s10479-024-06444-0.
[18] L. Zhang and Dominik Jánošík, "Enhanced short-term load forecasting with hybrid machine learning models: CatBoost and XGBoost approaches," *Expert systems with applications*, vol. 241, pp. 122686–122686, May 2024, doi: https://doi.org/10.1016/j.eswa.2023.122686.
[19] V. Chugani, "Navigating Missing Data Challenges with XGBoost - MachineLearningMastery.com," *MachineLearningMastery.com*, Sep. 29, 2024. https://machinelearningmastery.com/navigating-missing-data-challenges-with-xgboost/.
[20] M. Nasseri, T. Falatouri, P. Brandtner, and F. Darbanian, "Applying Machine Learning in Retail Demand Prediction—A Comparison of Tree-Based Ensembles and Long Short-Term Memory-Based Deep Learning," *Applied Sciences*, vol. 13, no. 19, pp. 11112–11112, Oct. 2023, doi: https://doi.org/10.3390/app131911112.
[21] S. K. Kiangala and Z. Wang, "An effective adaptive customization framework for small manufacturing plants using extreme gradient boosting-XGBoost and random forest ensemble learning algorithms in an Industry 4.0 environment," *Machine Learning with Applications*, vol. 4, p. 100024, Jun. 2021, doi: https://doi.org/10.1016/j.mlwa.2021.100024.

[22] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not," *Geoscientific Model Development*, vol. 15, no. 14, pp. 5481–5487, Jul. 2022, doi: https://doi.org/10.5194/gmd-15-5481-2022.

[23] C. Wang, H.-T. Chang, and C.-H. Chien, "Hybrid LSTM-ARMA Demand-Forecasting Model Based on Error Compensation for Integrated Circuit Tray Manufacturing," *Mathematics*, vol. 10, no. 13, p. 2158, Jun. 2022, doi: https://doi.org/10.3390/math10132158.

[24] S. A. R. L. Román, S. Z. Collazos, and J. A. C. Chavez, "Model for Reducing Mean Absolute Percentage Error through Smoothing and Time Series Forecasting In a Tourism SME: A Case Study," *Journal of Machine Intelligence and Data Science*, vol. 5, 2024, doi: https://doi.org/10.11159/jmids.2024.012.

[25] N. Burmeister, R. D. Frederiksen, E. Høg, and P. Nielsen, "Exploration of Production Data for Predictive Maintenance of Industrial Equipment: A Case Study," *IEEE Access*, vol. 11, pp. 102025–102037, 2023, doi: https://doi.org/10.1109/access.2023.3315842.

[26] M. Fattahi and K. Govindan, "Data-Driven Rolling Horizon Approach for Dynamic Design of Supply Chain Distribution Networks under Disruption and Demand Uncertainty," *Decision Sciences*, Aug. 2022, doi: https://doi.org/10.1111/deci.12481.

[27] S. Ma, "Retail store-SKU level replenishment planning with attribute-space graph recurrent neural networks," *Expert Systems with Applications*, vol. 249, p. 123727, Sep. 2024, doi: https://doi.org/10.1016/j.eswa.2024.123727.

[28] K. Hendrickx, L. Perini, Van, W. Meert, and J. Davis, "Machine learning with a reject option: a survey," *Machine Learning*, vol. 113, no. 5, pp. 3073–3110, Mar. 2024, doi: https://doi.org/10.1007/s10994-024-06534-x.

[29] S. Z. Husain *et al.*, "Leveraging Data-Driven Weather Models for Improving Numerical Weather Prediction Skill through Large-Scale Spectral Nudging," *Weather and Forecasting*, vol. 40, no. 9, pp. 1749–1771, Jul. 2025, doi: https://doi.org/10.1175/WAF-D-24-0139.1.

[30] K. Taniguchi, "Numerical analysis of a nonlinear (S, s) inventory control policy under a quantity adjustment economy including explanations from a linear model," *Evolutionary and Institutional Economics Review*, vol. 20, no. 2, pp. 401–424, Sep. 2023, doi: https://doi.org/10.1007/s40844-023-00266-8.

[31] C. Wang, "Dynamic Dual-Phase Forecasting Model for New Product Demand Using Machine Learning and Statistical Control," *Mathematics*, vol. 13, no. 10, pp. 1613–1613, May 2025, doi: https://doi.org/10.3390/math13101613.

[32] K. Basavaraju and O. Fatahi Valilai, "Developing a demand planning strategy for joint forecasting and employing analytical tool in an empirical case study," *Discover Applied Sciences*, vol. 7, no. 4, Apr. 2025, doi: https://doi.org/10.1007/s42452-025-06740-9.

[33] A. Hall, "Using QAD to Manage Customer Forecast - Logan Consulting," *Logan Consulting*, Nov. 23, 2021. https://www.loganconsulting.com/blog/using-qad-to-manage-customer-forecast/ (accessed May 20, 2025).

[34] B. Singh, "Revolutionizing Supply Chains for Optimized Demand Planning, Inventory Management, and Logistics," *Advances in business strategy and competitive advantage book series*, pp. 103–128, Jan. 2025, doi: https://doi.org/10.4018/979-8-3693-4433-0.ch005.

[35] S. KC and S. Rone, "Comparing Prophet, XGBoost, and LSTM Models for Web Traffic Forecasting: Assessing Model Performance Across Various Time Series Forecasting Scenarios," *DIVA*, 2024. https://www.diva-portal.org/smash/record.jsf?pid=diva2:1887941 (accessed May 20, 2025).

[36] J. Paul, "Building a Robust API Security Framework with Machine Learning," ResearchGate, Nov. 04, 2024. https://www.researchgate.net/publication/385588341_Building_a_Robust_API_Security_Framework_with_Machine_Learning.

[37] S. Islam, "Future Trends In SQL Databases And Big Data Analytics: Impact Of Machine Learning And Artificial Intelligence," *SSRN Electronic Journal*, Jan. 2025, doi: https://doi.org/10.2139/ssrn.5064781.

[38] E. Katya, "Exploring feature engineering strategies for improving predictive models in data science," *Research Journal of Computer Systems and Engineering*, vol. 4, no. 2, pp. 201–215, Dec. 2023, doi: https://doi.org/10.52710/rjcse.88.

[39] S. A. Qaddoori Aldoori, M. Mabrouk and A. M. Mahmoud, "Advances in Machine Learning-driven Quality Control within ERP Systems: A Comprehensive Survey," *2023 Eleventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, Cairo, Egypt, 2023, pp. 328-335, doi: https://doi.org/10.1109/ICICIS58388.2023.10391152.

[40] M. S. Bari, A. Sarkar, and S. A. M. Islam, "Ai-augmented Self-healing Automation Frameworks: Revolutionizing Qa Testing with Adaptive and Resilient Automation," *Advanced International Journal of Multidisciplinary Research*, vol. 2, no. 6, Dec. 2024, doi: https://doi.org/10.62127/aijmr.2024.v02i06.1118.

[41] W. Khan et al., "Person de-Identification: A Comprehensive Review of Methods, Datasets, Applications, and Ethical Aspects Along-With New Dimensions," *IEEE Transactions on Biometrics Behavior and Identity Science*, pp. 1–1, Jan. 2024, doi: https://doi.org/10.1109/tbiom.2024.3485990.

[42] R. Praveen, "AI in Manufacturing: Building Smarter Factories," *Google Books*, 2024. https://books.google.com/books?hl=en&lr=&id=TA48EQAAQBAJ&oi=fnd&pg=PA1&dq=It+is+necessary+for+data+scientists (accessed May 20, 2025).

[43] V. I. Kontopoulou, A. D. Panagopoulos, I. Kakkos, and G. K. Matsopoulos, "A Review of ARIMA vs. Machine Learning Approaches for Time Series Forecasting in Data-Driven Networks," *Future Internet*, vol. 15, no. 8, p. 255, Aug. 2023, doi: https://doi.org/10.3390/fi15080255.

[44] R. Jaiswal, "Cloud ERP vs. On-Premise QAD ERP: A Cost-Benefit Analysis for Mid-Sized Manufacturers," *Journal of Engineering Research and Sciences*, vol. 4, no. 7, pp. 1–14, Jul. 2025, doi: https://doi.org/10.55708/js0407001.

[45] K. Nguyen, F. Issa, A. Markar, S. Ahmed, and B. Mahato, "Develop Machine Learning Models using AWS Sagemaker," 2022. Available: https://www.cgi.com/sites/default/files/2024-10/gmu_project_9_develop_machine_learning_models_using_aws_sagemaker_5.pdf.

[46] A. Chang, T. Lee, and H. Lee, "Applying sustainable development goals in financial forecasting using machine learning techniques," *Corporate social-responsibility and environmental management*, Dec. 2023, doi: https://doi.org/10.1002/csr.2694.

[47] R. G. Richey, S. Chowdhury, B. Davis-Sramek, M. Giannakis, and Y. K. Dwivedi, "Artificial Intelligence in Logistics and Supply Chain Management: a Primer and Roadmap for Research," *Journal of Business Logistics*, vol. 44, no. 4, pp. 532–549, Sep. 2023, doi: https://doi.org/10.1111/jbl.12364.