Intelligent Methods in Engineering Sciences 5(2): 69-82, 2026

IMIENS

INTELLIGENT METHODS
IN ENGINEERING SCIENCES

https://www.imiens.org

International
Open Access

August, 2026

e-ISSN 2979-9236

Research Article

https://doi.org/10.58190/imiens.2026.176

Data-Driven Prediction of Processing Times and Delays Across Supply Chain Operational Stages Using Machine Learning

Muhammed Yıldıray SURMEN [a], ***Oya KILCI*** [b], ***Selime Sinem BAHAR*** [c], ***Murat KOKLU*** [d,*]

[a] *Operational Excellence Department, Alisan Inc, İstanbul, Türkiye*
[b] *KTO Karatay University, Vocational School of Trade and Industry, Department of Mechatronics, Konya, Türkiye*
[c] *Information Technologies and R&D Department, Alisan Inc, İstanbul, Türkiye*
[d] *Selcuk University, Technology Faculty, Department of Computer Engineering, Konya, Türkiye*

ARTICLE INFO

Article history:
Received 8 July 2026
Accepted 28 August 2026

Keywords:
Data Entry Time Prediction,
Gradient Boosting,
Machine Learning,
Operational Decision Support,
Supply Chain Operations,
Temporal Data Splitting

ABSTRACT

This study proposes an ensemble learning framework to predict personnel data entry durations across six operational stages using transaction records collected from an operational supply chain information system. To accurately represent operational performance, processing times were calculated in working minutes by excluding nonworking periods, including nights and weekends. The dataset comprised 110,695 unique shipment records collected between 2022 and 2026 and underwent stage-specific data quality assessment and preprocessing. A median-based baseline predictor and three ensemble learning algorithms, HistGradientBoosting, XGBoost, and LightGBM, were evaluated for regression tasks. All models were trained on a log-transformed target (log1p), and predictions were back-transformed to working minutes using the inverse transformation (expm1). To prevent data leakage and better reflect real-world deployment, a year-based temporal split was adopted, with 2026 records reserved for testing. Model performance was evaluated using mean absolute error, root mean square error, coefficient of determination, accuracy, precision, recall, and F1-score. A single regression model was trained per stage, and binary labels (normal/slow) were derived by thresholding the predicted durations at a stage-specific cut-off optimized on the validation set. The results showed that predictive performance varied substantially across operational stages, with LightGBM achieving the best overall performance for the Loading Arrival stage, while lower performance in the Transport Entry and unloading stages suggested the influence of external operational factors not represented in the available data. For the Loading Arrival stage, LightGBM achieved an MAE of 285.989 working minutes and an R^2 of 0.371 on the test set. Overall, the findings demonstrate that ensemble learning provides an effective and computationally efficient approach for predicting personnel data entry durations and supports real-time decision-making, personnel performance monitoring, and process optimization in data-driven supply chain operations.

1. INTRODUCTION

Personnel productivity is a critical determinant of organizational performance in supply chain and logistics operations. These operations consist of multiple interdependent stages, including shipment preparation, transportation planning, loading, and unloading. The timely completion of each stage is essential for maintaining operational continuity. Delays at any stage may propagate throughout the workflow, reducing operational capacity, increasing costs, and negatively affecting delivery performance, inventory turnover, and customer satisfaction.

Traditional productivity management primarily relies on historical reports, periodic inspections, and managerial experience. Such approaches identify operational problems only after they occur, limiting the ability to implement proactive interventions and optimize resource allocation. Furthermore, observation-based performance evaluations may vary across personnel, shifts, and facilities, reducing the consistency and objectivity of operational performance assessment.

Recent advances in artificial intelligence and machine learning have created new opportunities for predictive analytics in manufacturing and logistics. Machine learning techniques have been successfully applied to forecasting, anomaly detection, workforce management, and resource optimization. Among these techniques, ensemble learning algorithms have demonstrated strong predictive capability for complex operational datasets characterized by

* **Corresponding Author:** mkoklu@selcuk.edu.tr

nonlinear relationships, heterogeneous features, and noisy observations. HistGradientBoosting, XGBoost, and LightGBM have attracted considerable attention because of their high predictive accuracy, computational efficiency, and ability to model large-scale structured data.

Despite these advances, several challenges remain in predicting personnel performance in real-world supply chain operations. First, most existing studies focus on a single operational stage or a single performance indicator, although different stages exhibit distinct operational characteristics and variability. Second, many predictive models are evaluated using randomly partitioned datasets, which may overestimate predictive performance because future operational conditions are not adequately represented. Third, ensemble learning algorithms are often investigated independently, making objective comparisons under identical experimental conditions difficult.

The objective of this study is to develop and evaluate a unified machine learning framework for predicting personnel data entry durations across six operational stages of a real-world supply chain organization. The proposed framework formulates the prediction task as both a regression problem, estimating actual processing durations, and a binary classification problem, distinguishing normal and slow records. HistGradientBoosting, XGBoost, and LightGBM are compared with a median-based baseline using a common preprocessing pipeline, identical feature sets, and a year-based temporal data split to ensure a fair and realistic evaluation. In addition, processing durations are calculated using working minutes by excluding nonworking periods, thereby providing a more representative measure of operational performance.

The proposed framework is intended to support predictive workforce analytics by enabling early identification of potential operational delays and providing objective decision support for personnel performance evaluation and resource planning. By integrating multiple operational stages within a consistent experimental framework, this study contributes to the development of practical machine learning applications for data-driven supply chain management.

1.1. Literature Review

Employee productivity has become an important research topic in recent years. Organizations seek to improve efficiency, reduce operational costs, and optimize resource utilization. Traditional productivity assessment methods mainly rely on supervisor observations, performance reviews, and task completion rates. Although these methods are widely used, they often provide retrospective evaluations and limited decision support. Machine learning offers a more objective and data-driven alternative.

Several studies have applied machine learning techniques to employee performance prediction. Lather, et al. [1] used supervised learning algorithms to classify employee performance into different categories. Their results showed that Support Vector Machines achieved the best performance among the evaluated models. The study demonstrated that employee performance can be predicted using historical organizational data.

Artificial intelligence has also gained considerable attention in human resource management. Tambe, et al. [2] discussed the opportunities and challenges of applying artificial intelligence in HR processes. The authors highlighted the potential of predictive analytics for employee evaluation, talent management, and workforce planning. Similarly, Palos-Sánchez, et al. [3] reviewed artificial intelligence applications in human resource management and reported a growing interest in machine learning-based decision support systems.

More recently, Uppal, et al. [4] evaluated multiple machine learning algorithms for employee performance prediction. Their study compared Random Forest, XGBoost, LightGBM, AdaBoost, CatBoost, K-Nearest Neighbors, Decision Trees, and Support Vector Machines. The results indicated that ensemble learning algorithms provide highly accurate performance predictions. The study confirmed the effectiveness of machine learning for workforce analytics.

Employee productivity prediction has also been investigated using physiological and behavioral indicators. Awada, et al. [5] proposed a machine learning framework that integrates physiological, behavioral, and psychological features to predict office worker productivity. Their findings showed that XGBoost achieved the highest predictive performance. The authors also reported that psychological variables such as stress, mood, eustress, and distress significantly improve prediction accuracy.

Several studies have focused on psychological state prediction, which is closely related to productivity. Koldijk, et al. [6] developed machine learning models to detect workplace stress using physiological and behavioral signals. Their models achieved high prediction accuracy. Shu, et al. [7] used heart rate signals collected from wearable devices to recognize emotional states. Similarly, Li, et al. [8] investigated perceived stress and demonstrated the importance of distinguishing between positive and negative stress conditions. These studies indicate that human-related factors can provide valuable information for productivity prediction.

Labor productivity management has also been examined from a broader organizational perspective. Orlova [9] proposed a data-driven labor productivity management framework based on statistical analysis and machine learning methods. The study incorporated demographic, economic, social, and health-related variables. The results demonstrated that machine learning

can support managerial decision-making and productivity improvement strategies.

Machine learning applications have expanded rapidly in logistics and supply chain management. Nguyen, et al. [10] reported that data-driven analytics can improve forecasting, planning, and operational decision-making. Kannan, et al. [11] emphasized the importance of big data analytics for logistics optimization and supply chain performance improvement. Likewise, Riahi, et al. [12] showed that artificial intelligence techniques are increasingly used for forecasting, risk assessment, and process optimization in supply chain systems.

Another relevant research area is predictive process monitoring. This field focuses on predicting future process outcomes using historical event data. Van der Aalst, et al. [13] demonstrated that process mining techniques can estimate process completion times from event logs. Maggi, et al. [14] introduced predictive monitoring methods that support runtime process prediction. Later, Verenich, et al. [15] reviewed remaining-time prediction approaches and highlighted their usefulness for operational decision support. Since logistics activities consist of sequential operational stages, predictive process monitoring provides a strong theoretical basis for productivity prediction.

Ensemble learning algorithms have become particularly popular for structured operational datasets. Friedman [16] introduced the Gradient Boosting Machine framework, which forms the foundation of modern boosting algorithms. Chen and Guestrin [17] developed XGBoost to improve predictive performance and computational efficiency. Ke et al. [25] proposed LightGBM, which reduces training time through histogram-based learning and leaf-wise tree growth. More recently, Grinsztajn, et al. [18] showed that tree-based ensemble models frequently outperform deep learning models on tabular datasets. These findings support the use of boosting algorithms for operational prediction problems.

Despite significant progress in workforce analytics and machine learning, several research gaps remain. Most existing studies focus on overall employee performance rather than stage-level operational productivity. Many studies rely on demographic, behavioral, or psychological variables instead of operational process data. Furthermore, productivity prediction is often formulated only as a classification problem. Studies that simultaneously investigate regression and classification tasks are limited. In addition, business-minute-based productivity measurement has received little attention in literature.

To address these limitations, the present study develops a machine learning framework for stage-level personnel productivity prediction in supply chain operations. Six operational stages are modeled separately. Both regression and classification approaches are evaluated. Business-minute-based duration calculations are used to obtain realistic productivity measurements. Finally, a chronological train-validation-test strategy is adopted to provide a realistic assessment of future predictive performance.

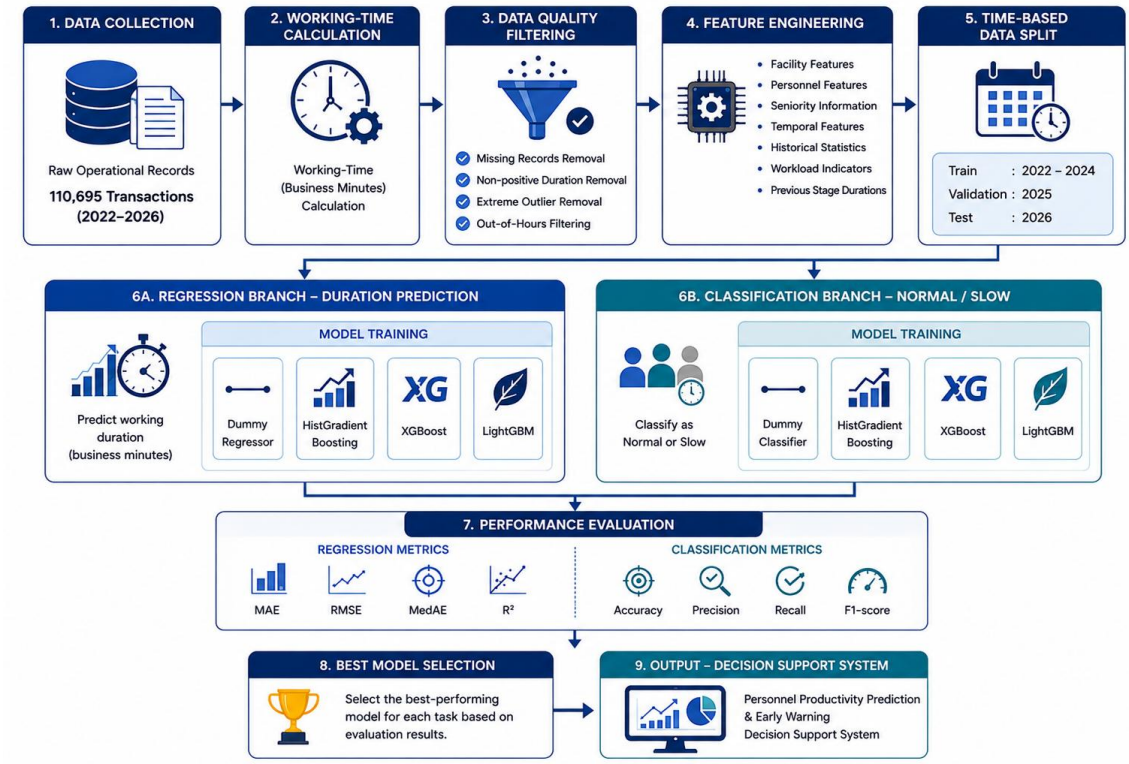

Figure 1. Proposed prediction framework

2. MATERIAL AND METHODS

This section describes the dataset, data preparation procedures, machine learning models, and evaluation metrics used in the study. The overall workflow of the proposed personnel productivity prediction framework is presented in Figure 1. The workflow consists of data collection, data preprocessing, dataset partitioning, model training, and performance evaluation stages.

2.1. Dataset

The dataset used in this study was collected from operational process records of a logistics organization between 2022 and 2026. It contains 110,695 unique shipment records covering different types of logistics operations. Each record includes timestamps from six personnel data entry stages throughout the operational lifecycle of a shipment. This structure enables detailed monitoring of operational timing. It also allows the processing time of each stage to be evaluated as an independent performance indicator.

The target variables in this study are the processing times between the start and end timestamps of each operational stage. These durations are calculated using working time rather than calendar time. Only actual working hours are included in the calculations. Time outside working hours is excluded. Therefore, the target variables are expressed in working minutes. This approach provides a more realistic measure of operational performance.

Working minutes were calculated according to the organization's official working schedule. Working hours were defined as 08:00–18:00 on weekdays and 08:00–14:00 on Saturdays. Sundays and all non-working hours were excluded from the calculations. This approach ensures that processing times reflect only periods when personnel were available to work. As a result, the calculated durations provide a more realistic measure of operational performance.

This approach avoids misleading delays caused by calendar time. For example, a process that starts near the end of the workday on Friday and finishes on Monday morning appears to have a delay of about 48 hours when calendar time is used. In contrast, the working-minute approach counts only the remaining working time on Friday and the actual working time after the start of business on Monday. As a result, the same process is measured as approximately two working hours. This eliminates artificial delays caused by weekends and non-working hours. It also allows the model to learn actual operational performance more accurately.

The productivity metrics examined in this study and their definitions for the six operational stages are presented in detail in Table 1.

Table 1. Dataset productivity metrics.

Metric	Stage	Definition (working minutes)
shipment_set_min	Shipment Entry	Time to create the shipment record after the order is received
transport_set_min	Transport Entry	Time to create the transport record after the shipment record
la_set_min	Loading Arrival (LA)	Time to complete the loading-arrival entry
ld_set_min	Loading Departure (LD)	Time to complete the loading-departure entry
ua_set_min	Unloading Arrival (UA)	Time to complete the unloading-arrival entry
ud_set_min	Unloading Departure (UD)	Time to complete the unloading-departure entry

2.2. Data Preprocessing

Data preprocessing and quality control were performed separately for each operational stage. This approach was adopted because a shipment record may contain missing or invalid information for one stage while remaining complete for others. Therefore, records identified as invalid at a specific stage were excluded only from the analysis of that stage. They were retained for all other valid stages. This strategy maximized the number of usable observations for each target variable and minimized data loss.

For each operational stage, records with missing start or end timestamps were removed because processing times could not be calculated. Records with zero or negative processing times were also excluded. These values were mainly caused by data entry errors or inconsistent timestamps. To reduce the influence of extreme outliers, records with processing times of 6,000 working minutes or more were excluded. This threshold corresponds to approximately 10 working days and was selected to remove exceptional cases outside normal operations. In addition, records outside the organization's defined working hours were excluded. This ensured that the target variables represented only operations performed during actual working hours. It also reduced timing bias caused by non-working periods.

After completing the data quality control and filtering procedures, the final number of records was determined for each operational stage. The number of valid observations used for each target variable is presented in detail in Table 2.

Table 2. Data quality summary by stage.

Stage	Total	Missing	Non-positive	Over cap (≥ 6000)	Out of hours	Usable
Shipment Entry	110,695	0	27,013	193	20,322	63,167
Transport Entry	110,695	0	16,499	1,114	13,664	79,418
Loading Arrival (LA)	110,695	1	22,111	624	23,098	64,861
Loading Departure (LD)	110,695	3	34,072	723	21,409	54,488
Unloading Arrival (UA)	110,695	4	22,861	2,043	21,415	64,372
Unloading Departure (UD)	110,695	4	31,775	2,014	20,343	56,559

2.3. Feature Engineering

A comprehensive feature engineering process was applied to improve prediction performance and better represent operational processes. Numerous explanatory variables were derived from timestamps and operational records. Only information available at the prediction time was used for each target variable. This ensured that the proposed models remained applicable in real-world operations. It also completely prevented data leakage from future information.

The generated features included operational and demographic variables such as the facility where the operation was performed, the personnel responsible, and employee seniority. Time-based features were also extracted from the timestamps. These included the year, month, day, hour, day of the week, and an indicator showing whether the operation was performed during working hours. These variables helped the models capture seasonal, monthly, weekly, and daily variations in operational workload.

Additional features were generated from the raw data to better represent operational processes. These features included historical performance statistics at both the facility and personnel levels. All statistics were calculated using only records completed before the current operation. They included historical indicators such as mean and median processing times. This produced dynamic features that reflected previous operational performance. In addition, the median processing time of recent operations was used to represent the current operational pace. The total number of operations scheduled on the same day was included to capture daily workload. Processing times from previous operational stages were also used as input features for predicting subsequent stages. This enabled the models to learn temporal dependencies between consecutive stages.

One of the main design principles of this study was to avoid using any information that would not be available at the prediction time. Therefore, all features were generated exclusively from historical data. No information derived from future records or target variables was used. This principle was systematically enforced through automated validation procedures and code-level verification throughout the data preparation process.

2.4. Experimental Design

The distributions of the six productivity metrics are presented in Figure 2 to illustrate the overall characteristics of the dataset. The yearly distribution of the records and the time-based training, validation, and test split used for model development are shown in Figure 3.

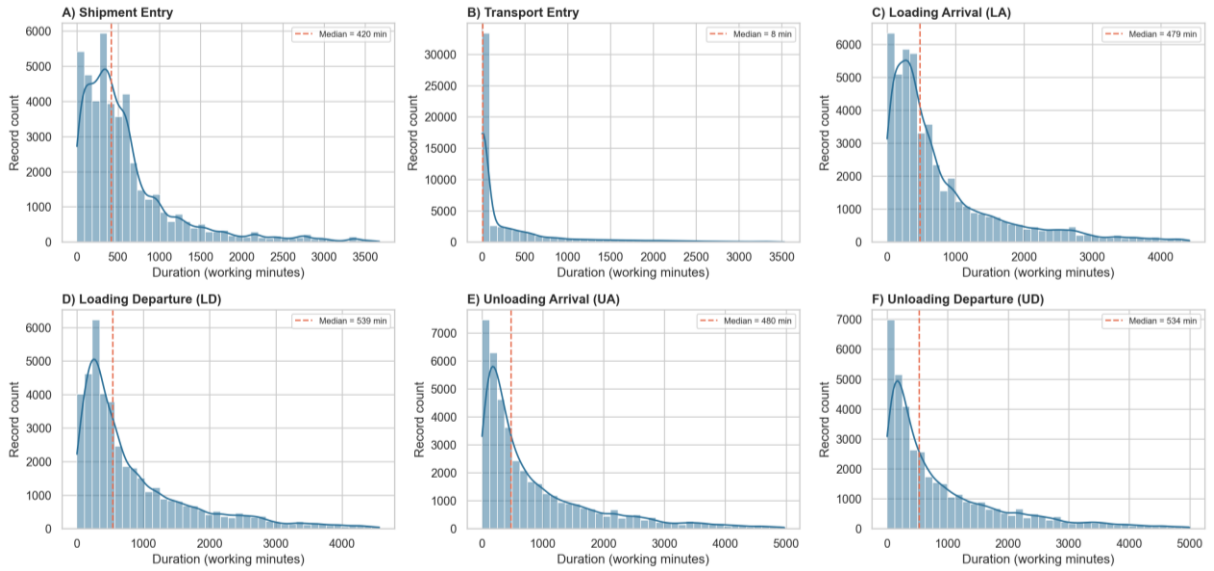

Figure 2. (a), Shipment Entry; (b), Transport Entry; (c), Loading Arrival; (d), Loading Departure; (e), Unloading Arrival; and (f), Unloading Departure.

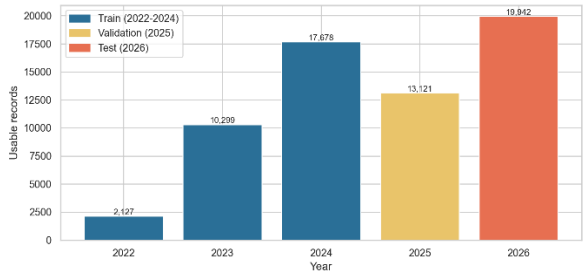

Figure 3. Year based data split

The dataset was divided into three subsets while preserving its temporal order. Records from 2022–2024 were used for training, records from 2025 for validation, and records from 2026 for testing. This time-based split was adopted to prevent data leakage and to provide a realistic evaluation of model generalization on future data.

For the classification task, a threshold was determined for each operational stage by maximizing the F1-score of the slow class on the validation set. Records above the threshold were labeled as slow, whereas those below the threshold were labeled as normal. Since the thresholds were optimized separately for each stage, the resulting classes represent relative slowness rather than a fixed delay duration.

2.5. Baseline and Machine Learning Models

2.5.1. Median Baseline Model

A median baseline is a simple reference predictor that establishes the minimum level of performance attainable without learning from the explanatory variables [19]. Rather than modeling the relationship between the input features and the target, it returns the median of the training target for every observation. Because it ignores all explanatory variables, it indicates whether the more advanced algorithms extract genuine information from the available features [20]. In this study, the median baseline was implemented using the scikit-learn DummyRegressor with the median strategy, and binary labels were obtained by applying the same stage-specific threshold used for the other models. Its simplicity, computational efficiency, and ease of interpretation make it a standard benchmark in predictive modeling studies.

2.5.2. HistGradientBoosting

HistGradientBoosting is a gradient boosting algorithm designed to improve the computational efficiency of conventional gradient boosting methods on large structured datasets. Instead of evaluating all unique feature values, the algorithm discretizes continuous variables into histogram bins before constructing decision trees [21]. This histogram-based learning strategy substantially reduces memory consumption and training time while maintaining predictive accuracy. Similar to other gradient boosting methods, HistGradientBoosting sequentially builds weak decision trees, with each new tree minimizing the residual errors produced by the previous ensemble. The algorithm also incorporates regularization techniques, including early stopping, to reduce overfitting and improve model generalization. Owing to its computational efficiency and scalability, HistGradientBoosting has become a widely used algorithm for regression and classification problems involving large tabular datasets [22].

2.5.3. Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) is an optimized implementation of the gradient boosting framework developed to improve both predictive performance and computational efficiency. The algorithm constructs decision trees sequentially by minimizing an objective function that combines prediction loss with regularization terms [23]. Unlike traditional gradient boosting methods, XGBoost utilizes first- and second-order gradient information during optimization, enabling more accurate tree construction and faster convergence [22]. Additional features such as L1 and L2 regularization, tree pruning, parallel computation, sparse data handling, and efficient memory management contribute to its robustness and scalability. Due to its high predictive accuracy and ability to model complex nonlinear relationships, XGBoost has become one of the most widely adopted machine learning algorithms for structured data analysis [24].

2.5.4. Light Gradient Boosting Machine

Light Gradient Boosting Machine (LightGBM) is an efficient gradient boosting framework developed for large-scale machine learning tasks involving structured data. The algorithm employs histogram-based learning, in which continuous feature values are discretized into histogram bins before tree construction. This approach significantly reduces memory consumption and computational complexity while maintaining high predictive performance. Unlike conventional gradient boosting algorithms that grow trees level by level, LightGBM adopts a leaf-wise tree growth strategy, where the leaf with the maximum loss reduction is expanded at each iteration. This strategy generally improves predictive accuracy and accelerates model convergence. In addition, LightGBM incorporates optimization techniques such as Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB), which reduce the computational burden associated with large datasets and high-dimensional feature spaces. Owing to its high efficiency, low memory requirements, and strong predictive capability, LightGBM has become one of the most widely used ensemble learning algorithms for regression and classification problems involving large tabular datasets [25].

2.6. Confusion Matrix

The confusion matrix is a fundamental tool for evaluating the performance of binary classification models. It compares the predicted class labels with the true labels and summarizes the results into four categories. This provides a detailed assessment of both correct and incorrect predictions. It also enables a more comprehensive evaluation than overall accuracy alone by examining performance for each class separately.

In the confusion matrix, True Positive (TP) represents the number of positive samples correctly classified as positive. True Negative (TN) represents the number of negative samples correctly classified as negative [26]. False Positive (FP) represents the number of negative samples incorrectly classified as positive, whereas False Negative (FN) represents the number of positive samples incorrectly classified as negative [27]. The confusion matrix forms the basis for calculating performance metrics such as accuracy, precision, recall, and the F1-score. In operational environments, correctly identifying slow processes is particularly important. Therefore, both false positive and false negative predictions should be considered in addition to overall accuracy. For this reason, the confusion matrix provides a comprehensive assessment of the strengths and limitations of a classification model. The general structure of the confusion matrix, including its four components, is shown in Figure 4 [28, 29].

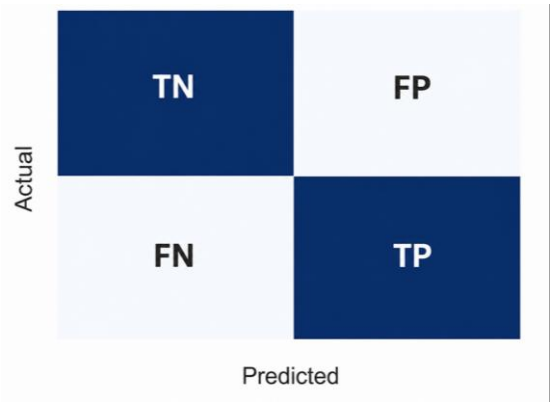

Figure 4. Binary classification confusion matrix

2.7. Performance Metrics

Model performance was evaluated separately for the regression and classification tasks. Regression models were assessed using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Median Absolute Error (MedAE), and the coefficient of determination (R^2). MAE was selected as the primary evaluation metric because it directly measures the average prediction error in working minutes and is easy to interpret. MAE is defined as the average absolute difference between the predicted and actual values, as shown in Equation 1 [30]. RMSE evaluates model performance by assigning greater penalties to large prediction errors and is defined in Equation 2. Median Absolute Error (MedAE) is a robust measure of prediction error that is less sensitive to outliers and is defined in Equation 3. The coefficient of determination (R^2) measures the proportion of variance explained by the model and is calculated using Equation 4 [31].

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{1}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2}$$

$$MedAE = median\left(\left|y_i - (\hat{y})_i\right|\right) \tag{3}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4}$$

In these equations y_i denotes the actual value, $\hat{y}_i$ the predicted value, $\bar{y}$ the mean of the actual values and n the number of observations. Classification performance was evaluated using accuracy, precision, recall, and the F1-score. Accuracy is the proportion of correctly classified samples among all samples and is defined in Equation (5). Precision measures the proportion of correctly identified slow records among all records predicted as slow and is defined in Equation 6 [32]. Recall measures the proportion of actual slow records that are correctly identified by the model and is defined in Equation 7. The F1-score is the harmonic mean of precision and recall and is calculated using Equation 8 [33].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{8}$$

3. EXPERIMENTAL RESULTS

The experimental setup and the obtained results are presented objectively. The dataset, model architectures, and evaluation metrics used in this study are described systematically.

3.1. Experimental Design

All experiments were conducted using the Python programming language. The scikit-learn, XGBoost, and

LightGBM libraries were employed for model development, training, validation, and performance evaluation. To ensure a fair comparison among the evaluated models and to improve the reproducibility of the experimental results, all experiments were performed under the same software environment and hardware configuration. Furthermore, identical data partitioning and evaluation protocols were applied to all models throughout the training, validation, and testing stages. The hardware specifications of the computer used to conduct the experiments are presented in Table 3.

Table 3. Hardware configuration.

Component	Specification
Processor (CPU)	AMD Ryzen 5 5600X (6 cores / 12 threads)
Graphics Processing Unit (GPU)	NVIDIA GeForce RTX 3060 Ti 8 GB
Memory (RAM)	32 GB
Storage	SSD
Operating System	Windows 11 Pro
Development Environment	Python 3.14.0, Jupyter Notebook
Software Libraries	scikit-learn 1.8.0, XGBoost 3.2.0, LightGBM 4.6.0, pandas 3.0.0, NumPy 2.4.2, matplotlib, seaborn

The model hyperparameters were selected on the validation set. For the two best-performing candidate algorithms at each stage, three predefined configurations were evaluated, and the one achieving the lowest validation MAE was retained; the selected configurations are listed in Table 5. All models were trained on a log-transformed target using log1p, and predictions were back-transformed to working minutes using expm1, which reduces the influence of the right-skewed duration distribution. All models were optimized with an L1 (MAE) objective so that the training loss is consistent with the primary evaluation metric, and a fixed random seed (random_state = 42) was used throughout the experiments. HistGradientBoosting was trained with the absolute_error loss, an upper bound of 800 iterations, a learning rate of 0.06, and early stopping (validation_fraction = 0.10, n_iter_no_change = 20). XGBoost used the reg:absoluteerror objective with an upper bound of 2000 trees and early stopping after 50 rounds, a maximum tree depth of 6, a learning rate of 0.05, and both subsample and colsample_bytree set to 0.85. LightGBM used the l1 objective with an upper bound of 2000 trees and early stopping after 50 rounds, 31 leaves, a learning rate of 0.05, and colsample_bytree set to 0.85. All remaining parameters were left at their library defaults unless otherwise specified. The complete model configuration is summarized in Table 4. The approximate training times of the models for the Loading Arrival (LA) stage are presented in Table 6. It should be noted that the model configurations reported in Table 5 were selected according to validation MAE for the regression task. Classification performance was evaluated separately using the stage-specific thresholds, and the models reported in Table 7 were ranked primarily according to the F1-score. Therefore, the best-performing classification model may differ from the regression model selected based on MAE.

Table 4. Model configurations

Model	Parameter	Value
All models	target transform	log1p / expm1 (TransformedTargetRegressor)
	random_state	42
	missing values	training-set median imputation
Median baseline	strategy	median (DummyRegressor)
HistGradientBoosting	loss	absolute_error
	max_iter	800 (upper bound)
	learning_rate	0.06
	early_stopping / validation_fraction / n_iter_no_change	True / 0.10 / 20
	categorical_features	from_dtype
XGBoost	objective / eval_metric	reg:absoluteerror / mae
	n_estimators	2000 (upper bound, early_stopping_rounds = 50)
	max_depth	6
	learning_rate	0.05
	subsample / colsample_bytree	0.85 / 0.85
	tree_method	hist
	reg_lambda	1.0 (library default)
LightGBM	objective	l1
	n_estimators	2000 (upper bound, early stopping = 50)
	learning_rate	0.05
	num_leaves	31
	colsample_bytree (feature_fraction)	0.85
	reg_lambda	0.0 (library default)

Table 5. Stage-specific configurations selected on the validation set

Stage	Selected model	Tuned parameters
Shipment Entry	LightGBM	n_estimators 400, num_leaves 15, learning_rate 0.08
Transport Entry	LightGBM	n_estimators 400, num_leaves 15, learning_rate 0.08
Loading Arrival (LA)	LightGBM	base configuration (no tuned variant selected)
Loading Departure (LD)	LightGBM	n_estimators 600, num_leaves 63, learning_rate 0.03
Unloading Arrival (UA)	LightGBM	n_estimators 600, num_leaves 63, learning_rate 0.03
Unloading Departure (UD)	HistGradientBoosting	min_samples_leaf 50

Table 6. Model training times for the Loading Arrival stage.

Algorithm	Training Time (s)
Median baseline	<0.01
HistGradientBoosting	7.95
LightGBM	1.01
XGBoost	4.23

3.2. Classification Performance

Model selection and the classification threshold were determined exclusively on the validation set (2025). The validation-set classification results are reported in Table 7, whereas the regression and classification results on the independent 2026 test set are presented in Tables 8 and 9, respectively. The independent test set was not used during model training or selection. The confusion matrices for the Loading Arrival stage, which achieved the highest regression performance, are presented in Figure 5. The test set for this stage comprised 18,806 samples, including 14,207 normal and 4,599 delayed records. The reference median baseline correctly classified all 14,207 normal records but classified all 4,599 delayed records as normal. The HistGradientBoosting model correctly classified 2,615 delayed records while misclassifying 1,984 delayed records as normal. The LightGBM model correctly classified 2,642 delayed records and misclassified 1,957 as normal, whereas the XGBoost model correctly classified 2,384 delayed records and misclassified 2,215 as normal. In addition, HistGradientBoosting, XGBoost, and LightGBM incorrectly classified 298, 1,322, and 362 normal records as delayed, respectively. The numbers of correct and incorrect classifications shown in the confusion matrices are consistent with the performance metrics reported for the test set.

The comparison of classification performance across all operational stages and models on the validation set is presented in Figure 6, while the classification metrics of the best-performing model for each stage are summarised in Table 7. Accuracy values ranged from 0.681 to 0.889 across all operational stages. The highest accuracy (0.889) was achieved by LightGBM for the Loading Arrival stage, whereas the lowest accuracy (0.681) was also obtained by LightGBM for the Shipment Entry stage. For the Shipment Entry stage, LightGBM achieved an accuracy of 0.681, a precision of 0.678, a recall of 0.607, and an F1-score of 0.640. For the Transport Entry stage, LightGBM achieved the best performance, with an accuracy of 0.858, a precision of 0.822, a recall of 0.852, and an F1-score of 0.837. For the Loading Arrival stage, the highest performance was also achieved by LightGBM, with an accuracy of 0.889, a precision of 0.803, a recall of 0.605, and an F1-score of 0.690. For the Loading Departure (LD) stage, HistGradientBoosting achieved an accuracy of 0.791, a precision of 0.569, a recall of 0.814, and an F1-score of 0.670. For the Unloading Arrival (UA) stage, HistGradientBoosting achieved an accuracy of 0.758, a precision of 0.546, a recall of 0.535, and an F1-score of 0.540, whereas for the Unloading Departure (UD) stage, the same model achieved an accuracy of 0.775, a precision of 0.583, a recall of 0.503, and an F1-score of 0.540. Because the class distribution was imbalanced, precision, recall, and F1-score are reported alongside accuracy.

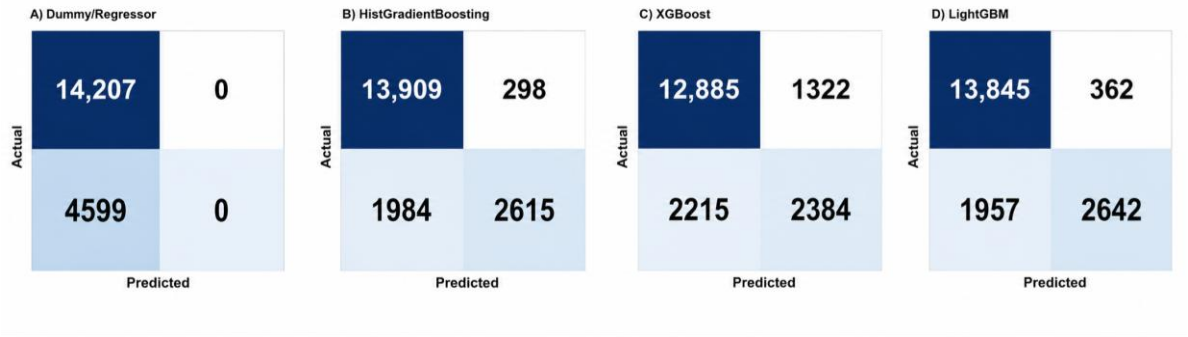

Figure 5. Loading arrival (a) Median baseline; (b) HistGradientBoosting; (c) XGBoost; (d) LightGBM

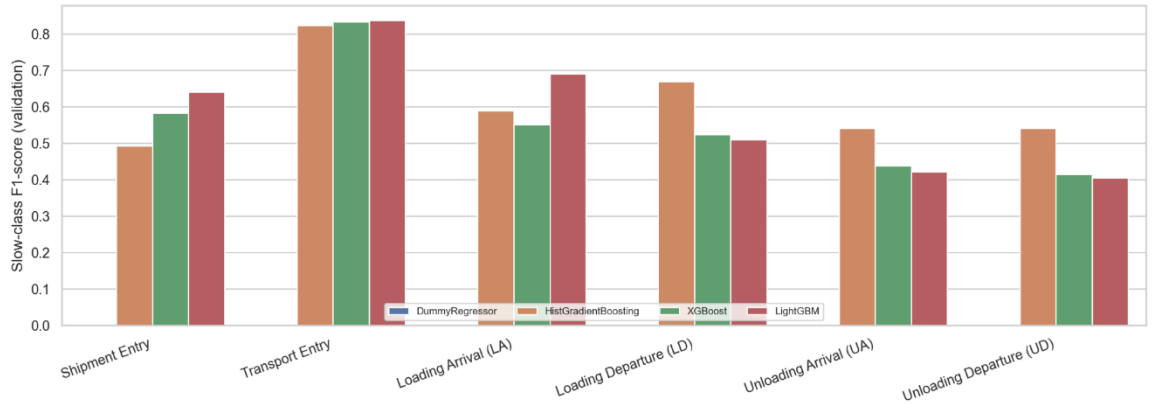

Figure 6. Comparison of model performance across stages

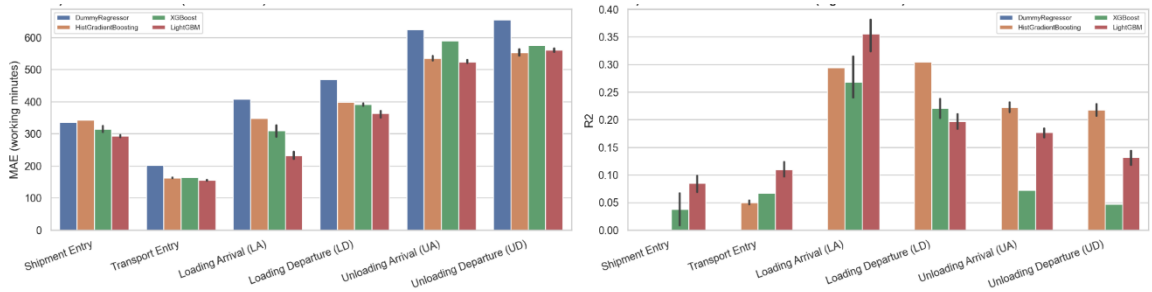

Figure 7. (a) Mean absolute error; (b) Coefficient of determination.

Table 7. Validation classification results

Stage	Best Model	Accuracy	Precision	Recall	F1-score
Shipment Entry	LightGBM	0.681	0.678	0.607	0.640
Transport Entry	LightGBM	0.858	0.822	0.852	0.837
Loading Arrival (LA)	LightGBM	0.889	0.803	0.605	0.690
Loading Departure (LD)	HistGradientBoosting	0.791	0.569	0.814	0.670
Unloading Arrival (UA)	HistGradientBoosting	0.758	0.546	0.535	0.540
Unloading Departure (UD)	HistGradientBoosting	0.775	0.583	0.503	0.540

3.3. Regression Performance

The results of the regression analysis on the validation set are presented in Figure 7 using Mean Absolute Error (MAE) and the coefficient of determination (R^2) as performance metrics for the four evaluated models. To improve the readability of the figure, negative R^2 values were displayed as 0 in the graphical representation; however, all numerical analyses and performance comparisons were based on the actual R^2 values.

The generalization performance of the models on the temporally held-out 2026 test set was evaluated using the best-performing model for each operational stage. The regression results are presented in Table 8, the corresponding classification results are presented in Table 9, and the distribution of test-set MAE values across the operational stages is illustrated in Figure 8.

Table 8. Test set regression results (2026)

Stage	Best model	MAE (working min)	RMSE (working min)	MedAE (working min)	R^2
Shipment Entry	LightGBM	275.774	546.011	138.018	0.138
Transport Entry	LightGBM	234.825	508.713	93.966	0.028
Loading Arrival (LA)	LightGBM	285.989	646.843	89.971	0.371
Loading Departure (LD)	LightGBM	434.068	828.552	179.789	0.168
Unloading Arrival (UA)	LightGBM	652.756	1067.006	345.749	0.024
Unloading Departure (UD)	HistGradientBoosting	675.993	1111.027	340.496	0.015

Table 9. Test set classification results (2026)

Stage	Model	Accuracy	Precision	Recall	F1-score
Shipment Entry	LightGBM	0.752	0.798	0.612	0.693
Transport Entry	LightGBM	0.800	0.865	0.808	0.836
Loading Arrival (LA)	LightGBM	0.877	0.879	0.574	0.695
Loading Departure (LD)	LightGBM	0.786	0.781	0.386	0.517
Unloading Arrival (UA)	LightGBM	0.682	0.565	0.336	0.422
Unloading Departure (UD)	HistGradientBoosting	0.695	0.639	0.379	0.476

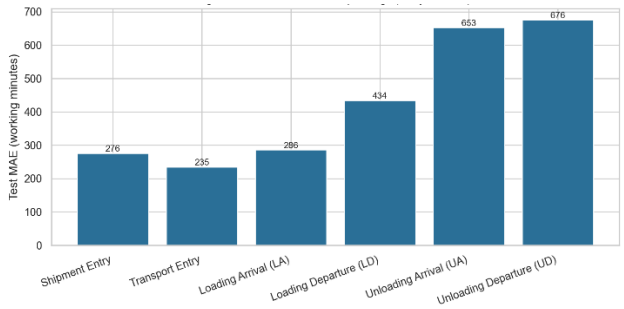

Figure 8. Test-set MAE of the best model by stage

As shown in Table 8, the highest coefficient of determination ($R^2 = 0.371$) was achieved by LightGBM for the Loading Arrival stage, where the corresponding test MAE was 285.989 working minutes. Because the absolute duration scale differs substantially across operational stages, the raw MAE values are not directly comparable between stages. Accordingly, although the Transport Entry stage achieved the lowest MAE (234.825 working minutes), its corresponding R^2 was only 0.028, indicating limited explanatory power. For the Shipment Entry stage, LightGBM achieved an MAE of 275.774 working minutes with an R^2 of 0.138, whereas for the Loading Departure stage it achieved an MAE of 434.068 working minutes with an R^2 of 0.168. For the Unloading Arrival stage, LightGBM was the best-performing model, achieving an MAE of 652.756 working minutes and an R^2 of 0.024. For the Unloading Departure stage, HistGradientBoosting achieved the best performance with an MAE of 675.993 working minutes and an R^2 of 0.015, representing the lowest regression performance among all operational stages.

The classification performance of the same best-performing models on the 2026 test set is presented in Table 9. The highest classification accuracy (0.877) was achieved for the Loading Arrival stage, where LightGBM also achieved a precision of 0.879, a recall of 0.574, and an F1-score of 0.695. In terms of F1-score, however, the best classification performance was obtained for the Transport Entry stage (0.836). Classification performance decreased noticeably for the unloading stages, where the Unloading Arrival and Unloading Departure stages achieved F1-scores of 0.422 and 0.476, respectively. This decline supports the interpretation that delays during the unloading stages are influenced by external operational factors that are not represented in the available dataset.

4. DISCUSSION

This study proposed a machine learning-based framework to predict personnel data entry times across six operational stages using transaction records collected from a real-world supply chain organization. Both regression and binary classification formulations were investigated, and HistGradientBoosting, XGBoost, and LightGBM models were evaluated against a median-based baseline using a common preprocessing pipeline, identical feature sets, and a temporal train-validation-test split. This experimental design enabled a fair comparison of different ensemble learning algorithms while preserving the chronological structure of the operational data.

The experimental results demonstrate that prediction performance strongly depends on the operational stage. The highest regression performance was achieved for the Loading Arrival stage, where LightGBM produced a mean absolute error of 286.0 working minutes with an R^2 value of 0.371, indicating that the processing time at this stage can be explained comparatively better by the available operational attributes. Similarly, the loading departure and shipment creation stages achieved relatively higher predictive performance with R^2 values of 0.168 and 0.138. In contrast, the transportation and unloading stages exhibited substantially lower coefficients of determination, implying that a considerable portion of their variability originates from external factors not represented in the available dataset, such as customer requests, traffic conditions, scheduling decisions, or other organizational constraints.

The classification experiments further confirmed these findings. Ensemble learning models consistently outperformed the baseline approach in stages where sufficient predictive information existed, while performance naturally declined in stages characterized by severe class imbalance and inherently unpredictable delays. Nevertheless, the obtained results indicate that binary early-warning systems remain feasible for operational stages where delays can be reliably anticipated.

An important contribution of this study is the use of working minutes instead of calendar time as the prediction target. By excluding non-working periods such as nights and weekends, artificial delays unrelated to actual

operational performance were removed. Consequently, the models learned genuine workforce processing behavior rather than organizational calendar effects, leading to more interpretable and operationally meaningful predictions.

From a practical perspective, the proposed framework is suitable for real-time decision support applications. If the estimated processing time exceeds a predefined threshold, the corresponding operation can be automatically flagged as slow, enabling proactive intervention by operations managers before significant delays occur. Furthermore, the same prediction outputs can be integrated into reporting dashboards to monitor facility-level and personnel level performance, identify bottlenecks, and support continuous process improvement. In practical deployment, these stage-specific thresholds should not necessarily remain fixed. As new operational records accumulate, the thresholds can be periodically recalibrated using a recent validation window by re-optimizing the F1-score of the slow class. Such periodic recalibration would allow the decision-support system to adapt to changes in workload, personnel behavior, and operational conditions.

Among the evaluated algorithms, HistGradientBoosting and LightGBM provided the most favorable balance between prediction accuracy and computational efficiency. Their fast training and inference times, together with their robust predictive performance, make them attractive candidates for deployment in industrial environments where computational resources and response times are important considerations. Overall, the results demonstrate that ensemble learning methods constitute an effective and computationally efficient solution for predicting personnel processing times in real world logistics operations.

Despite these encouraging results, several limitations should be acknowledged. The experiments were conducted using data from a single organization, limiting the direct generalizability of the findings to other industries or operational settings. In addition, the independent test set consisted only of records from 2026, representing a relatively limited temporal evaluation period. A further limitation concerns the gap between training and validation performance. Although early stopping and log-transformed targets were employed to improve generalization, all operational stages exhibited a noticeable train–validation performance gap, indicating that the models partly captured stage-specific noise. This observation reflects the high variability and limited predictability of several operational stages rather than a deficiency of the learning algorithms themselves, and suggests that stronger regularization or the inclusion of richer external operational features would be required to further improve generalization. Finally, stages such as Shipment Entry, Transport Entry, and the unloading stages contained highly imbalanced class distributions, which reduced classification performance and suggested that alternative approaches, such as cost-sensitive learning, class-weighted objectives, resampling strategies such as oversampling or undersampling, anomaly detection, or survival analysis, may be more suitable for these scenarios. Future research should investigate the transferability of the proposed framework using datasets collected from multiple companies and sectors. Incorporating explainable artificial intelligence techniques, such as SHAP or LIME, could improve model transparency and facilitate managerial interpretation of predictions. Furthermore, enriching the feature set with external operational variables including traffic conditions, weather information, workload indicators, and scheduling constraints may improve predictive performance, particularly for the early operational stages. Finally, evaluating the proposed system through real-world deployment and measuring its impact on operational efficiency, delay reduction, and managerial decision-making would provide valuable evidence of its practical effectiveness.

5. CONCLUSIONS

This study presented a machine learning framework for predicting personnel data entry times across six operational stages in a real-world supply chain organization. The results demonstrated that ensemble learning algorithms provide effective predictions for operational stages with sufficient predictive information, while prediction performance decreases in stages strongly influenced by external factors. Among the evaluated models, HistGradientBoosting and LightGBM achieved the best balance between predictive accuracy and computational efficiency. Furthermore, calculating target variables in working minutes rather than calendar time enabled a more realistic representation of operational performance. Overall, the proposed framework offers a practical solution for supporting personnel performance monitoring, operational planning, and real-time decision-making in data-driven supply chain management. Future studies should validate the framework using data from multiple organizations and investigate the integration of additional operational variables and explainable artificial intelligence techniques to further improve prediction performance and model interpretability.

Declaration of Ethical Standards

The article does not contain any studies with human or animal subjects.

Credit Authorship Contribution Statement

Muhammed Yıldıray SURMEN: Conceptualization, Methodology, Software, Data Curation, Formal Analysis, Investigation, Visualization, Writing – Original Draft.

Oya KILCI: Conceptualization, Methodology, Validation, Supervision, Writing – Review & Editing.

Selime Sinem BAHAR: Data Curation, Investigation, Validation, Resources, Writing – Review & Editing.

Murat KOKLU: Conceptualization, Methodology, Supervision, Validation, Writing – Review & Editing, Project Administration.

Declaration of Competing Interest

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

Funding / Acknowledgements

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

Data Availability

The operational data used in this study were provided and used with permission from the data-owning organization. The data are not publicly available because they contain confidential operational information. Data may be made available from the corresponding author upon reasonable request and with permission from the data owner.

References

[1] A. S. Lather, R. Malhotra, P. Saloni, P. Singh, and S. Mittal, "Prediction of employee performance using machine learning techniques," in *Proceedings of the 1st international conference on advanced information science and system*, 2019, pp. 1–6, doi: 10.1145/3373477.3373654.

[2] P. Tambe, P. Cappelli, and V. Yakubovich, "Artificial intelligence in human resources management: Challenges and a path forward," *California management review,* vol. 61, no. 4, pp. 15–42, 2019, doi: 10.1177/0008125619867910.

[3] P. R. Palos-Sánchez, P. Baena-Luna, A. Badicu, and J. C. Infante-Moro, "Artificial intelligence and human resources management: A bibliometric analysis," *Applied artificial intelligence,* vol. 36, no. 1, p. 2145631, 2022, doi: 10.1080/08839514.2022.2145631.

[4] A. Uppal, Y. Awasthi, and A. Srivastava, "Machine learningbased approaches for enhancing human resource management using automated employee performance prediction systems," *International Journal of Organizational Analysis,* vol. 33, no. 8, pp. 2307–2346, 2025, doi: 10.1108/IJOA-07-2024-4643.

[5] M. Awada, B. Becerik-Gerber, G. Lucas, and S. C. Roll, "Predicting office workers' productivity: A machine learning approach integrating physiological, behavioral, and psychological indicators," *Sensors,* vol. 23, no. 21, p. 8694, 2023, doi: 10.3390/s23218694.

[6] S. Koldijk, M. A. Neerincx, and W. Kraaij, "Detecting work stress in offices by combining unobtrusive sensors," *IEEE Transactions on affective computing,* vol. 9, no. 2, pp. 227–239, 2016, doi: 10.1109/TAFFC.2016.2610975.

[7] L. Shu *et al.*, "Wearable emotion recognition using heart rate data from a smart bracelet," *Sensors,* vol. 20, no. 3, p. 718, 2020, doi: 10.3390/s20030718.

[8] C.-T. Li, J. Cao, and T. M. Li, "Eustress or distress: An empirical study of perceived stress in everyday college life," in *Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing: Adjunct*, 2016, pp. 1209–1217, doi: 10.1145/2968219.2968309.

[9] E. V. Orlova, "Innovation in company labor productivity management: Data science methods application," *Applied System Innovation,* vol. 4, no. 3, p. 68, 2021, doi: 10.3390/asi4030068.

[10] T. Nguyen, Z. Li, V. Spiegler, P. Ieromonachou, and Y. Lin, "Big data analytics in supply chain management: A state-of-the-art literature review," *Computers & operations research,* vol. 98, pp. 254–264, 2018, doi: 10.1016/j.cor.2017.07.004.

[11] G. Kannan, T. C. E. Cheng, M. Nishikant, and S. Nagesh, "Big data analytics and application for logistics and supply chain management," *Transportation Research Part E: Logistics and Transportation Review,* vol. 114, pp. 343–349, 2018, doi: 10.1016/j.tre.2018.03.011.

[12] Y. Riahi, T. Saikouk, I. Badraoui, and S. Fosso Wamba, "Researched topics, patterns, barriers and enablers of artificial intelligence implementation in supply chain: a Latent-Dirichlet-allocation-based topic-modelling and expert validation," *Production Planning & Control,* vol. 36, no. 5, pp. 565–592, 2025, doi: 10.1080/09537287.2023.2286523.

[13] W. M. Van der Aalst, M. H. Schonenberg, and M. Song, "Time prediction based on process mining," *Information systems,* vol. 36, no. 2, pp. 450–475, 2011, doi: 10.1016/j.is.2010.09.001.

[14] F. M. Maggi, C. Di Francescomarino, M. Dumas, and C. Ghidini, "Predictive monitoring of business processes," in *International conference on advanced information systems engineering*, 2014: Springer, pp. 457–472, doi: 10.1007/978-3-319-07881-6_31.

[15] I. Verenich, M. Dumas, M. L. Rosa, F. M. Maggi, and I. Teinemaa, "Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring," *ACM Transactions on Intelligent Systems and Technology (TIST),* vol. 10, no. 4, pp. 1–34, 2019, doi: 10.1145/3331449.

[16] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics,* pp. 1189–1232, 2001, doi: 10.1214/aos/1013203451.

[17] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[18] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?," Advances in Neural Information Processing Systems, vol. 35, 2022 [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/0378c7692da36807bdec87ab043cdadc-Paper-Datasets_and_Benchmarks.pdf.

[19] H. Yang and L. Yang, "Asymmetric effects of attribute performance on transit satisfaction among older adults: machine learning regression with dummy variables," *International Journal of Environmental Science and Technology,* vol. 23, no. 3, p. 194, 2026, doi: 10.1007/s13762-025-07004-0.

[20] B. Marina, I. Memon, F. A. Alvi, U. Rajput, and M. Nabi, "Machine Learning-Driven Security and Privacy Analysis of a Dummy-ABAC Model for Cloud Computing," *Computers,* vol. 14, no. 10, p. 420, 2025, doi: 10.3390/computers14100420.

[21] M. Tamim Kashifi and I. Ahmad, "Efficient histogram-based gradient boosting approach for accident severity prediction with multisource data," *Transportation research record,* vol. 2676, no. 6, pp. 236–258, 2022, doi: 10.1177/03611981221074370.

[22] N. Lingling *et al.*, "Streamflow forecasting using extreme gradient boosting model coupled with Gaussian mixture model," *Journal of Hydrology,* vol. 586, p. 124901, 2020, doi: 10.1016/j.jhydrol.2020.124901.

[23] H. Incekara, I. H. Cizmeci, M. M. Saritas, and M. Koklu, "Classification of almond kernels with optuna hyper-parameter optimization using machine learning methods," *Journal of Food Science and Technology,* pp. 1–17, 2025, doi: 10.1007/s13197-025-06494-7.

[24] A. C. Citak, S. S. Bayram, and M. Koklu, "Classification of obesity levels using machine learning algorithms," *Intelligent Methods in Engineering Sciences,* vol. 4, no. 3, pp. 100–113, 2025, doi: 10.58190/imiens.2025.157.

[25] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient

boosting decision tree," in *Advances in Neural Information Processing Systems*, vol. 30, pp. 3146–3154, 2017.

[26] M. M. Saritas, Y. S. Taspinar, I. Cinar, and M. Koklu, "Railway track fault detection with ResNet deep learning models," in *2023 International Conference on Intelligent Systems and New Applications (ICISNA'23)*, Liverpool, United Kingdom, Apr. 28–30, 2023, pp. 66–71, E-ISBN: 978-605-72180-3-2.

[27] M. Koklu, B. Isgor, and C. Tumer, "Hybrid Feature-Based Two-Stage Framework for Audio Deepfake Detection and Generative Model Attribution," *Advances in Engineering and Intelligence Systems,* vol. 5, no. 01, pp. 241–259, 2026, doi: 10.22034/aeis.2026.566863.1400.

[28] K. Sabanci, M. Koklu, and M. F. Unlersen, "Classification of Siirt and long type pistachios (pistacia vera l.) by artificial neural networks," *International Journal of Intelligent Systems and Applications in Engineering,* vol. 3, no. 2, pp. 86–89, 2015, doi: 10.18201/ijisae.74573.

[29] B. Unver, M. T. Yoldar, S. Ozkan, and M. Koklu, "Vision Transformers in Person Re-Identification: A Review," *IEEE Access,* vol. 14, pp. 92748–92764, 2026, doi: 10.1109/ACCESS.2026.3704666.

[30] Ü. Yavuz, U. Ekim, and M. Köklü, "Üniversite Öğrencilerin Ortak Zorunlu Derslerdeki Başarılarının K-Means Algoritması İle İncelenmesi," *NWSA: Engineering Sciences,* vol. 6, no. 1, pp. 342–347, 2011.

[31] Ö. Yılmaz, A. Altun, and M. Köklü, "A new hybrid algorithm based on MVO and SA for function optimization," *International Journal of Industrial Engineering Computations,* vol. 13, no. 2, pp. 237–254, 2022.

[32] O. Kilci, Y. Eryesil, and M. Koklu, "Classification of Biscuit Quality With Deep Learning Algorithms," *Journal of Food Science,* vol. 90, no. 7, p. e70379, 2025, doi: 10.1111/1750-3841.70379.

[33] O. Kilci and M. Koklu, "Machine learning-based detection of solar panel surface defects using deep features from InceptionV3," in *4th International Conference on Trends in Advanced Research Konya, Türkiye*, Jul. 4–5, 2025, p. 59.