International Journal of Electrical and Computer Engineering (IJECE)
Vol. 15, No. 6, December 2025, pp. 5904~5913
ISSN: 2088-8708, DOI: 10.11591/ijece.v15i6.pp5904-5913

Enhancing supply chain agility with advanced weather forecasting

Imane Zeroual, Jaber El Bouhdidi
SIGL Laboratory, ENSATE, Abdelmalek Essadi University, Tetouan, Morocco

Article Info

Article history:

Received Dec 7, 2024
Revised Jul 15, 2025
Accepted Sep 14, 2025

Keywords:

Decision tree
Neural network
Random forest
Urban logistic
Weather forecasting

ABSTRACT

This article presents a solution that leverages artificial intelligence techniques to enhance urban freight transportation planning and organization through the integration of weather forecasting data. We identify key challenges in the current urban logistics landscape and introduce a range of machine learning models designed to predict delivery delays. Logistic regression serves as the foundational model, analyzing historical delivery data in conjunction with weather conditions to assess the likelihood of delays, thus enabling informed decision-making for companies. Additionally, we evaluate two other machine learning models to determine the most effective approach for our specific context, assessing their accuracy and capacity to deliver actionable insights. By improving the predictive capabilities of urban freight systems, this research aims to streamline operations, reduce costs, and enhance overall service reliability, contributing to more efficient and resilient urban transportation networks.

Corresponding Author:

Imane Zeroual
SIGL Laboratory, ENSATE, Abdelmalek Essadi University
Av. Khenifra, Tétouan 93000, Morocco
Email: imane.zeroual1@etu.uae.ac.ma

1. INTRODUCTION

Logistics in the supply chain involves the strategic transportation of goods from manufacturing sites to consumers, a process increasingly complicated by unpredictable weather conditions. Severe weather events such as heavy rain, snow, and strong winds pose significant challenges, leading to delivery delays [1], hazardous road conditions, and disruptions in port operations. These issues affect logistics efficiency and can result in financial losses and diminished customer satisfaction. The core problem lies in the inability to effectively predict and respond to these weather-related disruptions, which can cascade through the supply chain, impacting inventory management and overall operations. As such, logistics management companies must implement robust planning strategies to mitigate these risks, relying heavily on technology to monitor and adapt to changing weather conditions [2]. In response to this challenge, we propose a solution that integrates weather application programming interface (API) into supply chain management systems. These APIs provide real-time and forecasted weather data, enabling logistics firms to optimize routes and make informed decisions proactively. By analyzing up-to-date weather forecasts, delivery planners can adjust routes to avoid adverse conditions, ensuring safer and more timely deliveries. Additionally, predictive analytics offered by these APIs allow logistics companies to anticipate weather patterns, facilitating better resource allocation and inventory planning. This strategic approach aims to enhance the resilience of urban freight transportation, ultimately improving service reliability and operational efficiency.

2. METHOD

2.1. Architectural overview of the proposed solution

The proposed architecture illustrated in Figure 1 (blue bloc), the MuleSoft batch process "logistics-weather-enrichment-bch" plays a central role in orchestrating the data flow. First, it retrieves logistics data from a comma-separated values (CSV) file, which contains details such as delivery locations, delivery status, and other relevant information. Next, the batch queries a historical weather API for each delivery location to gather weather data such as temperature, humidity, or precipitation at the delivery time Once the logistics and weather data are collected, the batch process performs a series of transformations including cleaning, structuring the data into a consistent format that is suitable for further analysis, and joining the logistics data with the corresponding weather data for each delivery, ensuring that all the necessary information is aligned. The enriched data is then stored in "logistics-working-db" database, where it becomes ready for use in training machine learning models. This dataset will serve as the foundation for training a predictive model that can forecast delivery delays based on past weather and logistics data. This approach ensures that the model has access to historical data for both logistics operations and weather conditions, making it more accurate in its predictions.

The (green bloc) of Figure 1 focuses on training a machine learning (ML) model for predicting delivery delays based on historical logistics data and weather conditions stored in the database "logistics-working-db". The workflow involves data retrieval, model training, model storage in a model registry, and exposing an API for prediction. Here is how the entire process works:

a. The process begins with retrieving the historical logistics data and weather data stored in the database "logistics-working-db" where the data has already been enriched with historical weather conditions. This data includes delivery timestamps, locations, weather data (*e.g.*, temperature and precipitation), and other relevant logistics information.
b. Once the data is retrieved, the script processes the data, performs feature engineering such as handling missing values, encoding categorical features, and scaling numerical values, and trains the model using two algorithms: linear regression and random forests.
c. After training, the model is saved and stored in a model registry that tracks different versions of the model, which is essential for version control and reproducibility. Each model version has associated metadata, such as training configuration (hyperparameters used, dataset version) and evaluation metrics (*e.g.*, accuracy, F1 score).
d. Once the model is trained and registered, a model REST API is created to serve the trained model. The API accepts the same type of data (delivery details, weather conditions) that the model was trained on, performs the necessary pre-processing, and returns the predicted delivery delay.
e. The Process API "logistics-weather-enrichment-prc" which will be detailed later in the manuscript, calls the model REST API to make predictions. This interaction ensures that when new data is processed through the pipeline, it can trigger a prediction based on the trained model.

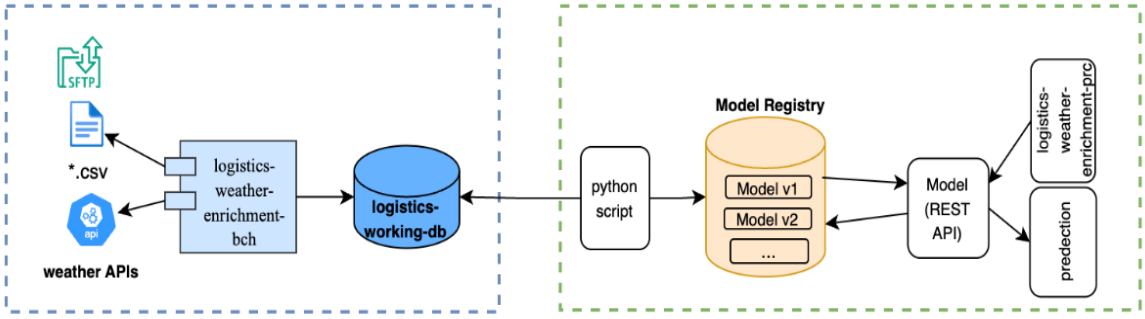

Figure 1. Architecture of machine learning and batch processing pipelines for delivery delay prediction and weather-enriched logistics data

Once the machine learning model has been trained on historical data, the architecture shown in Figure 2 is set up to process incoming delivery orders in real-time and predict potential delays. The process begins when the experience API receives logistic data from the incoming delivery order through a delivery Webhook. This initiates the flow of data into the system. The experience API then passes the order information to the Process API, which is responsible for further data processing and integration with external services. The Process API consolidates this data, including the delivery details and the predicted delay, and

sends it to the system API. The system API is responsible for storing the enriched data in a database for future reference and analysis. Let's break down this architecture illustrated in Figure 2 and explain the role and functionality of each API in detail:

a. Experience API: it handles the incoming logistics data and serves as the first point of interaction for the system. It exposes an endpoint that captures the incoming logistic data pushed via the delivery Webhook and forwards it to the MuleSoft "*q.logistics.delivery.v1*" queue where it can be subsequently handled by the Process API for further processing and analysis.
b. Process API: it is responsible for retrieving logistics data from the MuleSoft queue "*q.logistics.delivery.v1*". Once it receives the data, the API performs essential data enrichment tasks. This includes calling the external Weather API to fetch historical weather data relevant to the delivery location and timeframe. The Process API also applies core business logic, such as data transformation, validation, and necessary calculations, to ensure the logistics data is properly prepared for further analysis. In addition, the Process API calls the prediction ML API to generate accurate delivery delay predictions based on the enriched data. Once the prediction is generated, the Process API indexes the enriched data in Elasticsearch to enable real-time analytics. It then forwards the enriched and predicted data to the MuleSoft queue "*q.logistics.weather.delivery.v1*" for downstream consumption, allowing the system API to manage storage and indexing for future use.
c. System API: it is responsible for capturing the enriched logistics data, including the delivery predictions, from the MuleSoft queue "*q.logistics.weather.delivery.v1*". Once the data is retrieved, the system API stores the enriched data, including both the original logistics details and the generated delivery predictions in the database "logistics-working-db". This ensures the data is securely saved and made available for future reference, reporting, and further analysis. This architecture ensures that the system can process and predict delivery delays effectively in real-time, using historical data for prediction while also enabling continuous monitoring and reporting through Elasticsearch.

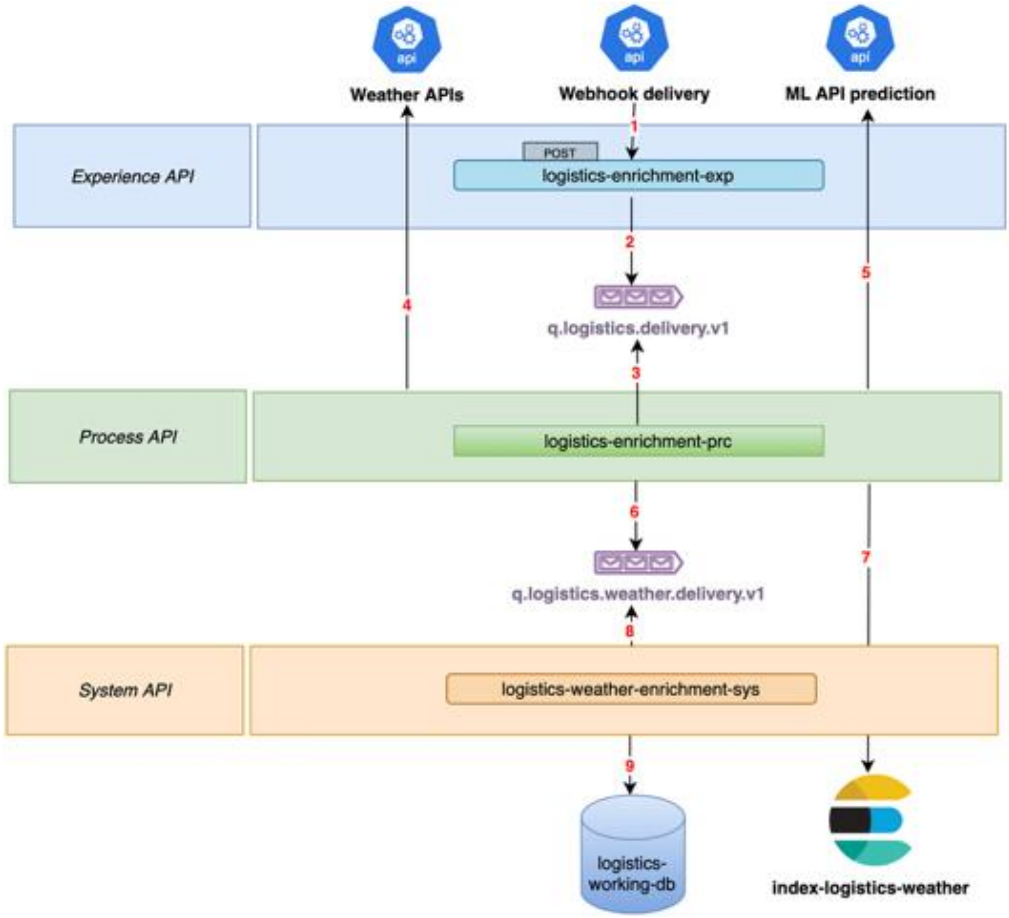

Figure 2. High-level design for logistics data processing and delivery delay forecasting

In the current architecture, the solution is designed to continuously evolve and enhance its predictions over time. By integrating real-time data and regularly retraining the machine learning model, the system ensures that its predictions remain both accurate and relevant. The system API plays a crucial role in this process by triggering model retraining. As shown in Figure 3, when the volume of new data stored in the database exceeds a predefined threshold, the system API activates a separate batch" logistics-weather-training-bch" (as detailed in Figure 2) to initiate the retraining of the model. This approach ensures that the machine learning model is continuously updated with the latest data, improving its accuracy and predictive capabilities [3].

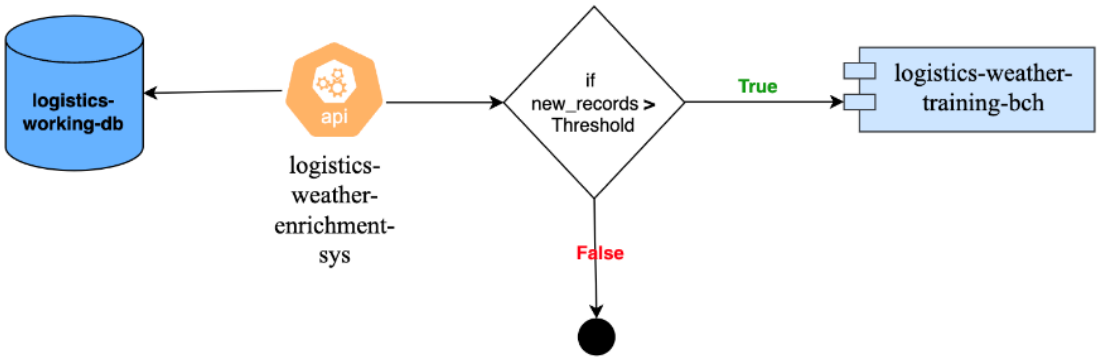

Figure 3. Continuous model improvement and retraining process

2.2. Data exploration and preprocessing

Our system leverages an application programming interface (API) to retrieve historical weather data, providing crucial insights into past weather conditions. The weather REST API (*https://archive-api.open-meteo.com/v1/era/5*) functions efficiently by using the input data from the CSV file as query parameters specified in Table 1 to fetch the relevant weather information. By integrating the weather API, our system gains access to historical weather patterns, which play a key role in predicting delivery delays. This historical weather data enriches our predictions, improving our forecasts' accuracy and reliability [4]. The historical weather API appears as:

Table 1. Weather API input specifications

Parameter	Format	Required	Value (example)	Description
Latitude, longitude	Floating point	Yes	18.35909462 66.07995606	Geographical WGS84 coordinate of the location
Start_date, end_date	String (yyyy-mm-dd)	No	2018-02-03 20218-02-06	The time interval to get weather data
Timezone daily	String	No	Auto	All timestamps are returned as local time
	String array	No	Rain_sum,weathercode, Precipitation_sum, precipitation_hours, snowfall_sum	A list of daily weather variable aggregations

Data cleaning is a crucial step in the machine learning pipeline [5], involving the identification and correction of errors, inconsistencies, and inaccuracies in the dataset. High-quality, clean data is fundamental for building reliable and accurate machine learning models. Below are some of the common data-cleaning techniques we applied to our dataset:

a. Handling missing values: identify and address missing data by removing rows and columns with excessive missing values and using imputation techniques [6] to fill in the gaps.
b. Data type conversion: ensure that the data types used in the dataset are compatible with the machine learning algorithm. For instance, we converted categorical variables into a numerical format using methods like one-hot encoding [7] and label encoding [8].
c. Removing duplicates: identify and eliminate duplicate [9] records to avoid redundancy and improve model accuracy.
d. Encoding categorical data: convert categorical variables into a format suitable for machine learning models.

We used one-hot or label encoding techniques, which transform non-numeric data into numerical values. Cleaning time-series data: for time-series data, we address challenges such as missing timestamps, irregular intervals, and seasonality by filling gaps or resampling the data to ensure consistency. Effective data cleaning [10] significantly improves the performance and reliability of machine learning models by ensuring that the data is well-prepared for analysis.

The logistics data contains various variables related to the delivery of different products. These variables include information such as "days for shipping (actual)," "days for shipment (scheduled)," "delivery status," "late delivery risk," and more. Table 2 provides an overview of some of the values in these columns.

Table 2. Examples of column values in the logistics dataset

Column name	Type	Values (example)	Description
Late delivery risk	Numerical	0 or 1	A binary variable, coded as 0 or 1. 1 indicates that the delivery is late, while 0 indicates it's on time
Delivery status	Categorical	Advance shipping, Late delivery, Shipping on time	Represents the status of the order delivery
Latitude	Numerical	18.35909462	Represents the geographical location of the delivery destination

The dataset includes several columns irrelevant to our analysis, as they either lack utility or do not significantly contribute to the predictive modeling tasks. As a result, we will initiate a data selection process to improve model efficiency. Feature selection is a key step in the machine learning pipeline [11] where we identify and retain the most relevant features from the original dataset. This process is important for several reasons:

a. Dimensionality reduction: by removing irrelevant or redundant features, we reduce the dimensionality of the dataset [12], which helps in faster model training and improves model generalization.
b. Improved model performance: selecting only the most informative features can enhance the model's accuracy while reducing the risk of overfitting.
c. Enhanced interpretability: fewer features typically make the model easier to interpret and understand, which is particularly important in practical applications.

There are several methods for feature selection, including:

a. Correlation-based methods: identifying highly correlated features [13] to eliminate redundant variables. Tree-based methods: using decision trees or tree ensemble methods (*e.g.*, random forests) to rank feature importance [14].
b. Dimensionality reduction: techniques such as principal component analysis (PCA) [15] to reduce the number of features.
c. Feature importance scores: estimating the relevance of each feature based on how it contributes to model predictions [16].

We have chosen to rely on feature importance scores because they are well-suited to our dataset and the machine learning algorithms we plan to use. Moreover, it is often beneficial to experiment with different feature selection techniques and assess their impact on model performance through cross-validation [17]. This process will guide us in selecting the most significant features for our predictive model. We proceeded to a feature important score from a feature selection analysis. The scores in Table 3 demonstrate how important each feature is in predicting the target variable (late delivery risk):

Table 3. Selected features from the feature selection process

	Feature	Importance
4	Latitude	0.290747
5	Longitude	0.261602
0	Elevation	0.226257
1	Rain avg	0.122533
2	Precipitation_hours_avg	0.084280
3	Snowfall_sum	0.014581

Feature importance scores help quantify the contribution of each variable to the model's predictions. Features with higher scores, such as Latitude and Longitude, are considered more influential in determining the target variable:

a. Longitude: ranking second with a score of about 0.2616, Longitude also plays a critical role in forecasting the target variable. Like Latitude, changes in Longitude substantially affect the model's predictions.
b. Elevation: with a feature importance score of around 0.2263, Elevation is important but less so than latitude and longitude. Despite its relatively lower importance, changes in Elevation still have a notable effect on the model's predictions.
c. Rain avg: with a score of approximately 0.1225, Rain avg contributes less than the geographical features (Latitude, Longitude, and Elevation), but it still holds significant predictive power in the model.
d. Precipitation hours avg: this feature has a significance score of about 0.0843. While it is less influential than Rain_avg, it still provides valuable information for the model's predictions.
e. Snowfall sum: at 0.0146, snowfall sum has the lowest feature importance score. This indicates that relative to the other variables, it has the least influence on predicting (late delivery risk).

2.3. Machine learning models used for predicting delivery delays

We employed two machine learning models to predict delivery delay: Logistic regression and random forest. These models were chosen for their proven effectiveness in classification tasks and their ability to handle different types of data relationships:

a. Logistic regression is a widely used machine learning algorithm for predictive tasks in various domains. It is commonly used for both binary and multi-class classification, making it versatile for different scenarios [18].
b. Random forest is a popular machine learning algorithm often used for classification and regression tasks. It is an ensemble learning algorithm that constructs multiple decision trees, each trained on a random subset of features at each split, to minimize the variance between correlated trees. By averaging the predictions of individual trees, it enhances predictive accuracy and helps mitigate overfitting, resulting in a more robust model [19].

3. RESULTS AND DISCUSSION

Let's now explore the accuracy of the results obtained through these methods and examine how each contributes to enhancing the overall performance of the delivery delay prediction system. As shown in Table 4, the logistic regression model achieves an accuracy of 0.61, indicating its ability to make correct predictions. However, the random forest model outperforms it significantly, with an accuracy of 0.98. This considerable difference suggests that the random forest model excels at identifying patterns and making precise predictions regarding delivery delays. Given these results, the random forest model has proven to be more effective for this prediction task, offering valuable insights for identifying and mitigating late deliveries.

Table 4. Accuracy metrics for delivery delay prediction models

Model	Accuracy
Logistic regression	0.61
Random forest	0.98

Next, we employed a confusion matrix to evaluate the performance of both models [20]. The confusion matrix is an N×N table, where N represents the number of target classes. It is used to compare the actual values of the target variable against the predictions made by the machine learning model. Since we are dealing with a binary classification problem [21], we used a 2×2 matrix. The outcomes of the confusion matrix for both models are presented below in Figures 4(a) and 4(b).

To further evaluate the performance of both models, we generated a classification report that provides a comprehensive view of each model's predictive capabilities as shown in Figure 5(a) and 5(b). The classification report includes key metrics such as precision, recall, F1-score, and support, which give insight into how well each model performs across different classes. These metrics not only assess overall accuracy but also highlight the model's behavior when handling imbalanced classes or more challenging predictions. By analyzing these key metrics, we can gain a deeper understanding of each model's strengths and weaknesses. This is crucial for identifying areas where the models may need improvement, particularly in cases where a higher precision or recall might be more important depending on the specific business requirements, such as minimizing false positives in delivery delays or reducing missed delays. Let's break down the key metrics in the classification report:

The classification report provides a detailed evaluation of a classification model's performance, such as logistic regression and random forest, using various metrics. Let's break down the key metrics presented in the classification report:

a. Logistic regression:
- Precision (0): measures how many instances predicted as *0* (no late delivery) were 0. It is calculated as $\frac{TP}{TP+FP}$. In this case, the precision is 0.62, meaning that 62% of the instances predicted as (no late delivery) were correctly classified as (No Late Delivery).
- Recall (0): measures how many actual *0* (No Late Delivery) instances were correctly predicted as *0*. It is calculated as $\frac{TP}{TP+FN}$. The recall for *0* is 0.71, indicating that 71% of the actual (no late delivery) instances were correctly predicted.
- F1-score (0): is the harmonic mean of precision and recall [22], providing a balance between the two. It is calculated as $\frac{2\times(Precision\times Recall)}{Precision\times Recall}$. For class *0*, the F1-score is 0.66.
- Support (0): represents the number of actual instances of class *0* in the test set. In this case, the support is 210,969.

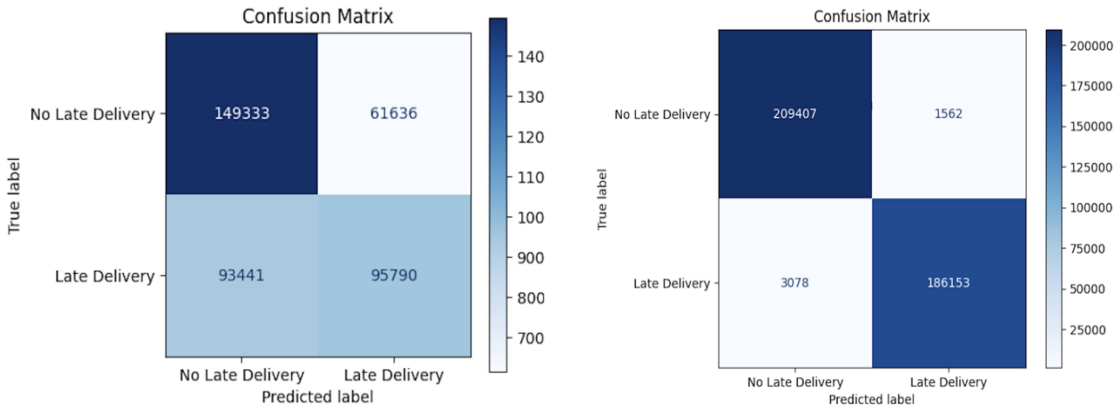

Figure 4. Confusion matrix comparison: (a) logistic regression and (b) random forest

Classification Report for Logistic Regression:

	precision	recall	f1-score	support
0	0.62	0.71	0.66	210969
1	0.61	0.51	0.55	189231
accuracy			0.61	400200
macro avg	0.61	0.61	0.61	400200
weighted avg	0.61	0.61	0.61	400200

(a)

Classification Report for Random Forest:

	precision	recall	f1-score	support
0	0.99	0.99	0.99	210969
1	0.99	0.98	0.99	189231
accuracy			0.99	400200
macro avg	0.99	0.99	0.99	400200
weighted avg	0.99	0.99	0.99	400200

(b)

Figure 5. Classification evaluating classification model performance: a comparison of (a) logistic regression and (b) random forest

Now, let's interpret the metrics for class *1* (late delivery):
- Precision (1): precision for class *1* is 0.61, meaning that among the instances predicted as (late delivery), 61% were correctly classified as (late delivery).

- Recall (1): recall for class *1* is 0.51, indicating that 51% of the actual (late delivery) instances were correctly predicted.
- F1-Score (1): the F1-score for class *1* is 0.55, balancing precision and recall for this class.
- Support (1): the support for class *1* is 189,231, representing the actual instances of (late delivery) in the test set.
- Accuracy: the overall accuracy of the logistic regression model is 0.61, meaning that the model correctly predicted the class labels [23] for 61% of the instances in the test set.
- Macro Avg: the macro average is the average of precision, recall, and F1-score for both classes, providing an overall summary of model performance across all classes [24]. In this case, the macro average is 0.61.
- Weighted Avg: the weighted average is the average of precision, recall, and F1-score, weighted by the number of instances for each class [25]. This gives a performance measure that takes class imbalances into account. In this case, the weighted average is also 0.61.

b. Random forest

The interpretation of the classification report for random forest is like that of logistic regression. However, the random forest model demonstrates exceptional performance with significantly higher precision, recall, and F1-scores for both classes (*0* and *1*). This indicates that random Forest achieved an impressive 99% accuracy in correctly classifying instances.

In summary, when comparing the two models, random Forest outperforms logistic regression across accuracy, precision, recall, and F1-score for both classes. This suggests that random Forest is more effective at classifying instances of both (no late delivery) and (late delivery) based on the given features. A good model is one with high true positive (TP) and true negative (TN) rates and low false positive (FP) and false negative (FN) rates.

The use of logistic regression and random forest algorithms to forecast late deliveries is a powerful way to address a significant challenge in the logistics industry [26], [27]. Logistic Regression is an effective method for predicting binary outcomes, such as whether a delivery will be late. By analyzing historical supply chain data, logistic regression can estimate the probability of late deliveries based on factors like prior delivery times, routes, and shipment characteristics. When combined with real-time weather data from APIs, logistic regression can further incorporate weather-related variables such as precipitation, temperature, and road conditions, offering a more comprehensive and accurate forecast. On the other hand, the random forest technique provides a more complex and robust modeling approach [28].

3. CONCLUSION

In conclusion, using logistic regression and random forest algorithms to predict late deliveries, in combination with supply chain data and weather API integration, offers a data-driven strategy with the potential to transform the logistics industry. These algorithms enable logistics professionals to anticipate and minimize disruptions by leveraging both historical data and real-time weather information. As a result, they enhance delivery reliability and customer satisfaction in an increasingly complex and unpredictable world.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Imane Zeroual	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			
Jaber El Bouhdidi	✓	✓		✓	✓			✓		✓		✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Derived data supporting the findings of this study are available from the corresponding author IZ on request.

REFERENCES

[1] W. Zhu, B. Castanier, and B. Bettayeb, "A dynamic programming-based maintenance model of offshore wind turbine considering logistic delay and weather condition," *Reliability Engineering & System Safety*, vol. 190, p. 106512, Oct. 2019, doi: 10.1016/j.ress.2019.106512.

[2] L. Bakke Lie, V. Lysgaard, and A. Kristoffer Sydnes, "Anticipating climate risk in Norwegian municipalities," *Climate Risk Management*, vol. 46, p. 100658, 2024, doi: 10.1016/j.crm.2024.100658.

[3] C. Peláez-Rodríguez, R. Torres-López, J. Pérez-Aracil, N. López-Laguna, S. Sánchez-Rodríguez, and S. Salcedo-Sanz, "An explainable machine learning approach for hospital emergency department visits forecasting using continuous training and multi-model regression," *Computer Methods and Programs in Biomedicine*, vol. 245, p. 108033, Mar. 2024, doi: 10.1016/j.cmpb.2024.108033.

[4] A. Mishra, H. R. Lone, and A. Mishra, "Decode: data-driven energy consumption prediction leveraging historical data and environmental factors in buildings," *Energy and Buildings*, vol. 307, p. 113950, Mar. 2024, doi: 10.1016/j.enbuild.2024.113950.

[5] J. Peng, D. Shen, T. Nie, and Y. Kou, "RLclean: an unsupervised integrated data cleaning framework based on deep reinforcement learning," *Information Sciences*, vol. 682, p. 121281, Nov. 2024, doi: 10.1016/j.ins.2024.121281.

[6] M. Tahir, A. Abdullah, N. I. Udzir, and K. A. Kasmiran, "A novel approach for handling missing data to enhance network intrusion detection system," *Cyber Security and Applications*, vol. 3, p. 100063, Dec. 2025, doi: 10.1016/j.csa.2024.100063.

[7] Y. Sun *et al.*, "Modifying the one-hot encoding technique can enhance the adversarial robustness of the visual model for symbol recognition," *Expert Systems with Applications*, vol. 250, p. 123751, Sep. 2024, doi: 10.1016/j.eswa.2024.123751.

[8] S. S. Dar, M. K. Karandikar, M. Z. U. Rehman, S. Bansal, and N. Kumar, "A contrastive topic-aware attentive framework with label encodings for post-disaster resource classification," *Knowledge-Based Systems*, vol. 304, p. 112526, Nov. 2024, doi: 10.1016/j.knosys.2024.112526.

[9] W. Lup Low, M. Li Lee, and T. Wang Ling, "A knowledge-based approach for duplicate elimination in data cleaning," *Information Systems*, vol. 26, no. 8, pp. 585–606, Dec. 2001, doi: 10.1016/S0306-4379(01)00041-2.

[10] N. Hou, H. Zhong, L. Sun, and L. Chen, "Enhancing acceleration data cleaning through time–frequency feature integration using deep learning," *Structures*, vol. 68, p. 107048, Oct. 2024, doi: 10.1016/j.istruc.2024.107048.

[11] Q. Han, Z. Zhao, L. Hu, and W. Gao, "Enhanced multi-label feature selection considering label-specific relevant information," *Expert Systems with Applications*, vol. 264, p. 125819, Mar. 2025, doi: 10.1016/j.eswa.2024.125819.

[12] Z. Yan, Y. Zhou, X. He, C. Su, and W. Wu, "High-dimensional expensive optimization by Kriging-assisted multiobjective evolutionary algorithm with dimensionality reduction," *Information Sciences*, vol. 691, p. 121620, Feb. 2025, doi: 10.1016/j.ins.2024.121620.

[13] X. Pan, H. Wang, M. Lei, T. Ju, and L. Bai, "A method for filling missing values in multivariate sequence bidirectional recurrent neural networks based on feature correlations," *Journal of Computational Science*, vol. 83, p. 102472, Dec. 2024, doi: 10.1016/j.jocs.2024.102472.

[14] G. Zhao *et al.*, "Enhancing interpretability of tree-based models for downstream salinity prediction: Decomposing feature importance using the Shapley additive explanation approach," *Results in Engineering*, vol. 23, p. 102373, Sep. 2024, doi: 10.1016/j.rineng.2024.102373.

[15] S. Saranya and S. Poonguzhali, "Principal component analysis biplot visualization of electromyogram features for submaximal muscle strength grading," *Computers in Biology and Medicine*, vol. 182, p. 109142, Nov. 2024, doi: 10.1016/j.compbiomed.2024.109142.

[16] A. Pathak, U. Barman, and T. S. Kumar, "Machine learning approach to detect android malware using feature-selection based on feature importance score," *Journal of Engineering Research*, vol. 13, no. 2, pp. 712–720, Jun. 2025, doi:10.1016/j.jer.2024.04.008.

[17] A. Yilmaz Adkinson *et al.*, "Assessing different cross-validation schemes for predicting novel traits using sensor data: An application to dry matter intake and residual feed intake using milk spectral data," *Journal of Dairy Science*, vol. 107, no. 10, pp. 8084–8099, Oct. 2024, doi: 10.3168/jds.2024-24701.

[18] F. Wang, H. Zhu, X. Liu, Y. Zheng, H. Li, and J. Hua, "Achieving federated logistic regression training towards model confidentiality with semi-honest TEE," *Information Sciences*, vol. 679, p. 121115, Sep. 2024, doi: 10.1016/j.ins.2024.121115.

[19] A. Balboa, A. Cuesta, J. González-Villa, G. Ortiz, and D. Alvear, "Logistic regression vs machine learning to predict evacuation decisions in fire alarm situations," *Safety Science*, vol. 174, p. 106485, Jun. 2024, doi: 10.1016/j.ssci.2024.106485.

[20] D. Valero-Carreras, J. Alcaraz, and M. Landete, "Comparing two SVM models through different metrics based on the confusion matrix," *Computers & Operations Research*, vol. 152, p. 106131, Apr. 2023, doi: 10.1016/j.cor.2022.106131.

[21] A. Fazekas and G. Kovács, "Testing the consistency of performance scores reported for binary classification problems," *Applied Soft Computing*, vol. 164, p. 111993, Oct. 2024, doi: 10.1016/j.asoc.2024.111993.

[22] S. Hinduja, T. Nourivandi, J. F. Cohn, and S. Canavan, "Time to retire F1-binary score for action unit detection," *Pattern* Recognition Letters, vol. 182, pp. 111–117, Jun. 2024, doi: 10.1016/j.patrec.2024.04.016.

[23] C. Ragupathi, S. Dhanasekaran, N. Vijayalakshmi, and A. O. Salau, "Prediction of electricity consumption using an innovative deep energy predictor model for enhanced accuracy and efficiency," *Energy Reports*, vol. 12, pp. 5320–5337, Dec. 2024, doi: 10.1016/j.egyr.2024.11.018.

[24] A. Berger and S. Guda, "Threshold optimization for F measure of macro-averaged precision and recall," *Pattern Recognition*, vol. 102, p. 107250, Jun. 2020, doi: 10.1016/j.patcog.2020.107250.

[25] J. Cheng and W. De Waele, "Weighted average algorithm: A novel meta-heuristic optimization algorithm based on the weighted average position concept," *Knowledge-Based Systems*, vol. 305, p. 112564, Dec. 2024, doi: 10.1016/j.knosys.2024.112564.

[26] X. Guo, A. Chauhan, J. Verschoor, and A. Margert, "A quality decay model with multinomial logistic regression and image-based deep learning to predict the firmness of 'Conference' pears in the downstream supply chains," *Journal of Stored Products Research*, vol. 109, p. 102450, Dec. 2024, doi: 10.1016/j.jspr.2024.102450.
[27] M. R. Ali, S. M. A. Nipu, and S. A. Khan, "A decision support system for classifying supplier selection criteria using machine learning and random forest approach," *Decision Analytics Journal*, vol. 7, p. 100238, Jun. 2023, doi: 10.1016/j.dajour.2023.100238.
[28] C. I. Allen Akselrud, "Random Forest regression models in ecology: accounting for messy biological data and producing predictions with uncertainty," *Fisheries Research*, vol. 280, p. 107161, Dec. 2024, doi: 10.1016/j.fishres.2024.107161.

BIOGRAPHIES OF AUTHORS

Imane Zeroual is a data engineer currently pursuing a Ph.D. in computer science at the research laboratory of information systems and software engineering (SIGL), University Abdelmalek Essaadi, National School of Applied Sciences-Tetouan, Morocco. She holds an engineering degree in information systems and decision support from the National School of Applied Sciences in Morocco and a master's degree in big data and machine learning from the University of Lille, France. Her research interests include machine learning, internet of things (IoT) applications, big data analytics, data engineering, and scalable data processing. She can be contacted at email: imane.zeroual1@etu.uae.ac.ma.

Jaber El Bouhdidi is an HDR professor of computer science at the SIGL Laboratory, University Abdelmalek Essaadi, National School of Applied Sciences-Tetouan-Morocco. His research interests include web semantic, multi-agents' systems, e-learning adaptive systems and big data. He has several papers in international conferences and journals. He can be contacted at email: jaber.elbouhdidi@uae.ac.ma.