IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151137

AeroPredict-X: Flight Delay Prediction System

Akshat Kumar[1], Rudrakash Agarwal[2], Varun K[3], Dr. Beenarani Manoj[4]

Student, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Vadapalani, Chennai, India[1-3]

Mentor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Vadapalani, Chennai, India[4]

Abstract: AeroPredict-X is a flight-delay prediction and analytics system built using United States flight records covering January 2024 to December 2025. The project standardizes monthly datasets and applies supervised learning to two tasks: classifying whether arrival delay reaches 15 minutes and estimating arrival-delay duration. A CatBoost-based classifier is evaluated with chronological data partitions and metrics suited to imbalanced classification, while regression performance is measured using error-based metrics. The reported test results include ROC-AUC of 0.6671 and PR-AUC of 0.3601 for classification; the displayed threshold of 0.24 gives an F1-score of 0.3805. The regression model reports a mean absolute error of 26.60 minutes and R^2 of 0.0164, indicating limited explanatory power. SHAP explanations and a Streamlit interface support interpretation, historical analytics, single predictions, and validated batch CSV scoring. AeroPredict-X is a prototype, not an operational guarantee; further validation, calibration, leakage checks, and real-time data integration are needed.

Keywords: flight delay prediction, CatBoost, classification, regression, SHAP, temporal validation, Streamlit, explainable artificial intelligence.

1. INTRODUCTION

Airline operations depend on a network of connected activities, including airport capacity, weather, flight schedules, air-traffic restrictions, airline procedures, and the availability of aircraft and crew. A disruption at one point can affect later services, leading to missed connections and additional delays. Being able to estimate delay risk in advance can help travellers and operational teams understand where uncertainty may arise.

Flight-delay prediction can be framed as both a classification and a regression task. Classification estimates whether an arrival delay will reach a defined threshold, while regression estimates the delay duration. These outputs answer different questions: a probability communicates the chance of a delay event, whereas a duration estimate attempts to quantify its size. They should therefore be interpreted separately.

AeroPredict-X uses historical flight records to build these two types of estimates. The work focuses on preparing large monthly datasets, preventing the use of information that would not be available before departure, evaluating models on later time periods, and presenting results through an interactive application. The current version is intended for pre-departure risk exploration and does not include confirmed live feeds for weather, air traffic, aircraft rotations, or airline operating conditions.

Project Objectives

The project is intended to combine the main stages of a flight-delay machine-learning workflow in one application. Its objectives are to standardize the historical flight records, check data quality, prepare predictors available before departure, train a classifier for delays of at least 15 minutes, develop a separate duration-estimation model, evaluate both tasks using time-ordered data, explain model behaviour with SHAP where supported, and provide single-flight and batch prediction features through Streamlit.

2. RELATED WORKS

Flight-delay forecasting has been explored through statistical techniques, conventional machine-learning algorithms, ensemble methods, and deep-learning approaches. These methods use historical patterns in airline, airport, route, schedule, and calendar information to estimate future outcomes. Their performance depends not only on the selected algorithm but also on the definition of the target, the information available at prediction time, and the way the data is divided for evaluation.

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.151137

Logistic regression is often used as a transparent baseline for binary prediction. Decision trees and random forests can represent non-linear relationships and interactions between features. Such baseline models are valuable because they provide a reference point against which more complex approaches can be assessed. A sophisticated model should demonstrate a measurable improvement rather than being assumed to perform better solely because of its complexity.

Gradient-boosting methods build an ensemble of decision trees in sequence, with later trees learning from errors made by earlier ones. CatBoost is designed to work with categorical inputs, which is useful for fields such as operating carrier, origin airport, destination airport, and scheduled-time categories. However, its suitability for a particular dataset must be established through validation on data that reflects the intended prediction setting.

Class imbalance is another important issue. When delayed flights form a smaller share of the dataset, accuracy alone can give an incomplete picture of model quality. ROC-AUC and precision-recall measures describe ranking behaviour across thresholds, while precision, recall, and F1-score describe the trade-off at a selected threshold. For duration prediction, MAE expresses the typical absolute error in minutes, while R-squared compares the model error with the variation in the target.

A complete prediction tool also needs to go beyond the training stage. Chronological evaluation, threshold analysis, a separate duration model, interpretable outputs, input validation, and a usable interface help make results easier to inspect. AeroPredict-X brings these elements together as a single educational workflow, while its current predictive performance remains subject to the limitations discussed later in this paper.

3. SYSTEM ARCHITECTURE

AeroPredict-X follows a sequence that starts with monthly flight-data archives and ends with predictions and evaluation views in a Streamlit application. The input files are read and standardized before quality checks are performed. The prepared records are then used to create features and targets, establish chronological training, validation, and test periods, fit the prediction models, and generate evaluation and explanation outputs. Historical analysis is kept distinct from model-generated estimates so that observed results are not confused with predictions.

System Workflow

The data-ingestion stage reads monthly ZIP archives and their CSV contents. Standardization aligns column names, date formats, and data types across files. Quality checks examine file readability, date coverage, missing target values, duplicates, invalid time fields, and cancellation or diversion indicators. Feature preparation then selects inputs that are expected to be available before departure and excludes fields that would reveal the eventual outcome.

The modelling stage creates time-ordered partitions and trains a CatBoost-based classification model for the ARR_DEL15 target, alongside a regression component for arrival-delay minutes. The classification model produces a probability that can be converted into a label using a chosen threshold. The regression model produces a duration estimate. SHAP is used to summarize feature contributions, and the application provides historical dashboards, individual predictions, batch CSV scoring, and model-performance information.

Implementation Tools

Python is used for data preparation and model execution. Pandas and NumPy support tabular processing and numerical operations, CatBoost provides the gradient-boosting model, and scikit-learn supplies evaluation utilities. SHAP is used for model explanations, while Streamlit provides the interactive interface. The project reads ZIP and CSV archives and retains processed data, trained models, configuration files, and evaluation outputs for reuse.

Key Features

- **Historical data processing:** Monthly flight archives are combined into a standardized dataset, with row counts and quality checks intended to make the preparation steps auditable.
- **Separate prediction tasks:** The system distinguishes the prediction of a 15-minute arrival-delay event from the estimation of delay duration.
- **Time-aware evaluation:** Training, validation, and test periods are separated chronologically to better represent forecasting on later records.
- **Threshold analysis:** The classification probability can be converted into an alert label using a configurable decision threshold.
- **Explainability:** Global and individual SHAP views are intended to show how model inputs contribute to predictions, without implying that those inputs cause delays.

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.151137

- **Interactive application:** Streamlit supports historical analytics, single-flight prediction, batch CSV scoring, model review, and export of prediction results.
- **Input validation:** Batch uploads are checked for required columns and invalid values, and records that fail validation are distinguished from valid predictions.

4. METHODOLOGY

Dataset Description

The project uses monthly United States flight-record archives intended to cover January 2024 through December 2025. The source report gives a standardized dataset size of 14,080,700 rows and describes an inventory of 24 monthly archives. The records include calendar information, operating carrier, origin and destination airports, scheduled departure and arrival times, scheduled elapsed time, distance, arrival-delay information, cancellation and diversion indicators, and delay-cause fields.

The principal classification target is ARR_DEL15, which represents whether arrival delay reaches 15 minutes. The regression target is ARR_DELAY, measured in minutes. These two targets are treated as separate learning tasks because a binary delay flag and a continuous duration estimate have different interpretations.

Data Ingestion and Preprocessing

The ingestion process locates monthly archives, reads the CSV files, aligns their schemas, parses dates, and converts columns into consistent data types. Source archives are retained, and row counts should be tracked before and after each transformation. Missing columns must be handled explicitly rather than filled with zero without justification. The cleaning checks include duplicate records, missing target labels, inconsistent dates, invalid time values, and cancellation or diversion status.

Duplicate handling should follow a documented rule. Exact duplicate rows need to be distinguished from separate records that happen to share a possible flight identifier. This distinction is important when combining monthly files, since an overly broad duplicate rule could remove legitimate observations.

Target and Feature Selection

For classification, ARR_DEL15 equal to 1 represents an arrival delay of at least 15 minutes, while 0 represents a delay below that threshold, subject to confirmation against the source data definition. ARR_DELAY is used as the continuous regression target. The training setup for regression should clearly state whether it includes all operated flights or only delayed flights, since those choices lead to different interpretations of the estimated duration.

Candidate pre-departure inputs include year, month, day of month, day of week, operating carrier, origin, destination, scheduled departure time, scheduled departure-time block, scheduled arrival time, scheduled arrival-time block, scheduled elapsed time, and distance. Actual arrival or departure times, the target columns themselves, and delay-cause fields recorded after the outcome must not be used as predictors for a pre-departure estimate. Any historical rate or derived feature must be calculated using information available before the prediction time.

Chronological Data Splitting

The records are separated by time rather than randomly. The intended training period is January to December 2024. January to June 2025 is used for validation and decisions such as model selection and threshold choice. July to December 2025 is held back for final testing. The held-out period should not be used to select features, tune hyperparameters, or choose the operational threshold.

Model Training and Evaluation

The classification workflow uses a CatBoost-based model. Model settings are selected using the validation period, with early stopping or another documented selection procedure where applicable. Categorical features must be treated consistently in training and prediction. Model files, feature definitions, configuration values, random seeds, and evaluation outputs should be preserved so that the results can be reproduced. The report records a displayed decision threshold of 0.24; this value should be checked against the final saved configuration.

Classification is assessed using ROC-AUC, PR-AUC or average precision, precision, recall, F1-score, Brier score, confusion-matrix results, and accuracy where appropriate. Regression is assessed using MAE, median absolute error, RMSE, and R-squared. A baseline should be evaluated on the same test observations, and any transformation of the regression target must be reversed before errors are reported in minutes. Validation results and final test results should be kept separate.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151137

Explainability and Batch Prediction

SHAP explanations are generated for a manageable sample of records. Global summaries describe which inputs contribute most across the sample, while local explanations show the contribution pattern for an individual prediction. These results describe model behaviour; they do not demonstrate that a feature causes a flight delay. The explanation process should use the same saved model as the prediction and should exclude information that would not be available at prediction time.

For batch prediction, the application accepts CSV files and checks required fields, missing values, airport and carrier codes, date and time formats, and numeric ranges. Rows that fail validation should be marked clearly and should not be assigned misleading predictions. Exported files should contain the prediction results and validation status without an unintended dataframe index column.

5. RESULTS AND DISCUSSION

The values below are reported in the AeroPredict-X project report. They should be checked against the final saved model configuration and JSON or CSV evaluation files before formal submission. They are presented as reported results and have not been independently re-computed as part of this document reformatting.

Dataset Processing Results

The project report records 14,080,700 standardized flight rows and 24 monthly source archives for the intended January 2024 to December 2025 coverage. It also reports that the Streamlit dashboard and prediction pages were displayed and that ten automated tests passed. The date coverage, row count, and test status should be supported by the final processed-data inventory and retained test output.

Classification Results

On the reported held-out test data, the classifier achieved a ROC-AUC of 0.6671 and a PR-AUC or average-precision value of 0.3601. The reported Brier score was 0.1667. At a displayed threshold of 0.24, the F1-score was 0.3805, precision was 0.3736, recall was 0.3877, and accuracy was 0.7126.

These values suggest that the classifier contains some ranking information, but they do not support describing it as highly accurate or ready for operational deployment. At the selected threshold, precision indicates that a substantial share of alerts may be false positives, while recall indicates that many delayed flights may not be identified. The preferred threshold therefore depends on the cost of missed delays compared with the cost of unnecessary alerts. Threshold selection should use validation data, leaving the final test results unchanged.

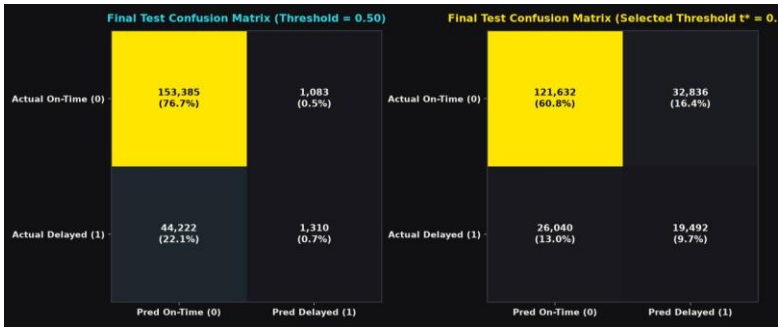


Fig. 1. Final Test Confusion Matrices at Default and Selected Classification Thresholds.

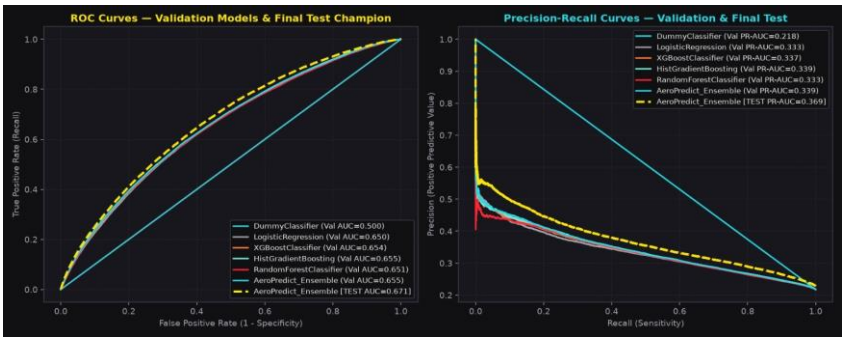


Fig. 2. ROC and Precision–Recall Curves Comparing Validation Models and the Final AeroPredict-X Test Model.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151137

Regression Results

The reported mean absolute error for arrival-delay duration is 26.60 minutes, and the reported R-squared value is 0.0164. The R-squared result indicates that the current regression model explains only a small part of the variation in delay duration. The MAE gives an interpretable measure of typical absolute error, but it should be compared with a simple baseline evaluated on the same test records. RMSE and median absolute error were not confirmed in the source report and should only be added after checking the saved evaluation output.

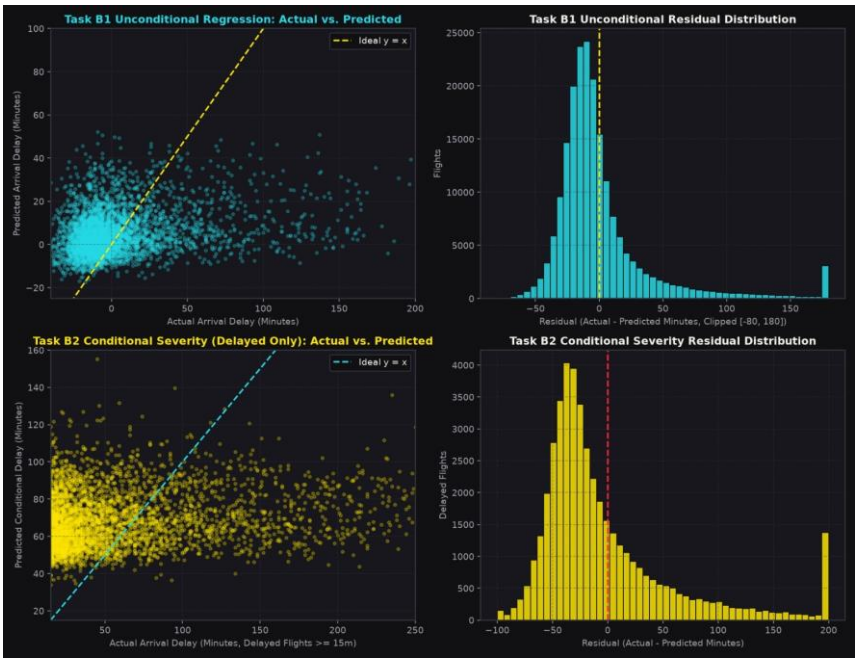


Fig. 3. Actual-versus-Predicted Arrival Delay and Residual Distributions for Unconditional and Conditional Regression Models.

Explainability and Application Behaviour

The application is designed to provide global and individual SHAP explanations, historical analytics, individual-flight predictions, batch CSV scoring, and model-performance review. The source report does not provide enough specific SHAP findings to support claims about which individual features were most influential, so no feature ranking is asserted here. Batch validation is intended to identify malformed files and invalid rows before the valid records are scored.

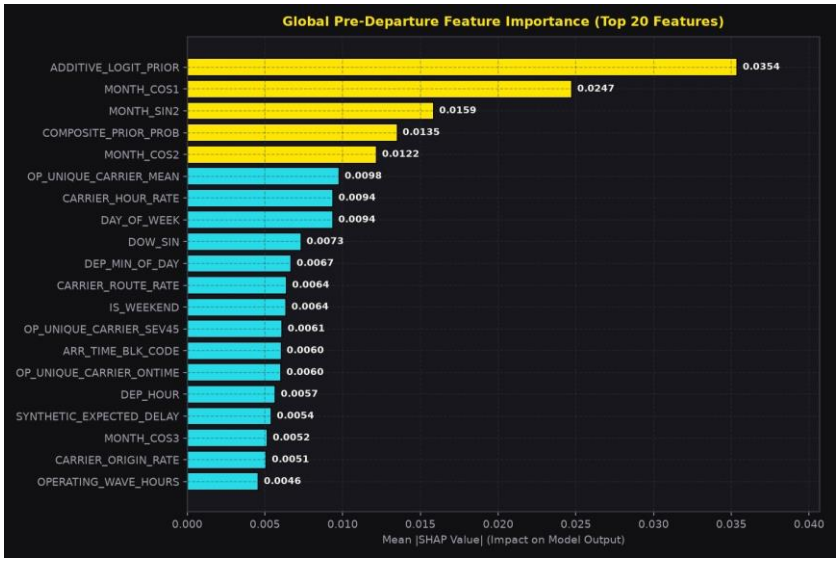


Fig. 4. Global Pre-Departure Feature Importance Based on Mean Absolute SHAP Values.



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.151137

6. CONCLUSION

AeroPredict-X brings together a historical flight-data pipeline, machine-learning models, evaluation procedures, explainability, and a Streamlit interface for flight-delay analysis. It uses a CatBoost-based classifier to estimate whether an arrival delay reaches 15 minutes and a separate regression component to estimate delay duration. The workflow also includes schema standardization, data-quality checks, chronological data partitions, individual and batch prediction, and SHAP-based interpretation.

The reported classification results are ROC-AUC 0.6671 and PR-AUC 0.3601, with an F1-score of 0.3805 at a displayed threshold of 0.24. The regression component reports an MAE of 26.60 minutes and an R-squared value of 0.0164, showing that the current duration model has limited explanatory strength. These values must be confirmed against the final saved evaluation artifacts. Overall, the project demonstrates an integrated and inspectable machine-learning application, but its present results support educational and exploratory use rather than dependable operational forecasting.

REFERENCES

[1]. Gui, G., Liu, F., Sun, J., Yang, J., Zhou, Z., and Zhao, D. (2020). "Flight Delay Prediction Based on Aviation Big Data and Machine Learning." IEEE Transactions on Vehicular Technology, vol. 69, no. 1, pp. 140–150. https://doi.org/10.1109/TVT.2019.2954094
[2]. Yu, B., Guo, Z., Asian, S., Wang, H., and Chen, G. (2019). "Flight Delay Prediction for Commercial Air Transport: A Deep Learning Approach." Transportation Research Part E: Logistics and Transportation Review, vol. 125, pp. 203–221. https://doi.org/10.1016/j.tre.2019.03.013
[3]. Li, Q. and Jing, R. (2021). "Generation and Prediction of Flight Delays in Air Transport." IET Intelligent Transport Systems, vol. 15, no. 6, pp. 740–753. https://doi.org/10.1049/itr2.12057
[4]. Dai, M. (2024). "A Hybrid Machine Learning-Based Model for Predicting Flight Delay Through Aviation Big Data." Scientific Reports, vol. 14, article 4603. https://doi.org/10.1038/s41598-024-55217-z
[5]. Alfarhood, M., Alotaibi, R., Abdulrahim, B., Einieh, A., Almousa, M., and Alkhanifer, A. (2024). "Predicting Flight Delays with Machine Learning: A Case Study from Saudi Arabian Airlines." International Journal of Aerospace Engineering, vol. 2024, article 3385463. https://doi.org/10.1155/2024/3385463
[6]. Beltman, M., Ribeiro, M., de Wilde, J., and Sun, J. (2025). "Dynamically Forecasting Airline Departure Delay Probability Distributions for Individual Flights Using Supervised Learning." Journal of Air Transport Management, vol. 126, article 102788. https://doi.org/10.1016/j.jairtraman.2025.102788
[7]. Wandelt, S., Chen, X., and Sun, X. (2025). "Flight Delay Prediction: A Dissecting Review of Recent Studies Using Machine Learning." IEEE Transactions on Intelligent Transportation Systems, vol. 26, no. 4, pp. 4283–4297. https://doi.org/10.1109/TITS.2025.3528536
[8]. Breiman, L. (2001). "Random Forests." Machine Learning, vol. 45, pp. 5–32. https://doi.org/10.1023/A:1010933404324
[9]. Chen, T. and Guestrin, C. (2016). "XGBoost: A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, USA, pp. 785–794. https://doi.org/10.1145/2939672.2939785
[10]. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. (2018). "CatBoost: Unbiased Boosting with Categorical Features." Advances in Neural Information Processing Systems, vol. 31.
[11]. Lundberg, S. M. and Lee, S.-I. (2017). "A Unified Approach to Interpreting Model Predictions." Advances in Neural Information Processing Systems, vol. 30, pp. 4765–4774.
[12]. Lundberg, S. M. et al. (2020). "From Local Explanations to Global Understanding with Explainable AI for Trees." Nature Machine Intelligence, vol. 2, pp. 56–67. https://doi.org/10.1038/s42256-019-0138-9
[13]. Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python." Journal of Machine Learning Research, vol. 12, pp. 2825–2830.
[14]. U.S. Bureau of Transportation Statistics. "Airline On-Time Statistics and Delay Causes." TranStats Database. Available at: https://www.transtats.bts.gov/
[15]. CatBoost. "CatBoost Documentation." Available at: https://catboost.ai/docs/
[16]. Scikit-learn Developers. "Model Evaluation: Quantifying the Quality of Predictions." Scikit-learn Documentation. Available at: https://scikit-learn.org/
[17]. Scikit-learn Developers. "TimeSeriesSplit." Scikit-learn Documentation. Available at: https://scikit-learn.org/
[18]. Streamlit. "Streamlit Documentation." Available at: https://docs.streamlit.io/
[19]. SHAP Developers. "SHAP Documentation." Available at: https://shap.readthedocs.io/

 This work is licensed under a Creative Commons Attribution 4.0 International License