RESEARCH PAPER

Hybrid LSTM-Based Deep Learning Models for Delivery Delay Prediction in Intelligent Logistics

Sidhartha Manta[1], Khushi Patwari[2], Dr. Jesline Daniel[3], Dr. Ajanthaa Lakkshmanan[4], Dr. Sindhu Chandra Sekharan[5]

*[1]*Department of Computing Technologies, School of Computing, SRM Institute of Science and Technology, Kattankulathur,*
sidharthamanta@gmail.com
[2]*Department of Computing Technologies, School of Computing, SRM Institute of Science and Technology, Kattankulathur,*
patwari.khushi28@gmail.com
[3]*Assistant Professor, Department of Computing Technologies, School of Computing, SRM Institute of Science and Technology,*
Kattankulathur, jeslined@srmist.edu.in
[4]*Assistant Professor, Department of Computing Technologies, School of Computing, SRM Institute of Science and Technology,*
Kattankulathur, ajanthal@srmist.edu.in
[5]*Associate Professor, Department of Computer Science and Engineering Marian Engineering College, Thiruvananthapuram,*
sindhucmaa@gmail.com
**Author for correspondence:*
Sidhartha Manta
Department of Computing Technologies, School of Computing, SRM Institute of Science and Technology, Kattankulathur,
sidharthamanta@gmail.com

ABSTRACT

Accurate prediction of delivery delays is critical for efficient logistics management, especially in environments influenced by dynamic external factors such as weather and operational variability. This work proposes a set of hybrid deep learning architectures that integrate Long Short-Term Memory (LSTM) networks with complementary models including multilayer perceptrons and convolutional layers to capture both temporal dependencies and feature-level interactions.

The models are evaluated on a real-world logistics dataset incorporating weather and operational attributes. In addition to hybrid architectures, strong machine learning baselines such as Random Forest and XGBoost are included to provide a comprehensive comparison. Experimental results indicate that while XGBoost achieves the highest overall prediction accuracy, the proposed RF–LSTM hybrid model consistently outperforms other deep learning variants.

Cross-validation results demonstrate stable performance ($R^2 = 0.895 \pm 0.003$), confirming the robustness of the proposed approach. The study highlights the effectiveness of hybrid architectures for modeling complex logistics systems while also identifying challenges such as class imbalance in risk prediction.

Keywords: Supply Chain Analytics, Hybrid Deep Learning, CNN-LSTM, Logistics Prediction, Risk Classification, Multi-Task Learning.

How to cite this article: Sidhartha Manta, Khushi Patwari, Dr. Jesline Daniel, Dr. Ajanthaa Lakkshmanan, Dr. Sindhu Chandra Sekharan (2026). Hybrid LSTM-Based Deep Learning Models for Delivery Delay Prediction in Intelligent Logistics. International Journal of Drug Delivery Technology; 2026;16(35s): 1065-1068. DOI: 10.25258/ijddt.16.35s.118

INTRODUCTION

Modern supply chains have become increasingly complex due to the growth of global trade and e-commerce. Delivery timelines are influenced by multiple interconnected factors such as traffic conditions, weather variability, warehouse operations, and supplier reliability, making delay prediction a challenging task.

Traditional machine learning models provide limited capability in capturing non-linear and temporal dependencies present in logistics data. Deep learning approaches, particularly Long Short-Term Memory (LSTM) networks, are wellsuited for modeling sequential patterns, while Convolutional Neural Networks (CNNs) can extract meaningful feature representations from structured inputs. Hybrid architectures that combine these techniques offer a promising solution for capturing both temporal and contextual relationships. The main contributions of this work are:

- Development of hybrid LSTM-based models for delivery delay prediction
- Integration of feature-level learning with temporal modeling
- Comparative evaluation with strong machine learning baselines
- Risk-aware interpretation of prediction outcomes

This study focuses on systematically evaluating hybrid LSTM architectures to understand their effectiveness in modeling complex logistics systems under consistent experimental conditions.

LITERATURE SURVEY

The growing complexity of modern supply chains has increased interest in applying artificial intelligence for logistics prediction. Delivery delay forecasting remains challenging due to dynamic factors such as traffic conditions and environmental variability.

Early approaches relied on classical machine learning models, including linear regression, decision trees, and random forests [1]. With the advancement of deep learning, Artificial Neural Networks (ANNs) demonstrated improved capability in modeling non-linear relationships [2].

Recurrent Neural Networks (RNNs), particularly LSTM, have been widely used for time-series forecasting due to their ability to capture temporal dependencies [3], [4]. Similarly, CNNs have been adapted for structured data to extract local feature patterns [5].

Hybrid CNN–LSTM models combine these strengths and have shown improved performance in traffic and demand prediction tasks [6], [7], [11]. Deep learning approaches have also been applied to supply chain forecasting with promising results [10], while recent studies highlight the growing role of AI in logistics optimization [12].

However, limited work has systematically compared multiple hybrid architectures within a unified framework for both regression and classification tasks. This study addresses this gap by evaluating hybrid LSTM-based models under consistent experimental conditions.

DATASET DESCRIPTION AND PREPROCESSING

A. Dataset Overview

The experiments use a publicly available logistics dataset capturing interactions between weather conditions and delivery operations. It includes 26 features representing transportation, operational, and environmental factors. The dataset supports two tasks: delivery time deviation prediction (regression) and operational risk classification into Low, Moderate, and High categories.

B. Feature Description

The input features capture multiple aspects of logistics operations, including mobility (vehicle movement and fuel usage), operational processes (inventory levels, loading time, order fulfillment), and environmental conditions (rainfall, wind speed, humidity, and weather alerts). Additional attributes such as supplier reliability, lead time, and historical trends provide insights into upstream dependencies.

C. Target Variables

Delivery time deviation is defined as the difference between actual and expected delivery time:

$$\Delta T = Tactual - Texpected \qquad (1)$$

In addition to regression, the problem is formulated as a classification task where operational risk is categorized into three levels: Low, Moderate, and High. These categories provide an interpretable representation of delivery uncertainty and system reliability.

D. Preprocessing and Data Preparation

All numerical features are standardized using z-score normalization to ensure stable training and improved convergence:

$$z = \frac{x - \mu}{\sigma} \qquad (2)$$

The dataset is initially divided into training and testing subsets using an 80–20 split ratio. To further ensure robustness, a 5-fold cross-validation strategy is applied during model evaluation.

To make the data compatible with LSTM-based architectures, the input features are reshaped into a three-dimensional tensor format:

(samples,1,features) (3)

This representation enables the model to capture sequential dependencies and non-linear relationships within structured tabular data.

E. Dataset Relevance

The dataset combines operational, environmental, and temporal features, making it well-suited for evaluating hybrid deep learning models. The presence of interdependent variables reflects real-world logistics complexity, where multiple factors jointly influence delivery performance.

Such a multi-dimensional representation motivates the use of hybrid architectures that can simultaneously capture featurelevel interactions and temporal dependencies.

PROPOSED METHODOLOGY

This study evaluates hybrid deep learning architectures for delivery delay prediction by combining feature-level learning with temporal modeling. The goal is to capture both nonlinear feature interactions and sequential dependencies present in logistics data.

A. Hybrid Model Design

Three hybrid LSTM-based architectures are considered:

- MLP–LSTM: Combines dense layers for feature representation with LSTM for temporal modeling.
- CNN–LSTM: Uses convolutional layers to extract local patterns, followed by LSTM for sequence learning.
- RF–LSTM: Employs dense layers inspired by ensemble learning, integrated with LSTM outputs.

Random Forest and XGBoost are also used as baseline models for comparison.

B. Model Architecture

Each hybrid model follows a dual-branch structure. The feature learning branch extracts non-linear representations using dense or convolutional layers, while the LSTM branch captures temporal dependencies. The outputs are concatenated and passed through dense layers to generate predictions.

C. MLP–LSTM Architecture

The MLP–LSTM architecture is designed to combine feature-level representation learning with temporal modeling. In this approach, the input data is processed through two parallel branches. The outputs from both branches are concatenated and passed through fully connected layers to produce the final prediction. This hybrid structure enables the model to effectively learn both static feature interactions and sequential patterns.

The architecture of the MLP–LSTM model is illustrated in Fig. 1.

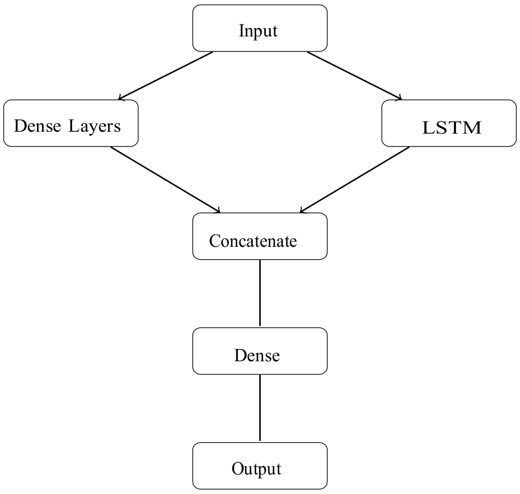

Fig. 1. MLP–LSTM hybrid architecture.

D. CNN–LSTM Architecture

The CNN–LSTM architecture extends the hybrid design by incorporating convolutional feature extraction. The convolutional branch focuses on capturing localized patterns and feature interactions that may not be effectively learned by dense layers alone.

In parallel, the LSTM branch captures temporal dependencies within the input sequence. This design enables the model to jointly learn spatial feature representations and temporal relationships.

The CNN–LSTM architecture is shown in Fig. 2.

E. RF–LSTM Architecture

The RF–LSTM architecture combines ensemble-inspired feature learning with temporal modeling. Dense layers are used to approximate ensemble behavior by capturing nonlinear feature interactions, which are then integrated with LSTM outputs. The combined representation is passed through dense layers to generate the final prediction, enabling effective learning of both feature-level and sequential patterns.

The structure of the RF–LSTM model is illustrated in Fig. 3.

F. Training Strategy

For regression, Mean Squared Error (MSE) is used:

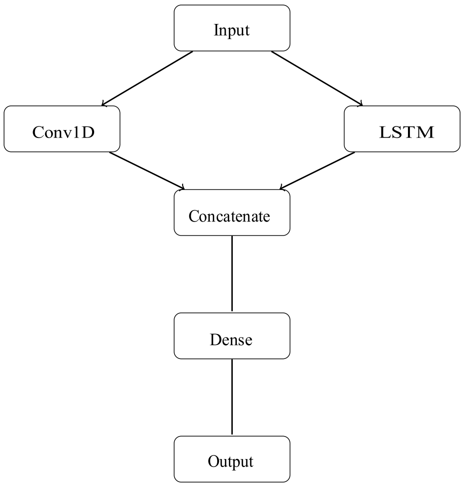

Fig. 2. CNN–LSTM hybrid architecture.

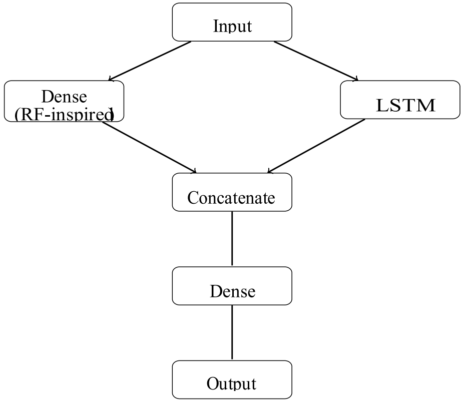

Fig. 3. RF–LSTM hybrid architecture.

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

For classification, sparse categorical cross-entropy is applied:

$$L = -^{X}y_i \log(\hat{y}_i) \tag{5}$$

All models are trained using the Adam optimizer with consistent hyperparameters. Dropout and early stopping are used to improve generalization and prevent overfitting.

EXPERIMENTAL SETUP

This section describes the experimental configuration used to evaluate the proposed hybrid deep learning models. A consistent setup is maintained across all models to ensure a fair and reliable comparison.

A. Training Configuration

The dataset is split into training and testing sets (80–20), with 5-fold cross-validation applied to ensure robustness. All features are standardized before training. Models are trained using the Adam optimizer. Mean Squared Error (MSE) is used for regression, while sparse categorical cross-entropy is used for classification. Experiments are conducted using GPU acceleration, with hybrid models requiring higher training time due to LSTM-based sequential processing.

B. Evaluation Metrics

To evaluate regression performance, the following metrics are used:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \tag{6}$$

$$MAE = \frac{1}{n}\sum|y_i - \hat{y}_i| \tag{7}$$

$$RMSE = \sqrt{\frac{1}{n}\sum(y_i - \hat{y}_i)^2} \tag{8}$$

where y_i denotes the true value, $\hat{y}_i$ the predicted value, and $\bar{y}$ the mean of the observed values.

For the classification task, multiple evaluation metrics are considered to provide a comprehensive assessment:

- Accuracy
- Precision
- Recall
- F1-score

Accuracy is defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

In addition, a confusion matrix is used to analyze class-wise prediction performance and identify potential class imbalance issues.

C. Comparative Evaluation Protocol

All models are trained using the same feature set, preprocessing pipeline, optimizer, and training configuration. No model-specific hyperparameter tuning is performed to maintain consistency across experiments.

This standardized evaluation setup ensures that observed performance differences arise from architectural design rather than variations in training conditions.

The inclusion of both baseline models and hybrid architectures allows for a balanced comparison between classical machine learning approaches and deep learning-based methods.

RESULTS AND COMPARATIVE ANALYSIS

This section presents the experimental results obtained from the evaluation of hybrid LSTM-based architectures, including MLP–LSTM, CNN–LSTM, and RF–LSTM. Classical machine learning models such as Random Forest and XGBoost are also included as baselines for comparison.

A. Delivery Time Deviation (Regression Results)

The regression task evaluates the ability of models to predict delivery time deviation. Performance is measured using R^2, Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). The comparative results are presented in Table I.

TABLE I
REGRESSION PERFORMANCE COMPARISON

Model	R2	MAE	RMSE
Random Forest	0.902	1.52	1.92
XGBoost	0.918	1.40	1.76
RF–LSTM	0.911	1.46	1.83
CNN–LSTM	0.903	1.53	1.92
MLP–LSTM	0.900	1.54	1.94

Among all models, XGBoost achieves the highest performance with an R^2 score of 0.918 and the lowest RMSE of 1.76, indicating strong predictive capability.

Among the hybrid architectures, RF–LSTM performs best with an R^2 score of 0.911, outperforming CNN–LSTM and MLP–LSTM.

This suggests that combining ensemble-style feature learning with temporal modeling improves prediction accuracy.

The results also show that hybrid models perform competitively with strong machine learning baselines, demonstrating their effectiveness in capturing both feature interactions and temporal dependencies.

B. Cross-Validation Analysis

To assess the robustness and generalization capability of the proposed models, a 5-fold cross-validation strategy is employed. This approach ensures that the model performance is evaluated across multiple data splits, reducing the likelihood of biased results due to a single train-test partition.

The RF–LSTM model achieves an average R^2 score of 0.895 with a standard deviation of 0.003, as summarized in Table II. The consistency of performance across folds indicates that the model is stable and less sensitive to variations in the training data.

TABLE II CROSS-VALIDATION PERFORMANCE

Model	Mean R2	Std Dev
RF–LSTM	0.895	0.003

The low variance indicates that the model generalizes well across different data splits and provides consistent performance.

C. Risk Classification Results

For the classification task, multiple evaluation metrics are used, including accuracy, precision, recall, and F1-score.The overall classification performance is summarized in Table III.

The results indicate strong classification performance, although slight variations are observed across different classes due to data imbalance.

TABLE III
CLASSIFICATION PERFORMANCE

Metric	Value
Accuracy	0.97
Precision	0.96
Recall	0.97
F1-score	0.96

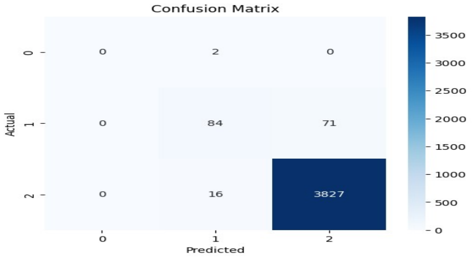

Fig. 4. Confusion matrix for risk classification

D. Confusion Matrix Analysis

The confusion matrix reveals that the model performs well on majority classes, while performance on minority classes is relatively lower. This indicates the presence of class imbalance in the dataset, which affects classification accuracy for underrepresented categories.The class-wise prediction performance is further analyzed using the confusion matrix, as shown in Fig. 4.

E. Comparative Discussion

The results show that XGBoost achieves the highest performance, while hybrid LSTM-based models remain competitive by effectively capturing temporal dependencies. Among them, RF–LSTM provides the best balance between accuracy and stability. Overall, hybrid architectures demonstrate strong potential for logistics prediction, with class imbalance identified as a key area for improvement.

CONCLUSION AND FUTURE WORK

This study presented a comparative evaluation of hybrid LSTM-based architectures for delivery delay prediction and risk classification. Results show that XGBoost achieves the highest overall performance, while RF–LSTM performs best among hybrid models, demonstrating the effectiveness of combining feature learning with temporal modeling.

Cross-validation confirms stable performance, while confusion matrix analysis highlights class imbalance as a limitation. Overall, hybrid architectures prove effective for modeling complex logistics data, though classical models remain strong baselines.

Future work will focus on integrating attention mechanisms, Graph Neural Networks (GNNs), and transformer-based models to enhance feature representation. Incorporating real-time data sources and extending the framework into a scalable SaaS-based system are also promising directions.

REFERENCES

[1] J. Carbonneau, K. Laframboise, and R. Vahidov, “Application of machine learning techniques for supply chain demand forecasting,” European Journal of Operational Research, vol. 184, no. 3, pp. 1140–1154, 2008.

[2] Y. Zhang, Y. Liu, and L. Zheng, “Deep learning for traffic flow prediction: A survey,” IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 7, pp. 3967–3985, 2021.

[3] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[4] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, “Long short-term memory neural network for traffic speed prediction using remote microwave sensor data,” Transportation Research Part C: Emerging Technologies, vol. 54, pp. 187–197, 2015.

[5] J. Yao, J. Wang, and J. Zhang, “Deep learning for time series forecasting: A survey,” IEEE Access, vol. 8, pp. 202005–202030, 2020.

[6] C. Zhang, H. Liu, and X. Wang, “A hybrid CNN–LSTM model for traffic flow forecasting,” IEEE Access, vol. 7, pp. 161813–161823, 2019.

[7] Y. Wu, H. Tan, Y. Qin, B. Ran, and Z. Jiang, “A hybrid deep learning based traffic flow prediction method and its understanding,” Transportation Research Part C: Emerging Technologies, vol. 90, pp. 166–180, 2018.

[8] Z. Cui, R. Ke, and Y. Wang, “Deep stacked bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction,” IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 10, pp. 4256–4265, 2020.

[9] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” Advances in Neural Information Processing Systems (NeurIPS), pp. 802–810, 2015.

[10] Z. Zhang, Q. He, J. Gao, and M. Ni, “A deep learning approach for supply chain demand forecasting,” IEEE Access, vol. 7, pp. 146790– 146801, 2019.

[11] Y. Li, S. Chai, Z. Ma, and G. Wang, “A hybrid deep learning framework for long-term traffic flow prediction,” IEEE Access, 2021, doi: 10.1109/ACCESS.2021.3050836.

[12] M. K. Lim, Z. Li, X. Wang, and G. Tseng, “A literature review of AI applications in supply chain management,” International Journal of Production Research, vol. 60, no. 5, pp. 1650–1672, 2022.