

Airline Ticket Price Prediction with Deep Learning Methods

1st Kutay Eroğlu

dept. of Computer Engineering

Boğaziçi University

İstanbul, Turkey

kutay.eroglu@std.bogazici.edu.tr

Abstract—Airfare pricing remains a challenging issue for both airlines and travelers due to the dynamic interplay of fluctuating demand, limited seat availability, and competitive market pressures. While airlines utilize sophisticated revenue management systems to optimize pricing strategies, travelers face difficulties in identifying the ideal time to purchase tickets. This study proposes a hybrid long short-term memory and convolutional neural network model designed to predict ticket prices by leveraging historical pricing trends and time-series data. The model’s performance was evaluated using various prediction horizons. Results indicate that while the model captures sequential dependencies and short-term pricing trends to some degree, it faces challenges in generalizing to broader contexts compared to simpler baseline models such as Random Forest Regressors. Key insights were drawn regarding the relationship between prediction horizon and prediction accuracy, as well as the trade-offs between training complexity and model performance. By addressing these challenges, this research aims to enhance price prediction accuracy for travelers.

Index Terms—Long short-term memory, Convolutional Neural Network, Random Forest, Time series data, Remaining days to departure (ndo), Prediction horizon

I. INTRODUCTION

Balancing airfare pricing is a critical issue for both travelers and airlines. Passengers strive to purchase tickets at the lowest possible cost, while airlines aim to maximize overall revenue and profits. This interplay is complicated by frequent mismatches between available seats and fluctuating demand, which can result in travelers paying more than necessary or airlines missing out on potential revenue. Although many airlines possess sophisticated tools and strategies to guide their pricing decisions, customers have also become more adept at comparing ticket costs through various online platforms [1]. The intense competition among airlines further contributes to the complexity of determining optimal ticket prices.

Over the past two decades, growing research has targeted both the consumer and airline perspectives. From the customer’s viewpoint, the primary challenge lies in pinpointing the best time to purchase a ticket at the lowest price. Contrary to the older notion that buying tickets earlier always guarantees a cheaper price [2], numerous factors can cause ticket prices to change unpredictably. In addition, purchasing tickets early introduces the risk of schedule inflexibility and possible change fees. Consequently, advanced techniques ranging from statistical methods like regression to data mining approaches

have been explored to support travelers in identifying ideal purchase times and to help airlines implement dynamic pricing strategies that safeguard profitability.

This study builds upon these advancements by proposing a hybrid long short-term memory (LSTM) and convolutional neural network (CNN) model designed to predict ticket prices based on historical trends and time-series data. Unlike traditional regression-based methods, the proposed model leverages the sequential nature of ticket pricing data, capturing both short-term fluctuations and long-term trends. By examining key factors such as the length of historical data used for prediction and the remaining days until departure, this research provides actionable insights into optimizing airfare pricing strategies. The findings aim to benefit travelers by improving price prediction accuracy.

II. LITERATURE REVIEW

Several studies have examined ticket price forecasting from different methodological angles, illustrating various ways in which researchers tackle the challenge of predicting airfare dynamics:

- Y. Chen et al. [3] introduce a model specifically tailored to predict the lowest future prices for a given route and departure date, focusing on consecutive days until takeoff. Although it leverages a modified ensemble-learning algorithm and performs iterative predictions based on prior outcomes, it is restricted to non-stop routes and cannot generate a price forecast for one specific flight segment.
- Anastasia Lantseva et al. [4] put forward an empirical, data-driven Regression Model that forecasts the per-kilometer cost for flights within 90 days prior to departure. Drawing data from two sources—AviaSales and Sabre—their investigation differentiates between local Russian flights (originating from Moscow and Saint-Petersburg to 50 domestic cities) and international flights (same origin cities but to 40 international destinations). The authors emphasize the advantage of early purchase for international flights; however, the study offers less conclusive results for local flights. This work does not report model performance metrics and employs data collected over a limited timeframe on a restricted set of routes.

- T. Janssen [5] explores a linear quantile mixed regression model to predict the minimum ticket price within 60 days of departure. Instead of analyzing all price observations, the model focuses on the lower price quantile to determine the anticipated minimum fare. Although the method works well for shorter windows, it loses accuracy as the departure date gets farther away. Data were collected for a single route, covering 2271 flights and yielding 126,412 records, which somewhat limits the method’s broader applicability.
- K. Tziridis et al. [6] compare eight top-performing machine learning regression models, including various neural networks, ensemble trees, and support vector machines. The study identifies Bagging Regression Trees as the most accurate (87.42%), closely followed by Random Forest (85.91%). They also investigate key factors influencing pricing—such as departure time, number of stops, and day of the week—using a dataset of 1,814 flights from one international route.
- T. Liu et al. [7] propose another ensemble method designed to estimate the lowest ticket prices from purchase up to departure date. They blend multiple base learners (k-Nearest Neighbors, Random Forest, and Bayesian approaches) and use feature clustering to adapt the model dynamically. The resulting mean absolute percentage error (MAPE) improved significantly (from around 7–12% to approximately 3.7–6%). Their dataset spans 19 routes over 92 days and integrates information on historical fares, holidays, and days before departure.
- William Groves and Maria Gini [8] present a stacked prediction model that merges outputs from Random Forests and Multilayer Perceptrons, identified as their two top-performing algorithms based on R-squared, MAE, and MSE metrics. Tested on 51,000 records from three domestic airlines covering round-trip, non-stop tickets, the stacked approach outperforms each individual algorithm by 4.4% and 7.7% on the R-squared measure. However, international and multi-stop flights fall outside the scope of this study.
- Boruah A. et al. [9] employ a Bayesian estimation strategy via the Kalman filter to predict ticket prices, using linear system modeling. Observations of prior fares serve as inputs that help the algorithm refine its estimates iteratively.
- Yuling Li and Zhichao Li [10] develop an airfare prediction system that merges the ARMA (AutoRegressive Moving Average) approach with Random Forest. Implemented in Python and underpinned by a SQL Server database, their method aims to capture both time-series trends and more complex, nonlinear relationships in ticket prices.
- Du et al. [11] achieved success in predicting airline ticket prices using a hybrid LSTM-CNN model. The model’s high performance was attributed to the LSTM’s capability to learn long-term patterns in time-series data. By leveraging historical price data from five distinct

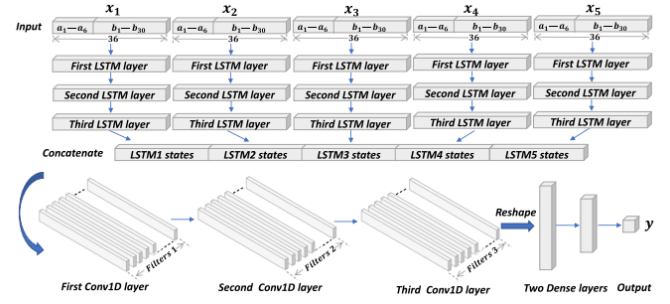


Fig. 1. The architecture of hybrid LSTM-CNN model proposed by Du et al. [11]. (reprinted from Du et al. [11]).

routes, the model was able to predict ticket prices for a single target route, as shown in Figure 1. The LSTM’s sequential processing and memory cells enabled it to effectively capture both short-term fluctuations and long-term dependencies, resulting in accurate predictions even when dealing with complex temporal data.

Collectively, these works underscore a variety of approaches, ranging from conventional linear models to ensemble and hybrid systems that tackle price forecasting with different scopes, datasets, and performance metrics. Each highlights unique modeling features, data constraints, and methodological insights that inform ongoing research in airfare prediction.

Based on the methods introduced by Du et al. [11], this research adopts LSTM to harness its ability to capture long-term dependencies in time-series data. However, it diverges from their approach in several significant ways. First, the data source utilized in this study differs, leading to variations in attributes and introducing challenges in extracting reliable temporal data. Additionally, rather than employing five parallel LSTM models each trained on a separate flight route to predict prices on only one route, as Du et al. [11] did, this work opts for a single LSTM layer to predict for one route, to enable faster experimentation cycles.

III. DATASET

A. Data Acquisition and Preparation

The publicly available Kaggle dataset entitled “Flight Prices” [12] is used, containing one-way flight data from Expedia between April 16, 2022, and October 5, 2022, for 16 airports in the United States. This dataset is substantial—31.09 GB in size, comprising 82 million records that represent roughly 6 million distinct flights. Although it does not explicitly define fare classes, about 14% of the entries relate to basic economy fares.

Due to the large size of the dataset, it was stored directly in Google Drive rather than on a personal computer. To address memory constraints in Google Colab, only the necessary columns and appropriate data types were specified before reading the data in chunks and concatenating them for further analysis. Given the time and memory limitations, running

a model on every possible route was deemed impractical. Instead, flights were grouped by origin and destination, and the five most common route pairs were identified, as illustrated in Figure 2. Among these, the route with the highest number of data points, ATL-LAX, was selected for experimentation. Additionally, to simplify price prediction and reduce variability introduced by multiple fare classes, basic economy fares, which accounted for only 14% of the dataset, were excluded from the analysis.

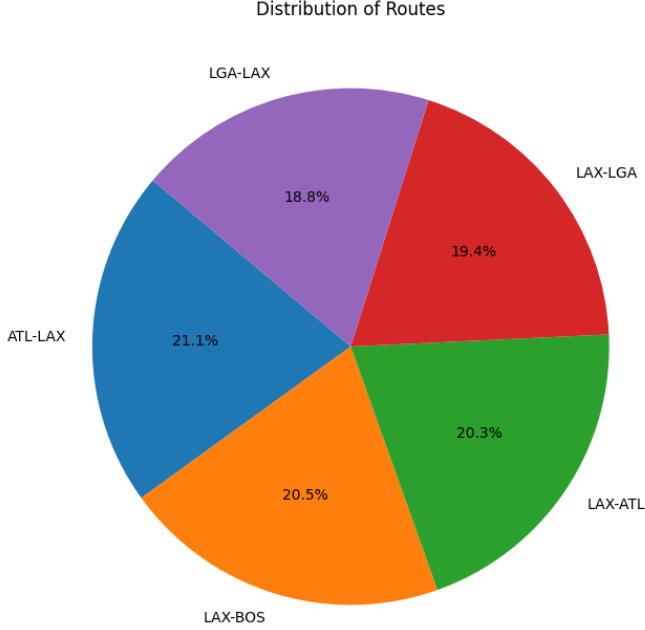


Fig. 2. Distribution of top 5 routes in filtered data.

B. Feature Engineering and Preprocessing

Preprocessing involved several key steps. First, relevant date/time fields were converted to proper date/time types, and the number of days to departure (i.e., the difference between the scheduled departure date and the price-observation date) was calculated. Arrival and departure times, provided as actual epoch timestamps due to being the only option available in the data source, were subsequently transformed into floating-point values ranging from 0.00 to 24.00 to represent the times of day. Next, day-of-week and ordinal date values were extracted to capture potential temporal patterns. Finally, 30 average fare prices were computed for each departure date, covering a period from 30 days before departure up to one day before departure; these values were stored in a matrix for subsequent retrieval.

C. Predictive Target and Input Features

When a specific price $p_{i,j}$ (where i is the calendar day and j is the number of days to departure) was chosen as the predictive target, the nearest 30 average prices were taken from this matrix and included as additional inputs.

IV. BASELINE

A. Motivation

A baseline model serves as a simple, interpretable reference against which more complex, specialized methods can be compared. By starting with a straightforward technique, it becomes possible to gain an initial understanding of the task's difficulty and what performance level can be achieved without intricate architectures or heavy hyperparameter tuning.

B. Model Choice: Random Forest Regressor

Random forest regressor from scikit-learn with 100 estimators (decision trees) was selected as the baseline. Random Forests tend to handle tabular data well, can capture non-linear relationships, and require minimal data preprocessing compared to neural networks. They also provide a relatively robust performance on airline ticket price prediction [6].

V. MAIN APPROACH

A. Introduction to the Model

The proposed method adopts a hybrid architecture combining an LSTM with a multi-layer CNN, closely resembling the design presented in Figure 1. The key distinction lies in the simplification of the architecture, where a single LSTM model is utilized instead of the five parallel LSTM models used in the original approach. This adjustment was made to streamline the model, reduce computational complexity, and enable more efficient experimentation. The LSTM layers are included to learn long-term dependencies and trends in flight prices over time, leveraging their ability to retain historical context in a time-series. In parallel, the 1D convolutional layers focus on local interactions among chronologically adjacent flights or prices, providing a structured way to identify short-range correlations within the data. By integrating these two components, the aim is to handle complex temporal relationships while also capturing localized patterns that influence flight ticket prices.

B. Input/Output

Input: Each flight is represented by a vector of 36 features that include six attributes describing flight conditions at the time of departure and 30 features illustrating price changes for a given flight.

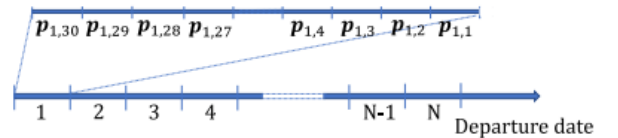


Fig. 3. The ticket prices arranged in time series sequence (reprinted from Du et al. [11]).

Du et al. [11] used departure time in 24-hour format, arrival time in 24-hour format, departure date, days to departure, day of week, and a holiday indicator. The current study follows a similar structure; however, due to constraints in the available data, departure and arrival times were taken from actual flight

times rather than scheduled times. Consequently, this does not accurately capture the specific departure hour for which the ticket is being sold. An extensive search of publicly available datasets was conducted, but none met the requirement of providing precise scheduled times.

The next 30 values in the dataset are taken from the reshaped ticket price sequence illustrated in Figure 3. If $p_{i,j}$ is selected as the model's output, then b_1 through b_{30} are the 30 values that immediately precede $p_{i,j}$ in the sequence (see Fig. 2). For instance, if the output is $p_{32,30}$, the corresponding b_1 to b_{30} are $p_{31,30}$ through $p_{31,1}$. Similarly, if the output is $p_{32,29}$, those 30 preceding values would be $p_{31,29}$ through $p_{31,1}$ along with $p_{32,30}$.

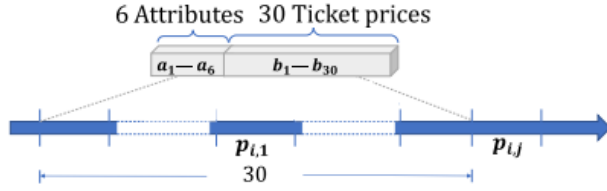


Fig. 4. The data structure of the input (reprinted from Du et al. [11]).

Output: A single predicted ticket price (a real-valued number). Each prediction is made for a given day, time, and days remaining to departure combination.

C. Model Architecture

The input layer expects a tensor of shape $(batch_size, sequence_length, 36)$, where 36 includes 6 flight attributes and 30 sequential price points; `batch_first=True` keeps the shape ordering as $(B, T, features)$. A three-layer LSTM block (`nn.LSTM(36, 64, 3, batch_first=True)`) then extracts temporal relationships among these features, producing a hidden representation of shape $(B, T, 64)$. Next, the output is permuted to $(B, 64, T)$ for a three-layer 1D convolution stack, where each layer applies a kernel of size 3 with stride 1, successively reducing or maintaining the time dimension, leading to a final shape of $(B, 16, T')$. This feature map is flattened to $(B, 16 \times T')$ and passed through two fully connected layers—first a hidden layer (e.g., `Linear(16 * T', 128)` with ReLU) and then a final `Linear(128, 1)` for ticket price prediction. The network is trained using Mean Squared Error (MSE) as the loss function, typically optimized by Adam with an initial learning rate (e.g., 2×10^{-5}).

D. Training Procedure

The data, shaped as $(B, T, 36)$, is first placed into a `TensorDataset` and provided to a `DataLoader`, which shuffles the training set each epoch and applies mini-batch gradient descent. During the forward pass, inputs move through

the LSTM, followed by convolution, flattening, and fully connected layers to generate an estimated ticket price. The backward pass computes the MSE loss against ground truth labels, then calls `loss.backward()` and `optimizer.step()` to update the model parameters. After every epoch, validation MSE is calculated to guide decisions on early stopping or to record the best model weights. A `ReduceLROnPlateau` scheduler was used to automatically lower the learning rate once the validation loss plateaus.

E. Strengths of the Proposed Model

The proposed architecture integrates temporal memory through LSTM layers and local receptive fields via Conv1D, effectively capturing short-term fluctuations in pricing data beyond what an RNN alone might achieve. By simultaneously learning the interplay between flight attributes and price sequences end-to-end, this approach demands less manual feature engineering than traditional methods, providing a more streamlined and powerful method for modeling complex pricing dynamics.

VI. EVALUATION METRIC

A. Root Mean Squared Error (RMSE)

The primary evaluation metric employed is Root Mean Squared Error (RMSE), which is derived from Mean Squared Error (MSE) and is expressed as:

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

where y_i represents the ground truth ticket price for the i -th sample, \hat{y}_i is the predicted price, and N denotes the total number of samples in the dataset or batch. RMSE provides an interpretable measure of error in the same unit as the target variable, making it particularly useful for assessing the accuracy of ticket price predictions. Furthermore, like MSE, RMSE penalizes larger errors more heavily, ensuring that the model minimizes significant deviations in predictions.

B. Prediction Accuracy (ACC)

In addition to RMSE, Prediction Accuracy (ACC) is employed as a secondary evaluation metric. ACC measures the proportion of predictions that fall within a predefined tolerance range of the actual values, ensuring the model's practical utility for approximating ticket prices.

The prediction is considered "correct" if it satisfies the following criterion:

$$|\hat{y}_i - y_i| \leq \text{tolerance} \cdot y_i,$$

where \hat{y}_i is the predicted value, y_i is the actual value, and tolerance is a user-defined threshold. In this study, a tolerance of 10% (i.e., tolerance = 0.1) is applied.

The accuracy is computed as:

$$ACC = \frac{\text{Number of Correct Predictions}}{N},$$

where N is the total number of predictions. If $N = 0$, the accuracy defaults to 0 to account for edge cases with no available predictions.

This metric complements RMSE by providing an interpretable measure of how often the model produces predictions within an acceptable range, making it particularly useful for practical decision-making scenarios.

VII. RESULTS & ANALYSIS

A. Overview of Final Results

This section details the performance of both the baseline model (Random Forest Regressor) and the proposed main approach (hybrid LSTM-CNN), presenting the results of training and evaluation experiments. A comprehensive analysis of these results is then provided, discussing their implications and informing potential future refinements to the main approach. The training and validation loss curves of the proposed model over training epochs are visualized in Figure 5.

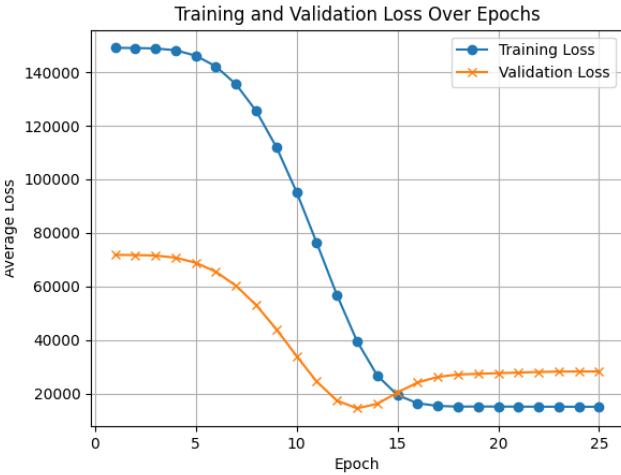


Fig. 5. Training and Validation Loss Curves over Training Epochs for ReduceLROnPlateau Learning Rate Scheduler.

B. Baseline Model Performance

The baseline model, a Random Forest Regressor, was trained with a fixed set of hyperparameters, employing 100 decision trees ($n_estimators=100$) and minimal further tuning. This choice was made to establish a readily reproducible and relatively simple benchmark against which to compare the performance of the more complex proposed model. The baseline model achieved a Validation Root Mean Squared Error (RMSE) of 45.0342. This metric provides a measure of the average magnitude of the errors between predicted and actual ticket prices.

C. Proposed Model (hybrid LSTM-CNN) Performance and Training Dynamics

The proposed model, a hybrid LSTM-CNN architecture, underwent a training process spanning 25 epochs. The training progress, as evidenced by the average training and validation

losses per epoch (depicted in Figure 5), reveals important insights into the model's learning behavior.

The initial epochs exhibited high training and validation losses, with the validation loss starting above 70,000. This is attributable to the initial random initialization of the model's weights and the inherent scale of ticket prices in the dataset. However, as training progressed, both training and validation losses demonstrated a consistent downward trend, indicating that the model was effectively learning to capture patterns in the data. This initial learning phase is clearly visible in Figure 5 as a steep decline in both loss curves.

Specifically, the validation loss decreased substantially over the first 13 epochs, reaching a minimum of approximately 14,500 at epoch 13. This rapid initial decrease, readily observed in Figure 5, suggests that the model quickly learned a preliminary representation of the underlying data structure, including the sequential dependencies inherent in the time series of ticket prices. This is a key advantage of the LSTM component, which is designed to capture temporal relationships.

Beyond epoch 13, a notable trend reversal occurred. The training loss continued to decrease (as shown in Figure 5), the validation loss began to increase, indicating the onset of overfitting. This phenomenon suggests that the model began to memorize the training data rather than generalizing to unseen data.

To mitigate overfitting and potentially improve generalization performance, a 'ReduceLROnPlateau' learning rate scheduler was implemented. This scheduler automatically reduces the learning rate by a factor of 0.1 if the validation loss plateaus for a specified number of consecutive epochs (in this case, 3). This intervention was triggered at epoch 18, as the validation loss, depicted in Figure 5, had begun to stagnate and subsequently increase. The reduced learning rate allowed the model to fine-tune its parameters in a more controlled manner, potentially escaping local minima and improving generalization. However, as the Figure 5 show, the validation loss continued to increase even with the reduced learning rate.

The proposed model achieved a Validation RMSE of 122.7148, which is substantially higher than the baseline model's RMSE of 45.0342.

Both models were evaluated on the following datasets:

- *Training set* (used for parameter fitting)
- *Validation set* (used for hyperparameter tuning)
- *Test set* (used for final performance evaluation)

The primary evaluation metric was Root Mean Squared Error (RMSE). Prediction Accuracy (ACC) was also tracked in some experiments for further analysis.

VIII. ERROR ANALYSIS

A. Summary of Experiments

This section details the various experiments conducted to evaluate the performance and robustness of the proposed models under different conditions. These experiments explored the impact of prediction horizon, and learning rate schedules on model performance.

B. Varying Prediction Horizon

The influence of remaining days to departure, representing the number of days which the prediction is being made in advance, was examined. Three different remaining days to departure (ndo) were analyzed: 1, 7, and 30.

TABLE I
PROPOSED MODEL PERFORMANCE METRICS

Metric	ndo 1	ndo 7	ndo30
RMSE	167.56	132.98	122.71
Accuracy	0.06	0.28	0.1

Analyzing the results in Table I, an unexpected trend emerges. It was initially hypothesized that predictions made further from the flight departure date, such as the ndo 30 scenario, would be characterized by greater uncertainty. This is because information about ticket price dynamics tends to become less reliable and harder to interpret as the prediction horizon increases, leading to a decrease in accuracy and an increase in error metrics like RMSE. However, the results indicate the opposite trend: the RMSE decreases from 167.56 for ndo 1 to 132.98 for ndo 7 and further to 122.71 for ndo 30, while the accuracy, although still relatively low, peaks at 0.28 for ndo 7 and remains higher for ndo 30 compared to ndo 1.

One possible explanation for this phenomenon lies in the architecture of the LSTM-CNN model and the nature of the ticket price data. The LSTM component, designed to capture sequential patterns and temporal dependencies, may perform more effectively when given data from longer horizons. This longer context could allow the model to identify more robust trends and patterns that stabilize over time, thereby reducing error rates for predictions further in advance. Similarly, the CNN component, responsible for extracting local features from input sequences, may also benefit from the richer information provided by longer time horizons, leading to improved feature extraction and, consequently, better predictions.

Another factor to consider is the variability of ticket prices closer to the departure date. As departure nears, ticket prices often exhibit higher volatility due to last-minute demand fluctuations, promotional offers, or changes in seat availability. This increased noise in the data for the ndo 1 scenario may have hindered the model's ability to accurately predict prices, leading to the observed higher RMSE and lower accuracy.

The intermediate performance at ndo 7 suggests that this time frame strikes a balance between the uncertainty associated with long-term predictions and the volatility of short-term predictions. The peak accuracy of 0.28 at ndo 7 supports this notion, as the model may have been able to leverage meaningful trends without being overly impacted by near-term price volatility or long-term uncertainty.

These results highlight the complexity of price prediction for varying prediction horizons and underscore the need for further investigation.

C. Effect of Learning Rate Schedule

The training dynamics of the LSTM-CNN model trained with a constant learning rate of 2×10^{-5} are shown in Figure 6. As depicted, the training loss demonstrates a consistent downward trend across all 25 training epochs, reflecting steady optimization. In contrast, the validation loss exhibits fluctuations, particularly between epochs 5 and 10, suggesting sensitivity to overfitting or noise in the validation data during this phase.

At the start of training, the validation loss is notably high, approximately 71,670 in the first epoch. This is expected due to the random initialization of model weights and the large scale of ticket price data. However, a rapid decrease in both training and validation losses is observed during the initial epochs, indicating the model's ability to quickly learn and capture key patterns in the data. This rapid convergence is evident as a steep decline in the loss curves during the early training stages.

Despite the initial progress, the fluctuations in validation loss highlight potential overfitting tendencies as the model continues to train. The learning dynamics underscore the importance of tuning hyperparameters, such as the learning rate schedule, to balance training efficiency and generalization performance.

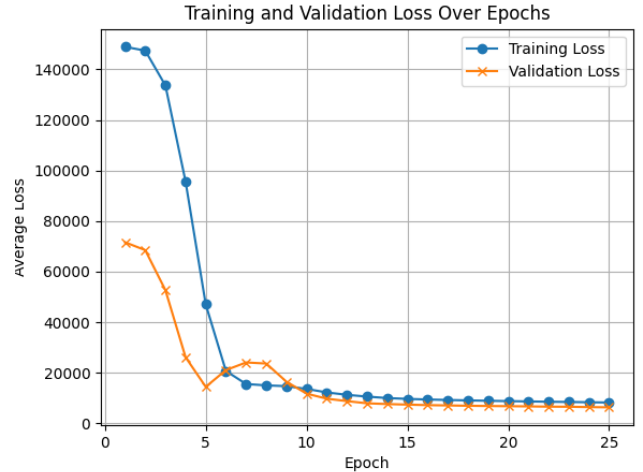


Fig. 6. Training and Validation Loss Curves over Training Epochs for Constant Learning Rate of 2×10^{-5} .

The validation loss shows a steady decline, dropping to approximately 14,481 by epoch 5 and further decreasing to around 10,123 by epoch 13. This consistent reduction highlights the model's capability to progressively extract more complex features and enhance its predictive accuracy. Notably, the continued decrease in validation loss beyond epoch 13 indicates that the model is still learning and refining its representations. By the final epoch (epoch 25), the validation loss reaches its lowest point of approximately 7,090 which reflects the model's improved performance over the training process.

Comparing these results with the proposed model's performance using a learning rate scheduler (as detailed in the results & analysis section and also shown in Figure 5), several key observations can be made. After epoch 13, the validation loss of the model with learning rate scheduler increases significantly, indicating overfitting. This suggests that while the learning rate scheduler initially helps the model find a better minimum, it subsequently leads to overfitting, preventing the model from achieving the lower validation loss obtained with the constant learning rate. The constant learning rate approach, while simpler to implement and not achieving the initial minimum of the scheduler, ultimately results in better generalization on the validation set in this specific case. This highlights that while dynamic learning rate scheduling can be beneficial, careful tuning and monitoring are crucial to prevent overfitting and ensure optimal performance. In this specific scenario, the constant learning rate appears to be more effective for generalization.

IX. FUTURE WORKS

A. Enhanced Data Preprocessing

One potential avenue for future research is the refinement of data preprocessing techniques. The high initial validation loss observed in the LSTM-CNN model suggests that the raw input data may not have been optimally scaled or normalized for deep learning models. Exploring advanced preprocessing methods such as robust scaling, log transformations, or feature engineering techniques could better align the data with the model's requirements. Additionally, addressing potential outliers and ensuring consistency in time-series length across different samples may improve overall model stability and performance.

B. Emphasizing the Importance of Scheduled Departure and Arrival Data

A key limitation of the current study lies in the reliance on actual departure and arrival datetime values, which reflect real-world delays and disruptions. While such data is valuable for understanding historical patterns, it introduces inherent noise and unpredictability, particularly for models aimed at forecasting ticket prices well in advance. Future research should prioritize the acquisition and incorporation of scheduled departure and arrival datetime values, as these provide a more stable and reliable foundation for prediction.

C. Regularization Techniques

The onset of overfitting observed beyond epoch 13 underscores the need for robust regularization strategies. Incorporating techniques such as dropout layers, weight decay, or data augmentation could prevent the model from memorizing training data and encourage better generalization. Moreover, early stopping based on validation loss trends could help terminate training before overfitting becomes significant.

D. Hyperparameter Tuning

The current study employed a fixed set of hyperparameters, but a systematic hyperparameter tuning process could uncover more effective configurations. Grid search, random search, or Bayesian optimization could be employed to fine-tune critical parameters such as the learning rate, number of LSTM units, CNN filter sizes, and dropout rates.

E. Incorporating External Features

The current model relies solely on historical ticket prices as input. Augmenting the dataset with external features such as macroeconomic indicators, seasonal trends, or competitor pricing information could provide additional context for the model. Temporal embeddings or feature interactions could also help integrate such auxiliary data effectively.

F. Transfer Learning and Pretraining

To improve model initialization and expedite convergence, transfer learning techniques could be explored. Pretraining the LSTM-CNN model on a related task or dataset, such as generic time-series forecasting, may provide a strong starting point for the ticket price prediction task, reducing the reliance on lengthy training processes.

G. Exploration of Advanced Loss Functions

The current model optimization relies on standard loss functions such as Mean Squared Error (MSE). Investigating alternative loss functions, such as Huber loss or quantile loss, may help mitigate the impact of outliers and better align the training objective with the ultimate evaluation metrics.

H. Temporal Cross-Validation

To ensure robust evaluation, future studies could adopt temporal cross-validation techniques specifically designed for time-series data. Methods such as walk-forward validation or expanding window validation could provide a more accurate assessment of the model's performance across different time periods.

X. ETHICAL CONSIDERATIONS

This study involves the prediction of airline ticket prices using historical pricing and time-series data. While the research is focused on advancing methodological approaches and providing actionable insights for both consumers and airlines, it is essential to acknowledge and address the ethical considerations associated with such work.

A. Potential for Misuse of Predictive Models

The predictive model developed in this study has the potential to influence consumer behavior and airline pricing strategies. While the intention of this research is to enhance transparency and improve decision-making for all stakeholders, the outcomes could be misused to manipulate ticket prices unfairly or disadvantage certain consumer groups. For instance, airlines could exploit predictive tools to maximize profits at the expense of equitable pricing. To mitigate such risks, this study emphasizes the need for transparency in model

deployment and encourages the adoption of ethical pricing practices.

B. Bias in Data and Model Outputs

Historical pricing data used in this study may inherently reflect biases present in existing airline pricing systems. These biases could arise from various factors, such as differential pricing practices based on geographic regions, travel seasons, or customer segments. The model may inadvertently perpetuate or amplify these biases, leading to inequitable outcomes for certain groups. Efforts were made to identify and minimize such biases during data preprocessing and model evaluation. Future work should include fairness metrics to quantify and mitigate these biases more effectively.

C. Consumer and Airline Welfare

The dual purpose of this research is to empower consumers to make informed purchasing decisions and to assist airlines in optimizing their pricing strategies. However, it is crucial to ensure that these goals do not conflict in ways that harm either party. For instance, while dynamic pricing strategies may maximize airline profits, they could also result in unpredictable and potentially unfair price surges for consumers.

XI. CODE

The code for this work is available at <https://github.com/kutayeroglu/airline-ticket-prices>.

REFERENCES

- [1] Li, Jun, Granados, Nelson, Netessine, Serguei, 2014. Are consumers strategic? Structural estimation from the air-travel industry. *Manage. Sci.* 60 (9), 2114–2137.
- [2] Groves, William, Gini, Maria, 2013. An agent for optimizing airline ticket purchasing, in *International conference on Autonomous agents and multiagent systems*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC.
- [3] Chen, Y., Cao, J., Feng, S., Tan, Y., 2015. An ensemble learning based approach for building airfare forecast service. In: *2015 IEEE International Conference on Big Data (Big Data)*, Santa Clara, CA, 2015, pp. 964-969
- [4] Lantseva, Anastasia, Mukhina, Ksenia, Nikishova, Anna, Ivanov, Sergey, Knyazkov, Konstantin, 2015. Data-driven Modeling of Airlines Pricing. *Procedia Comput. Sci.* 66, 267–276. ISSN 1877-0509
- [5] Janssen, T., 2014. A linear quantile mixed regression model for prediction of airline ticket prices. Radboud University.
- [6] Tziridis, K., Kalampokas, T., Papakostas, G.A., Diamantaras, K.I., 2017. Airfare prices prediction using machine learning techniques, 25th European Signal Processing Conference (EUSIPCO). Kos 2017, 1036–1039.
- [7] Liu, T., Cao, J., Tan, Y., Xiao, Q., 2017. ACER: An adaptive context-aware ensemble regression model for airfare price prediction. In: *2017 International Conference on Progress in Informatics and Computing (PIC)*, Nanjing, 2017, pp. 312–317.
- [8] William Groves, Maria Gini, 2011. A regression model for predicting optimal purchase timing for airline tickets. Technical report, University of Minnesota, Minneapolis, USA, Report number 11-025, 2011.
- [9] Boruah, A., Baruah, K., Das, B., Das, M.J., Gohain, N.B., 2018. A Bayesian Approach for Flight Fare Prediction Based on Kalman Filter. In: *Progress in Advanced Computing and Intelligent Engineering. Advances in Intelligent Systems and Computing*, Springer, Singapore, vol. 714, pp. 191–203, 2018 Boruah, A., Baruah, K., Das, B., Das, M.J., Gohain, N.B., 2018. A Bayesian Approach for Flight Fare Prediction Based on Kalman Filter. In: *Progress in Advanced Computing and Intelligent Engineering. Advances in Intelligent Systems and Computing*, Springer, Singapore, vol. 714, pp. 191–203, 2018
- [10] Li, Yuling, Li, Zhichao, 2018. Design and implementation of ticket price forecasting system 1967, 040009
- [11] Du, K.; Wang, Y.; Li, X.; Zhang, H.; Chen, Y. An LSTM Based Deep Learning Method for Airline Ticket Price Prediction. In *Neural Information Processing, Proceedings of the 27th International Conference, ICONIP 2020, Bangkok, Thailand, 18–22 November 2020*; Springer International Publishing: Berlin/Heidelberg, Germany, 2020
- [12] Kaggle, “Flight Prices,” 2022. [Online]. Available: <https://www.kaggle.com/dilwong/flightprices/data>