Comparative analysis of neural networks techniques to forecast Airfare Prices

Original
Comparative analysis of neural networks techniques to forecast Airfare Prices / Aliberti, Alessandro; Xin, Yao; Viticchie, Alessio; Macii, Enrico; Patti, Edoardo. - (2023), pp. 1023-1029. (Intervento presentato al convegno 47th IEEE Annual Computers, Software, and Applications Conference (COMPSAC 2023) tenutosi a Turin (Italy) nel 27-29 June 2023) [10.1109/COMPSAC57700.2023.00157].

Availability:
This version is available at: 11583/2980935 since: 2023-08-04T10:08:45Z

Publisher:
IEEE

Published
DOI:10.1109/COMPSAC57700.2023.00157

(Article begins on next page)

Comparative analysis of neural networks techniques to forecast Airfare Prices

Alessandro Aliberti*, Yao Xin*, Alessio Viticchié†, Enrico Macii* and Edoardo Patti*
*Politecnico di Torino, Turin, Italy. Emails: name.surname@polito.it
‡AlphaWaves S.r.l., Turin, Italy. Email: name.surname@awaves.it

***Abstract*—With the growth of tourism industry, airplanes have became an affordable choice for medium- and long-distance travels. Accurate forecasting of flights tickets helps the aviation industry to match demand, supply flexibly and optimize aviation resources. Airline companies use dynamic pricing strategies to determine the price of airline tickets to maximize profits. Passengers want to purchase tickets at the lowest selling price for the flight of their choice. However, airline tickets are a special commodity that is time-sensitive and scarce, and the price of airline tickets is affected by various factors.**

Our research work provides a systematic comparison of various traditional machine learning methods (i.e., Ridge Regression, Lasso Regression, K-Nearest Neighbor, Decision Tree, XGBoost, Random Forest) and deep learning methods (e.g., Fully Connected Networks, Convolutional Neural Networks, Transformer) to address the problem of airfare prediction, by keeping the consumers' needs. Moreover, we proposed innovative Bayesian neural networks, which represent the first exploitation attempt of Bayesian Inference for the airfare prediction task, to the best of our knowledge. Therefore, we evaluate the performance of our implemented and optimized models on an open dataset. The experimental results show that deep learning-based methods achieve better results on average than traditional ones, while Bayesian neural networks can achieve better performance among the other machine learning methods. However, taking into account both prediction performance and computational time, the Random Forest turns out to be the best choice to apply in this scenario.

***Index Terms*—Airfare prediction, Regression, Machine Learning, Deep Learning, Bayesian Neural Network**

I. Introduction

Research in the area of civil aviation has been encouraged by the industry's quick expansion and the rise in passenger traffic. The airline sector uses sophisticated pricing techniques, and even adjacent seats in the same service class may be marketed at wildly different costs for tickets on the same aircraft [1]. This is so that airlines may optimize their earnings using a variety of intricate pricing schemes in the yield management system. Even though the Internet growth enabled consumers to access more useful information, airlines can still create information asymmetry by keeping useful information like the number of available seats a trade secret and adjusting ticket prices on the fly to maximize profits [1]. The prediction service may incur significant computational costs and overhead if it is used too frequently. Therefore, creating an appropriate prediction service invocation approach is a critical issue that must be resolved. However, there are few research works on this topic due to the specificity of data in the sector of airline tickets [2].

Even while airlines have developed their theoretical understanding of air ticket pricing and revenue management, there is still a dearth of research on customer purchase behaviour. There are principally two causes for this: i) most airlines do not publicly publish their pricing methods; ii) there are not enough publicly accessible datasets [1].

In an effort to provide customers with better buying tactics, recent works set some suggestions based on scant data. Predicting the price of an airline ticket is a common time series prediction problem, but because the data is unique, the model is prone to error. Additionally, because there are several factors that impact airline ticket costs, it is essential to create a robust model.

In our research work, we provide a comprehensive and systematic comparison of different state-of-the-art Machine Learning and Deep Learning methods on the problem of airfare prediction. Moreover, we propose a novel Bayesian neural network as a first attempt to exploit Bayesian Inference for prices prediction in the touristic aviation industry (to the best of our knowledge). We evaluate the performance of different methods on an open dataset of 10683 routes in India from March 2019 to June 2019. The experimental results show that deep learning methods achieve on average more accurate results than traditional ones, however at the expense of computational time. Random Forest still proves to be the best choice for this application. The prediction results, in terms of accuracy, are comparable (and sometimes better) than those obtained by advanced deep learning methodologies. All this, at an acceptable computational cost. Instead, among the deep learning techniques, our proposed Bayesian neural networks achieve better performance according to the performance indexes.

The rest of the paper is organized as follows. Section III presents a literature review of airfare forecasting methods. Section II emphasizes the problem formulation and the novelty of applying machine learning techniques in airfare price forecast scenario. In Section IV, we present our methodology. Section V presents all the obtained experimental results. Finally, Section VI provides the concluding remarks.

II. Problem Formulation and Novelty

With the booming tourism industry, more and more people are choosing airplanes as a mean of transportation. Consequently, airline companies started to invest in R&D activities to realize even more accurate dynamic pricing strategies to determine the price of airline tickets and, then, maximize their profits [3]. In contrast, passengers would like to be able to purchase tickets at the lowest selling price possible [2]. However the price of airline tickets is affected by various factors, such as the departure time of the plane, the number of hours of advance purchase, and the airline flight, so it is difficult for consumers to know the best time to buy a ticket. Moreover, the scientific literature lacks of in-depth studies precisely geared to help consumers.

Based on this assumption, we propose a systematic comparison of traditional machine learning methods and deep learning methods to address the problem of airfare prediction, from a consumer's point of view. Therefore, we selected the most promising models used by airlines for dynamic pricing strategies. Then, we implemented and optimized this model against a public dataset. Moreover, inspired by the observation that XGBoost and Random Forest achieve good prediction accuracy results, we implemented novel Bayesian airfare prediction architectures. The final purpose is to create algorithms whose results can inspire new business models or enhance existing ones (e.g. Skyscanner [4], Kayak [5], etc.) with a special consideration for consumers.

III. Related Works

In recent years, different forecasting methods have been proposed for airfare price forecasting especially as artificial intelligence flourished. Therefore, in this section we review the studies that are most related to our methodology, analyzing their strengths and weaknesses.

By exploiting openly accessible datasets (i.e. DB1B [6] and T-100 [7] databases) and a cutting-edge machine learning framework, Tianyi Wang *et al.* tackled the issue of market segment-level flight price prediction [8]. Their suggested architecture aims to provide a thorough profile of each market segment and use machine learning techniques to forecast the typical airfare price for each segment. A regression model proposed by Groves *et al.* uses the history diagram to forecast the ideal time to book airline tickets [9]. From February 2011, to June 2011, they gathered their data, which included over 140000 records in total [10]. Their model consists of two phases. In the first, they predict the day price using a regression model. The second step creates a reliable rule based on the dependable threshold. If the price is less than the value, which is the predicted price less than the threshold, passengers should purchase the ticket. Their findings demonstrated that their methodology successfully reduces the average cost when the purchase date is more than two months out from the departure date. Buyrukolu *et al.* study [11] used 1814 one-way flights from Greece to Germany to explore different models (i.e. Ridge, Lasso and Elastic Net models). Based on their results, these methods were successful in producing compelling findings for flight pricing analysis. In [12], the Huang *et al.* used Artificial Neural Networks (ANN) and Genetic Algorithms (GA) to forecast the income from sales of airline tickets. The input characteristics included the price of crude oil on the world market, the weighted index of Taiwan's stock market, the monthly unemployment rate in Taiwan, and more. In order to boost the performance of the ANNs, the GA specifically chooses the best input characteristics. With a mean absolute percentage error of 9.11%, the model performed well. More sophisticated machine learning models have been taken into consideration to enhance the forecast of flight prices. For example, Tziridis *et al.* [13] investigated eight machine learning models, including ANNs, Random Forest (RF), Support Vector Machine (SVM), and Lasso Regression (LR) to forecast ticket prices and evaluate their effectiveness. The most accurate regression model obtained has an 88% accuracy rate. The Bagging Regression Tree, which is reliable and unaffected by utilizing various input feature sets, is the best model in their comparison. Deep Regressor Stacking was suggested in [13] as a way to make more accurate forecasts. The suggested solution is a brand-new multi-target strategy that uses RF and SVM as regressors, it is easily adaptable to other problem domains with a similar set of issues.

Since airline ticket data is rarely categorized and prepared for direct analysis, gathering and processing such data is always labour-intensive. Usually, given the literature, researchers seek private data from collaborative groups or web crawl the data to test the effectiveness of their models on different data. Hence, it is challenging to reproduce the study and make performance comparisons among the models. Our suggested system, in contrast to prior and current works, is able to solve the price prediction problem by utilizing just public data sources with a minimal set of characteristics. Additionally, the suggested framework may be used to estimate air travels costs in any market, as it is not constrained by any particular segment. Therefore, we first propose a systematic comparison of traditional machine learning and deep learning methods on the problem of airfare prediction. Then, we implement and optimize the state-of-the-art's most promising models. Afterwards, we observe how ensemble models (i.e. XGBoost and Random Forest) achieve better performance than other traditional machine learning methods. Hence, we propose a novel Bayesian airfare prediction network, which learns the data distribution from the training data and exploits the ensemble idea to boost the performance. Finally, we demonstrate the obtained performance by comparing our model to the previous optimized ones.

IV. Methodology

In this Section, we outline the methodology we propose to systematically compare different state-of-the-art machine learning methods for airfare prices prediction. Our goal is to find the model that best implements Airfare Prices forecast from the end customer's perspective. Moreover, we particularly emphasize our proposed Bayesian neural networks, which are the first methods that exploit Bayesian Inference for the airfare prediction task to the best of our knowledge.

Figure 1 summarizes the overall process. The rest of this section will describe in-depth the proposed methodology.

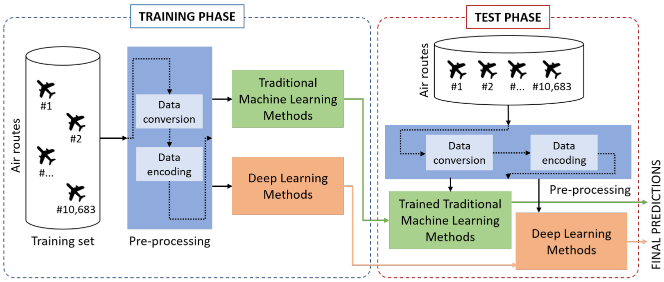

Fig. 1. Methodology outline

A. Flight Prices Dataset

The dataset used in this paper contains a total of 10683 routes between these cities within India: New Delhi, Bangalore, Cochin, Kolkata, Hyderabad, and Delhi from March 2019 to June 2019, and from this data, each raw data contains 11 fields of information [14]. The features of the dataset are detailed in Table I

TABLE I
DATASET FEATURES DETAIL

#	Feature Name	Description
1	Airline	Name of the airline company is stored in the airline column. It is a categorical feature having 6 different airlines.
2	Flight	Flight stores information regarding the plane's flight code. It is a categorical feature.
3	Source City	City from which the flight takes off. It is a categorical feature having 6 unique cities.
4	Departure Time	Derived categorical feature obtained created by grouping time periods into bins. It stores information about the departure time and has 6 unique time labels.
5	Stops	Categorical feature with 3 distinct values that stores the number of stops between the source and destination cities.
6	Arrival Time	Categorical feature derived by grouping time intervals into bins. It has 6 distinct time labels about the arrival time.
7	Destination City	The flight's landing city. Categorical feature having 6 unique cities.
8	Class	Categorical feature that contains information on seat class. It has two distinct values: Business and Economy.
9	Duration	Continuous feature that displays the overall amount of time it takes to travel between cities in hours.
10	Days Left	Derived characteristic that is calculated by subtracting the trip date from the booking date
11	Price	250, 500, 750, 1000, 1500

B. Data preprocessing

To enhance the performance of our models, we performed a comprehensive pre-processing of the selected dataset. In particular, we conducted an investigation to convert and standardize the dataset as much as possible, to improve the machine learning model performances.

1) Data conversion: to better handle data, we converted duration hours into minutes; split the journey date into journey day and journey month; split the departure time and arrival time into hours and minutes.

2) Data encoding: finally, to exploit the machine learning models, we converted categorical data into numerical. Categorical data encoding process allows to improve the prediction accuracy [15].

Table II exemplify the data obtained as a result of the pre-processing phase. For each feature in the dataset (i.e., one per column), the rows represent an air route. All values are expressed in numbers to facilitate the implementation of machine learning and deep learning methods.

C. Machine Learning models

For a thorough comparison in the context of airfare predictions, we implemented and optimized twelve Machine Learning methods. Specifically, we included seven traditional machine learning regression algorithms (i.e., Lasso Regression, Ridge Regression, Support Vector Regression, K-Nearest Neighbors, XGBoost, Decision Tree, Random Forest) and five deep neural network methods (i.e., Transformer, Fully Connected Network, Bayesian Fully Connected Network, Convolutional Neural Network, Bayesian Convolutional Neural Network). For a fair comparison, we split the dataset into 70% training data and 30% test and validation data. We use the same training and test data for all the implemented methods.

1) Traditional Machine Learning Methods: the traditional machine learning regression methods are implemented using the *scikit-learn* framework. We trained our models on Intel® Core™ i3-9100F CPU @ 3.60GHz × 4 with 32 GB of RAM memory. The settings of hyper-parameters are given for all the implemented models.

a) Lasso Regression: the Lasso (Least Absolute Shrinkage and Selection Operator) is a regularization technique used over regression methods for a more accurate prediction [16]. This technique penalizes the absolute magnitude of the regression coefficient. Additionally, it employs variable selection, which leads to the shrinkage of coefficient values to absolute zero.

We used [0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000, 100000] as Lasso parameters' set and we exploited the *Grid* search with validation method, by setting $K = 10$

b) Ridge Regression: it is applied when the independent variables are highly correlated, in a nutshell, when data exhibits multicollinearity [16]. While least squares estimates are unbiased in multicollinearity, their variances are significant enough to cause the observed value to diverge from the actual value. Ridge regression reduces standard errors by biassing the regression estimates.

We used [0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000, 100000] as Ridge parameters' set and we used the *Grid* search with validation method, by setting $K = 10$.

c) Support Vector Machine (SVM): it is a supervised machine learning model, suitable for classification, regression and outliers detection [17]. The objective of the SVM algorithm is to find a hyperplane in N-dimensional space (where N is the number of features) that distinctly classifies the data points on the hyperplane.

We used C: [1e0, 1e1, 1e2, 1e3] as regularization parameters in the SVM regression model and we used the *Grid* search with validation method. The strength of the regularization is inversely proportional to C and must be strictly positive. The penalty is a L2 penalty. We also used a set of Gaussian kernel coefficients in our model to obtain good performance.

TABLE II
EXAMPLE OF THE DATA INFORMATION STRUCTURE AFTER PRE-PROCESSING PHASE.

	Air route									
	#1		#2		#3		#...		#n	
	Data before pre-processing	Pre-processing result	Data before pre-processing	Pre-processing result	Data before pre-processing	Pre-processing result	Data before pre-processing	Pre-processing result	Data before pre-processing	Pre-processing result
Airline	IndiGo	3	Air India	1	Jet Always	4	...	...	Air India	1
Source	Bangalore	0	Kolkata	3	Delhi	2	...	...	Bangalore	0
Destination	New Delhi	5	Bangalore	0	Cochin	1	...	...	New Delhi	5
Route	BLR $\rightarrow$ DEL	18	CCU $\rightarrow$ IXR $\rightarrow$ BBI $\rightarrow$ BLR	84	DEL $\rightarrow$ LKO $\rightarrow$ BOM $\rightarrow$ COK	118	...	...	BLR $\rightarrow$ COK $\rightarrow$ DEL	17
Additional_Info	No info	3	No info	3	No info	3	...	...	No info	3
Duration	2h 50m	170	7h 25m	445	19h	1140	...	...	15h 20m	920
Total_Stops	non-stop	0	2 stops	2	2 stops	2	...	...	1 stop	1
Journey_day	24	24	1	1	9	9	...	...	21	21
Journey_month	3	3	5	5	6	6	...	...	3	3
Dep_hour	22	22	5	5	9	9	...	...	22	22
Dep_min	20	20	50	50	25	25	...	...	00	0
Arrival_hour	1	1	13	13	4	4	...	...	15	15
Arrival_min	10	10	15	15	25	25	...	...	20	20
Price	3897	3897	7662	7662	13882	13882	...	...	6955	6955

d) K-Nearest Neighbors (KNN): it is a non-parametric supervised learning classifier that uses proximity to make classifications or predictions about the grouping of an individual data point. Although it can be used also for regression problems, it is typically used as a classification algorithm, working on the assumption that similar points can be found near one another [16]. KNN can be useful in the case of non-linear data. The output value for the object is computed by the average of K closest neighbours value.

We set the depth of the tree to 3, the learning rate of the model generated by each iteration is 0.1, the number of sub-models is 100, and the loss function is set to squared loss.

e) XGBoost: Extreme Gradient Boosting is a supervised learning technique that uses an ensemble approach based on the Gradient boosting algorithm [18]. It is a scalable end-to-end tree-boosting system, widely used to achieve state-of-the-art results on many machine learning challenges. It can solve both classification and regression problems with good results and minimal effort.

For this model, we set the depth of the tree to 3, the learning rate of the model generated by each iteration is 0.1, the number of sub-models is 100, and the loss function is set to squared loss.

f) Decision Tree: it is a supervised machine-learning approach that uses a tree structure resembling a flowchart to represent decisions, outcomes, and predictions [19]. Such as a tree, each internal node of such architecture represents a test on a dataset feature (e.g. the outcome of a coin toss), each leaf node represents an outcome (e.g. the choice made after simulating all features), and branches represent the decision rules or feature conjunctions that result in the corresponding class labels.

In the Decision tree regression model, we used the *mean squared error function* to measure the quality of a split, which is equal to variance reduction as a feature selection criterion and minimizes the L2 loss using the mean of each terminal node. We used the “best” strategy to choose the split at each node. Meanwhile, when we set the maximum depth of the tree, nodes are expanded until all leaves are pure or until all leaves contain less than $min\_samples\_split$ samples. Then, we considered $min\_samples\_split$ as the minimum number required to split an internal node.

g) Random Forest: it is an ensemble of Decision Trees where each predictor is trained by using a different random subset of the training set, sampled via the bagging or the pasting methods [19]. Generally, Decision Trees performing regression tasks are called Regression Trees, while the corresponding ensemble model is called Random Forest regressor. The prediction of a Regression Tree is simply given by the mean target value of the training data reaching the leaf node. The Regression Tree algorithm tries to iteratively split the training data at each node such that the MSE between the target values and the mean of target values is as low as possible.

We set the random size search with a cross-validation approach to discover the best parameters, the number of cross-validations to 5, and utilize these parameters to construct a random forest regression model. The minimum number of samples needed to split an internal node is set to [2, 5, 10], while the minimum number of samples needed to be at a leaf node is set to [1, 2, 4]. We set the number of trees in the forest to [100, 200, 300, 400, 500].

2) Deep Learning methods: The deep learning methods are implemented based on the *Pytorch* framework. To optimize the parameters of our network, we adopted the Adam solver with batch size 128. We set the learning rate to 0.01. We trained our model on NVIDIA GeForce GTX 3090 GPU. The setting of hyper-parameters is reported below. For Bayesian neural networks, we set the number of ensembles during training as 5, the weight of KL loss as 2×10^{-5}, and the number of ensembles during testing to 128.

a) Transformer: the Temporal Fusion Transformer (TFT) is a novel attention-based deep neural network architecture for time series forecasting. This architecture was introduced to overcome i) Multi-horizon forecasting and heterogeneity and ii) Model explainability. The first issue, to predict a variable usually different data sources are needed, including known future information, other exogenous time series, and static metadata. The relationship between these sources is unknown and makes the multi-step prediction a challenging task. Instead, the Model explainability issue refers to the difficult to explain how a model arrives at its predictions. The explainability methods for deep neural networks are usually not suitable for time series prediction.

We designed a Transformer model with 3 self-attention layers and set the number of heads to 4. Since our input data do not contain sequential information, we replicated it to a dimension of 16 to simulate series data. Next, we applied linear embedding to input data to enrich the features from 13 to 256. Then, we fed these sequential high-dimensional features into the Transformer encoder, which is followed by a global average pooling to get the global features among the sequential features. Finally, we apply two fully connected layers, with the number of output nodes as 256 and 1, to get the final prediction.

b) Fully Connected Network (FCN): the fully-connected artificial neural network is a multi-layer Feed-forward neural network, trained using the error back-propagation technique. It is presently the most popular fully-connected ANN [20]. This model excels at multi-dimensional function mapping and can classify patterns of any complexity. Its objective function is the square of the network error, and the gradient descent method is used to get its least value [21].

We built a FCN with 7 linear layers, each of which is followed by batch normalization and ReLu. We set the number of neurons in the input layer is thirteen, according to the thirteen features in the airfare dataset. We set 1024 hidden layer's neurons and 1 output layer's neurons. We set the batch size to 128.

c) Convolutional Neural Network (CNN): it is a regularized version of multilayer perceptron inspired by biological process. Nowadays, this kind of ANN represents the state-of-the-art for image classification and pattern recognition. A CNN consists of an input layer, an output layer and multiple hidden layers in between. The hidden layers are typically a set of convolutional layers with a Rectified Linear Unit (ReLU) as activation function, followed by a flatten layer and fully connected layers (i.e. dense layers).

We built a convolutional neural network with 7 convolutional layers, each of which is followed by batch normalization and ReLu. We set the number of neurons in the input layer is thirteen, according to the thirteen features in the airfare dataset. We set 1024 hidden layer's neurons and 1 output layer's neurons. We set the batch size to 128.

d) Our proposed Bayesian FCN and CNN: Bayesian neural networks are popular due to their ability to quantify the uncertainty in their predictive output. With canonical neural networks, the weights between the different layers of the network take single values. In a Bayesian neural network, the weights take values according to a probability distributions [22]. The distributions finding process is called *marginalization*. Large enough amounts of data are nearly mandatory to train these networks and produce accurate probability distributions. This makes them more robust and reduce the likelihood of overfitting phenomena.

There are several benefits in adopting Bayesian neural networks: i) they are more resilient and generalizable than other neural networks; ii) they can quantify the uncertainty in their predicted output; and iii) they can be used for a wide range of practical applications [23]. In contrast, however i) they can be harder to train than other models and need an understanding of probability and statistics; and ii) they can be slower to converge than other models and frequently require more data. Because the network's weights are distributions rather than single values, more data is necessary to correctly predict the weights [24].

Firstly, we built a fully connected Bayesian neural network with two bayesian layers and five linear layers, each followed by batch normalization and ReLu. We set the number of neurons in the input layer is thirteen, according to the thirteen features in the airfare dataset. We set 1024 hidden layer's neurons and 1 output layer's neurons. We set the batch size to 128. The structure is depicted in Fig. 2.

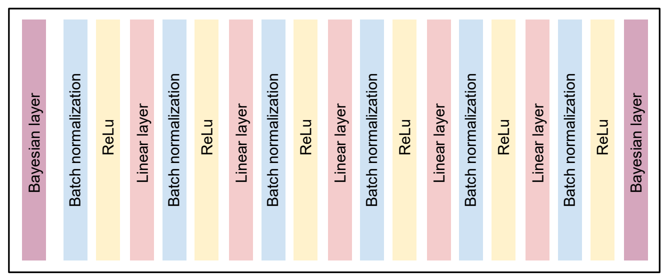

Fig. 2. Bayesian Fully Connected Neural Network implemented architecture

Finally, we implemented a Bayesian Convolutional Neural Network with one bayesian layer and six convolutional layers, each followed by batch normalization and ReLu. We set the number of neurons in the input layer is thirteen, according to the thirteen features in the airfare dataset. We set 1024 hidden layer's neurons and 1 output layer's neurons. We set the batch size to 128. The structure is depicted in Fig. 3.

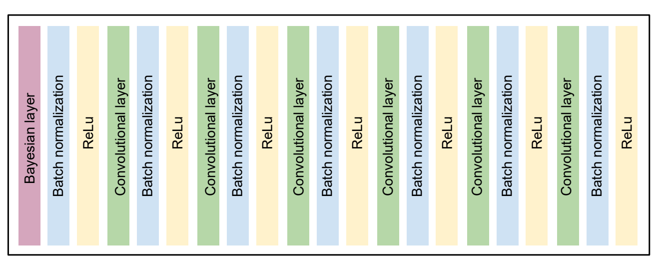

Fig. 3. Bayesian Convolutional Neural Network implemented architecture

V. Experimental Results

In this Section, we present our experimental results. First, we briefly describe the statistical indicators used to analyse and compare the predictions. Then, we show and compare all the implemented models. Moreover, we prove that our proposed deep learning, Bayesian-based models perform better overall. Finally, we also provide a computational running time detailed comparison.

A. Statistical indicators

State of the art has a large number of statistical indicators to evaluate neural networks performances in terms of predictions [25]. In the study of time-series, the three main indexes adopted are:

- $RMSE$ - the *Root mean square error* represents standard deviation of the residuals (prediction errors);
- MAE - the *Mean absolute error* between predicted and observed values;
- $MAPE$ - the *Mean Absolute Percent Error* that is, the error magnitude in percentage terms;
- R^2 - the *Coefficient of Determination*, defined as square of the correlation (R) between predicted and observed values.

Their mathematical expressions are shown in Equations 1, 2, 3 and 4 respectively, where y_i represents the true target value, $\hat{y}_i$ is the predicted value, n is the number of observed data and $\bar{y}_i$ stands for the mean value.

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

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2}$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{3}$$

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

Where $MAPE$ is expressed in percentage units, and $RMSE$ and MAE in absolute units. For $RMSE$, MAE and $MAPE$, a lower value indicates a smaller error, hence better performance. R^2, on the other hand, shows the correlation between real and predicted values, where a value of 1 means complete correlation, while lower values indicate a lower correlation factor.

B. Results remarks

Table III shows the numerical results in terms of statistical indicators (i.e. $RMSE$, MAE, $MAPE$ and R^2) by providing a complete overview about all the investigated methods. Specifically, we implemented representative traditional machine learning methods and deep neural networks presented in Section IV-C-1 and Section IV-C-2, respectively.

Experimental results highlight that, among traditional machine learning methods, Decision Tree, XGBoost, and Random Forest achieve significantly better performance than Lasso Regression, Ridge Regression, Support Vector Regression, and KNN.

TABLE III
PERFORMANCE COMPARISON OF PROPOSED TRADITIONAL AND DEEP LEARNING METHODS. BEST RESULTS ARE HIGHLIGHTED IN BOLD.

Methods	RMSE (↓)	MAE (↓)	MAPE (↓)	R^2 (↑)
Lasso Regression	3299.82	2396.99	31.96%	0.47
Ridge Regression	3289.24	2407.41	32.02%	0.47
Support Vector Regression	3167.65	1913.42	20.70%	0.51
K-Nearest Neighbors	2902.66	1690.64	18.34%	0.59
XGBoost	1567.57	749.70	8.76%	0.88
Decision Tree	1937.03	704.95	8.10%	0.82
Random Forest	1566.37	**645.78**	**7.57%**	**0.89**
Transformer	1733.29	835.79	10.18%	0.85
Fully Connected Network	1564.42	710.15	8.15%	0.88
Bayesian FCN (ours)	1491.36	**698.53**	**8.04%**	0.89
Convolutional Neural Network	1486.25	756.20	8.82%	0.89
Bayesian CNN (ours)	**1414.51**	749.70	8.81%	**0.91**

Besides, Random Forest achieves the best performance among traditional machine learning methods having $MAE = 645.78$, $MAPE = 7.57\%$ and $R^2 = 0.88$.

Deep learning methods (i.e. Fully Connected Network, Bayesian FCN, Convolutional Neural Network, and Bayesian CNN) achieve better performances than all traditional methods, with better values of $RMSE$ and R^2. We must highlight that with our proposed Bayesian layers, the performance of CNN and FCN can be both improved. Although Transformer is recently popular, it is the worst performing method among deep learning ones, which means it is not an ideal candidate for airfare prices prediction tasks. On the other hand, it should be noticed that Random Forest, as a traditional method, still achieves the best performance in MAE and $MAPE$ over all the involved methods, which means that traditional machine learning methods still play an important role in our task.

C. Running time comparisons

Finally, we compare the run times for all methods. Table IV shows the run times of all compared methods. Among the traditional models, the measured results highlight that Decision Tree Regression Model runs the fastest, and the Random Forest takes the longest time. However, among traditional machine learning models, the Random Forest performs better in terms of accuracy, as discussed in Section V-B. Instead, among the deep learning methods the Fully Connected Network runs the fastest and has the least number of parameters. The Bayesian CNN model takes the longest time and has the largest number of parameters.

VI. CONCLUSIONS

The objective of this manuscript is to propose a methodology for airfare prices prediction. In detail, we wanted to address this topic from the customer's point of view. Customers are in a disadvantaged position in comparison with airlines because they cannot easily access useful data to predict the dynamics of airline ticket prices. Usually, consumers cannot easily access useful data to predict the dynamics of airline ticket prices like airline companies do. In addition, the scientific literature proposes highly advanced dynamic

TABLE IV
RUNNING TIME AND # OF PARAMETERS OF IMPLEMENTED METHODS

	Methods	Time per Sample [ms]	# of parameters
Traditional machine learning methods [CPU-based computation]	Lasso Regression	0.10	n.a.
	Ridge Regression	0.10	n.a.
	Support Vector Regression	0.47	n.a.
	K-Nearest Neighbors	0.28	n.a.
	XGBoost	0.94	n.a.
	Decision Tree	0.04	n.a.
	Random Forest	4.56	n.a.
Deep learning methods [GPU-based computation]	Transformer	1.22	26.3
	Fully Connected Network	0.45	5.28
	Bayesian FCN	0.83	5.29
	Convolutional Neural Network	1.10	21.0M
	Bayesian CNN	11.03	23.1M

pricing strategy techniques for airlines, almost completely ignoring customers' needs. Basically, customers have two strikes against one. Therefore, the results of these algorithms can represent the foundation to develop new business models or enhance existing ones.

To this aim, we specifically designed, optimized and compared twelve state-of-the-art machine learning architectures. Then, we did a systematic comparison of seven traditional machine learning methods and five deep learning methods. Moreover, we proposed two innovative Bayesian neural networks for airfare prices prediction. We exploited an open dataset of 10683 domestic routes in India from March 2019 to June 2019.

The experimental results show that Random Forest achieves the best prediction accuracy among traditional methods despite longer computation time. Furthermore, deep learning-based methods achieve on average better accuracy than traditional ones. Instead, our proposed novel Bayesian neural networks further improve prediction accuracy. Finally, comparing the performance indices and execution times of each model, we can state that Random Forest is the best method among the evaluated ones. Despite the expense of execution times, Random Forest's performance is close to the best deep-learning models' one and, on top of that, they do exploit leaner neural architectures.

Lastly, we planned to extend this research for future work by exploiting a larger dataset and by applying feature selection techniques to improve forecast results and explore the possibilities of deep neural networks.

REFERENCES

[1] S. Saharan, S. Bawa, and N. Kumar, "Dynamic pricing techniques for intelligent transportation system in smart cities: A systematic review," *Computer Communications*, vol. 150, pp. 603–625, 2020.

[2] J. A. Abdella, N. M. Zaki, K. Shuaib, and F. Khan, "Airline ticket price and demand prediction: A survey," *Journal of King Saud University-Computer and Information Sciences*, vol. 33, no. 4, pp. 375–391, 2021.

[3] A. Hortaçsu, A. Oery, and K. R. Williams, "Dynamic price competition: Theory and evidence from airline markets," National Bureau of Economic Research, Tech. Rep., 2022.

[4] C. Giachino, L. Bollani, A. Bonadonna, and M. Bertetti, "Reinforcement learning for content's customization: a first step of experimentation in skyscanner," *Industrial Management & Data Systems*, vol. 121, no. 6, pp. 1417–1434, 2021.

[5] M. Heidari and S. Rafatirad, "Using transfer learning approach to implement convolutional neural network model to recommend airline tickets by using online reviews," in *2020 15th International Workshop on Semantic and Social Media Adaptation and Personalization (SMA*. IEEE, 2020, pp. 1–6.

[6] Bureau of Transport Statistics, "Airline origin and destination survey (db1b)," https://www.transtats.bts.gov/Tables.asp?QO_VQ=EFI&QO_anzr=Nv4yv0r%FDb4vtv0%FDn0q%FDQr56v0n6v10%FDf748rB%FD%FLQOEO%FM&QO_fu146_anzr=b4vtv0%FDn0q%FDQr56v0n6v10%FDf748rB, last access: 2023-01-29.

[7] Bureau of Transport Statistics , "Data bank 28ds - t-100 domestic segment data (world area code)," https://www.bts.gov/browse-statistical-products-and-data/bts-publications/data-bank-28ds-t-100-domestic-segment-data, last access: 2023-01-29.

[8] T. Wang, S. Pouyanfar, H. Tian, Y. Tao, M. Alonso, S. Luis, and S.-C. Chen, "A framework for airfare price prediction: a machine learning approach," in *2019 IEEE 20th international conference on information reuse and integration for data science (IRI)*. IEEE, 2019, pp. 200–207.

[9] W. Groves and M. Gini, "Optimal airline ticket purchasing using automated user-guided feature selection," in *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[10] Z. Wang, "Rwa: A regression-based scheme for flight price prediction," 2020.

[11] S. BUYRUKOĞLU and Y. YILMAZ, "An approach for airfare prices analysis with penalized regression methods," *Veri Bilimi*, vol. 4, no. 2, pp. 57–61.

[12] H.-C. Huang, "A hybrid neural network prediction model of air ticket sales," *Telkomnika Indonesian Journal of Electrical Engineering*, vol. 11, no. 11, pp. 6413–6419, 2013.

[13] K. Tziridis, T. Kalampokas, G. A. Papakostas, and K. I. Diamantaras, "Airfare prices prediction using machine learning techniques," in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 1036–1039.

[14] Kaggle, "Flight prediction," https://www.kaggle.com/datasets/chandruuk/flight-prediction?select=Data_Train.xlsx, last access: 2023-01-29.

[15] J. T. Hancock and T. M. Khoshgoftaar, "Survey on categorical data for neural networks," *Journal of Big Data*, vol. 7, no. 1, pp. 1–41, 2020.

[16] G. Bontempi, S. Ben Taieb, and Y.-A. L. Borgne, "Machine learning strategies for time series forecasting," in *European business intelligence summer school*. Springer, 2012, pp. 62–77.

[17] D. A. Pisner and D. M. Schnyer, "Support vector machine," in *Machine learning*. Elsevier, 2020, pp. 101–121.

[18] X. Liao, N. Cao, M. Li, and X. Kang, "Research on short-term load forecasting using xgboost based on similar days," in *2019 International conference on intelligent transportation, big data & smart city (ICITBS)*. IEEE, 2019, pp. 675–678.

[19] M. Fratello and R. Tagliaferri, "Decision trees and random forests," *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, vol. 1, p. 3, 2018.

[20] S. Ding, C. Su, and J. Yu, "An optimizing bp neural network algorithm based on genetic algorithm," *Artificial intelligence review*, vol. 36, no. 2, pp. 153–162, 2011.

[21] Z. Xiao, S.-J. Ye, B. Zhong, and C.-X. Sun, "Bp neural network with rough set for short term load forecasting," *Expert Systems with Applications*, vol. 36, no. 1, pp. 273–279, 2009.

[22] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.", 2022.

[23] L. V. Jospin, H. Laga, F. Boussaid, W. Buntine, and M. Bennamoun, "Hands-on bayesian neural networks—a tutorial for deep learning users," *IEEE Computational Intelligence Magazine*, vol. 17, no. 2, pp. 29–48, 2022.

[24] E. Goan and C. Fookes, "Bayesian neural networks: An introduction and survey," in *Case Studies in Applied Bayesian Data Science*. Springer, 2020, pp. 45–87.

[25] C. A. Gueymard, "A review of validation methodologies and statistical performance indicators for modeled solar radiation data: Towards a better bankability of solar projects," *Renewable and Sustainable Energy Reviews*, vol. 39, pp. 1024–1034, 2014.