DOI: https://doi.org/10.33103/uot.ijccce.22.3.8

# A Framework for Predicting Airfare Prices Using Machine Learning

Heba Mohammed Fadhil[1], Mohammed Najm Abdullah[2], Mohammed Issam Younis[3]

*[1]Department of Information and Communication, Al-Khwarizmi College of Engineering, University of Baghdad, Baghdad, Iraq*

*[2]Computer Engineering Department, University of Technology, Baghdad, Iraq*

*[3]Department of Computer Engineering, College of Engineering, University of Baghdad, Baghdad, Iraq*

*[1]ce.19.15@grad.uotechnology.edu.iq, [2]mohammed.n.abdullah@uotechnology.edu.iq, [3]younismi@coeng.uobaghdad.edu.iq*

***Abstract****—Many academics have concentrated on applying machine learning to retrieve information from databases to enable researchers to perform better. A difficult issue in prediction models is the selection of practical strategies that yield satisfactory forecast accuracy. Traditional software testing techniques have been extended to testing machine learning systems; however, they are insufficient for the latter because of the diversity of problems that machine learning systems create. Hence, the proposed methodologies were used to predict flight prices. A variety of artificial intelligence algorithms are used to attain the required, such as Bayesian modeling techniques such as Stochastic Gradient Descent (SGD), Adaptive boosting (ADA), Decision Trees (DT), K-nearest neighbor (KNN), and Logistic Regression (LR), have been used to identify the parameters that allow for effective price estimation. These approaches were tested on a data set of an extensive Indian airline network. When it came to estimating flight prices, the results demonstrate that the Decision tree method is the best conceivable Algorithm for predicting the price of a flight in our particular situation with 89% accuracy. The SGD method had the lowest accuracy, which was 38 %, while the accuracies of the KNN, NB, ADA, and LR algorithms were 69 %, 45 %, and 43 %, respectively. This study's presented methodologies will allow airline firms to predict flight prices more accurately, enhance air travel, and eliminate delay dispersion.*

***Index Terms****— Machine learning, Prediction model, Airline price prediction, Software testing,*

## I. INTRODUCTION

Computers that have the capability to mimic human thought and reason are considered a part of the larger field of Artificial Intelligence (AI), which is recognized in the computer science world. AI assists machines in mimicking human cognitive abilities by learning to build recommender systems [1],[2], prediction [3],[4],[5], identification [6], recognition [7],[8],[9], and decision-making ability. It is widely agreed that artificial intelligence has greatly boosted efficiency in several industries, and computers utilizing AI are becoming increasingly popular in the problem-solving arena [10]. Several industries have embraced the implementation of Machine Learning (ML), especially those involved in safety-critical endeavors. However, a significant increase in susceptibility was detected, leading to catastrophic failure.

Researchers are working hard to include ML algorithms in their study models for better identification, as seen by the large number of articles that have been carefully evaluated and

*Received 4/October/2021; Accepted 24/December/2021*

DOI: https://doi.org/10.33103/uot.ijccce.22.3.8

recently published. However, ML methods rely on datasets for training and testing. An ML algorithm model is difficult to perform without training and testing datasets. For many real-world applications, anticipating a price range is more feasible and desirable than predicting an actual value. In this scenario, price prediction may be considered a problem of categorization [11]. Predictive techniques are generally categorized into two categories based on their jobs. The first approach is meant to anticipate price trends in a time series format, such as stock prices and the prediction of oil prices. The second class of techniques focuses on forecasting the price of specific products based on their characteristics, such as house prices or flight ticket prices. This study focuses on the second type of prediction job.

However, numerous studies have been conducted in the research community to address these challenges. However, it is necessary to determine which ML algorithm is efficient for flight price estimation among various ML methods. For the most part, researchers focus on finding effective ways to use the ML algorithm's characteristics, while others want to make it run more smoothly. Although several studies have demonstrated that the performance of one ML algorithm is effective, no study has focused on this topic. It is thus necessary to investigate and determine which flight price estimation classification ML technique is successful in various ML algorithms. According to our understanding, It would be the first study to determine which machine learning algorithm performs optimally across a range of ML algorithms. [12].

ML algorithms, tasked with learning based on training data, use their newly acquired information to apply to real-world scenarios. Imagine a book-classification algorithm utilized in a bookstore. A training set (a table of training data and answers) can be used to classify books correctly. To obtain information about a book, details such as title, author, and even every word inside the book may be available. The ML algorithm uses a training set for learning. The Algorithm can categorize new books in the bookstore when they arrive by considering the information it has on book classification [13]. This study establishes a model for selecting the most efficient ML algorithm for a flight price estimate. However, the following are our most significant contributions: solving the problem of detecting abnormalities and intrusions in air traffic flight fares. Initially, a model was created to discover abnormalities and parasitism in the air network. It then determines the most efficient ML classification in various ML methods using a mathematical group approach to select an efficient ML.

The rest of the paper is structured as follows: In Section II, demonstrate relevant work relating to airfare prices. Section III gives a brief introduction to ML, and Section IV discusses the dataset used in this study and covers the study framework. In Section V, the approach and specification of the platform used are demonstrated, and in Section VI, the evaluation metrics and analyses are presented. Similarly, the results and discussion analyses are presented in Section VII, and finally, the conclusions are presented in Section VIII.

## II. RELATED WORK

Currently, airlines utilize complex policies and methods to assign airfare prices dynamically. These methods take numerous economic, marketing, commercial, and social aspects that are directly linked to ultimate ticket costs. Because the price models used by airlines are highly complex, it is very difficult for a customer to buy the lowest price of an air ticket because the prices change dynamically.

As a result, various approaches have been developed [14] and [15] that have recently been proposed to give the purchaser a ticket by predicting airfare prices. The majority of

Received 4/October/2021; Accepted 24/December/2021

DOI: https://doi.org/10.33103/uot.ijccce.22.3.8

these methods employ advanced prediction models from the field of computational intelligence research, called ML. Groves and Gini [15] used Partial least squares (PLS) regression to optimize the airline purchases of tickets with 75.3% accuracy.

Furthermore, Tziridis et al. [16] focused on the prediction of airfare costs for Greece. After a brief introduction to ML, the author has learned about their data collection approach, ML model selection of features, and assessment. The required information was gathered from the Greek airline business. After modifying the data, eight characteristics were examined for each trip. They used the regression tree, bagging regression tree, regression Support Vector Machine (SVM) (polynomial, linear), and linear regression. The external regression tree approach performs other strategies by achieving an accuracy of 87.42 percent.

Vu et al. [17] created interpretable predictions by stacking random forest and multilayer perceptron with two distinct prediction models. Researchers utilize the fine-determined weights to stack the measurement metric with R-squared. Their suggested prediction model is 7.7% better than multilayer perceptron and 4.4% better than random forest. Deep regressor stacking has been reported by Santana et al. [18] to improve prediction accuracy. Regressors for the proposed technique include Random forest (RF) and SVM, which may be readily applied to other comparable problem areas.

All of the studies mentioned above only used a limited number of ML models to predict airline prices worldwide, emphasizing certain classic models. However, the author's best knowledge still does not understand the performance of state-of-the-art ML models to address this problem. The efforts and information related to the selection of ML for flight price estimation are presented in this section. Researchers first explain the corresponding anomaly and then gradually get to the bottom of the pace, which will aid new researchers.

## III. MACHINE LEARNING ALGORITHMS

Several various learning algorithms exist. ML comprises the following features of training data from the standpoint of training data characteristics: a supervised learning algorithm has a training set, whereas an unsupervised learning algorithm has no training set. It is their job to learn from the data they have about the real world and not anyone else's. Learning algorithms that do not require supervision are mostly focused on locating subtle patterns in the data. For instance, consider an ML algorithm accessing a user's social network profile. An unsupervised learning system allows the social network operator to target ads more directly for certain groups of users. Finally, there may be a reinforcement learning strategy used by ML algorithms. Reinforcement learning occurs when machines and their environments can provide external input to train them. Teaching dogs to sit or jump is a good analogy for this strategy. When the dog gets it right, it is given a tiny treatment (positive feedback). It will be ignored if it fails to act appropriately. For instance, let us suppose an ML algorithm in the computer science field plays games against an opponent. Correct play results in winning; thus, players should practice and repeat plays that lead to victories, while they should avoid actions that result in losses. ML can be used for the following typical activities:

1) Classification: assign each instance category to each data, e.g., image classification and recognition of handwriting.
2) Regression: with every data instance to forecast a value, i.e., prediction of temperature, age, and revenue.
3) Clustering: in a homogenous region for partition instances, for example, pattern recognition and segmentation of market/image.
4) Reduction in size: reduction in training complexity, e.g., data set displays and pre-processing data.

Received 4/October/2021; Accepted 24/December/2021

DOI: https://doi.org/10.33103/uot.ijccce.22.3.8

5) Control: controlling reward actions, e.g., playing the game.

ML can also be classified as either traditional or deep learning methods. Classical ML includes algorithms such as decision trees, SVM, linear regression, and naive Bayes.

Increases in processor and memory capacity have helped ML grow rapidly in the last few years. Another notable development is the growing number of scientific papers that proposed new algorithms to use ML techniques, as evidenced by the increasing number of scientific papers proposing various versions or combinations of ML algorithms. Since then, ML algorithms have been divided into multiple groups according to their intended use [19].

## IV. METHODOLOGY

Complex and large-scale systems require creative techniques to discover, classify, and analyze vast volumes of data. An ML strategy for uncovering, collecting, classifying, and prioritizing data inside big datasets was proposed in this study. Flight planning, considered one of the most difficult problems in the industrial world, is subject to various uncertain variables. One such condition is delay occurrence, which arises from many factors and puts enormous costs on airlines, operators, and travelers. With these concerns in mind, researchers implemented flight price prediction using the provided methodologies based on ML algorithms to obtain the desired results. To estimate the occurrence and magnitude of delay in a network, the parameters that permitted effective estimation of delay were found. This was Followed by Bayesian modeling, stochastic gradient descent, adaptive boosting, decision tree, logistic regression, and the K-nearest neighbor technique to estimate their occurrences and magnitudes. These approaches were evaluated on an airline network of an Indian flight dataset. The following sections outline the suggested block diagram procedure sequence shown in *Fig. 1*, which includes data preparation, data labeling, and classification algorithms to categorize flight planning:

### A. Dataset Preprocessing

The modeling process was presented using the Kaggle Flight Prices dataset. This database contains 10,682 sequences, each of which is identified by an accession number and contains Comma-Separated Values of information (CSV) file providing details such as the airline, dates of the journey, origin and destination, route, departure and arrival times, duration, number of stops, and additional information.

An airline classifier was applied to the flight dataset. The names of the airlines are listed in Table I. Following duplicate removal, there were 12 different airlines in this dataset. Airline data were used to create a classifier for airline labels. Each record is labeled with a number in the range of 1 to 11, where only one number is used as a label, as specified by its airline. In the majority of instances, a researcher will use a framework to help them partition accessible data into two groups, one for training and one for testing. All labeled data were divided randomly with 7:3 ratios between a training set and a test set after the data labeling stage. Utilizing specific tools is one of the most popular ways to ensure that the training and testing data are appropriately separated. If significant changes are made to ML algorithms, prediction models, or training data, it may be necessary to recreate the ML fully. Such changes almost certainly lead to regression in the current functioning.

Received 4/October/2021; Accepted 24/December/2021

DOI: https://doi.org/10.33103/uot.ijccce.22.3.8

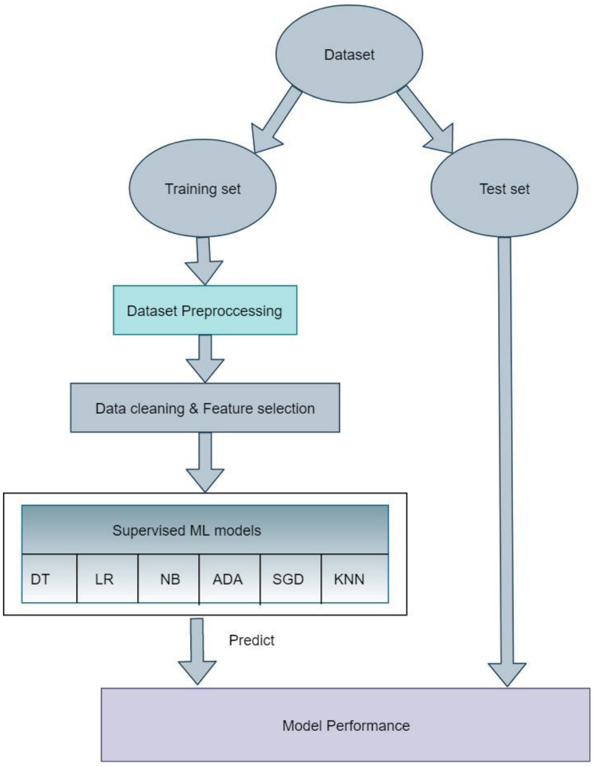


FIG.1. BLOCK DIAGRAM OF A PROPOSED METHOD SEQUENCE.

TABLE I. CODE NUMBERS OF AIRLINES

| Airlines number | Airlines name |
|---|---|
| 1 | IndiGo |
| 2 | Air India |
| 3 | Jet Airways |
| 4 | SpiceJet |
| 5 | Multiple carriers |
| 6 | GoAir |
| 7 | Vistara |
| 8 | Air Asia |
| 9 | Vistara Premium economy |
| 10 | Jet Airways Business |
| 11 | Multiple carriers Premium economy |

The alterations to the model will likely cause only minor changes to its design, but the entire model itself could conceivably transform. The model's final accuracy was estimated to be 70 percent training and 30 percent testing. It is best to set a ratio that lies somewhere between the training and testing processes. Thus, the outcomes of the system were not compromised. The training and testing data were used to teach the classification step, after which the final results were determined.

Received 4/October/2021; Accepted 24/December/2021

DOI: https://doi.org/10.33103/uot.ijccce.22.3.8

### B. Classification and Machine learning algorithms

After converting the data from the flight data set to numeric values that are part of 12 different classes, six different classifiers were employed in sequence to predict the price of a flight. Each category in this classifier represents a different attribute. One approach for classifying a classification system's expected outcome is to use a numerical or binary array format. The classifiers were linearly performed as follows:

1. **The decision tree** has a single root node, numerous branches, and numerous leaf nodes. This strategy begins with the user splitting the data into progressively smaller subsets until a decision tree with nodes and leaves is completed. Each branch represents a class, each leaf represents a specific attribute, and the test is used to obtain that class or attribute. The root node is located at the top of the tree [20], [21]. The detailed pseudocode of Algorithm (1) is as follows:

Algorithm 1: DT Pseudocode

```
Function DT(D, Feature_Attributes, Target_Attributes)
    Create a root node N
    Set N to the mode target feature value in D
          If all target feature values are the same:
               return N
         Else:
              pass
        If Feature_Attributes is empty:
            return N
        Else:
            AttF = Attribute from Feature_Attributes with the largest information gain    value
          N= AttF
    For values in AttF:
         Add a new node below N where node_values = (AttF == values)
        Sub_D = (AttF == values)
  If Sub_D == empty:
       Add a leaf node l where l equals the mode target value in D
  Else:
      Add Sub_Tree with DT(Sub_D,Feature_Attributes = Feature_Attributes without   AttF,
Target_Attributes)
```

2. **The naïve Bayes** algorithm uses a conditional probability and is simple. The model is a probability table that is updated using training data. The "probability table" is based on feature values where the class probabilities are needed to anticipate a new observation. The underlying assumption is conditional independence; hence, the term "naive" In reality, it is unlikely that all the input features are independent. The benefits of NB include simplicity, high performance, a low number of predictors and data points, handling of continuous and discrete data, binary and multi-class classification issues, and probabilistic predictions [20], [22]. The pseudocode of Algorithm (2) is as follows:

Received 4/October/2021; Accepted 24/December/2021

DOI: https://doi.org/10.33103/uot.ijccce.22.3.8

Algorithm 2: NB Pseudocode

```
Function NB(D, Feature_Attributes, Target_Attributes)
  Load the dataset
  Select target attribute and predictor attribute
     For (each preditor label)
       {
          calculate Mean;
          calculate Standard Deviation;
       }
     For (each class in target attribute )
      {
          For (each preditor label)
          {
            calculate Probability;
          }
      }
    For (each class in target attribute )
      {
        calculate Likelihood;
      }
```

3. **Logistic regression** was used to classify data. It calculates the probability of an event occurring (in terms of 0 and 1) depending on the values of the input variables. For example, predicting whether an email is spam is a binomial outcome of logistic regression. Logistic regression may also predict categorical target variables. Linear regression is used to predict the values of continuous variables, such as the price of real estate over three years [23], [20]. Below is the Algorithm's (3) pseudocode in its entirety.

Received 4/October/2021; Accepted 24/December/2021

DOI: https://doi.org/10.33103/uot.ijccce.22.3.8

Algorithm 3: LR Pseudocode

```
Function LR (predictor_attributes, target_attribute,weights)
  {
    Calculate gradient_descent;
    Return weights+ learning rate * gradient_descent;
  }
Normalize the dataset;
Repeat
  {
    Weights=grad (param);
    update weights;
  }
  until convergence
x= dot product of predictor variables and update weights;
prediction _limit=sigmoid function(x);
Predict target class
```

4. **The K-nearest neighbor** algorithm is a classification algorithm. It employs a database with data points classified into various classes, and the method attempts to categorize the sample data point provided to it as a classification issue. Because KNN makes no assumptions about the underlying data distribution, it is referred to as non-parametric. The following are some of the benefits of the KNN algorithm: it is a simple strategy that can be readily implemented. The construction of the model was inexpensive. It is an incredibly versatile categorization technique that is best suited for multimodal classes. It is sometimes the best technique for function prediction using expression profiles [24], [25]. The Algorithm (4) is detailed in the pseudocode provided below.

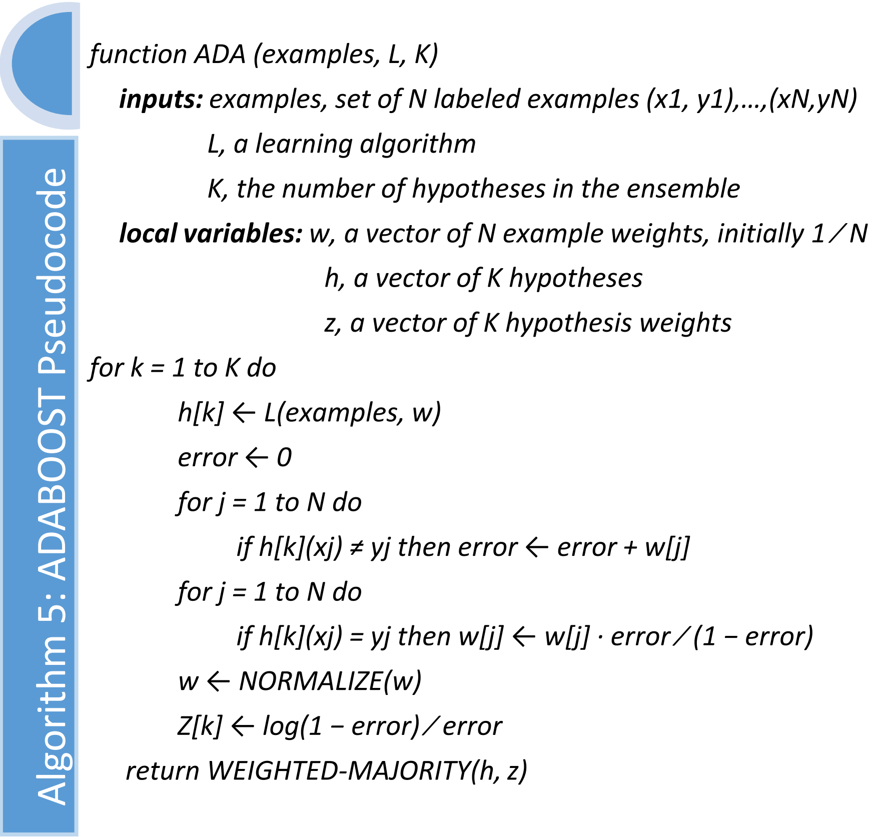
Algorithm 4: KNN Pseudocode

```
Function KNN (K, Feature_Attributes, Target_Attributes)
  Load the training and test data
  Choose the value of K

For (each point in test data)
  {
    find the Euclidean distance to all training data points

    store the Euclidean distances in a list and sort it

    choose the first k points

    assign a class to the test point based on the majority of  classes present in
    the chosen points
  }
```

Received 4/October/2021; Accepted 24/December/2021

DOI: https://doi.org/10.33103/uot.ijccce.22.3.8

5. ***Adaptive boosting,*** an adaptive boosting ML method, was used to improve the classification results. Meta-algorithms are useful when combined with other learning algorithms because they can boost the overall performance of learning algorithms. It is adaptive in that subsequent classifiers constructed have been modified to examples that have been misclassified by earlier classifiers. In other words, the fundamental principle of ADA is to repeatedly use a weak classifier and then tweak the weight given to each example with each call. In this way, the misclassified examples will be weighted higher than the correctly classified ones, causing the new classifier to favor the misclassified examples [26], [27]. The Algorithm (5) is detailed in the following pseudocode:



Algorithm 5: ADABOOST Pseudocode

```
function ADA (examples, L, K)
   inputs: examples, set of N labeled examples (x1, y1),…,(xN,yN)
           L, a learning algorithm
           K, the number of hypotheses in the ensemble
   local variables: w, a vector of N example weights, initially 1 ⁄ N
                    h, a vector of K hypotheses
                    z, a vector of K hypothesis weights
for k = 1 to K do
      h[k] ← L(examples, w)
      error ← 0
      for j = 1 to N do
         if h[k](xj) ≠ yj then error ← error + w[j]
      for j = 1 to N do
         if h[k](xj) = yj then w[j] ← w[j] · error ⁄ (1 − error)
      w ← NORMALIZE(w)
      Z[k] ← log(1 − error) ⁄ error
   return WEIGHTED-MAJORITY(h, z)
```

6. ***The stochastic gradient descent*** model is a highly effective learning algorithm for linear classifiers. Simply replace the real gradient obtained from the full dataset with an approximation based on a randomly selected portion. A stochastic (or "operational") gradient descent algorithm assigns a gradient to each learning element, approximating the gradient of the cost function. The settings were adjusted to reflect estimated gradients. The model parameters were recalculated after each learning object. The stochastic gradient descent technique is much faster than the normal gradient descent for large datasets [28],[29]. The pseudocode of Algorithm (6) is presented here.

*Received 4/October/2021; Accepted 24/December/2021*

DOI: https://doi.org/10.33103/uot.ijccce.22.3.8

Algorithm 6: SGD Pseudocode

```
Function SGD ( data set , model parameters , lr )
 {
    while (convergence condition not met)
      {
        for (each instance in data set)
           {
               gradient = compute gradient ( parameters , instance );
               parameters = parameters − l r ∗ gradient;
           }
      }
      return parameters
 }
```

## V. EXPERIMENTAL WORK

Researchers put the recommended strategies to the test airline networks and then compared their outcomes to find the optimum option. As mentioned, the data were divided into a training set (70 percent) and a test set (30 percent).

Flight price is a significant challenge in today's electronic environment. Various classification methods have been employed to address the problem of flight price detection techniques. Several stages must be completed to detect flight prices. For classification classifier implementation, a model was trained on each training set, and the performance of the model was tested on the testing dataset.

The model designed, which is based on the ML classifier, is trained and capable of supporting a broader range of algorithms as well as extremely large datasets. In addition, a model was developed based on the Python programming language using the ml library. The computer specifications are as follows.

- Processor: Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz, with a clock speed of 2.80 GHz.
- Memory: RAM (random access memory): 8.00 GB.
- Operating System: Windows 10 64-bit operating system

## VI. EVALUATION METRICS

The task of this phase was to compute the classifier's success rate. The classifier accuracy was measured by assessing the accuracy of the expected class labels corresponding to the actual class labels. Classification correctness can be evaluated by calculating the number of class examples correctly recognized (true positive), the number of well-recognized examples not class-relevant (true negatives), as well as examples that were either incorrectly classified (false positive) or which were not classified (false negatives) [30]. Quantitative analysis measurements were as follows:

- Accuracy is defined as the number of correct predictions divided by the total number of predictions, as shown in the equation stated in Eq. (1), which is also defined as the high viability of the model.

Received 4/October/2021; Accepted 24/December/2021

DOI: https://doi.org/10.33103/uot.ijccce.22.3.8

$$Accurcy = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}} \ldots\ldots\ldots.(1)$$

- Precision relates to the accuracy of documents placed in a class and how well the class represents what they describe. The precision of class ci, denoted by the symbol (Pi), is measured in the following way, represented by eq. (2):

$$\boldsymbol{P_i} = \frac{\boldsymbol{TP_i}}{\boldsymbol{TP_i+FP_i}} \ldots\ldots\ldots.(2)$$

- The extent to which a classifier recognizes documents belonging to a class is called recall, which is shown in Eq. (3). The following equation can be used to count the class ci recall (Ri):

$$\boldsymbol{R_i} = \frac{\boldsymbol{TP_i}}{\boldsymbol{TP_i+\ FN_i}} \ldots\ldots\ldots.(3)$$

In this case, ***TPi*** points to a true-positive value. ***FPi*** stands for false positives, and ***FNi*** represents false negatives.

- The F1 metric is the rate at which the precision and recall are synchronized. If F1 is high, this indicates that the system performs well as a whole. F1 is described as follows, as stated in Eqs. (4) and (5), as follows:

$$\boldsymbol{F1} = \frac{\boldsymbol{2\times precision\ \times recall}}{\boldsymbol{precision+recall}} \ldots\ldots\ldots.(4)$$

$$= \frac{\boldsymbol{2TP}}{\boldsymbol{2TP+FP+FN}} \ldots\ldots\ldots.(5)$$

## VII. RESULTS AND DISCUSSION

When factors such as airline, origin, and destination are considered, a better outcome with more precise output is obtained, according to the entire project. In addition, the decision tree method had the highest degree of precision among the algorithms used. When new attributes were used, researchers achieved the highest accuracy for the decision tree method, which was 89 percent. The decision tree method also assists in identifying the specific terms that are most frequently seen in the dataset set and, as a result, has the greatest impact on the accuracy of the calculation.

It is possible to assign or anticipate the target value of a new instance using the methods described above by comparing all of the instances' characteristics and their values to those shown in the decision tree model. Tables II and III present the experimental results. When estimating flight prices in Indian networks, the proposed decision tree approach had accuracy values of 89 percent. Class prediction displays the percentage of retrieved instances relevant to the class. For various flight price magnitudes in the Indian networks, the precision of class prediction using the decision tree is 89 percent for various flight price magnitudes. The recall rate was defined as the percentage of relevant instances retrieved. The Indian networks' class recall rates using the decision tree were 89 percent. Additionally, the results of the Bayesian classification provided in Tables II and III had accuracy levels of 45. There was also precision and recall for the Indian networks, with values of 61 percent and 45 percent, respectively.

When the K-means algorithm was used to predict the incidence of delays in the Indian networks, the system had an accuracy rate of 69 percent. Regarding forecasting the extent of delay in Indian networks, the precision and recall rates were 72 percent and 69 percent, respectively. As can be seen, the decision tree produced findings superior to those produced by the cluster classification method of categorization. The LR algorithm achieved 43 percent and 86 percent of accuracy, respectively. For

Received 4/October/2021; Accepted 24/December/2021

DOI: https://doi.org/10.33103/uot.ijccce.22.3.8

the ADA and SGD algorithms, the class precision ranged from 70 percent to 87%, and class recall decreased between 45 percent and 38 percent, respectively.

According to what researchers generally know, in ML, there are two crucial factors to consider: the time it takes to create the model and the quality of the ML algorithm that is being used to develop it. In this study, the accuracy was first evaluated as a metric for evaluating and analyzing the ML method. Then the second metric is considered: the time it takes to construct the model. Because computational complexity is now the most important and demanding challenge in ML, the time required to develop a model seems to be ML's most significant and critical problem. However, it is critical to keep track of the amount of time required to construct the model. As a result, when the time to construct the model is considered, the DT ML method is more effective than the other ML techniques. *Fig. 2* and *Fig. 3* show a more detailed comparison. However. According to the results of the experimental research, the DT algorithm is the most successful of the six ML algorithms tested for price prediction.

The results of the various methodologies were compared and confirmed based on characteristics such as class precision and recall. Tables II and III and *Fig. 2* and *Fig. 3* for the Indian flight networks, respectively, demonstrate the results of all the approaches used. Regarding the trustworthiness of outcomes, class recall is a crucial factor to consider. Our findings revealed that the proposed model was capable of predicting flight prices on the Indian network with an accuracy of 89%, which is a reasonable degree of accuracy.

This is useful for boosting the machine's performance. However, the overall performance of the ML algorithm used is quite efficient in terms of accuracy, precision, recall, and time required to build the model. Our proposed techniques yield promising results, but an in-depth study shows some insightful and important information that may be deployed to efficient ML choice and the effectiveness of ML algorithms. These insights and valuable information are provided below.

(1) The proposed model was found to be the most effective in picking an effective ML method from a set of ML algorithms with appropriate accuracy, precision, recall, and time is taken to create the model metrics.

(2) DT algorithm was chosen from a pool of six other ML algorithms, including Bayesian modeling techniques, stochastic gradient descent, adaptive boosting, decision trees, logistic regression, and K-nearest neighbor. It is noteworthy that the selected ML performance is extremely efficient for price prediction in-flight networks using ML techniques.

(3) As previously indicated, researchers selected only four different measures to evaluate the performance of ML algorithms for this study, including accuracy, precision, recall, and the amount of time it took to create a model. On the other hand, the model's accuracy and development time are the essential criteria. All the applied ML methods are efficient in terms of the selected ML metrics.

(4) Although all of the specified ML measures were shown to be successful for using a soft set, it is advantageous to apply additional parameters and ML techniques.

(5) For both selection and decision-making, the soft set method adopted in this study is beneficial. Researchers demonstrated that the DT technique is effective when using a soft set. However, it is critical to apply this method to other challenges that are relevant to selection and price prediction in flight network contexts.

Received 4/October/2021; Accepted 24/December/2021

DOI: https://doi.org/10.33103/uot.ijccce.22.3.8

TABLE 2. PERFORMANCE COMPARISON OF DIFFERENT CLASSIFIERS

| Classifiers | | Precision | Recall | F1-score |
|---|---|---|---|---|
| NB | weighted avg | 0.61 | 0.45 | 0.48 |
| SGD | weighted avg | 0.87 | 0.38 | 0.50 |
| ADA | weighted avg | 0.70 | 0.45 | 0.54 |
| DT | weighted avg | 0.89 | 0.89 | 0.89 |
| LR | weighted avg | 0.86 | 0.43 | 0.55 |
| KNN | weighted avg | 0.72 | 0.69 | 0.70 |

TABLE 3. APPLIED ML ALGORITHMS ACCURACY RESULTS

| Classifiers | Accuracy |
|---|---|
| NB | 0.45 |
| SGD | 0.38 |
| ADA | 0.45 |
| DT | 0.89 |
| LR | 0.43 |
| KNN | 0.69 |

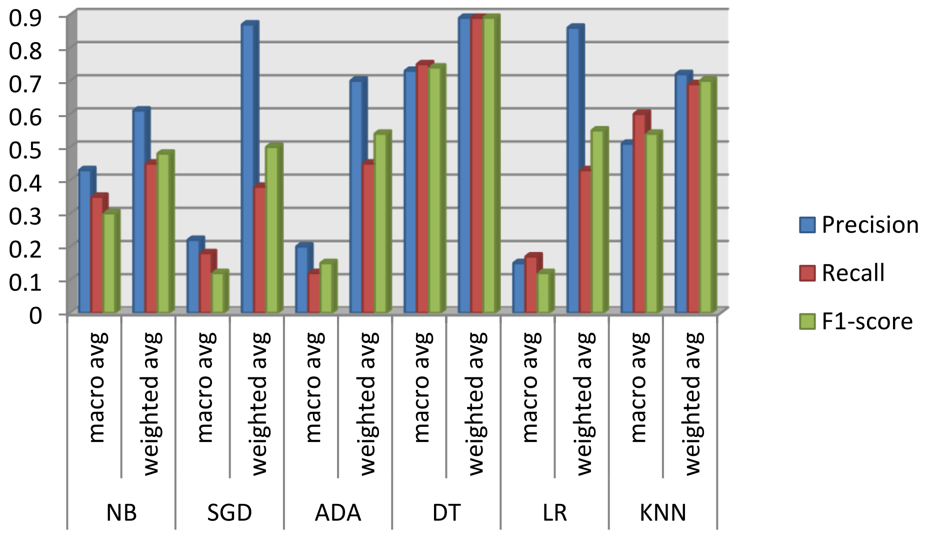


FIG. 2. PERFORMANCE EVALUATION OF ALL MODELS BASED ON CLASSIFIER EVALUATION MEASURE.

*Received 4/October/2021; Accepted 24/December/2021*

DOI: https://doi.org/10.33103/uot.ijccce.22.3.8

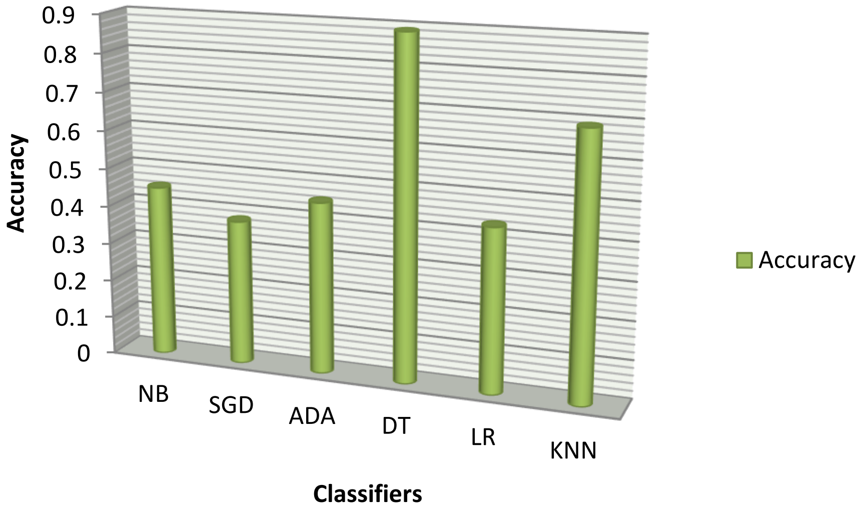


FIG. 3. ACCURACY RESULTS.

## VIII. CONCLUSIONS

In this study, six efficient ML approaches were selected for flight price prediction. These proposed algorithms are based on Bayesian modeling techniques, stochastic gradient descent, adaptive boosting, decision trees, logistic regression, and K-nearest neighbor. These techniques were tested on real-world datasets from Indian networks. The results showed that the DT technique outperformed the other methods. Flight prices in the Indian network are heavily influenced by factors such as the age of the aircraft and the type of aircraft. In the Indian networks, the decision tree method is the best conceivable Algorithm for predicting the price of a flight in a particular situation with 89% accuracy. The SGD method had the lowest accuracy, which was 38 %, while the accuracies of the KNN, NB, ADA, and LR algorithms were 69 %, 45 %, and 43 %, respectively. These findings may be important because they allow safeguards to be placed to prevent delay propagation.

According to reported results, the proposed ML algorithm model is effective at picking ML from a pool of ML algorithm numbers, and it is clear that the decision tree ML algorithm is effective at predicting. Researchers can use additional intriguing data mining technologies in future investigations and compare the results to determine which works best. The proposed combination approach of delay anticipation and its results can be investigated in greater depth in future studies. Consider combining the hybrid technique with robust flight scheduling as an example of an interesting research topic in the future.

## REFERENCES

[1] R. S. Alhamdani, M. N. Abdullah, and I. A. Sattar, "Recommender system for global terrorist database based on deep learning," Int. J. Mach. Learn. Comput., vol. 8, no. 6, 2018, doi: 10.18178/ijmlc.2018.8.6.747.

[2] I. A. S. Jabbar, R. S. Alhamdani, and M. N. Abdullah, "Analyzing restricted boltzmann machine neural network for building recommender systems," 2019, doi: 10.1109/IICETA47481.2019.9012981.

[3] S. A. H. Alazawi and M. N. Abdullah, "Online failure prediction model for open-source software system based on CNN," in Lecture Notes in Networks and Systems, 2021, vol. 243, doi: 10.1007/978-981-16-2094-2_70.

[4] Z. A. Mohammed, M. N. Abdullah, and I. H. Al-hussaini, "Predicting incident duration based on machine learning methods," 2020.

[5] K. Hamad, M. A. Khalil, and A. R. Alozi, "Predicting freeway incident duration using machine learning," Int. J. Intell. Transp. Syst. Res., vol. 18, no. 2, 2020, doi: 10.1007/s13177-019-00205-1.

[6] R. H. Ali, M. N. Abdullah, and B. F. Abed, "The identification and localization of speaker using fusion

*Received 4/October/2021; Accepted 24/December/2021*

DOI: https://doi.org/10.33103/uot.ijccce.22.3.8

techniques and machine learning techniques," Evol. Intell., 2021, doi: 10.1007/s12065-020-00560-z.

[7] W. H. Abdulsalam, R. S. Alhamdani, and M. N. Abdullah, "Emotion recognition system based on hybrid techniques," Int. J. Mach. Learn. Comput., vol. 9, no. 4, 2019, doi: 10.18178/ijmlc.2019.9.4.831.

[8] W. H. Abdulsalam, R. S. Alhamdani, and M. N. Abdullah, "Speech emotion recognition using minimum extracted features," in Proceedings - 2018 1st Annual International Conference on Information and Sciences, AiCIS 2018, 2019, pp. 58–61, doi: 10.1109/AiCIS.2018.00023.

[9] W. H. Abdulsalam, R. S. Alhamdani, and M. N. Abdullah, "Facial emotion recognition from videos using deep convolutional neural networks," Int. J. Mach. Learn. Comput., vol. 9, no. 1, pp. 14–19, 2019, doi: 10.18178/ijmlc.2019.9.1.759.

[10] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, "Explainable ai: A review of machine learning interpretability methods," Entropy, vol. 23, no. 1. 2021, doi: 10.3390/e23010018.

[11] G. Barash, E. Farchi, I. Jayaraman, O. Raz, R. Tzoref-Brill, and M. Zalmanovici, "Bridging the gap between ML solutions and their business requirements using feature interactions," in ESEC/FSE 2019 - Proceedings of the 2019 27th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Aug. 2019, pp. 1048–1058, doi: 10.1145/3338906.3340442.

[12] D. Marijan, A. Gotlieb, and M. Kumar Ahuja, "Challenges of testing machine learning based systems," Proc. - 2019 IEEE Int. Conf. Artif. Intell. Testing, AITest 2019, pp. 101–102, 2019, doi: 10.1109/AITest.2019.00010.

[13] S. Ray, "A Quick review of machine learning algorithms," in Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing: Trends, Prespectives and Prospects, COMITCon 2019, Feb. 2019, pp. 35–39, doi: 10.1109/COMITCon.2019.8862451.

[14] W. Groves and M. Gini, "A regression model for predicting optimal purchase timing for airline tickets," Dep. Comput. Sci. Eng. Univ. Minnesota - Tech. Rep., no. 1, pp. 1–17, 2011, [Online]. Available: https://pdfs.semanticscholar.org/be3a/231673885de15ab61cd2dd9a6b223940a49d.pdf.

[15] W. Groves and M. Gini, "An agent for optimizing airline ticket purchasing," in 12th International Conference on Autonomous Agents and Multiagent Systems 2013, AAMAS 2013, 2013, vol. 2, pp. 1341–1342.

[16] K. Tziridis, T. Kalampokas, G. A. Papakostas, and K. I. Diamantaras, "Airfare prices prediction using machine learning techniques," in 25th European Signal Processing Conference, EUSIPCO 2017, Oct. 2017, vol. 2017-Janua, pp. 1036–1039, doi: 10.23919/EUSIPCO.2017.8081365.

[17] V. H. Vu, Q. T. Minh, and P. H. Phung, "An airfare prediction model for developing markets," in International Conference on Information Networking, 2018, vol. 2018-January, doi: 10.1109/ICOIN.2018.8343221.

[18] E. Santana, S. Mastelini, and S. Jr., "Deep regressor stacking for air ticket prices prediction," pp. 25–31, 2019, doi: 10.5753/sbsi.2017.6022.

[19] J. M. Zhang, M. Harman, L. Ma, and Y. Liu, "Machine Learning Testing: survey, landscapes and horizons," IEEE Trans. Softw. Eng., pp. 1–1, Feb. 2020, doi: 10.1109/tse.2019.2962027.

[20] S. K. Singh, R. W. Taylor, M. M. Rahman, and B. Pradhan, "Developing robust arsenic awareness prediction models using machine learning algorithms," J. Environ. Manage., vol. 211, pp. 125–137, Apr. 2018, doi: 10.1016/j.jenvman.2018.01.044.

[21] P. Sokkhey and T. Okazaki, "Hybrid machine learning algorithms for predicting academic performance," Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 1, pp. 32–41, 2020, doi: 10.14569/ijacsa.2020.0110104.

[22] S. Uddin, A. Khan, M. E. Hossain, and M. A. Moni, "Comparing different supervised machine learning algorithms for disease prediction," BMC Med. Inform. Decis. Mak., vol. 19, no. 1, pp. 1–16, 2019, doi: 10.1186/s12911-019-1004-8.

[23] H. R. Pourghasemi, A. Gayen, S. Park, C.-W. Lee, and S. Lee, "Assessment of landslide-prone areas and their zonation using logistic regression, logitboost, and naïvebayes machine-learning algorithms," Sustain., 2018, doi: 10.3390/su10103697.

[24] G. Alimjan, T. Sun, Y. Liang, H. Jumahun, and Y. Guan, "A New technique for remote sensing image classification based on combinatorial algorithm of SVM and KNN," Int. J. Pattern Recognit. Artif. Intell., vol. 32, no. 7, Jul. 2018, doi: 10.1142/S0218001418590127.

[25] M. M. Hasan, M. T. Zahara, M. M. Sykot, A. U. Nur, M. Saifuzzaman, and R. Hafiz, "Ascertaining the fluctuation of rice price in bangladesh using machine learning approach," 2020, doi: 10.1109/ICCCNT49239.2020.9225468.

[26] A. Taherkhani, G. Cosma, and T. M. McGinnity, "AdaBoost-CNN: An adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning,"

Received 4/October/2021; Accepted 24/December/2021

DOI: https://doi.org/10.33103/uot.ijccce.22.3.8

Neurocomputing, vol. 404, pp. 351–366, Sep. 2020, doi: 10.1016/j.neucom.2020.03.064.

[27] A. Shahraki, M. Abbasi, and Ø. Haugen, "Engineering applications of artificial intelligence boosting algorithms for network intrusion detection : a comparative evaluation of real adaboost , gentle adaboost and modest adaboost," Eng. Appl. Artif. Intell., vol. 94, no. February, p. 103770, 2020, doi: 10.1016/j.engappai.2020.103770.

[28] Y. Zhang, A. M. Saxe, M. S. Advani, and A. A. Lee, "Energy–entropy competition and the effectiveness of stochastic gradient descent in machine learning," Mol. Phys., vol. 116, no. 21–22, pp. 3214–3223, Nov. 2018, doi: 10.1080/00268976.2018.1483535.

[29] A. Razaque and A. M. Alajlan, "Supervised machine learning model-based approach for performance prediction of students," J. Comput. Sci., vol. 16, no. 8, pp. 1150–1162, 2020, doi: 10.3844/jcssp.2020.1150.1162.

[30] A. Bibi et al., "Spam mail scanning using machine learning algorithm," J. Comput., vol. 73, no. 2, 2020, doi: 10.17706/jcp.15.2.73-84.

Received 4/October/2021; Accepted 24/December/2021