scientific reports

Check for updates

OPEN

Scalable flight cancellation prediction with ensemble distributed KNN and feature selection

Ho Yin Kan[1]✉, Keith Chau[2] & Patrick Cheong-iao Pang[3]

Flight cancellation prediction accuracy remains essential for airlines because it allows for automatic risk reduction of financial losses and passenger satisfaction decline. Heavy aviation big data presents challenges to traditional prediction methods which makes their practical use difficult. The proposed research brings forth an innovative approach utilizing distributed ensemble learning for conducting flight cancellation predictions at scale. The Artificial Bee Colony (ABC) algorithm operates within our method to determine the most essential predictors from an extensive dataset through optimal feature selection. The MapReduce framework enables distributed K-Nearest Neighbor (DKNN) model implementation to process features selected by the subsequent stage. The distribution of KNN models within this architecture allows the processing of extensive datasets effectively and delivers better accuracy through a collective model voting system. Our system performs computations on flight data collected from three New York City airports (JFK, LGA, and EWR) with a minimum computational advantage exceeding 25% above non-distributed KNN models. The ensemble strategy enhances prediction accuracy by 3.42% to obtain an average accuracy level of 95.79% which represents a 2.2% improvement above previous methods. Our distributed ensemble methodology proves its effectiveness for predicting flight cancellations accurately in big data environments through the presented experimental results.

Keywords Ensemble learning, Big data, Flight cancellation prediction, MapReduce, Distributed k nearest neighbors (DKNN)

Flight delays and cancellations are unpleasant events in the flight transportation system that may be caused by several factors such as adverse weather conditions, inefficiency in planning, plane breakdowns, and unexpected events[1]. The occurrence of these events in one flight may indirectly affect a chain of other flights[2]. On the other hand, flight cancellations and delays, in addition to imposing financial losses on airlines, cause passengers' dissatisfaction[3]. In such a situation, advance notice of flight cancellation is very important. In case of information about the possibility of flight cancellation, airline managers can prevent more losses by planning alternative strategies[4]. Also, having a flight cancellation prediction system can be useful in improving flight scheduling[5]. These advantages have caused many researchers to deal with the issue of flight cancellation or delay prediction based on flight data in recent years.

Most of the previous studies in this field have used machine learning (ML) techniques to solve the problem, and the goal of all of them was to achieve a more accurate prediction of flight delay cancellation. However, some specific challenges with flight data have made the applicability of these models in real conditions questionable. The first challenge in this field is the complexity of the problem of flight cancellation/delay prediction. Because this event can be caused by various direct (e.g., aircraft features) and indirect (e.g., delay in previous flights) factors[6]. This is despite the fact that most of the previous studies have used limited features for prediction and the absence of some indicators related to flight cancellation in these data causes the poor performance of learning models in problem modeling. To solve this deficiency, the set of features related to the flight cancellation/delay should be selected efficiently. On the other hand, specifications such as large volume and high production rate in flight data cause the correct processing of this information in real applications to require big data processing techniques[7]. This is while most of the previous methods did not consider this feature. Common ML methods

[1]Centre for Continuing Education, Macao Polytechnic University, Macao, China. [2]College of Professional and Continuing Education, The Hong Kong Polytechnic University, Hongkong, China. [3]Faculty of Applied Sciences, Macao Polytechnic University (MPU), Macao, China. ✉email: hykan@mpu.edu.mo

require the complete processing of training database samples to implement the training process. Due to the limitations of memory and computing power, it is not possible to perform this operation in big data processing applications. Therefore, the processing of flight big data requires the use of techniques that are compatible with the features of these data.

Therefore, this article is an attempt to solve the mentioned challenges by using the combination of ensemble learning techniques and distributed computing. In the proposed method, the limitations related to the accuracy of forecasting systems are reduced through the use of multiple learning models. Also, by using distributed computing techniques, an efficient method for flight big data processing is presented, which is effective in increasing the stability of the system in addition to improving the computing gain.

As with other approaches, this problem can be solved using different techniques; however, ensemble learning has some advantages. Firstly, it enhances the accuracy of the predictions by using the strengths of other models while at the same time avoiding the shortcomings of other models. Secondly, they combine the results, which helps to decrease the variance and to achieve stable results in definite practical tasks. Last, flight cancellation information is multifaceted; it includes a variety of aspects. When using different model types, each of them might be good at something, and this is a way of dealing with this kind of complexity. In the model that we are proposing in this paper, the ensemble uses three DKNN models with different settings. This increases model diversity by capturing different patterns in the data and also increases the model's ability to generalize and perform well on new scenarios. Incorporating ensemble learning, our model's objective is to attain a higher accuracy and stability than using a single model in the flight cancellation prediction. Although there are numerous previous works that have used popular distributed processing frameworks such as Apache Spark with a variety of models, our work in particular applies the ensemble of KNN models to a basic MapReduce framework, which offers a very scalable and fault-tolerant solution to this problem. The contributions of this article can be summarized as follows:

- This article presents a new DKNN model that will run in a MapReduce system. The given architecture is introduced as a powerful and efficient way of processing large aviation data, which shows improvements in the speed of computation and scalability over the traditional non-distributed models. Although other frameworks such as Apache Spark have also been applied in such a setting, our study aims at creating a custom KNN algorithm, which naturally fits the MapReduce paradigm and provides an effective and easy-to-use approach to big data processing.
- In this article, an ensemble system is presented to increase the accuracy of flight cancellation prediction. This ensemble system includes several DKNN models with different configurations, whose local detection results are integrated using the majority voting strategy. This mechanism makes it possible to cover the partial error of each model with other models.
- In this article, the problem of determining the most relevant indicators with the possibility of flight cancellation is discussed and a solution based on optimization techniques is presented to solve it. In this method, the importance of indicators is analyzed based on correlation criteria, and a set of the most effective indicators on flight cancellation is introduced. In the following, the effectiveness of this approach in increasing the accuracy of flight cancellation prediction is investigated.

The combination of ensemble and distributed learning systems to improve the accuracy and speed of big data processing is one of the innovative aspects of the proposed method. The structure of the continuation of the article is as follows. Section 2 includes a review of the research records, and in Sect. 3, a proposed solution for predicting flight cancellations is presented. Then, in Sect. 4, the results of this model implementation are reported and these findings are discussed. In Sect. 5, the research results are summarized.

Research background

In recent years, the issue of flight cancellation prediction has been investigated in various research. The most successful methods presented in this field have used ML techniques. Some studies have only focused on the prediction of flight cancellations; While some other studies have extended the application of their model to the prediction of flight cancellations and delays. In[8], a method for flight cancellation prediction based on ML techniques was introduced. In this method, first, 11 indicators are collected in the field of flight information and then four ML models are utilized to predict flight cancellation. These models include Support Vector Machine (SVM), Naive Bayes, Logistic Regression, and Decision Tree. The results of this research showed that the decision tree model can make predictions with higher accuracy than other models.

In[9], a method for predicting flight cancellations or delays based on the combination of feature selection techniques and ML was suggested. In this method, the feature selection process is performed using the recursive feature elimination strategy. The researchers by applying this algorithm to their dataset, reduced its dimensions to 14 features and used three models: LightGBM, Random Forest (RF), and Multilayer Perceptron (MLP) neural network to predict flight cancellations or delays. The reported results showed the superiority of the LightGBM model over the other two models. In similar research[10], the use of ML techniques to predict flight cancellations or delays was studied. This research did not introduce a strategy to reduce data dimensions, and for this reason, the accuracy values reported in this research cannot be compared with other studies.

In[11], a method based on deep learning was introduced to predict flight cancellations during the Covid-19 pandemic. In this method, 19 indicators were used to predict flight cancellation. The deep learning model in this article was obtained from the combination of two models, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), which are combined in the form of an ensemble prediction model. Although this model has high accuracy, its complexity makes its use for big data processing questionable.

In[12], the problem of Covid-19 impact on the cancellation of flights was investigated. In this research, one of the important issues of flight data related to flight cancellation, the imbalance of the data was mentioned. Because the number of canceled flights is less than 2% of the total number of flights. To solve this problem, different sampling techniques were used and the efficiency of each technique was evaluated through various ML methods. Based on the results of this research, the combination of random sampling with an MLP neural network can lead to higher prediction accuracy. In[13], the problem of data imbalance in the prediction of flight cancellation was discussed similarly. In this method, the sampling strategy based on the equilibrium rate is utilized, in which the appropriate equilibrium rate between the samples of both target classes is determined by an approach based on iteration. Then RF and MLP models are employed to predict flight cancellation or delay, and based on the results, the MLP model has a more suitable performance.

In[14], a method based on deep learning techniques to predict flight delay or cancellation was proposed. This research described the features of each flight in the form of 10 features and used three RF, LSTM, and Deep Recurrent Neural Network (DRNN) models for prediction. The results show the superiority of the RF model in terms of overall accuracy compared to deep learning models. On the other hand, the recall criterion maximizes success rate while using LSTM for prediction. In[15], a decision support system for determining flight cancellation costs is presented. By analyzing the flight and airport information through a logit model, this model provides the appropriate decision in the field of flight cancellation or providing an alternative flight. This model aims to minimize the losses caused by flight cancellations in airlines.

The research conducted in[16] provided a customized recommender system in order to provide alternative suggestions for canceled flights. In this model, customer and flight information is processed using an XGBoost model to provide alternative recommendations based on it. This model tries to maintain customer loyalty to airline companies by analyzing customer characteristics and habits. In[17], a decision support system for flight cancellation recommendations was presented. In this system, weather information and flight features are processed by an RF model to evaluate flight cancellation conditions. One of the prominent features of this model is to consider the factors of forecast uncertainty and possible damage due to flight cancellation at the same time.

In[18], a method for predicting flight delay using ML models was introduced. This method performs forecasting based on atmospheric and flight features. In this research, the performance of models such as logistic regression, decision tree, RF, and Gradient Boosting Regression (GBR) were compared in predicting the delay, research results showed that the RF model can achieve higher accuracy. In[19], a model based on the combination of deep learning and ML techniques was proposed to predict flight delay. In this model, the combination of convolutional neural network and LSTM was utilized in order to extract flight temporal and spatial correlation features. The correlation features extracted by these two models were classified by an RF model to determine the presence of flight delays.

A graph convolutional network (GCN)-based model for flight delay forecasting was introduced in[20]. This model successfully learned the associations between the temporal and spatial information by expressing them separately. A variety of deep learning methods were applied in[21] to forecast airline delays. CondenseNet, a CNN-based model that was originally used in this study, had its prediction accuracy increased by adding CBAM components. Furthermore, a CNN-LSTM hybrid model was presented for delay prediction. Through incorporating SimAM components, this hybrid model demonstrated a noteworthy enhancement over the separate CNN, LSTM, and CondenseNet models, achieving an estimation accuracy of 91.36%.

Deep learning methods were applied in[22] to anticipate aircraft delays. This method reduced the dimensionality of the data by preprocessing a set of flight- and weather-related features and using the ECA-MobileNetV3 model. Lastly, a SoftMax layer was used to categorize the features that were extracted.

In comparison to GCN independently the research done in[23] suggested a GCN-based approach to flight delay detection termed Geographical and Operational GCN (GOGCN), which showed superior capabilities in expressing geographical attributes and spatio-temporal variables. In this method, two GCN systems analyzed local geographical characteristics and global operational characteristics independently. After that, an output layer was used to combine the features that had been obtained from both models in order to estimate flight delays.

A random forest-based probabilistic model was utilized in[24] to assess the influence of each element on the precision of predictions. Weather-related and weather-unrelated variables were separated into two distinct groups. Finally, according to the chosen criteria, a random forest was employed to detect delays.

Flight delay forecasting was done in[25] using a model of random forests based on cluster computing techniques. The primary goal of this study was to use methods for processing big aviation data to speed up the analysis of flight records; nevertheless, a notable gain in prediction accuracy was not realized. Nonetheless, the random forest model demonstrated 92.7% accuracy in forecasting delays in flights, suggesting that this classification model is compatible with the flight delay prediction.

Research[26] presented an estimation model based on ensemble learning to predict the exact value of flight delay. This model, evaluated various ensemble learning techniques such as bagging, stachikng, and boosting and compared their performance in prediction of flight delays with classical ML algorithms. The results demonstrate the superiority of the ensemble learning techniques over other algorithms. However, this research can not provide a solution for handling big aviation data. Research[27], evaluated the efficiency of four ML algorithms including RF, logistic regression, gradient boosting, and decision tree in predicting flight cancellation. The reported results show that in this research problem decision trees can overcome the other classification algorithms.

Our work is dependent on recent progress in feature selection and dimensionality reduction. Sheikhpour et al.[28] discuss the problem of high-dimensional multi-label data by suggesting a semi-supervised feature selection algorithm. The relevance of their work is high since it, similarly to our approach, attempts to find the most informative features to minimize computational cost and enhance performance.

Likewise, a survey on Nonnegative Matrix Factorization (NMF) in dimensionality reduction is given by Saberi-Movahed et al.[29]. Their study of NMF in feature extraction and selection gives a wide background to our own feature selection method based on ABC which also aims at optimizing the feature space to achieve a more efficient and accurate model.

Lastly, the survey on spectral clustering by Berahmand et al.[30] highlights the significance of learning structure of underlying data using graph structure learning (GSL). Our correlation-based feature selection is based on this principle since we also aim to know the relationships between features in order to find a non-redundant and highly relevant subset. Table 1 summarizes the reviewed studies.

The problem of large aviation data has prompted some scholars to consider distributed computing systems. As an example[8], used a classification method based on Spark to forecast flight cancellations. Their work was able to show the importance of utilizing a distributed framework to process large datasets and reach a high level of accuracy using a decision tree model. In a similar manner[25], employed cluster computing in order to speed up a random forest model to predict flight delays. Nevertheless, these studies, though significant, either failed to concentrate on an ensemble approach or failed to show a significant increase in the accuracy of prediction using their distributed techniques. Frameworks such as Spark are typical to use due to their in-memory processing capabilities, but can often have a high architectural overhead. The difference is that our study suggests a customized DKNN algorithm to be used with MapReduce model, which is one of the fundamental frameworks that offers an alternative set of benefits, including architectural simplicity and increased fault tolerance in the context of batch processing.

Research method

Accurate prediction of flight cancellations and delays should be based on a complete set of related indicators so that patterns hidden in flight data can be mined more completely. On the other hand, the forecasting process should be based on a model compatible with the features of flight big data so that the correct functioning of the model can be assured in real conditions. In this section, a proposed method to meet these requirements is presented. For this purpose, first, the method of data collection in this research is explained and then the calculation steps of the proposed method are presented.

Reference	Year	Research Goal	Method	Limitation
Yanying et al.[8]	2019	Flight cancellation prediction	ML with Spark (SVM, Naive Bayes, Logistic Regression, Decision Tree)	Limited features (11), Not an ensemble model
Lambelho et al.[9]	2020	Flight cancellation/delay prediction	Feature selection with ML (LightGBM, Random Forest, MLP)	Imbalanced data
Shu[10]	2021	Flight cancellation/delay prediction	Machine Learning	No data dimension reduction, limits comparison
Bandyopadhyay et al.[11]	2020	Flight cancellation prediction during COVID-19	Deep Learning (LSTM, GRU)	High complexity for big data processing
Mohammed et al.[12]	2021	Impact of COVID-19 on flight cancellation	Machine Learning (MLP) to address data imbalance	Focuses on COVID-19 impact
Hendrickx et al.[13]	2021	Flight delay/cancellation prediction with imbalanced data	Sampling techniques with ML (RF, MLP)	Focuses on data imbalance
Ayaydin & Akcayol[14]	2021	Flight cancellation/delay/orientation prediction	Deep Learning (RF, LSTM, DRNN)	Raw flight data not suitable for deep learning without feature engineering
Diao et al.[15]	2019	Decision support system for flight cancellation costs	Logit model for flight information analysis	Focuses on cost minimization, not prediction accuracy
Gong et al.[16]	2023	Personalized recommendation system for canceled flights	XGBoost model for customer and flight information processing	Focuses on recommendations, not prediction
Taylor et al.[17]	2021	Decision support system for flight cancellation recommendations	Random Forest model for weather and flight data processing	Focuses on weather uncertainty and cancellation damage
Imran et al.[18]	2023	Flight delay prediction	Machine Learning (Logistic Regression, Decision Tree, RF, GBR)	Focuses on atmospheric and flight features
Li et al.[19]	2023	Flight delay prediction	Deep Learning (CNN-LSTM) with Random Forest classification	Complex model structure
Wu et al.[20]	2023	Flight delay forecasting	Deep Learning (Graph Convolutional Network)	Limited exploration of other deep learning methods
Qu et al.[21]	2023	Flight delay prediction	Deep Learning (CondenseNet, CNN-LSTM)	Limited exploration of other deep learning methods
Qu et al.[22]	2023	Flight delay prediction	Deep Learning (ECA-MobileNetV3)	Limited exploration of other deep learning methods
Cai et al.[23]	2023	Flight delay prediction	Deep Learning (Geographical and Operational GCN)	Lacks comparison with other GCN approaches
Li et al.[24]	2023	Flight delay prediction with weather priority	Random Forest with weather/non-weather feature separation	Limited exploration of other ML methods
Paramita et al.[25]	2022	Flight delay prediction	Random Forest with cluster computing	Limited improvement in prediction accuracy over other RF models
Wang et al.[26]	2022	Flight delay prediction	Ensemble learning	Low computational efficiency on big data

Table 1. Summary of the reviewed studies.

Data collection

In order to collect a complete set of possible features related to flight cancellation and delay conditions, research data has been collected through the FlightRadar24 website. In the process of data collection, a web crawler was employed to extract the incoming flight information of the three airports JFK, LGA, and EWR in New York City, United States during the years 2021 to 2023. By running this crawler, 816,096 data records were collected in the field of incoming flights of these airports. Each data record has been described through 36 indicators, which fall into three general classes: atmospheric features (7 indicators), flight features (19 indicators), and airport features (10 indicators). The collected samples have been divided into four classes based on the flight cancellation/delay status: 1- OnTime, 2- Late, 3- Very Late, and 4- Cancelled.

Predictive modeling of flight data poses a severe problem because of extremely skewed classes. In response to this, we used a hybrid data balancing approach. We have discovered that the traditional simple random sampling method was not sufficient enough because it could have lost valuable information and did not enrich the minority classes well. Our new two-step method is the following:

1. Oversampling of Minority Classes: We oversampled our minority classes, "Very Late" and "Cancelled" with the Synthetic Minority Over-sampling Technique (SMOTE) with a k-value of 5. This technique creates artificial data points of these classes which gives the model more data to learn.
2. Under-sampling of Majority Class: After the oversampling, we did a controlled random under-sampling of the majority classes, on-time and slightly late. This made them few in numbers to an extent that was equal to the newly oversampled minority classes.

This combination method makes the training data more balanced and at the same time retains the important characteristics of the original data. The last and balanced data set which was used to train consists of 162,750 samples where 60,391 samples were of OnTime flights, 48,000 samples were of Late flights, 26,058 samples were of Very Late flights and 28,301 samples were of Canceled flights. The approach gives a more robust and unbiased model that can predict the four classes more accurately. The set of features in each record of this data set is given in Table 2.

The set of indicators listed in Table 2 is used as input for the proposed forecasting model.

The proposed method for flight cancellation and delay prediction

The proposed method in this article is designed based on the combination of ensemble learning techniques and distributed computing. Therefore, first, it is necessary to describe the architecture of the proposed system for

Category	ID	Title	Description	Data Type
Weather Conditions	I_1	Temperature	Air temperature in the station near the origin airport (C)	Continuous
	I_2	Humidity	Air humidity in the station near the origin airport (%)	Continuous
	I_3	Cloud density	Cloud density at the origin airport	Categorical
	I_4	Wind direction	Average wind direction at the station near the origin airport	Categorical
	I_5	Horizontal view	Horizontal view of the station near the origin airport (m)	Continuous
	I_6	Wind speed	Average wind speed at the station near the origin airport (km/h)	Continuous
	I_7	Atmospheric pressure	Air pressure at the station near the origin airport (bar)	Continuous
Flight status	I_8	Day	The number of days of the month	Categorical
	I_9	Month	The number of months that have passed in the year	Categorical
	I_{10}	Weekday	The number of days that have passed in the week	Categorical
	I_{11}	Departure time	Scheduled time of departure from origin	Continuous
	I_{12}	Landing time	Scheduled time of landing at the destination	Continuous
	I_{13}	Flight Duration	Estimated time from departure to landing (min)	Continuous
	I_{14}	Path length	Flight distance between origin and destination airports (km)	Continuous
	I_{15}	Flight class	Airline flight class type	Categorical
	I_{16}	Flight type	National flight (1) or international flight (2)	Categorical
	I_{17}	Airplane	Aircraft Type	Categorical
	I_{18}	Number of passengers	Total number of adult passengers on the flight	Continuous
	I_{19}	Previous flight	Having a previous flight (1) otherwise (2)	Categorical
	$I_{20} \sim I_{26}$	Airline flight cancellation and delay History	Canceled (-1), Delayed (1+), or Ontime flight (0) in the last seven airline flights as a vector	Continuous
Airport status	I_{27}	Origin congestion degree	The number of scheduled flights at the time of departure from the origin airport	Continuous
	I_{28}	Destination congestion degree	The number of scheduled flights at the time of landing at the destination	Continuous
	$I_{29} \sim I_{35}$	Airport flight cancellation and delay history	Canceled (-1), Delayed (1+), or Ontime flight (0) in the last seven inbound flights of the airport in a vector form	Continuous
	I_{36}	Origin delay rate	Rate of delayed flights at the origin airport during the last 7 days	Continuous

Table 2. The set of features extracted for each database sample.

processing flight data and predicting flight cancellation or delay based on it. This structure is shown in Fig. 1. According to the system model drawn in this figure, the input data is collected through a fusion component. This component is responsible for performing high-level processes including pre-processing, feature selection, and results fusion. The fusion component sends the preprocessed flight data to the three classification components. Each classification component uses a DKNN model with a different configuration for pattern learning. In each of these components, the received data is divided into parts and each data part is analyzed by a separate process based on the MapReduce model. In this model, each of the classification components, based on its trained DKNN model, predicts flight cancellations or delays in new samples and finally sends its prediction results to the fusion component. Finally, the prediction result of the system is determined by combining the results and voting between the output of the detection components. In the following, the details of each step of the proposed approach are explained.

The proposed model in this research predicts flight cancellations and delays through three main steps:

1. Preprocessing.
2. Feature Selection.
3. Feature Classification.

As mentioned, according to the proposed system model (Fig. 1), the first and second steps are done by the fusion component. While the third step of the proposed method is done by using the MapReduce computing model and with the cooperation of detection components. This mechanism is shown in the form of a flowchart in Fig. 2.

The purpose of the pre-processing step is to prepare the data samples for processing in the next steps. For this purpose, in this step, the records with missing values are modified and the input features are normalized. In the second step, by ABC, the most relevant data features with the probability of flight cancellation or delay are selected. The purpose of this step is to increase the processing speed of the proposed method (due to the reduction of data dimensions) and increase its detection accuracy (due to the elimination of irrelevant features). Finally, in the third step of the proposed method, the selected features are classified using an ensemble model consisting of three DKNN distributed classifiers. This process starts with data partitioning. For this purpose, each DKNN model assumes the presence of N systems (or processors) in the distributed model, the training information is divided into N parts with common data, and each part is sent to a classification component. Each detection system, after receiving its data, uses the KNN algorithm to calculate the distance between test samples and training data locally. This process is repeated until the termination condition of the local classification model is met. Then, the local detection of DKNN models for the test samples is performed based on the majority voting in the detection components. After determining the local detection result of each DKNN model, these results are combined by the fusion component to determine the prediction result of the proposed ensemble model through majority voting between the output of these models.

Data preprocessing

The initial phase in the suggested procedure is data pretreatment, which gets the record set ready to be processed in the next stages. This step consists of two steps: normalizing and missing values management. First, 0 values are used to fill in the values that are missing in categorical characteristics. Additionally, the mean values of continuous characteristics are used to fill in the missing values. It should be noted that if more than 30% of the

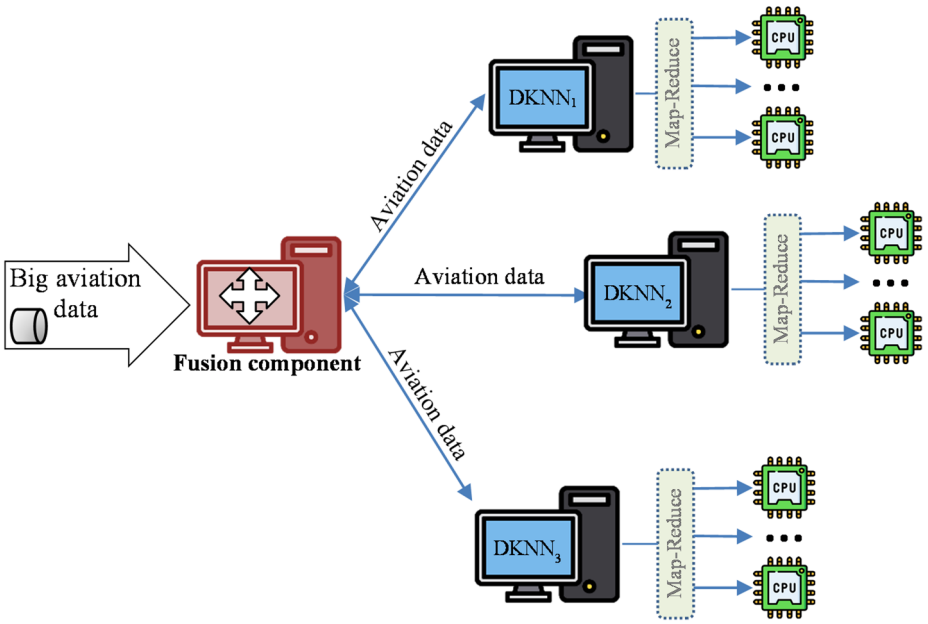

Fig. 1. The proposed system model based on ensemble learning and distributed computing model.

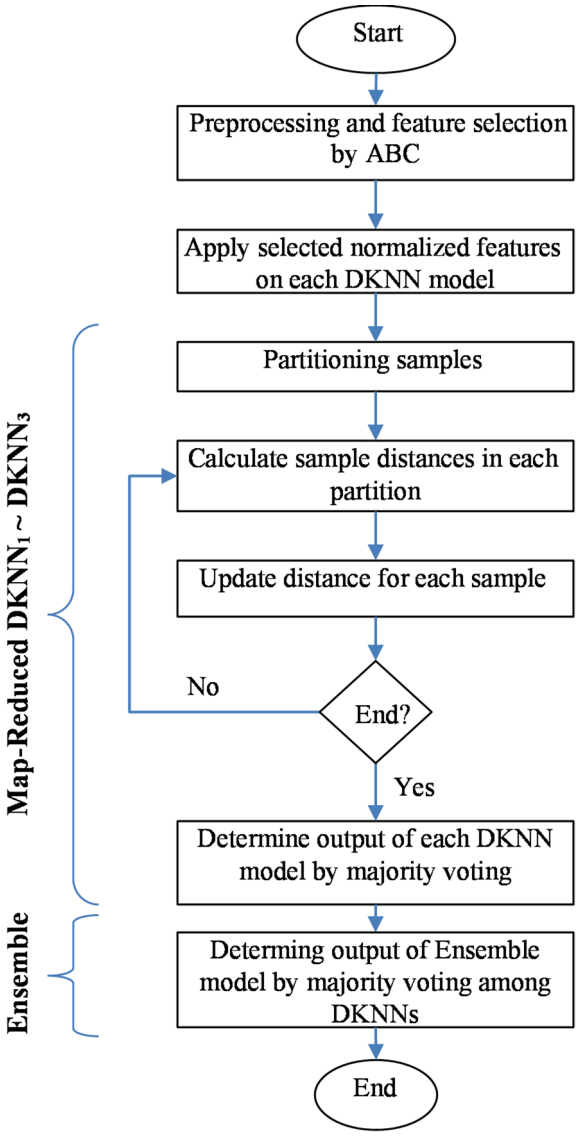

Fig. 2. Prediction steps of the proposed method.

attributes of a record contain missing values, that record is ignored. After managing the missing values, all the features are mapped to the range [0,1] based on Eq. (1)[31]:

$$\overrightarrow{N_i} = \frac{\overrightarrow{x} - \min(\overrightarrow{x})}{\max(\overrightarrow{x}) - \min(\overrightarrow{x})}, \tag{1}$$

Where $\overrightarrow{x}$ represents the input feature vector and $\overrightarrow{N_i}$ represents the corresponding normalization vector. Also, *min(.)* and *max(.)* are the minimum and maximum functions for the feature vector, respectively.

Feature selection by ABC

After normalizing the database features, feature selection and data dimension reduction are performed. Our feature selection algorithm is meant to maximize predictive accuracy and computational efficiency of our model by selecting the most relevant and non-redundant features. In order to do this, we selected the Artificial Bee Colony (ABC) algorithm because it has its own advantages in terms of solving complex optimization problems. ABC is especially qualified to do this due to its better balance of exploration and exploitation in the search process. It enables it to effectively search a high dimensional feature space and find an optimal subset without getting trapped in local optima, which is a common problem of other optimization-based feature selection algorithms. In the following, the structure of the solution vector and the objectives defined for the optimization algorithm are described first, and then the feature selection steps using ABC are explained.

In the proposed method, the number of optimization variables is equal to the number of features in the database (Table 2), which is represented by F. In other words, the solution vector length in the optimization

algorithm is equal to F. The ABC algorithm should be able to determine the selection or removal of a feature through the solution vector. In this way, each solution vector can be considered as a binary string in which each existing feature is assigned a place in the solution vector of the optimization algorithm. Each place can have a value of 0 or 1. If a location has a value of 0, that feature is not selected in the current solution, and otherwise, the feature corresponding to the current location is considered as the selected feature.

An optimal feature subset selection is a decisive procedure in the improvement of model accuracy and efficiency in computation. The Minimum Redundancy Maximum Relevance (mRMR) principle has been used to guide the feature selection process we have adopted and this is a well established principle. This principle assumes that the best subset of features is the one in which features are very relevant to the target variable and as little redundant as possible with each other. To this end we formulate a two-objective fitness function of the ABC algorithm, that attempts to meet the following two opposing objectives:

1. Maximizing Relevance (F_1): The former is to maximize the mean correlation between the chosen features and the target variable. The larger the correlation, the more significant a feature and the more predictive it is in terms of flight cancellation or delays. This may be represented as the objective function as follows:

$$F_1 = \frac{1}{|S|} \sum_{\forall\, i \in\, S} corr\,(i, T) \tag{2}$$

Where S represents the set of features selected in the current solution and |S| shows the number of these features. Also, T describes the target variable and $corr\,(i, T)$ is the correlation evaluation function between the selected feature i and the target variable.

2. Minimizing Redundancy (Minimum Redundancy, F2): The second objective is to reduce the repetition of the selected features. Features that are highly correlated tend to have similar information and adding all of them to the model will add noise and computation time without a corresponding increase in the predictive accuracy. The following is the minimization objective function:

$$F_2 = \frac{1}{|S|^2} \sum_{\forall\, i \in\, S} \sum_{\forall\, j \in\, S,\, (j \neq\, i)} corr\,(i, j) \tag{3}$$

Since the above two objectives are incompatible with each other (the first objective is defined as a maximization and the second objective is defined as a minimization); Therefore, in order to make these two objectives compatible in the optimization algorithm, they are combined by the following fitness relationship:

$$fitness = \frac{F_2}{F_1 + 1} \tag{4}$$

The 1 added to the denominator (F1 + 1) makes the division always well defined and avoids the possibility of a zero division since the correlation values may be negative. Optimization of this fitness function is a decent approach to identify a set of features that is as relevant as possible and as redundant as little as possible, therefore, giving a theoretically well-founded foundation to our feature selection procedure. The proposed algorithm to select the most relevant features with the probability of flight cancellation and delay by ABC includes the following steps:

Step 1) Determining the bounds of the problem variables: In this step, the search bounds for each optimization variable are determined as $r_i \in \{0,1\}$.

Step 2) Generation of the initial population: In this step, the scout bee agents are used to generate the initial solution vector S_N. Each solution vector is organized as a random binary string with length F. Then the fitness of each solution vector is evaluated based on Eq. (4).

Step 3) Employed bee search: In this step, employed bee agents try to discover more suitable solutions by searching the neighborhood points of each existing solution vector. For this purpose, in each solution vector, one of the variables is randomly selected and the bit value corresponding to that bit is reversed. After doing this, the vector fitness of the new solution is calculated and compared with the previous solution. If the fitness of the new solution is less than the previous one, the new solution is replaced and otherwise, it is discarded.

Step 4) Evaluation by onlooker bees: The solution vectors obtained from the previous step are checked by onlooker bee agents. In this phase, first, the probability of choosing each solution vector such as $\overrightarrow{x}_m$ is calculated as follows[32]:

$$P_m = \frac{fitness\,(\overrightarrow{x}_m)}{\sum_{i=1}^{SN} fitness\,(\overrightarrow{x}_i)} \tag{5}$$

Then, the roulette wheel selection method was employed to select the solutions of the next cycle based on the probability values (Eq. (5)). This algorithm tries to select solutions with a larger likelihood to be utilized in the next cycle, but in order to keep the comprehensiveness of the search, a small portion of non-optimal solutions are also moved to the next cycle.

Step 5) Searching by scout bees: in this step, solutions that their fitness could not improve after s cycles are replaced by new random solutions. This process prevents the solution from getting trapped in a local optimum.

Step 6) Evaluating the results: If the number of cycles of the algorithm reaches the threshold T, the algorithm is terminated. Otherwise, the algorithm repeats the new cycle from step 2.

The result of the above steps is a subset of the input features based on which the prediction accuracy can be maximized. After choosing the optimal features, in the last phase of the introduced approach, an ensemble system based on DKNN classifiers is utilized to predict flight cancellations or delays.

Prediction based on the combination of DKNN classifiers

One of the differences between the proposed model and previous studies for predicting flight cancellations and delays is the use of an ensemble of DKNN model in the detection step. Using an ensemble strategy can be effective in reducing the error of learning models[33]. On the other hand, using the distributed computing strategy in DKNN classification models can have two advantages. The first advantage is the increase in computing efficiency and the ability to process flight big data. Because this model can be used in situations where there is not enough memory to store and process all the data. The MapReduce mechanism in the DKNN model significantly increases the computational efficiency of the detection system by dividing the data and sending each part to an independent processing component. The second advantage of DKNN is improving fault tolerance. In other words, in case of deficiency in any of the forecasting components of the proposed distributed model, unlike centralized architectures, the system's overall efficiency will not decrease. The whole implementation that included the MapReduce framework was done on MATLAB 2020a and run on a cluster with 5 nodes and 16 GB RAM and 4 CPU cores. The suggested ensemble model is comprised of three DKNN models, each one of them being set with a different value of the number of neighbors (k). The particular odd numbers of k (3, 7, and 11) were chosen following an initial empirical study (grid search) in order to offer a trade-off between local pattern capture and ensemble diversity. We tend to choose small, odd values of k so that we can have no ties in the majority voting, and we can represent the local structure of the data well at many different resolutions. This combination in particular was empirically determined to provide the best trade-off between individual classifier performance and the stability of the ensemble as a whole on our data. Each of these models is formed independently based on training samples, and in the prediction phase, the majority voting strategy is used to determine the final output of the system. Since the number of target classes in the research problem is more than 2, if there are equal votes for some target classes, the output of the ensemble system is determined based on the DKNN model with the lowest training error. In the rest of this section, the DKNN algorithm is explained based on the MapReduce model, which is designed based on the mechanism introduced in[34]. In general, a MapReduce model performs computations in two main phases:

1. Map.
2. Reduce.

In the Map phase, classes and distances related to K nearest neighbors are calculated for each test sample in different divisions of the training data. Then, in the Reduce phase, K nearest neighbors' distances from each Map model are processed and by selecting those with the minimum distance, a definitive list of K nearest neighbors is created. Then, the majority voting method is used to determine the class in which the sample is placed. In the following, Map and Reduce phases in the DKNN model are explained. Figure 3 shows a high-level diagram of the DKNN algorithm in the proposed method.

Map phase In Map phase, the training set (TR) is divided into m partitions where m = number of mappers in our distributed system. It splits the data in a trivial block-wise fashion with all the mappers getting a contiguous block of the training data. This plan will make the workload evenly distributed among all mappers. These data are then partitioned into subsets and each mapper works on its own data. In this way, the training data set is divided into m parts, $(map_1, \ldots, map_m)$. In addition, TS test samples are not partitioned because each test sample must be accessible in all mappers.

In this way, in each mapper, the distance of each test sample, such as x_i, is calculated from all training samples map_j. In this phase, the class label corresponding to the nearest k neighbors (with minimum distance) for each test sample and their distance are saved as output. Thus, the output of this phase for each detection component is a matrix like CD_j with $n \times k$ dimensions, where n represents the number of test samples and k represents the number of neighbors in the KNN algorithm. Each element of the CD_j matrix contains distance information and labels of nearest neighbors in the form of $< class, dist >$. It should also be noted that each row of this matrix is sorted in ascending order based on the dist distance values. The steps of the Map phase are as follows:

Input: training subset map_j, test set TS, and parameter K (number of neighbors)

1. Initialize an empty list of neighbors for each test sample: $Output_{List}$ = [ ].
2. **For** each test sample x_i in TS:
3. Initialize a list for the **K** nearest neighbors: $Nearest_{Neighbors}$ = [ ].
4. **For** each training sample x_j in map_j:
5. Calculate distance between x_i and x_j using Euclidean distance.
6. **If** $Nearest_{Neighbors}$ has less than **K** members **OR** distance is less than the maximum distance in $Nearest_{Neighbors}$:
7. Add $(distance, class_{label})$ of x_j to $Nearest_{Neighbors}$.
8. If $Nearest_{Neighbors}$ has more than **K** members, remove the one with the maximum distance.
9. Sort $Nearest_{Neighbors}$ by distance.
10. Append $< x_i.id, Nearest_{Neighbors} >$ to CD_j.
11. **Return** CD_j.

Algorithm 1. Map phase.

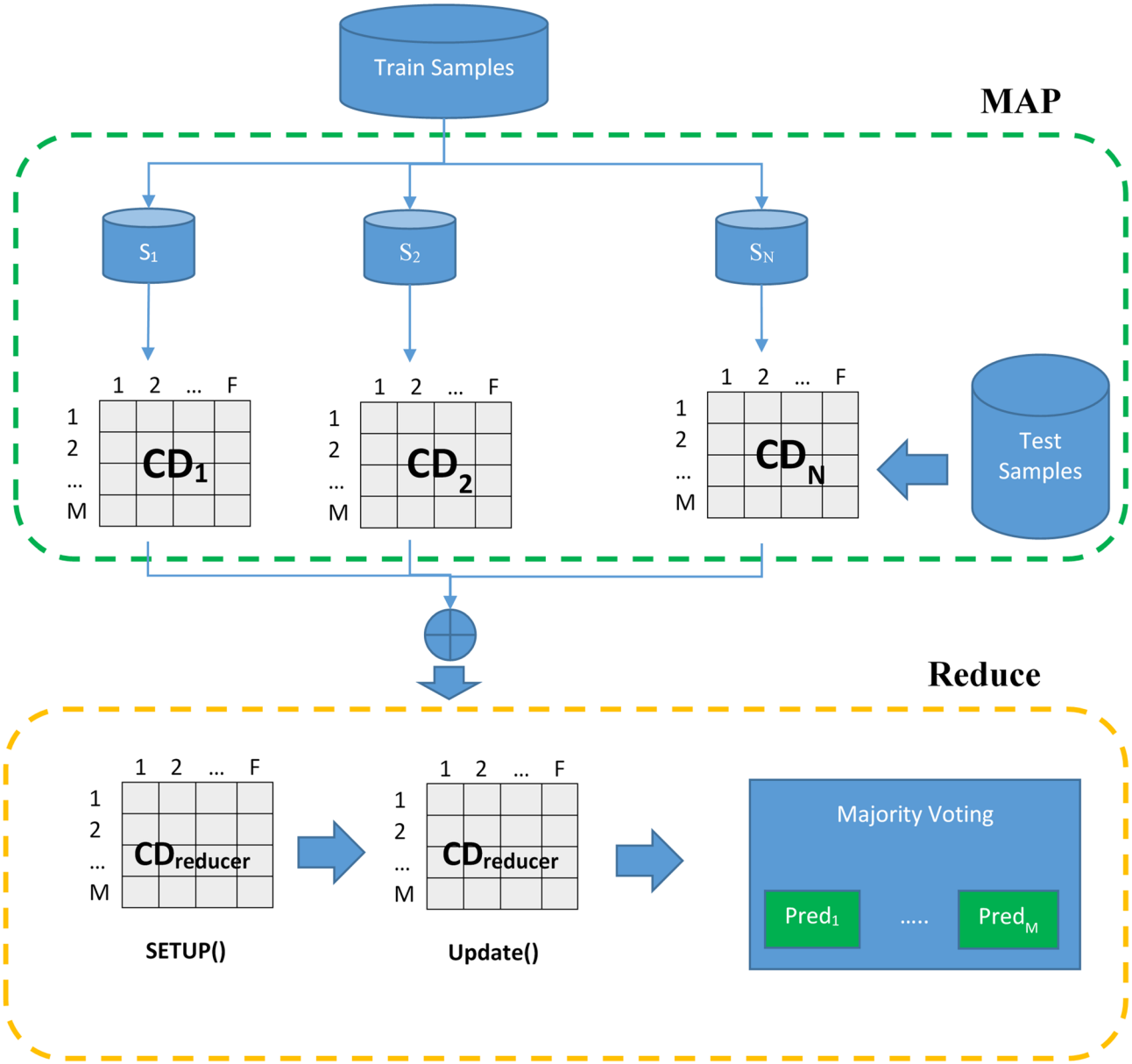

Fig. 3. Diagram of each DKNN model in the proposed ensemble system.

The result of the above steps is the CD_j matrix and the key ID, which are used in the Reduce phase.

Reduce phase In the Reduce phase, the results from all mappers are aggregated and analyzed to determine the final class label for each test sample. This phase begins with the configuration of the distributed system, which in our case is handled by the MATLAB environment. This phase begins with the configuration of the distributed attack detection system. The configuration step in the Reduce phase is shown as setup() in Fig. 3. In this phase, a matrix like $CD_{reducer}$ with n×k dimensions is defined. The initial value of the distance in the $CD_{reducer}$ matrix is considered equal to +∞. It should be noted that the configuration of the DKNN distributed system and the formation of the CDreducer matrix are performed by the detection component after receiving the first CD matrix (the result of the component Map phase).

When the Map phase is finished for each detection component, the *CDreducer* distance matrix is updated by comparing its current distance values with the Map result matrix (i.e., CD_j). Since the *CD* matrices obtained from the mapping are sorted by distance, the updating process can be done faster. Therefore, for each test sample such as x_{test}, we compare each neighborhood distance value peer-to-peer and start this work from the nearest neighbor. If the compared distance is less than the current value, then the class and the distance corresponding to this position in the matrix are updated with the new values. The *CDreducer* update process is executed by receiving the result of each mapper (by receiving each CD matrix). Therefore, it can be considered as an iterative step that fuses the results of the Map phase.

After processing all the results of the mappers and updating the *CDreducer* matrix for all the received CD parts; The majority voting method is utilized to determine the final label of the test samples. The voting step is the last step in the DKNN local detection process and is performed only once for each test set. At this step, the rows of the *CDreducer* matrix (which contains the definitive list of nearest neighbors for each TS test sample as $< class, dist >$) are considered as the basis of voting. In this step, each row of the *CDreducer* matrix (containing the information of the nearest neighbor for each TS test sample) is processed independently and the number of class labels in each row is counted. In this way, each test sample such as i belongs to the class whose label has the highest frequency in the i-th row of the *CDreducer* matrix. It should be noted that if more than one tag has the highest frequency, the tag with the lowest average distance is considered as the output. The result of this process is the predicted classes for all test samples in the TS set. The steps for implementing the Reduce phase are as follows:

Input: The outputs from all mappers: a collection of key-value pairs <test sample ID, list of K neighbors>

1. Group the mapper outputs by test sample ID.
2. **For** each test sample ID:
3. Initialize an empty master list: $Global_{Neighbors}$ = [ ].
4. **For** each list of neighbors received from a mapper:
5. Merge the list with $Global_{Neighbors}$.
6. Sort $Global_{Neighbors}$ by distance in ascending order.
7. Select the top **K** neighbors from $Global_{Neighbors}$.
8. Count the frequency of each class label among these top **K** neighbors.
9. Assign the class label with the highest frequency as the final prediction.
10. **If** there is a tie in frequency, select the class label whose neighbors have the lowest average distance.
11. Output the final prediction for the test sample.

Algorithm 2. Reduce phase.

Results and discussion

In order to evaluate the effectiveness of the proposed method in predicting flight cancellations and delays, MATLAB 2020a software was utilized. In this research, a test scenario based on cross-validation was planned with 10 iterations, in which the training and test phases are repeated 10 times, and in each iteration, 90% of the database samples are used to train prediction models. And the remaining 10% is used to analyze the effectiveness of the method in predicting flight cancellations and delays. At the end of each iteration, the labels determined by the prediction model for each test sample have been compared with its real labels. Then, based on the results of these comparisons, the performance of the forecasting model is described in terms of accuracy, precision, recall, and F-Measure. These criteria can be calculated based on the following equations[35]:

$$Accuracy = \frac{T}{N} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$FMeasure = \frac{2.Recall.Precision}{Recall + Precision} \tag{9}$$

where T represents the number of samples (of all target classes) for which the output label of the prediction model matches the real label. On the other hand, precision and recall criteria in relations 5 and 6 describe the efficiency of the prediction model separately for each class. For this purpose, these two criteria have been calculated for each target class and each time, one of the target classes is considered as a positive class and other classes as a negative class. In these relationships, TP describes the number of samples of the positive class whose labels are correctly predicted. Also, FP indicates the number of negative samples that the prediction model placed in the positive class. Finally, FN specifies the number of samples from the positive class that are labeled as negative.

In the implementation process, 36 features listed in Table 2 were used as the input of the proposed model so that after the pre-processing step, feature selection can be done by ABC. To ensure the correct performance of this algorithm in selecting relevant features, the feature selection process was repeated 10 times and the list of features selected in different iterations was compared. The results of this process are shown in Fig. 4.a. It should be noted that in the implementation of this experiment, the number of iterations and the size of the population in the ABC algorithm are set to 500 and 150, respectively.

Figure 4a shows that each of the classes of weather conditions, airport features, and flight features have several important features related to the possibility of flight cancellation or delay, which are detected as related features in more than 80% of iterations. Based on the set of features selected in different iterations, it can be concluded that most of the weather features have a high correlation with the target variable. On the other hand, the features related to flight time and path length have little relation with the probability of flight cancellation or delay and were ignored in most iterations. In the set of features describing the flight, the information related to the type of flight, the type of aircraft, and the history of airline cancellations or delays are of great importance. Also, the results showed that the delay rate of the origin airport and the cancellation or delay history of the last 4 flights are sufficient to describe the features related to the airport. Figure 4.b shows the selection rate of each feature during 10 iterations of the ABC algorithm. According to this figure, 22 features have been selected as indicators related to flight cancellation or delay in at least 80% of iterations. In the following, these features are used as input to the proposed model.

According to the process described in Sect. 3, the proposed strategy uses a combination of feature selection techniques, ensemble learning, and distributed computing to predict flight cancellations. In order to investigate the effect of each of these techniques on the performance of the proposed method, each of the following situations has been compared with the proposed method:

- All Indicators: The purpose of implementing this mode is to study the impact of the proposed feature selection strategy on forecast accuracy. Thus, in this case, all input features (36 indicators listed in Table 2) were used to predict flight delay cancellation.

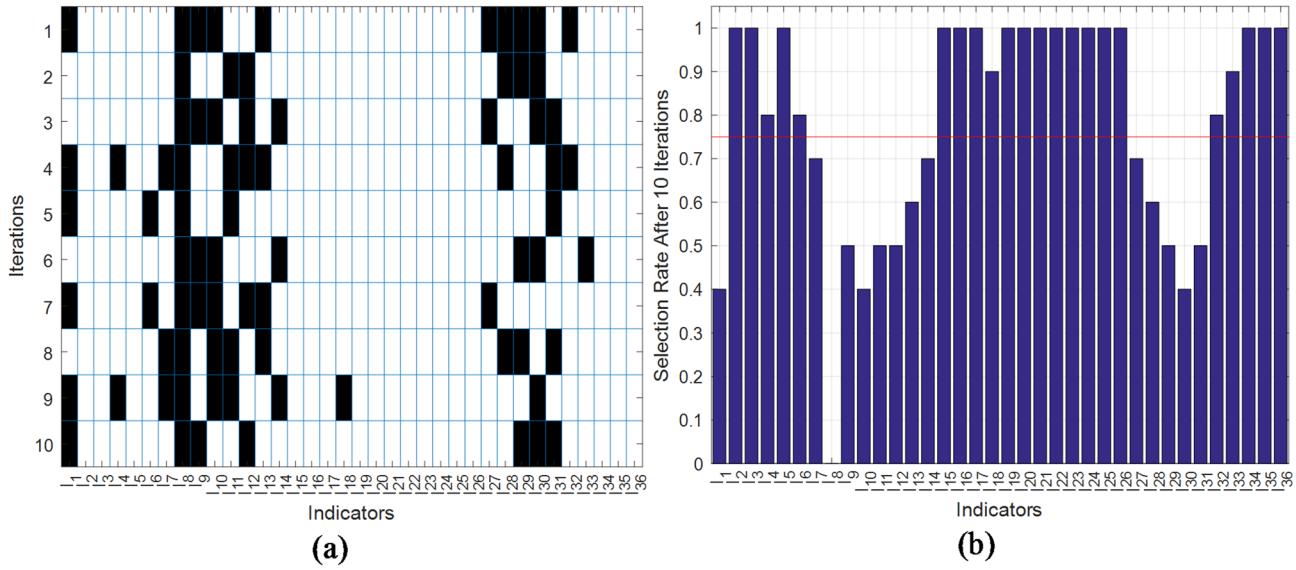

Fig. 4. The result of 10 iterations of feature selection (**a**) details of features selected in each iteration (**b**) selection rate of each feature in all iterations.

- DKNN (Without Ensemble): By comparing the results of the proposed method with this mode, the effect of the ensemble technique on increasing the accuracy of the model can be evaluated. Thus, in this case, only one DKNN model was used to predict cancellation or flight delay.
- Conventional KNN: In this case, the proposed ensemble classification model is replaced with a basic KNN model. In this case, it is possible to measure the effectiveness of distributed computing strategies on the performance of the proposed model.

In addition to the above situations, the results of the proposed method have been compared with the SVM model based on Spark cluster computing[8], the LightGBM model[9], along with the modern deep learning approaches such as GRU-LSTM[11] and LSTM[14]. It should be noted that the training and testing of all these methods were done through the same samples. Figure 5 depicts the average accuracy of different methods after 10 iterations of cross-validation.

Based on Fig. 5, the proposed method is superior in terms of accuracy compared to the other methods. If the feature selection process is ignored and all input indicators are used to predict flight cancellations and delays, the accuracy of the system is equal to 93.17%. However, by using the feature selection process of the proposed method, the prediction accuracy increases to 95.79%. Thus, the proposed feature selection strategy can increase the accuracy by 2.62%. On the other hand, if a DKNN model is used as a learning model and the ensemble learning strategy is removed, the prediction accuracy drops to 92.37%, based on which the effect of the ensemble learning model in increasing the prediction accuracy by at least 3.42% is confirmed. In addition, the comparison of the accuracy of the proposed classification model with the basic KNN shows the superiority of the proposed method by 4.47%. Finally, the comparison of the accuracy of the proposed method with the methods presented in[8,9,11], and[14] shows that the proposed method can increase the detection accuracy by at least 2.2% compared to the previous works. This superiority can be attributed to the combination of ensemble learning techniques and the distributability of the proposed classification model.

In order to check the performance of each method in more detail, it is possible to interpret the confusion matrices (Fig. 6). Figure 6.a shows the results of flight delay and cancellation prediction after 10 iterations of cross-validation by the proposed method. In this matrix, the columns correspond to the distribution of the test samples in the target classes, and the rows of this matrix show how these samples are labeled by the proposed method. The total values of the first column show that out of 60,391 OnTime flights, 57,875 samples were correctly classified by the proposed method, thus, the proposed model was able to correctly classify 95.8% of the OnTime flights. On the other hand, 46,004 samples of 48,000 Late flights have been correctly identified by the proposed method, which shows that our solution has correctly classified 95.8% of Late flights. Also, checking the performance of the proposed method for Vaery Late flights and canceled flights shows that this method was able to correctly classify 95.7% of the samples of these two classes. By checking the rows of the confusion matrix of the proposed method, the probability of each output label correctness in this method can be calculated. For example, the proposed method takes 59,329 samples as Ontime flights, of which 57,875 are correct. In this way, the probability of correctly detecting Ontime flights in the proposed method is equal to 0.9754. Similarly, the outputs determined as Late flights, VeryLate flights and canceled flights by the proposed method are correct with the probabilities above 0.95. Examining these results shows that the probability of correct output of the proposed

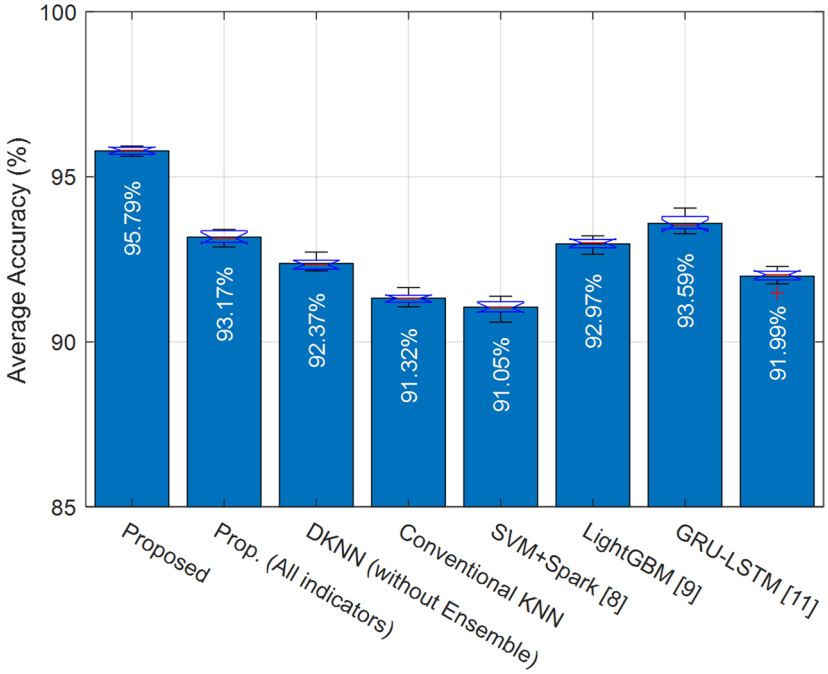

Fig. 5. The average accuracy of different methods in predicting flight cancellations and delays.

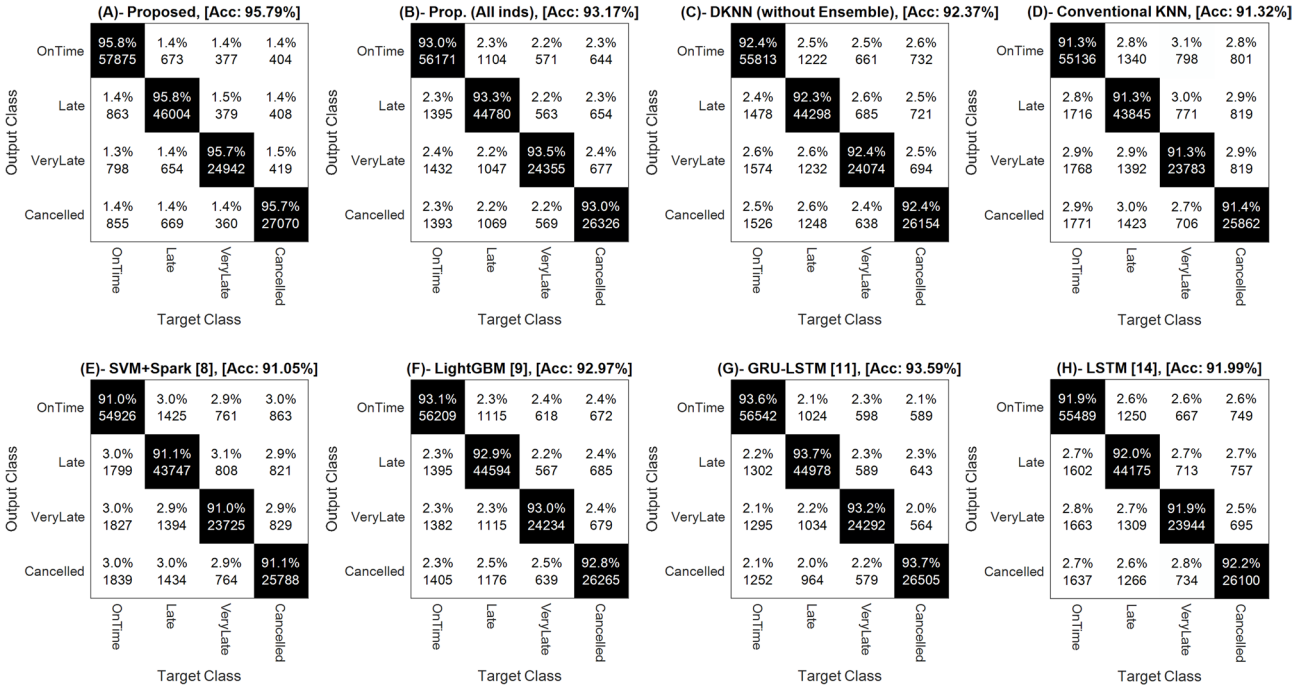

Fig. 6. Confusion matrix of different methods for predicting flight cancellations and delays.

method for each class is directly related to the number of training samples available for that class. In this way, it seems that by increasing the number of available samples for each class, the partial performance of the proposed model in detecting the samples of each class can be improved.

Interpreting the confusion matrices related to other methods in Fig. 6 can be performed similarly. In general, the comparison of these matrices shows that the outputs of the proposed method separately for each class have a higher probability of correctness and at the same time, the proposed method was able to correctly identify a higher rate of samples of each class. More accurate results can be obtained by using accuracy, recall, and F-Measure criteria. These results are presented in Fig. 7.

In Fig. 7a to c, precision, recall, and F-Measure criteria for each method are displayed separately for each target class. Also, the average performance of different methods is shown in Fig. 7d. Examining the graphs presented in Fig. 7 shows that the proposed method has been able to predict the cancellation status of delayed flights with higher quality compared to other methods. Figure 7a shows that the precision of the predictions provided by the proposed method was higher than other methods separately for each class. This superiority shows that the states predicted for each class by the proposed method are more likely to be correct compared to other methods. On the other hand, Fig. 7b confirms that the proposed method was able to recall the real label at a higher rate from the samples of each target class, which is the result of the superiority of the proposed method in terms of recall criteria. More precision and recall in the proposed method have led to an increase in the F-measure in Fig. 7c, which shows its more efficient prediction. In Fig. 8, the ROC curve obtained from 10 iterations of cross-validation of different methods are compared.

Based on Fig. 8, the proposed method predicts flight cancellation and delay in a way that leads to a lower false positive rate (FPR) and a higher true positive rate (TPR). This feature has caused the area under the ROC curve in the proposed method to be higher than other methods. Table 3 compares the numerical values obtained from the tests performed on different methods. Examining the results of the experiments shows that the proposed method is an efficient and accurate strategy for predicting cancellations and delays in domestic and foreign flights and can be used as a useful tool in real conditions.

In the continuation, we will study the efficiency of the DKNN model used in the proposed method in terms of computation gain. For this purpose, the speed of computation by the proposed distributed system with the centralized KNN model is compared. These experiments were conducted on a dedicated cluster consisting of one master node and five worker nodes. Each node was equipped with an Intel(R) Core(TM) i7-13700 CPU at 3.20 GHz and 16 GB of RAM, running on a 64-bit Windows 11 operating system with MATLAB 2020a. These results are shown in Fig. 9. In this figure, the computation gain criterion is calculated as follows:

$$CG = \frac{T_{KNN}}{T_{DKNN}} \tag{10}$$

In the above relationship, T_{DKNN} represents the execution time of the algorithm in distributed mode (DKNN) and T_{KNN} describes the execution time of the algorithm in centralized mode (KNN).

A distribution system has computational efficiency if $CG > 1$ and the larger value of CG means more computation gain of the distributed system. The results presented in Fig. 9 show that the computation gain

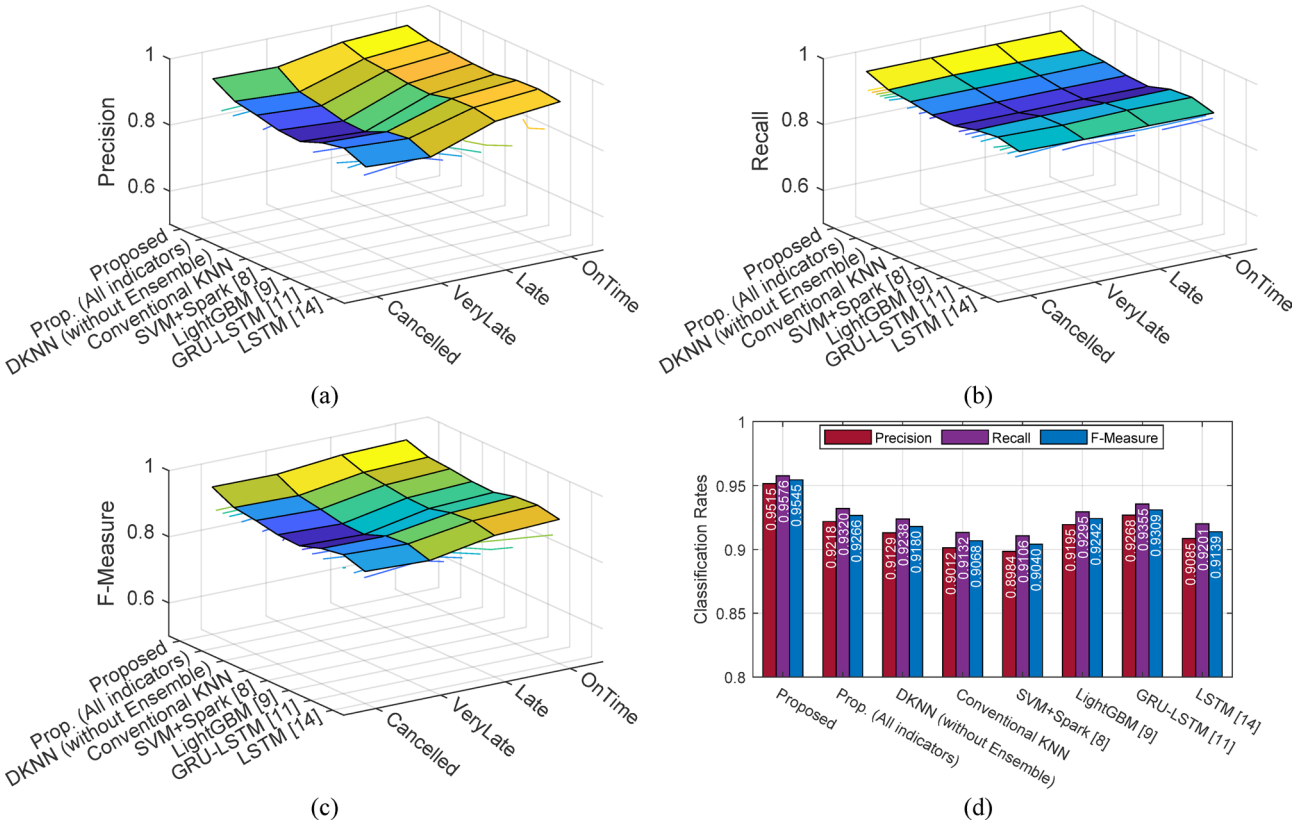

Fig. 7. Comparing the classification quality of different methods separately for each class in terms of (a) precision, (b) recall and (c) F-measure, and (d) the average of these measures for all target classes.

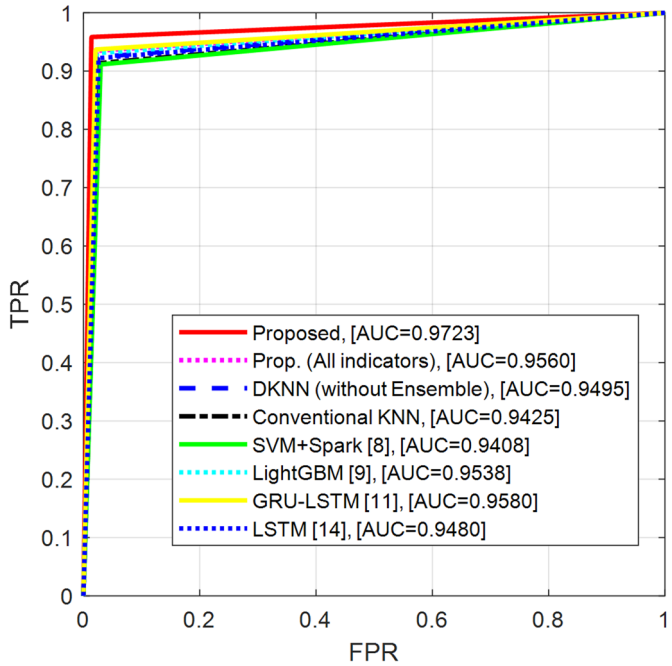

Fig. 8. ROC curve of different methods for predicting flight cancellation or delay after 10 iterations of cross-validation.

Method	Accuracy (%)	F-measure	Recall	precision	AUC
Proposed	95.7856	0.9545	0.9576	0.9515	0.9723
Proposed (All Indicators)	93.1687	0.9266	0.9320	0.9218	0.9560
DKNN (Without Ensemble)	92.3742	0.9180	0.9238	0.9129	0.9495
Conventional KNN	91.3217	0.9068	0.9132	0.9012	0.9425
SVM + Spark[8]	91.0513	0.9040	0.9106	0.8984	0.9408
LightGBM[9]	92.9659	0.9242	0.9295	0.9195	0.9538
GRU-LSTM[11]	93.5896	0.9309	0.9355	0.9268	0.9580
LSTM[14]	91.9865	0.9139	0.9201	0.9085	0.9480

Table 3. The performance of the introduced approach in contrast with other methods.

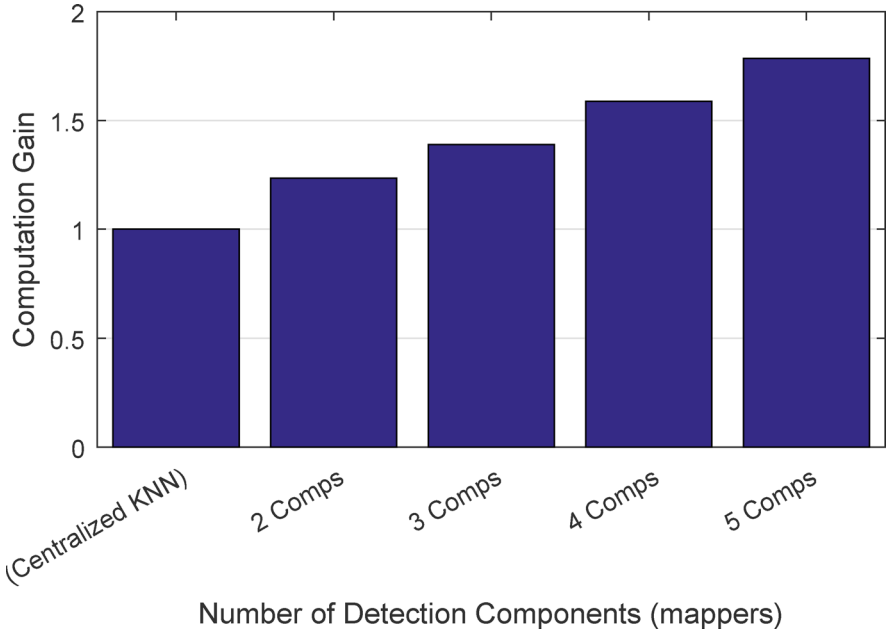

Fig. 9. Graph of the computation gain of the DKNN of the proposed method compared to the centralized KNN.

increases with the increase in the number of detection components (or in other words, the increase in the number of mappings in the DKNN model). To be able to statistically demonstrate it, we have conducted a Student t-test between the execution time of the distributed DKNN (5 components) and the centralized KNN. The obtained results of 30 independent runs provided a p-value of 0.00018 that is much less than the significance level of 0.05. This statistical data proves that the DKNN algorithm has a great computational advantage in comparison to the centralized approach. The results confirm that the DKNN algorithm in the proposed method can improve the efficiency of the forecasting system for flight data analysis in terms of processing power.

Lastly, in order to empirically justify the usefulness of our feature selection approach based on the ABC, we have conducted a comparative analysis with three other popular algorithms, i.e. "mRMR + SFS", "Relief + BFE" and a Genetic Algorithm (GA). "mRMR + SFS" is a filter using mRMR[36] and Sequential Forward Selection (SFS) algorithm to identify the best subset. "Relief + BFE" on the other hand applies Relief[37] algorithm to rank the features in terms of their discriminatory power to nearest neighbors and then apply Backward Feature Elimination (BFE) to drop the less important features one at a time. This choice of comparative methods is a solid assessment since it incorporates filter-wrapper hybrid techniques and another bio-inspired metaheuristic. As shown in Table 4, the results are used to test the capacity of each method to pick a sub-set of features that maximize the perfor-mance of our ensemble DKNN model.

Table 4 indicates that the proposed feature selection technique using ABC will outperform the rest of the methods in all the evaluation metrics. The Proposed (ABC) method recorded the best values of Accuracy (95.79%), F-measure (0.9545), and AUC (0.9723). This better performance is explained by the good balance of exploration and exploitation of the ABC algorithm that allows it to find a better and less redundant feature subset than the other techniques. Although the GA also showed good performance, it was a little lower than the ABC method which indicates that the particular search mechanism of the ABC algorithm is very appropriate in optimizing feature subsets in this field.

Method	Accuracy (%)	F-measure	Recall	precision	AUC
Proposed (ABC)	95.7856	0.9545	0.9576	0.9515	0.9723
mRMR + SFS	93.9994	0.9353	0.9400	0.9311	0.9607
Relief + BFE	94.2200	0.9378	0.9424	0.9336	0.9621
GA	94.3902	0.9397	0.9441	0.9357	0.9636

Table 4. Performance comparison of feature selection Methods.

Conclusion

Advance notice of flight cancellation can be useful in various applications such as airport flight schedules, setting up alternative flight schedules of airlines, and maintaining the loyalty of airline customers. In this research, a new method was presented in order to solve the challenges related to flight cancellation prediction. In the proposed method, optimization techniques were presented in order to determine the most relevant features with the probability of flight cancellation. This feature selection method selects an optimal subset to describe the flight features by analyzing the correlation information of the features. The use of this solution is effective in increasing the speed of data processing (due to the reduction of data dimensions) and reducing the prediction error (by filtering irrelevant features); Based on it, the forecasting accuracy can be increased by 2.62%. In the proposed method, the combination of several DKNN models with different configurations was used to predict flight cancellations. The use of this ensemble system covers the partial error of each classifier through cooperation with other classification models. The findings of the research show that this ensemble system can increase the prediction accuracy by at least 2.2% compared to the classical machine learning and modern deep learning strategies. On the other hand, the use of the MapReduce computing model in DKNN models has increased the ability of the proposed system in processing flight big data, and this distributed model can increase the computation gain by at least 25% compared to the centralized model. Checking the performance of the proposed method based on real flight data shows its accuracy of 95.79%, which is a 2.4% increase compared to the previous methods. These results confirm that the proposed strategy is an efficient model for predicting flight cancellations and delays and can be effectively used in real-world scenarios.

Limitations and future work

We propose a new distributed ensemble approach to scalable flight cancellation forecasting in big data in our work. Although our experimental findings prove its efficacy, we recognize a number of limitations that can offer clear and promising directions of future studies:

To begin with, our model was tested experimentally in one geographical area, using data of three busy airports in New York City (JFK, LGA, and EWR). As a result, the model was tested in terms of its performance given a certain combination of operational and meteorological conditions. We have not conducted any test of our findings in terms of generalizability to other regions that have diverse airport characteristics, weather, or flights.

Second, the current implementation of the ensemble architecture is homogenous, i.e. it consists of DKNN models with varying k values only. Though this design was deliberate in isolating the performance advantages of the distributed and voting mechanisms, it may not be in a position to exploit the full potential of an ensemble learning approach. A more heterogeneous, more diverse ensemble may be able to pool the benefits of different kinds of classifiers in such a way that the predictive accuracy and robustness are improved further.

Lastly, the model proposed will be based on historical data to determine predictive patterns. It is not dynamic in responding to unexpected, unpredicted events or anomalies that may affect flight cancellations, e.g. a mass power failure or a fast-developing, unprecedented weather situation. Although our model is effective on the historical data that it was trained on, its real-time flexibility to such new circumstances is an aspect that should be considered further.

On the basis of the limitations identified, we propose the following future research directions:

- Cross-Regional Validation and Generalizability: The methodology that we employed will need to be validated in the future on the datasets of other geographic regions with different operational and environmental conditions. This would give a better evaluation of how far the model can be generalized and be practically applied in the airline industry.
- Heterogeneous Ensemble Architecture: We plan to generalize our distributed system to a heterogeneous set of classifiers. This may include combining with other distributed models e.g. Distributed Support Vector Machines (SVM) or Distributed Random Forests, with our DKNN models. It can lead to a stronger and more accurate predictive system because it will be capable of recording a wider array of data patterns.
- Real-time Dynamic Adaptation: In future, there is a need to devise means of ensuring that the model adapts to unforeseen events. This can be done by incorporating a real time anomaly detection module or incorporating more dynamic, short term data feeds to make it more predictive to unexpected events.
- Integration of Real-Time Features: We will also look at the option of adding real-time or near real-time data, e.g., live weather radar data or real-time air traffic control data, as a way of making the model more responsive. This would allow the model to make better and timely forecasts of already ongoing flights.

Data availability

The dataset employed in this research is publicly available on GitHub through: (https://github.com/HoYinKan-Research/Flight-Cancellation-Data).

Received: 11 May 2025; Accepted: 3 September 2025
Published online: 07 October 2025

References

1. Gössling, S., Neger, C., Steiger, R. & Bell, R. *Weather, Climate Change, and Transport: a Review*1–20 (Natural Hazards, 2023).
2. Carvalho, L. et al. On the relevance of data science for flight delay research: a systematic review. *Transp. Reviews*. **41** (4), 499–528 (2021).
3. Zachariah, R. A., Sharma, S. & Kumar, V. Systematic review of passenger demand forecasting in aviation industry. *Multimedia Tools Appl.*, 1–37. (2023).
4. Afonso, F. et al. Strategies towards a more sustainable aviation: A systematic review. *Prog. Aerosp. Sci.* **137**, 100878 (2023).
5. Zhou, L., Liang, Z., Chou, C. A. & Chaovalitwongse, W. A. Airline planning and scheduling: models and solution methodologies. *Front. Eng. Manage.* **7** (1), 1–26 (2020).
6. Wang, F., Bi, J., Xie, D. & Zhao, X. Flight delay forecasting and analysis of direct and indirect factors. *IET Intel. Transport Syst.* **16** (7), 890–907 (2022).
7. Gui, G. et al. Flight delay prediction based on aviation big data and machine learning. *IEEE Trans. Veh. Technol.* **69** (1), 140–150 (2019).
8. Yanying, Y., Mo, H. & Haifeng, L. A classification prediction analysis of flight cancellation based on spark. *Procedia Comput. Sci.* **162**, 480–486 (2019).
9. Lambelho, M., Mitici, M., Pickup, S. & Marsden, A. Assessing strategic flight schedules at an airport using machine learning-based flight delay and cancellation predictions. *J. Air Transp. Manage.* **82**, 101737 (2020).
10. Shu, Z. Analysis of flight delay and cancellation prediction based on machine learning models. In 2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI) (pp. 260–267). IEEE. (2021), December.
11. Bandyopadhyay, S. K., Goyel, V. & DUTTA, S. *Prediction of Air Flight Cancellation during COVID-19 Using Deep Learning Methods* (ScienceOpen Preprints, 2020).
12. Mohammed, Z., Asghar, M. & Kanwal, N. Analyzing the impact of COVID-19 on flight cancellation using machine learning and deep learning algorithms for a highly unbalanced dataset. In 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET) (pp. 1–6). IEEE. (2021).
13. Hendrickx, R., Zoutendijk, M., Mitici, M. & Schäfer, J. Considering Airport Planners' Preferences and Imbalanced Datasets when Predicting Flight Delays and Cancellations. In 2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC) (pp. 1–10). IEEE. (2021), October.
14. Ayadin, A. & Akcayol, M. A. Deep Learning Based Forecasting Of Cancellation, Delay And Orientation On Flights. In 2021 1st International Conference On Informatics And Computer Science (p. 79). (2021).
15. Diao, Q., Taylor, C. P. & Wanke, C. R. Estimating Cancellation Costs for Real-Time Decision Support. In AIAA Aviation 2019 Forum (p. 3511). (2019).
16. Gong, Z., Wang, H., Nie, Q., Zhang, Z. & Xiao, Q. The personalized recommendation for OTA flight cancellation and change services during the pandemic. *J. Revenue Pricing Manag.* **22** (2), 157–165 (2023).
17. Taylor, C., Tien, S. L., Vargo, E. & Wanke, C. Strategic flight cancellation under ground delay program uncertainty. *J. Air Transp.* **29** (1), 5–15 (2021).
18. Imran, S. H., Anjum, A., Ananthasiri, C. H. & Viraja, D. U. Build an machine learning model for prediction flight delays with error calculation. *Int. J. Manage. Res. Reviews*. **13** (2), 12–17 (2023).
19. Li, Q., Guan, X. & Liu, J. A CNN-LSTM framework for flight delay prediction. *Expert Syst. Appl.* **227**, 120287 (2023).
20. Wu, Y., Yang, H., Lin, Y. & Liu, H. *Spatiotemporal Propagation Learning for Network-Wide Flight Delay Prediction* (IEEE Transactions on Knowledge and Data Engineering, 2023).
21. Qu, J., Wu, S. & Zhang, J. Flight delay propagation prediction based on deep learning. *Mathematics* **11** (3), 494 (2023).
22. Qu, J., Chen, B., Liu, C. & Wang, J. Flight delay prediction model based on lightweight network ECA-MobileNetV3. *Electronics* **12** (6), 1434 (2023).
23. Kaiquan, C. A. I. et al. A geographical and operational deep graph convolutional approach for flight delay prediction. *Chin. J. Aeronaut.* **36** (3), 357–367 (2023).
24. Li, Q., Jing, R. & Dong, Z. S. *Flight Delay Prediction with Priority Information of Weather and Non-Weather Features* (IEEE Transactions on Intelligent Transportation Systems, 2023).
25. Paramita, C., Supriyanto, C., Syarifuddin, L. A. & Rafrastara, F. A. The use of cluster computing and random forest Algoritm for flight delay prediction. *Int. J. Comput. Sci. Inform. Secur. (IJCSIS)*, **20**(2). (2022).
26. Wang, X., Wang, Z., Wan, L. & Tian, Y. Prediction of flight delays at Beijing capital international airport based on ensemble methods. *Appl. Sci.* **12** (20), 10621 (2022).
27. Ansari, A., Shaikh, A., Mapkar, S. & Khan, M. Cancellation Prediction for Flight Data Using Machine Learning. In 2nd International Conference on Advances in Science & Technology (ICAST). (2019), April.
28. Sheikhpour, R., Mohammadi, M., Berahmand, K., Saberi-Movahed, F. & Khosravi, H. Robust semi-supervised multi-label feature selection based on shared subspace and manifold learning. *Inf. Sci.* **699**, 121800 (2025).
29. Saberi-Movahed, F., Berahman, K., Sheikhpour, R., Li, Y. & Pan, S. Nonnegative matrix factorization in dimensionality reduction: A survey. arXiv preprint arXiv:2405.03615. (2024).
30. Berahmand, K., Saberi-Movahed, F., Sheikhpour, R., Li, Y. & Jalili, M. A comprehensive survey on spectral clustering with graph structure learning. *ArXiv Preprint arXiv*, 250113597. (2025).
31. Singh, D. & Singh, B. Investigating the impact of data normalization on classification performance. *Appl. Soft Comput.* **97**, 105524 (2020).
32. Kaya, E., Gorkemli, B., Akay, B. & Karaboga, D. A review on the studies employing artificial bee colony algorithm to solve combinatorial optimization problems. *Eng. Appl. Artif. Intell.* **115**, 105311 (2022).
33. Dong, X., Yu, Z., Cao, W., Shi, Y. & Ma, Q. A survey on ensemble learning. *Front. Comput. Sci.* **14**, 241–258 (2020).
34. Maillo, J., Triguero, I. & Herrera, F. A mapreduce-based k-nearest neighbor approach for big data classification. In 2015 IEEE Trustcom/BigDataSE/ISPA (Vol. 2, 167–172). IEEE. (2015), August.
35. Goutte, C. & Gaussier, E. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In European conference on information retrieval (pp. 345–359). Berlin, Heidelberg: Springer Berlin Heidelberg. (2005), March.
36. Jo, I., Lee, S. & Oh, S. Improved measures of redundancy and relevance for mRMR feature selection. *Computers* **8** (2), 42 (2019).
37. Urbanowicz, R. J., Meeker, M., La Cava, W., Olson, R. S. & Moore, J. H. Relief-based feature selection: introduction and review. *J. Biomed. Inform.* **85**, 189–203 (2018).

Author contributions

Conceptualization, X.Y.K.; methodology, X.Y.K. and K.C.; software, K.C. and P.C.P.; validation, H.Y.K and K.C.; formal analysis, H.Y.K and K.C.; investigation, K.C.; resources, H.Y.K and P.C.P.; data curation, H.Y.K and K.C.; writing—original draft preparation, H.Y.K.; writing—review and editing, H.Y.K and K.C.; visualization, P.C.P.; supervision, H.Y.K.; project administration, H.Y.K. All authors have read and agreed to the published version

of the manuscript.

Funding
This work was supported by Macao Polytechnic University (project code: RP/CEC-01/2022). The APC was funded by Macao Polytechnic University.

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to H.Y.K.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.