Engineering Journal issn: 0125-8281

Article

Flight Delay Prediction Using a Hybrid Deep Learning Method

Warittorn Cheevachaipimol[1,a], **Bhudharhita Teinwan**[1,b], **and Parames Chutima**[2,3,c,*]

1 Department of Aerospace Engineering, Faculty of Engineering, Chulalongkorn University, Thailand
2 Department of Industrial Engineering, Faculty of Engineering, Chulalongkorn University, Thailand
3 Academy of Science, The Royal Society of Thailand, Thailand
E-mail: [a]wcheevachaipimol@gmail.com, [b]teinwan.bhudharhita@gmail.com, [c,*]cparames.c@chula.ac.th (Corresponding author)

Abstract. The operational effectiveness of airports and airlines greatly relies on punctuality. Many conventional machine learning and deep learning algorithms are applied in the analysis of air traffic data. However, the hybrid deep learning (HDL) model demonstrates great success with superior results in many complex problems, e.g. image classification and behaviour detection based on video data. Interestingly, no previous attempts have been made to apply the concept of HDL in analysing structured air traffic data before. Hence, this research investigates the effectiveness of the HDL in the departure delays severity prediction (i.e. on-time, delay and extremely delay) for 10 major airports in the U.S. that experience high ground and air congestion. The proposed HDL model is a combination of a feed-forward artificial neural network model with three hidden layers and a conventional gradient boosted tree model (XGBoost). Utilising the passenger flight on-time performance data from the U.S. Department of Transportation, the proposed HDL model achieves a sharp rise of 22.95% in accuracy when compared to a pure neural network model. However, with current data used in this research, a pure machine learning model achieves the best prediction accuracy.

Keywords: Flight delay prediction, machine learning, XGBoost, feed-forward artificial neural network, deep learning, hybrid deep learning.

ENGINEERING JOURNAL Volume 25 Issue 8
Received 29 May 2021
Accepted 26 July 2021
Published 31 August 2021
Online at https://engj.org/
DOI:10.4186/ej.2021.25.8.99

DOI:10.4186/ej.2021.25.8.99

1. Introduction

Commercial air travel has been growing at an increasing rate in the past decade due to the ever-increasing influence of globalisation that the world has experienced in the 21st century. According to a 2019 report from International Civil Aviation Organization (ICAO), air transport has seen a clear sign of continuous growth in the aviation sector. From 2018 to 2019 alone, there was a total increase of 155 million commercial passengers resulting in average year-on-year growth of about 6.07% [1]. This upward trend can be quite problematic as traffic growth outpaces capacity expansion, which can lead to a noticeable impact on the aviation industry.

In 2007 alone, it was estimated that the direct cost of air transportation delays in the U.S. totalled up to nearly $33 billion, of which $16.7 billion was passed down to the passengers [2]. Therefore, one can presume that in a transportation system, the delay is something that management has to pay great attention to. Flight delays are proven not only to hurt the aviation business in both financial and operational aspects but also can induce a chain reaction to other industries. To improve the overall performance, whilst minimising unnecessary hidden costs, a flight delay prediction model is highly desirable for risk mitigation and avoidance preparations for commercial aviation stakeholders.

Factors that could lead to flight delays arise from both internal and external factors. Internal factors are those that are controllable by the airline, such as gate availability. Meanwhile, external factors are those dependent on other uncontrollable factors such as passenger and luggage handling delays and bad weather. Due to a myriad number of concerning data, flight delay prediction is recognised as one of the most challenging problems in the aviation industry. Moreover, unexpected events caused by internal and external factors increase the complexity of the problem significantly.

Completing tasks on time is one of the most prominent indicators for assessing the effectiveness of airports and airlines management. However, in reality, theoretical on-time work practices may be difficult to carry out due to uncontrollable factors such as bad weather, sudden sickness of the pilot and full apron. According to the Federal Aviation Administration (FAA) and aviation operational standards, flights that have landed more than 15 minutes past their originally scheduled slot is considered to be "*delayed*". Many airports also impose regulations that require an airline to compensate its passengers if the delay experienced exceeds a certain threshold. In major airports, congestion is quite common and as such, delays are quite common as well. As delays usually propagate to consecutive flights, the issue can easily compound and therefore, can cause detrimental effects. As such, it is likely to be beneficial to all stakeholders involved in air travel if the flight delays can be predicted as it will enable them to prepare for and make the necessary response to reduce further consequences.

Several known approaches could be used to create a predictive flight delay model such as conventional statistical analysis, machine learning, and deep learning. Artificial intelligence and data analytics have propelled researchers to find possible applications for flight delay prediction for commercial airlines to use. Therefore, it is necessary to understand how different techniques align with our objective and application domain.

Hybrid deep learning (HDL) has been found to have superior performance relative to state-of-the-art deep neural network models when working with complex data by achieving higher accuracy. However, those models were fed with complex data, which includes real-time video data and images, which are considered unstructured data. However, it is worth noting that the viability of using a hybrid deep learning model with structured data is worth studying as it may prove to perform similarly. Also, to the best of our knowledge, no publication has been contributed to the research area in which an HDL model is applied to predict flight delays before.

In this paper, the HDL approach is incorporated to predict flight delays for U.S. domestic departing flights, covering the top 10 busiest airports. It aims to investigate if incorporating the learning technique feature of a feed-forward neural network along with a gradient boosting, which is known to require feature selection, can help improve the accuracy of the model.

The following sequence is used to present this paper. Section 2 presents past studies in flight delay prediction tasks. Section 3 explains how the retrieved data is preprocessed. The proposed hybrid deep learning model is explained in detail in Section 4, where the network architectures are shown. Section 5 presents the experimental results using the machine learning model, feed-forward artificial neural network model, and our proposed hybrid deep learning model. The conclusion and future research directions are given in Section 6.

2. Literature Review

Accurately predicting flight delay has been a challenging issue for researchers and practitioners for decades. However, recent studies have demonstrated the applicability of using state-of-the-art computer-based approaches such as big data, machine learning, and deep learning to achieve better flight delay prediction results relative to conventional statistical approaches.

Sternberg et al. [3] explored multiple data science approaches and their corresponding taxonomy which includes machine learning, probabilistic models, statistical analysis, network representation, and operational research to create models to predict flight delays. Mueller and Chatterji [4] analysed and explored the characteristics of aircraft arrival and departure delay and developed a time-series model to determine the correlation between features and probability of a certain delay time that was determined based on a Normal and Poisson distribution probability density function.

Vandehzad [5] implemented linear and non-linear kernels, linear regression algorithm and combined method with mathematically weighted values to predict future delays with data provided from Aviolinx, a Swedish software provider which serves the airline industry. Gui et al. [6] explored factors that influenced flight delays by collecting data from an automatic dependent surveillance-broadcast aviation data platform and applied the data to LTSM-based and random forest-based prediction models to predict flight delays.

Balcastro et al. [7] attempted to predict arrival delay by incorporating weather data along with the use of parallel algorithms (MapReduce). Chen and Li [8] presented a combined multi-label random forest classification and delay propagation model with an optimal feature selection process to predict chained delay while taking initial departure delay into account.

Ye et al. [9] applied four supervised learning methods: multiple linear regression, support vector machine, extremely randomised trees, and Light Gradient Boosting Machine (LightGBM) along with meteorological data to predict departure delay at Nanjing Lukou International Airport. Manna et al. [10] applied a gradient boosted decision tree to analyse air traffic data, which resulted in better a higher accuracy air traffic delay prediction model. Chakrabarty [11] proposed a gradient boosting classifier model with grid search hyper-parameter tuning for predicting American Airlines arrival delay in the top 5 busiest airports using a binary classification approach.

Kuhn and Jamadagni [12] applied a decision tree, logistic regression, and single-layer neural network to classify whether the arrival flight will be delayed or not by using the top 3 features by importance as inputs. Takeichi et al. [13] applied queue analysis and Rectified Linear Unit (ReLU) function ANN along with congestion parameters to predict arrival delays at Tokyo Airport.

Gopalakrishnan and Balakrishnan [14] introduced a two-hour horizon prediction method based on the origin-destination pair link along with the use of Markov Jump Linear System (MJLS) and multiple Artificial Neural Network (ANN) architectures. Yu et al. [15] applied the combined deep belief network and support vector regression (DBN_SVR) model into the top layer to predict flight delays at Beijing International Airport with having novel factors such as the degree of crowdedness at an airport and the current air route situation.

Kim et al. [16] implemented multiple ways of building deep Long Short-Term Memory (LSTM) and Recurrent Neural Networks (RNN) architecture to predict flight delays with various sequences and thresholds. Ai et al. [17] employed convolutional long short-term memory (conv-LSTM) network to predict airport delays and apprehend the temporal and spatial characteristics in China's civil aviation area.

Khanmohammadi et al. [18] introduced a new Defect of Modules Prediction (DMP) ANN structure to predict delays of incoming flights at JFK airport. Lv et al. [19] proposed the application of the stacked autoencoders (SAE) model to a greedy deep learning model architecture to best learn the features for traffic flow prediction.

The new architecture of the hybrid model – combining multiple machine learning architectures – has been showing superior performance in multiple challenging tasks. Ding et al. [20] developed a hybrid model of CNN+LSTM to detect unsafe behaviour from video streaming. The results showed that the accuracy from employing this model exceeded the current state-of-the-art descriptor-based method.

Kim et al. [21] developed a deep neural network (DNN) hybrid model where diverse types of data were fed into various DNN structures, of which the data from the layers of each source were extracted, combined, and fed into a fully connected layer DNN to accurately predict consumer repurchasing behaviour. Bhattacharya [22] applied a hybrid model of DNN and classical machine learning to diagnose disease from a chest X-ray image. Results show that the model reaches over 93% accuracy.

With hybrid deep learning architecture's superior performance, all the revised previous studies have employed the model with complex data such as image and video data. However, there has not been any study on applying hybrid deep learning to the structured dataset with structured information. Therefore, the objective of this paper is to investigate the viability and efficacy of the hybrid deep learning model to flight data to predict the on-time performance of flights. This research intends to provide a new valuable asset for innovating accurate flight delay prediction to the aviation industry.

3. Data Analysis

In this study, the airline on-time performance dataset was retrieved from the U.S. Department of Transportation, Bureau of Transportation Statistics [23]. The dataset contains flight data from the beginning of 2008 to the end of 2019 by all mandatory reporting carriers in the USA of all domestic flights.

3.1. Data Preprocessing

Although the dataset contains information from 2008 to 2019, only air traffic data from the beginning of 2017 to the end of 2018 are used. The dataset is further reduced, due to computational restraints, to only include the data from the top 10 airports with regards to air traffic volume. The training and test data are split by annum, with the training receiving the year 2017 and the testing phase receiving the latter. The origin and destination airports used in the study are shown in Table 1.

Instead of using the exact times of delay, the delay is grouped into three blocks. The first block is defined as flights that took off no longer than 15 minutes past their predefined departure times. Flights that fall into this class are considered to be "*on-time flights*". The second block is defined as flights that took off within the following two hours past the initial block. These Class 1 flights are considered "*delayed flights*". Lastly, the third block consists

of all other flights past the second block. These Class 2 flights are defined as "*extremely delayed flights*". The delay grouping is also shown in Table 2.

From the available attributes, 11 significant features were selected for the model as these features would be one that could be known before the flight. The names and corresponding definitions can be seen in Table 3.

3.1.1. Gradient Boosted Trees Scaling

Due to certain features that were non-numerical, one hot encoding was performed to the following features to be able to apply them to the Gradient Boosted Tree model: OP_UNIQUE_CARRIER, ORIGIN and DEST.

Normalisation was also performed where the numerical features are normalised using a min-max scaling algorithm. The algorithm is defined in Eq. (1).

$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \quad (1)$$

where X_{norm} is the normalised values, X is the initial value and X_{min} and X_{max} are the minimum and maximum values within the list, respectively.

While normalisation is not particularly a big issue for gradient boosted models, it is necessary for an ANN model and as such, to keep consistency for a fairer comparison, it has been applied to both.

3.1.2. Artificial Neural Network (ANN)

Generally, the convergence assumption for neural network models is $\mu = 0$ and $\sigma = 1$. To build robust neural network models, standardization will be applied to the dataset fed to ANN models using the standard scaler algorithm, which is defined in Eq. (2).

$$z = \frac{X - \mu}{\sigma} \quad (2)$$

where X is the initial value, μ is a mean, shown in Eq. (3), and σ is a standard deviation, shown in Eq. (4).

$$\mu = \frac{1}{N}\sum_{i=1}^{N}(x_i) \quad (3)$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \quad (4)$$

Table 1. U.S. airports used in the study.

Airport	Airport Name
ATL	Atlanta
DEN	Denver
DFW	Dallas/Fort Wort
LAS	Las Vegas
LAX	Los Angeles
MSP	Minneapolis
ORD	Chicago
PHX	Phoenix
SEA	Seattle
SFO	San Francisco

Table 2. Reference table for departure block.

Departure Delay Group	Definition	Departure Delay in Minutes
0	On-time	[-∞,15]
1	Delay	[15,135]
2	Extreme Delay	[135,∞]

Table 3. Feature definition.

Feature	Definition	Type
YEAR	Year	Categorical
MONTH	Month	Categorical
DAY_OF_MONTH	Day of month	Categorical
OP_UNIQUE_CARRIER	Unique carrier code	Categorical
ORIGIN	Origin airport	Categorical
DEST	Destination airport	Categorical
CRS_DEP_TIME	Computer Reserved System departure time	Continuous
DEP_DELAY_GROUP	Departure delay, categorised in groups	Categorical
CRS_ARR_TIME	Computer Reserved System arrival time	Continuous
CRS_ELAPSED_TIMED	Difference between the Computer Reserved System departure and arrival time	Continuous
DISTANCE	Distance between airport in miles	Continuous

3.2. Dealing with Imbalanced Dataset

Due to the nature of air traffic, most flights are typically not delayed. Consequently, this results in a skewed dataset that does not equally represent the proportions of flights that are on time and flights that are delayed while training. Therefore, to prevent a biased result towards the majority label, oversampling using the

Synthetic Minority Oversampling Technique (SMOTE) [24] is used to deal with this imbalance. While applying SMOTE, the number of nearest neighbours used was 5 and the corresponding departure delay group distribution before and after oversampling can be seen in Fig. 1 and Fig. 2 respectively.

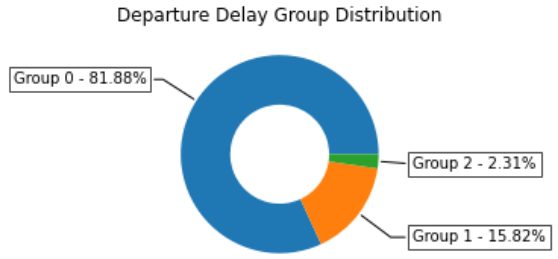

Fig. 1. Departure delay group distribution for training data prior to applying SMOTE.

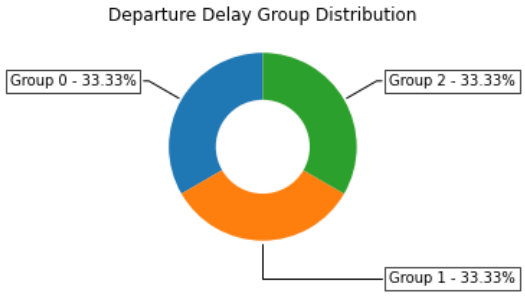

Fig. 2. Departure delay group distribution for training dataset post -application of SMOTE.

4. Model

4.1. Gradient Boosted Trees

Gradient boosted tree is an ensemble method that performs classification by combining the outputs from individual trees. In this research, XGBoost, which is an extreme gradient boosting tree model, was used. In particular, XGBoost is known to perform quite well in most situations due to its regularised model formalization which helps prevents overfitting. Since our objective is to train a multi-classification model, the objective function for the classifier is set to 'multi:softmax'. The evaluation metric used for the model is 'mlogloss', which is described in Eq. (5).

$$\log\text{loss} = -\frac{1}{N}\sum_{i=0}^{N} y_i \log(p_i) + (1 - y_i)\log(1 - p_i) \quad (5)$$

where N is the number of samples, y_i is the outcome, and p_i is the probability of the true output.

The model was constructed using the hyperparameter optimization framework (Optuna) [25] due to its efficient searching strategy. The study's goal was to maximise the accuracy score, which is defined in Eq. (6).

$$\text{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{sample}}}\sum_{i=0}^{n_{\text{sample}}} 1(\hat{y}_i - y_i) \quad (6)$$

where $\hat{y}_i$ is the predicted value, y_i is the true value, n_{sample} is the number of samples, and $1(x)$ is the indicator function.

The sampler used in the study was the Tree-structured Parzen Estimator Algorithm, with the total number of trials set to be 100. The parameter search space used can be seen in Table 4 along with the resulting hyperparameter importance in Fig. 3, the optimisation history plot in Fig. 4 and final parameter values in Table 5.

Table 4. Parameter search space – Gradient Boosted Trees.

Parameter Name	Parameter Key	Range	Step
Number of estimators	n_estimator	[0, 2500]	-
Maximum Tree Depth	max_depth	[1, 15]	-
Minimum Child Weight	min_child_weight	[1, 10]	-
Learning Rate	eta	[1e-08, 1]	-
Minimum Split Loss	gamma	[1e-08, 1]	-
Subsample Ratio	subsample	[0.5, 0.9]	0.1
Subsample Ratio by Tree	colsample_bytree	[0.5, 0.9]	0.1

Table 5. Final hyperparameters used for Gradient Boosting Tree Model.

Parameter Key	Value
n_estimator	1975
max_depth	2
min_child_weight	1
eta	3.18e-08
gamma	6.29e-07
subsample	0.6
colsample_bytree	0.5

DOI:10.4186/ej.2021.25.8.99

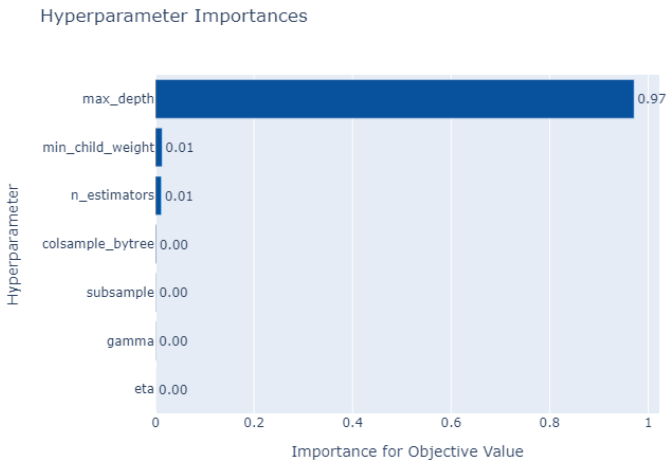

Fig. 3. Hyperparameter importance – Gradient Boosted Trees.

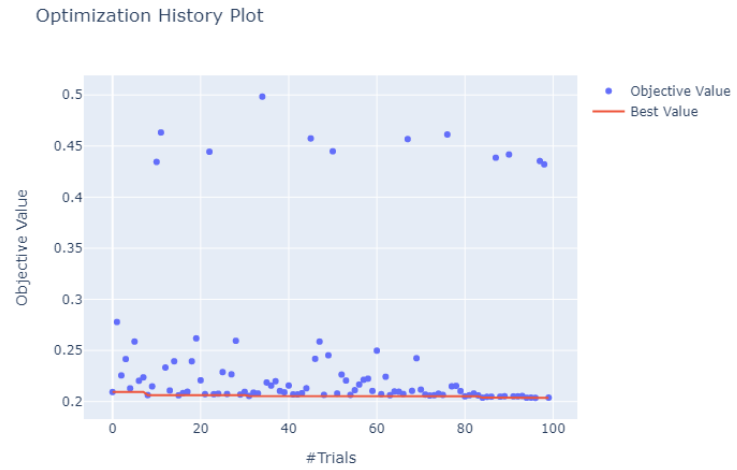

Fig. 4. Optimization history plot – Gradient Boosted Trees.

4.2. Artificial Neural Network (ANN)

The ANN model is built with the Keras framework [26]. Figure 5 shows the schematic diagram of the neural network used in this research. As shown, the constructed neural network consists of three hidden layers with 256 neurons each, which resulted from the hyperparameter tuning process. Before the feed-forward dense layers, the categorical features OP_UNIQUE_CARRIER, ORIGIN and DEST go through an embedding lookup process. The categorical variable under each feature is mapped to a dense vector representing each distinct category. The embedding approach allows the model to learn the relationships between each category through the dimensions of the resulting embedded vector [27].

The model has multiple inputs, as the features used consisted of both numeric and categorical data. Each categorical variable will be treated as separate input and needs to enter the embedding process before being feed into the dense layers. There are also three separate vectors as a result of each categorical feature. Then, the embedded vector for each categorical input is concatenated into a vector for the dense layers. For non-categorical data, the input can be directly fed into the dense layers. The resulting embedded vector is then merged with the numerical inputs and fed into the model for training.

To avoid the risk of overfitting, batch normalisation is applied between each layer which also accelerates training through standardisation. For the neurons in the hidden layers, Rectified Linear Unit (ReLU) was used as the activation function. In the output layer, the Softmax activation function, which is described in Eq. (7), is used to classify the result into multiple classes. A categorical cross-entropy loss function is applied, which is described in Eq. (8).

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_j^K e^{z_j}} \quad (7)$$

where $\vec{z}$ is the input vector to the softmax function, z_i is the element of the input vector, K is the number of classes in the multi-class classifier, and $\sum_j^K e^{z_j}$ is used for normalisation that results in a valid probability distribution.

$$\text{CE} = -\sum_i^C t_i \log\left(\sigma(\vec{z})_i\right) \tag{8}$$

where t_i is the ground truth for each class i in C and $\sigma(\vec{z})_i$ is the softmax output for each element in the vector.

The hyperparameter optimization framework Optuna was applied to the neural network model. As aforementioned in the Gradient Boosted Trees section, we aim at maximising the accuracy score.

Similar to the prior gradient boosted model, the hyperparameter optimization framework, Optuna, was used to determine the best performing parameters. The total number of trials for the study was set to be 25 due to computational limits and the following parameter search space used can be seen in Table 6. The resulting hyperparameter importance, optimization history plot and final parameter values are shown in Fig. 6, Fig. 7, and Table 7 respectively.

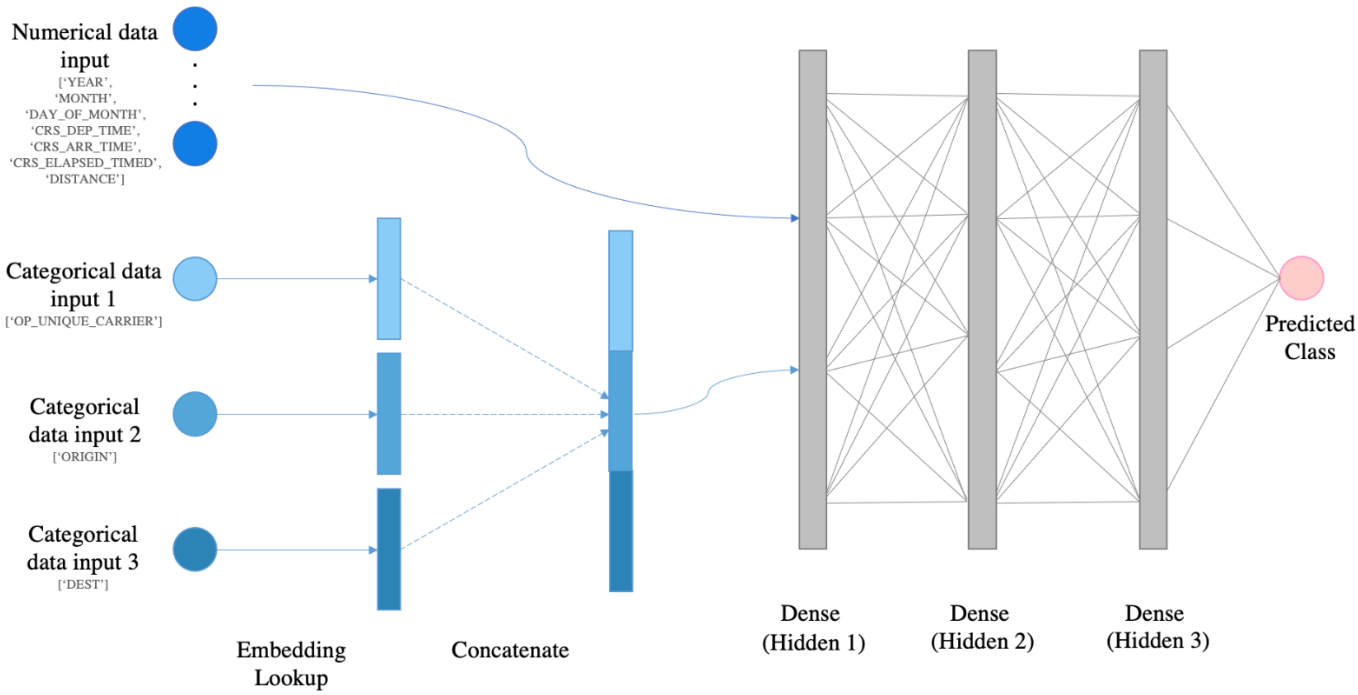

Fig. 5. Schematic diagram of feed forward neural network used in the project.

DOI:10.4186/ej.2021.25.8.99

Table 6. Parameter search space – Artificial Neural Network.

Parameter Name	Parameter Key	Range
Number of neurons in hidden layer 1	n_units_hidden_1	[64, 128, 256]
Number of neurons in hidden layer 2	n_units_hidden_2	[64, 128, 256]
Number of neurons in hidden layer 3	n_units_hidden_3	[64, 128, 256]
Learning Rate	lr	[1e-5, 1e-1]
Batch size	batch_size	[256, 512, 1024, 2048]
Epochs	epochs	[50, 100]

Table 7. Final hyperparameters used for Gradient Boosting Tree Model.

Parameter Key	Value
n_units_hidden_1	256
n_units_hidden_2	256
n_units_hidden_3	256
lr	1.16e-04
batch_size	1024
epochs	95

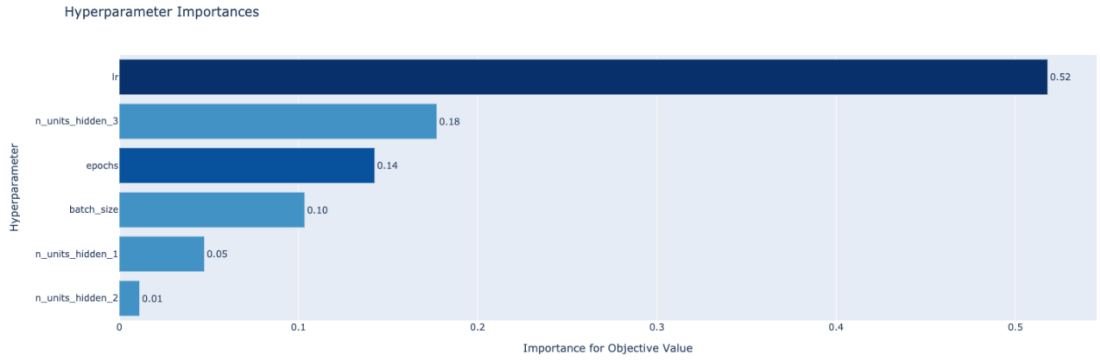

Fig. 6. Hyperparameter importance – Artificial Neural Network.

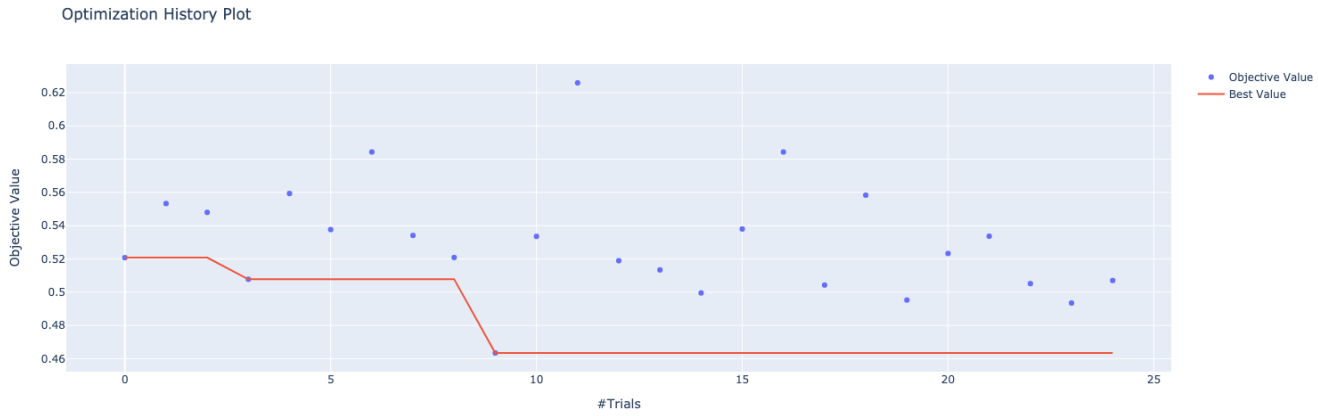

Fig. 7. Optimization History Plot – Artificial Neural Network.

4.3. Hybrid Deep Learning (HDL)

HDL is a fusion of conventional machine learning and deep learning. This combination leverages the benefits of both deep learning and machine learning to supposedly produce a model with higher accuracy whilst being less computationally expensive than traditional deep learning. The HDL architecture consists of a constructed ANN model that is succeeded by an ML classification layer as seen in Fig. 8. Tensors at the last layer, the layer before entering the prediction layer of the ANN model, are extracted and used as the inputs to the XGB model.

Both the training and testing data is required to pass through the DL portion of the HDL model as the ANN generated tensors are used as X (inputs) for the machine learning model.

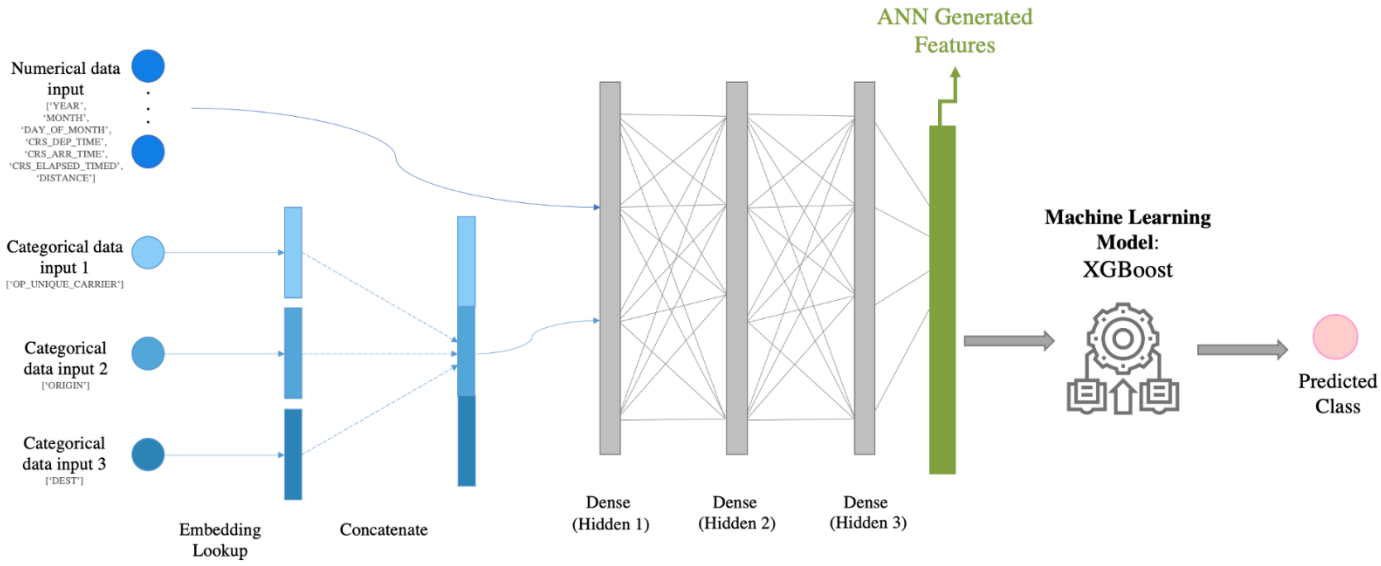

Fig. 8. Hybrid Deep Learning architecture used in the research.

With regards to the parameters used, the DL portion of the model used the same parameters as previously discovered through the hyperparameter optimization process however, the parameters for the ML portion were redone with the given outputs from the DL side. The total number of trials have been set to 50. The parameter search space can be seen in Table 8 and the resulting hyperparameter importance, optimization history plot and final parameter values can be seen in Fig. 9, Fig. 10, and Table 9, respectively.

Table 8. Parameter search space – Hybrid Deep Learning.

Parameter Name	Parameter Key	Range	Step
Number of estimators	n_estimator	[0, 2500]	-
Maximum Tree Depth	max_depth	[1, 15]	-
Minimum Child Weight	min_child_weight	[1, 10]	-
Learning Rate	eta	[1e-08, 1]	-
Minimum Split Loss	gamma	[1e-08, 1]	-
Subsample Ratio	subsample	[0.5, 0.9]	0.1
Subsample Ratio by Tree	colsample_bytree	[0.5, 0.9]	0.1

DOI:10.4186/ej.2021.25.8.99

Table 9. Final hyperparameters used for Hybrid Deep Learning Model.

Parameter Key	Value
n_estimator	1169
max_depth	15
min_child_weight	1
eta	2.18e-02
gamma	8.26e-07
subsample	0.8
colsample_bytree	0.6

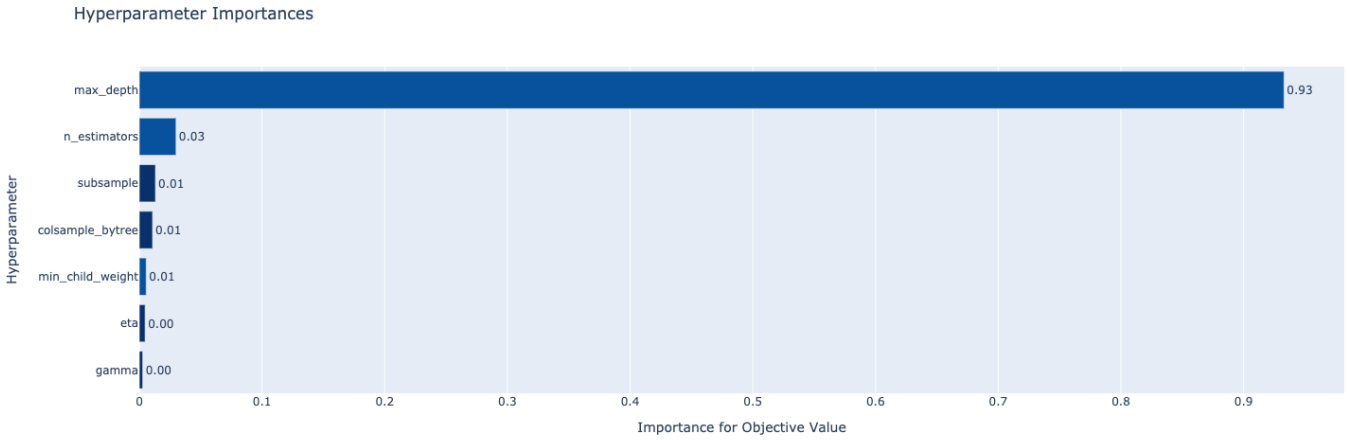

Fig. 9. Hyperparameter importance – Hybrid Deep Learning.

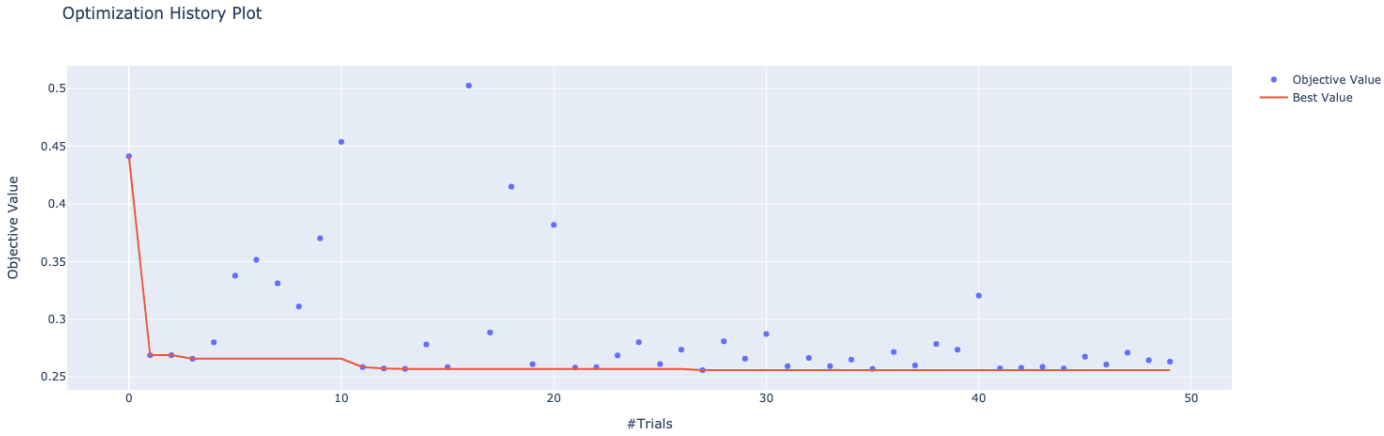

Fig. 10. Optimization history plot – Hybrid Deep Learning.

5. Results and Discussions

In this paper, the following metrics will be used to gauge the performance of the model: validation accuracy, precision, recall, F1-score and Mathew's correlation coefficient.

Validation accuracy is a metric that shows how the ratio of correct predictions is relative to the total predictions.

Precision is the proportion of correct positive identifications.

Recall is the proportion of actual positive that was correctly identified.

F1-Score, which is the harmonic mean of the model's precision and recall.

Matthew's correlation coefficient or MCC is the quality of the classification and is used to compare the results of each classes precision and recall relative to each other.

Accuracy is defined in Eq. (9).

$$\text{Accuracy} = \frac{\sum_{i=0}^{c}(\text{TP}_i + \text{TN}_i)}{\sum_{i=0}^{c}(\text{TP}_i + \text{TN}_i + \text{FP}_i + \text{FN}_i)} \quad (9)$$

Precision is defined in Eq. (10).

$$\text{Precision} = \frac{\sum_{i=0}^{c}\text{TP}_i}{\sum_{i=0}^{c}(\text{TP}_i + \text{FP}_i)} \quad (10)$$

Recall is defined in Eq. (11).

$$\text{Recall} = \frac{\sum_{i=0}^{c}\text{TP}_i}{\sum_{i=0}^{c}(\text{TP}_i + \text{FN}_i)} \quad (11)$$

F1-Score is defined in Eq. (12).

$$\text{F1} = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (12)$$

MCC is defined in Eq. (13).

$$\text{MCC} = \frac{\sum_{i=0}^{c}(TP_i \times TN_i) - (FP_i \times FN_i)}{\sqrt{\sum_{i=0}^{c}(TP_i+FP_i)(TP_i+FN_i)(TN_i+FP_i)(TN_i+FN_i)}} \quad (9)$$

The format of the Confusion Matrix, which is used to visualise the performance of the prediction model is shown in Fig. 11 [28].

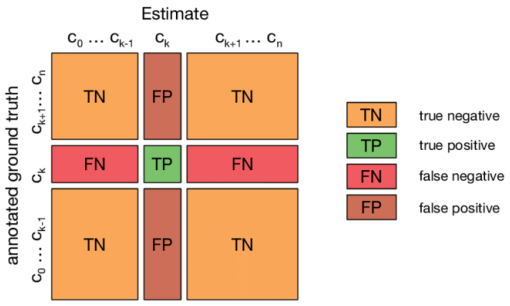

Fig. 11. Structure of confusion matrix for multi-class classification.

5.1. Gradient Boosted Trees

The results of the gradient boosted tree model returned a validation accuracy of 79.64% and its corresponding classification report can be seen in Table 10 along with the model's confusion matrix in Fig. 12. The model's performance in class 0 is decent across the board, where its precision, recall and F1-score are 0.8143, 0.9728, and 0.8865, respectively. Therefore, its performance in predicting no delay is relatively suitable. However, when predicting delays, the model's accuracy for class 1 falls short with a precision of 0.3390 and a substantially low recall and F1-Score of 0.0568 and 0.0973, respectively. Class 2 also underperforms with a precision score of 0.0358, a recall score of 0.094 and an F1-score of 0.0149.

The model also had an MCC score of 0.0723, which indicates a low correlation rate between the predicted classes and the true classes. This is substantiated by the low rate of correct labels in group 1 and group 2.

5.2. Artificial Neural Network (ANN)

The results of the feed-forward artificial neural network model returned a validation accuracy of 51.43%, which is comparatively lower than the gradient boosted tree model. The model however shows a better performance in predicting class 1 and class 2. Table 11 presents the classification report and Fig. 13 displays the confusion matrix. The model's precision, recall, and F1-score for class 0 are 0.8280, 0.5651 and 0.6718, respectively. When predicting classes 1 and 2, which are delays and extreme delays respectively, the model resulted in more solid prediction performance, when compared to the XGB model. Its precision, recall, and F1-score for class 1 are 0.1956, 0.3123, and 0.2405 respectively and it produced a precision score of 0.0242, a recall score of 0.2000 and an F1-score of 0.0432 for class 2.

The neural network model also resulted in an MCC score of 0.0420, which is significantly lower than the XGB model.

5.3. Hybrid Deep Learning (HDL)

The results of the HDL model returned a validation accuracy of 74.38%, a 22.95% increase when compared to the baseline ANN model. The model's performance in class 0 is solid, with an accuracy, recall and F1-score of 0.8143, 0.8996, and 0.8548, respectively. Although it does not perform as well as using XGBoost by itself, the vast proportion of the data is correctly identified as on time. Though the accuracy is 5.26% lower than the baseline XGBoost model, the results show that this model performs slightly better in predicting the delayed instances. The corresponding classification report is presented in Table 12 together with the model's confusion matrix in Fig. 14.

The model's precision for class 1 and class 2 is 0.2141 and 0.0285 respectively and its recall for class 1 and class 2 are 0.0912 and 0.0459 correspondingly. The MCC score for this model was 0.0345, the lowest amongst the models tested.

This leap in accuracy shows that the combination of neural network and gradient boosted trees results in a performance improvement, compared to using a neural network model by itself. As the feature set used in this study is quite small, the effects of using such an architecture may prove to be more effective, especially when using unsorted data.

To sum up the overall result, the conventional gradient boosted tree model provides the highest accuracy. However, the model has trouble in predicting delays and extreme delay instances. The ANN model shows lower ability in predicting on-time instances yet the potential in predicting delays and extreme delays is manifested.

Table 10. Classification Report for Gradient Boosted Tree Model.

	Precision	Recall	F1-Score	Support
Class 0	0.8143	0.9728	0.8865	430538
Class 1	0.3390	0.0568	0.0973	90587
Class 2	0.0358	0.094	0.0149	11381
Accuracy	-	-	0.7964	532506
Macro Average	0.3964	0.3463	0.3329	532506
Weighted Average	0.7168	0.7964	0.7336	532506

Table 11. Classification Report for Artificial Neural Network Model.

	Precision	Recall	F1-Score	Support
Class 0	0.8280	0.5651	0.6718	430538
Class 1	0.1956	0.3123	0.2405	90587
Class 2	0.0242	0.2000	0.0432	11381
Accuracy	-	-	0.5143	532506
Macro Average	0.3493	0.3591	0.3185	532506
Weighted Average	0.7033	0.5143	0.5850	532506

Table 12. Classification Report for Hybrid Deep Learning Model.

	Precision	Recall	F1-Score	Support
Class 0	0.8143	0.8996	0.8548	430538
Class 1	0.2141	0.0912	0.1280	90587
Class 2	0.0285	0.0459	0.0352	11381
Accuracy	-	-	0.7438	532506
Macro Average	0.3523	0.3456	0.3393	532506
Weighted Average	0.6954	0.7438	0.7136	532506

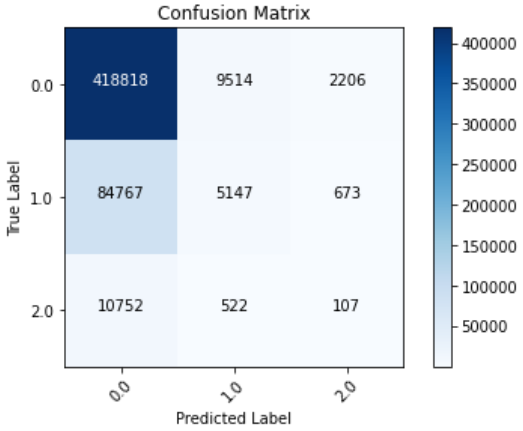

Fig. 12. Confusion Matrix for Gradient Boosted Tree Model.

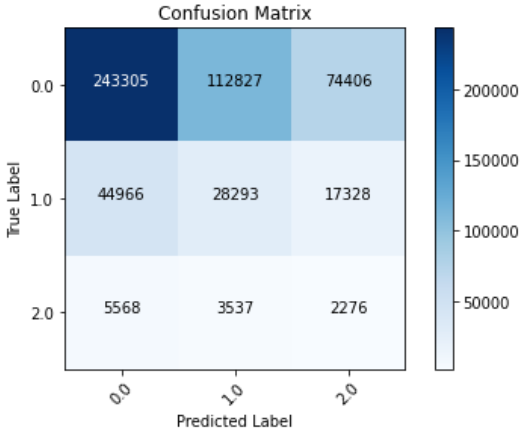

Fig. 13. Confusion Matrix for Artificial Neural Network Model.

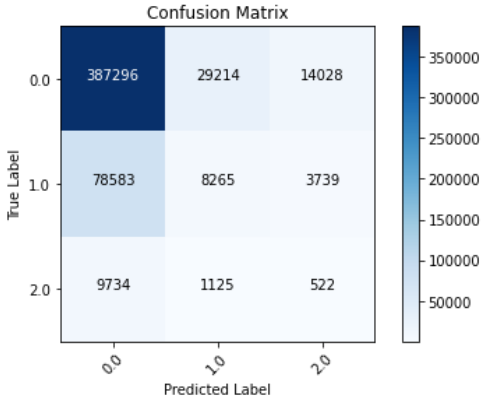

Fig. 14. Confusion Matrix for Hybrid Deep Learning Model.

6. Conclusion and Future Work

In this study, we have employed conventional machine learning, classical feed-forward deep learning, and a novel hybrid deep learning approach to predict departure on-time performance of U.S. domestic flights in the top 10 busiest airports. The goal of this study is to assess the viability and efficacy of the HDL model in comparison to a feed-forward ANN and gradient boosted tree machine learning (XGBoost). Though the HDL model did not result in the highest accuracy amongst the available model, the change in accuracy of the HDL model when compared to the ANN model shows the potential of using HDL model in predicting delayed instances.

Plausible reasons on why the HDL model did not outperform the XGBoost model may be due to data limitation. The data incompletion of this problem may not have allowed the HDL model to leverage any of the feature selecting capacity that it would have usually provided with the use of auto-encoders. Another factor could have also been due to the fact that boosting based models inherently perform well in cases with limited training data, particularly due to its intrinsic property of margin maximization during training.

However, it theorized that once more data such as weather data is included, which may or may not be structured, the model will likely outperform the XGBoost model. Therefore, more testing is required to be certain.

In future research, more cross-validation methods and larger sample size should be used to develop the model. Other methods of dealing with an imbalanced dataset should also be applied such as Edited Nearest Neighbor (ENN) in order to reduce noise from the majority sample class [29] post oversampling [30].

Other factors that can also be improved, with regards to the neural network portion, include modifying the architecture to achieve higher performance and better tuning of the model.

The dataset used in this study is also rather limited. Many other features can be implemented that may prove to be beneficial. For example, relevant data such as weather data, aircraft model, aircraft age, factors limiting the airport infrastructure such as runways available relative to average flights, may prove to help solve the issue relating to the low performance in predicting the delays.

References

[1] International Civil Aviation Organization. "International Civil Aviation Organization." https://www.icao.int/

[2] M. Ball, C. Barnhart, M. Dresner, M. Hansen, K. Neels, A. Odoni, E. Peterson, L. Sherry, A. Trani, B. Zou, R. Britto, D. Fearing, P. Swaroop, N. Uman, V. Vaze, and A. Voltes, "Total delay impact study: A comprehensive assessment of the costs and impacts of flight delay in the United States," NEXTOR, final report, 2010.

[3] A. Sternberg, J. Soares, D. Carvalho, and E. Ogasawara, "A review on flight delay prediction," 2017, arXiv:1703.06118.

[4] E. R. Mueller and G. B. Chatterji, "Analysis of aircraft arrival and departure delay characteristics," in *AIAA's Aircraft Technology, Integration, and Operations (ATIO) 2002 Technical Forum*, California, 2002.

[5] M. Vandehzad, "Efficient flight schedules with utilizing Machine Learning prediction algorithms," master's thesis, Department of Computer Science and Media Technology, Faculty of Technology and Society, Malmö University, 2020.

[6] G. Gui, F. Liu, J. Sun, J. Yang, Z. Zhou, and D. Zhao, "Flight delay prediction based on aviation big data and machine learning," *IEEE Transactions on Vehicular Technology*, vol. 69, pp. 140-150, 2019.

[7] L. Belcastro, F. Marozzo, D. Talia, and P. Trunfio, "Using scalable data mining for predicting flight delays," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, pp. 1-20, 2016.

[8] J. Chen and M. Li, "Chained predictions of flight delay using machine learning," in *AIAA Scitech 2019 forum, American Institute of Aeronautics and Astronautics*, 2019, p. 1661.

[9] B. Ye, B. Liu, Y. Tian, and L. Wan, "A methodology for predicting aggregate flight departure delays in airports based on supervised learning," *Sustainability*, vol. 12, p. 2749, 2020.

[10] S. Manna, S. Biswas, R. Kundu, S. Rakshit, P. Gupta, and S. Barman, "A statistical approach to predict flight delay using gradient boosted decision tree," in *International Conference on Computational Intelligence in Data Science (ICCIDS)*, West Bengal, 2017.

[11] N. Chakrabarty, "A data mining approach to flight arrival delay prediction for American Airlines," in *2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON)*, IEEE, pp. 102-107.

[12] N. Kuhn and N. Jamadagni, "Application of machine learning algorithms to predict flight arrival delays," CS229, Autumn 2017, 2017.

[13] N. Takeichi, R. Kaida, A. Shimomura, and T. Yamauchi, "Prediction of delay due to air traffic control by machine learning," in *AIAA Modeling and Simulation Technologies Conference*, Texas, 2017.

[14] K. Gopalakrishnan and H. Balakrishnan, "A comparative analysis of models for predicting delays in air traffic networks," in *Twelfth USA/Europe Air Traffic Management Research and Development Seminar (ATM2017)*, MA, 2017.

[15] B. Yu, Z. Guo, S. Asian, H. Wang, and G. Chen, "Flight delay prediction for commercial air transport: A deep learning approach," *Transportation Research Part E: Logistics and Transportation Review*, vol. 125, pp. 203-221, 2019.

[16] Y. J. Kim, S. Choi, S. Briceno, and D. Mavris, "A deep learning approach to flight delay prediction," in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, 2016.

[17] Y. Ai, W. Pan, C. Yang, D. Wu, and J. Tang, "A deep learning approach to predict the spatial and temporal distribution of flight delay in network," *Journal of Intelligent & Fuzzy Systems*, vol. 37, pp. 6029-6037, 2019.

[18] S. Khanmohammadi, S. Tutun, and Y. Kucuk, "A new multilevel input layer artificial neural network for predicting flight delays at JFK Airport," *Procedia Computer Science*, vol. 95, pp. 237-244, 2016.

[19] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, pp. 865-873, 2014.

[20] L. Ding, W. Fang, H. Luo, P. E. Love, B. Zhong, and X. Ouyang, "A deep hybrid learning model to detect unsafe behavior: Integrating convolution neural networks and long short-term memory," *Automation in Construction*, vol. 86, pp. 118-124, 2018.

[21] J. Kim, H. Ji, S. Oh, S. Hwang, E. Park, and A.P. del Pobil, "A deep hybrid learning model for customer repurchase behavior," *Journal of Retailing and Consumer Services*, vol. 59, p. 102381, 2021.

[22] A. Bhattacharya. "Radiographic Image Analysis." https://aditya-bhattacharya.net/2020/07/27/radiographic-image-analysis/2/

[23] United States Department of Transportation. "Bureau of Transportation Statistics." https://www.transtats.bts.gov/

[24] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357, 2002.

[25] T. Akiba, S. Sano, T. Yanase, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.

[26] F. Chollet. "Keras." GitHub.com. https://github.com/fchollet/keras

[27] C. Guo and F. Berkhahn, "Entity embeddings of categorical variables," 2016, arXiv:1604.06737.

[28] F. Krüger, "Activity, context, and plan recognition with computational causal behaviour models," University of Rostock, Rostock, 2018.

[29] Z. Xu, D. Shen, T. Nie, and Y. Kou, "A hybrid sampling algorithm combining M-SMOTE and ENN based on Random forest for medical imbalanced data," *Journal of Biomedical Informatics*, vol. 107, p. 103465, 2020.

[30] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, pp. 221-232, 2016.

Warittorn Cheevachaipimol, photograph and biography not available at the time of publication.

Bhudharhita Teinwan, photograph and biography not available at the time of publication.

Parames Chutima, photograph and biography not available at the time of publication.