A deep feedforward neural network and shallow architectures effectiveness comparison: Flight delays classification perspective

Conference Paper · November 2021

DOI: 10.1145/3505711.3505712

4 authors, including:

Desmond Bala Bisandu

Cranfield University

Salvatore Filippone

Cranfield University

The 5th International Conference on Advances in Artificial Intelligence (ICAAI 2021)
in QAHE at Northumbria University London Campus, UK, Virtual Event
DOI:10.1145/3505711.3505712

A Deep Feedforward Neural Network and Shallow Architectures Effectiveness Comparison: Flight Delays Classification Perspective

Desmond Bala Bisandu[1,2][0000-0002-5723-8429]*, Mohammed Salih Homaid[1,2][0000-0003-2226-4472], Irene Moulitsas[1,2][0000-0003-0947-9495] *, Salvatore Filippone[1,3][0000-0002-5859-7538]

[1]Department of Computational Engineering Sciences, Cranfield University, Mk43 0AL, United Kingdom
[2]Machine Learning and Data Analytics Laboratory, Digital Aviation Research and Technology Centre (DARTeC)
[3]Department of Civil and Computer Engineering, Università di Roma "Tor Vergata",
Via del Politecnico 1, I-00133, Roma, Italy
*Corresponding authors, E-mail addresses: desmond.bisandu@cranfield.ac.uk; i.moulitsas@cranfield.ac.uk

ABSTRACT

Flight delays have negatively impacted the socio-economics state of passengers, airlines and airports, resulting in huge economic losses. Hence, it has become necessary to correctly predict their occurrences in the process of decision-making because it is important for the effective management of the aviation industry. Developing accurate flight delays classification models depends mostly on the air transportation system complexity and the infrastructure available in airports, which may be a region-specific issue. However, no specific prediction or classification model can handle the individual characteristics of all airlines and airports at the same time. Hence, the need to further develop and compare predictive models for aviation decision system of the future cannot be over-emphasized. In this research, flight on-time data records from the United State Bureau of Transportation Statistics was employed to evaluate the performances of Deep Feedforward Neural Network, Neural Network, and Support Vector Machine models on a binary classification problem. The research revealed that different accuracies of flight delay classifications were achieved by the models. The Support Vector Machine had the worst average accuracy than Neural Network and Deep Feedforward Neural Network in the initial experiment. The Deep Feedforward Neural Network outperformed Support Vector Machines and Neural Network with best average percentage accuracies. Going further to investigate the Deep Feedforward Neural Network architecture on different parameters against itself suggest that training a Deep Feedforward Neural Network algorithm, regardless of data training size, the classification accuracy peaks. We examine which number of epochs works best in our flight delay classification settings for the Deep Feedforward Neural Network. Our experiment results demonstrate that having many epochs affects the convergence rate of the model unlike when hidden layers are increased, it does not ensure better or higher accuracy in a binary classification of flight delays. Finally, we recommended further studies on the applicability of Deep Feedforward Neural Network in flight delays prediction with specific case studies of either airlines or airports to check the impact on the performance of the model.

CCS Concepts

- **Computing Methodologies ➡ Machine Learning ➡ Machine Learning Approaches ➡ Neural Networks**

Keywords

Classification, deep learning, deep neural network, flight delays, Support Vector Machine

1 INTRODUCTION

Human social life in modern times has increasingly become dependent on the aviation industry and accurate classification of flight delays is profitable but difficult due to uncertainties and failures that might occur within the processes. Despite the possibility of using land or sea transport for travels, the efficiency and reliability of air transportation have made it a better option for most passengers [1]. Flight delays have increased due to the rapid increase in air traffic and have become a prominent problem in the aviation industry [1]. A report from the Bureau of Transportation Statistics (BTS) states that one out of four flights arrived destination later than 15 minutes [2]. Also, the annual cost of air transportation delays totals over $30 billion and this poses a challenge to the design and development of the Next-generation Air Transportation Decision Systems (NATDS) [3]. This is a motivation for the accurate and practical prediction of flight delays, especially on individual or group basis.

Flight delay propagation is usually caused by the late arrival of a certain flight which may cause the same aircraft to depart late and is beyond the limited airspace capacity delays [4]. Airlines fly their aircraft on a daily schedule itinerary requiring a visit to other airports, aircraft arriving late in the day has a significant impact on the delay performance thereafter [5]. For instance, an aircraft delayed for one hour from the departing airport is certainly going to delay arriving in the next airport; the lateness in arriving will affect the subsequent departure of the aircraft, which will lead to subsequent lateness in aircraft arrivals [6]. The United State (US) National Airspace System (NAS) is subject to delay propagation, which includes but not limited to many resources connected such as the crew, passenger, aircraft and gate space. The slack time between departure and arrival is forced to reduce in the NATSDS due to increasing air traffic demands, making the NAS suffer more delay from the networks. Therefore, modelling flight delays is key to successful and accurate flight delay predictions or classifications.

For several years, flight delays have been one of the major problems costing the airline industry. Several reasons can cause flight delays ranging from weather conditions (strong winds, winter storm, etc.), late-arriving aircraft, NAS (heavy traffic volume, air traffic control), security, and air carrier (aircraft cleaning, baggage loading, etc.) [7]. Additional operation expenses are incurred when a flight delayed, which is mostly due to increased spending on crews, maintenance and increase fuel consumption. This does not only create inconveniences to passengers but also negatively affect the overall economy [8]. Flight delays studies have been important research areas with much attraction which has led to the employment of data analytics and machine learning methods to predict flight delays by the airlines [9]. If airlines could predict flight delays, a proactive decision could be taken to effectively manage disruptions and minimise last-minute flight rescheduling, thus improving travel experiences. Approximately, every year about 20% of airline flights is delayed or cancelled mainly due to several causes [10]. These delays significantly incur a cost to both passengers and airlines. For example, in 2007 the US economy was affected by a cost of $32.9 billion [10]. More than half of that cost accounted for passengers claims to the airlines due to incurred delays.

Several studies exist the attempt to find the most suitable solutions to flight delays using Deep Learning (DL) methods. These methods were applied on datasets ranging from weather-caused to non-weather-caused delays. However, there is no superior or optimal solution to the problem of flight-delay classifications and predictions to date [11]. The use of weather data to predict Beijing Capital International Airport departure capacity was carried out by Xiangmin in [12]. Both Choi et al. in [13] and Belcastro et al. in [14] applied weather and flight data for predicting delays. In [15], weather forecast information of origin and destination airports were utilised for prediction, while in [16] the characteristics of weather such as the direction of the wind, speed rate, snow depth, precipitation and visibility were used for predicting flight delays. The author in [17] proposed a machine learning model focusing on weather-induced delays on individual flights. Binary classification accuracy for a specific individual origin-destination pair can be improved by fine-tuning some of the parameters such as activation function, number of hidden layers and feature selection etc. Overall, most of the previous studies have their limitation in feature selection techniques. The goal of this work is to implement deep learning classifier for a non-weather caused flight delays and compare the performance against other popular shallow architecture machine learning methods.

We can summarise the contribution of this paper in two points. Firstly, we introduce a new algorithm which supports classifying both departure and arrival delays that combines a feature selection method from [18] for a multi-label classification algorithm Deep Feedforward Neural Network (DNN). Based on the BTS flight on-time dataset, we applied a method to be able to select a feature for optimal training of the model for high classification accuracy. Secondly, according to the International Air Transport Association (IATA), one of the key causes of airline delays is the delayed arrival of an aircraft. In our model, we have captured this relationship by considering the knock-on effects. Also, we made a performance comparison for three models DNN and shallow architectures known as Support Vector Machine (SVM) and one-layer Neural Network (NN) on a flight delay classification task.

The rest of the paper is organised as follows. Section 2 discusses related work and main concepts. Section 3 discusses the methods implemented. Section 4 presents a discussion on data, performance metric for evaluating and assessing the model quality. Finally, Section 5 concludes and presents future direction.

2 RELATED WORK

Flight delays have been causing numerous problems for both passengers and the aviation industry [7]. Researchers have used collective efforts in the quest to address this problem by providing solutions through experimental findings [17]. Despite the huge successes recorded in addressing this problem, there is no single optimal solution. Therefore, there is need for more research and investigation of the issue [11]. The authors in [19] developed a methodology for flight schedules. A clustering approach was employed to group airports into clusters and these clusters were used as input into the flight scheduling system. However, the author suggests that the system can be improved using more sophisticated methods in the future. Fei in [20] performed a factor analysis of factual data from irregular flights which combine a Gaussian Mixture Model and Expectation Maximisation (GMM-EM) algorithm. The model was tested on the Air Traffic Management Bureau (ATMB). It was discovered that the upstream delay has a positive effect on the quality of downstream delay prediction. The authors in [21] applied Deep Recurrent Neural Network (DRNN) in discussing the architecture of Long Short-Term Memory (LSTM) and RNN networks. They explored the benefit of stacking the neural networks and ways of making the architectures deeper using RNN. The use of deep LSTM-RNN architecture improves the accuracy of the prediction model. Also, Kai-Quan Cai in [22] investigated the problem of multi-objective air traffic network flow optimisation (MATNFO). A method for systematic searches of flight routes and time of departure-arrival was proposed. A sequential time slot for flight routes searched to avoid computational overhead was implemented. The result of the experiment shows that their approach performs better than SVM.

In [17], the authors developed a model with two phases of prediction as follows: (i) Delay on departure; and (ii) Delay on arrival. This was for effective prediction of departure and arrival delays using a schedule of flight and weather forecast features. A binary classification to forecast delay occurrence and regression to find delay value in minutes were implemented. Gradient Boosting Classifier performed best during the classification while Extra-Tree regressor performed best in the regression task. The departure delay prediction has a higher error rate than arrival delay prediction. In [23], the author developed a unimodal model for predicting time delays. Neural networks and the binomial model with trees were implemented and compared. The author realises arrival delay and ground operation time to be the most significant indicators in predicting departure delay. In [24], the author implemented Gradient Boost model for flight delay prediction. The highest coefficient of determination of 92.3185% for arrival and 94.8523% for departure was achieved. However, the model was limited to only 70 airports. Also, the author in [25] implemented a Neural Network and Deep Belief Network for flight delay prediction. The accuracies of 92% and 77% were achieved for Neural Network with three hidden layers neuron and Deep Believe Network with four hidden layer neurons respectively.

In this work, we conducted a thorough performance comparison of three popular machine learning methods known as DNN, NN and SVM in a flight delay classification problem task. Also, to further investigate the performance of DNN architecture with different parameter settings.

2.1 Problem definition

Let flight F_1 be a tuple $(B_o, B_d, t_{sd}, t_{ad}, t_{sa}, t_{aa})$ where B_o represents the origin airport, B_d represents the destination airport, t_{sd} represent the scheduled departure time, t_{ad} represents the actual departure time, t_{sa} represents the scheduled arrival time to the gate, while t_{aa} represents the actual arrival time to the gate. All date, hours and minutes are included in time. As stated earlier this research is for classification of non-weather induced flight delays consisting of both departure and arrival delays. The classification takes into consideration of information related to both departure and arrival flights (origin airport, destination airport, departure scheduled time and arrival schedule time) according to the schedule. The delay classification of any flight F_1 departing from airport B_o at a time t_{sd} and arriving at an airport B_d, at a time t_{sa} is an estimated arrival delay $AD(F_1)$, while departure delay $DD(F_1)$ is when a flight depart the origin airport later than the scheduled time. If the delay prediction is below certain threshold T_1, then it is classified as an on-time flight. else, it is classified to be delayed flight.

3 METHODS

In this section, we describe the ML methods employed in our research. These are among the most popularly used ML methods in the literature for classification problems [25, 42]. Table 1 shows the parameters and levels used in training the three methods (SVM, NN and DNN) compared in this research.

3.1 Support Vector Machine

Support Vector Machine are belonging to general kernel methods [26, 27, 28], which depend on data only through dot products. A kernel function *n* is used to replace the dot product by computing a high dimensional feature space dot product of the data. It has two major advantages as follows: the ability to generate nonlinear decision with linear classifiers and the ability of the kernel to allow classifiers applied to a no obvious dimensional vector space representation data. In modelling SVM a hyperplane is constructed to form decision boundary creating margin between positive and negative class [30, 31]. Given a training set with $x_i \, \varepsilon \, \Re^d$ of input vectors with corresponding $y_i \, \varepsilon \, \{+1, -1\}$ labels the SVM model learns the classification into two classes.

A two-class problem of flight delays, having a label of 1 (there exist a delay) and 0 (there exist no delay). Let x be a vector with elements x_i in our case the input features containing i^{th} vector in a dataset composing of n labelled examples (x_i, y_i) where y_i is the label and x_i denotes the input variables. A linear classifier is defined as the dot product between two vectors also known as scalar or inner product. Equation (1) defines a linear classifier with a linear discriminant function:

$$f(x) = \mathrm{w}^T x + b \tag{1}$$

where x is the input variable, w the weighted vector and b is the bias. For the case of $b = 0$, this means that for the set of points x_i such that $w^T = 0$ are points that go through the origin and perpendicular to w making a hyperplane which divides the space according to the sign of discriminant function f(x) into two. The decision boundary of the classifier is the boundary region classified as positive or negative. A classifier that has a linear decision boundary is referred to as a linear classifier otherwise if dependent on non-linear data is known as non-linear classifier [31, 32]

The type of kernel function, the degree of kernel function (*d*: polynomial kernel; γ: for the radial base kernel) and regularisation constant *c* forms the parameter of an SVM. To determine efficient parameters we used the approach proposed in [34]. In our experiment, we applied five levels of *d*, ten levels of γ, while both four and five levels of *c* were tested in the parameter settings. We then selected the best polynomial and radial basis of the SVM model for comparison with the NN and DNN models.

3.2 Neural Network

Neural Network is a nonlinear prediction method and one of the popular methods applied in classification problems such as stock index prediction, fraud detection etc. [34, 35] It is referred to as Multilayer perceptron (MLP) or feedforward NN. The dense interconnected layers help the inputs to activate the neurons. In this study, we employ a one-layer feedforward NN. The input for the model is as simple as the same in DNN. We use a single layer output neuron, and a transfer function, known as a log function, which produces a continuous output value between 0 and 1. To determine the class of delay we use a threshold of 0.5. If the output is greater than or equal to 0.5 then this will result in a classification of delay. Using the threshold of 0.5 was found to be the best in producing the most accurate classification results. Additionally, in line with the work presented in [33, 34, 35, 36], a sigmoid function was employed as a transfer function for the hidden layers because it was the setting with the most accurate classification results in previous research. At each epoch, Gradient Descent was used to adjust the weights for attaining a global minimum.

The NN has several numbers of hidden layers neurons (*n*), the learning rate value (*lr*), the number of epochs (*ep*), and the momentum constant (*mc*). To determine the most appropriate parameters we tested ten different values of *n*, nine different values of *mc* and ten different values of *ep*. Then, we selected the best result for comparison with the DNN and SVM. We found that the learning rate *lr* for the top-performing settings in the model was within the interval [0.1, 0.2].

3.3 Deep Feedforward Neural Network

A Deep Feedforward Neural Network is a network consisting of at least an input layer, two or more hidden layers and an output layer. Several functions are attached to DNNs [29, 40]For instance, we might have five functions $f^{(1)}$, $f^{(2)}$, $f^{(3)}$, $f^{(4)}$ and $f^{(5)}$ all in a chain connection forming a function f(x) ≈ ($f^{(5)}$, $f^{(4)}$, $f^{(3)}$, $f^{(2)}$, $f^{(1)}$(x)). While in this case $f^{(1)}$ is known as the first layer, $f^{(2)}$ is known as the second layer, and it continues until the final layer which is called the output layer. The output layer is the same for either classification or regression task matching the output space. The depth of the model is given by the overall length of the network of the chain. The term deep learning is from this concept. We employ classical feedforward architecture as shown in Equation (2). The neurons in previous layers that are fully connected with all neurons in 1 + *l* subsequent layer with a weighted directed edge. Also, a bias unit connected to all non-output layers of the network, serve as subsequent layers activation threshold. Each weighted combination of **γ** of the n_l outputs of the neuron in the previous layer *l* are received as input by each neuron.

$$\boldsymbol{\gamma} = \sum_{i=1}^{n_l} x_i w_i + b \tag{2}$$

where w_i represents the weight of the output, x_i and *b* represent the input and bias respectively. The activation function $\boldsymbol{a}$ transforms the weighted combination **γ** so that output signal $\boldsymbol{g(\gamma)}$ is relayed on subsequent layer neuron which can be represented as in equation (3).

$$a = g(\gamma) = g\left(\sum_{i=1}^{n_l} x_i w_i + b\ \right) \tag{3}$$

where *a* is the activation function, $\boldsymbol{g(\gamma)}$ output signal, w_i are the weights and *b* is the bias that we adjusted during the training process of the network manually based on some rules until an optimal result is obtained. Following the work proposed in [42], we used sigmoid activation function on all the other layers except the output layer where a softmax activation function was applied to constrict real value into [0, 1] based on our target output. These activation functions can be represented as in Equation (4) and (5) respectively.

$$a = g(\gamma) = \frac{1}{1+e^{-\gamma}} \tag{4}$$

$$a = g(\gamma) = \frac{e^{\gamma}}{\sum_{l=0}^{k} e^{\gamma_l}} \tag{5}$$

where $e^{-\gamma}$, e^{γ} and e^{γ_l} are the exponential decay of input values and *k, l* are the initial value and terminating value during training. But sigmoid activation function is a differentiable function as shown in equation (6) where $g'^{(\gamma)}$ are the decomposed function and all other parameters maintain their usual meanings.

$$a' = g'^{(\gamma)} = \frac{e^{-\gamma}}{(1+e^{-\gamma})^2} = \frac{1}{1+e^{-\gamma}}\frac{e^{-\gamma}}{(1+e^{-\gamma})} = \frac{1}{1+e^{-\gamma}}\left(1-\frac{e^{-\gamma}}{(1+e^{-\gamma})}\right) = g(\gamma)\big(1-g(\gamma)\big) \tag{6}$$

Let *W* be the collection of $\cup_{l=1}^{L-1} w_l$, where w_l is the weighted matrix that connects two consecutive layers *l* and *l + 1* for a network with L number of layers. Similarly, assume *B* to be the collection $\cup_{l=1}^{L-1} b_l$ with c_1 column vector of biases of *l* layer. The output of the entire DNN network is fully determined by the collection of *W* and *B* as shown in Equation (7). To minimize the error in training on the data, we adapt the weight accordingly. Specifically, the objective is minimizing some loss function L (W, B|j) for *j* set of training examples. We are solving a classification problem related to flight delays and the cross-entropy loss function in Equation (7) is applied.

$$L\ (W, B|j) = -\sum_{y\in 0}(ln\left(\left(o_y^{(j)}\right)t_y^{(j)} + ln\left(1-o_y^{(j)}\right)\left(1-t_y^{(j)}\right)\right) \tag{7}$$

where the output units are represented by y, *o* represents the output layers, $o_y^{(j)}$ is the predicted class, and $t_y^{(j)}$ is the actual output class, for instance, j. We used Stochastic Gradient Descent (SGD) in minimising the loss function. Also, we employed Backward Propagation to calculate the Gradient loss function L (W, B|j). Since the input variables are simple (eg. seven variables), we used a straightforward architecture of deep feedforward neural network as shown in Figure 1, where the input layer, hidden layers and output layer are shown. As seen from Figure 1, the interconnections between hidden neurons can be complicated during training because of the random value generation for the weight and bias at each training epoch. This is one of the interesting aspects of dealing with deep neural networks.

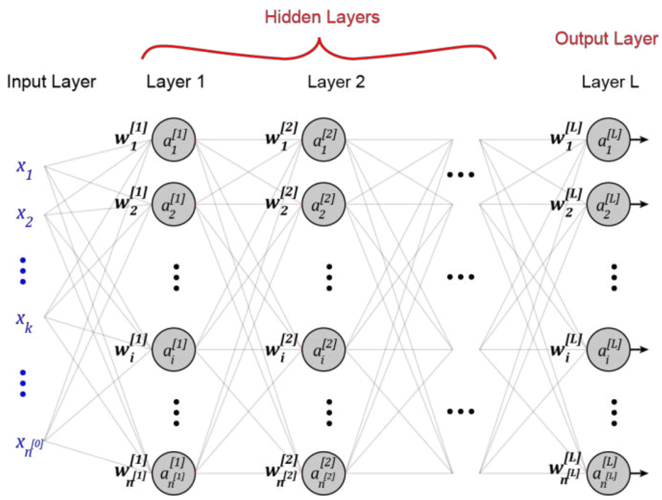

Figure 1: Architecture of Deep Feedforward Neural Network [43]

Table 1 shows the parameter settings used for each method during the model training. These parameters are specific to each of the architecture of the model and are fine-tuned to optimise the expected output of the models. This is important in every machine learning task, because the parameters have the overall control of any model before applying a special optimisation algorithm if needed. As shown in Table 1 for the three machine learning methods SVM, NN and DNN implemented in this paper, they all have their architectural characteristics for both the linear learner (SVM) as well as the nonlinear learners (NN, DNN). DNN is an extension of the NN in trying to improve the learnability of the model with more layers and fine-tuning parameters.

Table 1: Parameter settings for SVM, NN and DNN

S/No.	Method	Levels
1	SVM	polynomials
	Kernel function degree *(d)*	1, 2, 3, …, 5
	Kernel function Gamma (γ)	0, 0.1, 0.2, …, 5.0
	Regularization *(c)*	1, 10, 50, 100
2	NN	
	Epochs *(ep)*	1, 2, 3, …, 10
	Learning rate (*lr*)	0.1, 0.2
	Momentum constant *(mc)*	0.1, 0.2, 0.3, …, 0.9

	Number of neurons *(n)*	10, 20, 30, …, 100
3	DNN	
	Epochs *(ep)*	1, 2, 3, …, 10
	Learning rate (*lr)*	0.1, 0.2
	Momentum constant *(mc)*	0.1, 0.2, 0.3, …, 0.9
	Size of Mini Batch	5, 6, 7, …, 15
	Hidden layers sizes	2, 3, 4, …, 10
	Activation functions	Sigmoid/SoftMax
	Number of neurons *(n)*	10, 20, 30, …, 100

4 DISCUSSION OF RESULTS

In this paper, we evaluated the classification of individual flight delays for each one of the methods discuss earlier. The performance of SVM, NN and DNN was measured in the experiment in classifying flight delays to find out which of the methods perform best in our case. We also explored the differences in data handling of these methods during model training. To find out this we conduct a binary classification problem using the three methods with the target of classifying individual flight delays into a class of all flights that depart or arrive 10 minutes after the scheduled period to be considered delayed, otherwise not delayed.

4.1 Experimental setup

We conducted repeated experiments to clarify the factors influencing the predictability of the simple ML classifiers (SVM, NN). Figure 2 shows the steps in the general experiment. The implementation was performed with the Python programming language and relevant libraries embedded in Jupyter Notebook version 6.0.1 [40] on a Graphics Processing Unit (GPU). Detailed information about the software environment is provided in Table 2.

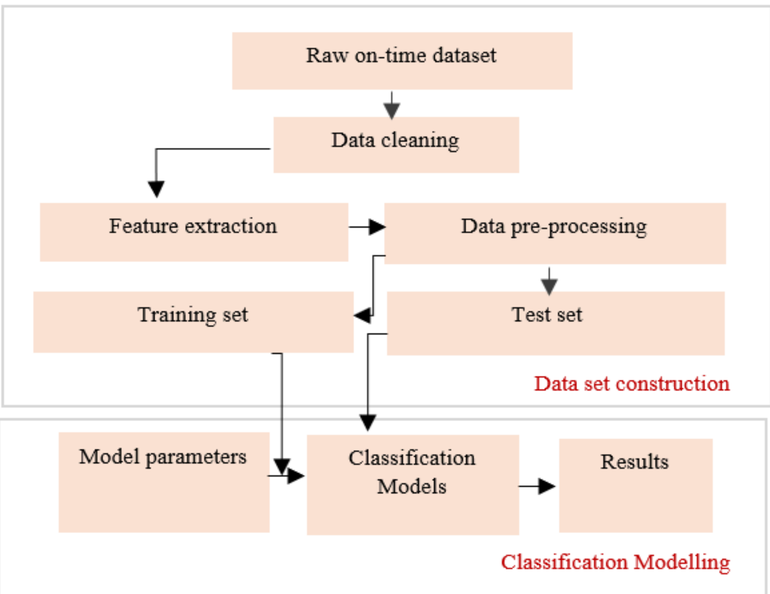

Figure 2: An overview of the experiment steps

Table 2: Software environment information

S/No	Name of library	Version
1	keras	2.3.1

2	numpy	1.19.5
3	pandas	0.25.1
4	itertools	0.25.1
5	matplotlib	3.1.1
6	sklearns	0.22.2.post1
7	tensorboard	2.1.1
8	tensorflow	2.1.0
9	tensorflow-gpu	2.0.0
10	ipython	7.8.0

4.1.1 Dataset Description

Table 3 contains the factors considered to influence the predictability of the flight delay classification. Real-world dataset for flight on-time data from the US BTS (https://www.transtats.bts.gov/) was used for the three models evaluated in this research. The flight on-time dataset used covers a time span from January to December 2018. More than 100 arrival and departure airports are covered in the data with more than 150,000 training samples. The weather feature was not considered because of its low impact on predictive accuracy improvement [13]. A preliminary dataset with a high probability of flight departure and arrival delay was selected. The original dataset contains 21 factors. However, some factors were removed due to their irrelevance in our targeted output. The summary of the employed factors is presented in Table 3.

Table 3: Dataset factors information

S/No.	Factor	Unit	Detail
1	Flight departure and arrival airport	Text	3 letters code eg. LHR
2	Scheduled flight departure and arrival time	Hours and Minutes	In four digits formats eg. 1456
3	Flight number	Numeric	Flights number eg. 1454
4	Actual departure and arrival time	Hours and Minutes	In four digits formats eg. 1456
5	Unique carrier	Numeric	3 letters code eg. PS
6	Scheduled and actual elapse time	Numeric	In two digits formats eg. 94
7	Departure and arrival delay	Numeric	Number of flights eg. 14

4.1.2 Dataset Cleaning

The data may not have a complete record due to message transmission problems. We deleted incomplete data. Also, obvious abnormal data were deleted in the quest to improve the accuracy of the model classification in the training set:

- Duplicate data in the departure and arrival airport.
- Data with very long delay time.

4.1.3 Feature Extraction

The feature extracted for this research is from the factors affecting the classification of flight delays using Algorithm 1. Flight plan formulation is subject to many departments such as airlines and airports, which all parties are interested in, even with their restricted factors. The features are extracted based on their impact on either the departure or arrival flight delay before applying some pre-processing techniques to fine-tune the input dataset for the ML model training.

Algorithm 1: Feature extraction approach	
1:	Procedure:
2:	**begin**
3:	Read data
4:	**for 1** to **N** features **do**
5:	select the best feature subset
6:	apply learning algorithm
7:	evaluate selected feature subset
8:	**end for**
9:	return final selected feature subset
10:	**end**

4.1.4 Dataset Pre-processing

Because SVM works with numerical values and NN/DNN work with both text and numerical values of data input, we had to pre-process the data differently. We define the delay time as the differences between the scheduled time and actual departure time, and in turn, we then calculate median delay time. We consider the delays of scheduled arrival and departure time according to the hour of the day and date of the month. Finally, the model considers the arrival and departure airports.

4.2 Results

We conducted a binary classification on individual flights departure, and arrival on-time record, the comparison of the results of the three ML approaches evaluated for the flight delay classification problem is discussed in this section.

4.2.1 Performance Metrics

To evaluate the quality of our results, we are using a number of metrics. The most significant metric to measure the performance of classification machine learning models is that of *Accuracy* [32, 40]. *Precision* and *Recall* are also important metrics because they help evaluate further the true effectiveness of a model. These metrics use the notions of True positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). More precisely, these metrics are defined as follows:

Accuracy (AC) measures the fraction of predictions that the model classified correctly [44]. Formally, accuracy can be defined as shown in Equation (8):

$$\text{Accuracy } (AC) = \frac{TP+TN}{TP+FP+TN+FN} \tag{8}$$

Precision (P) measures the exactness of a classifier as the truly classified percentage of positives. Is the weighted average of negative and positive precision [44]. Therefore, high precision means less False Positive and a good model. As expressed in equation (9) and equation (10) respectively.

$$\text{Precision}_{\text{positive}}\ (P_p) = \frac{TP}{TP+FP} \tag{9}$$

$$\text{Precision}_{\text{negative}}\ (P_n) = \frac{TN}{TN+FN} \tag{10}$$

Recall (R) measures the completeness of a classifier as the truly classified percentage of actual positives [44]. Is the weighted average of negative and positive recall. Therefore, high recall means less False Negative and a good model. As expressed in equation (11) and equation (12) respectively.

$$\text{Recall}_{\text{positive}}\ (R_p) = \frac{TP}{TP+FN} \tag{11}$$

$$\text{Recall}_{\text{negative}}\ (R_n) = \frac{TN}{TN+FP} \tag{12}$$

4.2.2 Model Evaluation

We implemented and evaluated SVM, NN and DNN in this research to compare their performance on flight delay classification modelling task. The results in terms of the earlier defined metrics are presented in Table 4.

Table 4: Performance comparison of SVM, NN and DNN on flight delays prediction using BTS dataset

S/No.	Model	Precision	Recall	Accuracy
1	SVM	0.6695	0.8619	0.6618
2	NN	0.7206	0.8990	0.7432
3	DNN	0.9070	0.9003	0.9001

In our experiment, the average recall, precision, and accuracy of the SVM are 86.19%, 66.95% and 66.18%. The measures for the NN are 89.90%, 72.06% and 74.32% respectively, while for the DNN the same measures are 90.03%, 90.70%, and 90.01%. Therefore, the DNN model outperformed the SVM and NN for all metrics. SVM as a non-parametric model performs worse than the parametric models NN and DNN.

In our investigation, we also wanted to find out if there are any advantages to having a deep neural network as opposed to other shallow models in flight delay binary classification when using DNN. The experiment shows that there is a huge improvement when having a deep neural network in the results, having at least 10% improvement in the recall, 20% in precision and 20% in accuracy when compared to the NN. This implies that in different classification problems there can be varied results for a method having many parameters to be fitted during training than other shallow methods.

In our study, we went further to investigate the accuracy performance of the DNN based on different parameters level during training, validation and testing. The results of this study are presented Table 5. We investigated the characteristics at different levels of the model building with the same number of features as inputs. Varying the number of hidden layers between 2 to 6 with an epoch of 100 to 500, we observed that when we used 2 hidden layers with 5 neurons and 150 epochs the accuracy was 80.00%. However, interestingly when there was an increase in the number of hidden layers, neurons, epochs, then the testing accuracy increased up to a maximum of 99.82%. Therefore, there were several experiments with different setups to check the ability of classification of the models to perform better. Surprisingly, the accuracy of the experiments was not too different on training, testing and validation with same parameters combination. But increasing the number of hidden layers does not have much significant impact on accuracy. Therefore, we can conclude that when the number of hidden layers increased it does not ensure better or higher accuracy. However, in our experiment, we achieved a 91.00% accuracy classifying flight delays, which is a significant improvement compared to a similar experiment presented in [25] where the accuracy was 76.76% with the use of a Deep Believe Network (DBN) using a sigmoid function. Our accuracy is also better when compared with the results presented in [45]. There are several reasons for this improvement which could be: feature availability, data quality or settings of the parameters of the algorithm. Summarily, even when fewer features are applied to DNN, the results are better compared to the other shallow ML algorithms (SVM, NN) implemented in this paper.

Table 5: Performance comparison of DNN on flight delays prediction using BTS dataset with different settings

S/No	Number of hidden layers	Number of neurons in hidden layers	Epoch	Accuracy		
				Training	Validation	Testing
1	2	5	150	0.8201	0.8221	0.8101
2	6	5	100	0.8267	0.8593	0.8213
3	3	50	150	0.8812	0.8991	0.8798
4	4	25	500	0.9701	0.9832	0.9982
5	5	13	200	0.9629	0.9734	0.9945

We then set out to further investigate the DNN model's performance on the same 500 epochs with different parameters. We observed that the model performance when we used 2 hidden layers with 50 neurons was better in terms of training and validation accuracy with 98.12% and 98.94%, respectively. However, on the testing, the best performance settings are observed when we used 4 hidden layers and 20 neurons with 99.85% accuracy. Finally, we observed that with smaller number of hidden layers and neurons generally the model performance dropped in training, validation and testing.

Table 6: Performance comparison of DNN on flight delays prediction using BTS dataset with different settings and 500 epochs

S/No	Number of hidden layers	Number of neurons in hidden layers	Accuracy		
			Training	Validation	Testing
1	2	5	0.9201	0.9221	0.9011
2	6	5	0.9265	09173	0.9152
3	3	50	0.9812	0.9894	0.9887
4	4	20	0.9610	0.9662	0.9985
5	5	13	0.9412	0.9504	0.9950

We used 500 epochs to train the model for the classification of the flight delays. Figure 3 shows how the DNN model can correctly classify 99.82% of the class of flights that are delayed and 78.42% of the class that are not delayed. This classification contains more than 150,000 flight records and the model correctly classified 128,609 of them in the delay class and 63,436 in the no delay classes respectively which is very good performance for any flight delay decision support system.

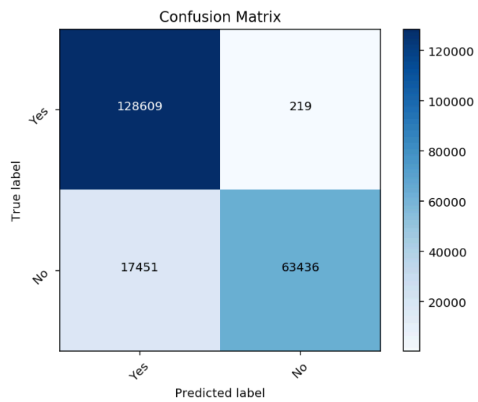

Figure 3: Confusion matrix of testing classification using DNN with an epoch=500

We then reduced the number of epochs to 200 when training the model for the same classification of the flight delays. Figure 4 shows how the DNN model can correctly classify 99.45% of the class of flight that are delayed and 81.54% of the class that are not delayed. This classification contains more than 150,000 flight records and the model correctly classified 128,129 of them in the delay class and 65,958 in the no delay class respectively which is again very good performance for any flight delay decision support system.

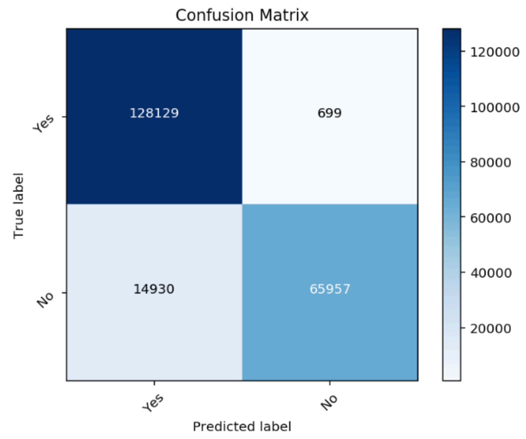

Figure 4: Confusion matrix of testing classification using DNN with an epoch=200

Figure 5 shows the training and validation accuracy of the DNN model. We observed that the model performs very well during both the training and the validation phase, with consistently good performance across every example of the training set, which is an advantage when developing a flight delay decision support system.

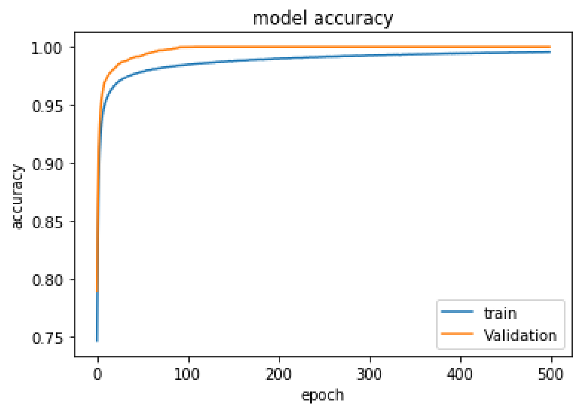

Figure 5: Model accuracy plot for DNN with epoch=500

Figure 6 shows the training and validation loss of the DNN model. We observed that the model again performs very during both the training and the validation phase, with consistently good performance across every example of the training set, which is an advantage when developing a flight delay decision support system.

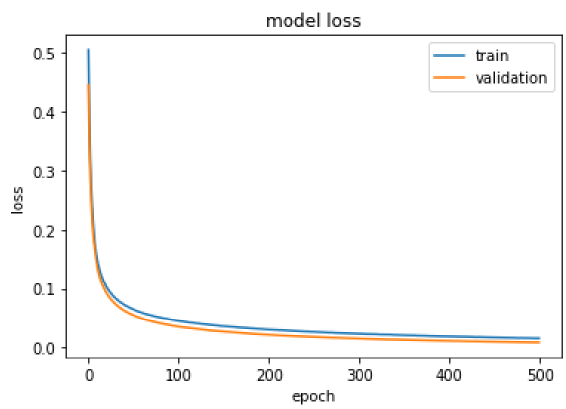

Figure 6: Model loss plot for DNN with epoch=500

Figure 7 shows the final model of the DNN and the architecture with all the elements as applied in the training of the model with 500 epochs.

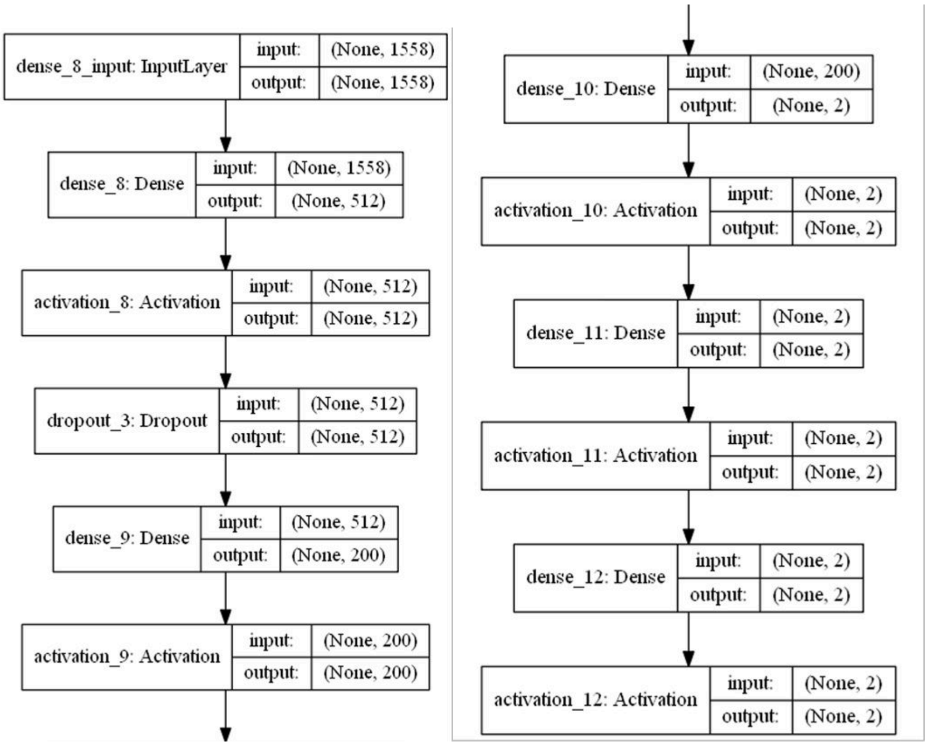

Figure 7: Final DNN Model architecture

5 CONCLUSION AND FUTURE DIRECTION

Major advances of deep learning have been recorded in solving problems that might have been difficult for years. Discovering the intricate structures in high-dimensional datasets made it possible. DNNs have recorded more successes and are predicted to have more robustness in the future of big data because they largely require little engineering by hand. DNN largely take advantage of data and computational ability consequently. Despite this huge success, not many have applied DNN in flight delay binary classification problems. Our research is among the few that has applied DNN and compares with other shallow architectures such as SVM and NN to study their performance in a classification-related problem. We set out to clearly understand if the performance will keep to the track in other experiments by comparing the predictive accuracy of the widely used DNN with other shallow methods at different epochs and parameter tunings. The main aim of our research was to identify the best conditions under which DNN performs better than SVM and NN.

Our experimental results show that the data size has a relative effect on the performance ability of the three classifiers DNN, NN and SVM at different parameter adjustments. However, the classification accuracy of DNN is better when it learns the underlying structure of the datasets. We observed that when the epochs were increased, there was a significant increase in the predictive accuracy. The comparison of the performance of DNN, NN and SVM on classifying delays using the BTS on-time dataset is a topic of continuous research due to the need for more flexible flight delay predictions.

Overall, this paper has contributed to the aviation industry and the air transportation unit specifically in improving passengers experience through better flight delay decision support systems. Empirically, it is evident that most deep learning approaches are more likely to outperform the shallow architectures in a classification task due to their ability to have flexibility in terms of parameters tuning. Based on our results, we believe that DNN can be further investigated in other predictive analysis such as aircraft maintenance. Finally, as evidenced from our experiments, we cannot claim that on its own merit any algorithm is inferior to another. Instead the ability of researchers to understand and tune these algorithms will go a long way in impacting their performance.

ACKNOWLEDGEMENT & FUNDING

The authors wish to thank the reviewers for their constructive feedback. The Petroleum Trust Development Fund (PTDF) Nigeria partially funded this work through grant number PTDF/ED/OSS/PHD/DBB/1558/19 for the first author's PhD studies.

REFERENCES

[1] Y. E. Tan, K. S. Teong, M. Shabbir, L. K. Foo, and S. L. Chua, “Modelling flight delays in the presence of class imbalance,” *ACM Int. Conf. Proceeding Ser.*, pp. 186–191, 2018, doi: 10.1145/3299819.3299847.

[2] European Union Aviation Safety Agency, “Artificial Intelligence Roadmap: A Human-Centric Approach to AI in Aviation,” no. February, pp. 1–33, 2020.

[3] X. Zhu and L. Li, “Improved Flight Time Predictions for Fuel Loading Decisions of Scheduled Flights with a Deep Learning Approach,” no. 11215119, pp. 1–12, 2020.

[4] M. T. Akpinar and M. E. Karabacak, “Data mining applications in civil aviation sector: State-of-art review,” *CEUR Workshop Proc.*, vol. 1852, pp. 18–25, 2017.

[5] R. Boggavarapu, P. Agarwal, and D. H. Rohith Kumar, “Aviation Delay Estimation using Deep Learning,” *2019 4th Int. Conf. Inf. Syst. Comput. Networks, ISCON 2019*, pp. 689–693, 2019, doi: 10.1109/ISCON47742.2019.9036276.

[6] Y. Borse *et al.*, “Flight Delay Prediction System,” *Int. J. Eng. Res. Technol.*, vol. 9, no. 3, pp. 88–92, 2020.

[7] F. Cheng, B. Baszczewski, and J. Gulding, “A hybrid optimization-simulation approach for itinerary generation,” *Proc. - Winter Simul. Conf.*, vol. 2015-Janua, pp. 1885–1896, 2015, doi: 10.1109/WSC.2014.7020036.

[8] Airports Commission, “Economy: Delay Impacts Assessment,” no. November, 2014.

[9] 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 Trans. Veh. Technol.*, vol. 69, no. 1, pp. 140–150, 2020, doi: 10.1109/TVT.2019.2954094.

[10] P. Balakrishna, R. Ganesan, and L. Sherry, “Application of reinforcement learning algorithms for predicting taxi-out times,” *Proc. 8th USA/Europe Air Traffic Manag. Res. Dev. Semin. ATM 2009*, pp. 255–261, 2009.

[11] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[12] G. Xiangmin and M. Li, “Departure capacity prediction for hub airport in thunderstorm based on data mining method,” *Proc. 29th Chinese Control Decis. Conf. CCDC 2017*, pp. 6004–6009, 2017, doi: 10.1109/CCDC.2017.7978246.

[13] S. Choi, Y. J. Kim, S. Briceno, and D. Mavris, “Prediction of weather-induced airline delays based on machine learning algorithms,” in *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, 2016, vol. 2016-Decem, doi: 10.1109/DASC.2016.7777956.

[14] L. Belcastro, F. Marozzo, D. Talia, and P. Trunfio, “Using scalable data mining for predicting flight delays,” *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 1, 2016, doi: 10.1145/2888402.

[15] A. Klein, S. Kavoussi, and R. S. Lee, “Weather forecast accuracy: Study of impact on airport capacity and estimation of avoidable costs,” *Proc. 8th USA/Europe Air Traffic Manag. Res. Dev. Semin. ATM 2009*, pp. 12–21, 2009.

[16] A. D. Evans, B. Sridhar, and D. McNally, “Improving operational acceptability of dynamic weather routes through analysis of commonly used routings,” *16th AIAA Aviat. Technol. Integr. Oper. Conf.*, pp. 1–18, 2016, doi: 10.2514/6.2016-3600.

[17] B. Thiagarajan, L. Srinivasan, A. V. Sharma, D. Sreekanthan, and V. Vijayaraghavan, “A machine learning approach for prediction of on-time performance of flights,” *AIAA/IEEE Digit. Avion. Syst. Conf. - Proc.*, vol. 2017-Septe, 2017, doi: 10.1109/DASC.2017.8102138.

[18] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, “of operators,” vol. 3, no. March, pp. 31–33, 2021, doi: 10.1038/s42256-021-00302-5.

[19] C. Ciruelos *et al.*, “Modelling delay propagation trees for scheduled flights,” *Proc. 11th USA/Europe Air Traffic Manag. Res. Dev. Semin. ATM 2015*, 2015.

[20] F. Bus, “The prediction of flight delays based the analysis of Random flight points,” vol. 00, pp. 3992–3997, 2015.

[21] Y. J. Kim, S. Choi, S. Briceno, and D. Mavris, “A deep learning approach to flight delay prediction,” in *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, 2016, vol. 2016-Decem, pp. 1–6, doi: 10.1109/DASC.2016.7778092.

[22] K. Q. Cai, J. Zhang, M. M. Xiao, K. Tang, and W. B. Du, “Simultaneous optimization of airspace congestion and flight delay in air traffic network flow management,” *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 3072–3082, 2017, doi: 10.1109/TITS.2017.2673247.

[23] H. M. Proenca, R. Klijn, T. Bäck, and M. Van Leeuwen, “Identifying flight delay patterns using diverse subgroup discovery,” *Proc. 2018 IEEE Symp. Ser. Comput. Intell. SSCI 2018*, pp. 60–67, 2019, doi: 10.1109/SSCI.2018.8628933.

[24] 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,” *ICCIDS 2017 - Int. Conf. Comput. Intell. Data Sci. Proc.*, vol. 2018-Janua, pp. 1–5, 2018, doi: 10.1109/ICCIDS.2017.8272656.

[25] V. Venkatesh, A. Arya, P. Agarwal, S. Lakshmi, and S. Balana, “Iterative machine and deep learning approach for aviation delay prediction,” in *2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics, UPCON 2017*, 2017, vol. 2018-January, doi: 10.1109/UPCON.2017.8251111.

[26] Y. Levi, S. Bekhor, and Y. Rosenfeld, “A multi-objective optimization model for urban planning: The case of a very large floating structure,” *Transp. Res. Part C Emerg. Technol.*, vol. 98, no. November 2018, pp. 85–100, 2019, doi: 10.1016/j.trc.2018.11.013.

[27] N. Zeng, H. Qiu, Z. Wang, W. Liu, H. Zhang, and Y. Li, “A new switching-delayed-PSO-based optimized SVM algorithm for diagnosis of Alzheimer’s disease,” *Neurocomputing*, vol. 320, pp. 195–202, 2018, doi: 10.1016/j.neucom.2018.09.001.

[28] W. Wu, K. Cai, Y. Yan, and Y. Li, “An Improved SVM Model for Flight Delay Prediction,” *AIAA/IEEE Digit. Avion. Syst. Conf. - Proc.*, vol. 2019-Septe, 2019, doi: 10.1109/DASC43569.2019.9081611.

[29] S. Lukas, D. Stefani, and P. Widjaja, “Comparing SVM and GLM in calculating insurance premium for flight delay,” *ACM Int. Conf. Proceeding Ser.*, no. September, pp. 141–145, 2019, doi: 10.1145/3369114.3369160.

[30] Y. Xu, L. Liu, X. Gao, and F. F. Zeng, “Analysis of Factors in Flight Delay,” vol. 93, no. Mmsta, pp. 170–176, 2019, doi: 10.2991/mmsta-19.2019.36.

[31] J. C. Moreano and N. B. L. S. Palomino, “Global facial recognition using gabor wavelet, support vector machines and 3d face models,” *J. Adv. Inf. Technol.*, vol. 11, no. 3, pp. 143–148, 2020, doi: 10.12720/jait.11.3.143-148.

[32] S. Ben Taieb and R. J. Hyndman, “Recursive and direct multi-step forecasting: the best of both worlds,” *Int. J. Forecast.*, no. September, 2014.

[33] M. Kadhum, S. Manaseer, and A. L. A. Dalhoum, “Evaluation feature selection technique on classification by using evolutionary ELM wrapper method with features priorities,” *J. Adv. Inf. Technol.*, vol. 12, no. 1, pp. 21–28, 2021, doi: 10.12720/jait.12.1.21-28.

[34] A. Neuron *et al.*, *and Machine Learning – ICANN 2018*. 2018.

[35] L. O. Orimoloye, M. C. Sung, T. Ma, and J. E. V. Johnson, “Comparing the effectiveness of deep feedforward neural networks and shallow architectures for predicting stock price indices,” *Expert Syst. Appl.*, vol. 139, p. 112828, 2020, doi: 10.1016/j.eswa.2019.112828.

[36] N. N. Nguyen and A. T. Duong, “Comparison of two main approaches for handling imbalanced data in churn prediction problem,” *J. Adv. Inf. Technol.*, vol. 12, no. 1, pp. 29–35, 2020, doi: 10.12720/jait.12.1.29-35.

[37] J. Schmidhuber, “Deep Learning in neural networks: An overview,” *Neural Networks*, vol. 61, pp. 85–117, 2015, doi: 10.1016/j.neunet.2014.09.003.

[38] S. van Cranenburgh and A. Alwosheel, “An artificial neural network based approach to investigate travellers’ decision rules,” *Transp. Res. Part C Emerg. Technol.*, vol. 98, no. November 2018, pp. 152–166, 2019, doi: 10.1016/j.trc.2018.11.014.

[39] F. Orsini, M. Gastaldi, L. Mantecchini, and R. Rossi, “Neural networks trained with WiFi traces to predict airport passenger behavior,” *MT-ITS 2019 - 6th Int. Conf. Model. Technol. Intell. Transp. Syst.*, pp. 1–7, 2019, doi: 10.1109/MTITS.2019.8883365.

[40] P. Koprinkova-Hristova, V. Mladenov, and N. K. Kasabov, *Artificial Neural Networks - Methods and Applications in Bio-/Neuroinformatics: Preface*. 2015.
[41] S. Bunrit, N. Kerdprasop, and K. Kerdprasop, "Improving the representation of cnn based features by autoencoder for a task of construction material image classification," *J. Adv. Inf. Technol.*, vol. 11, no. 4, pp. 192–199, 2020, doi: 10.12720/jait.11.4.192-199.
[42] B. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks.Imagenet classification with deep convolutional neural networks.," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2012.
[43] "An Introduction to Deep Feedforward Neural Networks _ by Reza Bagheri _ Towards Data Science." .
[44] L. M. Zintgraf, T. S. Cohen, T. Adel, M. Welling, and V. U. Brussel, "Flight delay error analysis for amsterdam airport schiphol using a deep neural network (dnn)," no. January, pp. 1–12, 2017.
[45] C. Hu, H. Wei, and M. C. B, *Enhancing Airlines Delay Prediction by Implementing Classification Based Deep Learning Algorithms*, vol. 1, no. 1. Springer International Publishing, 2019.