FLIGHT MANEUVER CLASSIFICATION USING ARTIFICIAL NEURAL NETWORKS

YAPAY SİNİR AĞLARI İLE UÇUŞ MANEVRASI SINIFLANDIRMA

FİKRİCAN PUSAT

PROF. DR. MEHMET ÖNDER EFE
Supervisor

Submitted to
Graduate School of Science and Engineering of Hacettepe University
as a Partial Fulfillment to the Requirements
for the Award of the Degree of Master of Science
in Computer Engineering

December 2022

ABSTRACT

FLIGHT MANEUVER CLASSIFICATION USING ARTIFICIAL NEURAL NETWORKS

Fikrican Pusat
Master of Science, Computer Engineering
Supervisor: Prof. Dr. Mehmet Önder Efe
December 2022, 81 pages

There are globally known basic flight maneuvers performed by fighter and aerobatics pilots with agile aircraft. These basic flight maneuvers have strict rules that the pilots should follow to complete the maneuver and can be used to evaluate the pilot's or aircraft's capabilities. Flight data is classified into flight maneuvers by aviation professionals before evaluation. The need for this classification created a research area to be filled: Automatic flight maneuver classification. This study proposes a solution to the automatic flight maneuver classification problem by exploiting artificial neural networks. It also contributes a flight maneuver classification dataset to the literature. This dataset was generated using professional flight simulation tools. The flight data attributes were evaluated and selected to give the optimum performance in terms of accuracy, precision, and recall. The types of artificial neural networks used and compared were single hidden layer neural networks, deep neural networks, and recurrent neural networks. Combinations of these types, activation functions, optimization methods, and gradient descent algorithms were tested against the problem to maximize the performance of the solution. There are no secondary studies on the subject of flight maneuver classification and this study also fills this gap by contributing a systematic literature review on "flight maneuver classification using machine learning methods". The solution proposed by this study successfully classified ten distinct flight maneuvers such as basic descent and ascent, but also more complex ones such as Immelman, split-s, and lazy-eight. The results were evaluated by calculating accuracy, precision, recall, and loss parameters on test data. The best-performing artificial neural network type, which is deep neural networks, gave over 96 percent accuracy and less than 0.1 loss in the test set. All the artificial neural network types and solutions gave over 90 percent accuracy with correctly chosen attributes. This study also contributed a software program that classifies the flight maneuver in real-time while the flight maneuver is being performed in a simulation. It was seen that this software was also accurately predicting the pilots' intended maneuvers in real-time.

Keywords: neural network, flight maneuver, classification, identification

ÖZET

YAPAY SİNİR AĞLARI İLE UÇUŞ MANEVRASI SINIFLANDIRMA

Fikrican Pusat
Yüksek Lisans, Bilgisayar Mühendisliği
Danışman: Prof. Dr. Mehmet Önder Efe
Eş Danışman: Doç. Dr. Adı Soyadı
Eylül 2022, 81 sayfa

Çevik uçaklarla savaş ve akrobasi pilotları tarafından gerçekleştirilen dünyaca bilinen temel uçuş manevraları vardır. Bu temel uçuş manevraları, pilotların manevrayı tamamlamak için uyması gereken katı kurallara sahiptir ve pilotun veya uçağın yeteneklerini değerlendirmek için kullanılabilir. Uçuş verileri, değerlendirilmeden önce havacılık uzmanları tarafından uçuş manevraları olarak sınıflandırılır. Bu sınıflandırma ihtiyacı, doldurulması gereken bir araştırma alanı yaratmıştır: Otomatik uçuş manevrası sınıflandırması. Bu çalışma, otomatik uçuş manevrası sınıflandırma problemine yapay sinir ağlarından yararlanarak bir çözüm önermektedir. Aynı zamanda literatüre bir uçuş manevrası sınıflandırma veri seti sağlamaktadır. Bu veri seti, profesyonel uçuş simülasyon araçları kullanılarak oluşturulmuştur. Uçuş verisi öznitelikleri, doğruluk, kesinlik ve geri çağırma açısından optimum performansı verecek şekilde değerlendirildi ve seçildi. Kullanılan ve karşılaştırılan yapay sinir ağlarının türleri, tek gizli katmanlı sinir ağları, derin sinir ağları ve tekrarlayan sinir ağlarıdır. Çözümün performansını en üst düzeye çıkarmak için bu türlerin kombinasyonları, aktivasyon fonksiyonları, optimizasyon yöntemleri ve gradyan iniş algoritmaları probleme karşı test edilmiştir. Uçuş manevrası sınıflandırması konusunda literatürde ikincil bir çalışma bulunmamaktadır ve bu çalışma aynı zamanda ”makine öğrenme yöntemleri kullanılarak uçuş manevrası sınıflandırması" konusunda sistematik bir literatür taraması içererek bu boşluğu doldurmaktadır. Bu çalışma tarafından önerilen çözüm, temel yükselme ve alçalma gibi on farklı uçuş manevrasını ve aynı zamanda Immelman, split-s ve lazy-eight gibi daha karmaşık olanları da başarıyla sınıflandırmıştır. Sonuçlar, test verileri üzerinde doğruluk, kesinlik, geri çağırma ve kayıp parametreleri hesaplanarak değerlendirilmiştir. En iyi performans gösteren yapay sinir ağı türü olan derin sinir ağları, test setinde yüzde 96'nın üzerinde doğruluk ve 0,1'den az kayıp vermiştir. Tüm yapay sinir ağı türleri ve çözümleri, doğru seçilmiş veri seti elemanlarıyla yüzde 90'ın üzerinde doğruluk sağlamıştır. Bu çalışma aynı zamanda, uçuş manevrası bir simülasyonda yapılırken, uçuş manevrasını gerçek zamanlı olarak sınıflandıran bir yazılım programını da literatüre sağlamaktadır. Bu yazılımın pilotların yapmak istedikleri manevraları da gerçek zamanlı olarak doğru bir şekilde tahmin ettiği görülmüştür.

Keywords: sinir ağları, uçuş manevrası, sınıflandırma, tanılama

ACKNOWLEDGEMENTS

Many thanks to Hacettepe University to adapt to the tough conditions of a pandemic throughout my time here and for embracing me. I would like to express my deepest appreciation to my advisor, Prof. Dr. Mehmet Önder Efe, for guiding me through my master's degree. He was always ready to help when I needed it.

I would like to pay my respect and gratitude to my family whom I deeply owe any success I have if I have any. My deepest love to my mother Nuriye Pusat, my father Fatih Pusat, my sister/flatmate Deniz Pusat, and my lovely wife Elvin Serim Pusat.

I had the pleasure of studying with dear Mustafa Can Güden, Olgu Erten, Niyazi Toker, Mert İlgüy, Nuri Koyuncu, Kerem Maden, Burhan Sünbül, and Burak Öztürk throughout my academic life. I would like to say thanks to each.

I would be remiss in not mentioning my pet cat Zeydan who never missed a chance to sit on my lecture notes. This endeavor would not have been possible without her watching me study for hours with a judgemental attitude.

CONTENTS

TABLES

FIGURES

ABBREVIATIONS

Symbols

ρ	:	Pearson Correlation
τ	:	Kendall Correlation
η	:	Learning rate
θ	:	Network parameters
φ	:	Activation function

Abbreviations

ANN	:	**A**rtificial **N**eural **N**etwork
ANOVA	:	**AN**alysis **O**f **VA**riance
SHLNN	:	**S**ingle **H**idden **L**ayer **N**eural **N**etwork
DNN	:	**D**eep **N**eural **N**etwork
RNN	:	**R**ecurrent **N**eural **N**etwork
MLP	:	**M**ulti **L**ayer **P**erceptrons
SLR	:	**S**ystematic **L**iterature **R**eview
DBSCAN	:	**D**ensity **B**ased **S**patial **C**lustering of **A**pplications with **N**oise
VVI	:	**V**ertical **V**elocity **I**ndicator
FTRL	:	**F**ollow **T**he **R**egularized **L**eader
SELU	:	**S**caled **E**xponential **L**inear **U**nit
ELU	:	**E**xponential **L**inear **U**nit
SGD	:	**S**tochastic **G**radient **D**escent
KL	:	**K**ullback-**L**eibler
ReLU	:	**Re**ctified **L**inear **U**nit
LSTM	:	**L**ong **S**hort-**T**erm **M**emory
SCCE	:	**S**parse **C**ategorical **C**ross-**E**ntropy
MSE	:	**M**ean **S**quared **E**rror

1. INTRODUCTION

Basic flight maneuvers have strict rules while performing them. The pilot strictly follows some aircraft data (angles, speeds, pressures, and so forth) to be in foreseen ranges to complete the maneuver. Completing these maneuvers is almost a mechanical process. Then, an automatic solution can help distinguish different maneuvers to help the pilots perform better, and the avionics professionals analyze flight data easier. This solution would take a burden from the avionic professionals manually classifying the aircraft data to different maneuvers, and remove the possibility of human error from the classification process. To achieve this solution, researchers developed methods to automatically classify flight maneuvers such as flight regime algorithms, data thresholds, and machine learning methods...

According to Jesan (2004), artificial neural networks are good algorithms for pattern recognition problems since they are error-prone, and can evolve for different recognition problems [1]. The problem at hand is a pattern recognition problem since it is simply trying to find some patterns of attributes while performing specific maneuvers, so it seems logical to apply back-propagation artificial networks for the problem at hand. Also, we have numerical input and categorical output, that is, our output is the selection of flight maneuvers from a finite set (classification).

There are studies that try to solve this classification problem using aerodynamics in the literature. For example, in a similar attempt to solve the problem of maneuver classification, Travert (2009) considered neural networks as advanced and not necessary to be considered in his paper. He, instead, used commercial flight regime recognition algorithms [2]. In another attempt at helicopter maneuvers, Barndt et al. (2007) tried to establish criteria and thresholds for determining flight maneuvers. They used the flight test database of The Navy [3].

Machine learning concepts are also used in numerous studies to solve the flight maneuver classification problem. The most utilized methods are density-based spatial clustering of applications with noise (DBSCAN), k-means, and Bayesian networks. DBSCAN and k-means clustering are together used by Blanks, Sedgwick, Bone, and Mayerchak (2017), and Dang, Tran, Alam, and Duong (2003) [4, 5]. Socha et al. (2018) used DBSCAN alone while Wang, Han, Hu, and Zhan (2019) used k-means clustering alone [6, 7]. Bayesian networks are mentioned as a solution to the problem by Wu et al. (2018), Meng et al. (2019), and Chen et al. (2019) [8–10]. Dynamic time warping is also a utilized machine learning method to classify flight maneuvers. It is used by Ruotsalainen et al. (2009), and Wei et al. (2020) [11, 12].

The concept of neural networks got more popular and easier to grasp with tools and special libraries over time, and they are a better choice now than ever. While they do not solely focus on artificial neural networks (ANN), there are some studies that show ANNs can be used for flight maneuvers and regime classification. Pechaud and Kim (2001), needed maneuver classification for determining in-flight loads in maneuvers. They used artificial neural networks for their applications. They obtained data through real flights with air data instruments attached to a Cessna 172P Aircraft and successfully classified maneuvers [13]. Oza et al. (2003) use multilayer perceptrons with one hidden layer and radial basis function networks for the problem at hand using real helicopter flight data. They evaluated their work by implementing cross-validation and confusion matrices and saw that they get high accuracy [14]. Bodin (2020), used logistic regression, support vector machines, and ANNs to classify flight maneuvers using professional simulation flight test data. She compared and evaluated these methods in her study by using recall, accuracy, precision, and confusion matrices. The study shows that artificial neural networks are the best algorithm among the others [15].

These studies that try to solve the maneuver classification problem are explained further in the Related Work (3.) section. The studies utilizing ANNs to solve the flight maneuver classification problem show promising results while not exploring the ANNs in depth. This study aims to fill the gap by testing and comparing numerous ANN structures and optimization methods against the classification problem with a multitude of classes and provide the first thorough analysis in the literature on this subject.

1.1. Scope Of The Thesis

This thesis focuses on providing a means to automatically classify flight data into multiple flight maneuvers by exploiting artificial neural networks. To achieve this goal, this thesis answers the following questions:

- Can artificial neural networks be used to classify flight data into flight maneuvers?
- Which artificial neural network types and optimization methods are best suited for this purpose?
- How do these different types and methods perform against the problem?

To answer the above questions, this thesis explores the below concepts:

- Creating a flight environment to collect flight data using simulation tools.
- Exploring flight data attributes best suited for flight maneuver classification.
- Determining flight maneuvers to classify.
- Exploring and inventing data feature evaluation and selection methods.
- Exploring and comparing neural network types for flight maneuver classification.
- Exploring optimization methods, functions, and algorithms of said neural network types.

1.2. Contributions

This thesis provides a solution to the automatic flight maneuver problem by contributing below items to the literature:

- It includes the first secondary study, a systematic literature review, on flight maneuver classification with machine learning methods.

- A flight simulation environment to collect flight data with a computer program collecting the dataset in real-time is introduced.
- An open-source flight maneuver dataset is contributed.
- It evaluates existing and invents new data feature selection methods to reveal the flight data feature importance in terms of flight maneuver classification.
- It covers the first thorough analysis of using and comparing different types of artificial neural network types for the purpose of flight maneuver classification in the literature. It explores and compares SHLNNs, DNNs, and RNNs on the said subject.
- It contributes an open-source highly accurate real-time flight maneuver classifier program.
- The results of the study show that flight maneuver classification can be done with high performance using artificial neural networks.

All the contributions introduced in this thesis are shared online [16] for reproducibility's sake. Any researcher can easily reproduce and improve the results of this thesis simply by using the source code, the dataset, and the documentation.

1.3. Organization

The organization of the thesis is as follows:

- Chapter 1 presents our motivation, contributions, and the scope of the thesis.
- Chapter 2 provides background information on:

  The data preprocessing methods which are absolute scaling and random shuffling.

  The feature selection methods which are Pearson correlation, Kendall correlation, and Spearman correlation as well as analysis of variance.

The information on ANNs starts from the neurons, and continues with perceptrons, activation functions, classification methods, different types of ANNs, loss functions, and finally the back-propagation and optimization methods.

The classification performance metrics.

- Chapter 3 includes the methodology and results of the systematic literature review on ”flight maneuver classification using machine learning methods”. The SLR was conducted using “Guidelines for performing Systematic Literature Reviews in Software Engineering” by Kitchenham and Stuart (2007) [17]. A total of 16 primary studies were found using specified research methodology. They were evaluated and explained in light of carefully crafted research questions.

- Chapter 4 explains how the information shown in Chapter 2 is implemented to achieve the flight maneuver classification goal step by step. This chapter starts with explaining the data collection method developed solely by this thesis, and continues with how the dataset is utilized with preprocessing, feature selection, and ANN methods.

- Chapter 5 demonstrates the results of the thesis. The performance of the mentioned ANNs with mentioned optimization methods can be seen in this chapter in the form of explained performance metrics.

- Finally, Chapter 6 states the summary of the thesis and possible future directions.

2. BACKGROUND OVERVIEW

2.1. Data Preprocessing

The dataset consists of different data features with different limits and units. Biases may occur when these kinds of data are used to train a neural network, such as some features dominate the output classification due to their scales being different from others. This is why the dataset is normalized between [-1, 1] using maximum absolute scaling before being used in training. This normalization method is given as Equation 1.

$$x_{scaled} = \frac{x}{\max(|x|)} \tag{1}$$

The next preprocessing is only applied when training without recurrent neural networks. The dataset is originally ordered by time in chronological order. The maneuvers are grouped together in the dataset. The lines in the dataset are randomly shuffled before being used in training with a single hidden layer and deep neural networks to reduce over-fitting caused by this grouping. Shuffling also helps to create a balanced training and validation split. Shuffling happens before normalization.

2.2. Feature Selection

The use of the correlation calculation provides both removing redundant features and giving an idea of which features are the most relevant to the categorical output when used with numerical input categorical output datasets.

The Pearson correlation was first pitched by Karl Pearson (1895). It can be calculated using Equation 2. The $\rho_{x,y}$ represents the correlation between two random variables x and y, $cov(x, y)$ represents the covariance between them, and finally, s_x and s_y represents the standard deviation of the random variables [18].

$$\rho_{x,y} = \frac{\mathrm{cov}(x, y)}{s_x s_y} \tag{2}$$

The Spearman correlation, first introduced by Spearman (1904), is defined as the Pearson correlation of the ranks of the random variables. It can be calculated using Equation 3 [19]. The R represents the rank of a random variable.

$$\rho_{R(x),R(y)} = \frac{\mathrm{cov}(R(x)R(y))}{s_{R(x)} s_{R(y)}} \tag{3}$$

The Kendall correlation, first introduced by Kendall (1938), can be calculated using Equation 4 [20]. The x_i represents the i^{th} observation of the random variable X.

$$\tau = \frac{2}{n(n-1)} \sum_{i<j} \mathrm{sgn}(x_i, x_j)\mathrm{sgn}(y_i, y_j) \tag{4}$$

Finally, the main feature selection method is ANOVA. Fisher (1921) pitched the ANOVA correlation [21] and it became a widely excepted data mean and variance analysis since then. The explanation of the ANOVA calculation is not in the scope of the thesis and it was calculated using an open-source library.

2.3. Artificial Neural Networks

The neurons are at the heart of artificial neural networks. The idea of simulating the neurons of the living brain as a computational processing unit came from Rosenblatt (1958). He called these processing units ”perceptrons”. These processing units include weight vectors that weigh the neuron inputs according to the importance for classification and an activation function that uses the weighted input values and outputs a neuron response. A neuron can be represented using Equation 5 and Figure 2.1. [22]. The training of the neural network is simply trying to find the weights of all the neurons in the network to satisfy the output goal.

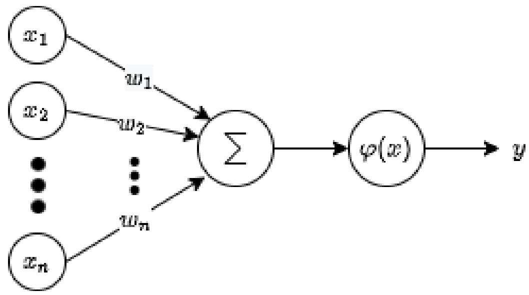


Figure 2.1 A model of a neuron.

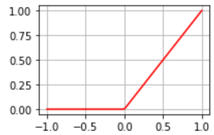
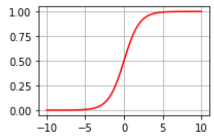
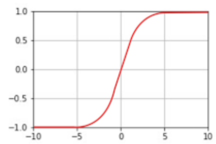
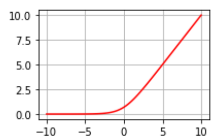
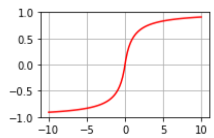
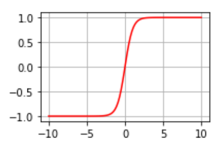
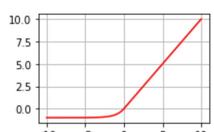
$$y = \varphi(\sum_{n=1}^{n_{max}} w_n x_n) \tag{5}$$

The x_n represents each input of the neuron, w_n represents the weights of each input, n_{max} represents the number of inputs, φ represents the activation function, and y represents the neuron response in Equation 5.

The activation functions are the functions that executes the output of the neurons. Rosenblatt (1958) simply used the Heaviside step function as the activation function in the perceptron model [22]. As the neural networks' popularity increased in time, which was not very popular when it was first pitched due to the high computation costs of the time, and the goals of the neural networks got more and more complex, other activation functions were tried and proven to be better for different kinds of problems. The activation functions explored in this thesis are defined in Table 2.1.

The neurons together form network layers. There are three layer types: Input, hidden, and output. Each neuron in the input layer accepts each input data feature as its input. Hidden layers are the layers between the input and output layers. The inputs of the neurons in the hidden layers are each output of the previous layer's neurons. The number of hidden layers determines the depth of the neural network. The output of the neurons in the output layer determines the classification result. An example of a neural network with one hidden layer can be seen in Figure 2.2. Hornik et al. (1989) showed that this kind of multilayer

Table 2.1 Activation functions.

Function name	**Function plot**	**Function**
ReLU [23]		$\varphi(x) = \max(x, 0)$
Sigmoid [24]		$\varphi(x) = \dfrac{1}{1+\exp(-\alpha x)}$
Softmax [25]		$\varphi(x)_i = \dfrac{\exp(x_i)}{\sum_j \exp(x_j)}$
Softplus [23]		$\varphi(x) = \log(\exp(x) + 1)$
Softsign [23]		$\varphi(x) = \dfrac{x}{\mathrm{abs}(x)+1}$
tanh [24]		$\varphi(x) = \tanh(x)$
SELU [23]		$\varphi(x) = \begin{array}{l} if\ x > 0:\ s * x \\ if\ x < 0:\ s * \alpha * (\exp(x) - 1) \end{array}$
ELU [23]		$\varphi(x) = \begin{array}{l} if\ x > 0:\ x \\ if\ x < 0:\ \alpha * (\exp(x) - 1) \end{array}$

feed-forward network can simulate any given function successfully. They showed that carefully chosen weights can result in an estimator function [26]. One run to get a neural network's output is called forward propagation. The forward-propagation can be represented with Equation 6. This equation is the output of any given neuron in the neural network. Calculating each neuron response from the input layer to the output layer in order results in an estimation which is the output layer neuron responses.

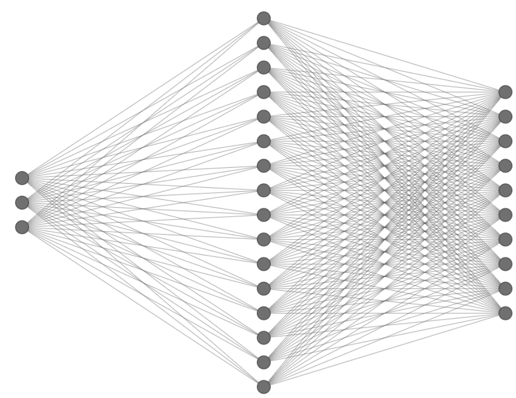


Figure 2.2 A neural network model.

$$a_j^l = \varphi^l(\sum_k w_{jk}^{\ l} a_k^{l-1} + b_j^l) \tag{6}$$

The φ^l represents the activation function of the l^{th} layer, a_j^l represents the output of the l^{th} layer's j^{th} neuron, $w_{jk}^{\ l}$ represents the weight of the neuron's k^{th} input, a_k^{l-1} represents the output of the $(l-1)^{th}$ layer's k^{th} neuron, and b_j^l represents the bias term of the neuron in Equation 6. The first layer neuron outputs can be considered the dataset inputs with this equation.

We will see in this thesis that the activation functions of the layers can be different from each other. A real-time ANN classifier simply would take a data sample as input and apply forward propagation to classify that sample. To give an example, the 10 output neurons seen in Figure 2.2 represents each of the flight maneuver to be classified. The responses of these neurons after a run of forward propagation can be seen in Figure 2.3. Brownlee (2020) suggests that the softmax function, when used in the output layer, gives a nice probability distribution between the classes to predict. Summing the output layer responses would result in 1 if this is the case [27]. By looking at Figure 2.3, we can determine that the maneuver is the second class, which is descent (The maneuvers are explained further in the thesis) with a 0.552 probability.

0	1	2	3	4	5	6	7	8	9
8.03404e-06	1.69715e-09	0.552171	4.66592e-13	3.35371e-08	0.154207	0.14446	1.36668e-08	2.29219e-06	0.149152

Figure 2.3 A prediction.

Single hidden layer neural networks are simple and shallow one hidden layered neural networks. It is the basis of neural networks and is useful for non-complex classification problems.

Deep neural networks are formed of multiple hidden layers and are generally better with more complex classification problems with numerical input datasets. Das and Roy (2019) also mention that DNNs are formed of more than one hidden layer and they can be referred to as stacked neural networks [28].

Recurrent neural networks differ from other types of neural networks due to exploiting historical data. The RNNs feed back the outputs to their input to include historical information, causing a sense of memory. This kind of architecture works brilliantly with sequential data, as opposed to feed-forward neural networks which assume no correlation exists between data features and data samples. Recurrent neural networks are mostly used in speech recognition and natural language processing where the sequence of the data matters [29]. The dataset of this study is a time series where the samples are logged with a frequency of 30 Hz. The goal and the form of the dataset are suitable to be used with recurrent neural networks. Since the sequence of the data matters with this kind of neural network, random shuffling preprocessing should not be applied to the dataset. Hochreiter and Schmidhuber (1997) first introduced the LSTM method and ANN layers [30]. LSTMs became the basis of recurrent neural networks after this point.

Brownlee (2019) explains the loss value as the value that shows how far the neural network is away from satisfying its goal. The purpose of the training is to minimize the loss value computed by the loss function [31]. A similar statement can be read by the Keras tutorial on loss [32]. The loss functions that are explored to get the success of the neural networks in this thesis are defined under Table 2.2.

Table 2.2 Loss functions.

Function name	**Function**
Categorical Cross-Entropy [31]	$L = \sum_k P_k \log(Q_k)$
KL Divergence [33]	$L = \sum_k P_k \log(\frac{P_k}{Q_k})$
MSE	$L = \frac{1}{k_{max}} \sum_k (P_k - Q_k)^2$
MAE [34]	$L = \frac{1}{k_{max}} \sum_k \mathrm{abs}(P_k - Q_k)$
Cosine Similarity [32]	$L = -\sum \vert\vert P\vert\vert_2^2 \vert\vert Q\vert\vert_2^2$
Huber Loss [35][32]	$L = For\ each\ value\ x\ in\ error\ =\ P_k\ -\ Q_k :$ $if\ \mathrm{abs}(x) <= d : 0.5x^2$ $if\ \mathrm{abs}(x) > \ \ d : d\vert x\vert - 0.5d^2$ $where\ d\ is\ delta$
Categorical Hinge [32]	$L = \sum_k \max(1 - P_k Q_k, 0)$

The P_k represents the response of a neuron while Q_k represents the expected result. The loss (and also the cost) is calculated using the response and the expected values.

The back-propagation algorithm was pitched by Rumelhart et al. (1986) and it runs right after the forward propagation in a training neural network [36]. As we discussed earlier, the purpose of the training of an ANN is to adjust the neuron weights to create an estimator. The back-propagation algorithm is the one adjusting the weights in training by minimizing the loss values calculated. It does that using calculating the gradient of the loss function.

Grant Sanderson (2017) explains the calculus behind the gradient with a back-propagation algorithm by using the MSE loss function for the simplest artificial neural network shown in Figure 2.4 [37].

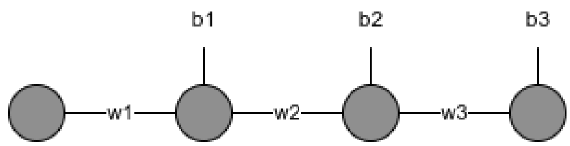


Figure 2.4 A simple ANN model.

By referencing Equation 6, we can give the response of the output neuron as Equation 7 for the simple ANN shown in Figure 2.4 as a^l [37].

$$z^l = w^l a^{l-1} + b^l \ , \ a^l = \varphi(z^l) \tag{7}$$

Assume we fed forward a training sample and the expected output of the sample is denoted as y. Then the cost would be calculated as Equation 8 using the MSE function [37].

$$C_0 = (a^l - y)^2 \tag{8}$$

Remember that we are trying to minimize the cost value and we are interested in the Equation 9 since it represents the cost value change rate as the weight w^l is changed. We can see that due to the dependencies of the variables, we can write a partial derivative chain [37].

$$\frac{\partial C_0}{\partial w^l} = \frac{\partial z^l}{\partial w^l}\frac{\partial a^l}{\partial z^l}\frac{\partial C_0}{\partial a^l} \tag{9}$$

When we actually compute the derivatives, we get Equation 10. This equation shows how a change in a specific weight, w^l in this case, affects the cost for a single data sample [37].

$$\frac{\partial C_0}{\partial w^l} = a^{l-1}\varphi'(z^l)2(a^l - y) \tag{10}$$

We can generalize this equation to the whole dataset, meaning we can find the full loss change rate for that specific weight w^l using Equation 11. Where n is the total number of samples and k is each data sample index [37].

$$\frac{\partial C}{\partial w^l} = \frac{1}{n}\sum_k \frac{\partial C_k}{\partial w^l} \tag{11}$$

Note that we can adjust the biases just like weights. Then in a similar fashion, we can calculate the derivative shown in Equation 9 by replacing w^l with b^l and get Equation 12 to find that it results in 1 [37].

$$\frac{\partial C_0}{\partial b^l} = \frac{\partial z^l}{\partial b^l}\frac{\partial a^l}{\partial z^l}\frac{\partial C_0}{\partial a^l} \tag{12}$$

The two gradients we found are both a part of the full gradient that we should be computing shown in Equation 13. The number of neurons and layers only changes the size of this matrix [37].

$$\nabla C = [\tfrac{\partial C}{\partial w^1}\ \tfrac{\partial C}{\partial b^1}\ \ldots\ \tfrac{\partial C}{\partial w^l}\ \tfrac{\partial C}{\partial b^l}] \tag{13}$$

Then finally we need to do the same thing we did for w^l, b^l for a^{l-1} too. Since a^{l-1} also depends on its own biases and weights, the same chain rule applies to that too. We did everything on the output layer's neuron but now we can apply it to the previous layer and find the full gradient matrix. This is simply back-propagation at work [37]. A run of the forward propagation and back-propagation is called an epoch. Weights and biases are updated in each epoch towards the loss goal. This change rate is called the learning rate. The learning rate is an important hyperparameter for a training ANN. A small learning rate would cause a slow or stuck training session while a large learning rate would cause unstable training. The learning rate is generally denoted using "η". This epoch loop is iterated until the threshold epoch is reached or the predetermined ANN loss is achieved. The flowchart of this classical training loop can be seen in Figure 2.5.

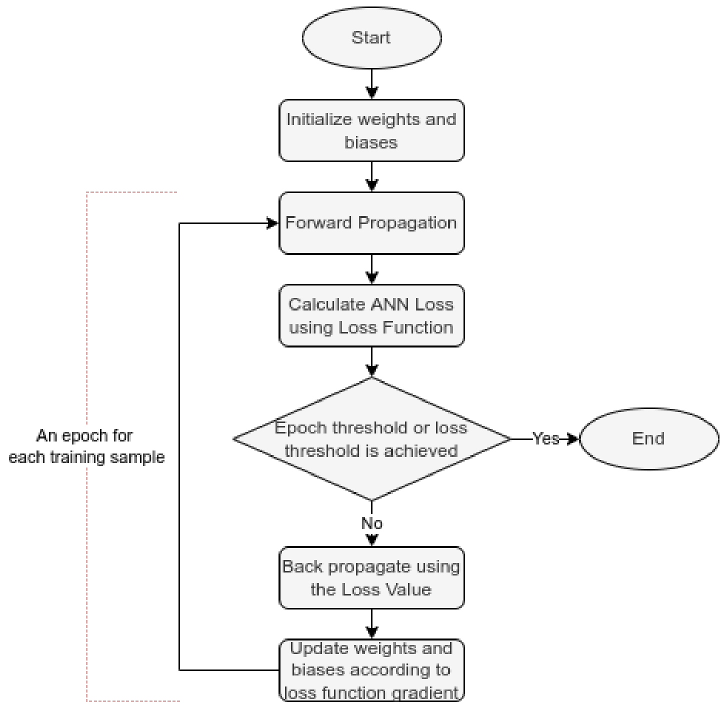


Figure 2.5 A training flow-chart.

Since the back-propagation algorithm is pitched, a lot of ANN optimizers are added to the literature which all are its variations. The one we covered above is batch gradient descent, which is the classical one. Ruder (2017) investigates the ANN optimizers included in Tensorflow with Keras [38]. Incidentally, the optimizers tried in this thesis are the same ones.

Ruder (2017), starts to lay the foundation by explaining that the purpose of the gradient descent is to minimize a $J(\theta)$ objective function where θ is the weights and biases of a network. The batch gradient descent can be summarized with Equation 14 where $\nabla_\theta J(\theta)$ is the derivative of the loss function w.r.t. θ. In batch gradient descent, the loss should be calculated using the whole dataset for one update of the weights and biases. This is why this method is computationally too slow but guaranteed to find the local minima [38].

$$\theta = \theta - \eta.\nabla_\theta J(\theta) \tag{14}$$

The SGD can be summarized with Equation 15. The SGD updates the weights and biases with each data sample. This is why SGD is a bit more sloppy than batch gradient descent while being much faster.

$$\theta = \theta - \eta.\nabla_\theta J(\theta; x^i; y^i) \tag{15}$$

The mini-batch gradient is the combination of both SGD and batch gradient descent to combine the goods of both. It simply creates batches of data samples and updates the parameters for each batch. When the batch size is 1, this method turns into SGD, when the batch size is dataset size, this method turns into batch gradient [38].

The Adagrad [39], introduced by Duchi (2011), adapts η accordingly with θ_i which is each network parameter. Since the η is changed by the algorithm and not given, the trainer is not concerned with choosing a learning rate with this method. The Equation 16 shows how the Adagrad method adapts the parameter-specific learning rate. The $\theta_{t+1,i}$ represents the value of i^{th} network parameter at data sample $t+1$, ϵ represents a smoothing term to avoid zero division, and Ruder defines "$G_{t,ii}$ is a diagonal matrix where each diagonal element i, i is the sum of the squares of the gradients w.r.t. θ_i" [38].

$$\begin{aligned} g_{t,i} &= \nabla_{\theta t} J(\theta_{t,i}), \\ \theta_{t+1,i} &= \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}}.g_{t,i} \end{aligned} \tag{16}$$

The term $G_{t,ii}$ gets larger in each iteration and the learning comes to a stop at one point with the Adagrad method [38]. The Adadelta method [40], pitched by Zeiler (2012) and the RMSprop method [41] pitched by Hinton (2012), successfully base their methods on top of Adagrad and solve the mentioned problem independently from each other [38].

Two other optimization methods that are adapting the learning rate are Adam and Adamax [42], which were pitched by Kingma and Ba (2014).

$$
\begin{aligned}
m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t \\
v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \\
\hat{m}_t &= \frac{m_t}{1-\beta_1^t} \\
\hat{v}_t &= \frac{v_t}{1-\beta_2^t} \\
\theta_{t+1} &= \theta_t - \frac{\eta}{\sqrt{\hat{v}_t}+\epsilon}\hat{m}_t
\end{aligned}
\tag{17}
$$

In these optimization methods, the weights and biases are updated with each data sample using Equation 17 for Adam optimization and Equation 18 for Adamax optimization [38]. Kingma and Ba (2014) initializes v_t and m_t with 0 and gives the default values of $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ [42].

$$
\begin{aligned}
u_t &= \max(\beta_2 . v_{t-1}, |g_t|) \\
\theta_{t+1} &= \theta_t - \frac{\eta}{u_t}\hat{m}_t
\end{aligned}
\tag{18}
$$

Finally, the Nadam optimization, introduced by Dozat (2015), incorporates Nesterov momentum into Adam optimization [43].

2.4. Performance Metrics

The successes of the classification models are evaluated using accuracy, precision, and recall for each maneuver [15]. The formula for these performance metrics can be seen in Equation 19.

The accuracy is the ratio of correctly classified vs. the total predictions for a specific class. This measure can be misleading in scenarios where the dataset is imbalanced and the number of negatives is high. For example, this measure becomes irrelevant with a classification problem with 10 classes.

Precision is a measure of how much a classification decision of a specific class should be trusted. Low precision for a class means it is highly likely this class is predicted but the actual result is another class.

Recall is a measure of identifying a class over the whole dataset without missing one. Low recall for a class means it is highly likely that another class is predicted but the actual result is that specific class.

$$\begin{aligned} accuracy &= \frac{TP + TN}{TP + TN + FP + FN}, \\ precision &= \frac{TP}{TP + FP}, \\ recall &= \frac{TP}{TP + FN} \end{aligned} \tag{19}$$

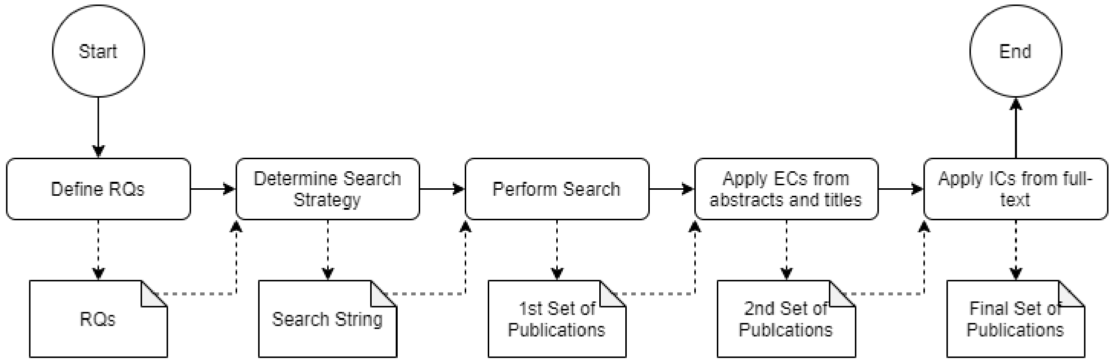


Figure 3.1 Literature review process flow

3. RELATED WORK

A systematic literature review was conducted to interpret all available studies and get a comprehensive understanding of the usage of machine learning methods for the purpose of flight maneuver classification. Through the research questions, this literature review contributes a way to explore flight data attributes, and machine learning algorithms used to classify flight maneuvers. It also provides a comparison between these methods, especially artificial neural networks, in the form of performance and accuracy.

The systematic literature review is conducted by a literature search in well-known libraries: IEEE Explore, ACM DL, arXiv, Wiley Online Library, Science Direct, ResearchGate, and Web of Science. It is conducted using "Guidelines for performing Systematic Literature Reviews in Software Engineering" [17] by Kitchenham and Stuart (2007).

3.1. Research Design

Activities to the research method can be seen below and a visual representation in the form of a flow-chart can be seen in Figure 3.1:

1. Define research questions (Output: Research questions).

2. Determine search strategy (Output: Search string).

3. Search is performed on a selection of digital libraries (Output: First set of publications – Table 3.1)
4. Apply exclusion criteria by reviewing titles and abstracts. (Output: Second set of publications)
5. Apply inclusion criteria to the second set of publications from the full text. (Output: Final set of publications)

3.1.1. Research Questions

PICOC(Population, Intervention, Comparison, Outcomes, and Context) that derive the research questions can be seen below:

- Population: Machine learning and aviation researchers.
- Intervention: Machine learning using flight data for flight maneuver classification.
- Comparison: Usage areas, different implementations, performance, and accuracy of machine learning algorithms with aircraft flight data for the purpose of flight maneuver classification.
- Outcomes: A good grasp and understanding of current developments in aviation academies and industry that use machine learning algorithms for the purpose of flight maneuver classification, and possible research areas to conduct studies to fill the gap with neural networks.
- Context: Academic and industrial studies.

Using this PICOC, research questions are defined below:

- **RQ-1.** Which aircraft flight data attributes are relevant to achieving flight maneuver classification?

- **RQ-2.** Which machine learning algorithms are used for flight maneuver classification?
  - **RQ-2.1.** How are machine learning algorithms and techniques implemented to achieve flight maneuver classification?
  - **RQ-2.2.** How are the performance and accuracy of these algorithms?
  - **RQ-2.3.** How the performance and accuracy of these algorithms can be increased? Which optimization methods are used with the mentioned algorithm?
- **RQ-3.** What is the role of ANNs in flight maneuver classification? How it is compared to other machine learning methods?

3.1.2. Search Strategy

The search string is a combination of logically formulized keywords to be used in search facilities of digital libraries. The researcher can find all the related studies and filter out others with a finely chosen search string. The search string chosen for this SLR is:

"(detect OR classification OR recognize OR recognition OR identify OR identification OR classify) AND (manoeuvre OR maneuver) AND (aircraft OR flight)"

This search string is used in multiple well-known digital libraries for computer science and software engineering: IEEE Explore, ACM DL, arXiv, Wiley Online Library, Science Direct, ResearchGate, and Web of Science. The search string is searched in the title and abstract of the studies to eliminate brief mentions. The search date range is from 1943 to 2021. 1943 is the year when the first computational model for a neural network is introduced to the literature. Results of the searches in digital libraries can be seen below in Table 3.1. The total number of publications found in these digital libraries combined was 674, which forms the first set of publications.

Note that there is no term containing "machine learning" related concepts in the search string. The reason for this decision is to make the first set of publications wider. The titles and abstracts of the publications are going to be skimmed using exclusion and inclusion criteria for the second set of publications. That step will be the point where the publications not including machine learning methods will be eliminated. There are lots of different approaches to machine learning, and a keyword may cause missing publications.

Table 3.1 List of searched digital libraries and the numbers of found publications

Digital Library	**URL**	**Number of Publications**
ACM DL	dl.acm.org	9
IEEE Explore	ieeexplore.ieee.org	369
arXiv	arxiv.org	15
Wiley Online Library	onlinelibrary.wiley.com	61
Science Direct	sciencedirect.com	103
ResearchGate	researchgate.net	100
Web of Science	apps.webofknowledge.com	17

3.1.3. Publication Selection

Exclusion criteria are applied by reviewing titles and abstracts of the first set of publications for the publication to get into the second set of publications. Publications satisfying any of the exclusion criteria are eliminated. The exclusion criteria's exclusion conditions can be seen below:

- **EC-1.** The researcher cannot access the publication;
- **EC-2.** The publication is not in English;
- **EC-3.** The publication is a duplicate that is found in another digital library;
- **EC-4.** The publication does not mention "flight maneuver classification" as explained in the 1.Introduction section;
- **EC-5.** The publication briefly mentions "flight maneuver classification" without proposing a method for it;

- **EC-6.** The publication does not mention any machine-learning method;
- **EC-7.** The search string is only found on the publication as a reference to other publications.

EC-4 and EC-5 are applied to remove any unrelated publications. Remember that there was no term containing “machine learning” in the search string: EC6 is used to introduce a “machine learning” filter to the publication selection. The number of the second set of publications was 16.

Inclusion criteria are applied by reviewing the full text of the second set of publications to get the final set of publications. To get into the final set of publications, the reviewed publication has to satisfy any of the conditions below:

- **IC-1.** The publication compares flight data attributes for better flight maneuver classification;
- **IC-2.** The publication mentions machine learning methods to achieve flight maneuver classification;
- **IC-3.** The publication includes comparisons of different flight maneuver classification methods, performance, and accuracy wise;
- **IC-4.** The publication includes comparisons of machine learning methods and traditional aviation methods for flight maneuver classification;
- **IC-5.** The publication is a primary study.

The first four inclusion criteria are strongly related to the research questions. These inclusion criteria ensure the included publications answer at least one of the research questions. Besides these criteria, the fifth criterion ensures the included publications are primary studies. The inclusion criteria did not eliminate any studies and the number of studies in the final set of publications was 16.

Study No	Study Title, Authors, Year	Venue	Ans RQs
1	Maneuver Identification Challenge by Samuel et al. (2021) [44]	IEEE HPEC Conference Paper	1
2	Automatic Detection of Flight Maneuvers with the Use of Density-based Clustering Algorithm by Socha et al. (2018) [6]	ISC-NTAD Conference Paper	1, 2
3	Identification of Flight Maneuvers and Aircraft Types Utilizing Unsupervised Learning with Big Data by Blanks et al (2017) [4]	SIEDS Conference Paper	1, 2
4	Research on High Precision Terrain Dynamic Loading Technology Based on Flight Trajectory Prediction by Wu et al. (2018) [8]	IEEE ITOEC Conference Paper	1, 2
5	Flight Motion Recognition Method Based on Multivariate Phase Space Reconstruction and Approximate Entropy by Qu, Lv, Yang, and Tang (2021) [45]	Chinese Control Conference Paper	1, 2
6	Threat Assessment for Rotte Based on Cooperative Tactical Recognition by Meng, Zhou, and Zhang (2019) [9]	IEEE IUCC – DSCI – SmartCNS Conference Paper	1, 2
7	An Automatic Method to Estimate the Calibration Quality of the Aeromagnetic Compensation by Wang, Han, Hu, and Zhan (2019) [7]	IEEE International Geoscience and Remote Sensing Symposium Conference Paper	1, 2
8	Optimal Guidance Method for UCAV in Close Free Air Combat by Chen et al. (2019) [10]	IEEE IUCC – DSCI – SmartCNS Conference Paper	1, 2
9	A Novel Algorithm for Identifying Patterns from Multisensor Time Series by Ruotsalainen et al. (2009) [11]	WRI World Congress on Computer Science and Information Engineering Conference Paper	1, 2
10	Trajectory Prediction For Low-Cost Collision Avoidance Systems by Baumgartner and Maeder (2009) [46]	IEEE AIAA Digital Avionics Systems	1, 2
11	Classification of aircraft maneuvers for fault detection by Oza et al. (2003) [14]	International Conference on Multiple Classifier Systems Paper	1, 2, 3
12	HMM relative entropy rate concepts for vision-based aircraft manoeuvre detection by Molloy and Ford (2013) [47]	Australian Control Conference Paper	1, 2
13	A flight maneuver recognition method based on multi-strategy affine canonical time warping by Wei et al. (2020) [12]	Applied Soft Computing Journal Paper	2
14	Link between Flight Maneuvers and Fatigue by Jylha et al. (2011) [48]	ICAF Journal Paper	1, 2
15	Automatic Flight Maneuver Identification Using Machine Learning Methods by Bodin (2020) [15]	Linköping University Master of Science Thesis in EE	1, 2, 3
16	A machine learning-based framework for aircraft maneuver detection and classification by Dang et al. (2021) [5]	ATM Seminar Paper	1, 2

Table 3.2 Included studies

3.2. Research Results

The studies included in this SLR can be seen in Table 3.2. below. This table also contains information on which research questions does the study answers in the “Ans RQs” column.

RQ-1. Which aircraft flight data attributes are relevant to achieve flight maneuver classification?

The study (No.1) introduces a challenge where they provide a large, verified, and professional dataset and expect AI developers to take this challenge and compete. This challenge and the dataset are introduced by MIT Air Force AI Accelerator and the writers are academicians from this department. There are three challenges introduced by using the same dataset: classify possible and impossible maneuvers, classify which maneuver is performed (18 classes), and finally score the performance of the maneuver. The dataset is a very detailed and verified dataset that, interestingly, includes PNG files where the flight is drawn in 2D from a bird's eye view. This dataset also includes these data attributes: time (sec), xEast(m), yNorth(m), zUp(m), vx(m/s), vy(m/s), vz(m/s), heading(deg), pitch(deg), roll(deg) [44].

The study (No.2) collects 414 maneuvers from flight training students using the TRD40 flight simulator. Only bank and vertical speed data attributes are used to classify maneuvers [6].

The study (No.3) gathers data from open-source ADS-B by U.S. Air Force. They perform feature engineering to select data attributes where they consider many attributes. By combining some ordinary flight data such as turn rate, climb rate, acceleration, heading, altitude, distance, G forces, etc. they create new data attributes. The derivation of each of these new data attributes is detailed in the study. They also include the contributions of these attributes to classification results [4].

The study (No.4) uses altitude, yaw angle, and roll angle to work with the Bayesian network to classify somersault, half-somersault, constant height, descent, climb, hoick, combat curve, hover, roll, dive, and half-roll reverse maneuvers. The maneuvers also act as an observation node state set [8].

The study (No.5) uses height, roll, pitch, and yaw to classify flight motions which are climb, landing, right turn up, left turn up, right S motion, left S motion, fixed height hover, and spiral upward. They use these attributes as time-ordered sequences and they try to distinguish these sequences. The height, roll, pitch, and yaw are the raw data attributes of the proposed method but the classification is performed on the approximate entropies and recurrence plots that are generated from the raw data [45].

The overall target of the study (No.6) is to assess the threat posed by the opponent aircraft. That is why this study focuses on classifying the opponent's maneuver from the possible maneuvers which are G-force left turn, G-force right turn, G- force diving, and G-force climbing. The data attributes used for this purpose are the opponent's relative distance (which includes latitude, longitude, and altitude of each aircraft), speed of each aircraft, heading of each aircraft, and tactical situations [9].

The study (No.7) uses the vector magnetic field of aircraft which is obtained from the vector magnetometer as the data to solve the classifying problem. The authors especially avoid attitude information because that information is not always available. This study classifies the maneuver into two classes: level flight, and maneuvering [7].

Similar to study (No.6), study (No.8) also tries to classify an opponent aircraft's maneuver. The data attributes to perform the method are relative distance, relative height, relative altitude of the opponent, and relative performance. The classes to classify from are straight flight, hover, loop, snake, and battle turn [10].

The study (No.9) uses multi-sensor time-series operational loads measurement flight recording of F-18 Hornet aircraft as the dataset. The operational load data consists of more than 100 data attributes which are time-ordered. The example data attributes are Mach number, phi angle, roll rate, and acceleration. One should note that this dataset is not pre-labeled into classes to recognize. The maneuvers to classify are split s, steep turn, and loop [11].

The study (No.10)'s classes of flight maneuvers are straight flight, slow turn, initiate left turn, initiate right turn, terminate turn, and circle. The data attributes of maneuver classification are generated from earlier sections of trajectory prediction state estimates. Also, a state machine model is used with a probability of state changes where states are maneuvers [46].

The study (No.11) uses data from real helicopter flights with AH1 Cobra and OH58c Kiowa. The data attributes are revolutions per minute of the planetary gear, torque, and vibration data from six accelerometers [14].

The study (No.12) uses morphologically processed image sequences as the data attribute. They classify aircraft maneuvers from the images mentioned [47].

The study (No.14) mentions altitude, angular velocities, accelerations, roll, pitch, roll rate, G, pitch rate, and yaw rate as useful to identify flight maneuvers which are split s, roll turn, turn, loop, push, roll, and oblique loop [48].

The study (No.15) uses load factor, roll, pitch, yaw, roll rate, pitch rate, angle of attack, angle of sideslip, delta Mach, delta altitude, stick inputs, pedal inputs, and throttle inputs to classify six flight maneuvers which are bleed off turn, wind up turn, slow down turn, steady turning sideslip, 360 degrees roll, and barrel roll. The dataset used was obtained from 24 flight tests at Saab Aeronautics [15].

The study (No.16) uses automatic dependent surveillance- Broadcast data collected from Singapore Flight Information Region. The dataset includes 2793 flights. This dataset is open-source and available for reference (No.19). The flight paths collected are drawn in the study for visualization. The classified maneuvers are sequencing, track shortening, weather, and coordinate optimization. They differ from agile aircraft maneuvers since this study focuses on civil airplane air traffic control [5].

RQ-2. Which machine learning algorithms are used for flight maneuver classification?

The study (No.2) uses density-based spatial clustering of applications with noise (DBSCAN) for flight maneuver classification [6].

The study (No.3) tries not only to classify flight maneuvers but also to classify aircraft types from the dataset using unsupervised learning. It uses k-means and k-medoids cluster models for classification. It also points out that the DBSCAN algorithm can be used for the same purpose which is later tried by the study (No.2) indeed [4].

The study (No.4) uses Dynamic Bayesian Networks, which is a model to work with stochastic processes, for flight maneuver classification and, furthermore, flight path estimation [8].

The study (No.5) uses Multivariate Phase Space Reconstruction and Approximate Entropy for flight maneuver classification. It calls this method: Flight Motion Recognition Model [45].

The study (No.6) takes a different approach and tries to classify the maneuvers of an opponent aircraft. The overall target of this study is not to classify maneuvers but to assess the threat posed by the opponent aircraft. To assess the thread this study proposes a Bayesian network with 3-tier architecture. The middle layer is the one that recognizes the intent of the opponent [9].

The study (No.7) uses k-means and the Gaussian Mixture Model for the aircraft maneuver classification problem [7].

The study (No.8) uses Dynamic Bayesian Networks to classify the opponent aircraft's maneuver [10].

The study (No.9) uses a template-based approach to flight maneuver classification problem since it is seen that the problem was a pattern recognition problem [11].

The study (No.10) uses Interacting Multiple Model filtering which is an extension of the Kalman Filter. Interacting Multiple Model filtering provides a solution to state estimation problems. Also, a state machine model is used where states are different maneuvers (namely straight flight, slow turn, initiate left turn, initiate right turn, terminate turn, and circle). The aircraft type to classify maneuver is a glider in this study [46].

The study (No.11) uses multilayer perceptrons and radial basis function networks for the purpose of maneuver classification of helicopters [14].

The study (No.12) uses Hidden Markov Model filters to classify maneuvers from morphologically processed image sequences. This study focuses on identifying if a maneuver is being performed at the moment or not instead of classifying which maneuver is being performed [47].

The study (No.13) uses multi-strategy affine canonical time warping to solve the problem. It also includes dynamic time warping for comparison with MACTW [12].

The study (No.14) uses an approximate pattern-matching algorithm in their AMANA software. It also backs this algorithm by manually creating templates for each flight maneuver manually and comparing them [48].

The study (No.15) uses logistic regression, support vector machines and ANNs to classify six flight maneuvers, and compare the results gained [15].

The study (No.16) uses DBSCAN, isolation forest, and k-means algorithms sequentially to identify flight maneuvers [5].

RQ-2.1. How are the machine learning algorithms and techniques implemented to achieve flight maneuver classification?

The study (No.2) identifies clusters using the two data attributes mentioned in RQ-1 with the DBSCAN algorithm. This was done without time being a parameter. This is why a second clustering was needed to distinguish the same maneuvers performed at different times where the time is added [6].

The study (No.3) first had 4 clusters to identify: Take-off, landing, cruising, and maneuvering. After this clustering, these clusters are also divided into sub-clusters. The total number of maneuvers classified is the number of total sub-clusters [4].

The study (No.4) shows their Dynamic Bayesian Network prediction models to perform the job which has 6 nodes. It also includes a flowchart for the implementation of the classification model [8].

The study (No.5) first unites the Phase Space Reconstruction of the data attributes mentioned in RQ-1 using Bayesian theory to achieve Multivariate Phase Space Reconstruction. At this point, they use Recurrence Plot and Approximate Entropy to distinguish maneuvers. It includes a detailed step-by-step implementation and flowchart of their method [45].

The study (No.7) uses the k-means method to distinguish magnetic field data from different headings. The Gaussian Mixture Model is used to identify aircraft attitude (phi, theta, psi angles) which reflects the maneuver being performed [7].

The study (No.8) proposes a 10-node DBN model to decide on a maneuver for the aircraft accordingly to the maneuver performed by the opponent aircraft. The nodes and states are explained thoroughly in the study [10].

The study (No.9)'s architecture to solve the problem is as follows: First, template definition and signal quantization on the dataset is performed; second, dynamic time warping matrices are found; and finally, the patterns, using DTW matrix, are classified [11].

The study (No.11) has one hidden layer of multi-layer perceptrons and radial basis function networks and a combination of the two. It includes the results of these combinations [14].

The architecture proposed by the study (No.14) includes manually created templates for each maneuver. The automatically generated rules for maneuver classification are checked with these templates [48].

The study (No.16) shares the architecture of their proposed solution to maneuver classification problem. This proposed solution follows the DBSCAN algorithm, major flow classification, flow-based nominal flight plan, isolation forest, and k-means as there is k number of clusters corresponding to each maneuver [5].

RQ-2.2. How are the performance and accuracy of these algorithms?

The study (No.2) successfully classifies 390 of 414 maneuvers. The sensitivity is found to be 0.9420. It is found that the sensitivity of the results was higher for the data created by the experienced flight students [6].

The study (No.3) figures that 12 of the 14 sub-cluster distributions were significantly different from the main dataset. The chi-squared results are included to show these results. The study concludes that the proposed methods can confidently classify maneuvers and flight phases, and also aircraft types [4].

The study (No.5) shows the Recurrence Plots of eight different aircraft motions and states that similar motions' RPs are similar. Also, the approximate average value significantly changes for four different types of aircraft motions. The study (No.5) does not include a result in the form of accuracy or performance [45].

The study (No.7)'s main purpose is to calculate the figure of merit of aeromagnetic compensation to reduce the magnetic interference of the aircraft. The maneuver classification is needed to automatically calculate the FOM. That is why automated vs. manual FOM calculations are used as the performance and accuracy criteria. The automated FOM calculation using the maneuver classification algorithm calculates the FOM the same as the manual calculation. One should note that maneuvers are only classified into two classes in this study since it was enough to calculate the FOM [7].

The study (No.8) also does not aim to classify maneuvers only. It proposes a decision-making solution for an aircraft fighting with another maneuvering aircraft. That is why it does not include any classification results. However, it shows simulation results where the decision-making solution uses the maneuver classification method and makes decisions correctly [10].

Remember that the study (No.9)'s dataset was not pre-labeled. The results are calculated by automatically detecting maneuvers and comparing them by letting an experienced analyst manually detect the same maneuver. For the split-s, the accuracy is 100% and for loops, the accuracy is 97% [11].

The study (No.10) shares a bird-vision view of a flight with time points marked, and a plot showing the probability distribution of each maneuver vs. time. The plot can be mapped into the flight and one can deduct that maneuver classification is done correctly. However, the study does not include detailed accuracy and performance results [46].

The study (No.11) implements cross-validation by dividing their dataset. A confusion matrix is shared which includes all 14 classes. It is seen that MLPs outperform RBFs. After similar maneuvers are united, accuracy gets up to 95%. Very detailed accuracy and correlation results are available for both MLPs and RBFs with different depths for this work [14].

A large set of results are shared by the study (No.13) [12].

The study (No.15) uses 4-fold cross-validation to optimize the algorithm parameters and calculate the results. Recall, accuracy, precision, TPR, and FPR are measured for algorithms. On average, LR TPR is 96%, precision is 65%, accuracy is 94%, and FPR is 3%. SVM TPR is 92%, precision is 80%, accuracy is 97%, and FPR is 3%. ANN TPR is 94%, precision is 77%, accuracy is 97%, and FPR is 2%. They included every result from every algorithm for every maneuver in the study [15].

RQ-2.3. How the performance and accuracy of these algorithms can be increased? Which optimization methods are used with the mentioned algorithm?

The study (No.2) increases the performance and accuracy by adjusting the DBSCAN algorithm's clustering parameters by trial and error and using other references [6].

The study (No.3) finds the optimal number of clusters in the first phrase is four which is explained in RQ 2.1. The optimal number of clusters and sub-clusters are different for different kinds of algorithms (k-means, k-medoids) [4].

The study (No.5) gives the approximate entropy calculation variables (initial dimension) m = 2, (threshold) r = 0.15std [45].

The study (No.7) uses the similarity criterion function, which is also shared in the paper, to optimize the k-means algorithm [7].

The study (No.9) performs quantization on data attributes to get higher performance from the over 100 data attributes. The attributes are quantized into three levels: high, mid, and low [11].

The study (No.11) noticed that, from the confusion matrix, some maneuvers were being confused with others (namely hover, hover turn left, hover turn right, and coordinated turn right, coordinated turn left). The maneuvers that are being confused are simply united because the maneuvers were very similar. This action increases the accuracy of the algorithm. Also, the MLP networks give better results than RBF networks and the accuracy gets higher as the network gets larger [14].

The study (No.15) uses L2 regularization with a logistic regression algorithm to optimize logistic regression parameters. It uses kernel functions to classify nonlinearly and multiclass classifications (multiple SVMs united) to classify a number of classes with support vector machines. The study also mentions feature selection to get higher accuracies with SVMs [15].

RQ-3. What is the role of ANNs for flight maneuver classification? How it is compared to other machine learning methods?

The study (No.11) uses ANNs, namely multilayer perceptron networks, and radial basis function networks, for the purpose of flight maneuver classification for helicopters. Real flight data is used and got very good results which are mentioned in RQ2.2. The study (No.11) does not mention any comparisons with other methods [14].

The study (No.15) uses ANNs and other algorithms for the problem and compares them by considering TPR, FPR, precision, recall, and accuracy. ANNs are found to be slightly better than other algorithms. Unfortunately, the study does not include ANN architecture from which they got the results to replicate the results [15].

3.2.1. Literature Review Discussion

RQ-1. Which aircraft flight data attributes are relevant to achieve flight maneuver classification?

Fifteen of sixteen studies answer this research question. Two of the studies share an open-source database on which they performed flight maneuver classification. There are lots of different data attributes mentioned for this research question since the studies' scopes are different, e.g. air traffic, helicopter, aircraft, etc. However, the most occurring ones can be said to be roll, pitch, yaw, roll rate, pitch rate, and yaw rate, which are aircraft axis and axis speed rates. Altitude and longitudinal G force are also mentioned multiple times. The author also included the classified maneuvers in this research question. The most classified maneuvers were landing, climb, rolls, split s, and loops. There were, again, different maneuvers mentioned since the scopes were different.

RQ-2. Which machine learning algorithms are used for flight maneuver classification?

Fifteen of sixteen studies answer this research question. DBSCAN (3 studies), k-means (3 studies), and Bayesian networks (3 studies) were the most utilized solution to the maneuver classification problem. They were followed by dynamic time warping (2 studies), and artificial neural networks (2 studies). The general performance and accuracies of the studies were all relatively high. The ones sharing accuracy results shared over 90better results which are specific to the algorithms they used or some data pre-processing such as normalization, etc. Some of the studies (such as (No.5), (No.7), (No.16)) mixed the algorithms in complex architectures to get better results.

RQ-3. What is the role of ANNs for flight maneuver classification? How it is compared to other machine learning methods?

Only two studies ((No.11), and (No.15)) used ANNs for the problem. Both of them did not share any ANN architectures and ANN optimization methods. However, the results were very high for both of them. The study (No.15) compared the results of ANNs with logistic regression, and vector support machines and found ANNs to be better at classifying flight maneuvers. The simplistic presentation of ANNs (without architectures, activation function definitions, optimization methods) in these studies show that there is a gap in the literature about ANNs and flight maneuver classification. This gap can be filled with a thesis with a whole ANN solution including data attributes, data processing, ANN architecture, and results.

3.2.2. Potential Threats to Validity

Reliability: The issue of reliability is strongly related to individuals' results in the same study. Since the author of this study is only one person, reliability issues can be questioned. To give an example, the search strategy is not peer-reviewed and only written by the single author of this SLR.

Internal Validity: Internal validity is strongly related to the factors that can be comprising the findings of the research. The search string for the SLR studies can be a cause of comprising if the search string is not defined properly. Missing publications can cause internal validity problems. In this In SLR's situation, the search string is chosen such that it gives a huge amount of studies to search from. The real elimination started when the author started to eliminate studies by reviewing the abstracts of the studies. Another threat to internal validity is the exclusion and inclusion criteria since it is determined and applied by the only author.

External Validity: External validity is strongly related to the generalizability of the results. The SLR is conducted on multiple well-known digital libraries to avoid threats to external validity. However, the number of publications in the final set of publications is low.

Construct Validity: Construct validity is strongly related to the research questions and the purpose of the researchers' correlation. This situation is avoided using PICOC and strongly relating the research purpose, questions, and method.

This SLR reviewed the literature on flight maneuver classification with machine learning methods with the purpose of exploring and getting a comprehensive understanding of it. Of the 674 studies in the first set of publications, 16 of them are selected by exclusion criteria. All of the 16 studies made the final set of publications since the inclusion criteria did not eliminate any studies. RQ1 is answered by 15, RQ2 is answered by 15, and RQ3 is answered by 2 studies. RQ4 is not answered by any of the studies. Generalization on 2 of the research questions was successfully reached.

It is found that flight maneuver classification is a hot topic in literature, and it is getting more popular each year. A good number of machine learning methods are used and got good results. It is also found that flight maneuver classification can be looked at from different perspectives, such as air traffic control, helicopter development, aircraft development…

This SLR shows that there is a gap in flight maneuver classification using artificial neural networks. Also, it shows that there is a gap in comparing machine learning methods and traditional aviation methods when it comes to flight maneuver classification.

4. PROPOSED METHOD

4.1. Data Acquisition and Dataset

This paper contributes a flight maneuver classification dataset to the literature. The dataset is available on GitHub [16]. This dataset is created using X-Plane Flight Simulator 11 as the flight environment (to perform flights and obtain data). Laminar Research (2020), the publisher of X-Plane 11, mentions X- Plane 11 is an engineering tool since it can simulate any given aircraft's flight efficiency with high accuracy and explains how they provide a realistic flight in documentation named "How X- Plane Works" [49]. This accuracy of The X-Plane Flight Simulator provides a level of confidence in the solution provided in this paper since the data to train the neural network will be accurate to the real aircraft. Flights were performed with F-4 Phantom 2, which is a supersonic aircraft by Boeing Company.

X-Plane has built-in properties such as logging the data of the user's choice or sending it through UDP. By choosing groups of data, users can interact with real-time aircraft data. One can use the open source program developed for this paper to read X-Plane data sent using UDP and create own dataset [16]. The mentioned dataset is created using the program and selecting the below data groups in the X-Plane output menu to send using UDP. Selectable data groups provided by X-Plane that one should select to obtain the dataset, and the flight data features that are included in these data groups are explained below. A detailed explanation of the data features can be found in the dataset output reference of X-Plane [50]. The dataset attributes as they appear on the open-source dataset can be seen in Table 4.1.

4.1.1. Flight Data Features

Speeds contain the aircraft speeds in indicated airspeed and true airspeed. Since these values have a high correlation between them, only the indicated airspeed in knots is placed on the dataset. The indicated airspeed is the value presented in the aircraft head-up display and the

Table 4.1 Data attributes.

No	Data attribute	Unit	Data group
0	Vind	kias	Speeds
1	VVI	fpm	Mach, VVI, g-load
2	Gload-norml	G	
3	Gload-axial	G	
4	Gload-side	G	
5	elev-stick	[-1, 1]	Joystics aileron/elevator/rudder
6	ailrn-stick	[-1, 1]	
7	ruddr-stick	[-1, 1]	
8	elev-surf	[-1, 1]	Flight controls aileron/elevator/rudder
9	ailrn-surf	[-1, 1]	
10	ruddr-surf	[-1, 1]	
11	M	ftlb	Angular moments
12	L	ftlb	
13	N	ftlb	
14	Q	rad/s	Angular velocities
15	P	rad/s	
16	R	rad/s	
17	pitch	degrees	Pitch, roll, & headings
18	roll	degrees	
19	hding-true	degrees	
20	hding-mag	degrees	
21	alpha	degrees	Angle of attack, sideslip, & paths
22	beta	degrees	
23	hpath	degrees	
24	vpath	degrees	
25	slip	degrees	
26	throttle	[0, 1]	Throttle
27	lift	lb	Aerodynamic forces
28	drag	lb	
29	side	lb	
30	L	lb-ft	
31	M	lb-ft	
32	N	lb-ft	
33	flight	enum	Maneuver

maneuvers were performed by checking the indicated airspeed. The speed data correspond to the attribute *Vind-kias*.

Mach, VVI, g-load contain good distinguisher information in the context of maneuver classification. VVI is the vertical velocity in feet per minute. G-load information is the axial, side, and across the aircraft, relative to the aircraft body-axis, pressures on the aircraft. G-load information depends on angular velocities and moments. Due to this reason, they give a good idea of the aircraft's state. These data correspond to the attributes *VVI-fpm*, *Gload-norml*, *Gload-axial*, and *Gload-side* in the dataset.

Joystics aileron/elevator/rudder contain the pilot input information. It represents the pilot input provided to the stick. The aileron provides roll, the elevator provides pitch, and the rudder provides yaw movements. They are the ratio to full upward or rightward deflection and range between -1, and 1. These data correspond to the attributes *ailrn-stick*, *elev-stick*, and *ruddr-stick* in the dataset.

Flight controls aileron/elevator/rudder contain the control surface response of the pilot input. The aileron is the control surface that provides the roll movement, the elevator is the control surface that provides the pitch movement, and the rudder is the control surface that provides the yaw movement. Control surfaces contain the pilot input information and the control computer's corrections and effects. They also range between -1, and 1 parallel to the joystick information. These data correspond to the attributes *ailrn-surf*, *elev-surf*, and *ruddr-surf* in the dataset.

Angular moments contains roll torque (L), pitch torque (M), and yaw torque (N) in foot-pounds. These data correspond to the attributes *L-ftlb*, *M-ftlb*, and *N-ftlb*.

Angular velocities contain the aircraft's pitch (Q), roll (P), yaw (R) axes turn rates measured in aircraft body axes in radians per second. These rates tend to change in different configurations and states of the aircraft. They provide a good distinguisher of aircraft's orientation at the moment as wanted from the data features. These data correspond to the attributes *Q-rad/s*, *P-rad/s*, and *R-rad/s* in the dataset.

Pitch, roll, & headings contain the aircraft's position angles measured in aircraft body axes in Euler angles with respect to the earth's surface. These values vary greatly as the aircraft moves. True heading and magnetic heading are included in the dataset. These data correspond to the attributes *pitch-deg*, *roll-deg*, *hding-true*, and *hding-mag* in the dataset.

Angle of attack, sideslip, & paths contains the vertical path angle, angle of attack (alpha), sideslip (beta), and the horizontal path angle all in degrees. These pieces of information vary greatly as aircraft are in different orientations and speeds. That is why they are useful in the context of flight maneuver classification. These data correspond to the attributes *alpha-deg*, *beta-deg*, *vpath-deg*, and *slip-deg*.

Throttle (actual) contains another pilot input which is the commanded throttle to gain thrust. It ranges between 0 (no thrust) and 1 (maximum thrust). The throttle data correspond to the attribute *thro1-part*.

Aerodynamic forces contains the forces acting on the aircraft in pounds. These forces are lift, drag, and side forces. These data correspond to the attributes *lift-lb*, *drag-lb*, and *side-lb*.

4.1.2. Maneuvers

In a similar attempt to classify flight maneuvers with neural networks, Bodin (2020) classified 6 maneuvers which are bleed-off turn, wind Up Turn, slow-down turn, steady turning sideslip, 360° roll, and barrel roll [15]. Similarly, Oza et al. (2003) classified 14 helicopter maneuvers which are hover, hover turn left, hover turn right, coordinated turn left, coordinated turn right, forward flight low speed, and 8 more that were not mentioned [14].

Samuel et al. (2021) and MIT invite academia and industry to a flight maneuver classification challenge. This challenge clearly defines 18 flight maneuvers [44]. Based on these definitions, this paper tries to classify below 10 maneuvers:

Level flight is performed by keeping the aircraft's altitude, attitude, and speed constant for a forward flight.

Sustained turn is performed by sustaining altitude and airspeed while applying a steady bank angle between 60° and 90° and pitching up with small corrections. Aircraft changes its' heading in the meanwhile.

Ascend is simply performed by pitching up command to an aircraft in level flight to gain altitude. **Descend** is performed by pitching down command to an aircraft in level flight to lose altitude.

Reverse flight is performed by rolling the aircraft to a 180° bank angle and keeping altitude and airspeed constant.

Immelman is performed to change aircraft heading 180° in a fast manner. It is performed by completing a 180° vertical climb turn and correcting the bank angle to 0° to go from reverse flight to level flight [44].

Aileron roll is performed by rolling the aircraft at a constant roll rate until 360° is completed and the aircraft is returned to the level flight at the same altitude and airspeed as before [44].

Split-S Split-S is considered to be the reverse of the Immelman maneuver. It is performed by rolling the aircraft from level flight to reverse flight and performing a vertical descending loop by a pitch-up command until the aircraft returns to level flight with a 180° heading difference [44].

Chandelle is performed to gain altitude and change heading at the same time. The aircraft gains altitude and changes heading at constant rates and performed similarly with sustained turn. The only difference is keeping the flight path angle more than 0° to gain altitude [44].

Lazy 8 is the most complex one to perform and classify among all the maneuvers in this paper. The aircraft follows the path to draw the number eight. The aircraft gains altitude while the bank angle is high and the aircraft is turning and returns to level flight at the same altitude as at the start [44].

4.2. Implementation

The steps taken to create and train the neural network models, and develop a real-time flight maneuver classifier program are as below:

1. Preprocessing of the dataset is performed before feature selection. The dataset is normalized using maximum absolute scaling. The features of the dataset are scaled between [-1, 1] after this step. Random shuffling is also performed on the dataset before feature selection.

2. Feature selection is performed using three techniques: Correlation calculation, ANOVA, and being used in single hidden layer neural network models.

The flight dataset includes some data features that are strictly relevant such as horizontal path and heading. Correlation matrices reveal these highly correlated features, and also reveal the correlation between the data features and the maneuver categories. The correlation matrices were created using an open-source Python library, Pandas and they can be seen in Figure 4.1. The indexes in the figure match the indexes in the dataset file. These correlation values range between [0, 1], 1 being the maximum correlation. Most highly correlated data features (exceeding 0.75 on every correlation method) can be seen in Table 4.2.

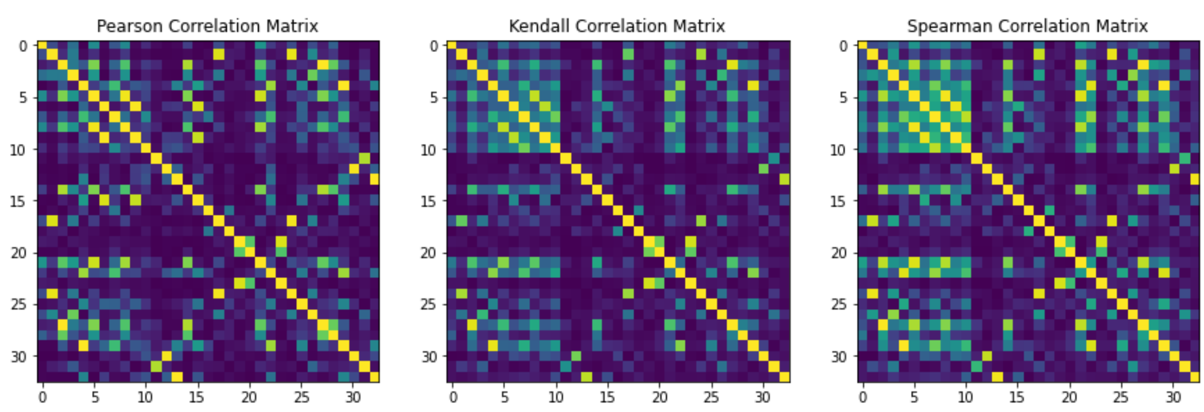


Figure 4.1 Correlation matrices.

Table 4.2 Correlation of data features.

Data feature 1	Data feature 2	Pearson	Kendall	Spearman
8 - elev-surf	5 - elev-stick	0.978	0.908	0.982
9 - ailrn-surf	6 - ailrn-stick	0.993	0.897	0.972
17 - pitch	1 - VVI	0.947	0.760	0.918
21 - alpha	5 - elev-stick	0.924	0.824	0.946
22 - beta	4 - Gload-side	0.852	0.825	0.949
23 - hpath	19 - hding-true	0.943	0.930	0.939
24 - vpath	1 - VVI	0.965	0.869	0.972
24 - vpath	17 - pitch	0.984	0.850	0.958
27 - lift	2 - Gload-norml	0.996	0.954	0.996
29 - side	4 - Gload-side	0.998	0.967	0.998
29 - side	22 - beta	0.856	0.817	0.945
30 - L-lb-ft	12 - L-ftlb	0.984	0.788	0.908
32 - N-lb-ft	13 - N-ftlb	0.999	0.902	0.978

From Table 4.2, we can clearly see that some of our features are highly correlated. To consider a few: elev-surf, elev-stick features, and airn-surf, ailrn-stick features are highly correlated due to control surface value being the direct result of a pilot providing input to the stick. We can also see that the vertical velocity (VVI) is highly correlated with the vertical path. There is no point in including both of these data features in the dataset when training.

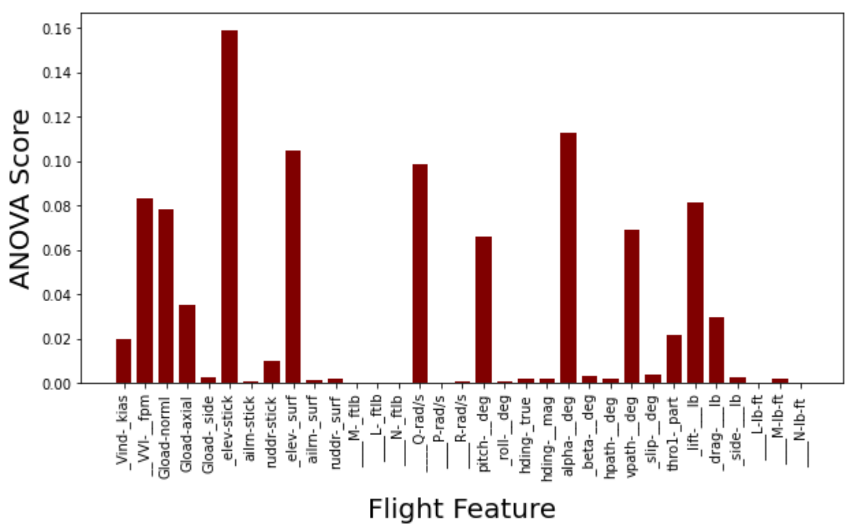


Figure 4.2 ANOVA scores.

The flight maneuver dataset that this study contributes is a numerical input categorical output one. Brownlee (2019) suggests the use of the ANOVA correlation coefficient when the dataset is numerical input and categorical output [51]. Scikit-learn, which is an open-source machine learning and data analysis library for Python, includes feature selection using ANOVA by using the f-test method. ANOVA was used as the primary data feature selection algorithm in this study.

ANOVA scores are calculated after the correlation calculation. The ANOVA scores of the features can be seen in Figure 4.2. The features sorted according to the highest ANOVA scores that do not have a correlation score of more than 0.75 with another data feature are: **'elev-stick'**, **'Q'**, **'lift'**, **'VVI'**, **'Gload-axial'**, **'drag'**, **'thro1'**, **'Vind-'**, **'ruddr-stick'**, **'slip'**, **'beta'**, **'M-lb-ft'**, **'hpath'**, **'ailrn-surf'**, **'hding-mag'**, **'ruddr-surf'**, **'R'**, **'roll'**, **'P'**, **'L-lb-ft'**, **'N'**, **'M'**.

The final feature selection method, invented for this thesis, was to use all the triple combinations of the above data features to train a simple single hidden layer neural network architecture shown in Figure 2.2 with 2 epochs. The resulting accuracy and the loss for each triple combination are logged for each data feature in the triple combination. The accumulated average accuracy and loss for each data feature can be seen in Figure 4.3. The activation function of the hidden layer was rectified linear unit, and the activation function of the output layer was softmax to provide a probability distribution. Loss is calculated using sparse categorical cross-entropy and the Adam algorithm is used as the optimizer.

The data features included and sorted by the success of classifying flight maneuvers are in Table 4.3. The main difference between ANOVA scores and ANN evaluation is the importance of roll is much higher when used in ANN than shown in ANOVA scores due to the artificial neural networks' success in learning complex patterns. We can clearly see that the accuracy rises and the loss lowers when **roll** is included in the dataset. The inverse applies to **heading** and **hpath**. We can see that the importance of heading to classify aircraft maneuvers is lower than the ANOVA score shows. We can clearly see that the results are

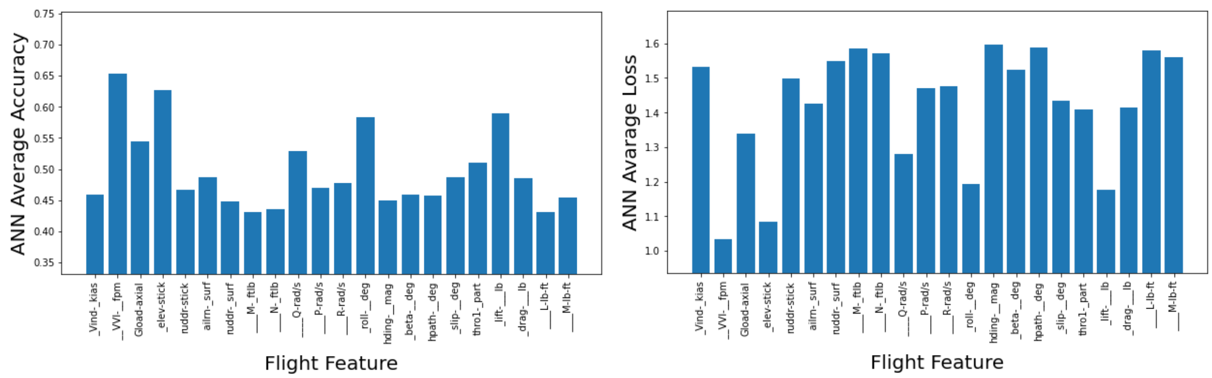


Figure 4.3 ANN scores.

parallel and ANOVA is a fierce data feature selection method for our case, other than these two deviations. Also, the invented feature selection method was also proven to be working.

Table 4.3 Data features sorted by feature selection methods.

ANOVA Score	ANN Avg Acc	Ann Avg Loss
5 - elev-stick	1 - VVI	1 - VVI
14 - Q	5 - elev-stick	5 - elev-stick
27 - lift	27 - lift	27 - lift
1 - VVI	18 - roll	18 - roll
3 - Gload-axial	3 - Gload-axial	14 - Q
28 - drag	14 - Q	3 - Gload-axial
26 - thro1-part	26 - thro1-part	26 - thro1-part
0 - Vind	25 - slip	28 - drag
7 - ruddr-stick	9 - ailrn-surf	9 - ailrn-surf
25 - slip	28 - drag	25 - slip
22 - beta	16 - R	15 - P
31 - M-lb-ft	15 - P	16 - R
23 - hpath	7 - ruddr-stick	7 - ruddr-stick
9 - ailrn-surf	0 - Vind	22 - beta
20 - hding-mag	22 - beta	0 - Vind
10 - ruddr-surf	23 - hpath	10 - ruddr-surf
16 - R	31 - M-lb-ft	31 - M-lb-ft
18 - roll	20 - hding-mag	13 - N-ftlb
15 - P	10 - ruddr-surf	30 - L-lb-ft
30 - L-lb-ft	13 - N-ftlb	11 - M-ftlb
13 - N-ftlb	30 - L-lb-ft	20 - hding-mag
11 - M-ftlb	11 - M-ftlb	23 - hpath

3. Building and training ANNs is the core process of the study. Different artificial neural networks were tested against the flight maneuver classification problem. The types of artificial neural networks used and compared were single hidden layer neural networks, deep neural networks, and recurrent neural networks. Combinations of these types, activation functions, optimization methods, and back-propagation algorithms were tested against the problem to maximize the performance of the solution, in terms of accuracy, precision, and recall.

Python Programming Language is used for the development of the flight maneuver classifier program. Python provides a nice, high-level but fast working environment for machine learning applications. Tensorflow for Python is used as the neural network library in the flight maneuver classification to build and train the models. It is an open-source machine learning library, a very good tool for artificial neural networks.

This step starts with building and training the single hidden layer neural network. Neural networks' foundation lies at the node weights. That is why neural networks are good at weighing the features that are more important according to the loss function. Accuracy and loss are calculated by training an ANN with 128 single hidden layer nodes with ReLU activation function, output layer nodes with softmax activation function, the loss calculation method of sparse categorical cross-entropy, and Adam optimization, one by one by removing the least feature in Table 4.3's average accuracy. Removing the features does not have a considerable positive effect. This situation can be seen in Figure 4.4. This is why all data features are used after removing the ones with a high correlation with another feature and a lower ANOVA score.

$$N_l \in [8, 16, 32, 64, 128, 254] \tag{20}$$

To see the impact of the number of nodes in the hidden layer, the values presented in Equation 20 are tried. The accuracy and loss changing according to the number of hidden layers are in Figure 4.5. Since the 128 nodes and 256 nodes both gave a %99.7

validation accuracy, 128 nodes were selected. These tests were also done with ReLU hidden activation function, softmax output activation function, the loss calculation method of sparse categorical cross-entropy, and Adam optimization.

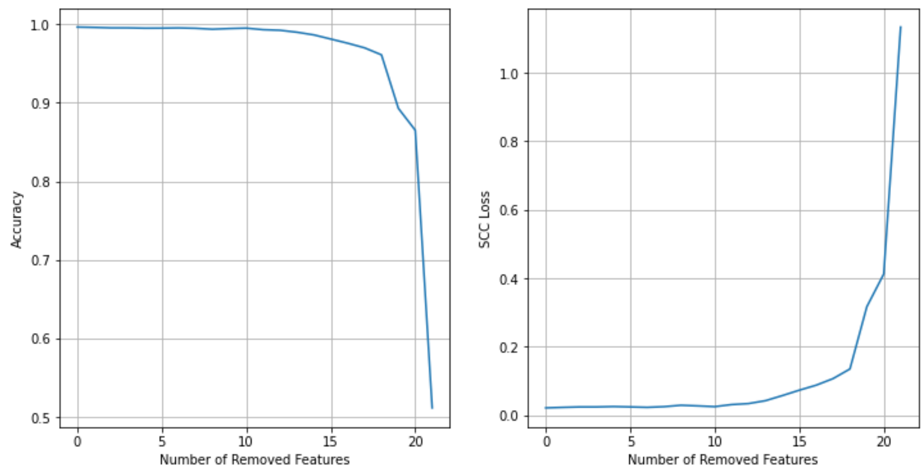


Figure 4.4 Single layer neural network accuracy and loss by removing data features.

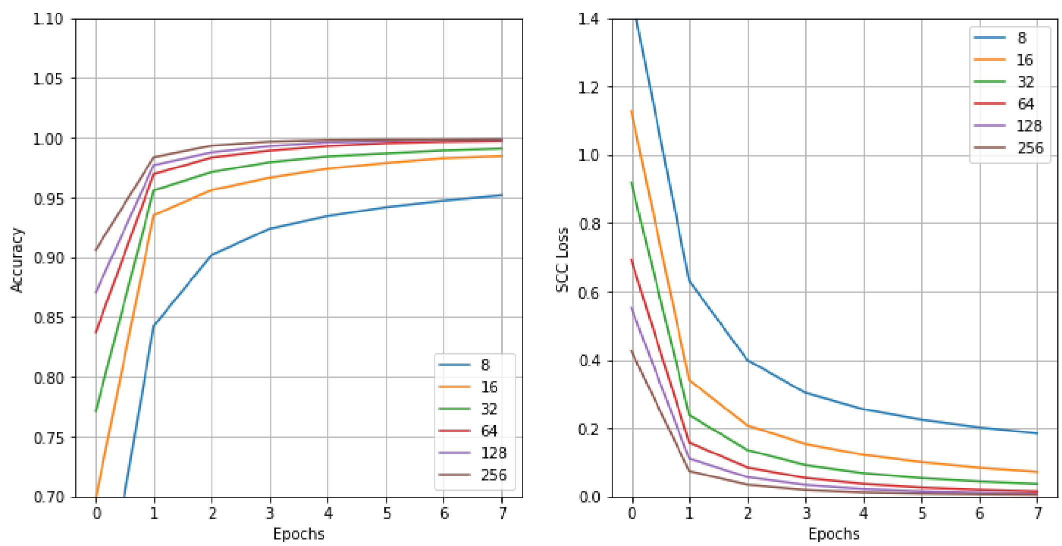


Figure 4.5 Single layer neural network accuracy and loss with different number of hidden nodes.

After selecting the features and the number of hidden nodes, different types of optimizers, hidden layer activation functions, and losses were combined to get the maximum validation accuracy. Note that the output activation function is kept as softmax to give a nice probability

Table 4.4 Optimizer, loss function, and activation functions.

Optimizers	Loss Functions	Activation Functions
SGD	Categorical Cross-entropy	RELU
AdaGrad	Sparse Categorical Cross-entropy	Sigmoid
RMSprop	KL Divergence	Softmax
Adadelta	Mean Squared Error	Softplus
Adam	Mean Absolute Error	Softsign
AdaMax	Cosine Similarity	tanh
NAdam	Huber Loss	SELU
FTRL	Categorical Hinge	ELU

distribution of flight maneuvers. The experimented alternatives are in Table 4.4. The combination of the Adam optimizer, ReLU activation function, and sparse categorical cross-entropy gave the maximum validation accuracy (%99.7). The trials to get the data features and the number of hidden layers were also done with these selections. When the neural network with chosen features and parameters was trained for ten epochs, the accuracy was %99.9, the validation accuracy was %99.86, the loss was 0.0051, and the validation loss was 0.0064.

$$N_l \in [8, 16, 32, 64, 128] \tag{21}$$

$$N_n \in [2, 3, 5] \tag{22}$$

Similarly to the SHLNN case, the combinations of values presented in Equation 21 number of hidden nodes in each hidden layer and values presented in Equation 22 number of hidden layers are tried. These tests were also done with ReLU hidden activation function, softmax output activation function, the loss calculation method of sparse categorical cross-entropy, and Adam optimization. The resulting accuracy and loss can be seen in Figure 4.6. A few options of the combinations converged with almost perfect accuracy. The three hidden layers with 128 hidden nodes were chosen to be the best deep neural network model.

The combination of the Adam optimizer, ReLU activation function, and sparse categorical cross-entropy gave the maximum validation accuracy (%99.7) again for DNN case too.

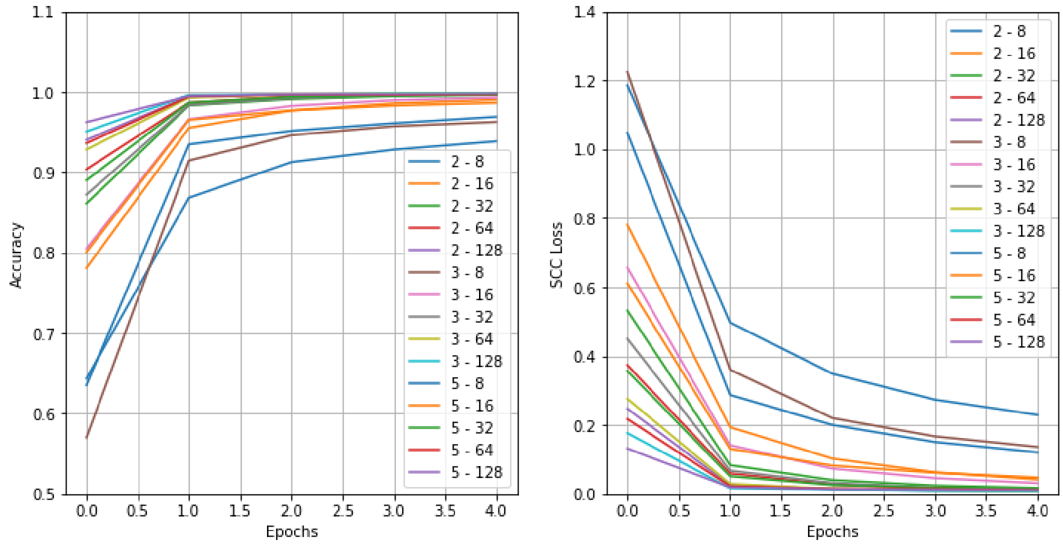


Figure 4.6 DNN accuracy and loss with different number of hidden nodes.

Table 4.5 RNN time frames.

Time Frame	Accuracy	Validation Accuracy	Loss	Validation Loss
0.5 seconds	94.7%	92.5%	0.19	0.26
1 seconds	94.6%	96%	0.19	0.15
2 seconds	93.2%	92.1%	0.24	0.31
3 seconds	94.1%	94.8%	0.21	0.27
4 seconds	92.8%	93.7%	0.24	0.25

Finally, a recurrent neural network with three hidden 128 nodes LSTM layers and 1 hidden 32-node dense hidden layer was created. The optimization method was Adam, the loss function was sparse categorical cross-entropy, and the activation function was rectified linear unit. One thing to consider when working with recurrent neural networks is the sequence length. Determining the time frame can drastically affect the result. The dataset is ordered by time and the period of sampling is 33.3 ms. By finding the sequence length, we are finding the time frame of a sample of a recurrent network. 1 second, 2 seconds, 4 seconds, and 6 seconds were the time frames. The accuracy, validation accuracy, loss, and validation loss are in Table 4.5. The optimal time frame was 1 second for the 30 Hz dataset.

4. Creating a real-time flight maneuver classifier program was the final step of the implementation. The algorithm to classify the maneuver in real-time can be seen in Figure 4.7 below.

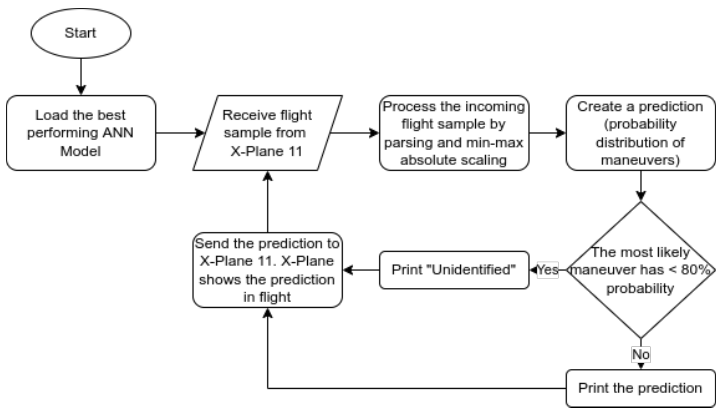


Figure 4.7 The flowchart of the classifier program.

If the most probable classification's probability (maximum valued output of the softmax layer) is less than 0.8, the sample is classified as "Unidentified". "Unidentified" classification generally occurs in maneuver transitions. For example, passing from ascending to level flight, there is a spot in which the neural network gives a high probability to both "ascend" and "level flight", naturally. At this point, it is meaningless to classify the maneuver since the sample does not belong to both of the maneuvers.

5. EXPERIMENTAL RESULTS

The thesis compared the three neural network types using accuracy, loss, precision, and recall. The validation accuracy and the validation loss values were presented in Table (5.2). The neural networks were trained using the main dataset and evaluated using the validation dataset. The information on these datasets can be seen in Table 5.1. The random shuffling of the dataset over and over again while developing showed no over-fitting. Also, two sets of validation data taken at different times were used to get the validation accuracy and performance metrics scores to show the absence of over-fitting. Since the data collection was easy, thanks to the developed data collection program, the independent flights were conducted to get the validation dataset that is not used in training at all. While the validation accuracy and validation loss values mentioned in Chapter 4. were obtained using cross-validation of the main dataset, the accuracy and loss values in Chapter 5. are obtained using independent validation data. Since we do not have a limit on data samples, k-fold cross-validation is not needed. This situation provides a level of confidence in the solution since it shows that the study can classify independent and never seen flight data to show no over-fitting. According to the results shown in Table 5.2, all three types of neural networks were highly successful at classifying ten flight maneuvers. We can clearly see from the validation accuracy and loss values (obtained from the independent validation dataset) that there was no over-fitting.

Table 5.1 Dataset statistics.

Dataset	**Samples**	**Samples per Maneuver**
Main Dataset	70031	9037 6924 10617 5706 9750 4438 4551 3500 5098 10410
Validation Dataset	14163	2008 1883 1319 1076 1546 763 990 582 1281 2715

5.1. Single Hidden Layer Neural Network

The results for the best performing single hidden layer neural network found in Chapter 4. are represented in this section. The confusion matrix can be seen in Figure (5.1) and the

Table 5.2 Results summed up with validation dataset.

ANN Type	**Validation Accuracy**	**Validation Loss**
Single Hidden Layer	96.0 %	0.15
Deep Neural Network	96.4 %	0.22
Recurrent Neural Network	95.6 %	0.24

performance metrics results can be seen in Figure (5.2) for the single hidden layer network model. The only seemingly bad result occurs with the precision of the split-s maneuver. The ”split-s” maneuver is more likely to be predicted when another maneuver is being performed. We can see this from the confusion matrix. While the ”split-s” maneuver is predicted correctly for 550 samples, it is predicted falsely 77 times when the actual maneuver was ”reverse” and 114 times when the actual maneuver was ”descend”. The higher number of false positives caused low precision. This situation is only logical since split-s starts when the aircraft is in reverse state and the aircraft descends while performing this maneuver. Since descend is likely to be predicted as split-s, we can see that the recall of descend move is lower than others with 89 percent due to a higher number of false negatives.

5.2. Deep Neural Network

The results for the best-performing deep neural network found in Chapter 4. are represented in this section. The confusion matrix can be seen in Figure (5.3) and the performance metrics results can be seen in Figure (5.4) for the DNN. The exact same arguments can be made with this type of neural network as in SHLNNs. The confusion matrix and performance metrics results are shared for reproducibility’s sake. We can see that the biggest differences are the precision of lazy-8 being improved with DNNs to an almost perfect with 99.9% and the precision of aileron roll being decreased due to a higher number of false positives (confusion with level flight).

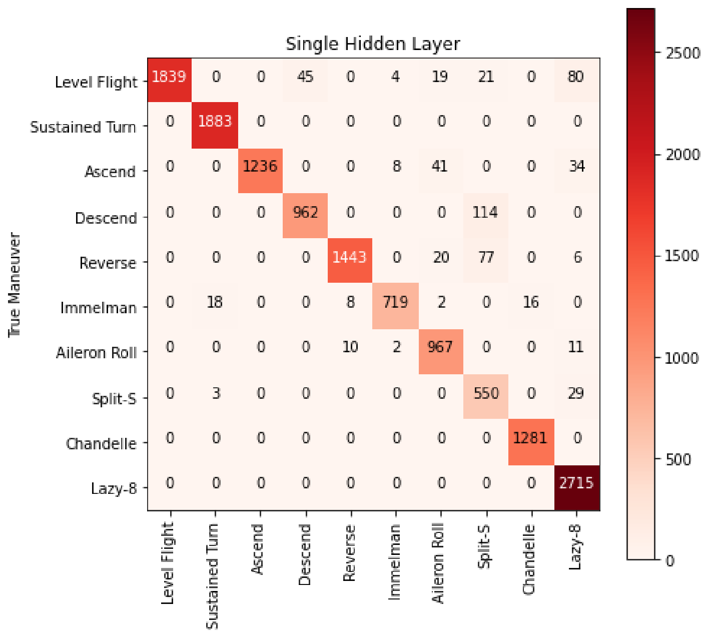


Figure 5.1 Confusion matrix of SHL Neural Network Model.

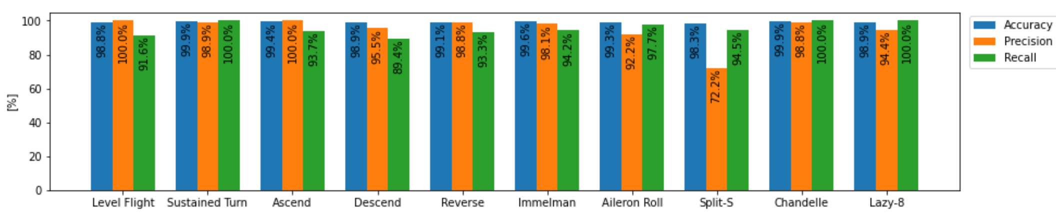


Figure 5.2 Performance metrics of SHL Neural Network Model.

5.3. Recurrent Neural Network

The results for the best-performing recurrent neural network found in Chapter 4. are represented in this section. The confusion matrix can be seen in Figure (5.5) and the performance metrics results can be seen in Figure (5.6) for the RNN model. The first thing to notice is that RNNs are better at classifying split-s maneuvers with more precision since they include the previous samples. The RNNs do not falsely predict a ”reverse” as a ”split-s” like the other ANN types but it still confuses ”descend” with ”split-s”. While split-s precision is higher than other types, it is still the worst result of the bunch. We can also notice that

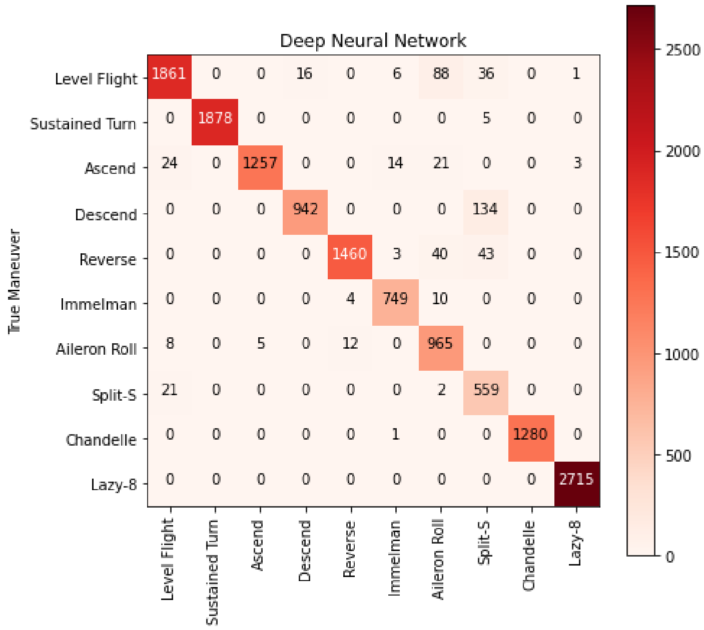


Figure 5.3 Confusion matrix of DNN Model.

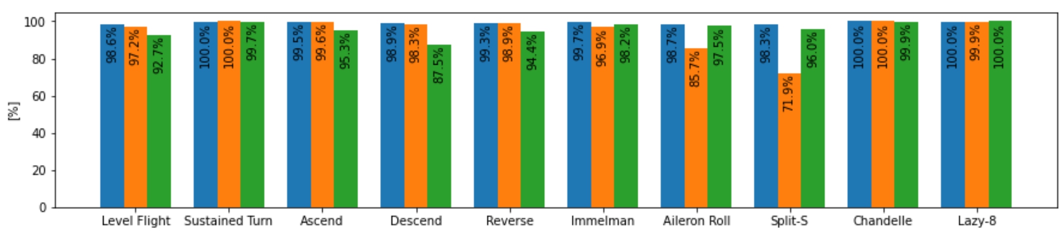


Figure 5.4 Performance metrics of DNN Model.

we have a decrease in ”Immelman” precision with RNNs. The other performance metrics are very similar to the SHLNNs and DNNs. This shows that all the artificial neural network types created similar flight maneuver models and predicted very parallel to each other. One should also note that the number of predictions is 29 less than the number of data samples. This is due to the last sample of this kind of neural network being the time series batch of the last 30 samples of the dataset.

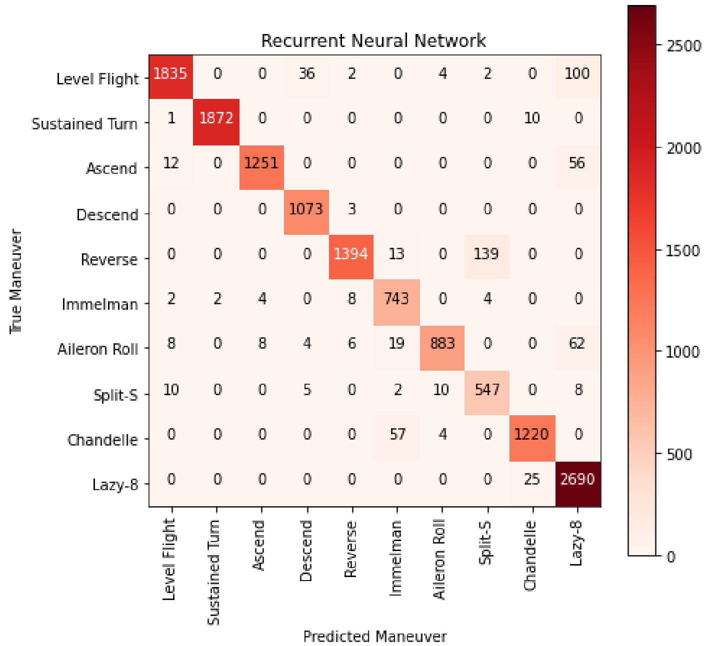


Figure 5.5 Confusion matrix of RNN Model.

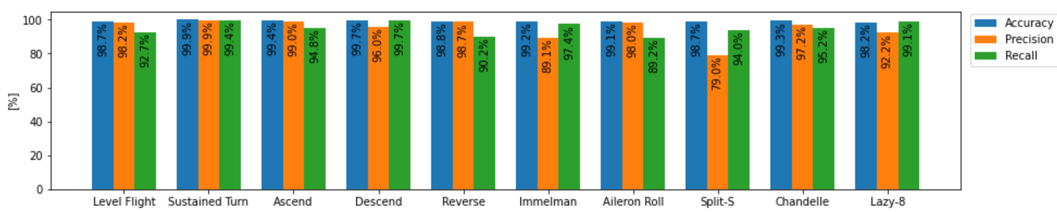


Figure 5.6 Performance metrics of RNN Model.

5.4. Combined Results

The average of the performance metrics results were calculated for each ANN type over every class and they can be seen in Figure 5.7. The maneuver accuracy is calculated using the accuracy values shown in Figures 5.2, 5.4, and 5.6 using Equation 19. We can see that the results are very close to each other. The DNN model is slightly better on recall and total accuracy while almost the same with others on maneuver accuracy and precision. Hence the DNN model is chosen to be the best performing ANN type for flight maneuver classification. Due to this reason, the DNN model was the model used in the real-time flight maneuver classification program. The best performing network model for each ANN type can be seen in Table 5.3.

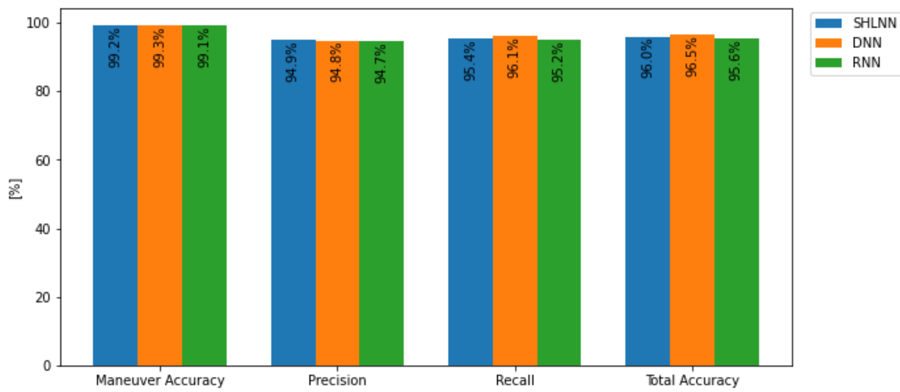


Figure 5.7 Performance metrics of ANNs over every class.

Table 5.3 Best performing models.

ANN Type	Hid. Layers and Neurons	Activation Function	Loss Function	Optimization Method	Time Frame
SHLNN	1 x 128	Hidden: ReLU Output: Softmax	SCCE	Adam	-
DNN	3 x 128	Hidden: ReLU Output Softmax	SCCE	Adam	-
RNN	3 x 128 LSTM + 1 x 32	Hidden: ReLU LSTM: tanh & sigmoid Output: softmax & sigmoid	SCCE	Adam	30 samples

The real-time flight maneuver classification program, which is available at GitHub [16], also was prospering. The program is shared openly for the reproducibility of this study. The program successfully classified and provided feedback to the X-Plane screen in real-time. Figure 5.8 shows a screenshot from the real-time flight maneuver classification program correctly classifying a maneuver.

Figure 5.8 Real-time flight maneuver classification program correctly classifying a maneuver.

5.5. Discussion

Some of the maneuvers in this thesis can also include another one. The models' success to find the more complex maneuver in these cases was happily surprising. For example, the models were successfully classifying the Immelman maneuvers where the altitude of the aircraft raises, just like an ascend. However, due to the lack of harmony, while collecting the data, the low precision in the split-s maneuver is introduced. This lack of harmony is caused by differing starting and ending times of maneuvers and sloppy maneuver executions. This concern was also mentioned by Bodin (2020) in her study [15].

Another concern about this thesis is that the flights to collect the dataset were performed by multiple people, who have little to no simulation flight experience. However, this can also be a strong side, since the trained models classified these flights with high performance, even if they were sloppy.

A thing to consider is the dataset having time jumps. The moments in between the flight maneuvers were not logged while collecting the dataset. While this is not a problem for SHLNNs and DNNs, it can be for RNNs. The first 30 samples of data batches of each maneuver will always have another maneuver's ending samples in this case. The reason for the slight underperforming of RNNs can be this.

In the end, having a program, which is an almost ready product, that is performing the classification in real-time was very satisfying to achieve. It was also very fun to play around with it. Please note that all the contributions introduced in this thesis are shared online [16] for reproducibility's sake. Any researcher can easily reproduce and improve the results of this thesis simply by using the source code, the dataset, and the documentation.

6. CONCLUSION

6.1. Concluding Remarks

This chapter states the summary of the thesis and possible future directions.

The thesis was able to provide means to classify flight maneuvers by exploiting three types of neural networks: Single hidden layer, deep, and recurrent. The best-performing model was found for each of the ANN types and they were compared in terms of validation accuracy, classification accuracy, precision, and recall.

The comparisons showed that each of the three types of neural networks was successful when it comes to flight maneuver classification, while the deep neural network model with 3 hidden layers of 128 neurons was the best with a small margin. Also, this study revealed the best activation functions, loss functions, and optimization methods for each ANN type to be used for flight maneuver classification. The ReLU was the best activation function for a hidden layer and the softmax was the best activation function for an output layer on all three ANN types. The best choice of activation function for an LSTM layer was the combination of tanh and sigmoid functions. The best loss function was sparse categorical cross-entropy and the best optimization method was the Adam method.

The dataset used in this study was also contributed by this study. This study provided means to collect flight data by explaining the process while also contributing an open-source program that is collecting the flight data. The resulting flight dataset was convenient and accurate to be used. The study also contributes the source code that is training the neural networks and the real-time flight maneuver classifier program for anyone that wants to reproduce the results achieved in this study.

The feature selection methods were also exploited. The correlation matrices showed the highly correlated flight features independent of the flight maneuver. The ANOVA feature importance method was used to find the best flight maneuver distinguisher data features. They were pilot elevator command, lift force, Q angular velocity, and vertical velocity.

In addition to ANOVA, a new feature importance method was invented. All the triple combinations of each possible data feature are fed into a simple shallow neural network and resulting accuracy and loss values were accumulated. This method showed that the most important features were vertical velocity, pilot elevator command, and lift force. The same features were found to be important with both methods which provided a level of confidence.

6.2. Future Work

This study can be scaled up simply by increasing the number of flight maneuvers and repeating the steps explained using the available sources. The study also can be scaled up by exploring different neural network types for the problem.

The dataset used in this study is coming from one source. All the flights were performed with one kind of aircraft. A possible future study can scale up this study by including other datasets obtained with different aircraft.

Data imputation methods to incorporate incomplete features in the flight maneuver dataset can be included as further development. These techniques aim to fill in incomplete features using the available data and can be implemented using statistical methods such as mean or median imputation, or machine learning algorithms. Using data imputation may potentially improve the performance of a neural network classifier on the dataset, as the presence of incomplete features can negatively impact the model's ability to learn effectively.

Another direction for future research is the use of neural network techniques for flight envelope detection. The flight envelope defines the limits of an aircraft, such as maximum speed and altitude, and it is important to monitor and enforce these limits to ensure safe operation. Currently, flight envelope detection is often done using rule-based systems. However, these systems can be inflexible and may not be able to adapt to changes in the aircraft or its environment. Machine learning algorithms, on the other hand, have the potential to learn and adapt to such changes, and could potentially provide a more robust and accurate method for flight envelope detection.

REFERENCES

[1] J.P. Jesan. The neural approach to pattern recognition. *Ubiquity*, 2004(April):2, **2004**. doi:10.1145/991108.985625.

[2] J.H. Travert. *Flight regime and maneuver recognition for complex maneuvers*. Master's thesis, Embry-Riddle Aeronautical University, Daytona Beach, **2009**.

[3] G. Barndt, C. Miller, and S. Sarkar. Maneuver regime recognition development and verification for h-60 structural monitoring. **2007**.

[4] Z. Blanks, A. Sedgwick, B. Bone, and A. Mayerchak. Identification of flight maneuvers and aircraft types utilizing unsupervised learning with big data. In *2017 Systems and Information Engineering Design Symposium (SIEDS)*, pages 180–185. **2017**. doi:10.1109/SIEDS.2017.7937712.

[5] P. Dang, P. Tran, S. Alam, and V. Duong. A machine learning-based framework for aircraft maneuver detection and classification. In *14th USA/Europe Air Traffic Management Research and Development Seminar*, volume 202, pages 52–. **2003**.

[6] V. Socha, L. Hanakova, L. Socha, S. Van den Bergh, A. Lalis, and J. Kraus. Automatic detection of flight maneuvers with the use of density-based clustering algorithm. In *2018 XIII International Scientific Conference - New Trends in Aviation Development (NTAD)*, pages 132–136. **2018**. doi:10.1109/NTAD.2018.8551680.

[7] Y. Wang, Q. Han, K. Hu, and D. Zhan. An automatic method to estimate the calibration quality of the aeromagnetic compensation. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 3554–3557. **2019**. doi:10.1109/IGARSS.2019.8898533.

[8] J. Wu, Z. Chen, G. Meng, and H. Zhang. Research on high precision terrain dynamic loading technology based on flight trajectory prediction. In *2018 IEEE*

4th Information Technology and Mechatronics Engineering Conference (ITOEC), pages 682–687. **2018**. doi:10.1109/ITOEC.2018.8740607.

[9] G. Meng, M. Zhou, H. Zhang, and D. Sun. Threat assessment for rotte based on cooperative tactical recognition. In *2019 IEEE International Conferences on Ubiquitous Computing & Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS)*, pages 490–494. **2019**. doi:10.1109/IUCC/DSCI/SmartCNS.2019.00107.

[10] Y. Chen, X. Sun, D. Liu, and S. Li. Optimal guidance method for ucav in close free air combat. In *2019 IEEE International Conferences on Ubiquitous Computing & Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS)*, pages 356–360. **2019**. doi:10.1109/IUCC/DSCI/SmartCNS.2019.00086.

[11] M. Ruotsalainen, J. Jylha, J. Vihonen, and A. Visa. A novel algorithm for identifying patterns from multisensor time series. In *2009 WRI World Congress on Computer Science and Information Engineering*, volume 6, pages 100–105. **2009**. doi:10.1109/CSIE.2009.958.

[12] Z. Wei, D. Ding, H. Zhou, Z. Zhang, L. Xie, and L. Wang. A flight maneuver recognition method based on multi-strategy affine canonical time warping. *Applied Soft Computing*, 95:106527, **2020**. ISSN 1568-4946. doi:https://doi.org/10.1016/j.asoc.2020.106527.

[13] L. Pechaud and Y. Kim. Improved methodology for the prediction of the empennage maneuver in-flight loads of a general aviation aircraft using neural networks. **2001**.

[14] N.C. Oza, Kagan Tumer, Irem Y. Tumer, and Edward M. Huff. Classification of aircraft maneuvers for fault detection. In *Multiple Classifier Systems*,

pages 375–384. Springer Berlin Heidelberg, Berlin, Heidelberg, **2003**. ISBN 978-3-540-44938-6.

[15] C. Bodin. *Automatic Flight Maneuver Identification Using Machine Learning Methods*. Master's thesis, Dept. Elect. Eng. Linköping University, Linköping, **2020**.

[16] F. Pusat. Flight maneuver classification, **2022**. `https://github.com/fikricanpusat` (Accessed: December 2022).

[17] B.A. Kitchenham and S. Charters. Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report, **2007**.

[18] K. Pearson. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240–242, **1895**. ISSN 03701662.

[19] C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, **1904**. ISSN 00029556.

[20] M.G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, **1938**. ISSN 00063444.

[21] R.A. Fisher. Studies in crop variation. i. an examination of the yield of dressed grain from broadbalk. *The Journal of Agricultural Science*, 11(2), **1921**. doi:10.1017/S0021859600003750.

[22] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, **1958**.

[23] F. Chollet. *Layer Activation Functions*. Keras, **2015**. `https://keras.io/api/layers/activations/` (Accessed: December 2022).

[24] S.S. Haykin. *Neural Networks and Learning Machines*. Neural networks and learning machines. Prentice Hall, Upper Saddle River, New Jersey 07458, 3 edition, **2009**.

[25] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, **2016**. `http://www.deeplearningbook.org`.

[26] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, **1989**. ISSN 0893-6080. doi:https://doi.org/10.1016/0893-6080(89)90020-8.

[27] J. Brownlee. Softmax activation function with python, **2020**. `https://machinelearningmastery.com/softmax-activation-function-with-python/` (Accessed: December 2022).

[28] H.S. Das and P. Roy. Chapter 5 - a deep dive into deep learning techniques for solving spoken language identification problems. In N. Dey, editor, *Intelligent Speech Signal Processing*, pages 81–100. Academic Press, **2019**. ISBN 978-0-12-818130-0. doi:https://doi.org/10.1016/B978-0-12-818130-0.00005-2.

[29] S. Amidi and A. Amidi. Recurrent neural networks cheatsheet. `https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks` (Accessed: December 2022).

[30] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, **1997**. ISSN 0899-7667. doi:10.1162/neco.1997.9.8.1735.

[31] J. Brownlee. Loss and loss functions for training deep learning neural networks, **2020**. `https://machinelearningmastery.com/` (Accessed: December 2022).

[32] F. Chollet. *Losses*. Keras, **2015**. `https://keras.io/api/losses/` (Accessed: December 2022).

[33] S. Kullback and R.A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, **1951**.

[34] C.J. Willmott and K. Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate Research*, 30:79–82, **2005**.

[35] P.J. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35:492–518, **1964**.

[36] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, **1986**.

[37] G. Sanderson. Backpropagation calculus, **2017**. `https://www.3blue1brown.com/lessons/backpropagation-calculus` (Accessed: December 2022).

[38] S. Ruder. An overview of gradient descent optimization algorithms. *ArXiv*, abs/1609.04747, **2016**. doi:10.48550/ARXIV.1609.04747.

[39] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12(null):2121–2159, **2011**. ISSN 1532-4435.

[40] M.D. Zeiler. Adadelta: An adaptive learning rate method. *ArXiv*, abs/1212.5701, **2012**.

[41] G. Hinton. Lecture notes in neural networks for machine learning lecture 6a overview of mini-batch gradient descent, **2012**.

[42] D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, **2015**.

[43] T. Dozat. Incorporating Nesterov Momentum into Adam. In *Proceedings of the 4th International Conference on Learning Representations*, pages 1–4. **2016**.

[44] K. Samuel, V. Gadepally, D. Jacobs, M. Jones, K. McAlpin, K. Palko, B. Paulk, S. Samsi, H. Chit Siu, C. Yee, and J. Kepner. Maneuver identification challenge.

In *2021 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–7. **2021**. doi:10.1109/HPEC49654.2021.9622788.

[45] J. Qu, M. Lv, Y. Yang, and Y. Tang. Flight motion recognition method based on multivariate phase space reconstruction and approximate entropy. In *2021 40th Chinese Control Conference (CCC)*, pages 7247–7253. **2021**. doi:10.23919/CCC52363.2021.9550605.

[46] T.I. Baumgartner and U. Maeder. Trajectory prediction for low-cost collision avoidance systems. In *2009 IEEE/AIAA 28th Digital Avionics Systems Conference*, pages 1.C.5–1–1.C.5–8. **2009**. doi:10.1109/DASC.2009.5347567.

[47] T.L. Molloy and J.J. Ford. Hmm relative entropy rate concepts for vision-based aircraft manoeuvre detection. In *2013 Australian Control Conference*, pages 7–13. **2013**. doi:10.1109/AUCC.2013.6697240.

[48] J. Jylhä, M. Ruotsalainen, T. Salonen, H. Janhunen, I. Venäläinen, A. Siljander, and A. Visa. Link between flight maneuvers and fatigue. In J. Komorowski, editor, *ICAF 2011 Structural Integrity: Influence of Efficiency and Green Imperatives*, pages 453–463. Springer Netherlands, Dordrecht, **2011**. ISBN 978-94-007-1664-3.

[49] Laminar Research. *How X-Plane works*. `https://www.x-plane.com/` (Accessed: December 2022).

[50] Laminar Research. *Data set output table*. `https://www.x-plane.com/kb/data-set-output-table/` (Accessed: December 2022).

[51] J. Brownlee. How to choose a feature selection method for machine learning, **2019**. `https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data` (Accessed: December 2022).