A Machine Learning Approach to Flight Safety Event Prediction

T. Harms

Technische Universiteit Delft

TU Delft Delft University of Technology

Challenge the future

Courtesy cover image: www.flickr.com

A Machine Learning Approach to Flight Safety Event Prediction

by

T. Harms

in partial fulfillment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at the Delft University of Technology,
to be defended publicly on Thursday March 26, 2020 at 12:30.

Supervisor:	Dr. O.A. Sharpans'kykh	TU Delft
Thesis committee:	Dr. ir. B. F. Dos Santos	TU Delft
	Dr. ir. G. La Rocca	TU Delft

This thesis is confidential and cannot be made public until March 31th, 2022.

TU Delft Delft University of Technology

TABLE OF CONTENTS

List of Acronyms iii

List of Figures iv

List of Tables v

I Paper 1

II Literature Study 19

1 Introduction 21

2 Safety Management 23
2.1 Evolution of Safety Management . . . 23
2.2 Modern Safety Management . . . 23
2.3 Safety Management System . . . 24
2.4 Safety Data Analysis . . . 25

3 Statistical Methods 27
3.1 General Classification . . . 27
3.2 Parametric Methods . . . 27
3.3 Non-Parametric Methods . . . 28
3.4 Applications of Statistical Methods . . . 28
3.5 Applicability of Statistical Methods . . . 30

4 Machine Learning 33
4.1 General Classification . . . 33
4.2 Domain-Based Algorithms . . . 36
4.3 Reconstruction-Based Algorithms . . . 37
4.4 Distance-Based Algorithms . . . 39
4.5 Rule-Based Algorithms . . . 41
4.6 Ensembled Methods . . . 42
4.7 Method Selection Criteria . . . 43
4.8 Quality Assessment . . . 44
4.9 Feature Selection . . . 46

5 Applications of Machine Learning Techniques 51
5.1 Applications in Safety Data Analysis . . . 51
5.2 Conclusive Remarks . . . 57
5.3 Research Gap . . . 59

6 Research Proposal 61
6.1 Research Problem . . . 61
6.2 Research Objective . . . 61
6.3 Research Questions . . . 61
6.4 Research Methodology . . . 63
6.5 Research Planning . . . 66

III Appendices **73**

A Unstable Approach Definition **75**

B Method Selection **79**

C Feature Selection **83**

D Early Stopping Verification **88**

E Utility Function Verification **92**

F Oversampling Analysis **97**

G Loss Analysis **100**

H Sampling Technique Verification **104**

LIST OF ACRONYMS

AI	Artificial Intelligence
AMC	Acceptable Means of Compliance
AMSL	Above Mean Sea Level
ASR	Air Safety Report
ATC	Air Traffic Control
BPN	Back Propagation Network
CATS	Causal Model for Air Transport Safety
CPN	Counter Propagation Network
DRNN	Deep Recurrent Neural Network
EASA	European Aviation Safety Agency
EI	Expected Improvement
FAA	Federal Aviation Administration
FC	Flight Crew
FDM	Flight Data Monitoring
FOQA	Flight Operations Quality Assurance
GEN	General
GM	Guidance Material
GMM	Gaussian Mixture Model
GRU	Gated Recurrent Units
HMM	Hidden Markov Models
IATA	International Air Transport Association
ICAO	International Civil Aviation Organisation
IFR	Instrument Flight Rules
IMC	Instrument Meteorological Conditions
k-NN	k-Nearest Neighbours
LCS	Longest Common Sequence
LTSM	Long Short Term Memory
MCC	Matthews Correlation Coefficient
METAR	Meteorological Aerodrome Report
MIL	Multiple Instance Learning
MKAD	Multi Kernel Anomaly Detection
ML	Machine Learning
MLP	Multi Layer Perceptron
MSE	Mean Square Error
NN	Neural Network
ORO	Organisation Requirements for Air Operations
PCA	Principal Component Analysis
POI	Probability of Improvement
QAR	Quick Access Recorder
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SMS	Safety Management System
SOP	Standard Operating Procedures
SVM	Support Vector Machine
UCB	Upper Confidence Bound
VFR	Visual Flight Rules
VMC	Visual Meteorological Conditions

LIST OF FIGURES

1.1 Document synthesis. . . . 22

2.1 Evolution of safety management. . . . 23
2.2 A data-based safety management system. . . . 26

3.1 Example of a GMM. . . . 28
3.2 Concepts of correlations and copulas. . . . 29
3.3 CATS architecture. . . . 29

4.1 Supervised learning process. . . . 34
4.2 Linear support vector machine for two class classification. . . . 36
4.3 Quadratic Kernel. . . . 37
4.4 Network diagram for a two layer Neural Network. . . . 38
4.5 Simple clustering algorithm. . . . 39
4.6 Progress of a K-means algorithm for a two-class classification. . . . 40
4.7 A Nearest Neighbour algorithm for a two-class classification. . . . 41
4.8 Two perspectives of a Decision Tree. . . . 42
4.9 Graphical representation of the interpretability-flexibility trade-off. . . . 45

5.1 ClusterAD methodology. . . . 54
5.2 ADOPT methodology. . . . 56
5.3 Airline planning. . . . 59

6.1 Data processing steps. . . . 63
6.2 General framework of the research. . . . 65
6.3 Thesis planning Gantt chart. . . . 67

C.1 Different approaches to feature selection. . . . 83
C.2 Accumulated weights for different fold settings. . . . 85
C.3 Accumulated weights of ten different feature selection iterations. . . . 86

D.1 Non-regularised loss plots. . . . 89
D.2 Convergence plots with undersampling. . . . 90

E.1 All probed points for $R_o = 0.7$. . . . 93
E.2 Bayesian optimisation at 20 iterations. . . . 94
E.3 Bayesian optimisation at 25 iterations. . . . 94
E.4 Bayesian optimisation at 30 iterations. . . . 95
E.5 Bayesian optimisation at 35 iterations (converged). . . . 95

F.1 Precision-recall curves with oversampling. . . . 98

G.1 Logarithmic losses for $y = 1$ as true label. . . . 101
G.2 General description of loss behaviour. . . . 101
G.3 Loss plots. . . . 102

H.1 Precision-recall curves with undersampling. . . . 105
H.2 Loss plots of undersampled experiments. . . . 106

LIST OF TABLES

2.1 Summary of two different approaches to safety. . . . 24

4.1 Overview learning categorisation. . . . 33
4.2 Hypotheses for model testing. . . . 44

5.1 Overview of applications in Safety Data Analysis. . . . 57
5.2 Feature selection per research. . . . 58
5.3 Data size per research. . . . 58

6.1 Method trade-off. . . . 65
6.2 Milestones and dates. . . . 66
6.3 Deliverables. . . . 66

A.1 VMC criteria according to ICAO. . . . 76
A.2 Airspace definitions. . . . 77

B.1 Method scores. . . . 80
B.2 Overview of applications in Safety Data Analysis. . . . 80
B.3 Nature of data criteria scores. . . . 80
B.4 Feasibility criteria scores. . . . 81
B.5 Total score assigned per method. . . . 81

E.1 Probed points by Expected Improvement utility function for $R_o = 0.7$. . . . 93

F.1 Best probed *MCC* iteration per experiment. . . . 97

H.1 Best *MCC* per undersampling experiment. . . . 104

Part I

Paper

A Machine Learning Approach to Flight Safety Event Prediction

T. Harms, *Master Student, Delft University of Technology*
Dr. O.A. Sharpans'kykh, *Assistant Professor*, *Delft University of Technology*
ir. S.C.M. Fakkert, *Airline Safety Department Representative*

***Abstract*—Safety occurrences in the aviation industry are nowadays commonly regarded as the outcome of a complex system. Due to this systemic view on safety airlines pursue to understand this complex, underlying system and aim to proactively act upon the occurrence of these events. The most prevalent implementation of flight safety event detection is however still threshold analysis, which has no such implications. On the other hand, Machine Learning methods have readily proven to be an efficient and valuable solutions in predicting the occurrence of anomalies in data, such as flight safety events. However, existing methods search for anomalies in datasets encompassing the anomaly, i.e. direct datasets. On the contrary, this study approached airline operations as a complex system which the outcome could be the occurrence of a flight safety event. Hence, the question was raised whether a set of indirect precursors could be significant in predicting flight safety events. That is why common airline processes were selected, in consultation with industry experts, and their indirect data considered. The aim of this study was to evaluate this concept by evaluating a set of precursors for a particular flight safety event (a case study). The Knowledge Discovery in Databases framework was the general guideline throughout this research, with a Relief and Neural Network algorithm as transformation and data mining step respectively. This study showed that the considered processes were significant in predicting the occurrence of a safety event, although the found precursors could not fully encompass the event under investigation. The classification performance of the methodology was characterised by a large number of false positives, which originated from the problem's class skewness. The Matthews Correlation Coefficient proved to be a well-balanced optimisation objective for such problems and overcame this drift. Locally, the weight optimisation showed a set of confidently classified false positives and negatives confined further improvement. These misclassifications were found to be the result of a lack of adequate information. Nevertheless, the considered information did display to be significant as the obtained Matthews Correlation Coefficient and recall underpinned, particularly in the light of the class imbalance and the anomalous nature of flight safety events.**

Keywords: Safety Data Analysis, Unstable Approaches, Precursor Mining, Anomaly Prediction, Aviation Safety, Machine Learning

I. Introduction

Safety management within the aviation industry has gradually evolved over the past decades. This evolution was initiated when commercial aviation emerged which induced rapid technological advancements. These developments shifted the focus from technical failures to human error as humans were exposed to increasingly complicated systems and circumstances. The subsequent era predominantly focused on individuals and did not to address their convoluted surroundings. With the ever increasing complexity of the operational environment this perspective could not persist. Eventually the environment itself was acknowledged as a contributing factor to the occurrence of safety events. That is why in modern day safety management an incident or accident is regarded as the outcome of a complex system [1].

Understanding the complex, underlying system is currently the ambition of airlines for the purpose of proactively taking mitigating actions. This pursuit originates from a proactive attitude towards safety which has found support in the industry [2]. Flight safety events are however still monitored by means of threshold analysis, with little, data-based proactive implications. Furthermore, this approach to safety data analysis does not align with the systemic view of modern safety management as the complex system is far from considered [3]. The industry is therefore looking for systemic, proactive analyses that fulfil their pursuits.

In academic literature many efforts in the field of safety data analysis are readily presented. These problems were approached using various data analysis techniques. In general, these techniques can be divided into two groups, statistical and Machine Learning techniques. The latter have become popular due to two major limitations of statistical methods, namely their need for prior knowledge and their inability to cope with dimensional data [4, 5, 6]. Machine Learning methods overcome these limitations and have for that reason obtained a prominent position in this research area [7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]. Nonetheless, applications of statistical methods are presented in literature [18, 19, 20, 21, 22].

Flight safety events are, in terms of data analysis, essentially anomalies, as *anomalies are patterns in data that do not conform to a well defined notion of normal behaviour* [23]. Machine Learning techniques have proven to be valuable and efficient solutions in detecting or predicting anomalies and nowadays seem a versatile solution to any data analytic task. However, the applicability of these methods is subject to a crucial and very delicate trade-off process [24]. This difficulty is simultaneously its strength as, if traded off properly, a suitable method could be found for any problem.

Currently, there exist methods that detect or predict anomalies in datasets encompassing the anomaly. These datasets are called direct datasets and contain the information required to evaluate the occurrence or severity of the flight safety event. In this case flight data is therefore often considered [7, 8, 9, 10, 11, 13, 14, 25, 26, 27, 28]. These researches do not adhere to the modern view on safety management as the complex environment is not acknowledged, similar to the present industry standard. In that perspective the industry's

standard and the currently available methods have a commonality, as both do not approach the occurrence of flight safety events systemically. Hence, this raised the following research question: *could a systemic, data-based approach yield a set of indirect precursors for predicting the occurrence of flight safety events?*

To answer this question this study approached airline operations as a complex system of which the outcome could be a flight safety event. It was therefore hypothesised that the probability of flight safety events occurring is affected by indirect processes and that their datasets could compromise precursors. These datasets were selected in consultation with industry experts. The aim of this study was to establish a set of precursors for a particular flight safety event. That is why a case study was undertaken to evaluate this concept. This case, unstable approaches, was subjected to a methodology based on the Knowledge Discovery in Databases (KDD) framework. The most noteworthy steps of this approach are the transformation and data mining, which were implemented with a Relief and Neural Network algorithm respectively [29].

This article is structured as follows. Section II will first specify the case study. Section III will then explain the methodology this case was subjected to. Subsequently, Section IV will discuss the obtained results, which are further discussed in Section V. Section VI will conclude this paper and suggest opportunities for future work.

II. Case Study

The case study for this research was chosen based on widespread concern in the industry. This concern is the occurrence of unstable approaches since this flight safety event is believed to be associated with more serious incidents and therefore has seen an increase in attention within in the industry [30]. According to *IATA* an unstable approach is defined as *any approach that does not meet the stabilised approach criteria defined by the operator in its SOPs* [30]. The Standard Operating Procedures (SOPs) are operator defined guidelines regarding speed, descent rate, thrust setting and configuration. An example of these guidelines can be found in *Appendix A*. These criteria aim to ensure a safe approach and are to be met at a so called stabilisation height, defined by the operator. The stabilisation height is defined as the altitude at which approach stability is established, i.e. the last altitude one of the SOPs is violated. The airline under investigation prescribes the following criteria: Any flight should be stabilised at either 500 ft in Visual Meteorological Conditions (VMC) or at 1,000 ft in Instrument Meteorological Conditions (IMC). The criteria for VMC conditions are summarised by Table I, a detailed description of these criteria is given in *Appendix A*.

TABLE I: VMC criteria according to ICAO [31].

Altitude band	Airspace	Visibility	Distance from clouds
At and above 10,000 ft AMSL	A to G	> 8km	$> 1,500$m horizontally > 300m vertically
Below 10,000 ft AMSL or 1,000 ft above terrain	A to G	> 5km	$> 1,500$m horizontally > 300m vertically
At and below 3,000 ft AMSL or 1,000 ft above terrain	A to E	> 5km	$> 1,500$m horizontally > 300m vertically
	F and G	> 5km	Clear of cloud and surface in sight

Based on the stabilisation height and the corresponding weather parameters seen in Table I, the two classes, stable and unstable, were derived. A histogram of the stabilisation height subsequently proved that unstable approaches are to be regarded as anomalies, because of their scarcity and their position in the distribution, i.e. the majority of occurrences is located beyond two standard deviations.

III. Methodology

The methodology of this research is in line with the Knowledge Discovery in Databases (KDD) framework. This scheme was chosen as there exists no Machine Learning methodology and the KDD methodology was in fact regarded a structured and unified approach of commonly seen methodologies in literature. Furthermore, its fundamental aim is well aligned with this study, namely *mapping low-level data too voluminous to understand and digest into other forms that might be more useful* [29]. Figure 1 summarises the steps of the KDD framework as well as the sections where each step is explained.

A. Data Selection and Preprocessing

The first step in this framework determined the validity of the to be selected dataset regarding the aim of this research, namely approaching the occurrence of a flight safety event systemically. This selection was conducted in consultation with industry experts. The first dataset was selected in the area of airline planning, namely pairings. Fundamentally, the process of airline planning consists of four sequential subprocesses, as Figure 2 depicts. The dataset at hand was deemed sufficiently representative for the purpose of this study as parameters originating from all these processes were present. The eventual result of these processes is a pairing, i.e. a sequence of flights or legs for a crew or crew member, and the data was structured as such. In addition, weather data was added to this analysis as safety data experts were convinced that the occurrence of safety events could be affected by weather conditions. This dataset was provided in the form of METAR reports, which were stripped according to the industry-wide conventions. This dataset was matched to the above pairings based on the landing time and airfield.

The selected data was subsequently preprocessed, which mostly consisted of cleansing the data by removing faulty, unrepresentative and incomplete flights. Afterwards, an Exploratory Data Analysis was conducted to familiarise with the characteristics of the dataset [33].

B. Feature Selection

Before choosing a Machine Learning method, it is important to consider whether all selected and preprocessed data is in fact useful to the investigation, as the dimensionality of the dataset affects the time to train and predictive performance [34, 35, 36]. A widely accepted technique to reduce the dimensionality of a dataset is feature selection. Feature selection methods are generally subdivided into three major categories, according to their interaction with the learning algorithm. These categories are filters, wrappers and embedders [37]. Considering the size

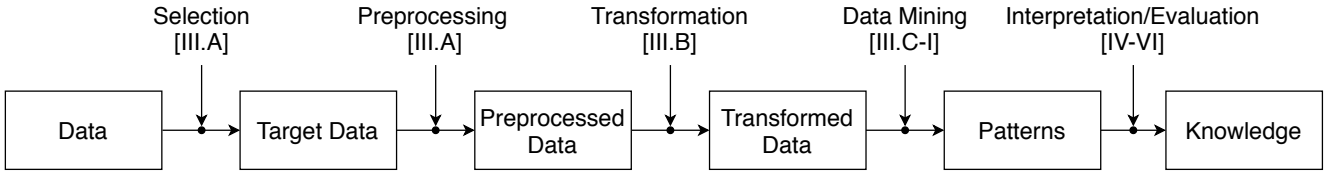


Fig. 1: Knowledge Discovery in Databases approach [29].

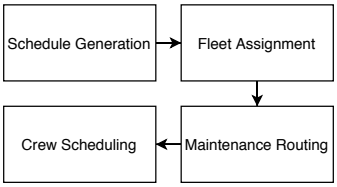


Fig. 2: Airline planning [32].

of the dataset at hand ($> 10^8$) and the available computational resources, a filter method was deemed the most viable solution. Filter methods function independent of the learning algorithm, in contrast with wrappers and embedders. Hence, these methods only require a limited number of iterations and are therefore computationally efficient. Filters are furthermore commonly applied in related literature [7, 8, 16, 17, 25, 26].

Amongst the observed filter algorithms Principal Component Analysis (PCA) and Relief were profound in the field of safety data analysis [7, 8, 16, 17, 25, 26]. For this particular problem Relief had two main advantages, namely its ability to deal with both continuous and categorical data as well as its supervised fashion. PCA can only consider continuous data and therefore cannot establish the same dimensionality reduction given the highly heterogeneous dataset at hand. Furthermore, a supervised feature selection algorithm does not implicitly assume that each flight safety event is to be explained by the same feature set, an assumption questioned by experts and researchers [13]. In other words, Relief's supervised nature tailors the feature set for each flight safety event. However, this benefit has a general computational disadvantage as feature selection has to be performed for each case. On the contrary, PCA yields a fixed set of features, under the condition that the size of the dataset is nearly constant. This condition is, in an operational setting, undoubtedly violated as the considered dataset is in fact growing rapidly. An elaborate description of feature selection and the working principle of Relief, including a mock-up algorithm, can be found in *Appendix B*.

C. Method Selection

The features were then used as inputs to a yet to be selected Machine Learning algorithm. The six most prominent criteria for selecting such algorithms were derived from the literature and these are listed below. Considering multiple processes lead to a large, heterogeneous and multi-dimensional dataset, a situation best coped with by Regression Trees or Neural Networks [5, 24]. The feasibility criteria were evaluated subsequently, but in a weighted manner as the interpretability criterion was disregarded. For this study proving the concept was a fundamental industry requirement, more meaningful than understanding the results to an extent Regression Trees renowned for [24, 38]. That is why the decisive role of the interpretability criterion was undesired and this criterion disregarded. Neural Networks were consequently chosen over Regression Trees, in particularly as the former had also readily proven its reliability in safety critical domains [4, 39]. A more elaborate discussion on the conducted model trade-off, including the scores and all studied algorithms, can be found in *Appendix C*. A Multi-Layer Perceptron (MLP) architecture was subsequently chosen since it is the most common architecture [40]. This architecture is characterised by links (weights) between all neurons (activation functions).

Nature of Data	**Feasibility**
– Scalability	– Complexity
– Heterogeneity	– Interpretability
– Dimensionality	– Robustness

D. Hyperparameter Identification and Tuning

Each learning algorithm has a set of parameters that affect its overall performance, called hyperparameters. Goodfellow et al. define hyperparameters as *several settings that we can use to control the behaviour of the learning algorithm* [41]. A neural network has the following hyperparameters:

Neural Network Layout
- Number of layers N_L.
- Number of units per layer N_U.
- Activation functions.

Weight Optimisation
- Optimisation algorithm.
- Gradient evaluation scheme.
- Learning rate.
- Loss function.
- Number of epochs N_E.

The following assumptions were made regarding the hyperparameters:

Neural Network Layout
- The number of units for all hidden layers is constant.
- The activation functions are rectified linear and Sigmoid for the hidden layers and final layer respectively.

Weight Optimisation

- The weight optimisation algorithm is ADAM.
- The learning rate is fixed at 0.001.
- The loss function is binary cross-entropy.
- The weight gradient is computed by backpropagation.

The layout assumptions reduced the hyperparameter space by twice the number of hidden layers. The activation functions and number of units would otherwise both have been a function of the number of hidden layers. Hence, this assumption saved computational resources and furthermore established a consistently sized input vector. Consequently, only two layout related hyperparameters remained: the number of layers N_L and the number of units N_U. The rectified linear activation functions were chosen for the hidden layers as these do not suffer from the vanishing gradient problem, which is particularly important in deep neural networks [42]. The final layer consisted of a single unit with a Sigmoid activation function, which is common for binary classification tasks. In contrast with the activation functions, there is little consensus with regards to the weight optimisation algorithms. According to literature ADAM is one of the most commonly applied algorithms, presumably because this optimiser was developed for highly dimensional problems [41, 43]. The learning rate was fixed at 0.001 accordingly, as suggested by the developers of ADAM [43]. Lastly, the loss function and weight gradient computation scheme were chosen according to best practises, considering both have proven to be efficient in prior applications [44, 45]. Figure 3 summarises the architecture of the Neural Network and depicts the two of the tuneable layout hyperparameters, N_U and N_L. Note that the size of the input layer N_F is not a hyperparameter but the result of the feature selection and that the number of epochs N_E is also still a tuneable hyperparameter.

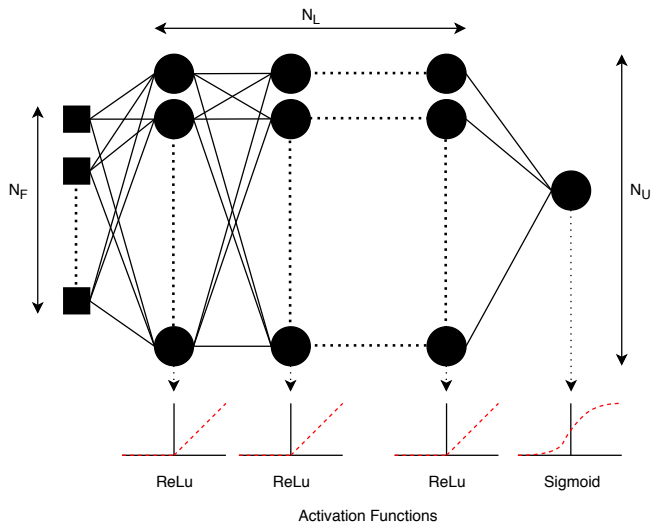


Fig. 3: Multi-Layer Perceptron Neural Network.

E. Hyperparameter Optimisation

The forthcoming hyperparameter optimisation problem is characterised by two important decisions: the optimisation strategy and the optimisation metric. Choosing the latter is a delicate task as a consequence of the potential bias, induced by the class imbalance. Accuracy for instance is a very poor measure for this task since predicting no unstable approaches would make any algorithm instantly 97% accurate [46]. Similarly, often used classification measures such as recall (Equation (1)) and precision (Equation (2)) would be biased as these only consider two aspects of the confusion matrix (Table II). Contrarily, the Matthews Correlations Coefficient (MCC) accounts for all aspects of this matrix, as Equation (3) shows [47]. This metric is therefore a more balanced performance measure and is for that reason commonly used for imbalanced classification tasks. Hence, this metric was chosen as objective of the hyperparameter optimisation. The MCC ranges from -1 to 1, where the predictor functions perfect if 1, opposite if -1 an random if 0 [48].

TABLE II: Confusion matrix.

		ACTUAL			
		Positive		**Negative**	
PREDICTED	**Positive**	True Positive	(TP)	False Positive	(FP)
	Negative	False Negative	(FN)	True Negative	(TN)

$$R = \frac{\text{TP}}{\text{TP + FN}} \tag{1}$$

$$P = \frac{\text{TP}}{\text{TP + FP}} \tag{2}$$

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{3}$$

With little prior experience required to tune hyperparameters manually, a conservative and justified approach would be a grid search [49, 50]. However, a grid search is computationally expensive in a multidimensional hyperparameter space. Alternatively, random search has shown to outperform grid search at significantly lower cost [49]. The downside of this more efficient approach is the potentially limited exploration of the domain. A Bayesian optimisation overcomes these limitations by building a probabilistic surrogate model. This surrogate is a posterior probability distribution consisting of Gaussian distributions. This surrogate establishes probabilistic surfaces, i.e. a mean and standard deviation surface, over the entire domain. These surfaces are then used to identify the best region for further investigation. Still, there might be regions of high uncertainty, i.e. standard deviation, but these are at least covered, in great contrast with a random optimisation. That is why this strategy was chosen for the hyperparameter optimisation.

F. Probing Strategy

A utility or acquisition function determines the probing strategy of a Bayesian optimisation, i.e. the regions to investigate. Amongst those, Expected Improvement (EI) and Upper Confidence Bound (UCB) have shown the most promising results in earlier research [50]. The definitions of these utility functions are provided by Equation (4) and Equation (5).

$$EI(x) = \begin{cases} \Big(\mu(x) - f(x^+) - \xi\Big)\Phi(Z) + \sigma(x)\phi(Z) & \text{if} \quad \sigma(x) > 0 \\ 0 & \text{if} \quad \sigma(x) = 0 \end{cases} \tag{4}$$

$$\text{where} \quad Z(x) = \begin{cases} \frac{\mu(x) - f(x^+) - \xi}{\sigma(x)} & \text{if} \quad \sigma(x) > 0 \\ 0 & \text{if} \quad \sigma(x) = 0 \end{cases}$$

$$UCB(x) = f(x^+) + k\sigma(x) \tag{5}$$

where

x	=	hyperparameter vector
x^+	=	best probed hyperparameter vector
ξ	=	exploration factor
k	=	confidence factor
f	=	black box function
μ	=	posterior mean function
σ	=	posterior standard deviation function
Z	=	normalised improvement
Φ	=	cumulative distribution function
ϕ	=	probability density function

The EI utility has a major two advantages compared to UCB. Firstly, the EI function computes the absolute expected improvement of the optimisation metric and that is why a convergence criterion can be set based on its maximum (relative) magnitude. And secondly, this function has the ability to probe exploratory, by means of the exploration factor ξ. This characteristic is important for general domain exploration and to move away from local minima. In addition, a patience criterion was applied to allow for some additional exploratory probing prior to convergence, ensuring confidence in the outcome. The optimisation was first initiated by random probing, such that the posterior mean and standard deviation surfaces could be readily established before the utility function would take control of the probing.

G. Regularisation

The hyperparameter optimisation could result in a very deep neural network which are known to be very prone to overfitting. The application of regularisation could become then essential to surmount this issue [51]. Two regularly applied techniques to overcome the occurrence of overfitting are dropouts and early stopping. The former simply deactivates units with a user-defined probability of deactivation. Dropouts could be applied between all layers and thus introduce new parameters to design and optimise for [51]. Early stopping literally stops the training early based on the loss of the weight optimisation. These criteria could be a quotient, a threshold or a patience [52]. Contrarily to dropouts, early stopping requires neither design nor optimisation considerations, but solely a set of stopping criteria. The applicability and necessity of these regularisation techniques depend on the behaviour of the learning algorithm. Hence, the subsequent section will discuss the relevance of these two methods.

H. Sampling

An appropriate optimisation metric is usually not sufficient to overcome the consequences of the data imbalance. To ensure a learning algorithm is presented with sufficient samples of each class, the class distribution of the training data can altered by means of sampling. In that respect random sampling has proven to be an efficient and regularly well performing solution [53, 54]. In addition to random sampling, there exist synthetic methods, such as *SMOTE* and *ADASYN*, which generate synthetic, unseen samples and overcome the same limitations of random sampling at the expense of computational effort [55, 56]. However, the size of the dataset made these synthetic sampling techniques readily infeasible since both methods are k-NN based and thus computationally expensive. According to literature random undersampling is preferred random oversampling. However, the latter was favoured for this particular problem as the class skewness would drastically cut back the size of the training dataset and consequently negatively affect the ability of the Neural Network to be trained. Hence, random oversampling was chosen as sampling technique.

As there exists neither a given optimality nor a well founded guideline regarding the amount of oversampling to be applied, this methodological step introduced a new hyperparameter R_o, the ratio between the number of instances in the minority class N_{min} and the number of instances in the majority class N_{maj}, as shown by Equation (6). The oversampling ratio was not considered as part of the Bayesian optimisation but a grid search was undertaken instead. This grid search ensured a sustained exploration of multiple settings, such that the effect of this methodological step could be addressed. Contrarily, as a part of the Bayesian optimisation it could be the case that R_o would only be evaluated in promising regions and there remains uncertainty in others. This situation then results in a limited understanding of the effect of oversampling on the classification metrics. This additional hyperparameter would also have introduced a very expensive dimension, requiring many randomly initiated points to have the same coverage as for a lower dimension problem.

$$R_o = \frac{N_{min}}{N_{maj}} \tag{6}$$

I. Validation Strategy

To validate the results of the above analysis a validation strategy had to be chosen. In that respect three strategies exist: hold-out, leave-one-out and k-fold cross-validation [45]. Figure 4 shows a schematic representation of these strategies. Hold-out was favoured as it has two advantages over the other methods. Firstly, hold-out is computationally cheap since it has fixed data split. On the contrary, leave-one-out and k-fold

have a moving data split, as depicted by Figure 4. Due to this moving split these techniques require N, the number of instances, and k, the number of folds, evaluations respectively. And secondly, hold-out allowed for a low variance implementation of random oversampling. The other two methods might have suffered from high variance as a result of the skewness of the classes. The non-oversampled validation set of these methods would have contained very few instances of the minority class and if this subset is not representative to the trained instances, i.e. too unique, high variance results would be obtained. Hence, hold-out was in favour as its validation set has a larger number of instances of the minority class. A training, development and test distribution of 60-20-20% was chosen according to best practices [57].

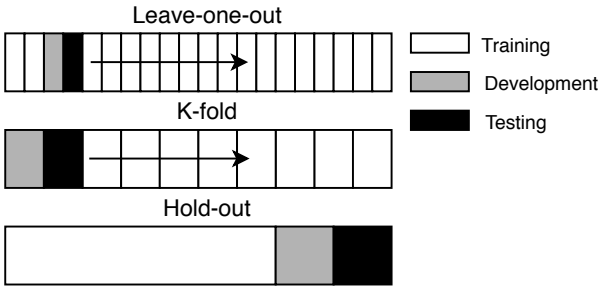


Fig. 4: Data splits for different validation strategies.

IV. Results

This section presents the results of the transformation and data mining step as introduced in Section III. These steps were conducted sequentially and the results are therefore presented accordingly. In addition, and prior to presenting the data mining results, the setup of the experiments is shortly discussed. At last, the sensitivity of the applied methods is assessed.

A. Feature Selection

Relief assigned weights to the variables and these are seen in Figure 5. Subsequently, the weight threshold τ was determined visually, as suggested by the authors of Relief [36]. The threshold was set at 0.1, as below this value the slope of the curve was significantly less steep implying a substantial amount of features would be added if the threshold would be lowered further. Hence, this value was deemed a fair trade-off between dimensionality reduction and relative importance, given the magnitude of the largest weight. Consequently, the number of features was reduced from 195 to 30, of which the weights are listed in Table III. Note that the weights of the categorical and ordinal features are accumulated in this table. According to this table features originating from all considered processes were selected. The identification of the selected processes was essential since their absence would readily require to reconsider the concept of this study. However, the weather parameters were noticeably dominant in this transformation step, a finding that will be discussed in the subsequent section.

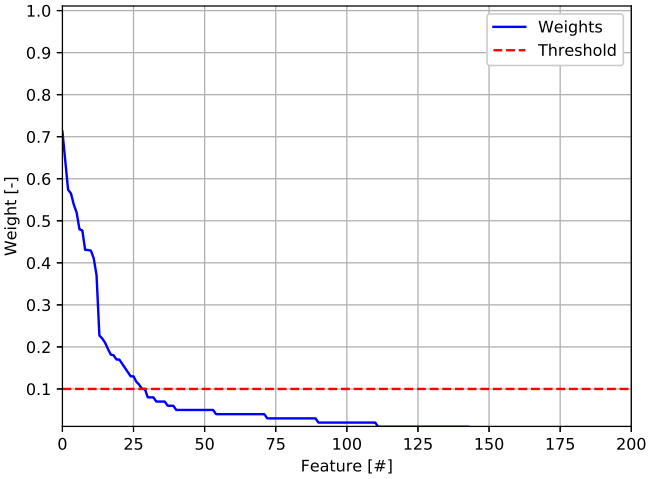


Fig. 5: Sorted feature weights.

TABLE III: Weights of selected features.

Description	(Acc.) weight [−]
Day of week	1.18
Difference in seniority	0.71
Age	0.65
Headwind	0.57
Crosswind	0.57
Pressure	0.54
Wind direction	0.52
Seniority	0.48
Planned flight time	0.48
Dew temperature	0.43
Temperature	0.43
Wind speed	0.43
Fogginess	0.41
Wind variation	0.37
Aircraft type	0.34
Month	0.25
Cloud base height	0.23
Ground time prior to flight	0.20
Wind gust	0.18
Visibility	0.12

B. Experimental Setup

After the features were selected, the hyperparameter tuning was done for a range of oversampling ratios by means of a grid search. This grid was defined from 0.2 to 1.0, with a spacing of 0.2, for the non-regularised experiments and from 0.1 to 1.0, with a spacing of 0.1, for the regularised experiments. The non-regularised experiments had a coarser grid as these were computationally expensive, as will be substantiated subsequently. Besides that, an experiment with no sampling, $R_o = 0.03$, was conducted for both cases. These experiments investigated the necessity of oversampling as methodological step. The Bayesian optimisation was undertaken with a convergence criterion of 0.01 and a patience of 5 iterations. The initialisation of the optimisation was done by randomly probing 20 points.

C. Non-Regularised Experiments

The applicability of regularisation depends on the behaviour of the learning algorithm, as Section III stressed. In the early non-regularised experiments some unsteadiness was observed which in fact made regularisation essential for the hyperparameter optimisation. Figure 6 presents the loss plot of such a non-regularised weight optimisation which shows significant instabilities in the training loss from 25 epochs onward. These instabilities are to be attributed to some poorly representative training batches, i.e. batches that share little to no characteristics with the learned instances so far. Note that the data in each epoch is divided over batches and that the computed loss is average over all the batches. As the training has already converged, the subsequent epochs present the network with irrelevant information. Due to this unpredictable behaviour epochs could not be considered as a tuneable hyperparameter within the Bayesian optimisation. In other words, finding the optimal number of epochs would be a subject to randomness as the result of the poor information encompassed by the later batches. Hence, early stopping was applied on the training loss to stop the training prior to the occurrence of these instabilities. A loss quotient of 0.01 and a patience of 5 iterations were defined as early stopping criteria, based the loss behaviour of the non-regularised experiments. In addition, the application of early stopping saved computational resources by reducing the number of epochs and the size of the hyperparameter space. In conclusion, the subsequent results were all regularised and only considered two hyperparameters: the number of hidden layers and the number of units per layer. *Appendix D* elaborates on the instability of the non-regularised experiments.

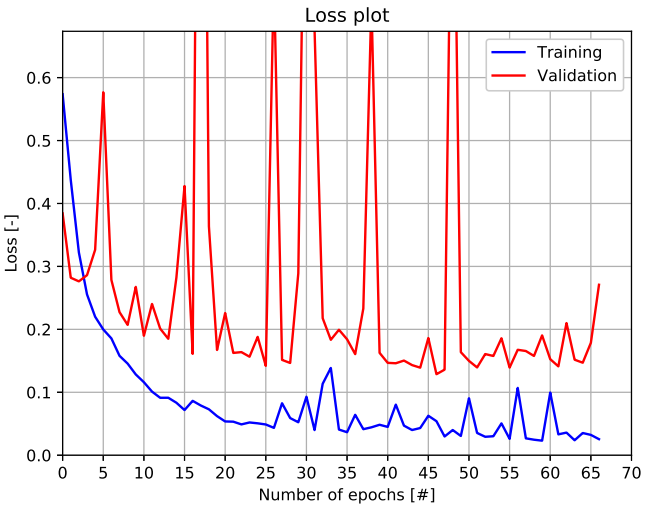


Fig. 6: Non-regularised loss plot.

D. Regularised Experiments

The remaining hyperparameters were tuned per oversampling experiment and the results thereof are listed in Table IV. From this table it is concluded that the optimisation objective was rather insensitive to the oversampling ratio, as seen from a value of 0.2 onward. These results also show that with no or little oversampling the network had limited ability to learn. The number of positive instances in the training was not sufficient to teach the network its characteristics, or, in other words, to overcome the dominance of the negative instances. This finding is underpinned by the increment in recall observed over the lower oversampling ratio experiments. Oversampling is therefore shown to be an essential methodological step for the learnability of the problem, but not for the optimisation as seen from an oversampling ratio of 0.2 onward.

TABLE IV: Best probed MCC iteration per experiment.

R_o [−]	P [−]	R [−]	MCC [−]	N_L [#]	N_U [#]
0.03	N/A	0.000	N/A	N/A	N/A
0.10	0.465	0.400	0.416	4	150
0.20	0.458	0.530	0.476	10	137
0.30	0.469	0.541	0.488	15	118
0.40	0.477	0.528	0.486	5	136
0.50	0.461	0.544	0.484	9	122
0.60	0.449	0.571	0.490	15	140
0.70	0.473	0.573	0.505	15	150
0.80	0.439	0.573	0.485	12	142
0.90	0.459	0.538	0.481	5	136
1.00	0.449	0.573	0.491	11	138

By isolating the iterations with the highest precision and recall per experiment, a thorough understanding of the optimisation was obtained as well as its limitation. These results are shown by Table V and Table VI respectively. When the former table is compared to Table IV, it is observed that the best performing precision iterations often coincide with the best MCC iterations. This indicates that the metrics involved in the evaluation of the precision, the positives, were dominant in the optimisation. Given that the precision was always behind the recall it is sensible that the optimisation attempted to progress in that manner. If a high recall was found, Table VI shows that this came at the cost of the precision. This drift originated from the class imbalance and was aggravated by the oversampling, as the network was presented with a larger number of positive instances. This phenomenon particularly happened at a low number of units when such narrow networks compressed the information and made the prediction of positives even more dominant, as Table VI underpins. Hence, the oversampling did affect the overall behaviour and performance of the optimisation as the precision and recall were skewed more frequently. These balanced results in Table IV can be attributed to the well-functioning of the objective. The MCC ensured the optimisation turned away from these skewed solutions as the MCC was extremely low. This demonstrates that the MCC was an appropriate optimisation metric for this problem, in clear contrast with recall or precision. Eventually all experiments yielded similar results showing a consistent understanding of the complex, underlying system. This consistency is sensible as, fundamentally, random oversampling does not add information to the problem since it solely duplicates existing instances and effectively adapts the mutual weights of the classes.

TABLE V: Best probed precision iteration per experiment.

R_o [−]	P [−]	R [−]	MCC [−]	N_L [#]	N_U [#]
0.03	N/A	0.000	N/A	N/A	N/A
0.10	0.465	0.400	0.416	4	150
0.20	0.458	0.530	0.476	10	137
0.30	0.469	0.541	0.488	15	118
0.40	0.477	0.528	0.486	5	136
0.50	0.461	0.544	0.484	9	122
0.60	0.458	0.530	0.476	7	135
0.70	0.473	0.573	0.505	15	150
0.80	0.446	0.553	0.480	15	138
0.90	0.459	0.538	0.481	5	136
1.00	0.449	0.573	0.491	11	138

TABLE VI: Best probed recall iteration per experiment.

R_o [−]	P [−]	R [−]	MCC [−]	N_L [#]	N_U [#]
0.03	N/A	0.000	N/A	N/A	N/A
0.10	0.404	0.448	0.407	10	133
0.20	0.400	0.536	0.445	9	143
0.30	0.379	0.563	0.443	10	138
0.40	0.209	0.592	0.321	2	110
0.50	0.195	0.631	0.318	14	59
0.60	0.119	0.634	0.230	15	50
0.70	0.116	0.653	0.229	12	36
0.80	0.099	0.685	0.210	12	33
0.90	0.065	0.716	0.147	10	20
1.00	0.074	0.711	0.166	8	22

The behaviour of the hyperparameters with respect to the objective is described by Table VII, which lists the correlations between the MCC of the development set and the number of layers, the number of units and the number of epochs, defined as r_L, r_U and r_E respectively. This table shows that the correlation between the units and the MCC was generally higher than that of the hidden layers and the MCC. Figure 7 illustrates the mean surface of the Bayesian optimisation for $R_o = 0.8$, which underpins this finding as the majority of the surfaces are horizontally stretched. These stretched surfaces indicate only a marginal progress in the direction of the hidden layers which can be explained as follows. Adding units to all layers, the number of weights, i.e. the degrees of freedom, increases exponentially. On the contrary, adding layers only linearly increases the number of weights. This is the consequence of the MLP architecture, which has links between all units. Due to the complexity of the underlying patterns the network required a substantial number of degrees of freedom to build an understanding and likely found these sooner by increasing the number of units per layer. Similarly, previously it was discussed how a low number of units compressed these patterns which resulted in a low precision and high recall. These networks lacked sufficient weights to establish a balanced comprehension.

The correlation with epochs was observed to be higher when the applied sampling was lower. As the networks of the lower oversampling experiments saw less positive samples per training epoch, their iterations benefitted by seeing these instances more often by means of epochs. This effect diminished with increasing oversampling ratio as the number of positives per epoch increased accordingly. This shows that there was general need for more positive instances during training, and that this was established by either epochs or sampling.

TABLE VII: Correlations between development set MCC and hyperparameters.

R_o [−]	r_L [−]	r_U [−]	r_E [−]
0.03	N/A	N/A	N/A
0.10	0.205	0.718	0.861
0.20	0.394	0.589	0.653
0.30	0.386	0.656	0.498
0.40	0.653	0.419	0.274
0.50	0.484	0.786	0.558
0.60	0.431	0.635	0.334
0.70	0.377	0.697	0.181
0.80	0.557	0.659	0.305
0.90	0.268	0.732	0.175
1.00	0.266	0.707	0.258

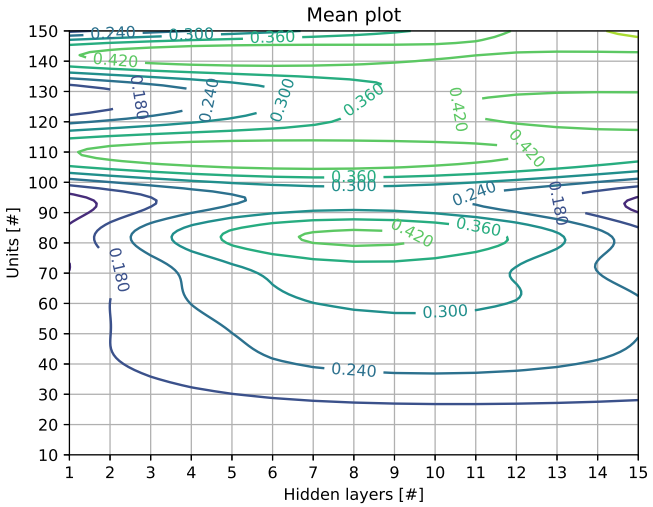


Fig. 7: Mean plot $R_o = 0.7$.

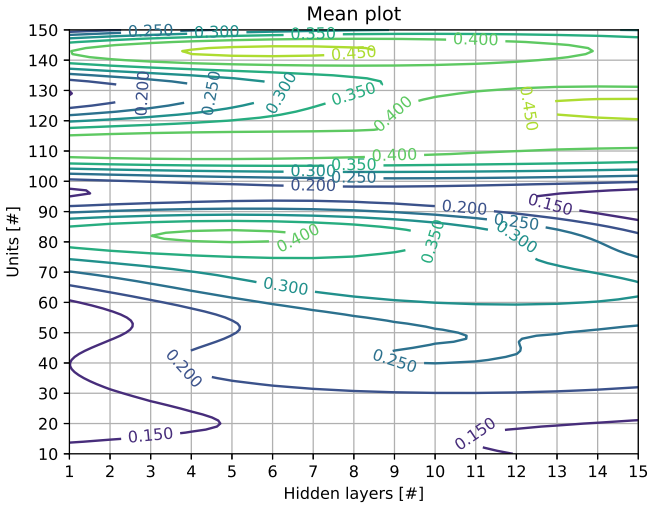


Fig. 8: Mean plot $R_o = 0.7$ at 30$^{\text{th}}$ iteration.

The EI utility function was chosen for its straightforward convergence criterion and its ability to probe exploratory. Due to its essential function in the optimisation its results

are evaluated by means of comparison to its best alternative according to literature, UCB. Figure 9 and Figure 10 depict the two standard deviation (k=2) and expected improvement of the best performing network at the 30th iteration respectively. These surfaces show that whilst the standard deviation was still significant, with respect to the local mean shown by Figure 8, the expected improvement had met the convergence criterion of 0.01 already. Figure 11 shows that convergence was established five iterations later due to the applied patience. Hence, in terms of efficiency EI outperformed UCB. In addition, this curve indicates some successful exploratory behaviour by means of the initial peak. These peaks occurred when the utility function probed a point in a region of high uncertainty and the accompanying shift in the surface exposed a new promising region. In *Appendix E* an elaborate discussion on the applicability of the utility function is provided.

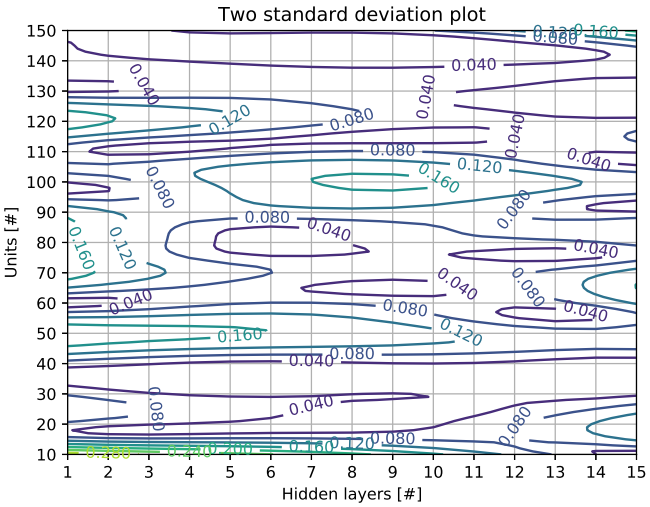


Fig. 9: Two standard deviation $R_o = 0.7$ at 30$^{\text{th}}$ iteration.

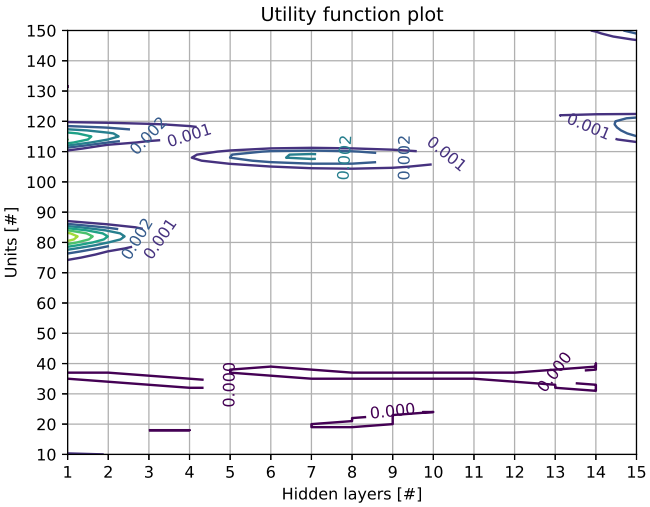


Fig. 10: Utility plot $R_o = 0.7$ at 30$^{\text{th}}$ iteration.

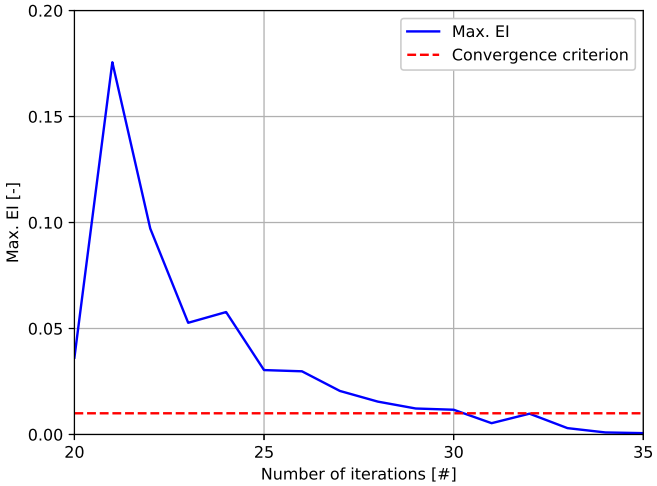


Fig. 11: Convergence plot $R_o = 0.7$.

E. Weight Optimisation

The weight optimisation of the non-regularised experiments showed instabilities in its training loss and consequently early stopping had to be applied. The effectiveness thereof is seen in Figure 12. This figure shows the loss plot of the best performing model presented in Table IV. The training loss was notably more steady and did not oscillate, legitimising the application of early stopping regularisation. The unsteadiness of the validation loss, during the early epochs, could be attributed to the model just starting to learn the problem and still misclassifying a substantial amount of instances. Particularly given the difference in class distribution and the application of the batch gradient descent such peaks were to be expected in the validation loss. Eventually, the losses are parallel and do not seem to converge any further, justifying the point of stopping since there was no further progress. Hence, this indicates that the validation set was not sufficiently representative with respect to the training set. In other words, the training did not contain adequate information to establish the same classification performance for the validation set. *Appendix G* confirms this finding and elaborates on more aspects of the loss plots.

The precision-recall curve is depicted by Figure 13. This figure shows the precision and recall for a moving decision threshold. Note that the output of the neural network is the probability of an instance belonging to a class and that for the global optimisation a threshold of 0.5 was applied. These curves are generally recommended for the evaluation of imbalanced problems as these plots are less biased [58]. Ideally, the area under the precision-curve should be 1.0, covering the entire domain and surpassing a recall and precision of 1.0, being a perfect classifier. The area under the curve represents the predictive performance of this network with respect to a random classifier. Hence, the current classifier has a significantly better performance than a random classifier. However, there still exists a large region of progress located

Fig. 12: Loss plot $R_o = 0.7$.

above the curve. This is in fact the region that could not be covered due to the lack of information addressed priorly. In addition, performance decisive misclassifications are seen at the both edges of the domain. On the right side of the domain an asymptote at the precision value of 80% is seen. This asymptote shows that if the threshold is moved towards classifying all instances negative, to the right over this curve, a consistent amount of 20% of the positive predictions are factually negatives. Similarly, on the left side of the domain a rapid exchange between the metrics, i.e. the steep slope, indicates that true positives are located in the region declared negative. According to Table IV all these instances are located beyond a confidence interval of 50%, which means that these instances could not be distinguished from the opposite class based on this feature set.

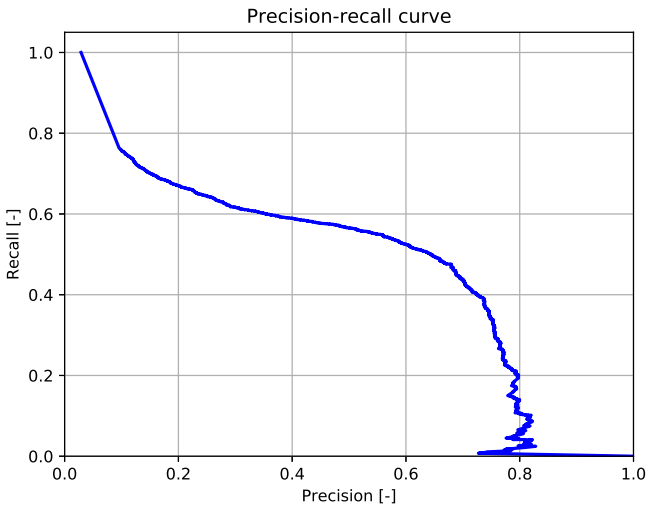


Fig. 13: Precision-recall plot $R_o = 0.7$.

From Table IV it became clear that the oversampling did not have a major effect on the global results of the optimisation. Still, locally there might be significant differences in terms of performance. These differences could be best exposed by the precision-recall curves. Figure 14 shows the precision-recall curve of an experiment with a oversampling ratio of 0.4. The general shape of this curve is similar to Figure 13 which indicates that understanding of the network was found to be rather independent of the applied oversampling. As mentioned previously, essentially oversampling does not add information to the problem, it duplicates the existing instances and thus adapts the mutual weights of the classes. The most significant discrepancy is seen near precision value of 0.8, where the curves are shaped differently as the consequence of the little number of positive samples that remain whilst shifting the threshold further towards the edge of the domain. *Appendix F* elaborates on and underpins this finding by discussing more precision-recall curves, which show there existed a general understanding amongst all experiments.

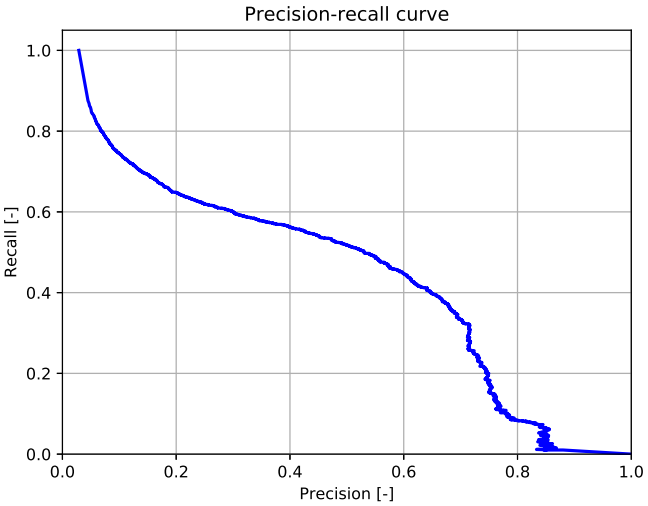


Fig. 14: Precision-recall curve $R_o = 0.4$.

F. Sensitivity Analysis

The sensitivity analysis was performed for three different (sets of) parameters. First, the sensitivity of the feature selection was evaluated. Given that there is no interaction with the neural network, any instability in the feature selection would readily imply uncertainty regarding the precursors. Second, the sensitivity with respect to the data distribution and random initiation on the Bayesian optimisation and the final results were evaluated. As there exist many variations on the hyperparameters, such as activation functions or loss function amongst others, their effect could not be investigated within this research due to the limited available resources. Besides, the relevance of those variations would be readily unjustifiable given that the majority of assumptions were made based on best practises.

1) Feature Selection: The sensitivity of the feature selection was assessed by repeating the procedure and by k-folded

validation. The latter would potentially identify local irregularities, by selecting a smaller subset prior to the selection. Figure 15 depicts the cumulative sum of the weights for $k = 6$ and from this figure it becomes clear that the general trend, i.e. slope, was consistent over the different folds. Therefore it was concluded that the feature selection was stable and could be fed to the neural network with confidence. *Appendix B* elaborates on the stability of the feature selection and underpins the stability of this methodological step.

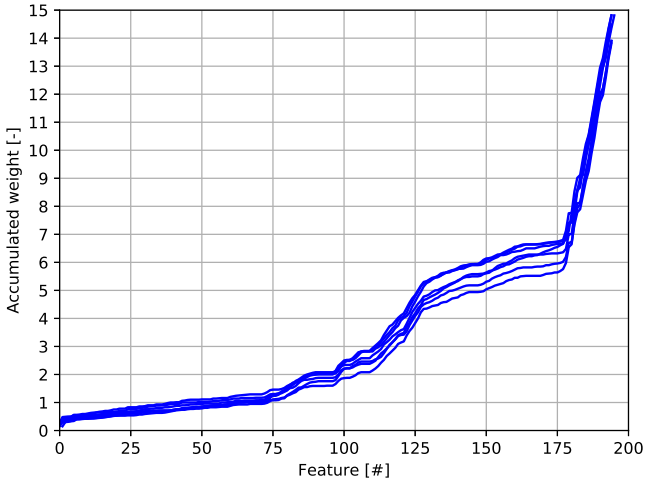


Fig. 15: Cumulative weight for $k = 6$.

2) Data Distribution: The data was split in three folds of 60%, 20% and 20% for training, development and testing respectively. This splitting was done randomly. Hence, by repeating the experiment the sensitivity to the data split could be evaluated. Figure 16 depicts the precision-recall curve for a different, random data split. The general shape of the curve is similar and so is the covered area, representing the predictive ability of the network. The confident misclassifications are again seen on both edges of the domain, indicating consistency in the predictions. This observation suggests that the classes were well distributed, which is sensible given the size of the dataset. The largest discrepancy is seen near a precision value of 0.8, where the number of positive instances is small and changes in threshold setting induce fluctuations in the assigned precision score.

In addition, a general sensitivity was observed in the validation loss, depicted by Figure 17. These instabilities occurred due to some poorly representative batches, similar to the instabilities in the training data discussed earlier. These batches are established randomly and are much smaller than the folds, consequently these batches do not have the same consistent distribution. The class distribution of the training and validation split were furthermore different due to the applied oversampling, which made the validation evaluation prone to this phenomenon. Moreover, some overfitting is seen when the general trend of the validation loss increases after 20 epochs. According to Figure 16 this overfitting did not impact the global performance substantially, i.e. the network basically became more confident about wrongly classified instances or less confident about rightly classified instances. *Appendix G* discusses a similar finding for different iterations of the initial dataset and underpins these fluctuations had limited effect on the results.

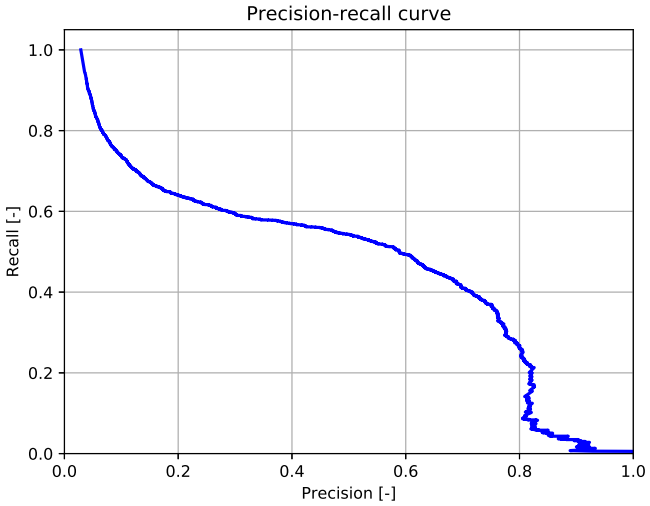


Fig. 16: Precision-recall curve $R_o = 0.7$ for different random data split.

Fig. 17: Loss plot $R_o = 0.7$ for different random data split.

3) Random Initialisation: The initialisation of the Bayesian optimisation was done by randomly probing 20 points, after which the utility function took over. This random initiation was strictly necessary to give the utility function a general idea of the domain and start probing efficiently. By repeating any experiment the initialisation changes and so the initial surface presented to this function. This surface determines its probing strategy for future iterations and the convergence and so the final results. Figure 18 and Figure 19 show a mean plot for two different runs. Both figures show strong

similarities in the upper region of domain, where many points were probed as wide networks were more successful as stated earlier. Similarly, the peak in the middle the domain was found during both optimisations. This peak is, compared to the best values, significant which explains that both runs identified this region. The most noteworthy discrepancy is seen at the bottom left of the domain, as Figure 19 does show a hill whilst the other figure does not. The second optimisation did not probe in this region and left a high uncertainty, i.e. standard deviation, instead. This discrepancy is the consequence of the level of exploratory behaviour the utility function introduces in each experiment.

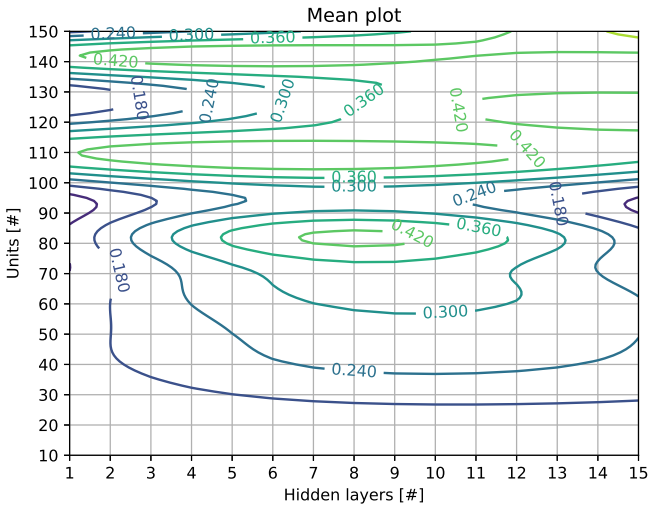


Fig. 18: Mean surface $R_o = 0.7$.

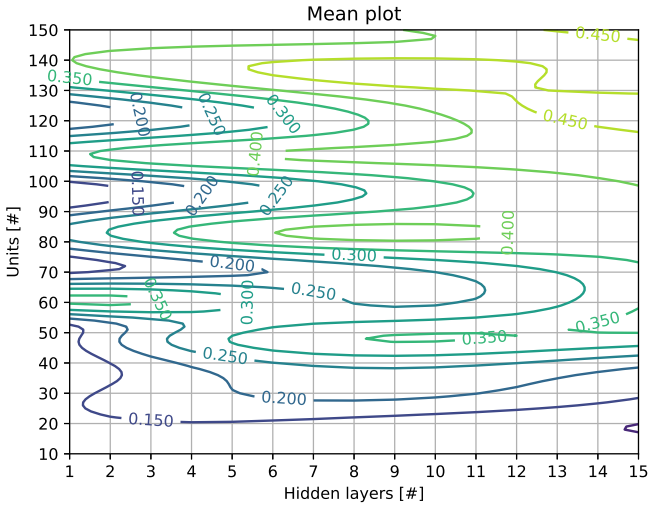


Fig. 19: Mean surface $R_o = 0.7$ at convergence with different random initiation.

V. Discussion

This research aimed at identifying a set of precursors for a safety event that currently gains a lot of attention in the aviation industry, namely unstable approaches. The prior section showed how the applied methodology could exceed the performance of a random classifier by means of the area under the precision-recall curve and MCC. The latter cannot be interpreted as a measure of accuracy but rather as a correlation measure. According to this measure there only existed a moderate correlation between the considered (airline) processes and unstable approaches. If the other two metrics are interpreted from an operational perspective one could say that 57.3% of all unstable approaches were found by this method. If an unstable approach was predicted it was 47.3% correct in stating it was actually an unstable approach. Finding more unstable approaches came at the cost of significantly more false positives. Whether this is acceptable depends on the severity of the safety event. Regardless, this is the biggest limitation in day to day application of this methodology since its uncertainty in predicting positives, i.e. the large number of false positives, could be very costly.

The results were limited due to the lack of information. This shortcoming came forward from the loss as well as the precision-recall plot. The latter indicated the network could not differentiate some instances as actual negatives and positives were classified to the opposite class with great confidence. However, whether this shortcoming can in fact fully be attributed to a lack of information is yet to be further investigated as the hyperparameter space was not explored to its best extent. Many assumptions were made regarding the hyperparameters, to allow for an affordable optimisation. These assumptions reduced the hyperparameter space severely to only three parameters, the number of hidden layers, the number of units and the number of epochs. The latter was later discarded as the result of the discussed instabilities. The effect of the assumptions could not be investigated due to the many variations that should be considered and inherently would have costed resources. Furthermore, the relevance of many of those variations could be readily questionable as the majority of the applied assumptions was well founded. The limited exploration of the domain is however a limitation of this methodology.

It is sensible that not all occurrences could be predicted as the dataset at hand governed extremely time distant data, with respect to the actual occurrence of the safety event. In addition, these datasets essentially ignored all physical relations. The considered weather data was, too a large extent, the sole exception for both of these conditions. The feature selection showed this particular set of information to be prevailing, underpinning the relevance of less time distant datasets. Similarly, other studies considered directly related and less time distant datasets and showed to be capable of identifying and predicting anomalies [7, 8, 9, 10, 11, 13, 14]. These datasets generally have a higher correlation with the occurrence of the event and that made this systemic approach readily more complex. Still, features originating from other processes were also deemed relevant according to the feature

selection and whether the dominance of these weather features was translated to the predictive ability of the neural network could not be assessed because of the nature of the neural network. On the other hand, this is actually also a general limitation as neither of the processes nor features could be evaluated individually.

The behaviour of the hyperparameter optimisation was found to be affected considerably by the applied oversampling, as many of the iterations showed to be imbalanced in terms of precision and recall. Still, sampling was found to be essential to the network's ability to learn the problem since the non-sampled experiments were underfitted, i.e. could not learn the problem at hand. The MCC overcame this drift by re-establishing some balance in these classification metrics as it considered all aspects of the confusion matrix. Hence, the well-functioning of the optimisation objective is a strength of this methodology.

Another strength of this methodology was the application of Bayesian optimisation and the accompanying utility function. The EI utility function showed to be a cost effective acquisition function for the Bayesian optimisation. This optimisation technique was chosen for its efficiency and maximum domain exploration and proved to be, in combination with the EI utility function, a very efficient technique for the two parameter problem proposed. Although for future research, that might include more hyperparameters, it should still prove its consistency. This would also cost more computational resources in terms of random initiation, at least to have the same initial domain coverage as applied during this study. In addition, the extent to which the assumption of Gaussian processes holds for a highly dimensional optimisation is to be carefully reviewed.

VI. Conclusions and Future Work

This paper presented and evaluated a KDD based methodology for predicting the occurrence of flight safety events. This approach was, to the best of our knowledge, unique as it approached the occurrence of flight safety events systematically. Consequently, indirect datasets, generated by different airline (related) processes, were selected in consultation with industry experts. These datasets were, in contrast with existing methods, time distant and not physically related to the flight safety event. A case study showed that a set of precursors, originating from these processes, had the ability to predict flight safety events by outperforming a random classifier and were significant considering the data characteristics mentioned previously. However, the event under investigation could not be encompassed in its entirety and this yields openings for future work as will be discussed subsequently.

The lack of information was identified as a major limitation in the predictive performance and therefore asks for the identification of other potentially valuable data sources. From that perspective it is important to note that all data in this research represented planned airline operations (weather data excepted) and was therefore very time distant. However, these datasets were aligned with the industry's safety perspective. Less time distant processes could encompass valuable patterns which are simply to complex to be governed by the planned data. As an example, operational disturbances might result in operational drift and rushed, i.e. unstable, approaches. Besides, even actual data could evaluated in a proactive fashion if for instance sequences of flights are considered prior to the occurrence of a subsequent flight. Such an application would then be run live and flag safety events before performing the actual flight. Recurrent Neural Networks would be a perfect fit to such an approach as these can consider such sequential inputs. Evaluating both approaches could then address the significance of the less time distant precursors compared to those identified by a methodology similar to this research.

In addition, the effect of the individual processes could be further investigated by for instance an ensemble of Neural Nets where a network is trained for features of particular process. This hypothesises that a particular process is perhaps more significant in predicting certain safety events. Additionally, it would, too a large extent, rule out the necessity of feature selection as each network is trained with a significantly smaller subset of the dataset.

Another opening would be to consider the actors involved in airline operations. Many actors are involved in the airline operations, also prior to the occurrence of a flight safety event. Pilots, air traffic controllers and crew controllers would be interesting actors to consider amongst others. These actors could be regarded as agents and agent-based modelling approach could therefore be an appropriate research line as many cognitive processes take place prior to the occurrence of a safety event. These processes differ substantially from this study as there is no data readily available. This data is to be gathered by modelling all agents. Besides, according to experts, the involved actors are believed to be impact on the outcome of a flight.

References

[1] International Civil Aviation Organisation. *Safety Management Manual (SMM)*. 2013. www.icao.int, Accessed: 31-07-2019.

[2] E. Hollnagel. *Safety-I and Safety-II*. CRC Press, 2014.

[3] G.W.H. Van Es. "Advanced flight data analysis". In: *Proceedings of the 14th European Aviation Safety Seminar, Budapest, Hungary*. 2002.

[4] M.A.F. Pimentel, D.A. Clifton, L. Clifton, and L. Tarassenko. "A review of novelty detection". In: *Signal Processing* 99 (2014), pp. 215–249.

[5] M.G. Karlaftis and E.I. Vlahogianni. "Statistical methods versus neural networks in transportation research: Differences, similarities and some insights". In: *Transportation Research Part C: Emerging Technologies* 19.3 (2011), pp. 387–399.

[6] M. Markou and S. Singh. "Novelty detection: a review—part 1: statistical approaches". In: *Signal processing* 83.12 (2003), pp. 2481–2497.

[7] B.G. Amidan and T.A. Ferryman. *APMS SVD methodology and implementation*. Tech. rep. Pacific Northwest National Lab., Richland, WA (US), 2000.

[8] T.R. Chidester. “Understanding normal and atypical operations through analysis of flight data”. In: *Proceedings of the 12th International Symposium on Aviation Psychology*. 2003, pp. 239–242.

[9] S. Budalakoti, A.N. Srivastava, and M.E. Otey. “Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety”. In: *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews* 39 (2009), pp. 101–113.

[10] S. Das, B.L. Matthews, A. Srivastava, and N. Oza. “Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study”. In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, Washington, DC, USA*. 2010, pp. 47–56.

[11] S. Das, B.L. Matthews, and R. Lawrence. “Fleet level anomaly detection of aviation safety data”. In: *2011 IEEE Conference on Prognostics and Health Management*. IEEE, 2011, pp. 1–10.

[12] L. Tanguy, N. Tulechki, A. Urieli, E. Hermann, and C. Raynal. “Natural language processing for aviation safety reports: from classification to interactive analysis”. In: *Computers in Industry* 78 (2016), pp. 80–95.

[13] A. Nanduri and L. Sherry. “Anomaly detection in aircraft data using Recurrent Neural Networks (RNN)”. In: *2016 Integrated Communications Navigation and Surveillance (ICNS)*. 2016, pp. 5C2–1.

[14] V.M. Janakiraman. “Explaining Aviation Safety Incidents Using Deep Temporal Multiple Instance Learning”. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, United Kingdom*. ACM, 2017, pp. 406–415.

[15] F.F. Herrema, R. Curran, H.G. Visser, D. Huet, and R. Lacote. *Taxi-Out Time Prediction Model at Charles de Gaulle Airport*. Vol. 15. 2018, pp. 1–11.

[16] F.F. Herrema, V. Treve, R. Curran, and H.G. Visser. “Evaluation of feasible machine learning techniques for predicting the time to fly and aircraft speed profile on final approach”. In: *International Conference on Research in Air Transportation*. Vol. 8. Delft University of Technology, 2016, pp. 4–8.

[17] F.F. Herrema, V. Treve, B. Desart, R. Curran, and H.G. Visser. “A novel machine learning model to predict abnormal Runway Occupancy Times and observe related precursors”. In: *Proceedings of the 12th USA/Europe Air Traffic Management Research and Development Seminar, Seattle, WA, USA*. 2017.

[18] L. Drees and F. Holzapfel. “Predicting the occurrence of incidents based on flight operation data”. In: *AIAA Modeling and Simulation Technologies Conference*. 2011.

[19] L. Drees, J. Siegel, P. Koppitz, and F. Holzapfel. “Quantifying probabilities of exceeding the maximum Mach number in cruise flight using operational flight data”. In: *European Safety and Reliability Conference (ESREL)*. 2017.

[20] L. Höhndorf, J. Sembiring, and F. Holzapfel. “Copulas applied to Flight Data Analysis”. In: *Probabilistic Safety Assessment and Management PSAM* 12 (2014).

[21] B.J.M. Ale et al. “Further development of a Causal model for Air Transport Safety (CATS): Building the mathematical heart”. In: *Reliability Engineering & System Safety* 94.9 (2009), pp. 1433–1441.

[22] C. Wang, L. Drees, N. Gissibl, L. Höhndorf, J. Sembiring, and F. Holzapfel. “Quantification of incident probabilities using physical and statistical approaches”. In: *6th International Conference on Research in Air Transportation. Istanbul, Turkey*. 2014.

[23] V. Chandola, A. Banerjee, and V. Kumar. “Anomaly detection: A survey”. In: *ACM computing surveys (CSUR)* 41.3 (2009), p. 15.

[24] S.B. Kotsiantis, I. Zaharakis, and P. Pintelas. “Supervised machine learning: A review of classification techniques”. In: *Emerging artificial intelligence applications in computer engineering* 160 (2007), pp. 3–24.

[25] L. Li, M. Gariel, R.J. Hansman, and R. Palacios. “Anomaly detection in onboard-recorded flight data using cluster analysis”. In: *Proceedings of the 2011 IEEE/AIAA 30th Digital Avionics Systems Conference, Seattle, WA, USA*. 2011.

[26] L. Li, S. Das, R.J. Hansman, R. Palacios, and A. Srivastava. “Analysis of Flight Data Using Clustering Techniques for Detecting Abnormal Operations”. In: *Journal of Aerospace Information Systems* 12 (2015), pp. 1–12.

[27] V.M. Janakiraman, B.L. Matthews, and N. Oza. “Discovery of Precursors to Adverse Events using Time Series Data”. In: *Proceedings of the 2016 SIAM International Conference on Data Mining*. 2016, pp. 639–647.

[28] V.M. Janakiraman, B.L. Matthews, and N. Oza. “Finding precursors to anomalous drop in airspeed during a flight’s takeoff”. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada*. 2017, pp. 1843–1852.

[29] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, et al. “Knowledge Discovery and Data Mining: Towards a Unifying Framework”. In: *KDD*. Vol. 96. 1996, pp. 82–88.

[30] International Air Transport Association. *Unstable Approaches: Risk Mitigation Policies, Procedures and Practises 2nd Edition*. 2016. www.iata.org. Accessed: 18-09-2018.

[31] International Civil Aviation Organisation. *Annex 2: Rules of the Air*. 2005. www.icao.int, Accessed: 14-11-2019.

[32] C. Barnhart, A.M. Cohn, E. Johnson, D. Klabjan, G. Nemhauser, and P.H. Vance. “Airline Crew Scheduling”. In: 2003, pp. 517–560.

[33] J.W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

[34] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*. Vol. 1. 10. Springer series in statistics New York, 2001.

[35] I. Guyon and A. Elisseeff. “An introduction to variable and feature selection”. In: *Journal of Machine Learning Research* 3 (2003), pp. 1157–1182.

[36] K. Kira and L.A. Rendell. “The Feature Selection Problem: Traditional Methods and a New Algorithm.” In: *Proceedings of the 10th National Conference on Artificial Intelligence, San Jose, CA, USA*. 1992, pp. 129–134.

[37] A. Jović, K. Brkić, and N. Bogunović. “A review of feature selection methods with applications”. In: *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE. 2015, pp. 1200–1205.

[38] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An introduction to statistical learning*. Vol. 112. Springer, 2013.

[39] C.M. Bishop. “Novelty detection and neural network validation”. In: *IEE Proceedings: Vision, Image and Signal Processing* 141.4 (1994), pp. 217–222.

[40] M. Markou and S. Singh. “Novelty detection: a review - part 2: neural network based approaches”. In: *Signal Processing* 83.12 (2003), pp. 2499–2521.

[41] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[42] F. Agostinelli, M. Hoffman, P. Sadowski, and P. Baldi. “Learning activation functions to improve deep neural networks”. In: *arXiv preprint arXiv:1412.6830* (2014).

[43] D.P. Kingma and J. Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[44] P.Y. Simard, D. Steinkraus, J.C. Platt, et al. “Best practices for convolutional neural networks applied to visual document analysis.” In: *Icdar*. Vol. 3. 2003.

[45] C.M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[46] F. Provost. “Machine learning from imbalanced data sets 101”. In: *Proceedings of the AAAI’2000 workshop on imbalanced data sets*. Vol. 68. 2000. AAAI Press. 2000, pp. 1–3.

[47] S. Boughorbel, F. Jarray, and M. El-Anbari. “Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric”. In: *PloS one* 12.6 (2017).

[48] B.W. Matthews. “Comparison of the predicted and observed secondary structure of T4 phage lysozyme”. In: *Biochimica et Biophysica Acta (BBA)-Protein Structure* 405.2 (1975), pp. 442–451.

[49] J. Bergstra and Y. Bengio. “Random search for hyper-parameter optimization”. In: *Journal of Machine Learning Research* 13 (2012), pp. 281–305.

[50] J. Snoek, H. Larochelle, and R.P. Adams. “Practical Bayesian optimization of machine learning algorithms”. In: *Advances in neural information processing systems*. 2012, pp. 2951–2959.

[51] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. “Dropout: a simple way to prevent neural networks from overfitting”. In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[52] L. Prechelt. “Early stopping-but when?” In: *Neural Networks: Tricks of the trade*. Springer, 1998, pp. 55–69.

[53] T.R. Hoens and N.V. Chawla. “Imbalanced datasets: from sampling to classifiers”. In: *Imbalanced Learning: Foundations, Algorithms, and Applications* (2013), pp. 43–59.

[54] Z. Zhou and X. Liu. “Training cost-sensitive neural networks with methods addressing the class imbalance problem”. In: *IEEE Transactions on knowledge and data engineering* 18.1 (2005), pp. 63–77.

[55] N.V. Chawla, K.W. Bowyer, L.O. Hall, and W.P. Kegelmeyer. “SMOTE: synthetic minority over-sampling technique”. In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[56] H. He, Y. Bai, E.A. Garcia, and S. Li. “ADASYN: Adaptive synthetic sampling approach for imbalanced learning”. In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE. 2008, pp. 1322–1328.

[57] A. Ng. *Train / Dev / Test sets*. 2020. www.coursera.com, Accessed: 2020-03-10.

[58] P. Branco, L Torgo, and R.P. Ribeiro. “A Survey of Predictive Modelling under Imbalanced Distributions”. In: *CoRR* (2015).

Part II

Literature Study

Previously graded under AE4020

1 Introduction

The number of incidents and accidents in commercial aviation has been relatively low over the past years [1], substantiating that the industry has become extremely safe. Due to the expected growth in demand it is however key for operators to invest in future safety enhancements [2, 3]. The absence of events, moreover, challenges airlines in assessing and subsequently improving their safety performance. As a consequence, the industry's point of view has gradually changed from reactive to proactive [4]. Regardless, their performance assessment more than ever relies on data, another challenging aspect as airlines are confronted with large amounts as well as rapid developments [5]. The wide variety of data in any airline is one of the major challenges. Operators presently lack solutions to adequately mine huge, independent data sets for the purpose of safety management, while it is deemed essential for improvement [6]. In the current context the aim of such safety data analysis would be to expose relevant prior relations for proactive safety management [7].

This literature study will review and discuss the best academic practises in the field of safety management and data analysis to identify a gap in the existing literature. In current day safety management these two topics are inherent to one another as airline safety is nowadays mostly assessed and managed based on data. The subsequent research will be conducted in collaboration with an industry partner. This partner will make the relevant data sets available, such as flight data and crew scheduling data amongst other sources, depending on the identified gap. This data will be used for a case study concerning a current industry-wide safety concern.

This document is structured as follows. Chapter 2 discusses the most general aspect of the upcoming research, safety management. It highlights the past and present as well as the involved data generating processes. Subsequently, Chapter 3 shortly reviews statistical methods, one of the two 'schools of thought'. Afterwards, Chapter 4 continues with the second, namely Machine Learning. Chapter 4 also elaborates on the feature selection, a common process in Machine Learning. With the theoretical background in entirely place, Chapter 5 reviews current applications before identifying the research gap. Chapter 6 then formally sets out the research proposal. The synthesis of this document is depicted by Figure 1.1. Each chapter belongs to a circle which is part of an overarching circle that eventually lead to the outcome, the research proposal.

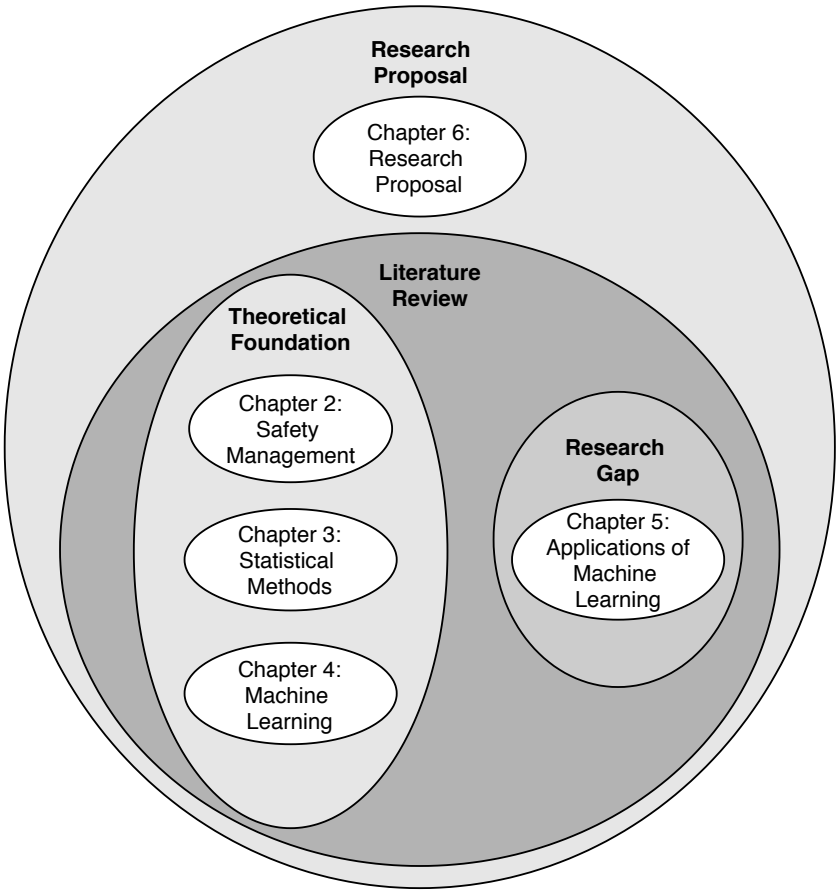


Figure 1.1: Document synthesis.

2 Safety Management

This chapter shortly highlights the past and the present of safety management in Section 2.1. Followed by an ongoing discussion in the field, one of the fundamental motivations of the upcoming study in Section 2.2. Afterwards, Section 2.3 discusses the regulatory system that is currently in place to maintain safe operations in the aviation industry. Lastly, Section 2.4 explains how data is involved in this process.

2.1 Evolution of Safety Management

As addressed earlier, the view on this topic has changed drastically over time. It has been, and still is, subject to change due to fundamental discussions as well as rapid developments. Nowadays, in great contrast with the past, the nearly perfect safety record actually challenges operators in the assessment of their performance regarding safety. Despite that, progress is a necessity since the aviation industry is expected to grow substantially [2]. This might also require to rethink the concept of safety, as one of the subsequent section will argue.

Retrospectively, the field can be categorised in three broad eras with fuzzy transitions, as can be seen in figure Figure 2.1. This figure shows safety management reflected the (believed) causes of events. As an example, in the earlier days of commercial aviation technical and design flaws had more significant share in the incidents and accidents. As a consequence, safety was mostly focused on technical issues [8]. Technological advancements shifted the focus to the human error as the major cause of events. In the 90's however, a new understanding came to be where the complex environment humans operate in was acknowledged as an important, contributing factor. An accident was regarded as the outcome of a system, caused by failures of individual components or layers failing successively [8]. In other words, the first movements towards pro-activeness took place in this decade and that is still of major importance as the subsequent section will discuss.

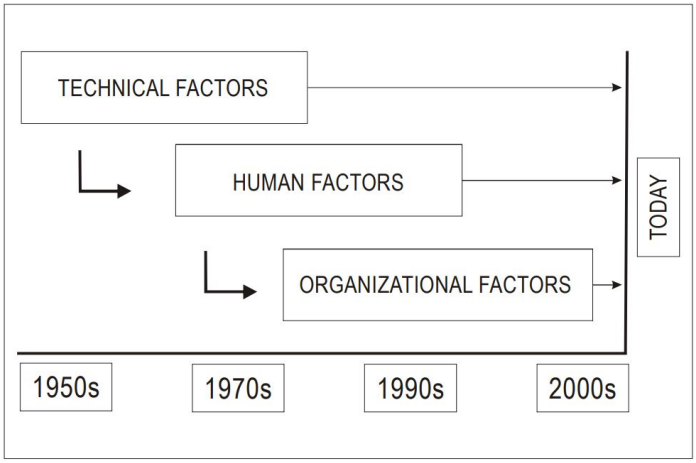


Figure 2.1: Evolution of safety [8].

2.2 Modern Safety Management

In the year 2017 the least fatal accidents took place since recordings started in 1970 [1], suggesting that aviation is safer than ever. Nevertheless, vivid discussions are held in the domain of safety management due to the flourishing prospect and the absence of events. One widespread discussion

is on the most fundamental definition of safety. Safety is often defined as the absence of incidents and accidents i.e. negatives [9]. This point of view is referred to as Safety-I. Safety-I is a reactive approach focusing on things that 'go wrong' and resolves these issues by investigating the causes. Contrarily, its "successor", Safety-II, focuses on ensuring things go right and intends to proactively deal with developments and events [10]. Safety-II is currently gaining much popularity as the result of the lack of serious events required to reason according to Safety-I. Safety-II is on the other hand fairly academic, practical implementations not widely in use yet.

Table 2.1: Summary of two different approaches to safety [10].

	Safety-I	Safety-II
Definition of safety	That as few things as possible go wrong	That as many things as possible go right
Safety management principle	Reactive, respond when something happens or is categorised as an unacceptable risk	Proactive, continuously trying to anticipate developments and events

2.3 Safety Management System

The synthesis of Figure 2.1, in other words the lessons learnt, has been bundled in the safety management system or shortly SMS, which is to be implemented mandatorily by operators in the aviation industry. The Safety Management System is defined by International Civil Aviation Organisation as *a systematic approach to managing safety, including the necessary organisational structures, accountabilities, policies and procedures* [11]. The Safety Management Manual, published by ICAO as well, states that safety management consists of four main processes. These processes are listed below, including the sub-processes that will relate to the subsequent section by positioning that particular process in the current regulatory environment. The upcoming research is obviously part of the safety assurance process, as the upcoming study came forth from a change in perspective and improvement is definitely the target.

- **Safety policy and objectives**
  - Management commitment and responsibility
  - Safety accountabilities
  - Appointment of key safety personnel
  - Coordination of emergency response planning
  - SMS documentation
- **Safety risk management**
  - Hazard identification
  - Safety risk assessment and mitigation
- **Safety assurance**
  - Safety performance monitoring and measurement
  - The management of change
  - Continuous improvement of the SMS
- **Safety promotion**
  - Training and education
  - Safety communication

2.4 Safety Data Analysis

Safety is more than ever related to data analysis. In fact, the assessment of an airline's safety performance is mainly based on data. Therefore many safety related data processes are regulated by authorities and two of these processes are highlighted in this section.

Flight Data Monitoring (FDM) or Flight Operations Quality Assurance (FOQA) is the regulated process for flight data analysis. Civil aviation authority EASA prescribes regulations as well as requirements regarding safety data analysis, such as a quota on the percentage of data an airline should be able to capture. The Acceptable Means of Compliance and Guidance Material (AMC1 ORO.FC.A.245) say that at least 80% of the data (of relevant flights) should be retrieved [12]. This data is supposed to be supplied to a FDM application containing a data analysis algorithm. This step is part of the hazard identification as well as safety performance monitoring process addressed in the previous section. The most common method of FDM is threshold or exceedance analysis, as observed by many researchers [3, 13, 14, 15] amongst others. Threshold analysis triggers events if values exceed a certain threshold. This method is therefore to be classified as Safety-I, since it has no inherently proactive characteristics. In other words, exceedance analysis cannot discover precursors or novelties of any kind. On the other hand, it is convenient, especially with respect to the operating procedures. It also easily understood by a wide public, an important consideration for applications within an airline. The transition towards Safety-II is ongoing, but in the field of data analysis in particular the practical implementation is challenging. Safety-II demands more sophisticated, computationally expensive methods for discovering a priori knowledge. The aim of such proactive data analysis is to identify precursors that are related to certain events and reduce the probability of occurrence by taking mitigating actions regarding the found precursors.

In airlines there is another common, regulated data source that is explicitly monitored for the purpose of safety as its nature is purely safety, namely reporting. According to AMC1.ORO.GEN.200 any operator is obliged to have a reporting system and culture in place that allows staff to report unsafe situations. One of the most well known examples of such a report is the Air Safety Report, or shortly ASR. These reports are filed by air crew and describe any issues or observations related to air safety.

Figure 2.2 shows the interaction between the line operations, the data generating processes, and the evaluation of safety as it is nowadays commonly done within airlines. Note the human factors data and QAR data refer to reporting and flight data respectively. It becomes obvious experts play an important role in this system as their knowledge is needed throughout the entire loop. Ideally, a model would try to capture this knowledge. The feedback loop is closed by training and decision-making, essential to establish the safety promotion process within the SMS.

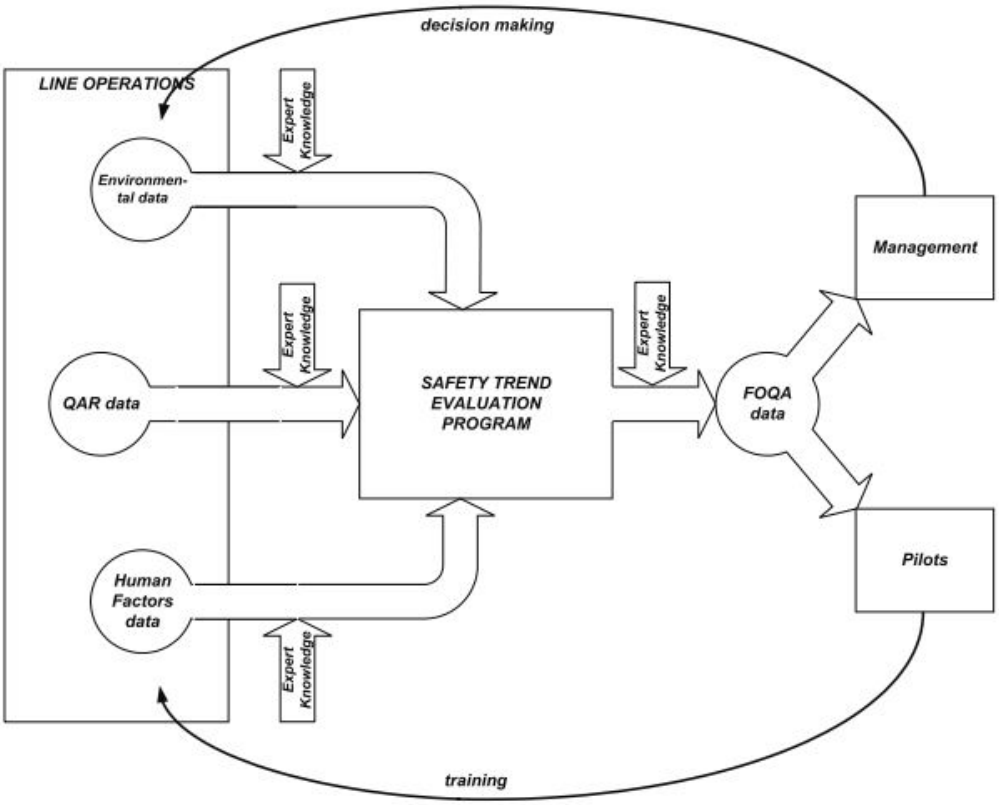


Figure 2.2: A data-based safety management system [13].

3 Statistical Methods

Data analysis or data science is nowadays explored intensively due to the increasing number of data generating processes. The airline industry is no exception in that sense and is currently looking to apply these techniques for data-based improvements [5]. It is important to highlight that in this field two 'school of thought' exist, statistics and learning [16]. Hence, this review is structured accordingly. This chapter discusses statistical methods whilst the subsequent chapter focuses on Machine Learning techniques. Section 3.1 discusses the situation of statistical methods in the field of data science followed by some applications briefly highlighted in Section 3.4

3.1 General Classification

Statistical methods is usually subdivided over parametric and non-parametric methods [17]. Parametric methods assume a dataset can be represented by an empirical distribution defined by a set of definite parameters, such as the normal distribution which is a function of the mean and variance. Contrarily, non-parametric approaches do not no use the empirical parameterised distributions. Histograms would be a common example of a non-parametric method. Pimentel et al. state that in novelty detection the following methods are considered the state-of-the-art [18]:

Parametric Methods

- Mixture Models
- State-Space Models

Non-parametric Methods

- Kernel Density Estimators
- Negative Selection

3.2 Parametric Methods

This section discusses the two most commonly used parametric models according to Pimentel et al., namely mixture models and state-space models. The following paragraphs treat these accordingly [18].

3.2.1 Mixture Models

Mixture models are basically superpositions formed by linear combinations of well known density distributions such as the Gaussian distribution. A renowned mixture model is the Gaussian Mixture Model or simply GMM of which an example is shown by Figure 3.1. Each parameter set is usually obtained by maximum likelihood expectation maximisation algorithm.

3.2.2 State-Space Models

State-space models are regularly utilised for novelty detection in time series. It is assumed that an underlying hidden state generates the instances. This state evolves over time, likely as a function of the given inputs. The two most commonly applied state-space models are the Hidden Markov Model (HMM) and the Kalman filter [18].

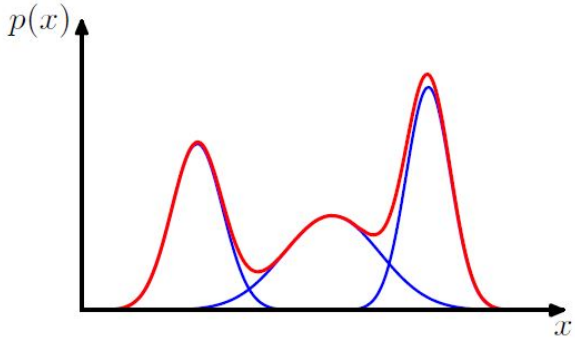


Figure 3.1: Example of a GMM [19].

3.3 Non-Parametric Methods

This section highlights the non-parametric models by discussing the two most popular methods according to Pimentel et al., being Kernel density estimators and negative selection [18]. The information regarding these methods is also retrieved from this particular paper.

3.3.1 Kernel Density Estimators

Kernel density estimators estimate the probability density function by using a large numbers of kernels distributed over the instance space. The estimate at a location relies on the instances that lie within the neighbourhood of the kernel. The kernel density estimator puts a kernel on an instance and sums the local contributions. A Gaussian kernel is popular kernel for instance. This method is often termed the Parzen windows estimator. The Parzen Windows estimator centers an anisotropic Gaussian kernel on a training point, with a single shared variance hyperparameter. Training the Parzen density estimator is done by determining the variance of the kernels, controlling the smoothness of the distribution.

3.3.2 Negative Selection

The negative selection algorithm is inspired by the working principles of the human immune system. An immune system is capable of detecting so called antigens i.e. anything that does not belong to the body. A T-cell receptor recognises these antigens and binds them to it. These receptors are developed by a random process of genetic rearrangements. The cells that are bound to a T-cell receptor are negatively selected and will be destroyed. Hence, a negative selection algorithm requires a probabilistic recognition mechanism to filter the 'bad' instances.

3.4 Applications of Statistical Methods

Recently a research group at the TU München showed how the concept of copulas is capable of quantifying the occurrence probabilities of certain safety events based on flight data [14, 20, 21]. Copulas are, in contrast with correlations, not constant over the domain spanned by the parameters i.e. copulas construct a dependence structure. Figure 3.2 clearly depicts this difference for two arbitrary features X_1 and X_2. The methodology of these researches is generally in line with the earlier work of Drees and Holzapfel and is as follows [22]. First, an incident metric and boundary are defined. This divides the domain in two groups, incidents and non-incidents. Höhndorf et al. for example defined the distance between the end of the runway and the point where the aircraft reached 80 kts as the measure for their runway overrun analysis [14]. Subsequently, contributing factors were identified by for instance establishing a convenient Contributing Factor Tree or applying expert judgement. Probability density functions were then fitted to these parameters based on

the available flight data. If necessary, the physics of the problem were modelled to establish the functional relations between contributing factors. As an example, Drees and Holzapfel applied the equations of motion for the on-ground braking of their runway incursion analysis [22]. Lastly, the contributing factors were combined into copulas of which samples were taken to obtain the distribution for the incident metric.

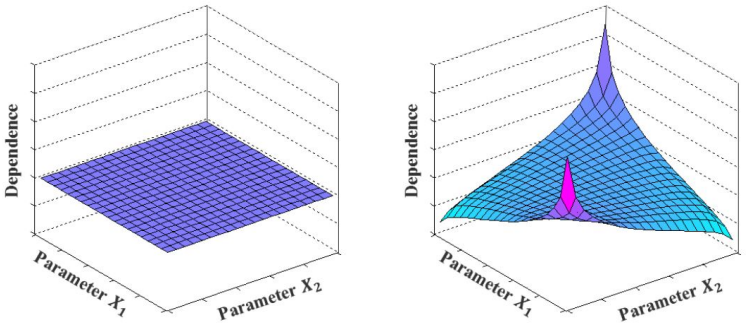


Figure 3.2: Cncepts of correlations and copulas [14].

In 2006 a group of Dutch researchers announced a causal safety model named CATS, this model was promised as the result of an ongoing debate in the Netherlands on causal modelling in the aviation industry [7]. This particular approach was chosen for its high end representation, friendly user interface and monitoring capabilities. The concept was proposed as a combination of fault trees and Bayesian Belief Nets that would yield event categories and subsequently event sequence diagrams, as seen in Figure 3.3. Three years later the 'backbone' of the model was published that identified a total of 36 accident categories. The fault trees probabilities were retrieved from incidents and accidents reports [23].

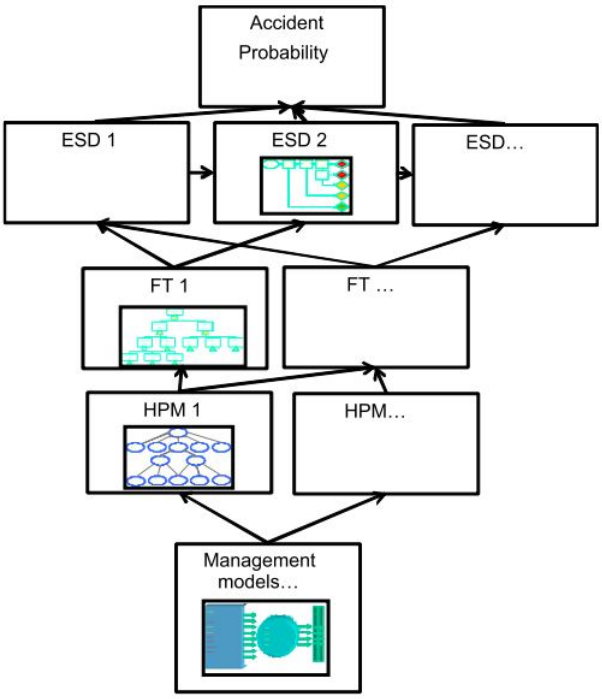


Figure 3.3: CATS architecture [23].

3.5 Applicability of Statistical Methods

In the field of data science Machine Learning is extremely popular, though statistical methods have some major advantages and the position of these methods will be reviewed in this section. This section ends with a conclusive statement regarding the applicability of these methods for the problem at hand.

Markou and Singh review the application of statistical methods in their paper and argue that these are computationally cheap and straightforward in their explanation [24]. Pimentel et al. similarly mention that the required storage is minimal due to the absence of a training and that above all the methods are transparent [18]. In other words, the results could be analysed and explained forthrightly. These authors also independently stress the need for sufficient data for the distribution fitting and point out how dimensionality negatively affects this requirement. This is the so called curse of dimensionality. Chandola et al. explicitly discuss the advantages and disadvantages of statistical methods in their survey on anomaly detection [25]. These are listed below.

Advantages

i. If the assumed distributions are sufficiently accurate, statistical techniques are a legitimate solution.

ii. A statistical technique could provide a confidence interval, which can then be utilised for deciding upon the label of any instance.

iii. If the distribution fitting is resistant to anomalies, statistical methods might be applied unsupervised.

Disadvantages

i. A distribution has to be fitted to the data. This assumption is regularly found to be invalid, especially for highly dimensional data sets (curse of dimensionality).

ii. If the above is true, there arises another difficulty in choosing the appropriate statistic for anomaly detection.

iii. Histogram methods are straightforward to apply, but cannot expose the interactions between features, whilst the interaction in particular might be scarcely occurring.

Statistical methods have some major advantages over the massively popular learning, such as the well founded mathematical background and the capability of providing insights on the data generating mechanisms. However, these methods also suffer from the curse of dimensionality, meaning highly dimensional data yields a very poor performance. The researches at the TU München discussed earlier are an illustration of how fitting empirical distributions can become impracticable as the result of this "curse". If the physics of the model, required to establish the functional relations, become more complicated, i.e. higher dimensionality, poor fits will be the unequivocal result due to local sparsity. Furthermore, statistical methods imply a principal assumption on the data generating process that often does not hold for the huge, complex data sets. That is why learning algorithms are currently so widely applied, as it more conveniently assumes the data is retrieved from a mechanism of unknown dynamics. In other words, no prior knowledge regarding the problem is strictly necessary [16, 18]. The CATS showed the substantial amount of prior knowledge necessary to build the model, as Fault Trees, like Contributing Factor Trees, require a thorough understanding of the system and its potential flaws. Nevertheless, Karlaftis and Vlahogianni suggest to use such methods, compared to Neural Networks, a learning algorithm, if the four conditions below hold [16].

i. There exists a statistical approach that solves a given problem better than Neural Networks.

ii. Researchers have knowledge, or a priori information, regarding the functional relationship of the variables in the problem.

iii. Researchers need to verify the statistical properties of the underlying mechanism that produced the problem.

iv. When interpretability of results and causalities are important.

A common thread is observed in these review papers, pointing out some of the major benefits as well as shortcomings of statistical methods. These are mostly related to the prior knowledge of the problem, the backbone of these methods. If there is any, and the data is not highly dimensional, these methods have a good position in a trade-off. In real life problems this is however generally not the case [18]. In a multi dataset problem, concerning data generated from different, fairly independent processes, this is particularly implausible as the data will be dimensional and heterogeneous. Hence, it is already unlikely statistical methods would be an appropriate solution to the problem at hand. Furthermore, many promising developments are taking place in the field of Machine Learning or Artificial Intelligence. It is noticed that the focus of the entire data science community has shifted towards this rapidly evolving field of studies.

4 Machine Learning

Machine Learning (ML) or Artificial Intelligence (AI) is nowadays widely applied for many different purposes. Therefore it seems as if it is an all-round solution to any data analysis problem with little effort involved. The opposite is true as the variety of methods and applications results in a crucial decision-making process [26]. To select the appropriate method it is first and foremost important to specify and classify the general fields within Machine Learning. This is done in Section 4.1. Afterwards, each section treats different group of algorithms by giving a short description of commonly used algorithms in that particular group. The focus will there be on the crucial considerations in applying that method. Section 4.7 summarises these considerations by identifying the selection criteria, used later on to determine the most suitable method. Section 4.8 concludes this chapter by discussing quality assessment metrics and validation methods as well as two important trade-offs that determine the actual quality of the model.

4.1 General Classification

This section shortly discusses the general classification of Machine Learning technique as it commonly done by researchers in the field. In addition, some advantages and disadvantages are pointed out. This classification is done according to the book of Bishop unless explicitly mentioned otherwise [19].

4.1.1 Supervised and Unsupervised Learning

In many problems the outcome of a certain input vector is given. Hence, this allows to compare the desired situation to the modelled situation and make systematic adaptations to the model to minimise the error. This process is shown by Figure 4.1 as parameter tuning. This is the principle of supervised learning where the training data is thus labelled. Contrarily, in unsupervised learning there is no knowledge on beforehand and the training data is unlabelled. The algorithm is in place to identify groups with significant similarity. This method implicitly assumes the number of normal instances is significantly larger than the number of anomalies [25]. In addition, Chandola et al. define semi-supervised learning as the condition where only the normal instances are classified a-priori [25].

4.1.2 Classification and Regression

There is another distinct categorisation in Machine Learning, namely regression or classification. This categorisation is based on the structure of the output. A method belongs to the classification category if it tries to assign a set of inputs to classes. Ergo, the algorithm has a labelling purpose. These classes could either be known, supervised, or unknown, unsupervised. Contrarily, a regression problem returns one or more predicted variables. Similarly, the predicted variables could be known, supervised, or unknown, unsupervised. Table 4.1 summarises the categorisation of ML algorithms.

Table 4.1: Overview learning categorisation.

	Supervised	**Unsupervised**
Classification	Supervised classification	Unsupervised classification
Regression	Supervised regression	Unsupervised regression

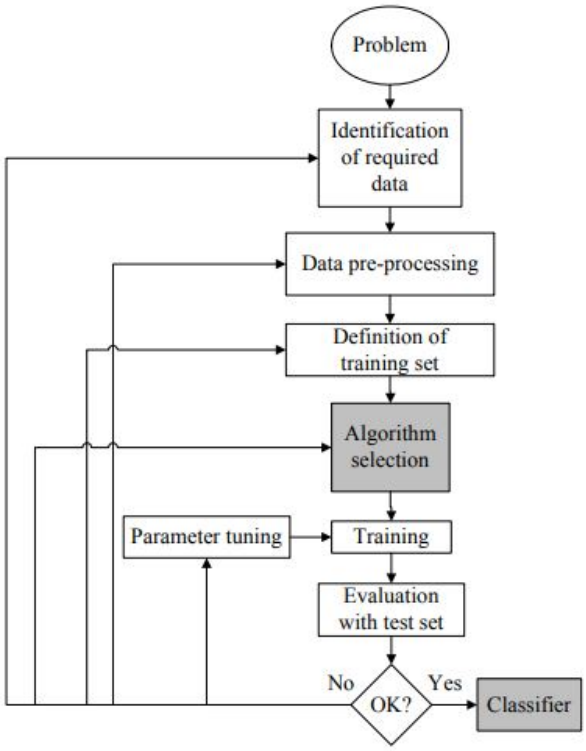


Figure 4.1: Supervised learning process [26].

Chandola et al. state two advantages and disadvantages of the classification methods in their survey. These are listed below [25].

Advantages

i. Use of powerful algorithms that can categorise between different classes.

ii. Fast testing phase since each instance is compared with the model.

Disadvantages

i. Availability of reliable labels especially in multi-class classification.

ii. Labelling not sufficient for meaningful results.

Especially the latter disadvantage is the opposite for regression methods as regression analyses quantify the results which could be more meaningful. It obviously depends on the desired result and therefore the problem at hand.

4.1.3 Reinforcement Learning

Reinforcement Learning is the third group of learning algorithms which is not a straightforward counterpart of either two of the above. Reinforcement Learning involves finding suited actions for a given situation so that a certain reward is maximised. A reinforcement algorithm typically consists of states and actions and learns by trial and error, in contrast with supervised learning where the labels are known in advance. In other words, it reinforces its decision-making by learning from its 'own' mistakes.

Given this general classification it is already possible to draw some important conclusions regarding the layout of the to be applied method. It is likely the safety concern is currently monitored, albeit by other means such as exceedance analysis. Note a serious safety concern is going to be considered and that makes this scenario plausible. Regardless of the method, the classification or identification of flights has taken place and these have usually been validated by an expert. Consequently, a set of instances with outputs might exist, making a supervised setting the most appropriate category. Janakiraman et al. call this form of learning weakly-supervised, since labels can come from for instance flight crew, through ASR's, or an exceedance report, generated by the FDM application [27].

Depending on the complexity of the event, i.e. the number of parameters involved, and the desired results regression or classification is to be decided upon. If the severity of the event, usually a numerical value, is of operational value, regression is obviously the preferable layout. If the event is complex and a much simpler label suffices, classification is preferred. Drees and Holzapfel for instance defined an incident metric, dividing the domain over two classes, incidents and non-incidents [22]. It is furthermore not uncommon to categorise the severity in multiple classes either, making classification a more logical layout in that case. This step is only undertaken if the added value of an exact prediction is of lesser significance. Recall that exceedance analysis also labels and often classifies flights on severity, suggesting this form of classification is well accepted. Note this entire reasoning is basically the consideration addressed by Chandola et al. mentioned above, which can conveniently be summarised by an example [25]. If for instance high speed approaches are to be predicted, the question rises whether the actual flown air speed is of importance. Clearly in this example it might be though if events become complicated a label would be suitable, especially if it already exists. The latter refers to the weakly supervised layout pointed out earlier. Besides, windows of an arbitrary size, say 5 kts, could be equally useful meaning classification is more suited.

Reinforcement Learning is only a relevant layout if decision-making is involved. An example of its capabilities in terms of precursor mining is given by Janakiraman et al. [28]. These authors developed an reinforcement algorithm that can discover precursors in time series, called ADOPT. This paper in fact discusses a totally different philosophy regarding precursor mining, namely the following: A precursor might occur in nominal data and these are usually followed by mitigating actions. The frequency based rule mining methods consider these events as false positives and consequently discard these events. Hence, these algorithms use a 'hard' label whilst ADOPT uses 'soft' information by identifying decisions that increase the risk of occurrence instead. The extent to which mitigating actions are expected is thus a crucial consideration in this application. ADOPT does not search for a random precursor from a data set but approaches the problem as a *search for sub-optimal actions in the adverse time series*. To conclude, the applicability is largely dependent on the considered data sets and the processes these datasets belong to as the latter determines if and what decision-making is involved. If so, it is still the question whether another set of rules would interfere with the existing process and if this is situation is desirable.

The subsequent sections discuss some of the most common Machine Learning algorithms and thereby respect the categorisation set out by Pimentel et al., if this survey covered the particular category [18]. The fundamental description of the methods is retrieved from the books of Bishop, James et al. and Goodfellow et al. [17, 19, 29].

4.2 Domain-Based Algorithms

Domain-based algorithms are decision boundary techniques that opt to divide the overall domain in sub-domains by creating boundaries between instances. This section will highlight two renowned examples, Support Vector Machines (SVM) and Kernels.

4.2.1 Support Vector Machines

Support Vector Machines, as the name suggests, make use of vectors for the purpose of learning. A decision boundary is established based on support vectors that maximise the distance, i.e. margin, between boundary and the data points by solving a constraint maximisation problem. Equation (4.2) shows this optimisation for a two class classification problem. Note the λ term is the so called regularisation term, preventing the system for overfitting. This method is frequently applied for classification problems where the decision boundary basically determines to what class an instance belongs. An optimisation is usually prone to local minima, but this is not the case as this optimisation is convex by nature [19]. Figure 4.2 summarises this method by means of an example. Clearly H_1 fails in its task to be a boundary between the two classes. H_2 and H_3 do succeed, it is however obvious that H_3 performs better in maximising the distances to the boundary.

Markou and Singh discuss some pro's and con's of this method in their survey. A pro is said to be the absence of any probability density fitting. This is however a fairly common advantage of learning algorithms and is therefore a general remark for most methods discussed in this paper. According to Markou and Singh, a major disadvantage is the need for a large data set, particularly in case of highly dimensional input vector [30]. Besides that, the inability to cope with variation in density is troublesome since instances in low density areas might be rejected too easily. Nevertheless, Kotsiantis et al. prefer SVM's, and Neural Nets, for multi-dimensional, continuous data [26]. Furthermore, these authors state that Support Vector Machines do not require a large data set relative to the number of features considered. This is because the boundary is a linear combination of instances that make the boundary.

$$y(\mathbf{x}) = \mathbf{w}^T \Phi(\mathbf{x}) + \mathbf{b} \tag{4.1}$$

$$J(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \left(\mathbf{w}^T \Phi(\mathbf{x_n}) - t_n \right)^2 + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w} \tag{4.2}$$

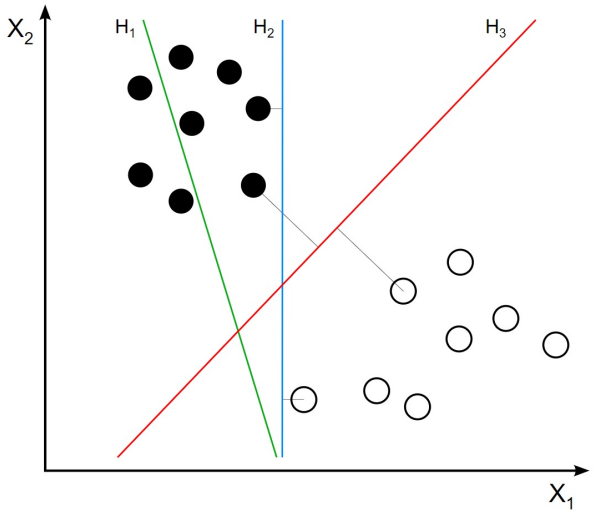


Figure 4.2: Linear support vector machine for two class classification. *Courtesy:* Wikimedia Commons

4.2.2 Kernels

If the data is not linearly separable, Kernel functions are a potential solution. Kernel functions translate the data points to a hyperplane where the data might become linearly, or at least more linearly, separable than in its original plane. Figure 4.3 illustrates this principle where a quadratic Kernel is

applied. Equation (4.3) depicts the most general Kernel function $k(\mathbf{x},\mathbf{x}')$. With some mathematically sound operations new Kernels can be constructed. Multiplication with a constant or summation of two Kernel functions are for instance valid Kernel operations [19]. The most frequently applied Kernel is the Gaussian Kernel, shown by Equation (4.4) [29, 30].

$$k(\mathbf{x},\mathbf{x}') = \Phi(\mathbf{x})^T \Phi(\mathbf{x}') \tag{4.3}$$

$$k(\mathbf{x},\mathbf{x}') = \mathrm{e}^{\frac{||\mathbf{x}-\mathbf{x}'||^2}{2\sigma^2}} \tag{4.4}$$

The selection of suitable Kernels is one of the major challenges in the application of this technique. Hence, the freedom in building new Kernels seems a welcome solution, but due to the many options is expensive to consider in an optimisation for instance. Pimentel et al. call this decision-making "problematic" and points out that selecting the number of variables that control the size of the boundary region is challenging [18]. They conclude with addressing the computationally complexity of domain-based methods i.e. extensive training but relatively fast testing. James et al. however state that, in a trade-off situation, Kernels do have a computational advantage over enlarging the feature space [17].

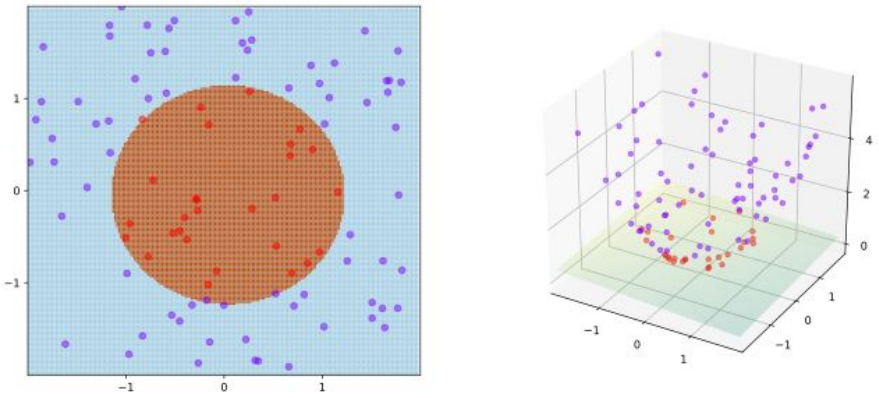


Figure 4.3: Quadatric Kernel. *Courtesy:* Wikimedia Commons

4.3 Reconstruction-Based Algorithms

Reconstruction-based algorithms are capable of autonomously modelling hidden data by presenting test data to the trained model. Neural Networks are the most prevailing in this category.

4.3.1 Neural Networks

Neural Networks (NN) are a network-based learning algorithms consisting of inputs, outputs, hidden layers and weights. Figure 4.4 illustrates an arbitrary two-layer network for input vector $\mathbf{x}$, output vector $\mathbf{y}$, hidden layer $\mathbf{z}$ and weights matrix $\mathbf{W}$. In addition, two bias terms, indicated with zeros, are part of this simple network. The most widely used network topology is the Multi Layer Perceptron (MLP) network [30].

The network works as follows. Inputs are multiplied with the first row of weights and then subjected to an activation function. This function quantifies the activity of that cell which usually ranges from 0 (inactive) to 1 (active). This activation function is often the renowned Sigmoid function, also seen in Logistic Regression. This function yields a binary which is then again multiplied with the weights of the subsequent row. This procedure is repeated $N+1$ times, where N is

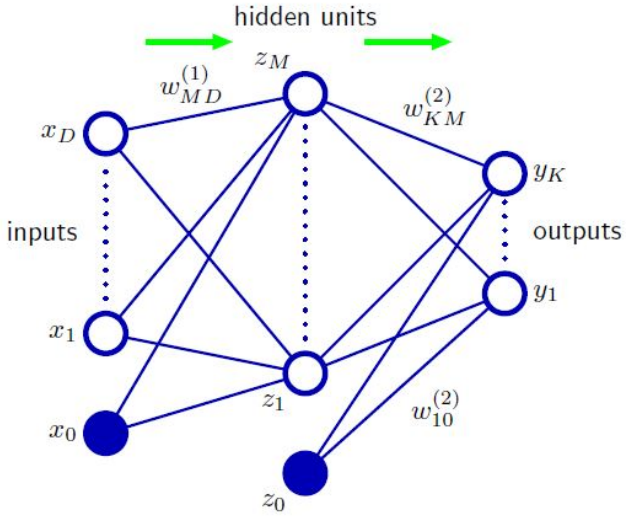


Figure 4.4: Network diagram for a two layer Neural Network [19].

the number of hidden layers. Equation (4.5) shows this process mathematically, note σ is the Sigmoid function. In a supervised context the error is computed and the weights adjusted accordingly. The error is defined as shown by Equation (4.6). A common method for weight tweaking is backpropagation, a first order method that backpropagates the error found in the final layer to the prior layers.

$$y_k(x,w) = \sigma\Big(\sum_{j=0}^{M} w^{(2)}_{kj} h\Big(\sum_{i=0}^{D} w^{(1)}_{ji} x_i\Big)\Big) \tag{4.5}$$

$$E(w) = \frac{1}{2}\sum_{n=1}^{K} ||y(x_n,w) - t_n||^2 \tag{4.6}$$

Karlaftis and Vlahogianni state in their survey that Neural Nets have been in used in their research area (transportation) primarily due to its ability to deal with large data sets of highly dimensional data, its resilience and its ability to generalise, besides the good predictive results obviously [16]. The difficulty in applying this method is in finding the right architecture, as the results could be sensitive to the topology of the network i.e. prone to overfitting [18, 26]. The architecture is defined by the number of hidden units per layer, the number hidden layers and the number and orientation of the links i.e. weights. Compared to other methods, Markou and Singh find this number of parameters fairly small suggesting the problem is limited [30]. Pimentel et al. however says its troublesome in highly dimensional data sets [18]. In addition, a solution for this problem is discussed, namely constructive algorithms. These algorithms allow the structure of the network to grow, which usually yields better results. Potential overfitting is now a consequent struggle which relies on the stopping criteria, a new set of hyperparameters that is to be decided upon. Irrelevant features are another feature related concern according to Kotsiantis et al., potentially making the neural net training inefficient [26].

Bishop says novel input data is the largest source of error for Neural Nets [31]. Therefore he presents a measure for the novelty of inputs by modelling the unconditional probability density of the input data during training. Error bars can give a quantification to the outputs or assign the novelty of the inputs. This remark however does not stand alone as the subsequent will show by elaborating on the effect of training and novel data on the quality of the results.

The weight updating mechanism, such as backpropagation mentioned earlier, might also affect the computational time as well as the error. Dharia and Adeli argue that backpropagation is computationally expensive since it alters the weights of all links [32]. The authors show that a counter propagation network (CPN) yields similar or better accuracy at significantly fewer iterations for a travel time forecasting problem. The major difference between the methods is in building the architecture. While the BPN topology is determined by mostly trial-and-error, a CPN consists of three layers: input, competition and interpolation. In short, choosing the right updating mechanism is crucial to the time performance.

To conclude, Bishop and Pimentel et al. say neural nets are regularly used for safety critical applications, an observation that implies good reliability in this critical research area and aligns with the upcoming study [18, 31].

4.4 Distance-Based Learning

Distance-based or density-based learning is a group of learning algorithms that have a particular feature in common, the use of a distance measure. This distance measure could be either between data points or randomly specified points and its computation is not unambiguous either. Though the fundamental decision-making mechanism is a distance measure, of which a common example would be the Euclidean distance.

4.4.1 K-Means Clustering

Clustering in general is an learning approach that tries to establish clusters with somehow significantly related features. Clustering is often used in an unsupervised context, where the clusters of data are to be identified amongst many instances. Note this algorithm implicitly assumes that all instances of a particular class are in close vicinity. This very strong assumption is a downside of the method. Figure 4.5 shows an example of the results of such an application. The instances in this domain are labelled by different colours, as the result of a clustering algorithm that basically ascribes an instance to a cluster until all instances are assigned. Instances outside the cluster are indicated as anomalies.

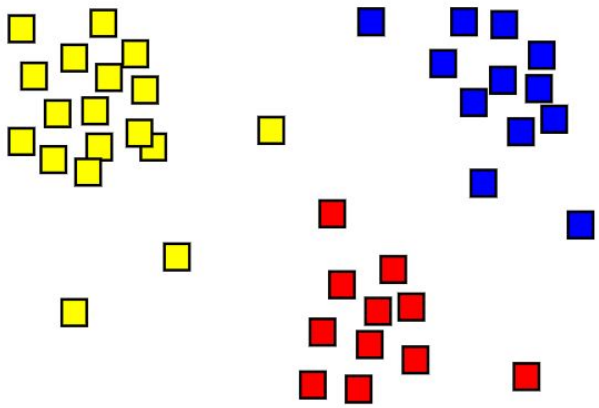


Figure 4.5: Simple clustering algorithm. *Courtesy:* Wikimedia Commons

K-Means clustering is one of many clustering method that is widely utilised. K-Means randomly initiates the centroids of K clusters and assigns instances to these K clusters based on the distance to the centroids. The mean of cluster instances is then computed and the centroid shifted to this

position. The instances are reassigned and the mean again calculated in the new situation. This iterative process is repeated until the shift of the centroid is within a certain threshold, i.e. the solution converges. A disadvantage of this method is that due to its iterative nature the distance to the centroid is computed for each iteration and for each data point, making it rather expensive.

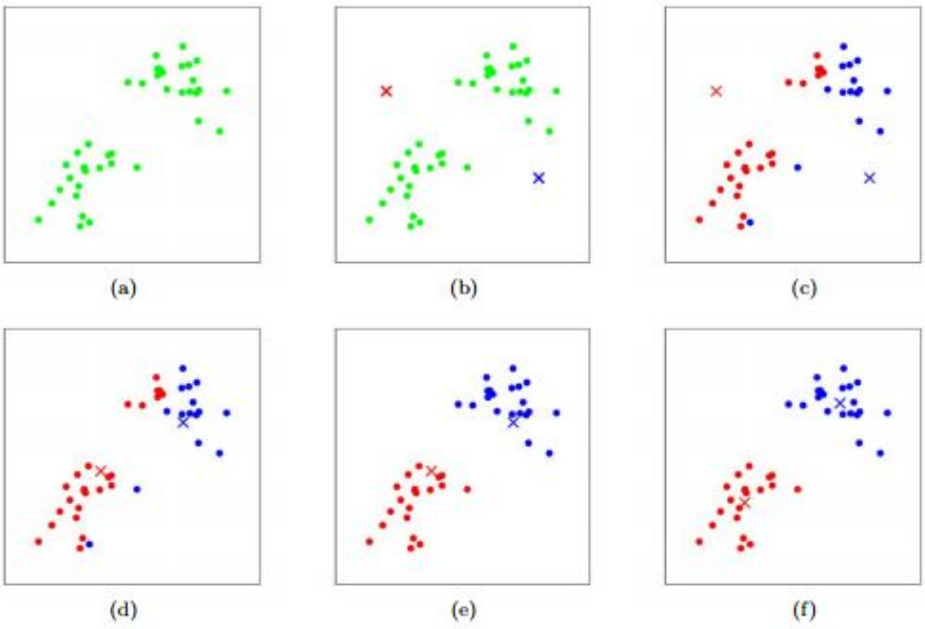


Figure 4.6: Progress of a K-means algorithm for a two-class classification. *Courtesy:* Stanford University

It is evident that the fairly random initiation of the cluster cores has an impact on the final results. James et al. therefore suggest to repeat the procedure for multiple initial values and select the best result [17]. Extreme sensitivity to this initiation would make the results already questionable. Furthermore, the effect of number of clusters is to be evaluated. All in all, this method is deemed fairly prone to its hyperparameters. It is similarly sensitive to outliers that could affect the mean of a cluster drastically during the iterative procedure. The right number of clusters should however prevent this phenomenon of occurring. Alternatively, K-Medians could be applied instead. K-Medians computes the median rather than the mean of each cluster.

4.4.2 Nearest Neighbours

A Nearest Neighbours (k-NN) algorithm decides upon the nature of a new instance based on the k nearest neighbours of that particular instance. In a classification problem this means an input belongs to A if a significant number of its neighbours are A. In contrast with the above, this method assumes an instance is surrounded by instances of the same class [25]. Hence, these objects do not have to clutter all together in a single cluster i.e. class. Still, all points are to be considered for the assessment which remains computationally expensive. Figure 4.7 shows a two-class classification problem that tries to assess a new instance, depicted in green. It seems obvious this instance belongs to the red instances but by increasing the number of k the assessment might yield a different outcome.

Chandola et al. state that semi-supervised approaches are preferred over fully unsupervised approach for these kind of applications, since the likelihood of an anomaly to be in the same 'neighbourhood' is low [25]. Kotsiantis et al. address the sensitivity of this method to irrelevant features, a characteristic of how distance-based algorithm function and therefore cannot easily handle dimensionality. This working principle is fairly intuitive, an advantage of k-NN, although its interpretabil-

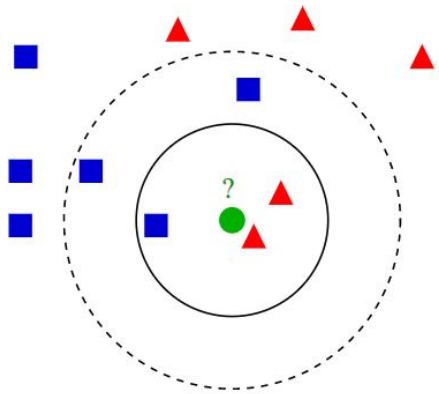


Figure 4.7: A Nearest Neighbour algorithm for a two-class classification. *Courtesy:* Wikimedia Commons

ity is poor because of *the unstructured collection of training instances.* Its resistance to noise is also poor, as the similarity measure to determine its nearest neighbours could be affected significantly by random variations [26].

The difficulty in all distance-based methods is the distance metric. Finding the appropriate measure is a delicate task that is determined by the nature of the data. If found however, any problem can be implemented in an algorithm by simply changing the distance measure [25]. Nonetheless, it might suffer from the curse of dimensionality. That is because in a highly dimensional space the distance between vectors is relatively small, meaning Nearest Neighbours for instance performs poorly in discovering novelties. These methods are usually also fairly computationally expensive as the distances are computed many times [26]. Clustering on the other hand handles this scalability issue to a much better extent. Current research is focusing on improving the time performance by for instance defining a cluster radius c.q. width [18]. An application of such a method will be discussed subsequently. This method however introduces more hyperparameters that are to be decided upon and these demand analysis on the sensitivity thereof. The selection of clusters is likewise a sensitive number that affects the performance, particularly in a highly dimensional feature space where instances could be assigned to particular classes too easily. Contrarily, in low density regions the absence of nearby instances could result in false conclusions as well [25].

4.5 Rule-Based Algorithms

Rule-based techniques base their prediction on learned rules that represent the regular behaviour of a dynamic system [25]. Decision Trees belong to this category and are explained here.

4.5.1 Decision Trees

Decision Trees are learning algorithms that are characterised by their accompanying tree representation. This representation is the outcome of a tree building procedure consisting of roughly two steps. First, the feature space is divided over N regions. Figure 4.8 illustrates this principle for the features X_1 and X_2 of which the spanned domain is divided over N regions R_i with $N-1$ boundaries t. This step is often done by recursive binary splitting and is repeated until a minimum number of instances per region is found. Recursive binary splitting is more efficient than considering all possible regions, but solves the problem greedily i.e. it only considers the best solution at that particular point in time. In other words, it lacks the ability to consider future steps, a shortcoming of this region identification method. The decision boundary t is defined by minimising the residual sum of squares or classification error rate, depending on whether the problem is a regression or classifica-

tion tree respectively. This criterion applies for all regions. If all regions are established, pruning is done by a regularisation like trade-off taking into account the number of leaves T. The aim of this trade-off to obtain the best sequence of sub-trees as previously the solution might have been overfitted. Lastly, it is important to highlight the difference between regression and classification trees. The response of regression tree is simply the mean of that region, whilst for a classification tree it is the most occurring instance in that particular region.

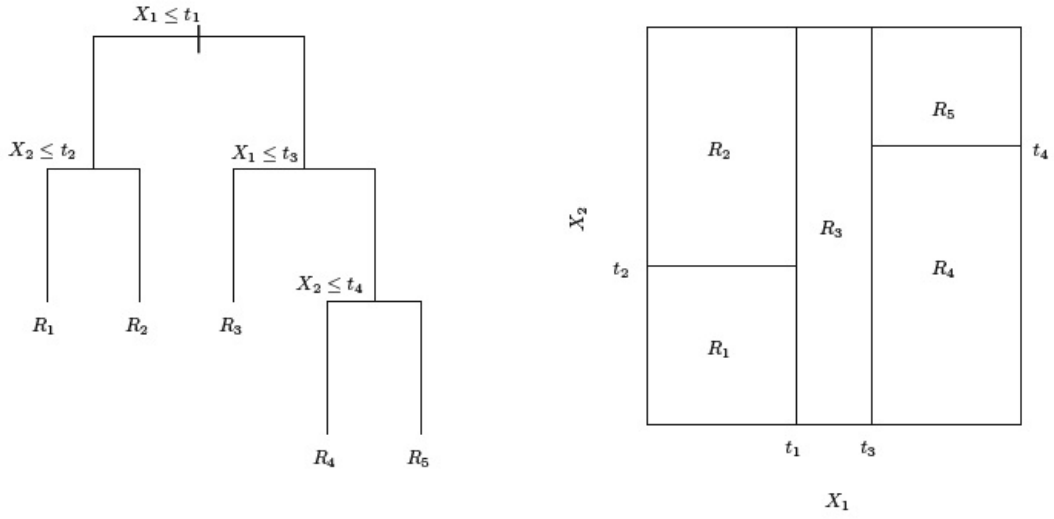


Figure 4.8: Two perspectives of a Decision Tree, with on the left the actual tree representation and on the right the splitted domain [17].

James et al. mention some advantages and disadvantages that can be reduced to two straightforward statements. First, Decision Trees are very convenient in terms of explanation and can serve a wide public due to its simple graphical representation [17]. And second, that this comes at the cost of accuracy, though with by aggregating trees with other methods, such as Random Forest and bagging, the predictive performance can be enhanced. Kotsiantis et al. agree with the convenience and add that trees tend to have a better performance in a classification problem when presented with discrete and categorical features [26]. Furthermore, compared to Neural Nets, the training is usually less time-consuming and accuracy in the same order of magnitude. Pruning is pointed out as a benefit in dealing with noise, since trees usually do not overfit which is clearly beneficial for noise.

4.6 Ensembled Methods

Ensembled methods are 'simply' a combination of the algorithms discussed above and are discussed in the review by Kotsiantis et al. [26]. Though it seems overcomplicated, it has the advantage of combining the benefits or strengths of the individual methods. The aim of this approach is to tackle the weaknesses of one method by adding another method with that particular weakness as distinct strength. In the subsequent chapter an example will be discussed where a simple ensemble indeed achieves a higher accuracy. Ensembled methods have also some downsides, the unique consequence of combining the methods and not the particular shortcoming of a single method. First, more storage is required as the number of predictions proportionally scales with the size of the ensemble i.e. the number of methods involved. Second, the computation effort increases similarly, as each prediction is computed n times more. Third, the interpretability is affected negatively since an ensemble integrates the independent results to a perhaps less meaningful result. Hence, Kotsiantis

et al. suggest to use ensembled methods only if high accuracy is demanded and not achievable by a single method [26].

4.7 Method Selection Criteria

From the elaborate discussion of the methods above, the paramount criteria for selecting the most appropriate method can be derived. The advantages and disadvantages of each method were highlighted in this chapter and these are compiled in a set of six criteria: scalability, heterogeneity, dimensionality, complexity, interpretability and robustness. These criteria are listed below, including a short description. Note the first three criteria are depending on the nature of the data, these therefore function as requirements. The applicability of the methods is assessed by this criteria set.

- **Scalability** the ability to scale the problem by changing the size of the dataset without compromising performance
- **Heterogeneity** the ability to cope with heterogeneous data
- **Dimensionality** the ability to cope with dimensional data
- **Complexity** the effort required to build the method and the number of design decisions/variables
- **Interpretability** the extent to which the results are understandable or the effort needed to become (more) understandable
- **Robustness** the extent to which the method is sensitive to noisy features and/or changes in hyperparameters

In fact, Budalakoti et al. specified five criteria which are shown below [33]. By comparing the criteria it is possible to conclude on the validity of the criteria. It is noticed that the criteria by Budalakoti et al. are to a large extent aligned with the method selection criteria identified by reviewing the methods, validating the selected criteria. Comprehensibility and interpretability as well as robustness and robust focus on the same method characteristics. Though the latter identifies a somewhat different source of instability, namely expert input. This criterion is nonetheless governed by complexity criterion as it not only affects the applicability but also the general sensitivity of the methods. Uniqueness and repeatability are to be assessed by validation techniques discussed in Section 4.8.

- **Unique** The system should find a unique solution.
- **Repeatable** The system must provide a repeatable solution. Thus, each time the system is run it should identify exactly the same set of outliers.
- **Comprehensible** If the system identifies a set of outliers, it must also generate an explanation as to why the sequences were called outliers.
- **Robust** The performance of the system should not be critically dependent on the quality and amount of expert input. The initial solution should be as off-the-shelf as possible.
- **Scalable** The system performance should not substantially degrade as the number of sequences increases. Thus, the complexity of the algorithm must be better than an $O(n^2)$ computation.

4.8 Quality Assessment

Since many methods exist, it is sensible to review the quality assessment measures and considerations, as the best quality is obviously opted for. The assessment of quality is commonly done with widely accepted techniques and measures. This ensures there is a general understanding in the field. Throughout this section the book of James et al. is the general guideline [17].

4.8.1 Quality Considerations

Bias-Variance Trade-off

In the application of these kind of algorithms an intrinsic trade-off occurs that is worth explaining since the results and feasibility are at stake. This trade-off is called the bias-variance trade-off. The explanation of this trade-off demands a definition of the two aspects, bias and variance. As James et al. say *variance refers to the amount of which f would change if we estimated it using a different training data set.* In other words, variance assesses the sensitivity of the model with respect to the training set. This is undesired as any training set should more or less result in the same prediction. Then one could argue why not to use the entire training set instead, so the issue of variance is ruled out. However, this potentially introduces much bias in the outcome as *bias refers to the error that is introduced by approximating a real-life problem, which may be extremely complicated, by a much simpler model.* A highly biased model is often faced if the model is overfitted. A classic example is that of the n^{th} order polynomial that can fit $n-1$ data points precisely but has poor performance in predicting values of novel inputs.

Interpretability-Flexibility Trade-off

James et al. address another major trade-off in their book that determines the practical quality of the method, namely between interpretability and flexibility. Figure 4.9 illustrates the relation between the two concepts. This representation shows the relation is somewhat proportional and that the two are therefore to be traded off carefully. Hence, this underpins interpretability should be one of the feasibility selection criteria, derived in Section 4.7. In the operational context this research will be conducted in, both are almost equally decisive. Interpretability is paramount since this more sophisticated procedure, with respect to exceedance analysis, is to be widely accepted. Flexibility on the other hand since there might exist the desire to analyse more safety events, perhaps by applying a similar technique or even the same model. However, due to industry wide nature of the concern this requirement is somewhat less stringent.

4.8.2 Quality Metrics

Precision, recall, accuracy and the F1 score are four commonly used measures in the field of Machine Learning for assessing the quantitative quality of classification algorithms. These metrics are fairly simple and are fully based on model hypothesis testing as depicted by Table 4.2.

Table 4.2: Hypotheses for model testing.

		ACTUAL	
		Positive	**Negative**
MODELLED	**Positive**	True positive	False positive
	Negative	False negative	True negative

Precision is defined as the rate of true positives over all true positives. Hence, a low precision means the false positive rate is high. Equation (4.7) shows this quality metric.

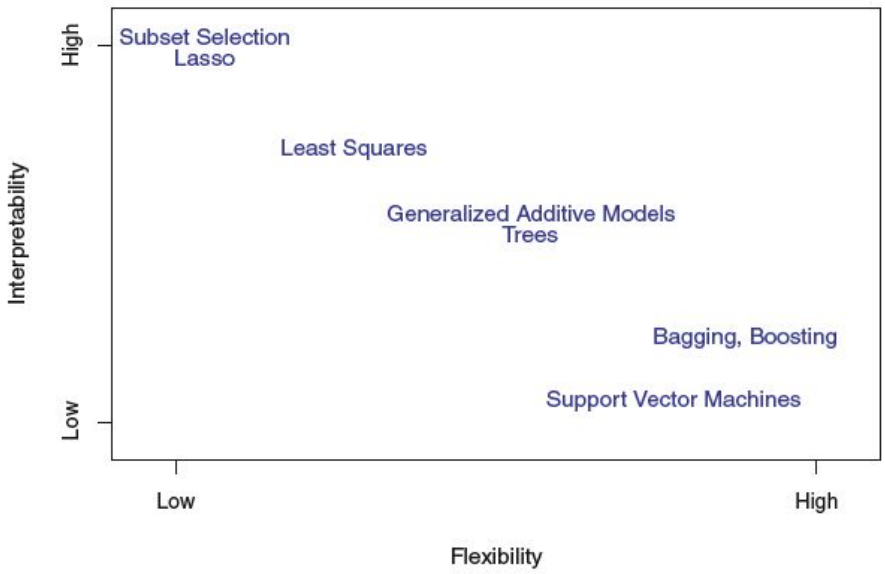


Figure 4.9: Graphical representation of the interpretability-flexibility trade-off [17].

$$P = \frac{\text{True Positive}}{\text{True Positive + False Positive}} = \frac{\text{True Positive}}{\text{All Predicted Positive}} \tag{4.7}$$

Recall represents the accuracy in terms of determining positives, it is also called the true positive rate accordingly. A low recall means many positives are missed by the model. Equation (4.7) shows this quality measure mathematically.

$$R = \frac{\text{True Positive}}{\text{True Positive + False Negative}} = \frac{\text{True Positive}}{\text{All Actual Positive}} \tag{4.8}$$

Accuracy is the most straightforward measure and is simply the correct predictions with respect to all predictions.

$$A = \frac{\text{True Positive + True Negative}}{\text{All}} \tag{4.9}$$

The F1 score is a combination of the precision and recall measures and more generally assesses the quality of a model. The definition is shown by Equation (4.10). Obviously, if the precision and recall are both 1, so is the F_1.

$$F_1 = 2\frac{PR}{P+R} \tag{4.10}$$

The subsequent sections treat different validation techniques that are frequently used and are discussed in the book of James et al. The validation techniques face a similar bias-variance trade-off as will be argued after the high level explanation of each method.

4.8.3 Validation Methods

Validation Set

Validation set or hold-out set is a method that splits the available data set in a training set and validation set. The latter is then the so called hold-out set. The training set is used to train the model and the validation set to assess the eventual quality. Herrema et al. apply this method by dividing their data set in sets of 70-15-15% for the training, testing and validating respectively i.e. two hold-out sets [34]. Testing and validating are fundamentally different since the testing set is to tweak

the model hyperparameters. As an example, the testing set would be used to determine number of layers and cells in a Neural Network context.

Leave-One-Out Cross Validation

As the validation set approach, leave-one-out involves dividing the data set into two separate sets. It is important, for all validation methods, that these sets are independent i.e. have no overlap. However, instead of two subsets of similar size, a single observation is taken for the validation while the remaining data points make up the training set. This procedure is repeated n times, where n equals the number of unique instances so that the independence criterion is indeed met. The cross validation error is simply the averaged sum of the n errors, as shown below.

$$CV_n = \frac{1}{n} \sum_{i=1}^{n} MSE_i \tag{4.11}$$

K-Fold Cross Validation

K-fold cross validation is an alternative to leave-one-out. This approach randomly divides the set of observations into k folds of approximately equal size. The first group is the validation set, and the method is trained on the remaining $k-1$ groups. The MSE is computed on the observations in the held-out fold. These steps are repeated k times. This results in k estimates of the test error. The k-fold estimate is calculated by averaging these errors as depicted by the equation below. Note the error computation is analogous to the leave-one-out error.

$$CV_k = \frac{1}{k} \sum_{i=1}^{k} MSE_i \tag{4.12}$$

K-fold clearly has a computational advantage over the leave-one-out approach, as the dataset is divided over smaller number of folds. K-fold often has a more accurate test error rate estimate too, because of a bias-variance trade-off similar to the one discussed above. Obviously the validation set could result in overestimations of the test error rate since it uses only small slice of the available data. It is on the other hand the cheapest in terms of computational effort. Logically, the leave-one-out approach has least bias as the variations in validation sets are small. According to this reasoning the K-fold is supposed to be somewhat in between. As one of the previous sections mentioned, there is an intrinsic trade-off with variance. Leave-one-out has higher variance due to the strong positive correlation between the data folds. K-fold has less overlap and therefore results in a marginally lower variance. As the validation set has no overlap, it has the least variance.

From the above it becomes clear that the quality metrics are fairly straightforward, and that the methods are not overly difficult either. Though the latter seem to have a major effect on the quality and performance of the quality assessment. To decide upon what method to use a trade-off is to be made between the computational effort and accuracy, i.e. bias-variance, to select the most appropriate as well as feasible validation technique.

4.9 Feature Selection

Feature selection or variable selection is the process of selecting the parameters that will be fed to a ML algorithm. Feature selection is in fact a field of studies, called feature engineering, due to its major impact on the time to train and predictive performance [35, 36]. This section first discusses the general definition and considerations in Section 4.9.1. Section 4.9.2 describes some methods frequently observed in literature. At last, Section 4.9.3 establishes a procedure for tackling this problem during the ensuing research.

4.9.1 General Definition and Considerations

Before discussing the topic of feature selection it is essential to lay down a definition. Kira and Rendell defined this process of feature selection as follows [36]:

> *Feature selection is the problem of choosing a small subset of features that ideally is necessary and sufficient to describe the target concept*

This definition requires elaboration of the key terms, sufficient and necessary. Sufficient in this context basically means such that the method returns acceptably accurate results, or in other words, the results are reliable. As an example, an algorithm could yield poor performance due to overfitting, as the result of having too many features. This was already touched upon in Section 4.4 where it was stated that the effectiveness of a distance metric degrades with increasing features. Necessary relates to the computational effort needed, the small subset should ideally be as small as possible to be able to feed large data sets at a low cost, i.e. train fast. Besides these two there is third objective according to Guyon and Elisseeff, namely providing insight into the underlying data generating process [35]. Recall one of the fundamental assumptions of Machine Learning applications is that the underlying data generating process is dynamic and unknown, hence setting this objective on feature selection is the logical consequence of this assumption. Chandola et al. also stress the importance of this process regarding method selection by stating *the nature of attributes determine the applicability of anomaly detection techniques* [25]. Hence, the nature of the features i.e. data is again proven to be a very important in method selection. Figure 4.1 shows this principle by deciding upon the features first.

4.9.2 Feature Selection Techniques

The subsequent paragraphs shortly discuss some of the observed techniques for feature analysis to familiarise with the concepts and the reasoning behind these techniques. This shortlist are the most frequently observed techniques in the papers discussed in Chapter 5.

Principal Component Analysis

Principal Component Analysis (PCA) is a dimension-reduction method, applied to scale down the number of features present in a feature space by only considering the features with the largest variation. In other words, PCA ensures only the 'interesting' features are taken into account for future analyses and inherently assumes variation is a good indication. Though some researchers argue that latent features might be as important and feature selection negatively effects the results, as will be discussed in the next chapter. In a PCA the first step is the normalisation, so that effects due to the magnitude of parameters are ruled out. The first principal components are the features $\mathbf{X}$ of which the variance of the linear combination, shown in Equation (4.13), is highest. The loadings Φ_{i1} have to fulfil the condition depicted by Equation (4.14). The remaining principal components are also ranked based on this parameter and ranked list is established [17]. This list allows to pick the N most varying parameters as resulting features. The number N strongly depends on the computational cost of adding features. It is furthermore a parameter that should be subjected to accuracy as well as sensitivity analysis.

$$PC_1 = \Phi_{11}X_1 + \Phi_{12}X_2 ... + \Phi_{p1}X_p \tag{4.13}$$

$$\sum_{j=1}^{N} \Phi_{j1}^2 = 1 \tag{4.14}$$

Relief
Kira and Rendell developed a feature weight based methodology named Relief [36]. This algorithm requires two inputs, the relevance threshold τ ($0 \leq \tau \leq 1$) and a sample size m. For each instance X Relief determines the near-hit and near-miss instance of m triplets. The near-hit and near-miss are calculated with the Euclidean distance. A routine is subsequently called upon to update the feature weight vector **W** for every triplet, this is seen in Equation (4.15), where NH and NM represent the near hit and miss respectively. The relevance is simply the mean of the weight vector. If the relevance exceeds the predefined threshold, then the features are selected for further use.

$$W_i = W_i - (x_i - NH_i)^2 + (x_i - NM_i)^2 \tag{4.15}$$

Expert Judgement
Expert judgement is often regarded as one of the weakest feature selection methods as it has no mathematical foundation at all. It is merely an individual with a domain knowledge who points out relevant features in his or her personal opinion. Experts however have years of experience and gained extremely valuable knowledge that cannot easily be gathered by looking at data. As illustration of its valued contribution, it is in fact the first question posed by Guyon and Elisseeff, these questions are treated subsequently [35]. Furthermore, there do exist methodologies to soundly combine and analyse these findings as Goossens et al. review in their paper discussing fifteen years of developments in this particular area [37]. Chapter 5 will proof how widely preferred this technique still is over more advanced mathematical methods, or how expert judgement is combined with these methods for the purpose feature selection.

4.9.3 Feature Selection Procedure
Guyon and Elisseeff pose ten questions any researcher should answer in order to find the right features for their data analysis problem [35]. These questions are listed below. By answering these questions a procedure is subsequently derived for tackling the problem at hand.

1. Do you have domain knowledge? If yes, construct a better set of "ad hoc" features.
    - Yes, the upcoming study is done in collaboration with the industry so sufficient domain knowledge is present.
2. Are your features commensurate? If no, consider normalising them.
    - Likely not, as an example, the gross weight of an aircraft affects the selected approach speed.
3. Do you suspect interdependence of features? If yes, expand your feature set by constructing conjunctive features or products of features, as much as your computer resources allow you.
    - Yes, interdependence is expected as for instance flight data contains control variables that affect the physics of an aircraft.
4. Do you need to prune the input variables (e.g. for cost, speed or data understanding reasons)? If no, construct disjunctive features or weighted sums of features (e.g. by clustering or matrix factorisation).
    - Pruning variables is plausible as overcomplicated features would be the understanding of the results and have no operational functionality.
5. Do you need to assess features individually (e.g. to understand their influence on the system or because their number is so large that you need to do a first filtering)? If yes, use a variable ranking method; else, do it anyway to get baseline results.

- Yes, the influence of features is of major importance for enhancing processes within an airline. Significant precursors obviously get higher priority.

6. Do you need a predictor? If no, stop.
   - A predictor or precursor is desired considering the current point of view regarding safety management.
7. Do you suspect your data is “dirty” (has a few meaningless input patterns and/or noisy outputs or wrong class labels)? If yes, detect the outlier examples using the top ranking variables obtained in step 5 as representation; check and/or discard them.
   - If the event is monitored, the classification has been validated by a safety data analyst, hence it is unlikely a significant number of labels is wrong. Raw flight data might be dirty and require a ranking check, an example will be given in the next chapter.
8. Do you know what to try first? If no, use a linear predictor. Use a forward selection method with the “probe” method as a stopping criterion or use the 0-norm embedded method. For comparison, following the ranking of step 5, construct a sequence of predictors of same nature using increasing subsets of features. Can you match or improve performance with a smaller subset? If yes, try a non-linear predictor with that subset.
   - These steps will be followed unless explicit knowledge is gathered. Regardless, sensitivity of the features will be assessed for quality and operational purposes.
9. Do you have new ideas, time, computational resources, and enough examples? If yes, compare several feature selection methods, including your new idea, correlation coefficients, backward selection and embedded methods. Use linear and non-linear predictors. Select the best approach with model selection.
   - Time and computational resources are the sole restrictions in trying different methods, with a data base of 500,000 flights the number of examples is probably excessive.
10. Do you want a stable solution (to improve performance and/or understanding)? If yes, subsample your data and redo your analysis for several “bootstraps”.
    - One of the cross-validation techniques discussed in Section 4.8 will definitely be applied for this purpose.

The answers given above can be summarised in a step-by-step procedure that will be applied for the future study. This procedure is repetitive for a to be determined number of feature selection techniques. In an early stage however the exact number cannot be determined yet due to unknown time restrictions. Nonetheless, for verification purposes it should be larger or equal to two.

1. Clean data if data suspected to be "dirty".
2. Apply operations to features if necessary/relevant.
3. Normalise all available features.
4. Assess independence of feature set $\mathbf{X}$.
5. Mathematically determine relevant features $\mathbf{X}'$ by means of method A.
6. Cross-validate the results for different subsets.
7. Select a new method A.
8. Repeat 1-7 until acceptable results.
9. Prune feature set $\mathbf{X}'$ in close consultation with industry expert.
10. Assess the effect of the expert pruning set
11. Finalise feature set

5 Applications of Machine Learning Techniques

This chapter starts off with discussing the applications of the techniques explained previously in Section 5.1. The focus of this section will be on the relevant assumptions and considerations of fellow researchers. These are summarised in Section 5.2. The general aim of this chapter is to identify the research gap, which laid down in Section 5.3.

5.1 Applications in Safety Data Analysis

This section discusses applications of Machine Learning techniques in the field of Safety Data Analysis in particular. Each of the following paragraphs is uniquely indicated by the author(s) of the research(es).

Chidester & Amidan et al. One of the earliest applications of ML in safety data analysis is that of the Morning Report [38, 39]. This unsupervised multivariate clustering method detects anomalous flights based on an atypicality score. Morning Report calculates statistical signatures across the parameters selected by PCA. Clusters of the flights were formed according to these signatures. The Mahalanobis distance, with respect to the centroid of a cluster, is calculated to identify anomalous flights. The overall data was reduced by focusing on data between certain events, such as T/O power and gear retraction at take-off, as well as narrowing down the problem to a single runway. The quality could however not be assessed due to the absence of labelled flights, a situation that is deemed highly unlikely for the problem at hand, as argued in Section 4.1. The major difficulty of this method was the determination of the anomaly boundary, i.e. the set of anomalous flights to be reviewed by the expert. Hence, this application was operationally sensitive, a characteristic that Budalakoti et al. explicitly state as unwanted since it will not meet the robustness criterion laid down by these researchers [33].

Budalakoti et al. SequenceMiner is an unsupervised clustering algorithm developed by Budalakoti et al. [33]. The aim of this research was to detect and characterise discrete sequences of button switching in the approach phase. Unsupervised clustering was chosen for its good scalability characteristics, a criterion identified earlier. The authors explain the concepts of their aim carefully by stating the following definitions:

- **Anomaly detection:** Given a set of sequences S, identify anomalous sequences, i.e., sequences that are considerably different from the other sequences in S.
- **Anomaly characterisation:** For a sequence S_i in set S that is already identified as anomalous, describe the reasons why S_i was, or should be, identified as anomalous.

Based on these two definitions it can be said that the upcoming study is foremostly focusing on characterising the anomalies as the detection is likely already done, again referring to Section 4.1. The SequenceMiner algorithm does not perform any form of feature selection as the input vectors consist of discrete sequences. Hence, the size of the input vector presented to SequenceMiner varied widely (1,500 points on average), an advantageous feature of this clustering algorithm. To lower the computational effort the focus of the study was reduced to the approach phase at a single runway. This ensured procedures at all 2,200 flights were similar to the large extent, as ideally these would

be when applying a clustering algorithm. It was assumed that *the probability of generation of the outlier from each sequence is proportional to the normalised LCS score between the sequence and the outlier*. The centroid of that cluster was the representation of the entire cluster and other sequences were modelled as if generated by the centroid with a certain probability. Besides that, a probabilistic model was developed that could find and reconstruct missing symbols. This is basically an entirely different approach to the data cleansing mentioned in Section 4.9. Especially in sequential data sets shorter, common patterns might exist, even within the sequences itself. These could for instance originate from procedures. Reconstruction is then definitely worth the effort since otherwise valuable data is lost too easily. The most anomalous flights were validated by a pilot, one of many examples of an expert in the loop.

Das et al. Das et al. applied a multi kernel technique named Multi Kernel Anomaly Detection (MKAD) to detect anomalies in operational flight data [3]. This method was chosen for its flexibility since it has the ability to cope with heterogeneous data from multiple sources. This characteristic is very valuable as the selection criteria found in Section 4.7 underpin. To apply this method it was assumed that the normal flights have fairly consistent pattern, an assumption that Herrema et al. explicitly deny in their paper discussed later on [34]. Furthermore, it was assumed that discrete data streams affect the continuous streams and that atypical sequences in discrete data can result in non-nominal performance. This hypothesis is strictly necessary if the relations between the data (sub-)sets are uncertain i.e. are part of the system of unknown dynamics. Moreover, two measures were taken to reduce the size of the immense data set. First, it was decided to only consider data points below 10,000 ft. This reduction was deemed reasonable as the majority of events occur in these phases. In addition, an expert was consulted to further cut down the size of the set by selecting relevant parameters, indeed showing how experts are consulted for feature selection as argued in Section 4.9.2. These parameters were both continuous and discrete, verifying the selected method. A total of 2,500 'clean' flights were then presented to the model, dropping almost 1,000 of the available flights as the result of a conservative cleansing procedure, the first step of the feature method selection procedure established in Section 4.9.3. The significant loss in data stresses the need for such a step when using for instance raw flight data. Equation (5.1) shows the chosen Kernel, a summation of two Kernels, one for discrete and one for continuous data. This is a valid operation according to Bishop, an unique characteristic of Kernel methods that can aid in coping with highly heterogeneous data [19]. Subsequently, the results of the MKAD were verified with two baselines, Orca and SequenceMiner, since MKAD attempts to combine both philosophies. This verification technique is an important notion in this paper as regularly no perfect verification method exists. This is simply the consequence of the distinct impact of nature of the data on the method selection. Therefore an 'ensembled' verification is an inventive solution to this problem. However, this verification is more time consuming as well as computationally expensive but regularly unavoidable.

In 2011 the same research group conducted a case study on a different significantly larger data set, 174,000 flights. This study was done under the same assumptions and conditions mentioned above [40]. Though a different set of variables was chosen, with only 26 variables of different data types. Despite not being explicitly stated, the computational effects of this new set are minimal considering total data size did not different substantially. Furthermore, this application showed some flexibility in dealing with different inputs. This is a particularly valuable feature in an operational context where a model would preferably be used for other investigations as well. Results were this time not verified but validated with an safety data analysis expert, underpinning the value of experts in this sense too. Das et al. conclude this work with proposing text as additional future feature, requiring another text-oriented Kernel. This text could be retrieved from ASR's the author state, a process explained in Section 2.4. This statement furthermore suggests that relevant relations are expected between in-flight anomalies and the text in ASR's. This train of thought sounds reasonable

as for many safety related events or occurrences reporting is mandatory.

$$k(\mathbf{x_i}, \mathbf{x_j}) = \eta K_d(\mathbf{x_i}, \mathbf{x_j}) + (1-\eta) K_c(\mathbf{x_i}, \mathbf{x_j}) \tag{5.1}$$

$$K_d(\mathbf{x_i}, \mathbf{x_j}) = \frac{||LCS(\mathbf{x_i}, \mathbf{x_j})||}{\sqrt{l_{x_i} l_{x_j}}} \tag{5.2}$$

Li et al. Li et al. worked on an unsupervised clustering algorithm named ClusterAD [15]. This research aimed at flagging anomalies in highly dimensional time series i.e. QAR data. Dimensionality is here proven to be a motive as argued in Section 4.7. As other researchers, it was assumed that anomalous flights would significantly deviate from the common pattern or, in other words, would not belong to a compact cluster. The methodology is divided over three steps, translation, reduction and analysis. First, the data is translated from the time series to a highly dimensional vector. Therefore, specific events and windows were defined to reduce the data set to vectors. An example of such an event and window are T/O thrust and the 90 seconds afterwards respectively. Still, an additional cutback was required and that is why PCA was applied and the K Principal Component that explained more than 90% of the variance selected. Hence, these researchers implicitly assumes variance is a good indicator for features, a point of view not all researcher support as will be discussed subsequently. These reduced vectors were inputs to the density-based clustering algorithm. The methodology is summarised in Figure 5.1. The clustering method required two inputs, a minimal number of data points per cluster and a maximum radius. A sensitivity analysis was undertaken that showed ClusterAD was only sensitive to the mean radius but had a rapidly converging trend. The clusters were generated by the DBSCAN algorithm developed by Ester et al. [41]. The reason for applying this algorithm was a three-fold as density-based clustering models 1) can handle outliers in the data 2) do not have the need to know the number of clusters in advance 3) can discover clusters of arbitrary shape.

A case study was done with 365 Boeing 777 flights where each flight contained 69 flights parameters. To reduce the size of the data, only the 90 seconds after T/O thrust were considered. The found anomalies were classified according to the top x percentage accordingly. In this data set three clusters were discovered: normal flights, take-offs from certain airports and reduced power take-offs. This showed ClusterAD is capable of identifying nominal data patterns, although it is fairly limited by the transformation step as a predefined event is required. To conclude, Li et al. announced further work such as verification with MKAD and validation with an industry expert [15].

In 2015 actually published a paper that discussed the verification with MKAD and threshold analysis with a larger data set [42]. Results showed that ClusterAD and MKAD assisted the domain expert in exposing abnormalities in a large set of flights, some that were even identified as high level risks. This comparison made clear that both methods have certain strengths based on the marginal overlap. ClusterAD is more reliable in detecting deviations in continuous data patterns whilst MKAD recognised abnormal sequences of discrete parameters. Conclusively, Li et al. suggest a hybrid of the two methods, an ensemble, would yield more satisfactory results. This struggle is observed in more researches, i.e. two independent methods seem to be functioning well for a particular issue and generalisation is still needed for improving the operational feasibility of the method.

Herrema et al. Herrema et al. conducted a feasibility study with ten different techniques to find the best performing solution for their runway throughput problem [43]. This performance based method selection is ideal in finding the best method though computationally expensive. The goal of this study was to predict the time to fly and aircraft speed profile with flight data so controllers could improve their decision-making without compromising safety. A total of 15 flight parameters,

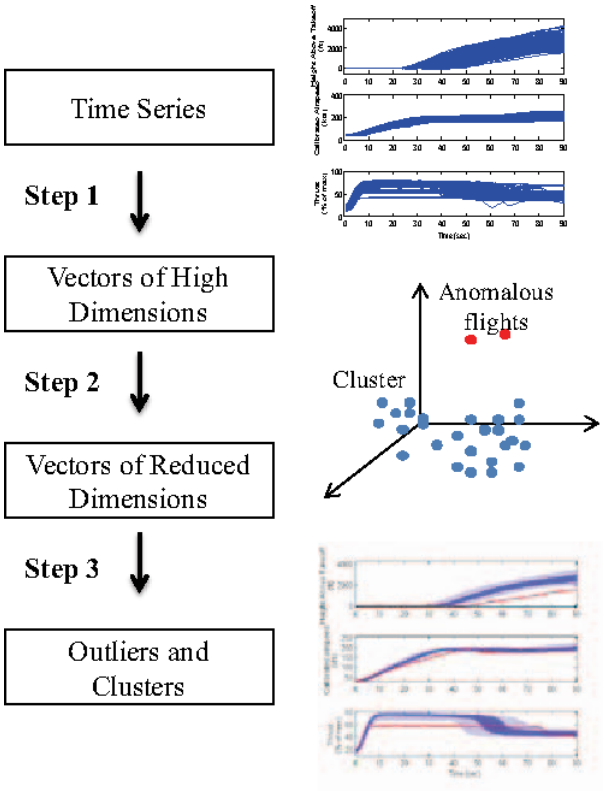


Figure 5.1: ClusterAD methodology [15]

both continuous and discrete, were sampled in segments of 0.5 NM, after which RreliefF, an extended version of Relief, and PCA determined these most relevant features from a data set, reducing it to only 10. The authors assessed the methods on training time (speed), number of parameters and performance indicators such as RMSE. It was found that MLP and Lasso perform best for this particular problem. In addition, both techniques led to a combined third technique called Ensemble. The combination was simply the average of the predictions of the two most feasible options. A case study showed that this in fact improved the results even further. The major findings of the research were the effective use of PCA and RreliefF and that a simple combination of two promising techniques can enhance the outcome. The former shows that also 'ensembled' feature selection techniques, i.e. RreliefF and PCA sequentially, can improve the effectiveness of features.

Similarly, in 2017 Herrema et al. worked on a merger of three methods to compute the expected runway occupancy time. Now Sequentialfs and RreliefF were used for feature selection instead of PCA and RreliefF earlier [34]. A set of 22 parameters was considered, both continuous and discrete. Though eventually 12 parameters were discarded by the feature selection procedure, making the model more robust to fresh data. A data set containing CDG flights was then used for the assessment of the merged methods. Subsequently, a Regression Tree was added to identify precursors since the goal was again to assist controller in their decision-making and trees are obviously intuitive. Herrema et al. announced further work on the implementation of this method and furthermore stressed the flexibility for similar problems. One of the key features of this model was its ability to update in real time, an appreciable aspect enhancing its decision-making capabilities.

A year later a comparable trade-off was performed by Herrema et al. for taxi-out time predictions [44]. Again, different methods were evaluated and the best selected, Regression Tree respectively. The same procedure for feature selection was used and the 10 best out of 42 chosen. These 42 parameters were heterogeneous as before. A case study was conducted with a set of 500,000 CDG flights. The results were verified with other cases researched by fellow researchers and based on that

it became clear only a case at ARN had a marginally higher accuracy. Taking into account the complex operations at CDG this result was deemed a marginal error. This shows that putting the results in perspective could lead to an entirely different interpretation and stresses that for verification and validation conditions are to be carefully reviewed before drawing conclusions on the performance of the algorithm.

Nanduri et al. Nanduri and Sherry developed a Recurrent Neural Network (RNN) for identifying anomalies in flight data [45]. Varying architectures were considered, variations of Long Short Term Memory (LTSM) as well as Gated Recurrent Units (GRU) techniques. Recurrent Neural Networks (RNN) are different from the basic neural networks discussed previously. RNN allows the output of neurons to be fed back and serve as inputs for the other neurons. Hence, the network uses the history as a way to get a grasp on the sequential nature of the data. That is why RNN are specifically useful for this kind of data. LTSM enables to learn long term dependencies in the input. The architecture has a set of recurrently connected structures, the so called memory blocks. Gated Recurrent Units are in fact a variation on LTSM that fuse the forget and input gates into an update gate simplifying the model. These techniques have the capability to manage multivariate sequential, time series data without dimensionality reduction. In addition, the authors address feature selection as a crucial step in both MKAD and ClusterAD and regard this as a restriction in real-time applications. Besides, the algorithms are insensitive to short term anomalies and cannot identify anomalies in latent features either. Latent features are indirect features that affect the direct features. The insensitivity to direct features could be caused by the severe data compression, since nuances, affected by latent features, are lost. The authors of this paper have a remarkably different perspective regarding features since feature selection in their opinion could also be a shortcoming. Data was gathered from the *X-Plane* simulation software, 500 flights in total. Similar to Das et al. [40] and Li et al. [15], only the approach phase was considered for the purpose of comparing the quality of the different methods. That is why the canonical anomalies were introduced according to the earlier work by Das et al. [3]. GRU and LTSM RNN showed to have a much higher recall for exactly the same precision of 1. Since the F_1 score considers both values, the model outperformed MKAD in all three measures.

Janakiraman et al. Janakiraman et al. developed an algorithm that can discover precursors in time series, ADOPT discussed earlier [28]. ADOPT consists of multiple steps such as data preparation, reward and value function modelling and action identification. Figure 5.2 summarises its method.

A case study was conducted on 1,000 flights, of which each time series contained 41 parameters and had an average length of roughly 250 time steps. The results were promising though an important remark was made regarding threshold put on the precursor index. The recall and precision of the method showed notable sensitivity to this parameter, requiring further analysis.

In 2017 Janakiraman et al. published a research that focused on establishing precursors for anomalous drops in airspeed with the use of ADOPT [27]. The authors again stress how prior events can assist and enhance decision-making by being root causes of the event to a certain extent. These root causes could subsequently be fed back to training departments for instance, in coherence with Figure 2.2. This philosophy is obviously extremely in line with the future research. The threshold issue was coped with by learning this parameter from two classes. Additionally, Granger's Causility was used to select features from a flight data set, this eliminated 150 of the 370 heterogeneous variables in the 4,000 available flights. Granger's was chosen for its simplicity and scalability. In addition, an expert was consulted, who added and removed features based on his domain knowledge. Immediate relations to the anomalous drop in air speeds were already discarded, such as air speed, angle of attack etc. as these would not be meaningful precursors due to its physical relation. This consideration is crucial as it proves potentially very accurate precursors could be meaningless. Eventually, the outcome was convenient plot showing how the risk developed over time based

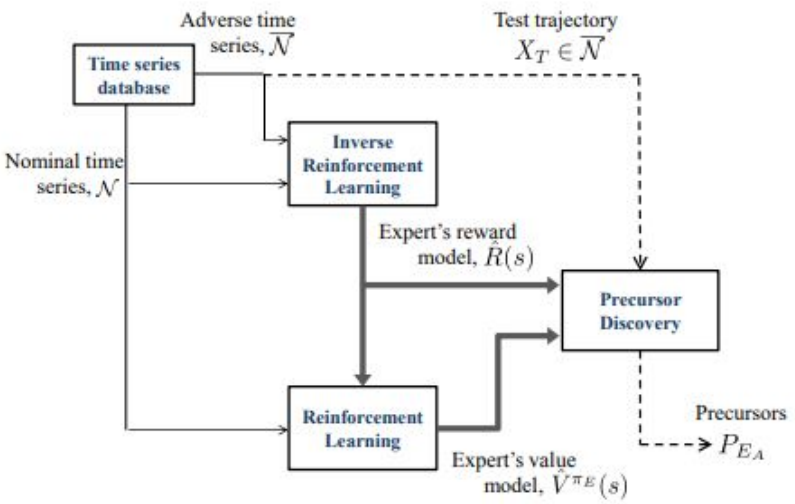


Figure 5.2: ADOPT methodology [28].

on the precursors. The need for feature selection was stressed for reducing the overall complexity. Complexity that was also dealt with for operational purposes by reducing the number of precursors by means of ranking. This eased the interpretation by the consulted expert and could be a potential solution if many features are to be considered. This additional reconsideration on the found precursors once more underpins the importance of the interpretability criterion.

Janakiraman In 2017 Janakiraman proposed once more a precursor mining algorithm that identifies correlated patterns in multidimensional time-series to explain a safety event [46]. The motivation for this study was to identify root causes, as previously, and to reduce the general workload. Additionally, the author argues that existing precursor mining techniques have poor scalability when facing high dimensional time series as well as poor performance in capturing long-term memory, similar to the issue tackled by Nanduri and Sherry [45]. By merging multiple-instance learning (MIL) and deep recurrent neural networks (DRNN) this novel ensemble has two advantages at once. First, MIL is able to model weakly supervised data. Second, DRNN captures long term memory processes and more easily copes with highly dimensional data and large data sets. A case study was conducted on high speed approaches defined as the air speed at a fixed 1,000 ft above airfield. Relevant flight parameters were selected by the researchers by means of expert judgement. These parameters were sampled each 0.25 NM from 25 NM out onward, meaning each flight consisted of 100 time steps. 500 occurrences were reported in a certain time period of which the algorithm learned its precursors. Consequently, a probability timeline was established, showing the likelihood occurrence of the high speed event over time.

Tanguy et al. Tanguy et al. analysed the other obligatory dataset discussed in Section 2.4, safety reports [47]. Two industry problems were tackled, report classification and topic modelling. The former is done in a supervised sense by a SVM, classifying a report to a list of categories based on the textual content. This content is converted to words, stems, character n-grams and stem n-grams, the features. The features became numerical vectors by computing the relative frequency of each unit. Stems showed to produce the best results, whilst words in fact worsened the accuracy. If including text in a learning algorithm, stems seem to have the best potential. Afterwards, the topic modelling was set up by first of all assuming a report is a bag of words expressing topics of varying importance according to varying distributions i.e. a multivariate data set. In other words,

the distributions document-topic and topic-words were to be retrieved. The need for this analysis originated from trend monitoring purposes, as experts wondered whether there exist similar occurrences worth investigating. The similarity was assessed by a lexical overlap. After stemming and tokenising the 'bags', the inner product of two vectors, containing the weighted terms, defined the similarity. The same learning algorithm as before was used, only considering the proven valuable stems. Active learning was applied with an expert in the loop where the expert improved the performance of the algorithm iteratively by classifying the 20 instances closest to the decision boundary. This step was repeated each iteration until satisfactory results were obtained. The results did enhance by the human interaction and illustrated the relevance of such iterative methods with an expert in the loop.

5.2 Conclusive Remarks

This sections summarises the common observations as seen in the above papers. The aim hereby is to highlight the most important considerations of fellow researchers which are expected to contribute to the future research.

Throughout the discussion of these papers the method selection criteria as established in Section 4.7 are repeatedly mentioned as being decisive in these applications. Table 5.1 proofs this statement by summarising the reason(s) for the fundamental method selection per research. Hence, this validates the criteria even more than already done by the comparison with Budalakoti et al. [33]. That is why these will definitely be sufficient to find the appropriate method at hand with this insight.

Table 5.1: Overview of applications in Safety Data Analysis.

Author	**Year**	**Method(s)**	**Reason(s)**
Chidester	2003	Unsupervised clustering	Nature of data (multivariate)
Das et al.	2010	Multi Kernel	Nature of data (heterogeneous)
Das et al.	2011	Multi Kernel	Nature of data (heterogeneous)
Budalakoti et al.	2009	Unsupervised clustering	Scalability
Li et al.	2011	Unsupervised clustering	Limited number of hyperparameters (complexity)
Li et al.	2015	Unsupervised clustering	Limited number of hyperparameters (complexity)
Herrema et al.	2016	MLP and Lasso	Performance trade-off
Herrema et al.	2017	Merged Regression Tree	Combined performance assessment
Herrema et al.	2018	Regression Tree	Performance trade-off
Nanduri et al.	2016	RNN based on GRU and LTSM units	Nature of data (multivariate, sequential, highly dimensional), latent features, no feature selection
Janakiraman et al.	2016	Reinforced Learning	Use of 'soft' information
Janakiraman et al.	2017	Reinforced Learning	Use of 'soft' information, improved threshold
Tanguy et al.	2016	Supervised SVM	Performance trade-off, nature of data (text)
Janakiraman	2017	MIL & DRNN	Weakly supervised, nature of data (highly dimensional), scalability, long term relations

It is furthermore noticeable that the feature selection techniques applied in these researches are not overly complex, PCA for instance is still widely used. Experts are frequently consulted for feature selection as well, to capture knowledge by assessing the relevance of the mathematically retrieved features or to pick features from scratch. These experts are also involved in training and validation, illustrating their understanding is very much appreciated. Besides, interpretability is secured at the same time if experts are involved in an early stage of the research, such as feature selection. This is indeed governed in the procedure laid down in Section 4.9.3.

Almost all researchers face a massive amount of data that simply cannot be processed due to the lack of computational power. Consequently, the overall data is drastically compressed by feature selection as well as scope reduction. The latter is done in a fairly inconsistent manner, flight phase or single runway, depending on the events considered or the best practises by the researchers. Table 5.3 shows the estimated data size per research. These numbers are perfect benchmark figure for the future study and could indicate of the number feature times instances is of the right order of magnitude.

Table 5.2: Feature selection per research.

Author	**Year**	**Method(s)**
Chidester	2003	PCA
Das et al.	2010	Expert
Das et al.	2011	Expert
Budalakoti et al.	2009	N/A
Li et al.	2011	PCA
Li et al.	2015	PCA
Herrema et al.	2016	RreliefF/PCA
Herrema et al.	2017	RreliefF/Sequentialfs
Herrema et al.	2018	RreliefF/Sequentialfs
Nanduri et al.	2016	N/A
Janakiraman et al.	2016	Granger's Causality
Janakiraman et al.	2017	Granger's Causility
Tanguy et al.	2016	N/A
Janakiraman	2017	Expert

Table 5.3: Data size per research.

Author	**Year**	**Data size**
Chidester	2003	Unknown
Das et al.	2010	10^5
Das et al.	2011	10^6
Budalakoti et al.	2009	10^6
Li et al.	2011	10^6
Li et al.	2015	10^7
Herrema et al.	2016	10^7
Herrema et al.	2017	10^7
Herrema et al.	2018	10^6
Nanduri et al.	2016	10^6
Janakiraman et al.	2016	10^7
Janakiraman et al.	2017	10^6
Tanguy et al.	2016	10^6
Janakiraman	2017	10^6

5.3 Research Gap

After carefully reviewing all the papers above, the gap in the existing literature can be derived. It is observed that earlier researches primarily focus on datasets generated by a single process and investigate events or anomalies within the parameter domain of that process i.e. the direct dataset. In other words, the focus lies on the direct dataset that contains the necessary information to evaluate the event or the anomaly itself. As an example, flight anomalies are investigated with the parameters retrieved from flight data, examples are the researches conducted by Das et al. and Li et al. [3, 15]. Every now and then features originating from other processes are added, but these indirect data sets are not yet considered structurally. Airlines have more data available thanks to the numerous data generating processes, such as the pre-flight planning process shown by Figure 5.3. This schedule generation results in data such as legs, pairings, origins, destinations etc., to which the fleet assignment assigns an aircraft per member. Maintenance routing reroutes the schedule to ensure maintenance tasks can be performed before a crew is assigned. This crew also has certain characteristics such as age and maturity which are not taken into account but might still be relevant features. ASR's could contain relevant text or descriptors, note an ASR is more than a flat piece of text. In short, these indirectly related datasets could encompass relevant features that affect the outcome of a flight. This however demands a strong assumption, or hypothesis, regarding the influence of these processes on the flight safety. These hypotheses are discussed subsequently in Chapter 6. This assumption is not unorthodox as it is similar, to a certain extent, to the assumption made by Das et al. on how button sequences affect the continuous data stream [3]. Das et al. already suggested to include textual data retrieved from air safety reports, a process explained in Section 2.4, suggesting these interactions between the different processes do exist [40]. This is basically the gap in the literature and defines the aim of the future study, namely identifying relevant precursors in indirect data sets generated by common, non-safety related operational data generating processes. The commonality is important to keep the method generic, a characteristic that more easily allows a broader application. This is also in line with the philosophy of Safety-II, discussed in Section 2.2. Considering additional processes potentially increases the size, but definitely increases dimensionality and heterogeneity of the data set and that is obviously the one of the biggest challenges recalling the nature of the data is crucial to the method selection, as addressed in Section 4.7. In addition, this research will tackle the struggle pointed out by Li et al., considering multiple data streams in a single method [42].

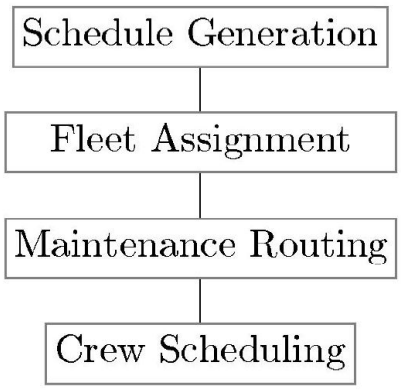


Figure 5.3: Airline planning [48].

6 Research Proposal

This chapter formally introduces the research proposal as it is derived from the research gap identified in the previous chapter. The following formalities are presented accordingly: the research problem, objectives, questions, methodology and planning.

6.1 Research Problem

As addressed in the introduction of this literature review, airlines are confronted with large data sets which are not yet fully mined for the purpose of proactive safety management. Ideally, in the light of the current view on safety management, this mining would result in a set of precursors that could enhance the safety performance proactively. Current academic literature tends to focus on the direct dataset and does not yet consider the indirect datasets generated by other processes such as crew scheduling, flight planning etc. Hence, the industry problem and the research gap are in fact well aligned. The difficulty in utilising these unexplored datasets is in its dimensionality and heterogeneity, two criteria that have a distinct impact on the applicability of methods as discussed in Section 4.7. Besides, the physical or direct relation is no longer existent which requires a very strong hypothesis or suspicion regarding the relations between these datasets.

6.2 Research Objectives

The research objective is important in explaining what can and cannot be expected from the research and to give a general idea of the activities [49]. As the research is undertaken in collaboration with an industry partner, two objectives are defined, an academic and an industry objective. It is important for any researcher in such a situation to continuously evaluate these objectives to live up to the expectations of the industry partner and simultaneously maintain the academic standard. The academic objective is defined as follows:

> *To identify a set of safety precursors from operational datasets by applying a data mining methodology*

The industry objective is somewhat more general, it serves the greater purpose and describes the industry's major interest, as seen below.

> *To proactively manage safety by data-driven decision-making based on a precursor identification tool*

From the statements above two clear distinctions can be derived. First, from an academic perspective a model will be developed whilst from an industry perspective it is a tool. The implementation and user-friendliness of the model are for instance two valuable aspects for the industry that are not relevant from a scholar point of view. Contrarily, the model will attempt to represent the underlying data dynamics between the event and the datasets. Second, the focus of the objectives differs between the academic and industry objective, namely data analysis and safety management respectively. The duty of a researcher is to find and manage the proper balance between the two at any given time during the research.

6.3 Research Questions

Next, the research questions are posed. Research questions are usually subdivided over a multitude of sub-questions. These sub-questions are derived from the main research question. The main question is depicted below.

What multi-process based precursors set is significant in predicting the occurrence of a particular flight safety event?

The extracted sub-questions are listed below.

1. What is the required data?
   (a) What are the common operational data generating processes in airlines?
   (b) What is the structure and size of the generated datasets?
   (c) What is the type of the data in these datasets?
   (d) Are these datasets currently monitored for the purpose of safety and how?
   (e) Is the dataset considered to be clean?
2. What is currently an industry-wide safety concern i.e. event with particular interest or focus?
   (a) What are the characteristics of this event?
   (b) What data is required to evaluate this event?
   (c) Is this event currently monitored and if so, how?
   (d) What are suspected (root) causes of this event?
3. What is the most suitable data analysis method for the problem at hand?
   (a) What are the currently used methods for data analysis?
   (b) What are the criteria for the data analysis method selection?
   (c) What is the best analysis method according to a fair trade-off based on these criteria?
4. What features are to be fed to the data analysis method?
   (a) What are the currently used methods for the feature selection?
   (b) What are the criteria for choosing the feature selection method?
   (c) What is the best feature selection method according to a fair trade-off based on these criteria?
   (d) Do the features need operations or modifications to become (more) relevant or usable?
   (e) Do the features align with the suspected (root) causes?
5. What is the quality of the obtained results?
   (a) What methods and measures are used to assess the quality of these kind of models?
   (b) What is the sensitivity of the control parameters with respect to the output?
   (c) What is a suitable verification method?
   (d) What is the operational feasibility of the results i.e. the resilience of the model for considering other concerns?
   (e) To what extent do the features affect the performance and results of the model?

The second and third question seem to have quite some overlap. However, recall Figure 4.1 formulates multiple independent steps for the data processing. First, the required data is to be chosen and preprocessed, before the feature selection takes place. As both steps have different goals the questions are separated accordingly.

6.4 Research Methodology

This section presents the methodology, the theoretical basis and the underlying hypotheses of this practice-oriented research.

Chapman et al. laid down a procedural approach to a data mining problem. This convenient procedure consists of multiple phases and will the general guideline thorough this research [50]. Figure 6.1 shows the different phases and how these are related. Below a short description of each phase is discussed.

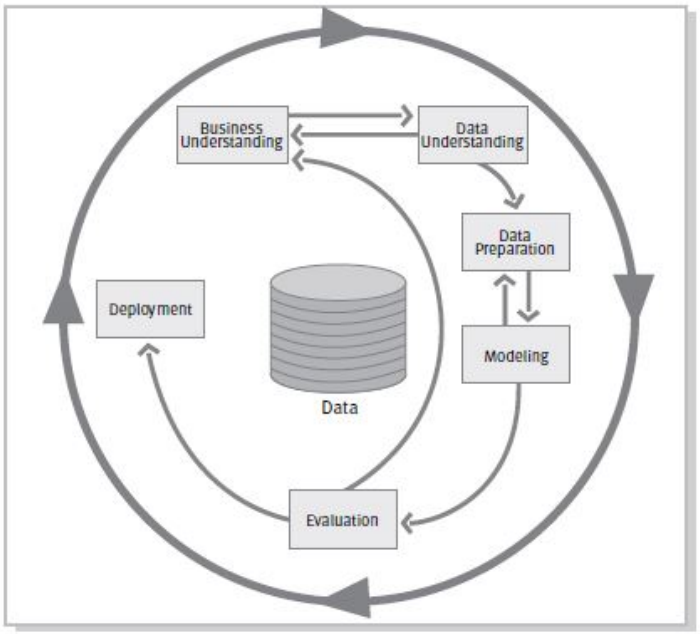


Figure 6.1: Data processing steps [50].

Business/Academic Understanding The first phase is about gaining a thorough understanding of the problem and converting this problem into a data mining problem. The former is the goal of this review as the previous chapter identified the research gap. Similarly, the case study requires the understanding from a business perspective. This will be done during a concern analysis.

Data Understanding The second phase starts off with collecting the data and some steps that try to establish familiarity with the dataset. The quality of the data is assessed and the first insights gained. A description report and exploration report will be the results of this phase.

Data Preparation The preparation phase covers all activities required to build the final dataset, i.e. the set data is ready to be fed to the model. Transformations and cleansing take place in this phase. These transformations are part of the feature selection procedure.

Modelling Modelling techniques are selected and applied, and the hyperparameters are calibrated to obtain highest performance. Considering several techniques for the same problem is a commonly done, in this research these will be traded off based on the method selection criteria. As proven earlier, modelling techniques might have specific requirements on the nature of data. Hence, going back and forth between data preparation phase and this phase is often necessary.

Evaluation Before the finalising the model it is important to evaluate and review the executed phases so far. This is the verification and validation of the model that ensure the quality of the model. At the end of this step, a decision on the usefulness of the results is to be made.

Deployment The aim of a model is to gain knowledge on relations within the data. This knowledge will need to be organised and presented such that a customer can interpret it, the interpretability criterion. This

With this general framework in place, the research at hand is formulated accordingly. First and foremost, the hypotheses are specified. The upcoming study is based on two hypotheses, shown below. These formulations came to be based on the early discussions with the industry and from the research gap established previously. The first hypothesis is fairly straightforward and can even be proven by fundamental statistics with for instance correlations. It is however important to pose this rationale since the second hypothesis is to a large extent an elaboration on the first. It is unlikely to proof the second without satisfying the first. The added feature of the latter hypothesis is the interaction, i.e. dynamics of the system, that can actually establishing good predictions by being exposed.

> *There exist statistically significant relations between certain safety events and indirect operational data (features)*
>
> *The occurrence of safety events can be predicted by learning the dynamics of these relations from historical data*

A research framework is a schematic representation of the content of the research. Figure 6.2 shows the research framework of this proposal. As can be seen from this figure, the research is divided in four phases, *Preliminary Study*, *Model Development and Implementation*, *Verification&Validation and Testing* and *Wrap-up*. On the basis of this framework the subsequent paragraphs will elaborate on the research methodology. Note the first column primarily the literature review and its content which is not elaborately treated here.

The first step in the methodology is to identify the required data, as Figure 4.1 suggested. This was in fact broadly done by establishing the research gap in the previous chapter. This was however only a general description of the data sets whilst a more careful description is yet necessary for further processing. Recall the nature of data is crucial in method selection and that is why is a first principal step. One of the subsequent steps is the preprocessing, which is done according to the procedure laid down in Section 4.9. These steps are depicted in the research framework as *Dataset and Feature Analysis*. Concurrently, the *Concern Analysis* takes place, answering the second research question. This simultaneously affects the relevance of certain datasets and features based on the experts understanding. This knowledge is going to be gathered during interviews. A set of features is the final result if both of these steps are completed. Subsequently, the method trade-off and analysis is conducted aiming to find the best method for the problem at hand. The feature set is here of major importance as it determines the applicability of the method based on the requirements. Recall the first three criteria are depending on the nature of the data i.e. features and that these function as requirements. The best method is then assessed by the remaining criteria set, deciding upon the feasibility. This trade-off is done according to Table 6.1 where the methods discussed in Chapter 4 are depicted. The scale works as follows: 1 for poor and 5 for best performing respectively. As illustration, a score of 1 on complexity means a very complex algorithm. Note this is an equal weight trade-off were none of the criteria is ranked. In this setting Neural Nets and Decision Trees seem to have the most all round performance. If ranking of criteria is applied this could change the outcome and therefore this consideration is also part of the trade-off analysis. Experts might again be consulted to in particularly review the feasibility criteria. In addition, elaborations

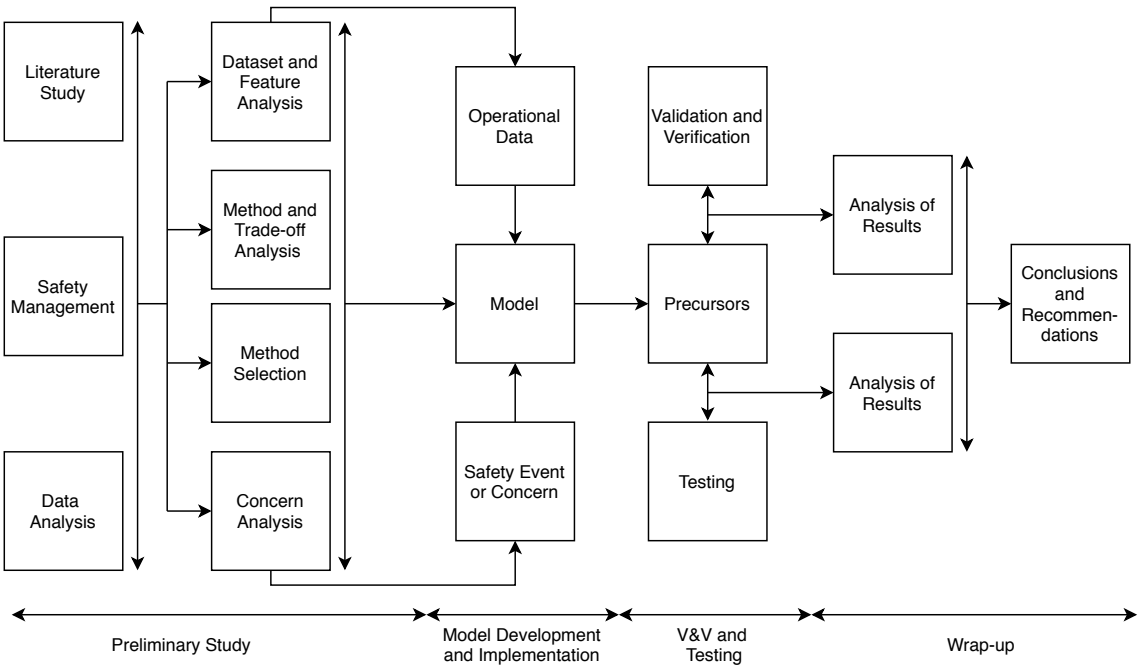


Figure 6.2: General framework of the research.

of the current methods are researched if the synthesis of this step is near, before the final method is decided upon. This step is seen as *Method Selection* in the diagram. Note that although this seems very decisive, features for instance could still be added and removed to either enhance the performance, thanks to new insights, or to assess the sensitivity of the model. If results are unsatisfactory this entire procedure could even be repeated or the second method in line could be considered. The latter is regularly observed by fellow researchers, Herrema et al. for example made performance the sole criterion [34]. If the trade-off is not conclusive either, as for the equal weight case seen below, the performance could be the obvious though deciding criterion. Despite this nuance, the feature set retrieved by the methodology should be a firm start towards the problem to ensure continuation of the research.

Table 6.1: Method trade-off.

	Nature of data			**Feasibility**		
	Scalability	Heterogeneity	Dimensionality	Complexity	Interpretability	Robustness
Domain						
SVM	1	1	5	3	2	3
Kernels	2	4	4	2	1	4
Reconstruction						
NN	5	5	5	2	1	5
Distance						
K-Means	1	1	2	5	3	2
k-NN	2	1	2	5	3	1
Rule						
Decision Tree	3	4	5	4	5	3
Ensembled						
X	1	4	4	1	2	3

As soon as the synthesis of the second column is nearly reached, the building the model is initiated. If the model is almost finalised the testing, validation and verification steps are executed. At it smaller scale is done continuously though a thorough testing and validation is undertaken to assess the quality. The testing and validation steps are often confused but do differ fundamentally. Testing is done for tweaking the final control parameters whilst validation is there to assess the quality of the final model. The experiment is going to be a case study. Hence, this step aims at in-depth investigation of a particular event, the result of the *Concern Analysis*. Contrarily, the model is ideally resilient, so it is capable of investigating other unique cases. This is important for the operational feasibility of the model. With the results in place, the project is finalised with a report discussing the findings and documenting the steps as well as drawing conclusions and making recommendations for further studies, the final step seen in the framework.

6.5 Research Planning

The project planning is conveniently summarised by means of a Gantt chart and the most relevant features of this chart are discussed in this section.

The Gantt chart is shown in Figure 6.3, at the end of this document. The milestones are indicated with diamonds, with the particular milestone date on the right. These important dates are also listed in Table 6.2. The work packages and tasks are depicted by the black and white bars respectively. The duration of each work package is aside of the bar. In this preliminary stage, a time window in terms of weeks is deemed sufficiently accurate. The work packages clearly show a strong link with the research framework, illustrated by Figure 6.2. Note the relations between certain tasks are indicated with arrows. Be aware the preliminary study has already taken place in this chart. The research questions are translated to particular tasks, except for those answered during the literature study. Meetings are illustrated by the blue, vertical dashed lines. The intention is to meet at least once per two weeks, and to intensify this structure near milestones and the end of work packages. Finally, it is worth mentioning that a one week buffer is accounted for, the week before the Christmas holidays. These holidays are depicted by the dashed grey lines.

Table 6.2: Milestones and dates.

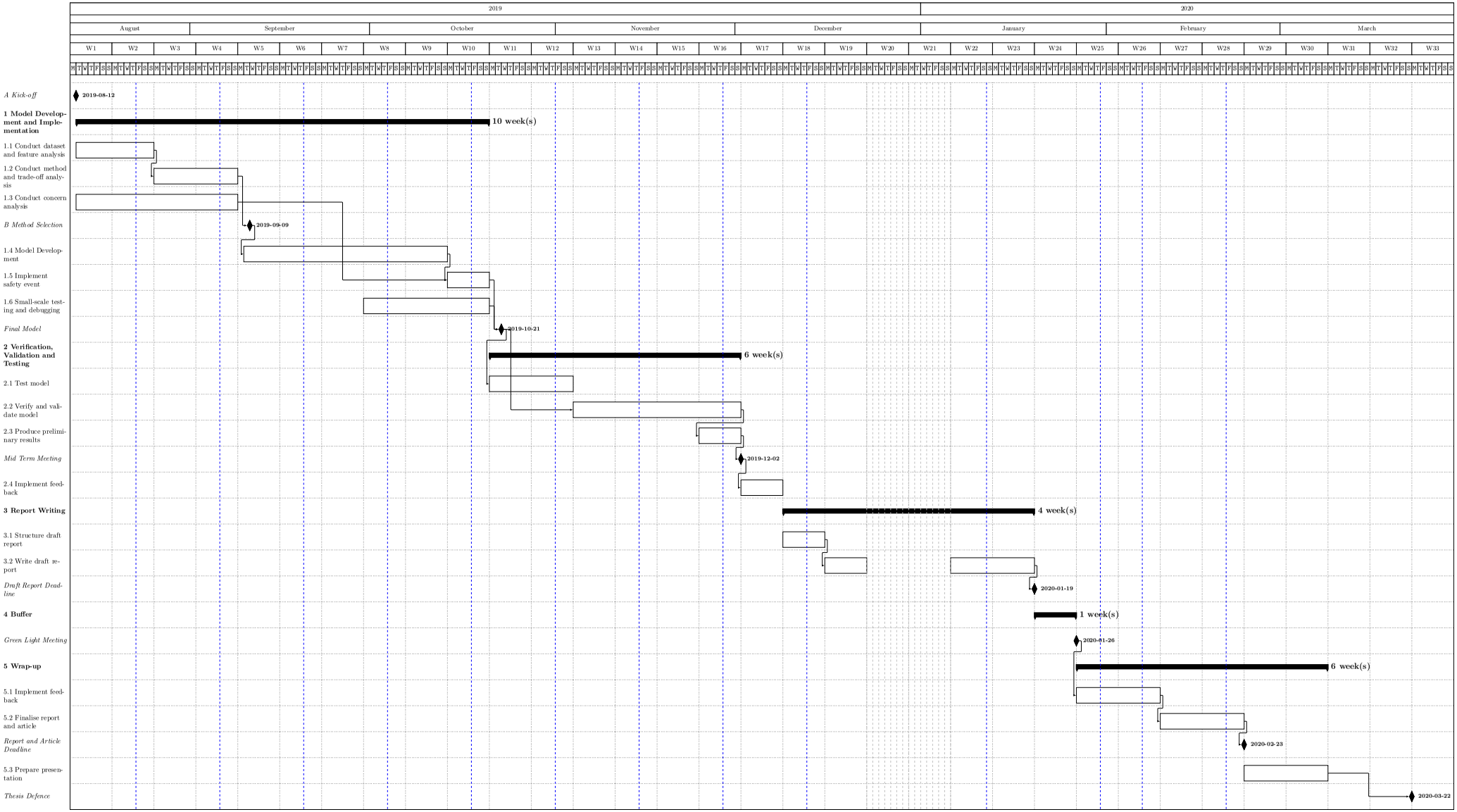
Milestone	**Date**
Kick-off Meeting	2019-07-25
Method Selection	2019-09-09
Final Model	2019-10-21
Midterm Meeting	2019-11-05
Deadline Draft Report	2020-01-12
Green Light Meeting	2020-01-20
Deadline Final Report and Article	2020-02-17
Thesis Defence and Graduation	2020-02-30

Throughout the research deliverables have to be handed in as part of the progress. Some deliverables have an obvious link to the meetings but others do not and that is why it is important to specify these. Table 6.3 summarises all the important deliverables, the deadline date and the people these documents belong to.

Table 6.3: Deliverables.

Deliverable	**Date**	To be delivered/presented to:
Literature Review	2019-08-13	Supervisors
Midterm Presentation	2019-11-05	Supervisors and others[1]
Draft Report	2020-01-12	Supervisors
Final Report and Article	2020-02-17	Supervisors
Thesis Defence	2020-02-30	Supervisors, thesis committee and others[2]

[1] Open to TU Delft researchers/staff
[2] Open to TU Delft researchers/staff, friends, family and general public


2019
2020
August
September
October
November
December
January
February
March
W1 W2 W3 W4 W5 W6 W7 W8 W9 W10 W11 W12 W13 W14 W15 W16 W17 W18 W19 W20 W21 W22 W23 W24 W25 W26 W27 W28 W29 W30 W31 W32 W33
A Kick-off
2019-08-12
1 Model Development and Implementation
10 week(s)
1.1 Conduct dataset and feature analysis
1.2 Conduct method and trade-off analysis
1.3 Conduct concern analysis
B Method Selection
2019-09-09
1.4 Model Development
1.5 Implement safety event
1.6 Small-scale testing and debugging
Final Model
2019-10-21
2 Verification, Validation and Testing
6 week(s)
2.1 Test model
2.2 Verify and validate model
2.3 Produce preliminary resilts
Mid Term Meeting
2019-12-02
2.4 Implement feedback
3 Report Writing
4 week(s)
3.1 Structure draft report
3.2 Write draft report
Draft Report Deadline
2020-01-19
4 Buffer
1 week(s)
Green Light Meeting
2020-01-26
5 Wrap-up
6 week(s)
5.1 Implement feedback
5.2 Finalise report and article
Report and Article Deadline
2020-02-23
5.3 Prepare presentation
Thesis Defence
2020-03-22

Bibliography

[1] European Aviation Safety Agency. *Annual Safety Review 2018*. 2018. www.skybrary.aero, Accessed: 13-06-2019.

[2] Federal Aviation Administration. *FAA Aerospace Forecast*. 2018. www.faa.gov, Accessed: 17-06-2019.

[3] S. Das, B.L. Matthews, A. Srivastava, and N. Oza. "Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study". In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, Washington, DC, USA*. 2010, pp. 47–56.

[4] E. Hollnagel, L. Jörg, T. Licu, and S. Shorrock. "From Safety-I to Safety-II: A White Paper (Eurocontrol)". In: *The resilient health care net: published simultaneously by the University of Southern Denmark, University of Florida, USA, and Macquarie University, Australia* (2013).

[5] S. Maire and C. Spafford. "The Data Science Revolution That's Transforming Aviation". In: *Forbes Magazine* (2017). www.forbes.com, Accessed: 25-06-2019.

[6] R.L. Helmreich. "On error management: Lessons from aviation". In: *British Medical Journal* 320.7237 (2000), pp. 781–785.

[7] B.J.M. Ale, L.J. Bellamy, R.M. Cooke, L.H.J. Goossens, A.L.C. Roelen, A.R. Hale, and E. Smith. "Towards a causal model for air transport safety: an ongoing research project". In: *Safety Science* 44.8 (2006), pp. 657–673.

[8] International Civil Aviation Organisation. *Safety Management Manual (SMM)*. 2013. www.icao.int, Accessed: 31-07-2019.

[9] S. Dekker. *Safety differently: human factors for a new era*. CRC Press, 2014.

[10] E. Hollnagel. *Safety-I and Safety-II*. CRC Press, 2014.

[11] International Civil Aviation Organisation. *Safety Management Systems (SMS) and Cabin Safety*. 2011. www.icao.int, Accessed: 31-07-2019.

[12] European Aviation Safety Agency. *Acceptable Means of Compliance (AMC) and Guidance Material (GM) to Part-ORO*. 2014. www.easa.europa.eu, Accessed: 04-07-2019.

[13] G.W.H. Van Es. "Advanced flight data analysis". In: *Proceedings of the 14th European Aviation Safety Seminar, Budapest, Hungary*. 2002.

[14] L. Höhndorf, J. Sembiring, and F. Holzapfel. "Copulas applied to Flight Data Analysis". In: *Probabilistic Safety Assessment and Management PSAM* 12 (2014).

[15] L. Li, M. Gariel, R.J. Hansman, and R. Palacios. "Anomaly detection in onboard-recorded flight data using cluster analysis". In: *Proceedings of the 2011 IEEE/AIAA 30th Digital Avionics Systems Conference, Seattle, WA, USA*. 2011.

[16] M.G. Karlaftis and E.I. Vlahogianni. "Statistical methods versus neural networks in transportation research: Differences, similarities and some insights". In: *Transportation Research Part C: Emerging Technologies* 19.3 (2011), pp. 387–399.

[17] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An introduction to statistical learning*. Vol. 112. Springer, 2013.

[18] M.A.F. Pimentel, D.A. Clifton, L. Clifton, and L. Tarassenko. "A review of novelty detection". In: *Signal Processing* 99 (2014), pp. 215–249.

[19] C.M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[20] L. Drees, J. Siegel, P. Koppitz, and F. Holzapfel. "Quantifying probabilities of exceeding the maximum Mach number in cruise flight using operational flight data". In: *European Safety and Reliability Conference (ESREL)*. 2017.

[21] C. Wang, L. Drees, N. Gissibl, L. Höhndorf, J. Sembiring, and F. Holzapfel. "Quantification of incident probabilities using physical and statistical approaches". In: *6th International Conference on Research in Air Transportation. Istanbul, Turkey*. 2014.

[22] L. Drees and F. Holzapfel. "Predicting the occurrence of incidents based on flight operation data". In: *AIAA Modeling and Simulation Technologies Conference*. 2011.

[23] B.J.M. Ale et al. "Further development of a Causal model for Air Transport Safety (CATS): Building the mathematical heart". In: *Reliability Engineering & System Safety* 94.9 (2009), pp. 1433–1441.

[24] M. Markou and S. Singh. "Novelty detection: a review—part 1: statistical approaches". In: *Signal processing* 83.12 (2003), pp. 2481–2497.

[25] V. Chandola, A. Banerjee, and V. Kumar. "Anomaly detection: A survey". In: *ACM computing surveys (CSUR)* 41.3 (2009), p. 15.

[26] S.B. Kotsiantis, I. Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques". In: *Emerging artificial intelligence applications in computer engineering* 160 (2007), pp. 3–24.

[27] V.M. Janakiraman, B.L. Matthews, and N. Oza. "Finding precursors to anomalous drop in airspeed during a flight's takeoff". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada*. 2017, pp. 1843–1852.

[28] V.M. Janakiraman, B.L. Matthews, and N. Oza. "Discovery of Precursors to Adverse Events using Time Series Data". In: *Proceedings of the 2016 SIAM International Conference on Data Mining*. 2016, pp. 639–647.

[29] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[30] M. Markou and S. Singh. "Novelty detection: a review - part 2: neural network based approaches". In: *Signal Processing* 83.12 (2003), pp. 2499–2521.

[31] C.M. Bishop. "Novelty detection and neural network validation". In: *IEE Proceedings: Vision, Image and Signal Processing* 141.4 (1994), pp. 217–222.

[32] A. Dharia and H. Adeli. "Neural network model for rapid forecasting of freeway link travel time". In: *Engineering Applications of Artificial Intelligence* 16.7 (2003), pp. 607–613.

[33] S. Budalakoti, A.N. Srivastava, and M.E. Otey. "Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety". In: *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews* 39 (2009), pp. 101–113.

[34] F.F. Herrema, V. Treve, B. Desart, R. Curran, and H.G. Visser. "A novel machine learning model to predict abnormal Runway Occupancy Times and observe related precursors". In: *Proceedings of the 12th USA/Europe Air Traffic Management Research and Development Seminar, Seattle, WA, USA*. 2017.

[35] I. Guyon and A. Elisseeff. "An introduction to variable and feature selection". In: *Journal of Machine Learning Research* 3 (2003), pp. 1157–1182.

[36] K. Kira and L.A. Rendell. "The Feature Selection Problem: Traditional Methods and a New Algorithm." In: *Proceedings of the 10th National Conference on Artificial Intelligence, San Jose, CA, USA*. 1992, pp. 129–134.

[37] L.H.J. Goossens, R.M. Cooke, A.R. Hale, and L.J. Rodić-Wiersma. “Fifteen years of expert judgement at TU Delft”. In: *Safety Science* 46.2 (2008), pp. 234–244.

[38] T.R. Chidester. “Understanding normal and atypical operations through analysis of flight data”. In: *Proceedings of the 12th International Symposium on Aviation Psychology*. 2003, pp. 239–242.

[39] B.G. Amidan and T.A. Ferryman. *APMS SVD methodology and implementation*. Tech. rep. Pacific Northwest National Lab., Richland, WA (US), 2000.

[40] S. Das, B.L. Matthews, and R. Lawrence. “Fleet level anomaly detection of aviation safety data”. In: *2011 IEEE Conference on Prognostics and Health Management*. IEEE, 2011, pp. 1–10.

[41] M. Ester, H.P. Kriegel, J. Sander, X. Xu, et al. “A density-based algorithm for discovering clusters in large spatial databases with noise.” In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, OR*. Vol. 96. 34. 1996, pp. 226–231.

[42] L. Li, S. Das, R.J. Hansman, R. Palacios, and A. Srivastava. “Analysis of Flight Data Using Clustering Techniques for Detecting Abnormal Operations”. In: *Journal of Aerospace Information Systems* 12 (2015), pp. 1–12.

[43] F.F. Herrema, V. Treve, R. Curran, and H.G. Visser. “Evaluation of feasible machine learning techniques for predicting the time to fly and aircraft speed profile on final approach”. In: *International Conference on Research in Air Transportation*. Vol. 8. Delft University of Technology, 2016, pp. 4–8.

[44] F.F. Herrema, R. Curran, H.G. Visser, D. Huet, and R. Lacote. *Taxi-Out Time Prediction Model at Charles de Gaulle Airport*. Vol. 15. 2018, pp. 1–11.

[45] A. Nanduri and L. Sherry. “Anomaly detection in aircraft data using Recurrent Neural Networks (RNN)”. In: *2016 Integrated Communications Navigation and Surveillance (ICNS)*. 2016, pp. 5C2–1.

[46] V.M. Janakiraman. “Explaining Aviation Safety Incidents Using Deep Temporal Multiple Instance Learning”. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, United Kingdom*. ACM, 2017, pp. 406–415.

[47] L. Tanguy, N. Tulechki, A. Urieli, E. Hermann, and C. Raynal. “Natural language processing for aviation safety reports: from classification to interactive analysis”. In: *Computers in Industry* 78 (2016), pp. 80–95.

[48] C. Barnhart, A.M. Cohn, E. Johnson, D. Klabjan, G. Nemhauser, and P.H. Vance. “Airline Crew Scheduling”. In: 2003, pp. 517–560.

[49] P. Verschuren, H. Doorewaard, and M. Mellion. *Designing a research project*. Vol. 2. Eleven International Publishing The Hague, 2010.

[50] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth. *CRISP-DM 1.0*. 1999. www.kddlab.com, Accessed: 15-08-2019.

Part III

Appendices

A

UNSTABLE APPROACH DEFINITION

The case study for this research, unstable approaches, was chosen because of a widespread concern in the aviation industry. According to *IATA* an unstable approach is defined as *any approach that does not meet the stabilised approach criteria defined by the operator in its SOPs* [1]. According to the guidelines of the operator under investigation any flight should be stabilised at 500 ft in Visual Meteorological Conditions (VMC) or at 1,000 ft in Instrument Meteorological Conditions (IMC). If these procedural criteria are not met at the respective altitudes, a go-around should be initiated. However, if this manoeuvre is not undertaken, an approach is deemed unstable.

STANDARD OPERATING PROCEDURES

According to the standard operating procedures of the operator under investigation the following criteria have to be met for an approach to be considered stable:

- The airplane is on the correct flight path.
- Only small changes in heading/pitch are required to maintain the correct flight path.
- The airplane approach speed is V_{REF} + wind speed correction, not exceeding V_{REF} + 20 kts and not less than V_{REF}.
- The aircraft is in the correct landing configuration.
- Sink rate is not greater than 1,000fpm. If an approach requires a sink rate greater than 1,000 fpm, conduct a special briefing.
- Power setting is appropriate for the airplane configuration.
- All briefings and checklists have been conducted.

The correctness of the flown flight path is determined by the applicable glide slope, which is a 3 ° path for the majority of airports. The reference speed V_{REF} is derived from the aircraft's operating manual and accounts for a margin with respect to the stall speed. Given the glide path and the corrected reference speed, an aircraft could, by definition, not surpass a sink rate of 1,000 fpm. However, if an aircraft deviates from this predefined path it should still remain within this limit. The landing configuration is correct if landing flaps and gear are set. The power setting is deemed appropriate if non-idle, as this allows the engine to spool up within reasonable time in case of a go-around. The latter criterion is the sole criterion that cannot be monitored by data and was therefore outside the scope of this research.

VISUAL METEOROLOGICAL CONDITIONS CRITERIA

The criteria for VMC are listed in Table A.1. As an unstable approach by definition occurs below 3,000 ft AMSL only the final row of this table is relevant to this research. This table shows the applicability of these conditions is dependent of the airspace class, the horizontal visibility and the distance from the clouds. The airspace

definitions are depicted by Table A.2. According to these class specifications the airline under investigation solely operates in class C or D during approach. Hence, the emphasised criteria in Table A.1 are applicable to this research. The visibility and distance to clouds were derived from METAR reports of which an example is seen below. The visibility (four digit numeric) and distance to clouds (six digit mixed) are depicted in bold. The visibility is readily provided in m and requires no further interpretation, in contrast with cloud height. The cloud height is defined by an intensity, such as FEW, and altitude 024 (2,400ft). The cloud intensity is divided over octants based on the actual sky coverage. The four most intense octants have sufficient coverage to limit the view of an pilot. and therefore only those octants were considered in this study.

METAR EHAM 191125Z 27022KT **9999 FEW024** 07/02 Q1018 NOSIG

Table A.1: VMC criteria according to ICAO [2].

Altitude band	**Airspace**	**Visibility**	**Distance from clouds**
At and above 10,000 ft AMSL	A to G	> 8km	> 1,500m horizontally > 300m vertically
Below 10,000 ft AMSL or 1,000 ft above terrain	A to G	> 5km	> 1,500m horizontally > 300m vertically
At and below 3,000 ft AMSL or 1,000 ft above terrain	**A to E**	**> 5km**	**> 1,500m horizontally** **> 300m vertically**
	F and G	> 5km	Clear of cloud and surface in sight

Table A.2: Airspace definitions [3].

Class	*Type of Flight*	*Separation Provided*	*Service Provided*	*Speed Limitation*	*Radio Communication Requirement*	*ATC Clearance*
A	IFR only	All A/C	ATC service	N/A	Continuous two-way	Yes
B	IFR	All A/C	ATC service	N/A	Continuous two-way	Yes
	VFR	All A/C	ATC service	N/A	Continuous two-way	Yes
C	IFR	IFR from IFR IFR from VFR	ATC service	N/A	Continuous two-way	Yes
	VFR	VFR from IFR	ATC for separation from IFR VFR/VFR traffic information	250 kts below 10,000 ft AMSL	Continuous two-way	Yes
D	IFR	IFR from IFR	ATC service, traffic information VFR	250 kts below 10,000 ft AMSL	Continuous two-way	Yes
	VFR	No	IFR/VFR and VFR/VFR traffic information	250 kts below 10,000 ft AMSL	Continuous two-way	Yes
E	IFR	IFR from IFR	ATC service and VFR traffic information	250 kts below 10,000 ft AMSL	Continuous two-way	Yes
	VFR	No	Traffic information	250 kts below 10,000 ft AMSL	No	No
F	IFR	IFR from IFR	ATC advisory service Traffic information	250 kts below 10,000 ft AMSL	Continuous two-way	No
	VFR	No	Flight information service	250 kts below 10,000 ft AMSL	No	No
G	IFR	No	Flight information service	250 kts below 10,000 ft AMSL	Continuous two-way	No
	VFR	No	Flight information service	250 kts below 10,000 ft AMSL	No	No

B

METHOD SELECTION

The method selection is one of the most delicate tasks of any research as its impact on the results is potentially significant whilst the actual effect can hardly be assessed. This effect could be quantified by considering multiple methods and trading these off based on their performance. A performance-based trade-off is however computationally expensive and often not even essential because of the limited applicability of some of methods. Still, examples of such applications are presented in literature [4, 5]. The subsequent paragraphs will underpin that this approach to method selection would be overoptimistic for this problem since the conducted trade-off will show only two candidate methods come forward.

Table B.1 lists the considered Machine Learning methods for the ensuing trade-off. These methods were selected by considering their presence in fundamental literature, [6, 7], as well as their applications within the field of safety data analysis, Table B.2. Similarly, the method selection criteria were derived from fundamental literature as well as their motivations for their applications. The motives and criteria are shown by Table B.1 and Table B.2 respectively. In addition, the method selection criteria were validated with set of criteria found in literature [8]. The resulting definitions are listed below.

Scalability	the ability to scale the problem by changing the size of the dataset without compromising performance
Heterogeneity	the ability to cope with heterogeneous data
Dimensionality	the ability to cope with dimensional data
Complexity	the effort required to build the method and the number of design decisions/variables
Interpretability	the extent to which the results are understandable or the effort needed to become (more) understandable
Robustness	the extent to which the method is sensitive to noisy features and/or changes in hyperparameters

The discrepancy between the methods in Table B.1 and Table B.2 can be explained as follows. Some methods, such as Reinforcement Learning, were simply not compatible with the problem at hand. Reinforcement Learning techniques are for instance useful for decision-making or time-series problems. Some other methods, such as Recurrent Neural Networks (RNN) are a more advanced application of an elementary technique, namely a Neural Network.

The unweighted scores of the methods are seen in Table B.1. These scores were assigned based on a fundamental understanding of the methods as well as the decision-making by researchers in the field. The former means that some methods naturally do not suit certain problems. The latter is summarised by Table B.2. The six criteria were then divided over two groups according to a principal difference. That is to say, the nature of the data criteria depend on the considered data and these characteristics cannot be altered. As an example, the heterogeneity of a dataset cannot be reduced to make certain methods more suitable. Heterogeneity is the consequence of the dataset which belongs to the problem. Feature selection might affect the heterogeneity but cannot control this criterion. In other words, the feature selection does not aim to make a problem less heterogeneous, in contrast with its aim to reduce its dimensionality. However, the level of dimensionality reduction also depends on the outcome of feature selection and is, strictly speaking, not guaranteed either. Contrarily, the feasibility criteria are subject to a researcher's preference and the implications of the study.

Table B.1: Method scores.

	Nature of data			**Feasibility**		
	Scalability	Heterogeneity	Dimensionality	Complexity	Interpretability	Robustness
Domain						
SVM	1	1	5	3	2	3
Kernels	2	4	4	2	1	4
Reconstruction						
NN	5	5	5	2	1	5
Distance						
K-Means	1	1	2	5	3	2
k-NN	2	1	2	5	3	1
Rule						
Decision Tree	3	4	5	4	5	3
Ensembled						
X	1	4	4	1	2	3

Table B.2: Overview of applications in Safety Data Analysis.

Author	**Year**	**Method(s)**	**Reason(s)**
Chidester	2003	Unsupervised clustering	Nature of data (multivariate)
Das et al.	2010	Multi Kernel	Nature of data (heterogeneous)
Das et al.	2011	Multi Kernel	Nature of data (heterogeneous)
Budalakoti et al.	2009	Unsupervised clustering	Scalability
Li et al.	2011	Unsupervised clustering	Limited number of hyperparameters (complexity)
Li et al.	2015	Unsupervised clustering	Limited number of hyperparameters (complexity)
Herrema et al.	2016	MLP and Lasso	Performance trade-off
Herrema et al.	2017	Merged Regression Tree	Combined performance assessment
Herrema et al.	2018	Regression Tree	Performance trade-off
Nanduri et al.	2016	RNN based on GRU and LTSM units	Nature of data (multivariate, sequential, highly dimensional), latent features, no feature selection
Janakiraman et al.	2016	Reinforced Learning	Use of 'soft' information
Janakiraman et al.	2017	Reinforced Learning	Use of 'soft' information, improved threshold
Tanguy et al.	2016	Supervised SVM	Performance trade-off, nature of data (text)
Janakiraman	2017	MIL & DRNN	Weakly supervised, nature of data (highly dimensional), scalability, long term relations

When isolating the nature of the data criteria for an equally weighted trade-off Table B.3 is obtained. The weights were set equally as the dataset at hand was dimensional, heterogeneous and rapidly growing. Consequently, Neural Networks outperformed the remaining models with Decision Trees as sole competitor.

Table B.3: Nature of data criteria scores.

	Scalability	Heterogeneity	Dimensionality	
Weight	**1**	**1**	**1**	**Total**
Domain				
SVM	1	1	5	**7**
Kernels	2	4	4	**10**
Reconstruction				
NN	5	5	5	**15**
Distance				
K-Means	1	1	2	**4**
k-NN	2	1	2	**5**
Rule				
Decision Tree	3	4	5	**12**
Ensembled				
X	1	4	4	**9**

Table B.4 lists the scores of the feasibility criteria and shows that the interpretability criterion was disregarded for this study. Although interpretability is an important notion in an operational context it was not an essential industry requirement. Evaluating the systemic approach to predicting safety events prevailed the potential understanding of how these events come to be. Besides, interpretability could, especially in an operational context, also be misleading as it is more than often misinterpreted as causality. Given its decisiveness in this trade-off and the above perspective the weight of this criterion was set to zero. Alternatively, more significant weights could be applied to the remaining criteria. However, this would not affect the position of the interpretability criterion since their combined weight is equally large as Table B.4 shows. Hence, to overcome its influence this criterion had to be discarded.

Table B.4: Feasibility criteria scores.

	Complexity	Interpretability	Robustness	
Weight	**1**	**0**	**1**	**Total**
Domain				
SVM	3	0	3	**6**
Kernels	2	0	4	**6**
Reconstruction				
NN	2	0	5	**7**
Distance				
K-Means	5	0	2	**7**
k-NN	5	0	1	**6**
Rule				
Decision Tree	4	0	3	**7**
Ensembled				
X	1	0	3	**4**

Table B.5 sums up this trade-off by showing te total score per method. From the above discussion it becomes clear that the nature of the data criteria were decisive. Both distance methods as well as the Support Vector Machines were readily unfeasible for this problem based on the assigned score. Neural Networks outperformed Decision Trees by three points basically due to absence of the interpretability criterion. The opposite would be true if the criterion was taken into account in a equal weight trade-off, confirming the trade-off's sensitivity to this criterion.

Table B.5: Total score assigned per method.

	Total
Domain	
SVM	13
Kernels	16
Reconstruction	
NN	**22**
Distance	
K-Means	11
k-NN	11
Rule	
Decision Tree	19
Ensembled	
X	13

C

FEATURE SELECTION

Before choosing a Machine Learning method, it is important to consider whether all data selected is valuable to the investigation, as the dimensionality of the dataset affects the time to train as well as the predictive performance. A common method to reduce dimensionality is feature selection [20, 21, 22]. Because of its importance in current day applications, an abundance of feature selection methods exist. These methods are divided into three major categories: filters, wrappers and embedders. Each of these categories has a fundamentally different interaction with the learning algorithm. In the methodology of this research the interaction between the transformation and data mining step would be affected. Figure C.1 summarises the three different approaches to feature selection. Filter methods function entirely independent from the learning algorithm, meaning that features are selected based on a separate performance measure. In contrast, wrapper methods select feature (sub)sets and evaluate the performance of the learning algorithm for that particular set. Ergo, the learning algorithm and the feature selection algorithm share the same performance metric. Lastly, embedder methods pick features whilst learning and evaluate the effect hereof in a more integrated fashion [23]. An example of an embedder is a Decision Tree algorithm which selects features at each split by selecting features through recursive binary splitting.

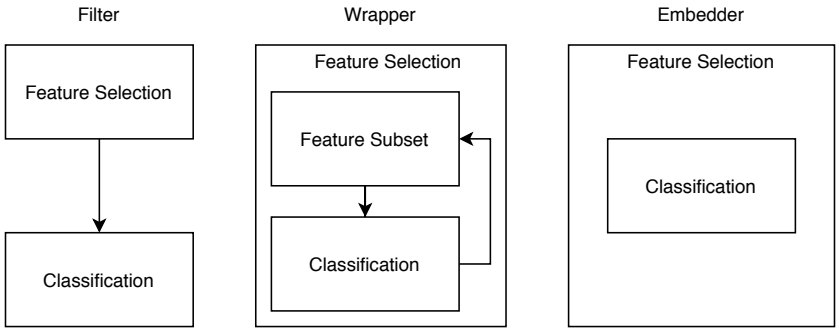


Figure C.1: Different approaches to feature selection [24].

For this particular application a filter method was regarded as the most appropriate category, particularly considering the size of the dataset ($> 10^8$) and its wide application within the domain of safety data analysis. The most commonly applied filter algorithms in related literature were observed to be Relief and PCA [4, 5, 9, 12, 13, 14]. Relief was subsequently chosen over PCA for its dimensionality reduction capacity and its supervised fashion.

Relief selects features according to assigned weights, the independent performance measures of this filter method. These weights are computed as follows. Relief randomly picks N instances, of which the nearest hit and miss are then determined. A hit is defined as an instance of the same class whilst a miss is one of the opposite The distance between these instances is computed using the Euclidean distance, depicted by Equation (C.1). The weight updating is done by computing the difference between the normalised feature vectors. Normalisation is essential in feature selection to rule out the effect of the magnitude of the different features. This normalisation is done by subtracting the mean of a feature column μ of that particular column

and subsequently dividing its by the standard deviation σ. This standardisation procedure is shown by Equation (C.2). The categorical and ordinal features are standardised by means of one-hot-encoding. If the weight of feature exceeds a fixed threshold τ, the feature is selected. The threshold can be set either by visual inspection or by evaluating Equation (C.3), where α is a confidence level and N the number of randomly selected instances [22].

$$d(p,q) = \sqrt{\sum_{i=1}^{N} (p_i - q_i)^2} \tag{C.1}$$

$$\bar{\mathbf{x}} = \frac{\mathbf{x} - \mu}{\sigma} \tag{C.2}$$

$$\tau = \frac{1}{\sqrt{\alpha N}} \tag{C.3}$$

The stability of a filter method is essential as, contrarily to the wrapper and embedders, these have no interaction with the learning algorithm. The stability was assessed by two different means: repeated and k-folded selection. The former simply selects different instances per iteration whilst the latter could expose local inconsistencies by considering a smaller subset of the data. Regardless, if the assigned weights change significantly, either over different iterations or folds, this would imply inconsistency in the selected features and subsequently uncertainty regarding the precursors. Note that the feature selection does not assess the predictive ability of the features but produces a suggestion for potentially valuable precursors. Whether these features are in fact good precursors is to be determined by the Machine Learning algorithm.

Algorithm 1 Relief [22].

Input : Feature matrix, target vector, number of iterations, threshold
Output: Feature weights, features

$N \rightarrow$ Number of iterations
$\mathbf{X} \rightarrow$ Feature matrix
$\mathbf{y} \rightarrow$ Target vector
$\tau \rightarrow$ Threshold

Initialise weights vector $\mathbf{W} \leftarrow [0...0]$

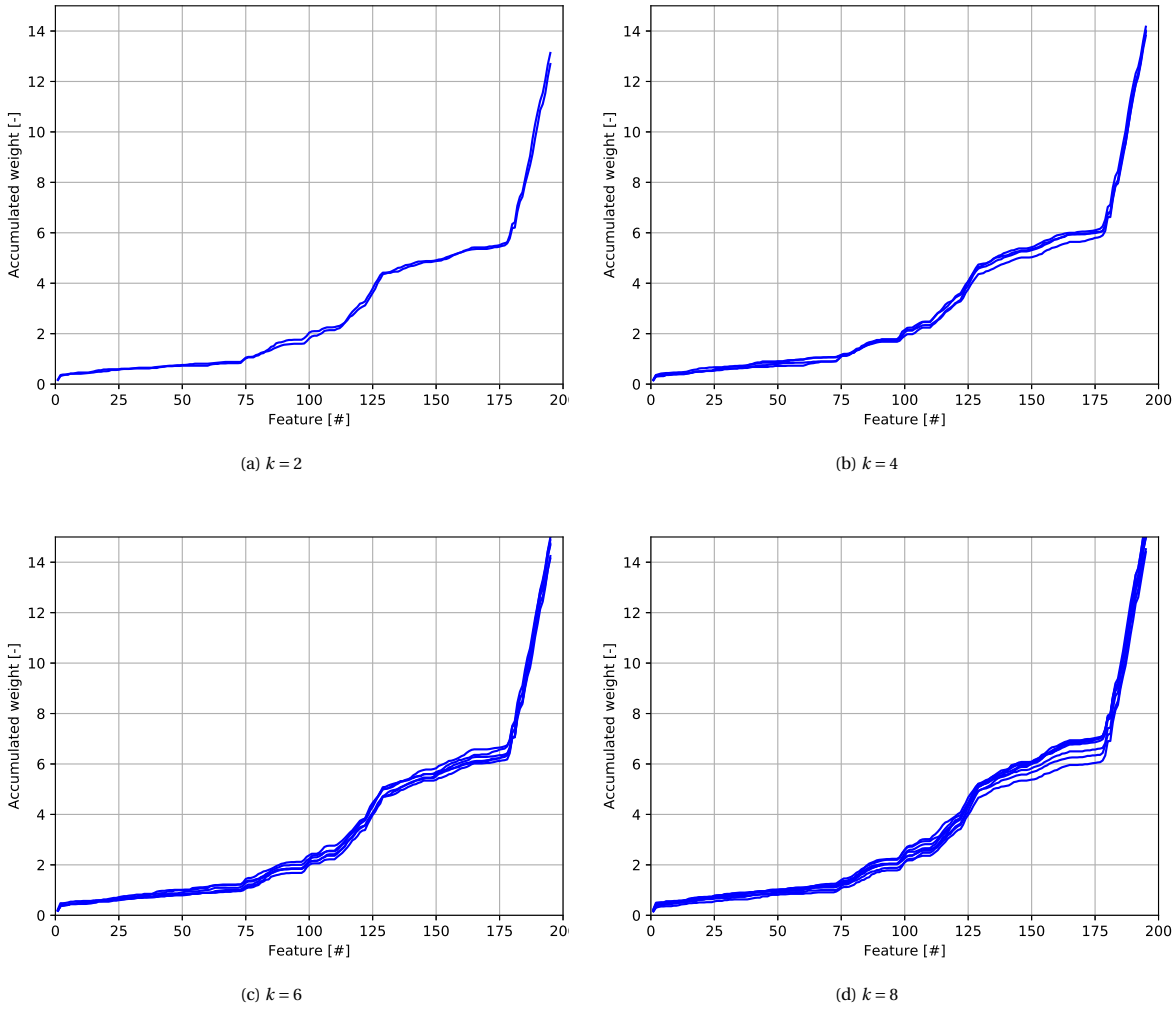
for $i \leftarrow 1$ **to** N **do**
  Pick random feature vector and target $x_i \epsilon \mathbf{X}$ and $y_i \epsilon \mathbf{y}$
  Calculate distance to remaining instances
  Find nearest hit ($\mathbf{x_j}$, $\mathbf{y_j}$) and ($\mathbf{x_k}$, $\mathbf{y_k}$)
  Calculate difference between nearest hit and nearest miss and selected instance $\mathbf{dW} \leftarrow (\mathbf{x_i} - \mathbf{x_k}) - (\mathbf{x_i} - \mathbf{x_j})$
  Update weights vector $\mathbf{W} \leftarrow \mathbf{W} + \mathbf{dW}$
end

Normalise weights $\mathbf{W} \leftarrow \mathbf{W}/N$

for w *in* $\mathbf{W}$ **do**
  if $w \geq \tau$ **then**
    Select feature
  else
    Reject feature
  end
end

Figure C.2 illustrates the accumulated weights for different fold settings. These figures show a similar trend for each fold setting, indicating a strong consistency in the assigned weights. Noteworthy is the steep increase in accumulated weight on the right side of the domain, which implies that important features are

located there each iteration. Figure C.3 shows ten different feature selection iterations, each considering the dataset as a whole. Once again a good consistency is seen in the accumulative weight of the features ensuring the features were selected with confidence. Note that for both cases the weights were accumulated such that the order of the feature vectors could be kept consistent throughout the folds or iterations.



(a) $k = 2$

(b) $k = 4$

(c) $k = 6$

(d) $k = 8$

Figure C.2: Accumulated weights for different fold settings.

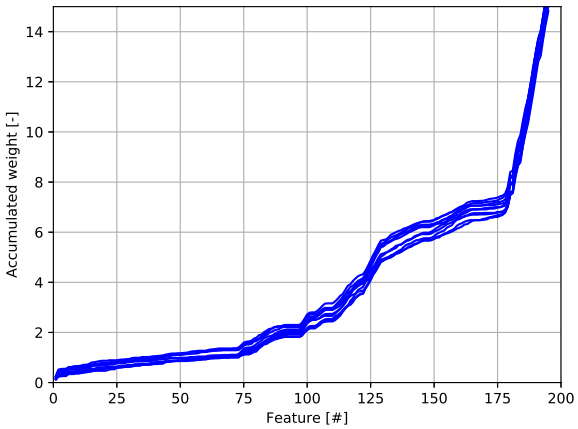


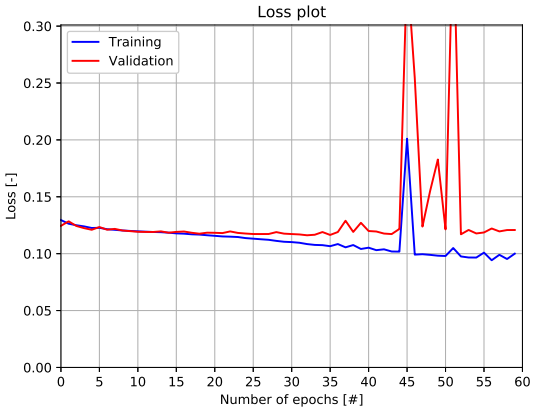
Figure C.3: Accumulated weights of ten different feature selection iterations.

D

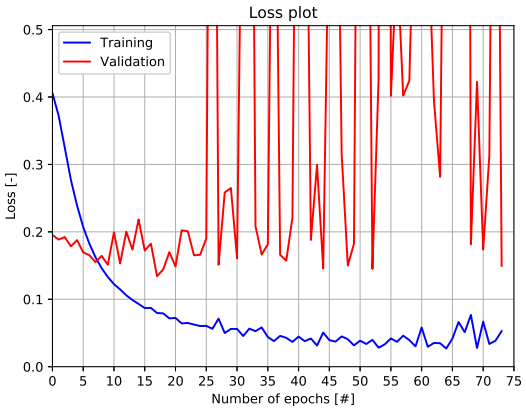
EARLY STOPPING VERIFICATION

Early stopping is a regularisation technique that literally stops the training phase of the learning algorithm early. The point of stopping is determined by criteria set on losses of the weight optimisation. These criteria could be a quotient, a threshold or a patience applied on either the training or validation loss [25]. Their applicability depends on the behaviour of the loss. The most common reason for applying early stopping is to prevent overfitting. Overfitting occurs when the variance between the training and validation loss increases per epoch i.e. the losses of these two datasets diverge.

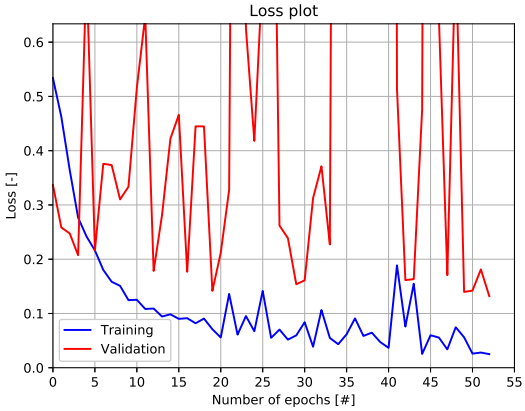
The application of early stopping was found to be essential due to the unsteady behaviour of the training loss of non-regularised experiments. Figure D.1 shows the loss plots of these experiments which were conducted for different oversampling settings. All these curves indicate unsteadiness in training loss. These instabilities were the consequence of poorly representative batches i.e. batches that have little cohesion with the learned batches. These batches inherently updated the weights and subsequently affected the activity of neurons. Eventually this could drift the solution away from its found optimum. As maximum confidence was opted for i.e. minimum loss, this behaviour was undesirable and had to be overcome. Prior to the occurrence of these instabilities the loss did show convergent behaviour. In fact, the training loss had readily levelled off prior to their occurrence and these instabilities only occurred when the training already had a thorough understanding of the problem. This made early stopping a straightforward solution as it could move the point of stopping forward in time i.e. requiring fewer epochs. Moreover, given the insignificant progress it was computationally inefficient to explore this dimension of the hyperparameter domain. Figure D.2 shows the number of iterations required for the non-regularised experiments was substantial, in particular compared to those presented in Part I. This large number of iterations would furthermore imply Bayesian optimisation is not significantly more efficient than a 2D grid search, making this methodological decision questionable. Consequently, early stopping made the hyperparameter tuning much more efficient.



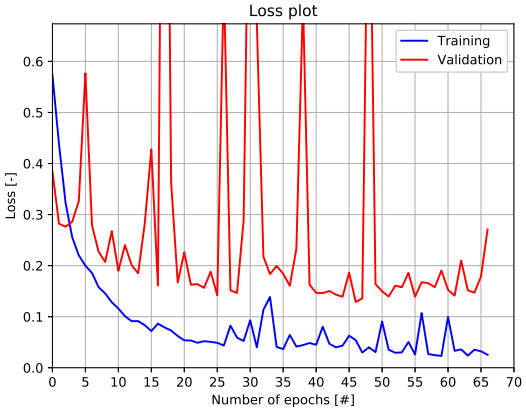
(a) $R_o = 0.03$



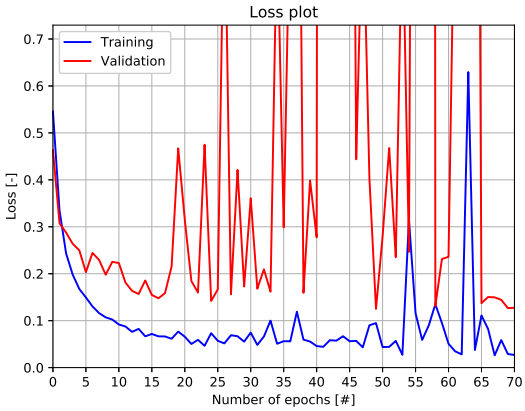
(b) $R_o = 0.20$



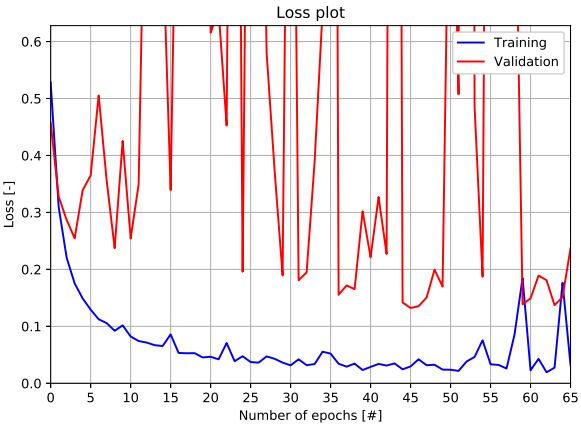
(c) $R_o = 0.40$



(d) $R_o = 0.6$

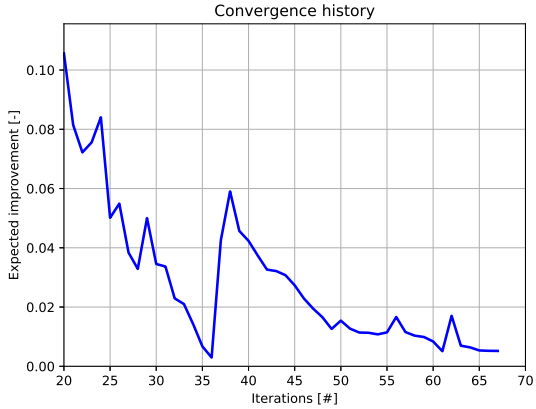


(e) $R_o = 0.80$

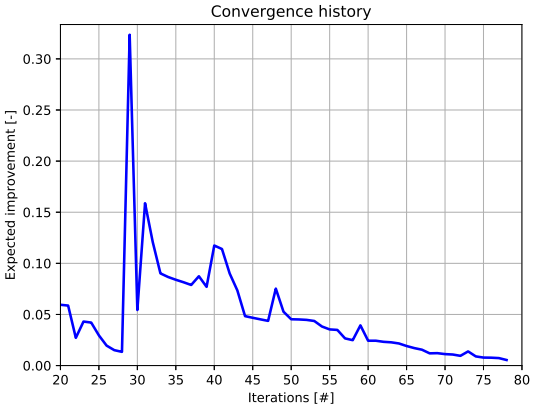


(f) $R_o = 1.00$

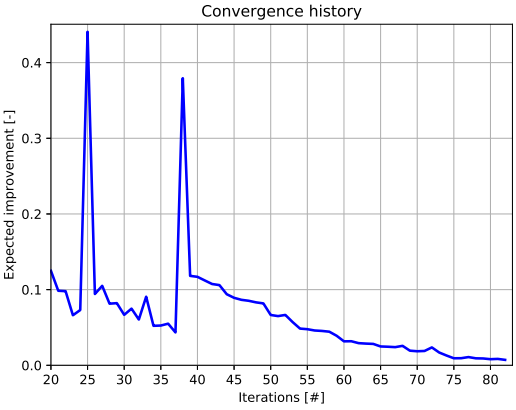
Figure D.1: Non-regularised loss plots.



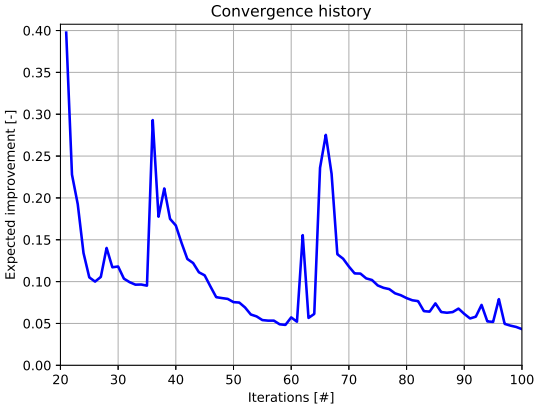
(a) $R_u = 0.03$



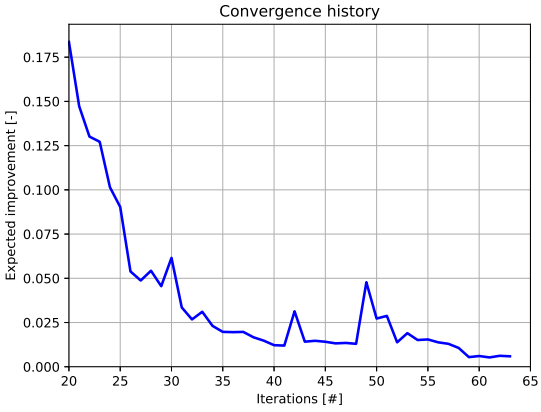
(b) $R_u = 0.20$



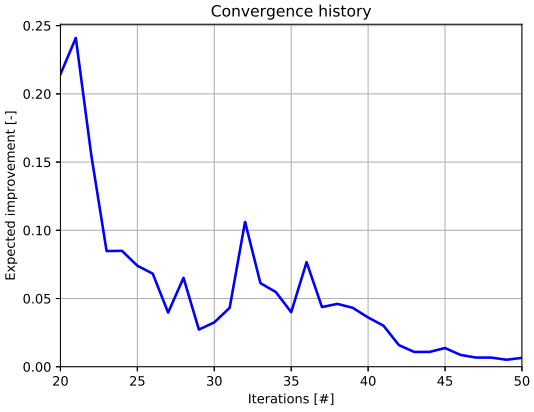
(c) $R_u = 0.40$



(d) $R_u = 0.60$



(e) $R_u = 0.80$



(f) $R_u = 1.00$

Figure D.2: Convergence plots with undersampling.

E

UTILITY FUNCTION VERIFICATION

The utility or acquisition function is key in a Bayesian optimisation as it determines the probing strategy of the optimisation. The most commonly applied utility functions are Probability of Improvement (POI), Expected Improvement (EI) and Upper Confidence Bound (UCB). Snoek et al. state that Expected Improvement and Upper Confidence Bound showed the most promising results in past research [26]. Below the definition of both EI and UCB are provided by Equation (E.1) and (E.2) respectively.

$$EI(x) = \begin{cases} \Big(\mu(x) - f(x^+) - \xi\Big)\Phi(Z) + \sigma(x)\phi(Z) & \text{if} \quad \sigma(x) > 0 \\ 0 & \text{if} \quad \sigma(x) = 0 \end{cases} \qquad \text{where} \quad Z(x) = \begin{cases} \frac{\mu(x) - f(x^+) - \xi}{\sigma(x)} & \text{if} \quad \sigma(x) > 0 \\ 0 & \text{if} \quad \sigma(x) = 0 \end{cases} \tag{E.1}$$

$$UCB(x) = f(x^+) + k\sigma(x) \tag{E.2}$$

where

x	=	hyperparameter vector
x^+	=	best probed hyperparameter vector
ξ	=	exploration factor
k	=	confidence factor
f	=	surrogate function
μ	=	posterior mean function
σ	=	posterior standard deviation function
Z	=	normalised improvement
Φ	=	cumulative distribution function (CDF)
ϕ	=	probability density function (PDF)

UCB evaluates its next probed point at the location where the mean plus k standard deviations is highest. Contrarily, EI probes points at locations where it expects to obtain the highest improvement with respect to its current best. The latter in particular has a major advantage as it represents the absolute expected improvement of the objective function. Hence, the convergence criterion can straightforwardly be set based on the maximum (relative) magnitude of the optimisation metric. In addition, this particular function has the ability to, autonomously, probe exploratory. This ability is important in general domain exploration and, above all, to move away from possible local minima. That is why the EI function was chosen over UCB.

Figure E.2 to E.5 show the mean, two standard deviation and utility function plot for different iterations as well as the general convergence plot of this experiment. These plots summarise how the Bayesian optimisation came to its solution and how EI, compared to UCB, was more efficient in terms of obtaining these results.

Figure E.2a and Figure E.2b show the mean and two standard deviation after the random initialisation. Note that the latter plot is basically the right hand term of the UCB equation for k=2, i.e. 95% confidence. Both figures display that there existed a significant uncertainty as two standard deviation was generally significant with respect to the mean. Prior to this plot only random probing had taken place which is why high uncertainty remained in this large domain. In fact, the utility function was applied to overcome this uncertainty efficiently. Figure E.2c depicts the utility function evaluated at the same iteration count. This function identified a region of progress located at the right outer edge of the domain, where the standard deviation was substantial as well. In that sense both functions, EI and UCB, would have identified the same region of interest. Hence, points were probed in this area as Table E.1 lists. In addition, some fairly random points were probed as a result of the exploration factor ξ, which was set to 0.01 for all experiments, according to best practises. Figure E.3b and E.3c illustrate the standard deviation and utility function plot after 25 iterations. The utility function identified some new regions of interest where the standard deviation was significant too. In other words, both functions would have, again, identified the same regions of interest. However, only the expected improvement already showed to be close to convergence, expecting little future progress. Furthermore, the standard deviation was still significant in other regions which could, potentially, outperform the best solution so far. Such a region is seen at 5 to 9 layers and 80 to 85 units. At 30 iterations the optimisation had readily converged, as Figure E.4c depicts. Note that the convergence criterion was set at a 0.01. The utility function did continue probing because of the patience criterion, which was set to five iterations. Contrarily, Figure E.4b shows that numerous regions that could result in better solutions, although the progress was already less significant. Figure E.5 depicts the converged solution of this particular oversampling experiment. These plots finally show some consistency in the potential, as neither the expected improvement nor the standard deviation expected significant improvement. Still, EI outperformed UCB in terms of efficiency as it had readily converged to a solution five iterations earlier and already established some confidence by the final five iterations.

Table E.1: Probed points by Expected Improvement utility function for $R_o = 0.7$.

i	[#]	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
N_L	[#]	1	15	1	9	1	8	15	1	15	15	15	1	7	15	1	15
N_U	[#]	131	108	98	115	150	141	124	60	131	10	92	82	108	150	10	148

Figure E.1 depicts the probed points of both the random initiation as well as the utility function. The domain coverage of these combined methods is significant underpinning the validity of the results and justifying the application of EI.

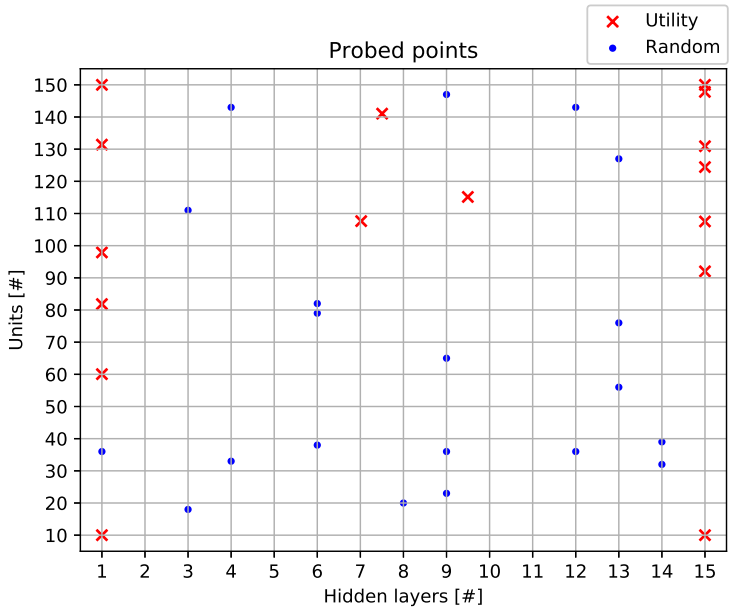
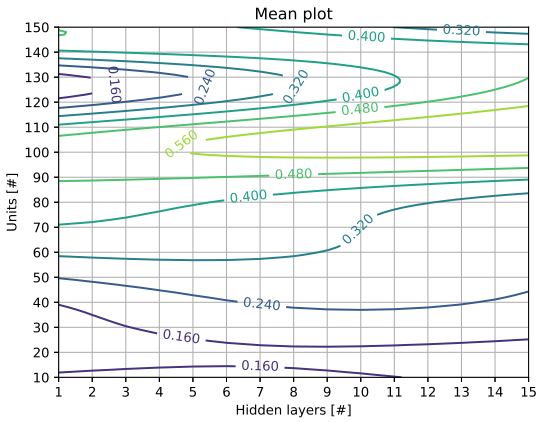
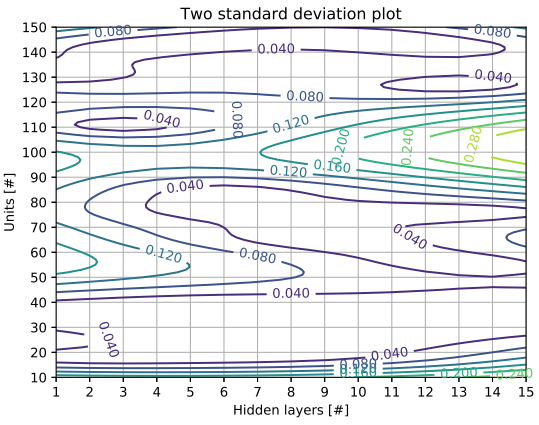


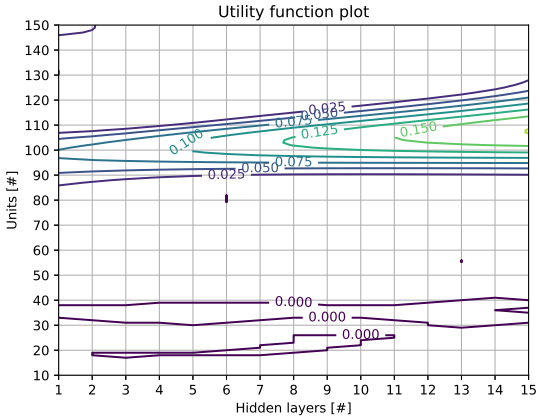
Figure E.1: All probed points for $R_o = 0.7$.



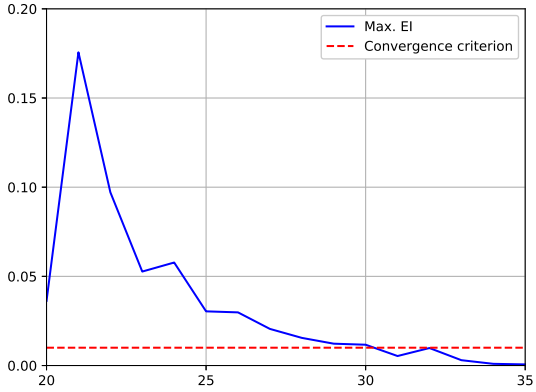
(a) Mean plot $R_o = 0.7$ at 20 iterations.



(b) Two standard deviation plot $R_o = 0.7$ at 20 iterations

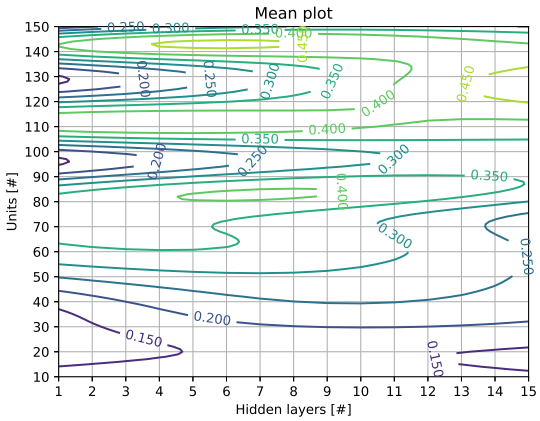


(c) Utility plot $R_o = 0.7$ at 20 iterations.

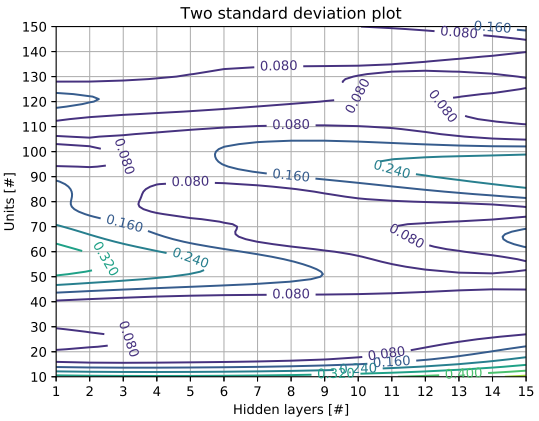


(d) Convergence plot $R_o = 0.7$

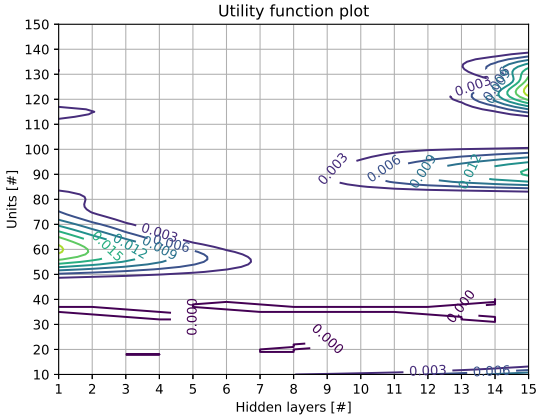
Figure E.2: Bayesian optimisation at 20 iterations.



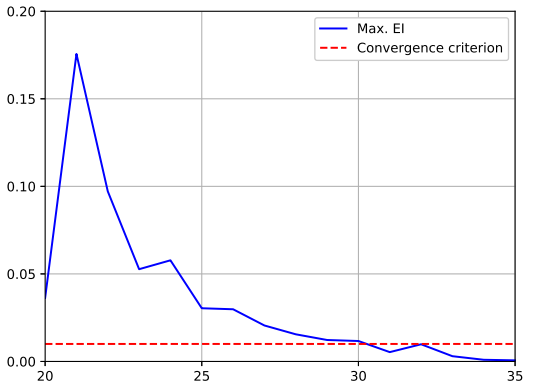
(a) Mean plot $R_o = 0.7$ at 25 iterations.



(b) Two standard deviation plot $R_o = 0.7$ at 25 iterations.

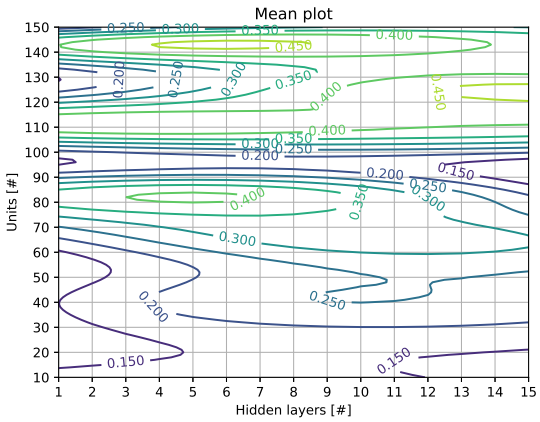


(c) Utility plot $R_o = 0.7$ at 25 iterations.

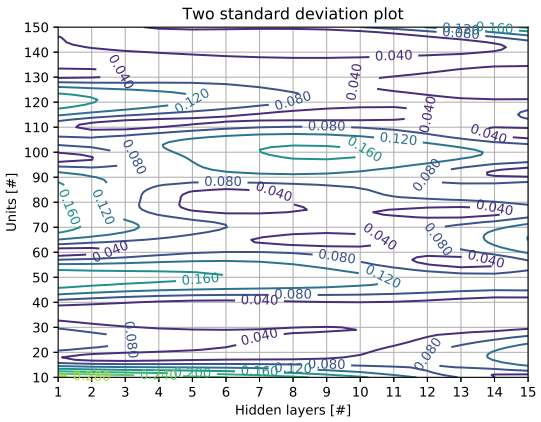


(d) Convergence plot $R_o = 0.7$

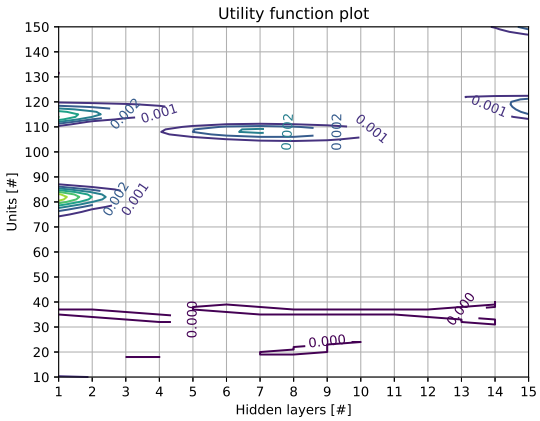
Figure E.3: Bayesian optimisation at 25 iterations.



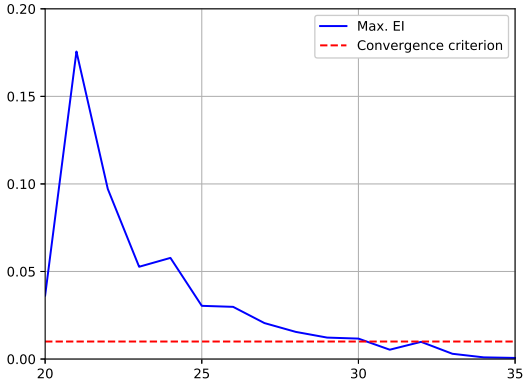
(a) Mean plot $R_o = 0.7$ at 30 iterations.



(b) Two standard deviation plot $R_o = 0.7$ at 30 iterations.

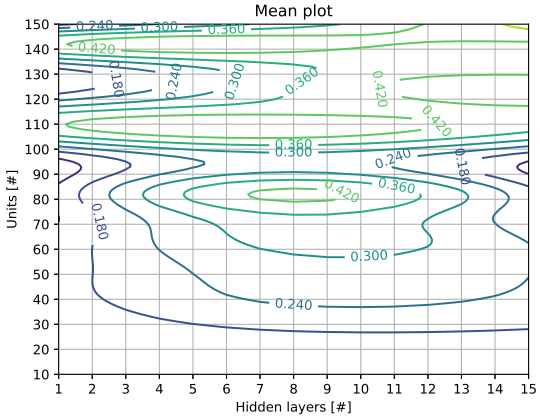


(c) Utility plot $R_o = 0.7$ at 30 iterations.

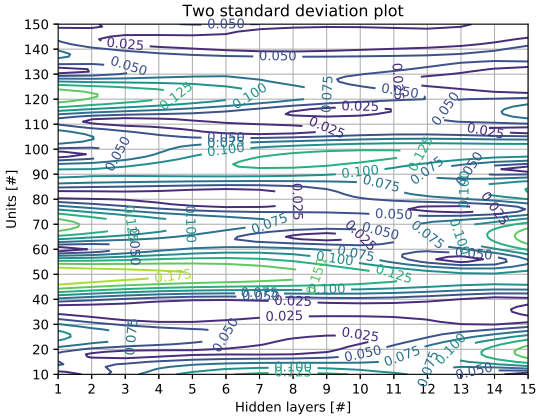


(d) Convergence plot $R_o = 0.7$.

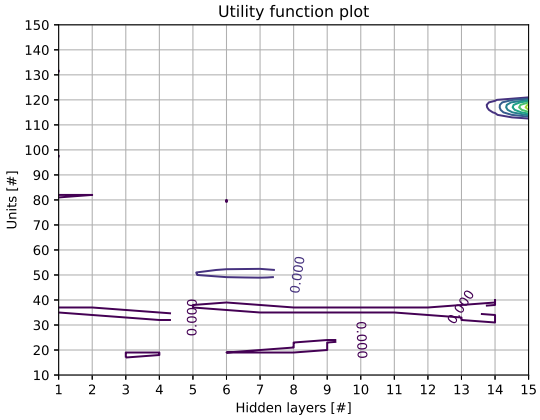
Figure E.4: Bayesian optimisation at 30 iterations.



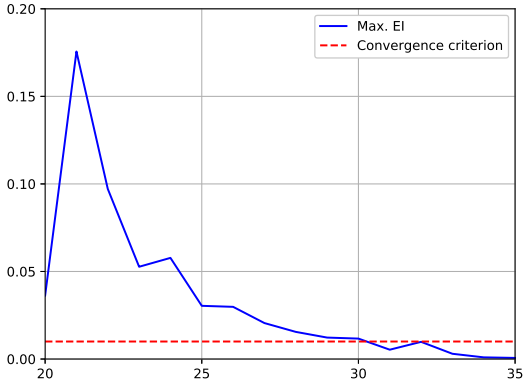
(a) Mean plot $R_o = 0.7$ at 35 iterations.



(b) Two standard deviation plot $R_o = 0.7$ at 35 iterations.



(c) Utility plot $R_o = 0.7$ at 35 iterations.



(d) Convergence plot $R_o = 0.7$.

Figure E.5: Bayesian optimisation at 35 iterations (converged).

F

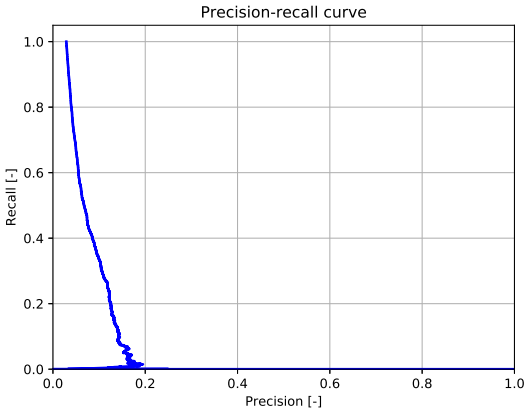
OVERSAMPLING ANALYSIS

Random oversampling was applied to overcome the dominance of the majority class and thereby to enhance the neural network's ability to understand and predict the minority class. In Part I and Appendix G it was readily proven to be an essential methodological step for the trainability of the neural network. Contrarily, it was observed that the Bayesian optimisation rather insensitive as the objective, the *MCC*, levelled off from an oversampling ratio of 0.2 onward, as Table F.1 shows. Hence, the global results, i.e. with a fixed threshold of 0.5, showed a strong coherence in terms of its understanding of the complex patterns it attempted to expose.

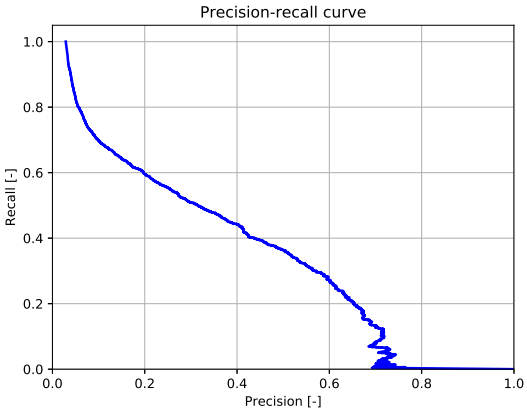
Table F.1: Best probed *MCC* iteration per experiment.

R_o [−]	P [−]	R [−]	MCC [−]
0.03	N/A	N/A	N/A
0.10	0.465	0.400	0.416
0.20	0.458	0.530	0.476
0.30	0.469	0.541	0.488
0.40	0.477	0.528	0.486
0.50	0.461	0.544	0.484
0.60	0.449	0.571	0.490
0.70	0.473	0.573	0.505
0.80	0.439	0.573	0.485
0.90	0.459	0.538	0.481
1.00	0.449	0.573	0.491

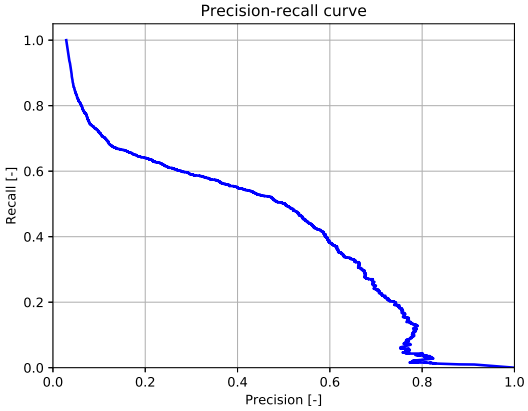
Figure F.1 describes a similar finding but from a local perspective. From Figure F.1b until F.1d it is seen how oversampling enhanced the understanding of the problem by means of the increasing area under the curve, which is representative for the predictive ability of the network. Figure F.1e and Figure F.1f subsequently show very little progress as the general shape of the curve as well as the area under the curve are very similar. In addition, all plots show the performance decisive misclassifications, seen at the both edges of the domain. On the right side of the domain an asymptote at and around a precision of 80% is observed. This asymptote shows that if the threshold is moved towards classifying more instances negative, to the right over this curve, 20% of the still positive predictions are in fact negatives. Similarly, on the other side of the precision-recall curve the steep slope indicates that true positives are located in the small region then declared negative. According to Table F.1 all these instances are located beyond the confidence interval of 50% and therefore these could not be distinguished from the opposite class based on this feature set. The minimal differences in terms of shape, i.e. confidence, can be attributed to some randomness that, insurmountably, occurs. A poorly representative final batch, causing the validation loss to converge at a slightly higher level, is for instance such a source of randomness.



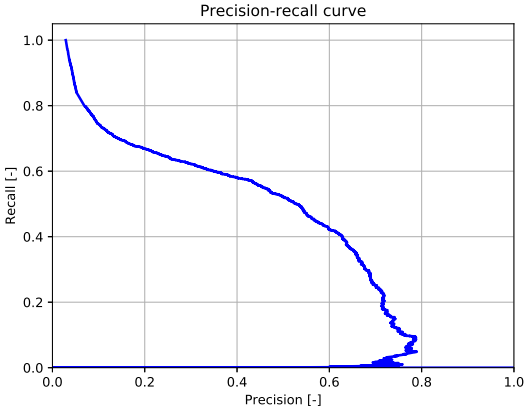
(a) $R_o = 0.03$



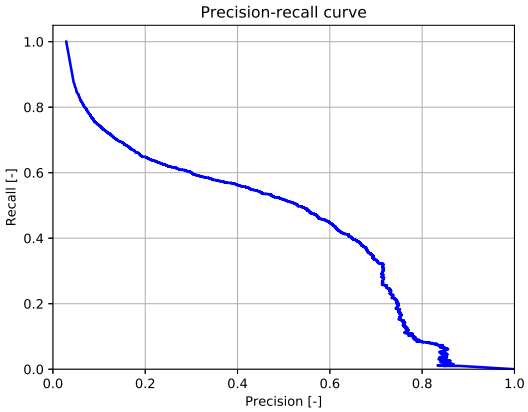
(b) $R_o = 0.10$



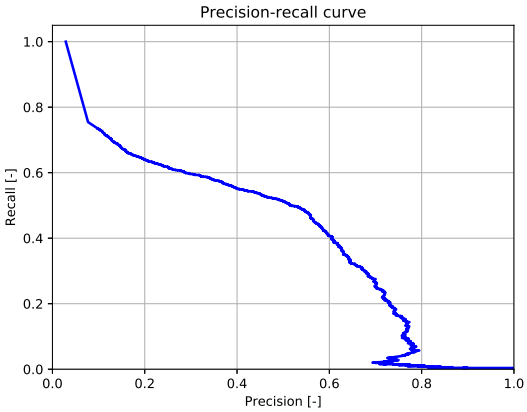
(c) $R_o = 0.20$



(d) $R_o = 0.30$



(e) $R_o = 0.4$



(f) $R_o = 0.5$

Figure F.1: Precision-recall curves with oversampling.

G

Loss Analysis

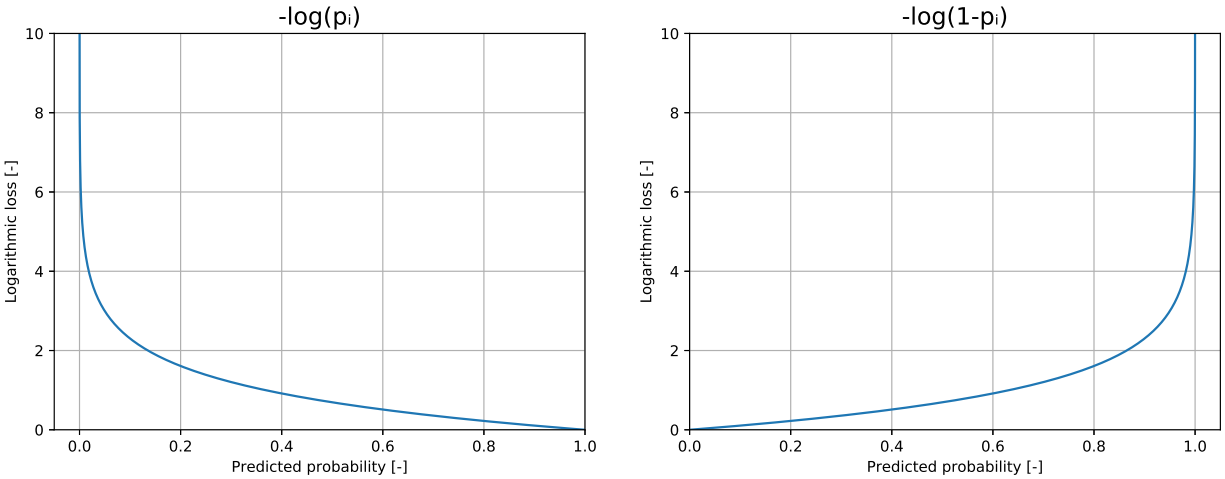
The loss function of a Machine Learning algorithm is the objective function of the weight optimisation. This optimisation aims to establish the best predictive performance for a given set of hyperparameters. For binary classification tasks it is common to use the binary cross-entropy loss function. This loss function is shown by Equation (G.1), where y_j is a vector of actual labels and p_j a vector of probabilities of these instances belonging to the true class ($y = 1$). The binary cross-entropy function is a logarithmic function that evaluates the loss by computing the probability of an instance belonging to the true class, given that it belongs to either the true (y=1) or false (y=0) class. The severity of the misclassification is penalised i.e. confident misclassifications have a significantly higher loss. Figure G.1 shows the evaluation of both components of the loss function. Note that the (y_i-1) in this function serves as an on-off switch based on the actual, binary label of the class. The left curve shows that if the actual label is 1 (true), and the probability of this class is close to 1, the loss approaches 0. Contrarily, the loss is high when the probability of an instance belonging to the true class is low, as can seen from the asymptote on the left side of the domain. The opposite is true for the actual false instances as can be derived from the figure on the right. For the sake of computational efficiency losses are commonly computed in batches. Equation (G.2) shows the loss function computation for an arbitrary number of batches M. This equation shows that the average loss of all batches is evaluated as the general loss of the model.

$$L_j(y_j, p_j) = \frac{1}{N}\sum_{i=1}^{N} -(\log(p_{i,j}) + (1 - y_{i,j})\log(1 - p_{i,j}) \tag{G.1}$$

$$L(y, p) = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{N}\sum_{i=1}^{N} -(\log(p_{i,j}) + (1 - y_{i,j})\log(1 - p_{i,j}) \tag{G.2}$$

The loss of a Machine Learning application should, ideally, behave properly. is notion is defined by the independent trend of the training loss and the interaction between the training and validation loss. An appropriate learning rate is here essential. Figure G.2a shows the effect of the learning rate on the training loss, which can be explained as follows. A very high learning rate rapidly changes the weights between the neurons and consequently the classification. This change is so severe that the loss actually diverges as it cannot locate any minimum in the domain. If the considered learning rate is high, the loss is close to a (local) minimum but jumps back and forth around this point. That is why it initially progresses rapidly but then levels off since it cannot continue its way towards the minimum due to the too large step size. The opposite is true if the learning rate is too low. The loss does go down but at a very low pace. Eventually, the loss will, potentially, converge towards the optimum, but at the expense of many additional epochs. Hence, the perfect learning rate in in between the high and low learning rate and converges to the optimum efficiently.

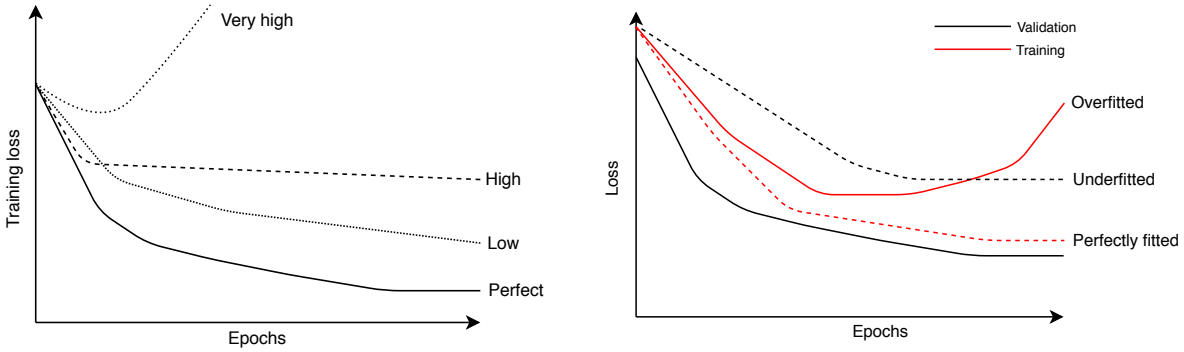
Figure G.2b shows the different, possible interactions between the training loss (depicted in black) and the validation loss (depicted in red), assuming a successful training. A perfectly fitted model roughly converges to the same final loss and, subsequently, has the same overall performance. Overfitting occurs when high variance between the training and validation loss arises over time. In other words, with each epoch a more thorough understanding of the training data is obtained, but this does not result in a better comprehension of the validation data. Consequently, the losses diverge with increasing epochs. The opposite of overfitting is underfitting, which appears if the training is not progressing, i.e. the training loss is not decreasing with increasing epochs.



(a) Logarithmic loss for $y = 1$.

(b) Logarithmic loss for $y = 0$.

Figure G.1: Logarithmic losses for $y = 1$ as true label.



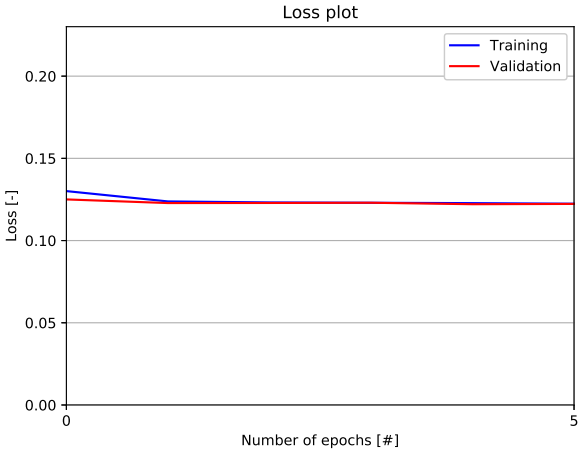
(a) General depiction of the effect of learning rate on the training loss.

(b) Different interactions between training and validation loss.

Figure G.2: General description of loss behaviour.

Figure G.3 shows the loss curves of a number of experiments with varying oversampling ratio. From these curves it becomes clear that without oversampling, $R_o = 0.03$, the network could not learn the problem and that, due to the applied patience, it attempted to do so for five epochs. Furthermore, these curves show that the applied learning rate was appropriate for this problem since all curves are shaped similar to the perfect learning rate curve depicted in Figure G.2a. Hence, the guideline by Kingma and Ba, a fixed learning rate of 0.001, was suitable for this application [27]. Furthermore, Figure G.3 illustrates the non-oversampled experiment suffered from underfitting. Subsequent experiments showed to be marginally overfitted or merely parallel. The latter indicates the neural network lacked information to progress further.

In addition, the loss curves show instabilities in validation data, which apparently had little effect on overall results. Note that the overall results i.e. classification metrics were obtained for a fixed threshold of 0.5. This implies that loss is not proportional to the classification metrics but is mostly a measure of confidence. The origin of these instabilities lies in how the loss of the model is computed and how the required batches are formed. The loss seen in these figures is the averaged loss over M batches, as discussed previously. These batches are established and the representativeness of each batch, with respect to the learned instances, is subject to randomness as each epoch these batches are established and then shuffled randomly. That is why batches could arise with a significant loss although the global classification performance is only moderately affected. During the weight optimisation such peaks in training loss were observed and deemed undesired as each batch updated the weights and affected the overall confidence of the training. To overcome this instabilities, it was decided to apply early stopping.



(a) Loss curve $R_o = 0.03$.

(b) Loss curve $R_o = 0.2$.

(c) Loss curve $R_o = 0.4$.

(d) Loss curve $R_o = 0.6$.

(e) Loss curve $R_o = 0.8$.

(f) Loss curve $R_o = 1.00$.

Figure G.3: Loss plots.

H

SAMPLING TECHNIQUE VERIFICATION

Although oversampling was deemed the more appropriate sampling technique, due to the skewness of the classes, a justification for this decision was looked for by conducting experiments in which random undersampling was applied instead. This sampling method randomly discards instances of the majority class. The undersampling ratio R_u is defined as shown by Equation (H.1), where N_{min} equals the number of instances in the minority class N_{min} and N_{maj} the number of instances in the majority class.

$$R_u = \frac{N_{min}}{N_{maj}} \tag{H.1}$$

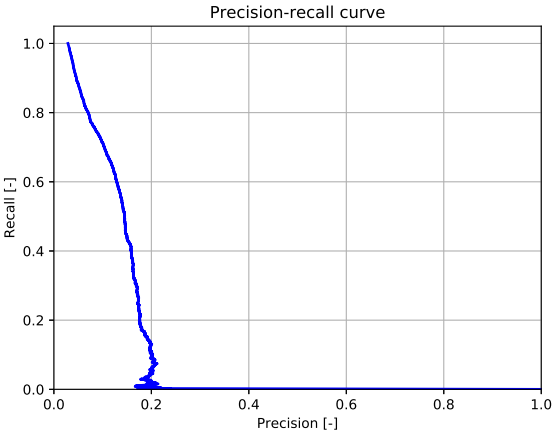
As with the oversampling experiments a grid search was conducted over a range of undersampling ratios. The results of these experiments are listed in Table H.1. The *MCC* was found to be significantly lower for all experiments, with a maximum of 0.236 at the lowest undersampling ratio considered. From this table it can be concluded that undersampling, as oversampling, aggravated the diversion between the precision and recall for an increasing undersampling ratio. The considered classes had lacked the uniqueness required to establish a distinct separation between the two classes. Furthermore, in great contrast with the oversampling experiments, the *MCC* could balance the problem. Recall that, although a similar phenomenon occurred by the application of oversampling, the *MCC* found an optimum number of iterations where the precision and recall metric were of the same order of magnitude. Contrarily, undersampling yielded skewed results which could not be compensated by the *MCC*.

Figure H.2 depicts the loss curves of the undersampled networks which also show overfitting occurred since the training and validation loss diverge. This effect aggravated with an increasing undersampling ratio, the consequence of an even more severe loss of information which made the problem more prone to overfitting. Note that although the magnitude of the loss is similar to that of the oversampling experiments, the performance is poorer because the number of instances is less due to the applied undersampling. Still, the same loss was obtained and that is why the general performance was worse.

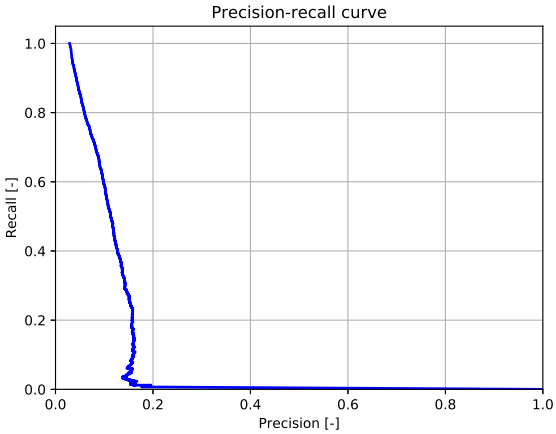
Figure H.1 shows the precision-recall curves of all experiments. The slopes of the curves are steep, which indicates that these classifiers are sensitive to the changing threshold. Consequently, there is little separation between the classes, as mentioned earlier. The area under the curve is negligible indicating little predictive ability. In fact, this ability degraded with increasing the undersampling ratio, stressing the importance of the lost information. In conclusion, undersampling was not an appropriate sampling technique for the problem at hand as the information of the positive instances was shown to be essential for the learning process.

Table H.1: Best *MCC* per undersampling experiment.

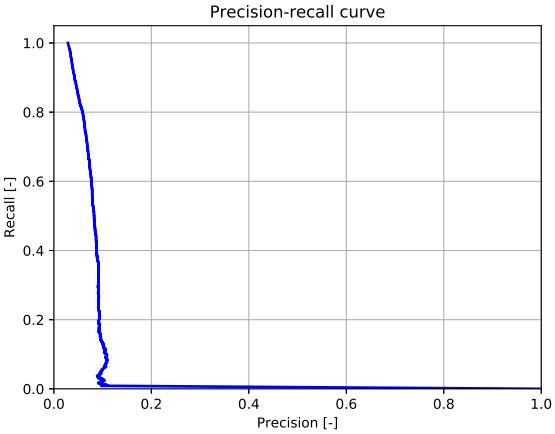
R_u [−]	P [−]	R [−]	MCC [−]
0.25	0.132	0.583	0.236
0.50	0.097	0.637	0.197
0.75	0.073	0.688	0.162
1.00	0.066	0.719	0.150



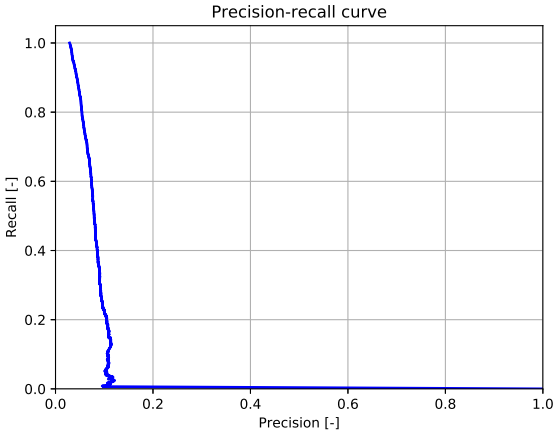
(a) $R_u = 0.25$



(b) $R_u = 0.50$

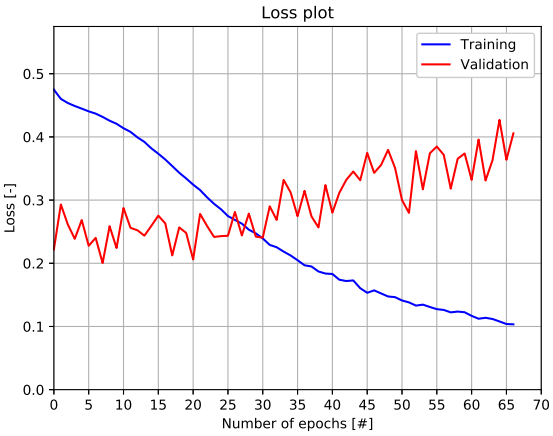


(c) $R_u = 0.75$

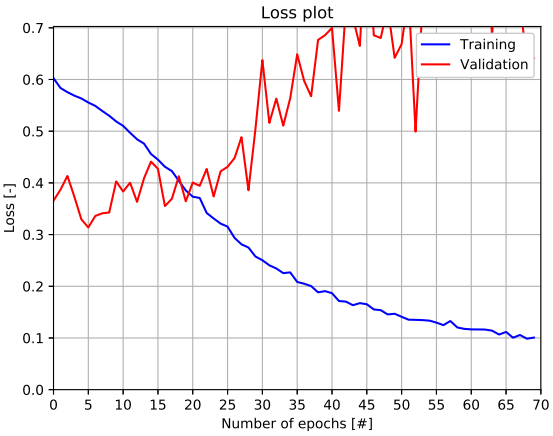


(d) $R_u = 1.0$

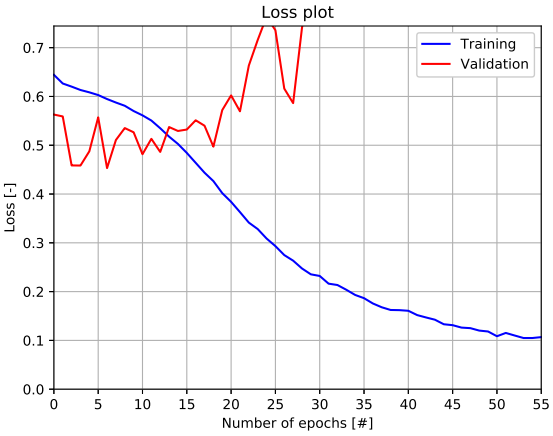
Figure H.1: Precision-recall curves with undersampling.



(a) $R_u = 0.25$



(b) $R_u = 0.50$



(c) $R_u = 0.75$

(d) $R_u = 1.00$

Figure H.2: Loss plots of undersampled experiments.

BIBLIOGRAPHY

[1] International Air Transport Association. *Unstable Approaches: Risk Mitigation Policies, Procedures and Practises 2nd Edition*. 2016. www.iata.org. Accessed: 18-09-2018.

[2] International Civil Aviation Organisation. *Annex 2: Rules of the Air*. 2005. www.icao.int, Accessed: 14-11-2019.

[3] International Civil Aviation Organisation. *Annex 11: Air Traffic Services*. 2001.

[4] F.F. Herrema, V. Treve, R. Curran, and H.G. Visser. "Evaluation of feasible machine learning techniques for predicting the time to fly and aircraft speed profile on final approach". In: *International Conference on Research in Air Transportation*. Vol. 8. Delft University of Technology, 2016, pp. 4–8.

[5] F.F. Herrema, R. Curran, H.G. Visser, D. Huet, and R. Lacote. *Taxi-Out Time Prediction Model at Charles de Gaulle Airport*. Vol. 15. 2018, pp. 1–11.

[6] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An introduction to statistical learning*. Vol. 112. Springer, 2013.

[7] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[8] S. Budalakoti, A.N. Srivastava, and M.E. Otey. "Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety". In: *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews* 39 (2009), pp. 101–113.

[9] T.R. Chidester. "Understanding normal and atypical operations through analysis of flight data". In: *Proceedings of the 12th International Symposium on Aviation Psychology*. 2003, pp. 239–242.

[10] S. Das, B.L. Matthews, A. Srivastava, and N. Oza. "Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study". In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, Washington, DC, USA*. 2010, pp. 47–56.

[11] S. Das, B.L. Matthews, and R. Lawrence. "Fleet level anomaly detection of aviation safety data". In: *2011 IEEE Conference on Prognostics and Health Management*. IEEE, 2011, pp. 1–10.

[12] L. Li, M. Gariel, R.J. Hansman, and R. Palacios. "Anomaly detection in onboard-recorded flight data using cluster analysis". In: *Proceedings of the 2011 IEEE/AIAA 30th Digital Avionics Systems Conference, Seattle, WA, USA*. 2011.

[13] L. Li, S. Das, R.J. Hansman, R. Palacios, and A. Srivastava. "Analysis of Flight Data Using Clustering Techniques for Detecting Abnormal Operations". In: *Journal of Aerospace Information Systems* 12 (2015), pp. 1–12.

[14] F.F. Herrema, V. Treve, B. Desart, R. Curran, and H.G. Visser. "A novel machine learning model to predict abnormal Runway Occupancy Times and observe related precursors". In: *Proceedings of the 12th USA/Europe Air Traffic Management Research and Development Seminar, Seattle, WA, USA*. 2017.

[15] A. Nanduri and L. Sherry. "Anomaly detection in aircraft data using Recurrent Neural Networks (RNN)". In: *2016 Integrated Communications Navigation and Surveillance (ICNS)*. 2016, pp. 5C2–1.

[16] V.M. Janakiraman, B.L. Matthews, and N. Oza. "Discovery of Precursors to Adverse Events using Time Series Data". In: *Proceedings of the 2016 SIAM International Conference on Data Mining*. 2016, pp. 639–647.

[17] V.M. Janakiraman, B.L. Matthews, and N. Oza. "Finding precursors to anomalous drop in airspeed during a flight's takeoff". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada*. 2017, pp. 1843–1852.

[18] L. Tanguy, N. Tulechki, A. Urieli, E. Hermann, and C. Raynal. "Natural language processing for aviation safety reports: from classification to interactive analysis". In: *Computers in Industry* 78 (2016), pp. 80–95.

[19] V.M. Janakiraman. “Explaining Aviation Safety Incidents Using Deep Temporal Multiple Instance Learning”. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, United Kingdom*. ACM, 2017, pp. 406–415.

[20] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*. Vol. 1. 10. Springer series in statistics New York, 2001.

[21] I. Guyon and A. Elisseeff. “An introduction to variable and feature selection”. In: *Journal of Machine Learning Research* 3 (2003), pp. 1157–1182.

[22] K. Kira and L.A. Rendell. “The Feature Selection Problem: Traditional Methods and a New Algorithm.” In: *Proceedings of the 10th National Conference on Artificial Intelligence, San Jose, CA, USA*. 1992, pp. 129–134.

[23] A. Jović, K. Brkić, and N. Bogunović. “A review of feature selection methods with applications”. In: *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE. 2015, pp. 1200–1205.

[24] A. Suppers, A. van Gool, and H. Wessels. “Integrated Chemometrics and Statistics to Drive Successful Proteomics Biomarker Discovery”. In: *Proteomes* 6 (Apr. 2018).

[25] L. Prechelt. “Early stopping-but when?” In: *Neural Networks: Tricks of the trade*. Springer, 1998, pp. 55–69.

[26] J. Snoek, H. Larochelle, and R.P. Adams. “Practical bayesian optimization of machine learning algorithms”. In: *Advances in neural information processing systems*. 2012, pp. 2951–2959.

[27] D.P. Kingma and J. Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).