Wake Vortex Characterisation of Landing Aircraft using Artificial Neural Networks and LiDAR Measurements

- MEng Mechanical Engineering with Aeronautics -

Niklas Louis Wartha

(2262567W)

ENG5041P - James Watt School of Engineering

Advisors: Dr. Anton Stephan (DLR)
Dr. Rene Steijl
Dr. Angela Busse

January 2021

Abstract

An often proposed method for increasing airport capacities to tackle the ever-growing flight demand, is the re-categorisation of currently used aircraft separations. There are good reasons for the existence of these separations: wake vortices generated by an aircraft can be a hazard to those following. The danger is greatest for landing aircraft, as most follow the same glide path before hitting the runway tarmac. The goal is to have dynamic aircraft separations, allowing individual judgement for each aircraft pair, airport and atmospheric condition. The fast-time strength and location characterisation of wake vortices with Light Detection and Ranging (LiDAR) scans is suggested for monitoring predictive Wake Vortex Advisory Systems (WVASs). Current, partly manual, algorithms cannot accommodate this for all LiDAR types. This work proposes the use of Artificial Neural Networks (ANNs) to automate the characterisation of wake vortices. Artificially generated LiDAR proxy data and Multilayer Perceptrons (MLPs) are used to develop suitable network architectures and evaluate different feature engineering. Findings are thereafter applied to measured LiDAR scans, made up of radial velocity measurements from Vienna International Airport, and state-of-the-art perceptual ANNs - Convolutional Neural Networks (CNNs). Feasible feature engineering includes the removal of crosswind using LiDAR scans before overflights, the use of ideal and global measurement grids, and the disregarding of faulty scans as well as velocity measurements at low altitudes. CNNs prove suitable for monitoring WVAS and could be of value for processing large data sets of future wake vortex LiDAR campaigns and routine measurements. ANN characterisation predictions, with circulation errors as low as $26\,\mathrm{m}^2/\mathrm{s}$ and localisation errors as low as $13\,\mathrm{m}$, can be obtained in the fraction of a second, enabling fast-time wake vortex characterisation. The reliability of these results is up to $91\,\%$, suggesting the enormous capabilities of the proposed approach and coming one step closer to dynamic aircraft separations.

Acknowledgements

Throughout the course of this thesis, I received advice and support from several people. First and foremost, I would like to thank my supervisors Dr. Anton Stephan, Dr. Rene Steijl and Dr. Angela Busse. Anton, thank you for holding such interested, enthusiastic and open discussions with me - having an open ear for any small question or doubt, and giving me plenty of freedom in choosing the direction of the thesis. I was given the feeling that the outcome of my work is truly relevant and directly contributes to the wake vortex group's efforts. Thank you, Dr. Steijl, the extensive Zoom sessions we had were very informative and enjoyable, your fascination for aeronautics and research is inspiring. Dr. Busse, I greatly appreciate your willingness to take on an additional final year project, on top of your internally offered ones.

Furthermore, I would like to express my gratitude to the entire Transport Meteorology department for the warm welcome and conversations, even if primarily via Skype meetings. In particular, I would like to thank Dr. Frank Holzäpfel for his deep interest in my work and Grigory Rotshteyn for the countless number of emails and calls explaining, as well as providing, several custom Python modules used for the processing of Vienna campaign data. Moreover, thank you to my colleagues Dr. Norman Wildmann, for providing further Python modules and LiDAR/CNR filter knowledge, Dr. Thomas Gerz, Dr. Simon Unterstrasser, Brigitte Ziegele and Winfried Beer. I am deeply grateful for having had the opportunity to write my MEng Final Year Report within the department of Transport Meteorology at the Institute of Atmospheric Physics, German Aerospace Center (DLR) in Oberpfaffenhofen, Germany during such unprecedented and uncertain times.

I also want to thank Felix Schmitt who allowed me to see results from different viewpoints and gave me the opportunity for several coffee breaks throughout the day. Last, but certainly not least, I would like to thank my family for their continuous support throughout my entire academic career, as well as their collective effort to be my role models.

Contents

Acronyms

ADAM . . . . .Adaptive Moment Estimation

ADE . . . . .Absolute Distance Error

AI . . . . .Artificial Intelligence

ANN . . . . .Artificial Neural Network

CNN . . . . .Convolutional Neural Network

CNR . . . . .Carrier-to-Noise Ratio

CPB . . . . .Convolution-Pooling Block

DLR . . . . .German Aerospace Center

ICAO . . . . .International Civil Aviation Organization

LiDAR . . . . .Light Detection and Ranging

LOS . . . . .Line of Sight

MAE . . . . .Mean Absolute Error

MLP . . . . .Multilayer Perceptron

MSE . . . . .Mean Squared Error

P2P . . . . .Probabilistic Two-Phase Wake Vortex Decay Model

PCDL . . . . .Pulsed Coherent Doppler LiDAR

RCR . . . . .Roll Control Ratio

RECAT . . . . .Eurocontrol Re-Categorisation Program

ReLu . . . . .Rectified Linear Unit

RHI . . . . .Range Height Indicator

RMSProp . . . . .Root Mean Square Propagation

RV . . . . .Radial Velocities

SVM . . . . .Support Vector Machine

SVS . . . . .Secondary Vortex Structure

VE . . . . .Velocity Envelopes

WVAS . . . . .Wake Vortex Advisory System

WVE . . . . .Wake Vortex Encounter

YOLO . . . . .You Only Look Once

Abbreviations

CCW . . . . .counterclockwise

CW . . . . .clockwise

L . . . . .LiDAR

List of symbols

Physical symbols

(l_y, l_z)	LiDAR position in Cartesian coordinates, m
Γ	Vortex circulation, m^2/s
λ	Optical LiDAR wavelength, m
ω	Vorticity, revolutions per second
ρ	Fluid density, kg/m^3
θ	Azimuth angle of LOS around the z-axis, degrees
φ	Elevation angle of LOS from the ground, degrees
A	Area of a surface, m^2
$a(j,k)$	Discrete scan field in polar coordinates with j and k components, -
B	Aircraft wing span, m
b	Wake vortex separation, m
C	Line contour, -
D	Gradient function for finding vortex locations, (m/s)^2
dA	Element of a surface, -
ds	Tangential element along a line contour, -
f_D	LiDAR Doppler frequency, Hz
g	Gravitational acceleration, m/s^2
J	Set of LiDAR beam range gates, -
K	Set of LiDAR LOS, -

L . . . . Lift force, N

M . . . . Aircraft mass, kg

O . . . . Vortex origin/centre, -

R . . . . Range from a LiDAR, m

r . . . . Radius from a vortex centre, m

r_c . . . . Vortex core radius, m

s_w . . . . Wing load factor, -

u, v, w . . . . Velocity components in Cartesian coordinates, m/s

V . . . . Speed, m/s

V_θ . . . . Tangential velocity component of a vortex, m/s

V_r . . . . Radial velocity component of aerosols as seen by a LiDAR, m/s

x, y, z . . . . Spatial components in Cartesian coordinates, m

y_s . . . . Spanwise wing position, m

Artificial Neural Network symbols

η . . . . Training learning rate, -

A . . . . Activation function, -

a . . . . Interim-output of a neuron without activation function, -

B . . . . Mini-batch of LiDAR scans, -

F_h, F_w . . . . Convolutional filter height and width, -

I . . . . Set of inputs to a specific neuron, -

IC . . . . Input channels of data to a layer, -

IS . . . . Input shape of data to a layer, -

J . . . . Set of neurons, -

K . . . . Number of outputs, -

L . . . . Total loss, -

M . . . . Number of epochs used in training, -

N . . . . Data set of LiDAR scans, -

OC . . . . Output channels of data to a layer, -

OS . . . . Output shape of data from a layer, -

P . . . . Trainable parameters, -

t . . . . Target (solution to input), -

w . . . . Neuron connection weight, -

x . . . . Radial velocity input, m/s

y . . . . Parameter output (prediction from inputs), -

z . . . . Interim-output of a neuron with activation function, -

Subscripts

0 . . . . Initial/root

∞ . . . . Freestream/atmosphere

c . . . . Convolutional layer

d . . . . Dense layer

p . . . . Prediction

t . . . . Target

Superscripts

$'$ . . . . Derivative/per unit span

List of figures

List of tables

1. Introduction

1.1. Motivation

This thesis is motivated by two drivers. First, the worldwide economic growth causing an increase in flight demand and second, the advances of Artificial Intelligence (AI) during this decade, specifically Artificial Neural Networks (ANNs), in scientific problem solving and their associated acceleration. The 2018 'Challenges of Growth' report by Eurocontrol concludes that even with currently planned airport expansions, the future demand of flights will not be met [1]. Prior to the COVID-19 pandemic it foresaw up to $1.5\,\mathrm{million}$ unaccommodated flights in 2040. Although the current situation is unpredictable, even the report's most devastating economic scenario forecasts $3\,\%$ of the flight demand to be unaccommodated, resulting in high economic loss [2].

The most straightforward measure an airport can take to increase its capacity is the construction of additional runways, however political opposing, environmental concerns and upset neighbouring communities often hamper such proposal. Another measure is to employ larger aircraft, however it did not take long after the introduction of the Boeing 747 until it was realised that these result in more pronounced turbulent kinetic energy flow in the runway corridor [3]. The complex flow behind an aircraft can be dangerous for following aircraft by for example inducing rolling moments. So-called trailing vortices are a widely studied phenomenon in fluid dynamics as they are an unavoidable consequence of aircraft lift generation [4]. Wake Vortex Encounters (WVEs) can occur throughout any flight state. Most commonly they take place during the approach phase, as all landing aircraft follow the same glide path and in ground proximity wake vortices are unable to glide below the flight path [5].

In order to deal with the increased danger behind the Boeing 747, the International Civil Aviation Organization (ICAO) introduced landing separations that take into account wake vortices [3]. These initial vortex-based separation distances focus on three aircraft weight classes: light, medium and heavy. The recommended separation distances range from 2.5 nautical miles to 6.0 nautical miles [3]. On the one hand, experiments and experience show that these traditional separation distances are conservative [3], opening the door for separation reductions. On the other hand, even with these conservative separations, some aircraft still report WVEs when landing [6]. Both arguments lead to the re-evaluation of aircraft separations. It is suggested that having an universal system for all airports, aircraft and atmospheric conditions is a major problem. Instead, aircraft type combinations (generator-follower) and current weather conditions (primarily wind speed and direction), but also ground obstacles and runway constellations should be considered for less WVEs and higher airport capacities. These factors lead to today's desire of aiming for a Wake Vortex Advisory System (WVAS), which is capable of real-time wake vortex hazard advice, by tracking the strength and location of the wake vortices, and/or by using theoretical models to evaluate safe aircraft separations. Such a system could allow the implementation of dynamic aircraft separations.

In a joint effort, the Federal Aviation Administration and Eurocontrol have outlined a three-phase agenda to obtain and certify such WVAS [2, 7]: the Eurocontrol Re-Categorisation Program (RECAT). Phase one introduces six new aircraft categories in a generator-follower aircraft matrix. For completeness, the new categories and the resulting matrix are illustrated in Appendix A. The second RECAT phase focuses on implementing pair-wise static separations, while RECAT phase three aims to introduce dynamic aircraft separations by also considering environmental conditions. This latter phase is currently worked on and ideally implements prediction, as well as monitoring systems as safety nets for those predictions. However, fast-time monitoring is currently not possible for all measurement instruments. An alternative programme looks at the implementation of temporal aircraft separations, which perform superior with headwinds [6]. Their implementation would also benefit from fast-time characterisation with instruments for routine measurements, as would the evaluation of future measurement campaign data.

1.2. Thesis aims and objectives

The aims of this thesis are as follows: make use of AI for automatic, fast-time wake vortex characterisation - strength and two-dimensional location - from measurements of state-of-the-art Light Detection and Ranging (LiDAR) instruments, balancing accuracy and rapidity, understand its contribution to dynamic aircraft separations, as well as fast data processing for related scientific wake vortex studies and campaigns.

To achieve the above, several objectives are introduced: to clarify the relevance of AI in fluid dynamics, to discuss the suitability of ANNs for wake vortex characterisation, to generate artificial LiDAR scans for developing network structures, to choose a suitable LiDAR scan coordinate system, to normalise the data and select favourable feature engineering, to develop data pre-processing and ANN Python scripts as well as to compare and contrast several ANN types in their characterisation performance and processing time.

1.3. Thesis overview

- **Chapter 2** introduces the evolution of wake vortices and the workings of LiDAR devices.
- **Chapter 3** compares and discusses ways to characterise wake vortices, current algorithms to do so and the aspects of AI which make it attractive for this application.
- **Chapter 4** outlines two data sets, artificial and measurement LiDAR data, their processing and the building of appropriate architectures using the proxy data.
- **Chapter 5** presents and discusses the application of the developed networks using LiDAR measurements to perform feature engineering and statistical analyses.
- **Chapter 6** summarises the conclusions, while also providing an outlook for future tasks.

2. Background theory

2.1. Wake vortex principles

2.1.1. Potential vortex

Vortex flow is one of the key elements required in fluid dynamics for understanding the lift generation of objects [8], with vorticity defined as the curl of the velocity, $\omega = \nabla \times \mathbf{u}$. A simple potential vortex is shown in Figure 2.1. It is made up of concentric circular streamlines, such that the vortex is incompressible and irrotational everywhere, except at its origin O. The velocity along each streamline is constant and equals the tangential velocity V_θ - its magnitude follows an inverse relationship to the radial distance r.

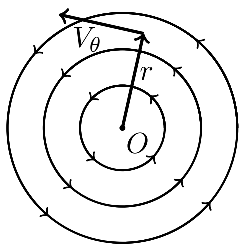


Figure 2.1.: Potential vortex model (inspired by [8], p.263).

It is often desired to understand the flow impact of an entire vortex. This is done via the circulation Γ, defined as the line integral of the velocity in a closed contour C. As described with (2.1), the application of Stokes' theorem allows expressing the circulation by the above vorticity definition over an area A of the surface enclosed by C [8]. In the case of a single, axisymmetric vortex as sketched in Figure 2.1, the circulation can be found via the tangential velocity as shown on the right hand side of (2.1) [4]. This expression suggests an infinite vorticity at the vortex origin, which is why a potential vortex is often also named 'point vortex'. Such a vortex can be extended to a vortex filament by infinitely elongating the origin into and out of the page, with a constant circulation along the filament. Placing an infinite amount of such vortex filaments perpendicular to their elongation direction leads to a vortex sheet.

$$\Gamma \equiv \oint_C \mathbf{u} \cdot d\mathbf{s} = \iint_A (\nabla \times \mathbf{u}) \cdot d\mathbf{A} = \iint_A \omega \cdot d\mathbf{A} = V_\theta (2\pi r) \tag{2.1}$$

2.1.2. Aircraft lift generation

Fundamental concepts on the generation of lift were first introduced by Lanchester [9] and Prandtl [10, 11] in the early 20th century. The circulation around an aircraft's wing enables it to develop lift according to (2.2) - the Kutta-Joukowski lift theorem, where the lift force per unit span of a wing L' is obtained from a certain circulation Γ, flight speed V_∞ and air density ρ_∞ [8].

$$L' = \rho_\infty V_\infty \Gamma \tag{2.2}$$

The lift produced by a wing is often modelled using two semi-infinite straight vortex filaments, forming the so-called bound vortex [8]. According to Helmholtz's second theorem, a filament cannot be terminated in a fluid [8]. Thus, Prandtl developed the lifting line theory [8]: at $-B/2$ and $+B/2$, with B being the wing span, vortex filaments bend towards the downstream direction into infinity to model the counter-rotating trailing wake vortices. This is often referred to as a horseshoe vortex depicted in Figure 2.2.

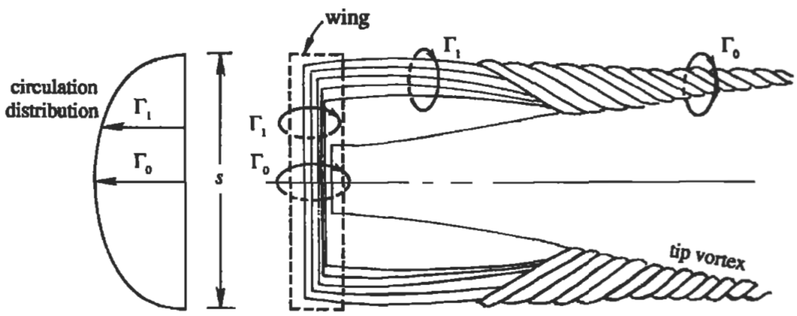


Figure 2.2.: Sketch of the flow around a wing rolling up to a horseshoe vortex (note: in this figure s denotes the wingspan B) (taken from [12], p.646).

These trailing vortices develop due to the circulation around the wing, causing pressure differences with the suction found on top of the wing. The flow curls around the wingtips and sheds vorticity from respective sheets [13]. This causes vortices to roll up, as visualised for a rectangular wing in Figure 2.2. The figure shows that in reality some filaments already detach closer to the root of the wing. Thus, usually several horseshoe vortices are superimposed for modelling different wing geometries and loadings [14]. In the literature, elliptical lift distributions are most widely used, as the majority of wings can be approximated as such [3]. The realistic lift distribution of an Airbus A320 at different altitudes, also taking into account the landing gear, can be found in [15]. In the elliptical case, the circulation at an arbitrary spanwise position y_s, on the bound vortex is defined with (2.3), where $y_s = 0$ is the root of the wing and Γ_0 the root circulation [3, 8]. The latter is computed by balancing the lift and weight forces of the aircraft [3]. Here, M represents the aircraft mass, g the gravitational acceleration, b_0 the initial vortex spacing and s_w the wing load factor ($s_w = b_0/B$ or $\pi/4$ in the elliptical case) [3].

$$\Gamma(y_s) = \Gamma_0 \sqrt{1 - \left(\frac{2y_s}{B}\right)^2} \quad \text{where} \quad \Gamma_0 = \frac{Mg}{\rho_\infty b_0 V_\infty} \quad \text{and} \quad b_0 = s_w B \tag{2.3}$$

2.1.3. Vortex models

Besides the potential vortex discussed in Section 2.1.1, a variety of more sophisticated and realistic models exist [3, 13]. The two most widely used vortex models are Lamb-Oseen [16] and Burnham-Hallock [17], defined by (2.4) and (2.5), respectively. These replace the inner potential vortex region with a core flow that is said to rotate like a solid body of constant vorticity [3]. The core is scaled using the core radius r_c, which is defined via the maximum tangential velocity of a vortex [13]. In this way, the unphysical infinite vorticity at the origin of a potential vortex is accounted for. Flight tests indicate further distinctive regions in a vortex, splitting the core into viscous and vorticity behaviour of a wake vortex [3].

$$V_\theta = \frac{\Gamma_0}{2\pi r}\left[1 - e^{-1.2526\left(\frac{r}{r_c}\right)^2}\right] \tag{2.4}$$

$$V_\theta = \frac{\Gamma_0}{2\pi r}\left[\frac{r^2}{r^2 + r_c^2}\right] \tag{2.5}$$

Each vortex model follows a different philosophy, for instance the Lamb-Oseen model is an analytical solution to the Navier-Stokes equations [18], describing the vorticity dissipation across the surface of a vortex core [3, 13]. While the Lamb-Oseen model features high tangential velocities and a large pressure deficit, the Burnham-Hallock model has significant vorticity dissipation, leading to a dragging tangential velocity reduction [18]. Regardless of the model, for the velocity field of a counter-rotating vortex pair, the flow of two oppositely circulating vortices must be superimposed [3].

2.1.4. Wake vortex evolution and instabilities

The evolution of a wake vortex can be divided into two major categories, the near and far field. The former discusses the influence of the aircraft's geometry on the generated wake. The far field is primarily concerned with instabilities causing the decay of the vortices by, amongst other effects, environmental influences.

The near field flow focuses on co-rotating wake vortices. Wing tips do not exclusively generate vortices, especially an aircraft's high lift wing configuration also results in major flap tip, landing gear, engine and wing-fuselage intersection vortices [15]. However, usually the wing tip and flap tip vortices build the primary wake vortices by merging before reaching the far field. The mutual induction merging process of each co-rotating vortex pair can be separated into four stages [19, 20]:

1. **First diffusive stage:** the laminar core and hence high viscosity of vortices leads to a nearly meta-stable stage. Solely viscous diffusion causes the vortex cores to grow.
2. **Convective stage:** upon reaching a critical core size, the vortex spacing drastically shrinks. This is caused by vorticity leaking from the cores to outer regions, increasing angular momentum. Therefore, to conserve total momentum, the vortices must approach one another.
3. **Second diffusive stage:** merging is finalised via Biot-Savart induction, resulting in a single vortex.
4. **Merged diffusive stage:** the single vortex is received significantly deformed from the previous stage. Further diffusion leads to an axisymmetric vortex by the end of this last stage.

Co-rotating wake vortex pairs are also subject to elliptic instabilities triggered by minor atmospheric turbulence. These cause resonance between the two fundamental vortex structure modes - the base flow of each vortex - and the adaptation of this structure by the strain field of the neighbouring vortex [19]. Hence, elliptical streamlines from one vortex are induced into the neighbouring vortex's core, which can lead to premature vortex merging.

The far field is primarily concerned with the primary counter-rotating vortex pair. Under good atmospheric conditions [4], weak turbulence, wind and neutral thermal stratification, two instability types dominate the vortex pair merging and decay - Crow [21] and elliptic instabilities. The latter is equivalent to the co-rotating case, whereas the former is considered the leading decay mechanism of counter-rotating vortices [19]. Crow instabilities are of low frequency and lead to vortex merging, formation of vortex rings and ultimately turbulent vortex decay by self-induction as depicted in Figure 2.3 [22]. Once vortices are turbulent, decay is rapid, emphasising that their main decay parameter is the growth rate of instabilities [19].

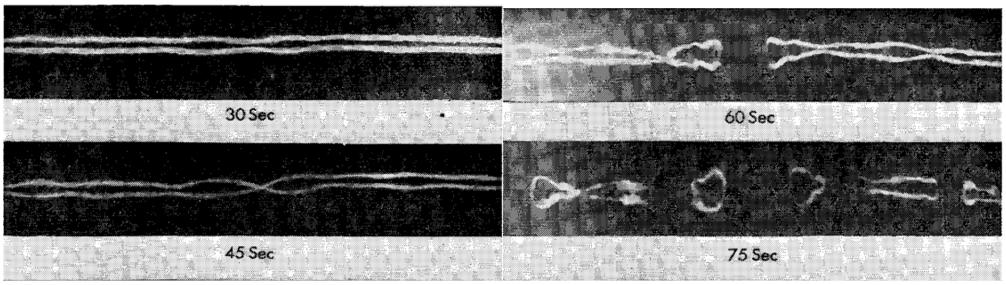


Figure 2.3.: Decay of rolled up wake vortices by Crow instabilities (taken from [21], p.2173).

Crow instabilities are also triggered by minor atmospheric turbulence, however in contrast to elliptic instabilities, they convect the entire vortex filament, not just the core [19]. The straight vortex filaments are subject to sinusoidal vortex oscillations which unfold three mechanisms in the vortex pair [19]: self-induced rotation in the opposite sense to their core flow, induced motion by the neighbouring vortex and motion of the combined perturbations of the vortex pair, which induce rotation and radial stretching in the perturbation planes. Decay is onset once these mechanisms cancel one another.

On top of the discussed instabilities, several atmospheric conditions can alter the decay of wake vortices. These include [3, 23]: strong ambient turbulence, convection, stable/unstable thermal stratification with related buoyancy forces, wind and associated shear forces.

2.1.5. Wake vortices in ground proximity

Close to the ground, the no-slip condition and related shear generation in boundary layers dominate the vortex decay [24]. Ground effects appear when aircraft fly at an altitude lower than their wingspan [13]. The effects on aircraft performance are summarised in [25], however the trajectories of vortices alter as well.

Vortices close to the ground follow outward diverging, hyperbolic trajectories [19]. Once mutual induction has caused the primary vortex pair to descend down to an altitude of $b_0/2$, it is subject to Secondary Vortex Structures (SVSs) caused by its interaction with the ground. Below the described altitude, the pair induces vorticity of the opposite circulation sense on the ground, creating a boundary layer [26]. Secondary vortices can roll up once the adverse pressure gradient in this boundary layer is strong enough [26]. These may cause the rebound or accelerated decay of the primary wake vortices via upward velocities from the SVSs or secondary vortices whirling around the primary [19].

When a crosswind is present, an additional boundary layer is formed, causing an asymmetric trajectory situation in terms of decay and vortex transport as observable in Figure 2.4 [26]. The crosswind supports the downwind secondary vortex, but attenuates the upwind secondary vortex. As a consequence, the primary vortex pair tilts and the downwind primary vortex decays more swiftly. Furthermore, crosswind supports the downwind primary vortex in its lateral movement, but hinders the upwind primary vortex. In the worst case, the upwind primary vortex (with high strength) may stall over the runway and the downwind primary vortex may move towards a parallel running runway [23]. The latter results in parallel runways requiring a $2500\,\mathrm{ft}$ separation for independent operation [27].

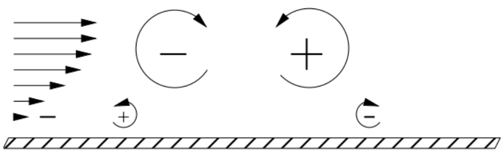


Figure 2.4.: Vortices in ground proximity under crosswind conditions. Large and small circular arrows represent primary and secondary vortices, respectively (taken from [26], p.1253).

With the touchdown of an aircraft, there is a dramatic lift reduction and its bound vortex vanishes [28]. Helmholtz's theorem dictates that the trailing vortices must also diminish and descent, allowing them to reconnect with the boundary layer [26]. Inside the vortex cores pressure differences cause axial flow, and outside the cores self-induced Omega-shaped vortices travel against the flight direction, mitigating the primary vortices [26, 28]. These touchdown phenomena are commonly termed end effects.

Objects on the ground can also trigger these and may therefore be used to mitigate wake vortex hazards [26]. Simulations and field experiments verify that the vortex circulation can be reduced by up to a third, and the decay accelerated in comparison to a flat ground scenario, when so-called plate lines [29] - upright truck tarpaulin plates - are installed underneath the glide path of landing aircraft [26, 30].

2.2. Wake vortex measurements

Measurements for the detection of wake vortices have been successfully done with both LiDAR and Radio Detection and Ranging (RADAR) instruments [31]. The former is the preferred device, as it performs superior under desirable weather conditions [31]. LiDAR capture the movement of air particles, termed aerosols, in the atmosphere. To do so, they emit a laser beam which one-by-one scans individual Line of Sight (LOS). This beam has two positional arguments, an azimuth angle θ and an elevation angle φ, allowing LiDAR to perform several scan types. Additionally there is the range R, defined as the distance from the instrument to a specific measurement point along the beam. Capturing wake vortices requires the scanning of a vertical plane with respect to the ground and perpendicular to the runway, as the vortex planes are perpendicular to the flight direction (out of the page). This gives a Range Height Indicator (RHI) scan type, as sketched in Figure 2.5, with the vortex origins located at (y_O, z_O) and the LiDAR (L) position (l_y, l_z) for both the clockwise (CW) wake vortex and the counterclockwise (CCW) wake vortex.

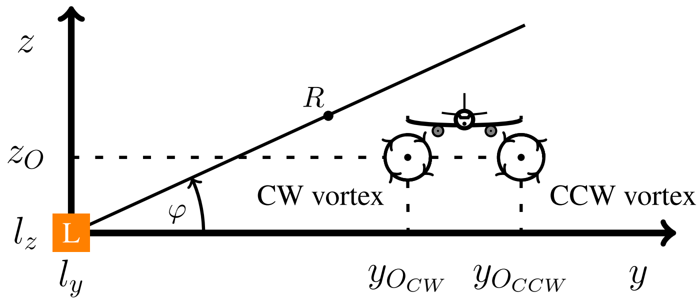


Figure 2.5.: Sketch of measurement geometry (θ rotates around the z-axis).

The two main LiDAR types are continuous wave LiDAR and Pulsed Coherent Doppler LiDAR (PCDL), while the former constantly emits and receives signals, the latter switches between transmitting a beam and receiving the radiation [32]. PCDLs are preferred in the wake vortex application as they have a higher precision at larger distances to the device, allowing them to observe wake vortex evolution for a significantly longer period of time [32, 33]. For the same precision with continuous wave LiDAR, two devices are required in order to perform triangulation [33].

PCDLs obtain radial velocity measurements for several ranges within each LOS as follows [34, 35]: every pulse in the beam has an associated measurement volume, defined by its pulse length and the laser radius. The pulse replicates a normal distribution, such that the backscattered signal from aerosols is not measured at a single point and the LiDAR detector obtains a signal of intensities in time. Through the pulse's time of flight, the range to the LiDAR can be obtained. The time series signal is thereafter Fourier transformed within a range gate and summed to obtain the signal within the frequency spectrum. Comparison of the transmitted and backscattered beam frequency gives a Doppler shift and hence the radial velocity of the aerosols can be obtained with (2.6), where λ represents the laser wavelength and f_D the Doppler frequency [18]. A positive radial velocity indicates aerosols moving away from the LiDAR [15].

$$V_r = \frac{\lambda f_D}{2} \tag{2.6}$$

LiDAR measurements are rated via the Carrier-to-Noise Ratio (CNR), the amount of signal with respect to the atmospheric noise in the backscattered signal. Valid measurements lie between a maximum and minimum CNR. Unrealistically high CNR values can result from hitting hard targets, while low CNR measurements track back to high noise [35]. CNR boosting techniques exist. For instance, the laser beam can be focused onto a focal range. Alternatively, CNR filters such as described in [36] can be applied. Such filters remove invalid and recover acceptable measurements outside the focal range of the LiDAR.

3. Literature review

3.1. Wake Vortex Encounters, severity characterisation and modelling

Wake vortices can, and increasingly will, pose a hazard to aircraft in nearly all flight states, as the relationship of increased flights with WVEs is said to be quadratic [37]. Besides, a likely tropopause rise above the European airspace could also result in additional WVEs [37].

WVEs can cause induced roll, altitude loss or a decreased climb rate and structural stresses, depending on whether the follower aircraft flies through a vortex core, the vortex pair's symmetry line or first through one and then the other vortex, respectively [38, 39]. For completeness, aircraft reactions to a WVE with different paths are shown in Appendix B - these may alter in different flight phases and encounter angles [3].

Thus far, no cruise WVE related accident has occurred, yet several incidents have been reported [40]. For instance, in 2008 a cruising Airbus A319 experienced an intense, unannounced rolling motion after following a Boeing 747 [41]. Serious passenger injuries and interior aircraft damage were the result. Over 73 similar cruise WVE incidents have been reported in the European airspace from 2009 to 2012 [37]. Reduced vertical cruise spacing or free flight concepts, to be implemented in the future due to airspace congestion [40], thus require a thorough understanding of wake vortex behaviour.

Both landing and take-off can generally be considered the most critical phases of flight, given the proximity to the ground and the manual control by the pilots. Consequently, WVEs should be avoided during these flight segments. Still, the greatest fraction of WVEs are experienced at an altitude below $300\,\mathrm{ft}$ [42]. Landing aircraft all follow the same glide path to the runway, whereas for lift-off the path heavily depends on the aircraft type, weight and configuration. Given the low speed, thrust, reaction time and altitude during approach, it is the most hazardous flight state in terms of WVEs [38]. In fact, most research effort is focused on this flight phase [43]. Investigations show that most aircraft conduct a go-around once a potential WVE has been identified, as it is often too late for aircraft restabilisation at low altitudes [2, 6].

In order to model WVEs and quantify the hazard, several metrics have been developed. The strength characterisation of a wake vortex has been suggested to involve the maximum tangential velocity of a vortex or its entire circulation, where the latter is the preferred parameter as it correlates well with the effects expected by WVEs [4]. It is common practice to use the averaged circulation over radii from $5\,\mathrm{m}$ to $15\,\mathrm{m}$ for the evaluation of a wake vortex encounter hazard, since measuring the entire circulation is impossible in practice - a vortex and the ambient flow field cannot be perfectly distinguished [4, 26]. Alternatively, the wake-vortex-induced rolling moment with respect to the possible roll control of the aircraft - the required Roll Control Ratio (RCR) - is employed. A RCR above one corresponds to an aircraft no longer being controllable [44]. In practice, manual control requires a RCR below 0.2 [45]. Further metrics are the wingspan ratio of the follower-generator aircraft and the Roll Moment Coefficient, which is a similar metric to the RCR and has been used in the RECAT scheme for rating encounter severity [37, 43].

The strength characterisation of vortices via their circulation is controversial. Additional vortices in the velocity field can result in the strength being judged differently by the various vortex model types. However, the maximum tangential velocity and RCR are even less appropriate for wake vortex characterisation. In theory, the former is similar to the circulation, but the filtering nature of the measurement process can result in high velocities being missed from the vortex core, underestimating WVE hazards [46]. Also the RCR is less suitable than the circulation, if only wake vortex characterisation is necessary. The RCR requires knowledge about the involved aircraft, which leads to the delay of measurement evaluation. The circulation of a vortex is the most straightforward way for judging the strength of a wake vortex, given the amount of information required and the available methods for high accuracy with this metric.

Characterisation of WVEs is traditionally realised by fitting measured velocity fields to vortex models [31]. Most models used in the literature include a vortex core, however they feature different rates of dissipation, leading to associating the same velocity field to different circulation magnitudes [47]. In this context it is vital to know which model most accurately, reliably and conveniently describes vortices encountered by aircraft. Previously, the Lamb-Oseen and Burnham-Hallock models were presented as suitable, due to their frequent occurrences in simulations. In the research community, it is well known that neither model is perfect, the Lamb-Oseen model is known to underestimate hazards and the Burnham-Hallock model is known for its lower velocity gradients [18]. In the context of pattern recognition, it might be seen beneficial to have distinct patterns with high magnitudes generated by the Lamb-Oseen model.

3.2. State-of-the-art wake vortex characterisation and advisory systems

Throughout the past decades, several attempts have been made to develop wake vortex prediction models and install measurement devices at airports. These can build a WVAS to warn aircraft about potential WVEs and are a key component towards dynamic aircraft separations. Predictive models, such as the Probabilistic Two-Phase Wake Vortex Decay Model (P2P) from the German Aerospace Center (DLR) estimate the location and strength of wake vortices based on theoretical models, aircraft configuration knowledge, weather data and ground proximity information [44, 48]. The problem with predictive models is their accuracy, given the limited input parameters they receive [31, 48]. On the other hand, LiDAR measurements convince with less uncertainty and their ability to capture atmospheric turbulence. However, measurements have high computational costs, often making their evaluation too slow - predictions come too late for overflights [31].

As a consequence, fusion concepts are being developed, combining prediction and measurement systems [31]. Other systems, such as the Wake Vortex Prediction and Monitoring System (WSVBS), also use both principles [44]: the P2P model predicts vortex characteristics and a LiDAR is used for monitoring these predictions [44]. The measurements provide a safety net by verifying the correctness of the predictive model. Characterisation algorithms of LiDAR measurements must be fast and automatic for ensuring the safety and efficiency at airports. The rapid correction of predictions may increase the safety given the reports of WVE related injuries [41]. It could allow to evaluate large campaign data sets more swiftly.

State-of-the-art characterisation precision of automatic algorithms is reported in [33], which uses the Velocity Envelopes (VE) method to characterise wake vortices [32, 49]. They report vortex origin localisation standard deviations of $4.5\,\mathrm{m}$ and $6.5\,\mathrm{m}$ in the vertical and horizontal direction of the Cartesian coordinate system, respectively. Circulation errors are quoted to be $13\,\mathrm{m}^2/\mathrm{s}$ [33]. Unfortunately, the VE method is only applicable to LiDAR with a beam wavelength of $2\,\mu\mathrm{m}$ (PCDL). Although these LiDAR are more common for wake vortex measurements, micro-PCDL with a wavelength of $1.5\,\mu\mathrm{m}$ are preferred when higher precision measurements are desired [50]. These have an inadequate CNR to be evaluated with the VE method, but their higher pulse rate also gives them the capability for precise measurements. Consequently, the Radial Velocities (RV) method has been developed [50, 51]. Its characterisation error greatly depends on the CNR of the measurement, for instance a CNR of $-10\,\mathrm{dB}$ gives errors of $1.8\,\mathrm{m}$, $0.21°$ and $10.3\,\mathrm{m}^2/\mathrm{s}$ for range, elevation angle and circulation, respectively [51]. Overall, both characterisation methods have comparable precision, but their usage depends on the LiDAR type. Currently, the major drawback of the RV method is that it can only be processed accurately with a time-intense manual procedure, briefly described in Section 4.1.1. With its lack of automation, the RV method is not ready for fast-time characterisation - the processing of one LiDAR scan can take around $6\,\mathrm{s}$ [46].

3.3. Artificial Intelligence for identifying wake vortices

Since the turn of the millennium, AI developments have attracted its application in fluid dynamics [52]. Its ability to decipher noisy data also makes it attractive for scientific measurements, especially the wake behind an aircraft, which deals with an enormous amount of dimensions and non-linear flow - most machine learning algorithms can deal with this [53, 54].

ANNs are the most famous machine learning algorithms, performing non-linear regression throughout their hierarchical structure. Linear regression algorithms are not suitable for the non-linear problem at hand. ANNs can automatically extract patterns from data, setting them apart from other AI types such as Support Vector Machines (SVMs), which require significant pattern knowledge to split data into categories [55]. Automatic feature recognition allows ANNs to learn more complex patterns than SVMs, as these do not require to be physically interpretable [56]. On the flip side, ANN results are often hard to interpret [57].

At first, ANNs focused on the classification of images, for example identifying handwritten postcodes [58]. However, also regression tasks followed after, enabling the prediction of coordinates for the localisation of objects, outputting real numbers. First fluid dynamic applications of ANNs occurred in 2002, where only wall information was used to reconstruct turbulence in a channel flow [59]. Since 2012, the industry-wide renounced ImageNet challenge [60] is dominated by Convolutional Neural Networks (CNNs) - a type of ANNs allowing computationally efficient image processing [61]. This then allowed not only recognising objects and patterns in images, but also where they are.

The three most common localisation methods are ‘sliding window’, ‘bounding box’ and ‘key-point detection’. Sliding window crops the original image into many small ones and performs individual classification on each sub-image [55]. It is computationally inefficient [62]. Instead, bounding box approaches combine localisation and classification into one step and return boxes within the entire image, where they believe the desired object is. An example of such approach is the You Only Look Once (YOLO) algorithm [62]. While it can be extremely fast, the localisation is often rather inaccurate, as it simply uses the centre of the predicted bounding box as the object’s location [63]. An additional disadvantage of bounding box algorithms is their inability to output real numbers other than the object’s location, making them impractical for characterising the strength of wake vortices without major architectural adjustments [64]. A novel alternative are key-point detectors. They allow coordinates of the detected points to be extracted [65, 66]. First applications detected features in human faces [67, 68], and later human poses [63, 69] followed. Advantageously, the same architecture can be used to detect other real number parameters, making key-point detectors suitable for the case of wake vortices: both key-points - the wake vortex origins - and their circulations must be predicted. In comparison to pre-built algorithms such as YOLO, key-point algorithms are of lower complexity and shallower (smaller hierarchy), giving more control for adjusting settings for the specific application. In fact, they normally rely on simpler network architectures. While also bounding box approaches may be applicable and potentially provide higher performance, in order to understand the nature of ANNs and receive more control over the algorithm for the present application, shallower networks are seen as beneficial.

Recently, vortex wakes have been classified using fundamental ANNs called Multilayer Perceptrons (MLPs) [70]. Additionally, first studies attempting to detect wake vortices in LiDAR scans by means of ANNs and SVMs have been published [71, 72]. Detection accuracies of 94 % using the ANN YOLO algorithm and 70 % using SVM were achieved, validating the superiority of ANNs for the present application [71, 72]. However, these works were limited to detecting the presence of wake vortices in a LiDAR scan and made no effort to localise nor characterise the vortices. Moreover, CNNs have previously been successfully applied to LiDAR scans for the fast-time prediction of pedestrian positions (10 scans per second) [73].

3.4. Literature deficits and resulting research questions

- Currently, automatic, fast and accurate characterisation of landing aircraft wake vortices is not attainable for all LiDAR types. Thus, how fast and accurate can the characterisation with ANNs be?
- So far, wake vortices have not been characterised in LiDAR scans with ANNs. Additionally, complex networks were adapted to the current problem only to a limited extent. Thus, are key-point detection and shallow CNNs capable of monitoring prediction systems?
- Straightforward MLPs have previously classified vortices. Could they be sufficient?
- Measurement campaigns are cumbersome and their data evaluation impractical, given that current micro-PCDL algorithms are manual. Therefore, to what extent are ANN LiDAR scan evaluations suitable for routine measurement, and can large data sets be tackled with them?

4. Data sets and methods

4.1. LiDAR data

This section describes the two data sets utilised in this work. One data set contains measurement data from LiDAR installed at Vienna International Airport from May until November in 2019 by DLR. The second data set contains artificial ‘proxy’ scans which aim to model LiDAR measurements, but are based on vortex models. LiDAR measurements are highly variable, they include atmospheric effects, SVSs, noise and are potentially erroneous. Instead, the proxy data can be seen as ideal and clean, only entailing the primary counter-rotating vortex pair. As a result, they are less realistic and not suitable for application at airports.

The proxy data is used for developing ANN architectures, the lower variability of the velocity fields allows judgements of network changes to be made more confidently. Afterwards, the measured scans are applied to these architectures, enabling a verdict to be made about the capabilities of ANNs for this application.

Data sets contain the radial velocity RHI LiDAR scans, as well as the vortex origin locations and vortex circulations of the vortices in each scan, the so-called targets R_O, φ_O and Γ for both the clockwise vortex and the counterclockwise vortex (see Section 2.2). Targets are required for the success evaluation of the ANN outputs and represent the known solution to the problem.

4.1.1. Measurement data

Originally, the Vienna measurement campaign was conducted to evaluate the wake vortex mitigation effectiveness of the plate lines mentioned in Section 2.1.5. Therefore, roughly half of the LiDAR scans from the campaign include the effects of plate lines (‘plates up’), while the other half does not (‘plates down’). Both plate states are utilised throughout this report. Details of the measurement campaign are given in [2].

The campaign setup is mapped in Figure 4.1, consisting of five LiDAR positions (L1-L5) with three Leosphere Windcube 200S (1.543-μm) micro-PCDLs used at once [2]. Two plate lines were installed, line one and two consisted of eight and nine $4.5\,\mathrm{m} \times 9\,\mathrm{m}$ plates, respectively [2]. Additional meteorological instruments are highlighted at positions A to C in Figure 4.1.

Aircraft approached the runway (shown on the bottom of Figure 4.1) from the top of the image, hence each LiDAR position captured different overflight flow phenomena [30]: L2 and L4 were positioned in-line with plates, observing wake vortex decay above these, L1 captured the end effects and their interactions with the plates, L3 saw the interplay of the disturbances which spread from both plate lines and L5 looked

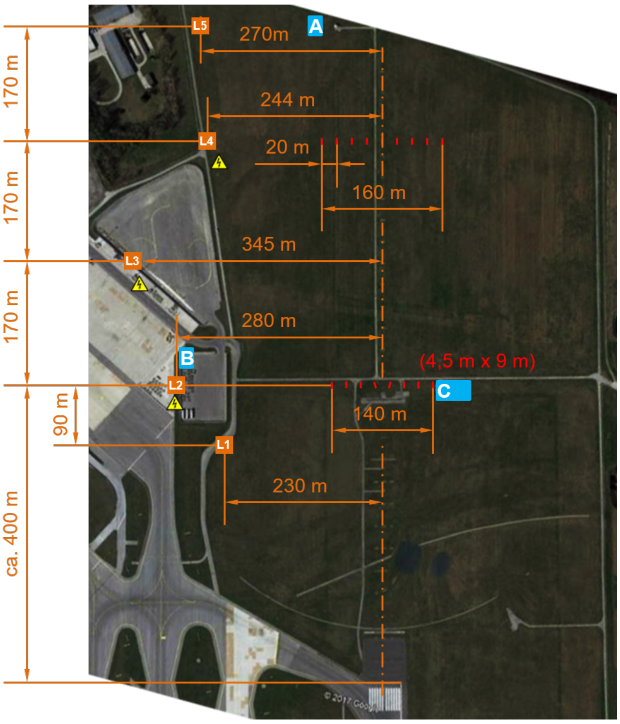


Figure 4.1.: Campaign instrumentation positioning (red dashes = plates) (taken from [74], slide 12).

at plate line disturbances travelling against the flight direction. In addition, each LiDAR position dealt with aircraft at different heights, experiencing unequal amounts of ground effects.

LiDAR measurements of this campaign are structured as visualised in Figure 4.2. A raw LiDAR scan is of the form illustrated in Figure 4.2a, while the measurement points in the Cartesian coordinate system follow the pattern of Figure 4.2b. Each LiDAR measured the radial velocities along LOS in discrete steps of $3\,\mathrm{m}$, from $80\,\mathrm{m}$ to $530\,\mathrm{m}$ (hence, 151 discrete range gates) at an azimuth angle such that the LiDAR was oriented perpendicular to the runway. It is assumed that vortices shared the same plane as the azimuth angle, as only minimal aircraft yaws are expected at low crosswinds, which was the case for most overflights. Additionally, only overflights with headwinds below $2\,\mathrm{m/s}$ are considered. As tabulated in Table 4.1, the discrete elevation angles used for a scan depended on the LiDAR position. Regardless of the position, the elevation angle step was $0.2°$; $50\,\mathrm{ms}$ were required for each LOS change.

Two measurement data inaccuracies must be emphasised. First, since it took some time to change the LOS, a LiDAR scan cannot be instantaneous. For LiDAR position L5, time differences between radial velocity measurements can be up to $7.25\,\mathrm{s}$ in the same scan. Since the LOS was initiated at either end, with enough scans, the effect can be considered negligible. Second, the elevation angle spectra given in Table 4.1 are ideal values, however the actual measurements initiated slightly above or below these.

The simplified grids in Figure 4.2 depict how LiDAR scans consist of range-elevation radial velocities, ordered in terms of range and elevation angle. The grid resolution is constant for scans in the polar coordinate

Table 4.1.: LiDAR elevation angle spectra with respect to their position.

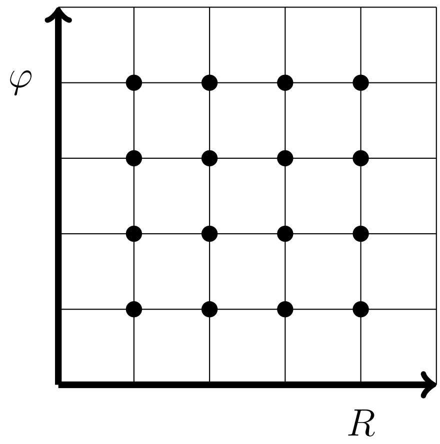
LiDAR position	φ range (degrees)	LOS beams
L1	0-25	125
L2	0-20	100
L3	0-18	90
L4	1-28	135
L5	0-29	145

system, this is not the case for scans in the Cartesian coordinate system. LiDAR measurements occur in the polar coordinate system, reducing the computational time and avoiding transformations addressed in Section 4.4.2. Hence, scans in the polar coordinate system are considered as the standard and the targets for the key-point detection are in polar coordinates. For reference, Figure 4.2 also shows an arbitrary scan from the campaign, as well as a randomly selected proxy scan for illustration purposes only - its generation process can be found in Section 4.1.2. These arbitrary scans reveal two differences between LiDAR scans in polar and Cartesian coordinates. First, the Cartesian coordinate system shows wake vortices at almost equal altitude, while the polar coordinate system distorts the altitude perception. Second, the closer wake vortices are to a LiDAR, the more distorted their representation in the polar coordinate system.

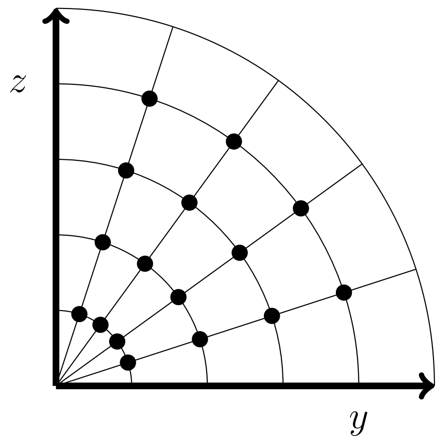
The authors of the measurement campaign assessed the LiDAR scans using the RV method (since micro-PCDLs were used), providing the targets for this data set. The RV method looks at radial velocity LiDAR scans as seen in Figure 4.3a. There are two parts to the RV method, the location and the strength characterisation [50, 51]. In the polar coordinate system case, the localisation consists of the range and elevation angle to the vortex centre of both the clockwise and counterclockwise wake vortex. The vortex strength is defined by the circulation, as discussed in Section 3.1.

Focusing on the localisation first, for each range gate number j, (4.1) is computed along all K LOS (integration of elevation angle) [51]. Once this has been done for all range gates, function D results as shown in Figure 4.3b. To identify which peak belongs to what vortex, it must be recalled that the primary vortex pair is counter-rotating and positive radial velocities are away from the LiDAR. Consider Figure 2.5 and the circular arcs sketched in Figure 4.2b. If the elevation angle of the maximum radial velocity along the corresponding arc of the peak is higher than the elevation angle for the minimum radial velocity, then the peak of D belongs to the clockwise wake vortex, otherwise it belongs to the counterclockwise one. Once the range to both vortices has been established, the corresponding elevation angles can be found by examining the maximum and minimum radial velocities along the arcs. The mean values of the two elevation angles, corresponding to the extreme radial velocities within each arc, are the elevation angles of the vortex centres.

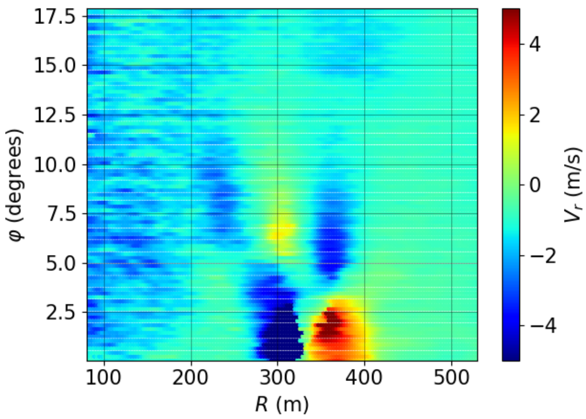
$$D(R_j) = \sum_{k=1}^{K} [V_r(R_j, \varphi_k)]^2 \tag{4.1}$$



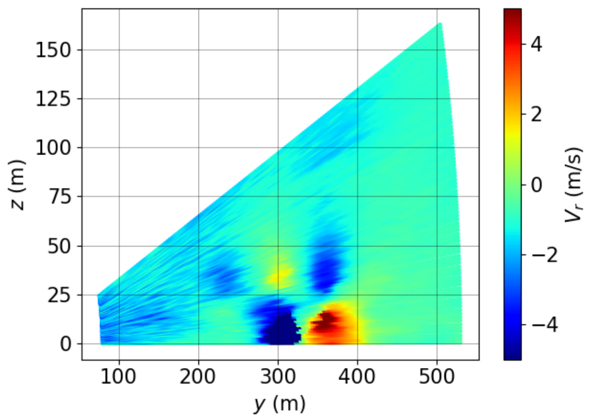
(a) Cartesian scan measurement grid in the polar coordinate system.



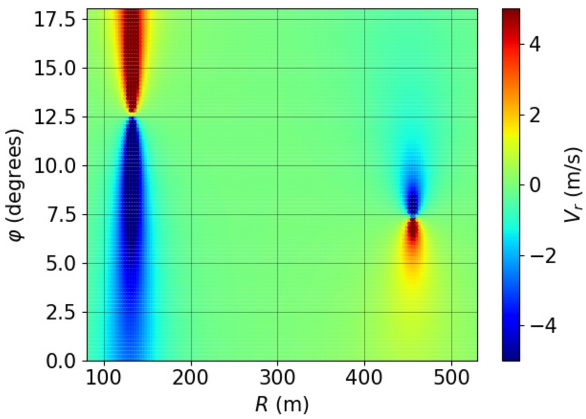
(b) Polar scan measurement grid in the Cartesian coordinate system.



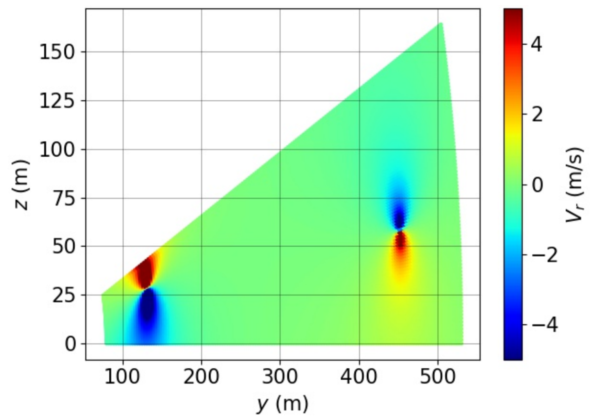
(c) Arbitrary measured scan in the polar coordinate system.



(d) Arbitrary measured scan in the Cartesian coordinate system.

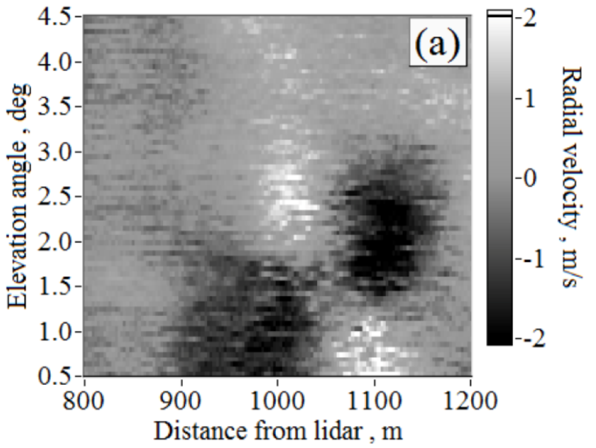


(e) Arbitrary proxy scan in the polar coordinate system.

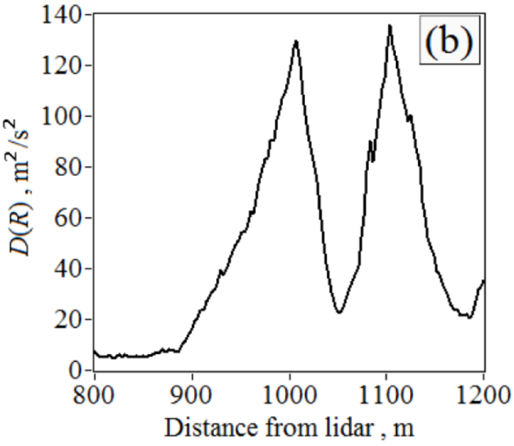


(f) Arbitrary proxy scan in the Cartesian coordinate system.

Figure 4.2.: Measurement points (dots) in a simplified LiDAR scan, as well as one arbitrary measured and one arbitrary proxy scan in both coordinate systems.



(a) Arbitrary radial velocity scan.



(b) $D(R)$ for the arbitrary scan.

Figure 4.3.: Wake vortex localisation with the RV method (taken from [51], p.A1206).

Secondly, the circulation of both vortices is determined as proposed in [50]: a functional, based on (4.1), must be minimised. However, prior to squaring and integrating V_r, a theoretical velocity field generated from an arbitrary circulation value is subtracted. It is an iterative process, and the circulation value with the lowest functional is decided to be the strength of the vortex.

During the measurement campaign, there were around 500 overflights, each measured by up to three LiDAR at once. These LiDAR operated for several moments after an overflight, to observe the morphing and changing of the wake. Each LiDAR recorded around twenty wake vortex scans per overflight. For 289 of these overflights, no plate lines were erected, while for the remainder both were raised. The most common aircraft of this campaign was the Airbus A320, in fact the majority of aircraft belonged to the medium ICAO weight class, with around a quarter being heavy and only five super.

4.1.2. Proxy data

The generation of artificial LiDAR scans aims to have similar discrete elevation-range pairs to the measured LiDAR scans. To keep the data simple and not add too much variation, it is decided to only model LiDAR position L3 from the campaign. The Lamb-Oseen vortex model introduced in Section 2.1.3 is chosen to generate the velocity fields, given its common usage in the literature.

The two vortices in a scan are located independently and may have different strengths. However, to keep the vortices within realistic bounds, their altitudes are restricted to $100\,\mathrm{m}$ above the ground. Furthermore, each vortex can only be generated within one side of the runway. The core radius and strength of each vortex are also randomised within common bounds observed for an Airbus A320 [75]. The randomised parameter bounds are summarised in Table 4.2, location parameters are illustrated in Figure 2.5.

Location bounds are given in the Cartesian coordinate system, within which vortices are initialised. Thereafter, the vortex origins are transformed to polar coordinates. To model measured LiDAR scans, discrete elevation-range pairs are defined in a field $a(j, k)$, where for the proxy data $j \in \{0, 1, ..., 150\}$ belongs to

Table 4.2.: Proxy data parameters with associated bounds.

Location parameters	Vortex parameters
$0\,\mathrm{m} \leq z_O \leq 100\,\mathrm{m}$	$1.5\,\mathrm{m} \leq r_c \leq 5\,\mathrm{m}$
$80.0\,\mathrm{m} \leq y_{O_{CW}} \leq 345.0\,\mathrm{m}$	$100\,\mathrm{m}^2/\mathrm{s} \leq \Gamma \leq 500\,\mathrm{m}^2/\mathrm{s}$
$345.0\,\mathrm{m} \leq y_{O_{CCW}} \leq 520.5\,\mathrm{m}$	

$R \in \{80\,\mathrm{m}, 83\,\mathrm{m}, ..., 530\,\mathrm{m}\}$ and $k \in \{0, 1, ..., 90\}$ belongs to $\varphi \in \{0°, 0.2°, ..., 18°\}$.

The process is analogous for both vortices, details are therefore given for solely the clockwise rotating vortex - when appropriate, disparities to the counterclockwise case are mentioned. For each measurement point in $a(j, k)$, the radial velocity must be found. This is accomplished by taking the approach that follows below. The tangential velocities of a vortex are obtained from the Lamb-Oseen vortex model at specified radial distances from the vortex origin. The radial distance from the vortex origin to each measurement point in a is equivalent to the Euclidean distance to the vortex origin. Therefore, the tangential velocity at each discrete point in a can be found via (2.4).

The tangential velocities must then be transformed to the general velocity components of the Cartesian coordinate system, v and w, for y and z, respectively. The transformation is defined with (4.2) (the signs in the vector on the right hand side must be swapped for a counterclockwise vortex) [18].

$$\mathbf{V_{CW}} = \begin{bmatrix} v \\ w \end{bmatrix}_{CW} = \frac{V_\theta}{\sqrt{(z_O - z_k)^2 + (y_O - y_j)^2}} \begin{bmatrix} (y_O - y_j) \\ -(z_O - z_k) \end{bmatrix} \tag{4.2}$$

The resulting Cartesian velocity components are projected onto the radial vector of a LiDAR positioned at the origin of the Cartesian coordinate system - the LOS to the elevation-range measurement point - to obtain the radial velocity as it would be measured. For this, (4.3) is utilised. The magnitude of the distance from the LiDAR to the elevation-range point is divided, such that only the velocity magnitude is projected onto the LOS.

$$V_r = \frac{\mathbf{V} \cdot \left\{ \begin{bmatrix} y_j \\ z_k \end{bmatrix} - \begin{bmatrix} l_y \\ l_z \end{bmatrix} \right\}}{\sqrt{(y_j - l_y)^2 + (z_k - l_z)^2}} \tag{4.3}$$

Once the radial velocities are found for the clockwise vortex case, the counterclockwise radial velocity field is superimposed, which leads to the final radial velocity field of a counter-rotating wake vortex pair - a proxy scan. Clearly, with this approach the mutual induction, atmospheric effects or influences from the ground are not taken into account. Although standard vortex models are incapable of capturing the complex flow phenomena during landing [15], their simplicity helps to develop ANNs. The generated proxy data nicely corresponds, qualitatively and quantitatively, with measured LiDAR scans of an Airbus A320 aircraft [15]. An arbitrarily selected proxy scan is shown in Figures 4.2e & 4.2f.

4.2. Artificial Neural Network methods

ANNs can be seen as non-linear function approximators. They are comprised of processing units, named neurons, which are ordered in different layers - each layer is a function approximator. As discussed in Section 3.3, the characterisation of wake vortices requires regression ANNs, allowing the output of real numbers. Given the availability of targets in the LiDAR data sets, supervised learning is the preferred AI type. With supervised learning, the network compares its outputs to the desired values - the targets - and aims to reduce their difference. Oppositely, unsupervised learning is the clustering of data without knowing targets. However, supervised learning is computationally less expensive and obtains more accurate predictions [52]. Furthermore, feedforward networks are used - predicted outputs are not fed back into the networks once they have been trained. Key-point detection is compatible with this and the chosen approach of this thesis due to its advantages described in Section 3.3. Note that predictions refer to vortex characterisation estimations by the ANNs and not temporal forecasts.

4.2.1. Artificial Neural Network structure

To comprehend the workings of an entire ANN, the technicalities of a single neuron must first be understood. Consider a single neuron with three inputs x_i and output y as depicted in Figure 4.4. For now, let the inputs be the radial velocities of a single LiDAR scan, each of which is associated with a weight w_i.

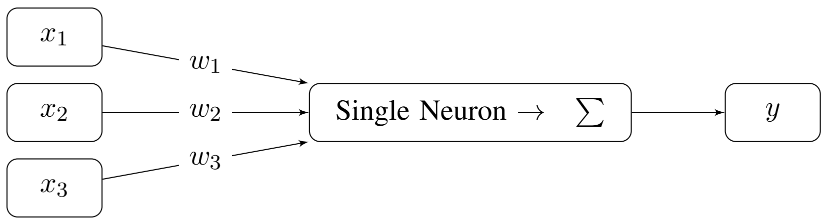


Figure 4.4.: Schematic of a single neuron approximator.

In the simplest configuration, the neuron computes the weighted sum of all inputs, giving the output (or prediction) via a dimensional reduction. This can be summarised with (4.4) for I inputs to a neuron, giving a linear inputs-output relationship [76]. A bias b is added to account for relationships which do not pass the origin (for clarity, the bias is omitted for the description of the algorithm only). Activation functions A add non-linearity to the relationship and control the output type - suitable for this non-linear application.

$$y = A\left(\sum_{i=1}^{I} x_i w_i + b\right) \tag{4.4}$$

The Rectified Linear Unit (ReLu) activation function, defined by $\max(0, \text{input})$, is common in regression problems [55]. As illustrated in Figure 4.5, it sets negative values to zero, but does not alter positive values. ReLu facilitates the fastest training by only highlighting scan regions where a pattern is recognised [61, 77].

The above idea can be extended to multiple neurons, grouped into layers, allowing additional non-linearity for finding more realistic input-output relationships. An ANN with one hidden layer and three neurons is

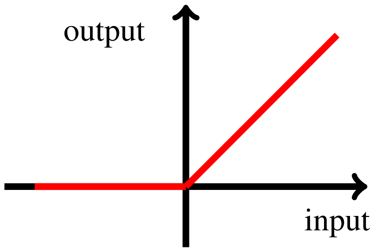


Figure 4.5.: Sketch of the ReLu activation function.

depicted in Figure 4.6, where hidden refers to neither the input, nor the output being directly connected [78]. Each neuron performs the weighted sum of its inputs, giving an interim-outcome a and z once the activation function is applied as described with (4.5) [76]. Here, j refers to a specific neuron of which the interim-output is calculated of and i are the inputs to this neuron. When (4.5) is used for the first hidden layer, $z_i = x_i$ and in the output layer, $z_j = y_j$. More hidden layers allow more complex input-output relationships to be obtained - low level features combine to high level ones.

$$z_j = A\left(\sum_{i=1}^{I} z_i w_{ji}\right) = A\left(a_j\right) \tag{4.5}$$

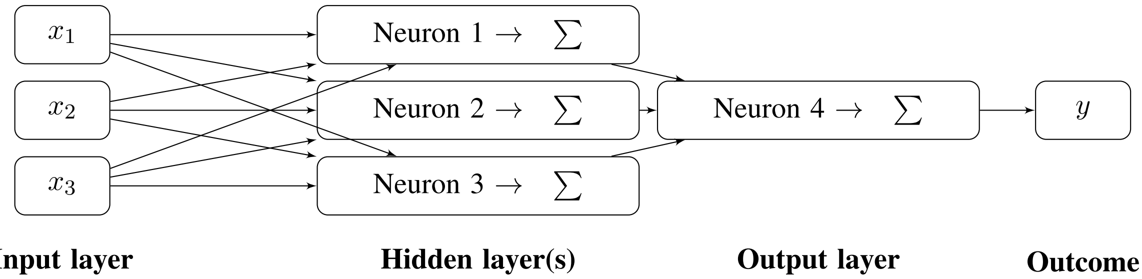


Figure 4.6.: Schematic of a simple ANN (for clarity weight labels are omitted).

4.2.2. Artificial Neural Network training

The aim is to minimise the difference between the targets and the outputs, the loss. This is achieved by finding the best input-output relationship for modelling the problem and varying the weights. ANNs use a ‘gradient descent’ procedure to find/train suitable weights [55]. This iterative method switches between so-called forward and backward passes, in between which the loss is calculated. First, the weight values are initialised using the default ‘Glorot uniform initialiser’ - close to an uniform distribution [79]. Now consider the loss function in Figure 4.7. To find a minimum, its gradient with respect to each weight is computed. Thereafter, each weight is updated in the opposite direction to the associated gradient [55].

Mathematically, the process is described with (4.6) in tensor format, where L represents the total loss function (average loss from a set of N scans), $\eta > 0$ the learning rate (a coefficient deciding how far to move into the opposite gradient direction) and m the epoch [76]. An epoch is completed once all LiDAR

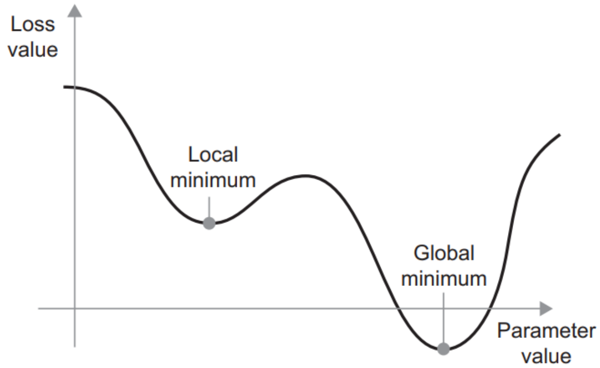


Figure 4.7.: Sketch of a loss function with respect to parameters of an ANN (taken from [55], p.51).

scans have been considered one time - when saving ANN training histories, epochs are used to keep track of the training progress. This thesis uses 100 epochs. Optimiser algorithms can accelerate this iterative process and avoid local minima by adapting the algorithm described herein.

$$\mathbf{w}^{m+1} \longleftarrow \mathbf{w}^{m} - \eta \nabla L(\mathbf{w}^{m}) \tag{4.6}$$

The choice of the loss function depends on the problem, yet regression applications usually employ the Mean Squared Error (MSE) defined by (4.7), where $\mathbf{t}$ is the target tensor of the corresponding outputs [76].

$$L(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} |\mathbf{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n|^2 \tag{4.7}$$

The method used to obtain the gradients depends on the number and types of layers. The mathematics of how the gradients are computed for a network purely made up of dense layers - MLPs - are in Appendix C. Here, solely the final result is given with (4.8), where K represent the outputs of the output layer, n a single LiDAR scan and $'$ a derivative [76]. To understand the specifics for CNNs, [80] should be consulted.

$$\frac{\partial L_n}{\partial w_{ji}} = A'(a_j) \sum_{k=1}^{K} w_{kj} \delta_k z_i \quad \text{where} \quad \delta_k = y_k - t_k \tag{4.8}$$

Gradient descent in the classical sense, termed stochastic gradient descent, evaluates each scan individually, performing as many weight updates as scans in the data set. Stochastic gradient descent is quick, but inaccurate, as redundant computations lead to a heavily fluctuating loss function [81]. Instead, this thesis employs mini-batch gradient descent, where B scans are evaluated in a group. Only once all B scans have had their gradients computed, the parameters are updated. Finding a suitable mini-batch size is not straightforward. On the one hand, huge mini-batches can lead to too few weight updates throughout all epochs, not allowing sufficient model training [82]. On the other hand, tiny mini-batches can lead to high error fluctuations, as weight updates are performed for specific scans rather than a representative data set sample [83]. While a mini-batch size of 128 is common [61, 84], smaller mini-batch sizes deliver superior performance with the data of this thesis. Therefore, a mini-batch size of 10 is used for training and evaluation throughout the remainder of this report. Parameter predictions use the algorithm's default mini-batch size of 32.

The overall training process is visualised with the schematic in Figure 4.8.

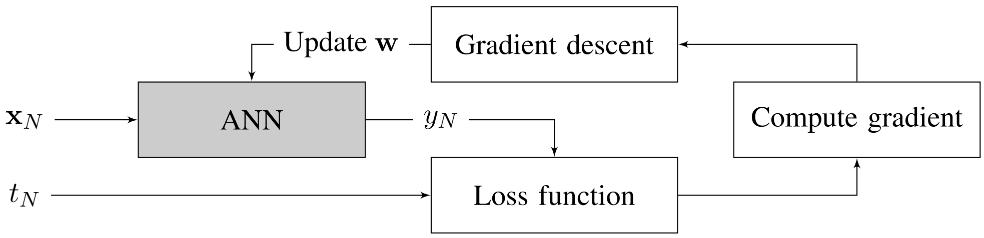


Figure 4.8.: Schematic of the ANN optimisation problem (inspired by [70], p.5).

4.2.3. Artificial Neural Network layer types

Two Artificial Neural Network types are used in this work: Multilayer Perceptrons, which consist of solely dense layers, and Convolutional Neural Networks, state-of-the-art perceptual ANNs that contain convolutional layers. CNNs are more reliable in making assumptions about the nature of the input, requiring fewer connections and parameters [61]. Nonetheless, MLPs may be sufficient for the current task (see Section 3.3), allowing faster training. These networks make use of different layer types introduced below.

Dense layer:

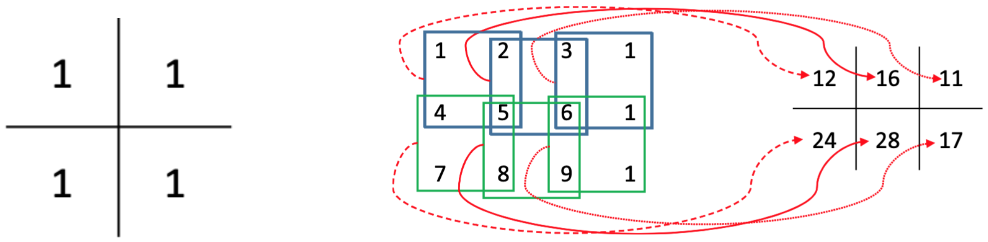
Its structure corresponds to the hidden layer in Figure 4.6, where each neuron is connected to all neurons in the previous and following layer. Within the layer itself, neurons are not interconnected [85]. The dense layer is considered the basic ANN layer. It requires a one-dimensional array of radial velocities as input shape. The number of neurons in the first hidden layer is less than the number of inputs, thus the neurons can look at several radial velocities at once. This is particularly relevant for MLPs, where the radial velocities are in the order of increasing elevation angle, and within each LOS, in order of increasing range. This implies that neurons in the first hidden layer are observing a one-dimensional region of a LiDAR scan - a part of a single LOS. Within each LOS part, the alignment and context of radial velocities is understood. The number of trainable parameters of such a layer is calculated using (4.9), where P_d is the number of trainable parameters from a dense layer l, OS is the output shape and IS is the input shape. The parameters are made up of the weights and bias of each neuron in the layer.

$$P_d^l = OS^l \left(IS^l + 1 \right) \tag{4.9}$$

Convolutional layer

This layer performs convolution using 'filters' that lead to activation maps, indicating where and at what magnitude patterns are found [55]. Filters are learned weight matrices representing patterns. Instead of inspecting a single radial velocity, two-dimensional receptive fields provide ANNs with context understanding, enabling them to comprehend velocity arrangements. Regardless of a vortex's position, the same resources are required to identify it. Convolutional filters in this work have a standardised size of 3×3 [55].

The input to a convolutional layer is three-dimensional, comprised of the width and height of a LiDAR scan grid and the number of channels (also referred to as depth of the data): usually, RGB images have three channels, but the scalar radial velocity scans only have one channel - each grid position is associated with one real number. An example of the convolutional operation is given in Figure 4.9, where the filter is applied separately to each marked box in the input on the left hand side of Figure 4.9b. Corresponding elements are multiplied and the results summed for that box. The outcome is an activation map (right hand side of Figure 4.9b), which by means of the magnitude of each field, indicates where the filter's pattern is most probable.



(a) 2×2 filter. **(b)** Input (left) and resulting activation map (right).

Figure 4.9.: Example of the convolutional layer operation (taken from [80], p.12).

Two behaviours should be noted for the operation of the convolutional layer: padding and stride [55]. Padding means that the activation maps resulting from a convolutional layer are post-processed by attaching additional borders. This avoids the data size to not be affected by the convolution operation. Stride refers to how many pixels the blue boxes - receptive fields - are shifted from one to another in Figure 4.9b. Both padding and stride influence the format of the output of a convolutional layer. For the purpose of this work, standardised values are chosen [55]: padding is applied such that the two-dimensional input and output size of the data is equivalent, moreover a stride of one is selected.

By using the same filters across the entire scan, less weight parameters are required in comparison to dense layers. However, this shift invariance could be problematic with LiDAR scans in the polar coordinate system. Figure 4.2 already revealed that vortices of similar strength change their scale depending on their vicinity to the LiDAR. Consequently, the overestimating or underestimating of vortices close or far from the measurement device could occur. Thus, Section 4.4.2 investigates whether it is advantageous to first transform radial velocity scans to the Cartesian coordinate system.

The number of trainable parameters of such layer is calculated using (4.10), where P_c are the number of trainable parameters from a convolutional layer, OC is the output channel size, IC is the input channel size and $F_h \times F_w$ the filter height and width, respectively. Again, the '1' represents the bias.

$$P_c^l = OC^l \left[(IC^l \times F_h \times F_w) + 1 \right] \tag{4.10}$$

Pooling layer:

Its purpose is to select salient features, reduce the number of ANN parameters and perform a dimensional reduction, forcing high-level features to be learned from many small ones in activation maps [85]. It is applied after a convolutional layer to form a Convolution-Pooling Block (CPB) and therefore uses the same data shape as a convolutional layer. 'Max pooling' is usually the preferred pooling type [86], as in a receptive field of an activation map, only the highest value is passed - the most salient feature. It can detect subtle local features, which 'average pooling' would miss [86]. While convolutional layers have shift invariance, max pooling layers also possess built-in rotation invariance [87]. In this work, common max pooling layer settings are chosen [54]: a filter size of 2×2 and a stride of 2, such that the scan's two-dimensional shape is reduced by a factor of 2 after the layer's application. The pooling layer has no trainable parameters.

Flatten layer:

It converts highly dimensional data to one dimension. Regression ANNs require a single neuron as their output layer (with no activation function) such that the output can be any number range [88]. This can only be facilitated using a dense layer. As a consequence, when using convolutional layers the data structure must be transformed to the format used by dense layers. In other words, three-dimensional data is converted to one-dimension. The flatten layer concatenates each row of a specific activation map, one after another, and then attaches these in one large one-dimensional array. This layer has no trainable parameters.

4.2.4. Artificial Neural Network implementation

The focus of this work is put on the performance of ANNs, as a consequence existing machine learning frameworks are employed. The majority of these use Python as a programming language due to its open-source nature. Python has earned a large user base in industry and academia due to its wide availability of libraries, ease of use and accessibility. The machine learning framework of choice is Keras [89], it has the second highest ANN user base in Python according to [55]. The first place is dominated by TensorFlow [90], but since Keras builds on TensorFlow's basic ANN operations (tensor manipulation) and is more user-friendly, it is considered advantageous for this thesis. Here, TensorFlow 1.15.3 and Keras 2.3.1 are used. Although newer versions exist, the open-source nature of Python can lead to newly released versions having compatibility issues. Additionally, most documentation is available for these older frameworks.

4.3. Artificial Neural Networks with LiDAR data

The aim of this work is to predict real numbers for R_O, φ_O and Γ of the two wake vortices in individual LiDAR scans. Manually computed values from the Radial Velocities method are used as the targets for the training of the ANNs. Suitable metrics and pre-processing for radial velocity LiDAR scans must be chosen.

4.3.1. Measures of success

Besides the loss function MSE, Keras facilitates the evaluation of further metrics. These are implemented analogous to the loss function, however their results are not used for updating weights. The Mean Absolute Error (MAE) is usually recommended as a metric for regression problems, giving superior interpretability of the results [84]. Already a small difference in, for instance the predicted φ_O compared to the target, can lead to an enormous relative error, when the MAE difference is nearly negligible. Mathematically, the MAE represents the average error of the ANN output compared to the target as shown with (4.11).

$$MAE = \frac{1}{N}\sum_{n=1}^{N}|\mathbf{y}_n - \mathbf{t}_n| \tag{4.11}$$

The 'Absolute Distance Error (ADE)' is defined with (4.12) and represents the Euclidean distance between the predicted wake vortex origin to the target wake vortex origin in the Cartesian coordinate system (subscripts p and t, respectively). It gives an overall key-point localisation error, allowing the comparison of different coordinate systems. There are real and virtual ADEs. The latter is utilised when different data sets are employed for the φ_O and R_O predictions. Thus, fake elevation-range pairs are built to compare the vortex origin localisation. These associate elevation angles and ranges to vortex origins which in reality do not exist. The approach is representative as it is performed for both the targets and the predictions.

$$ADE = \sqrt{(y_{O_t} - y_{O_p})^2 + (z_{O_t} - z_{O_p})^2} \tag{4.12}$$

4.3.2. Validation techniques

In order to validate the results of an ANN, several data sets are required [91]. These data sets are: 'train', 'validate' and 'test'. The train data is used for the learning phase of the ANN models, while the validate data set is used for evaluating the trained models on previously unused LiDAR scans. This reveals the real performance of the networks - the train data set normally performs superior, as the models are optimised towards it. Hence, overfitting towards the train data must be avoided. The validate data set is usually utilised in the optimisation of network architectures. Therefore, many optimisation cycles can also lead to overfitting towards the validate data [55]. In these cases, the test data set is used to confirm the errors retrieved by the validate data.

Data sets are constructed such that they contain a variety of LiDAR scans. Scans are randomly selected, regardless of individual overflights. No universal set sizes can be given for this project, as studies use different sizes for computational efficiency and data availability reasons. Nevertheless, across-the-board train data sets are roughly ten times the size of the validate and test data sets. It should be mentioned that different data sets may result in lower, but more likely, in higher errors. Some overfitting of ANNs is unavoidable and the current data set underrepresents some aircraft categories, LiDAR positions or similar.

Further reasons for uncertainty in the obtained results are [92]: the stochastic nature of the optimisation algorithm, the evaluation method in Keras, the randomised weight initialisation and the data set shuffling during gradient descent. To deal with this common ANN problem, the mean errors of multiple attempts are given. These result from three separately trained networks on the same data set, yet randomly chosen scans each time. The mean errors give an overview of the general ANN performance, rather than the performance of a single data set. When larger data sets are employed, the variation is automatically reduced. Performing the process once is considered sufficient in these cases.

Section 4.4 focuses on the ANN architectures, where also techniques for avoiding overfitting - regularisation methods - are discussed and evaluated. To make sure the best models are used for evaluation, a less elegant version of regularisation is also implemented in Keras: a callback. This allows a model to stop training once the loss of the validate data set no longer decreases. Within all epochs, the model with the lowest validate loss is saved. This selected model guarantees better generalisation on new, previously unused, LiDAR scans. Normally, the callback uses the train loss, such that the saved model is not performing well only on the current validate data, however as previous implementations show, the thesis' approach is common when large and representative enough validate data sets are used [84, 92]. In fact, using the train data set for early stopping can mean that the validate data metrics already show significant overfitting - the train data is not representative. Unless otherwise stated, all tables and figures printed in this report state the metric values for a validate data set and the epoch which gives the lowest validate loss.

4.3.3. LiDAR scan preparation and normalisation

LiDAR data from the campaign is stored in three one-dimensional arrays, one for the range, the elevation angle and the radial velocities at those measurement points. MLPs and CNNs require different input data shapes as introduced in Section 4.2.3. Since MLPs see all radial velocities as individual features, each scan remains in this one-dimensional row vector, but sorted for increasing elevation angles, and within each LOS for increasing ranges. The scans in one data set are then concatenated into a matrix, where one row entails one scan. CNNs require each scan to be represented in three dimensions. With multiple scans, this requires a four-dimensional tensor as follows: the first dimension separates the scans, the second and third dimensions are radial velocities - stored analogous to an image - with elevation angles in descending order and within each LOS, ascending ranges. The last dimension makes up the number of channels. The targets are stored equivalently for both ANN types, in a separate column vector - one target for each scan.

ANNs perform superior when scans contain radial velocities of similar (low) magnitude [55]. Otherwise, inputs with a great variance may incorrectly dominate inputs with a smaller variance [72]. Two normalisation techniques may deal with this: scan-wise and feature-wise [55]. Both aim for a radial velocity mean of zero and standard deviation of one within the train data set part considered, for low magnitude radial velocities and limited variation [55]. Resulting normalisation factors are also applied to the validate data set.

Feature-wise normalisation separately normalises across each point in the measurement grid, but regarding all scans in the train data set. It is a common approach for MLPs, as they normally contain inputs of different types. Therefore, feature-wise normalisation might distort the scan velocity distributions, leading to the consideration of employing scan-wise normalisation instead - jointly considering all radial velocities of the entire train data set, regardless of where they are located in a scan. ANNs may understand unphysical relationships of feature-wise normalised scans, which is why proxy data is generated to investigate the superior normalisation method. The comparison is implemented using a simple ANN with six outputs - one for each parameter. The exact architecture is of little relevance here, as all normalisation techniques use the same one. The results are tabulated in Table 4.3, also showing the option of no normalisation.

Table 4.3.: Validate errors comparing normalisation techniques with proxy data of 1000 train & 100 validate scans using MLPs (blue = the lowest parameter error).

Normalisation	Γ MAE (m^2/s)		φ_O MAE (degrees)		R_O MAE (m)	
	CW	CCW	CW	CCW	CW	CCW
None	14.33	12.27	2.83	2.60	49.95	13.69
Scan-wise	14.36	12.03	2.35	1.64	42.10	9.67
Feature-wise	15.25	19.50	1.55	1.59	17.28	6.28

Blue cells in Table 4.3 reveal that the lowest errors are obtained using feature-wise normalisation. Upon closer inspection, Table 4.3 indicates that this is true mainly for the localisation parameters, while the circulation is more accurately predicted using scan-wise and no normalisation. Feature-wise normalisation transforms scans into another dimension, which could lead to scan relevant information - in terms of vortex patterns - being lost. Additionally, scan-wise normalisation shows favourable behaviour when applying feature engineering (see Section 5.1). Regardless, CNNs are known to perform superior with scan-wise normalisation and therefore it is used for CNNs throughout this thesis [55]. The normalisation approach for MLPs is the following: due to the results from Table 4.3, feature-wise normalisation is applied to achieve the highest precision. Scan-wise normalisation is utilised when comparisons to CNNs are desired.

A further preparation deals with the individual faulty radial velocity measurements from LiDAR. For simplicity, these are set to zero. While this might add small gradients to the scans, these errors are rare and consequently unlikely to disrupt wake vortex characterisation. A description of crucial Python ANN scripts is given in Appendix F. Furthermore, a complete overview of an example pre-processing and Python implementation is given in Appendix G.

4.4. Artificial Neural Network architectures

This section aims to build suitable architectures for MLPs and CNNs focused on LiDAR scan evaluation. As discussed in Section 3.3, CNNs are assumed to be more suitable for evaluating LiDAR scans. Nonetheless, MLPs are necessary to understand the influence of hyperparameters and could prove to be sufficient. Hyperparameters are all variables other than trainable parameters such as weights and bias. For instance, the filter size of convolutional layers is a hyperparameter. Additionally, training MLPs is dramatically faster (for an example see Section 5.2.3), allowing to investigate more feature engineering. First, the MLP architecture is found by specifying the fundamental hyperparameters, and subsequently the CNN architecture is developed by adjusting its unique hyperparameters, but retaining settings from the MLP architecture. The proxy data, described in Section 4.1.2, is used to obtain the network architectures. To recapitulate, this is done to see the effects of the architecture more clearly, avoiding the effects from noise, SVSs and atmospheric turbulence that are present in the measurement data.

Finding appropriate network architectures is a trial and error process. As a consequence, this text does not aim to find ideal architectures, but to provide trends as a starting point for future optimisation. Literature often uses Bayesian optimisation, which trains many different ANNs with hyperparameters chosen at random (within appropriate bounds) to speed up optimisation, however the computational time still goes beyond the scope of this thesis [92]. The alternative approach taken here is as follows: the architectures recommended in [55] are used as a starting point and hyperparameters are varied above and below their suggested value.

Architectures which prove beneficial for the proxy data, might not be ideal for the measurement data. However, high velocity field variations of the measurement data already lead to fluctuations in the detection performance of the networks. This obscures the effects of different architectures, preventing measurement data from being used in the hyperparameter studies. It is expected that there are enough similarities between the two data sets (velocity magnitudes, gradients and data sparseness) to give an indication of a network architecture which might be suitable for measured LiDAR radial velocity scans.

The main goal, directly after generally obtaining low errors, is to avoid overfitting of the ANNs to the train data set, where the features learned are not general enough and the focus is on the unique features of the train data. Consequently, this section also briefly investigates some regularisation techniques. The techniques explored here are dropout, batch normalisation and weight regularisation:

- **Dropout** [93] sets a percentage of randomly selected weights to zero during each epoch, defined by the dropout rate. When using dropout, ANN training makes use of more features. Trained models should be more generalisable to new data, even when salient features are not present. The primary problem with dropout is that in certain cases too much significant information is removed, alleviating network performance, complicating the interpretability of results and in turn reducing the model's generalisability [86]. Nevertheless, dropout is often praised to be highly effective [93].
- **Batch normalisation** [94] is data normalisation throughout ANN training. The initial data normalisation is normally only effective for the first network layer, but data variation is also not desirable throughout the rest of training [94]. Batch normalisation ensures that the standard deviation and the mean of a mini-batch are consistent throughout the entire ANN training [95].
- **Weight regularisation** limits the magnitude of weights during training. Therefore, weights are distributed more evenly, resulting in a similar effect as dropout: the final model output is not dominated by a single feature [55]. This regularisation is implemented by adding a cost (higher loss) for having large weights to the loss function [55].

The following investigations primarily focus on predicting the circulation of the clockwise wake vortex. When the single parameter is not sufficient for identifying a trend, the remaining parameters are consulted. After all, this approach is not too limiting: the scans are the same for all parameters, with the targets changing only. In other words, the information is the same, but different input-output relationships must be found.

ANN tuning is performed using 1000 train and 100 validate proxy scans, all data sets are randomly shuffled for each training case. The employed data set sizes are a compromise between computational efficiency and accuracy.

4.4.1. Multilayer Perceptron

Fundamental hyperparameter tuning is described in this section. The considered hyperparameters are shown in Table 4.4. All possible settings of one hyperparameter are investigated for all possible settings of the other hyperparameters, leading to 54 MLPs. The final settings are also given in the table. Dropout is added between all layers. It has been shown to be beneficial compared to sparse usage as described in [93]. Adaptive Moment Estimation (ADAM) [96] and Root Mean Square Propagation (RMSProp) [97] are chosen as the two optimiser options. Comparisons of popular optimisers identify them to be among the most competent, due to incorporating adaptive learning rates [81, 84]. The majority of the literature investigated for this thesis makes use of them. Mathematics of the different optimisers can be found in [81]. The number of dense layers is limited since extremely deep networks, with a greater number of parameters, tend to more easily overfit. Lastly, the early training stopping is set to 20 epochs throughout this investigation.

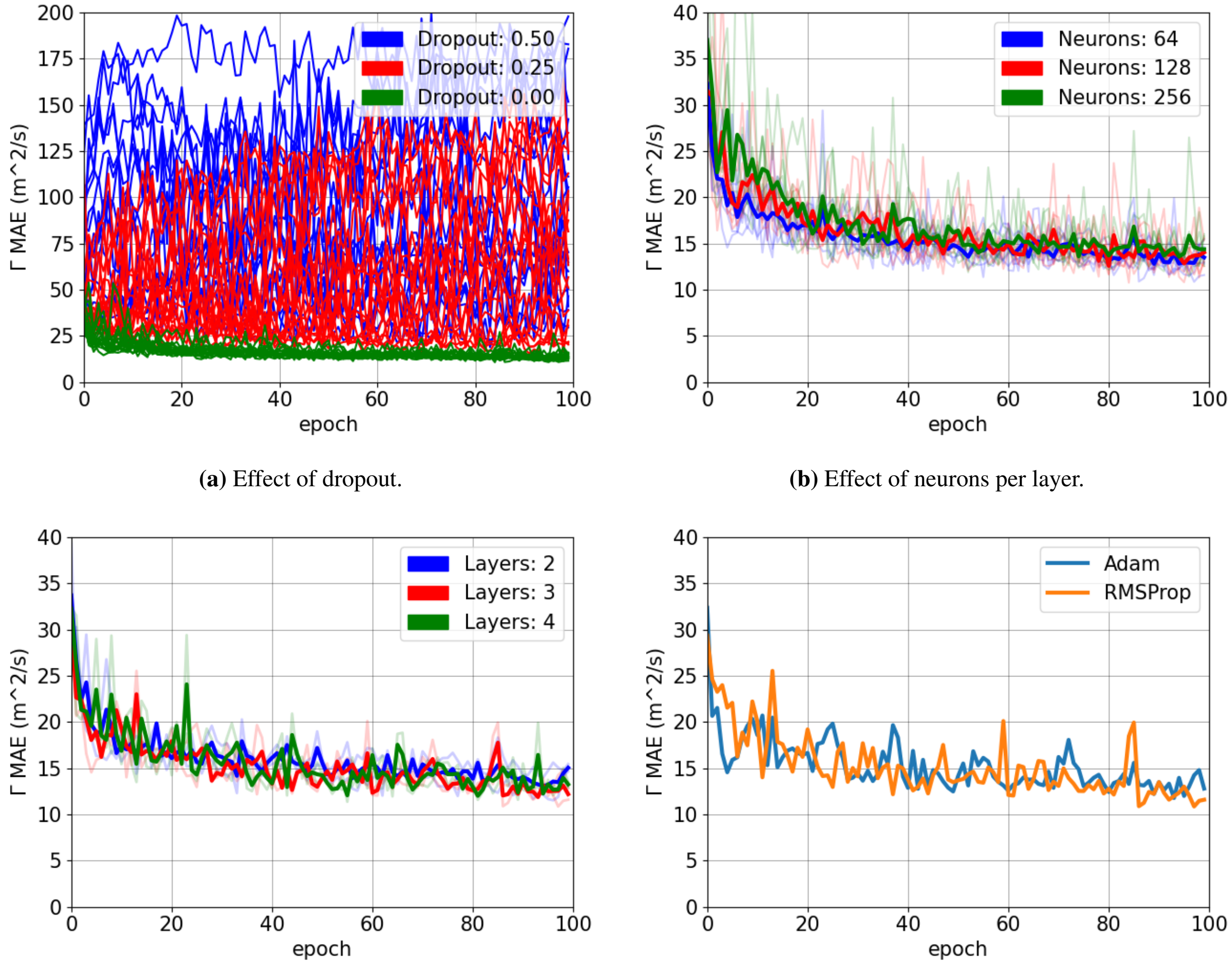
Table 4.4.: Investigated MLP hyperparameters with their possible and selected settings.

Hyperparameter	Possible settings	Chosen setting
Hidden dense layers	2, 3 and 4	3
Neurons per dense layer	64, 128 and 256	64
Dropout rate	0.00, 0.25 and 0.50	0.00
Optimiser	RMSProp and ADAM	ADAM

Figure 4.10 illustrates the training history of all considered hyperparameter combinations, grouping histories featuring the same hyperparameter studied in that figure, and consequently the robustness of model convergence. The vertical axis represents the Γ MAE of the clockwise vortex, while the horizontal axis is the epoch count. One epoch signifies that the entire train data set has been considered once for updating weights. As a consequence, with each epoch Γ MAE should decrease. Normally, training progress is depicted via the train data. Instead in the present case, the validate data metrics are computed after each epoch, facilitating to recognise overfitting towards the train data and giving more realistic model performance. The figures are in the order of how hyperparameters have been selected, such that Figure 4.10a includes more plotted histories than Figure 4.10d. In other words, once a hyperparameter choice has been made in the previous figure, the next figure only shows relevant architectures with that hyperparameter setting.

Figure 4.10a shows that dropout is unusable. Rather than model overfitting, the problem seems to be training divergence or the lack of model convergence. Certain features in the scans appear to be vital for vortex detection. While normally dropout aims to prevent this, it seems that these features are present in all scans. Therefore, dropout is not required for this work's application. Weight normalisation and batch normalisation give similar results. Thus, none of the regularisation techniques are used, as they promote overfitting and limit the vortex characterisation performance.

The decision of how many neurons should be used is less obvious, however Figure 4.10b shows that the lowest errors can be obtained with 64 neurons per layer (blue-coloured line). For clarity, the moving averages of the observed histories are plotted in the foreground. Fewer spikes towards high errors are observed with this setting. Furthermore, the lower neuron count results in faster network training and a lower risk of overfitting. Consequently, 64 neurons are recognised as appropriate in a dense layer. In fact, closer inspection reveals that 256 neurons perform the worst, confirming the trend that lower neuron numbers are beneficial.



(a) Effect of dropout.

(b) Effect of neurons per layer.

(c) Effect of layer amount.

(d) Effect of optimiser.

Figure 4.10.: MLP hyperparameter comparison with validate proxy data & the clockwise vortex circulation.

Thereafter, the number of hidden layers are selected. Again, the foreground of Figure 4.10c illustrates the moving averages. While 3 and 4 hidden layers perform similarly and there is no trend based on the data available, the MLP with 2 layers cannot reach the same low error magnitude, leading to 2 layers not being considered further. The two best performing architectures, based on the overall lowest Γ MAE and amount of epochs with low Γ MAEs (seen in Figure 4.10d), contain 3 hidden layers, thus this option is chosen.

The two best performing MLPs are set apart by their optimiser. Their performance is rather similar, although towards later epochs the RMSProp optimiser leads to lower Γ MAEs. Nevertheless, ADAM is chosen, since it takes $123\,\mathrm{s}$ to reach this result, while RMSprop requires almost twice that time ($235\,\mathrm{s}$). Hence, ADAM allows training twice as many networks in the feature engineering part of this report, facilitating more significant performance improvements.

One further setting utilised for the remainder of this thesis, is the early training stop callback. It is increased to a more appropriate 30 epochs, due to the heavy fluctuation in network training - using 20 epochs occasionally cut off training too early. The final MLP network architecture is modelled as shown in Figure 4.11. The input layer is not drawn, as it does not contain any neurons and simply consists of various elevation-range radial velocities. The architecture is built to predict one parameter of a wake vortex, so-called scalar regression. The total number of trainable parameters depends on the size of each scan. In the case of a proxy scan with 91 range gates for 151 LOS (13 741 radial velocities), the number is 887 873, resulting directly from the sum of all parameters calculated via (4.9).

LAYER 1: DENSE LAYER
Activation: ReLu
Input Shape: Range Gates $\times$ Elevation Angles
Output Shape: 64

$\downarrow$

LAYER 2: DENSE LAYER
Activation: ReLu
Output Shape: 64

$\downarrow$

LAYER 3: DENSE LAYER
Activation: ReLu
Output Shape: 64

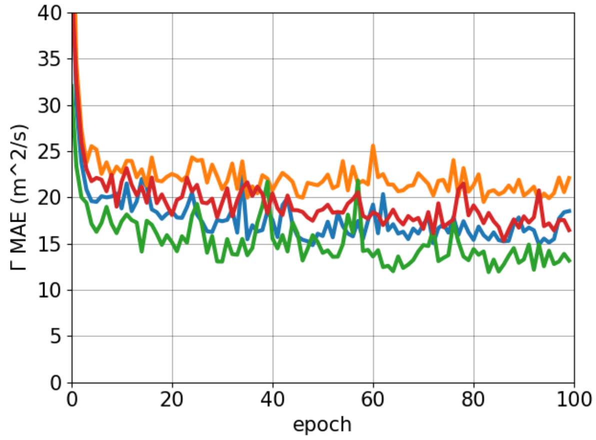
$\downarrow$

LAYER 4: DENSE LAYER
Activation: None
Output Shape: 1

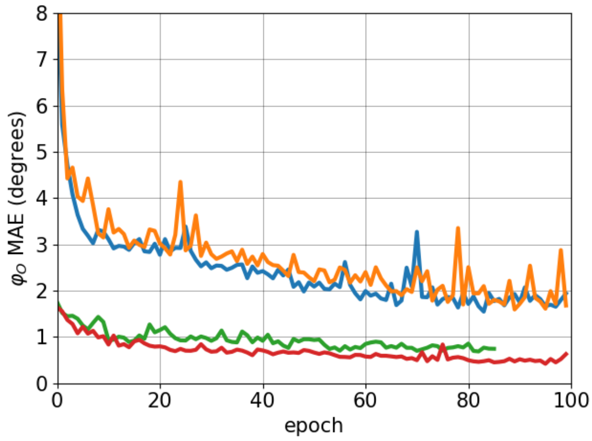
Figure 4.11.: Scalar regression MLP architecture.

This ANN predicts one parameter of the six desired: the range and elevation angle to the vortex origin and the circulation of each vortex. ANNs also have the ability to output multiple parameters, so-called vector regression in the case of a regressive problem. In practice, vector regression could be faster by sharing features. For comparison, the vector architecture is equivalent to that of the scalar regression network, with the addition of five more layers in the style of Layer 4 emerging from Layer 3 in Figure 4.11 - the last hidden layer. The entire vector regression architecture is shown in Appendix D.

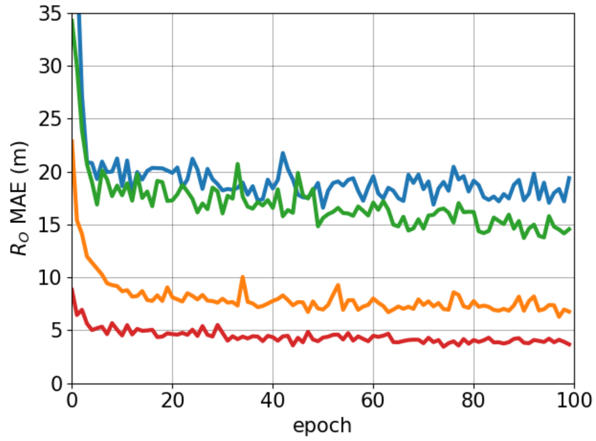
While the training and computation time for this vector network is comparable to a single scalar network, Figure 4.12 illustrates that its performance is poor. Differences are not significant for all parameters, however for the elevation angle predictions, the errors of the scalar networks are substantially lower than vector network errors of either vortex. For the other two parameter categories, the scalar error usually only outperforms the vector error of the same vortex. Still, conversely to arguments in [88], the independent scalar networks perform superior to the vector regression network. This indicates that the patterns learned from the LiDAR scans are not necessarily the same for each parameter and hence are not reused within the vector network. As a consequence, the remainder of this thesis utilises separate scalar networks.



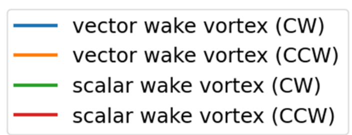
(a) Circulation.



(b) Elevation angle.



(c) Range.



(d) Compared ANNs.

Figure 4.12.: Comparison of vector and scalar regression for MLPs with validate proxy data.

In order to verify the results obtained with this network architecture and validate data set, a test data set is utilised - this set is made up of 100 new proxy scans. The tabulated values in Table 4.5 indicate that the circulation estimation is slightly worse, while the localisation of wake vortex origins can even be superior with new scans. Overall, the network architecture used for the MLPs is general enough for the desired application. Thus, the test data set is no longer applied for this architecture's verification of the validate data set - quoting errors of the validate data set is sufficient.

Table 4.5.: Validate and test errors confirming the MLP architecture with proxy data of 100 validate & 100 test scans and feature-wise normalisation (blue = the lowest parameter error).

Data set	Γ MAE (m^2/s)		φ_O MAE (degrees)		R_O MAE (m)	
	CW	CCW	CW	CCW	CW	CCW
Validate	11.92	15.30	0.70	0.42	13.94	3.45
Test	15.74	17.65	0.83	0.48	11.43	3.63

4.4.2. Convolutional Neural Network

Building an appropriate CNN architecture is more involved. Before choosing hyperparameter settings, it must be considered that for CNNs the LiDAR scan coordinate system can affect the network performance in a significant manner. Previous discussions in Section 4.2.3 hinted the differences in ANN layers and the superior capability of contextual understanding with convolutional layers. Additionally, the raw LiDAR scans are of polar coordinate system type. However if a photo were taken of the flow behind an aircraft, it would be in the Cartesian coordinate system. In the latter case, regardless of the distance to the LiDAR, the flow would not be stretched or distorted. It must be evaluated which of the two coordinate systems performs superior. While the polar coordinate system would be preferred to avoid extensive pre-processing and obtain faster predictions, it is able to distort velocity fields. The format of scans in the polar coordinate system versus the Cartesian coordinate system has been previously shown in Figure 4.2 - scans in the polar coordinate system offer homogeneous grid point distributions, while scans in the Cartesian coordinate system become sparsely populated with larger ranges from the LiDAR.

One solution for Cartesian coordinate system scans to feature homogeneous grid point distributions, would be the interpolation onto a grid with preset measurement locations where radial velocities can be located. However, the selection of a grid spacing would be far from straightforward. Figure 4.2d aids in visualising the following arguments. On the one end, if a coarse grid is chosen, too much information at lower ranges is lost. On the other end, a grid with minimal spacing leads to scans of enormous resolution, without the gain of additional information at larger ranges. This would slow down learning and predictions.

In order to avoid this trade-off, proxy wake vortex scans, which do not replicate the structure of a classical LiDAR scan, are created. They do not have LOS restrictions and a Cartesian measurement grid can be employed in the Cartesian, rather than the polar coordinate system. Its size is chosen to be of equivalent shape to the polar coordinate system proxy scan, in terms of measurement points, range spectrum and elevation angle spectrum. This approach is not applicable for measurement data. Different measurement points for both scan types also lead to different input radial velocity data, however this bypasses the question of what interpolation is most suitable. It must be noted that this approach leads to a lower scan resolution at lower ranges than would be seen with a measured LiDAR scan, potentially limiting the location prediction capabilities for this data. Nonetheless, it is assumed that the results from the following study are representative, as the circulation errors should decrease enough for this coordinate transformation to be justifiable in the first place. Clearly, to move from polar to Cartesian coordinate system scans, the latter must deliver a significant benefit compared to the polar coordinate system scans, as ANNs are known to perform worse with sparse data and interpolation only degrades the quality of the scans.

For the results of Table 4.6, the focus is not on the CNN architecture, however both coordinate systems use the same one. The results in Table 4.6 emerge from a new set of 1000 train and 100 validate scans. Only comparable location parameters are given, as polar coordinate networks use the parameters R_O and φ_O, while Cartesian coordinate networks localise vortices using the parameters y_O and z_O. The tabulated results partly confirm the expected trends. Only for the clockwise vortex circulation, Cartesian coordinate system scans show lower errors. All other errors are lower for polar coordinate system scans. Hence, the additional effort to transform scans to the Cartesian coordinate system, potentially degrading location prediction accuracy and requiring longer training and characterisation times, is not justifiable. As a result, the polar coordinate system is also chosen for LiDAR scans fed to CNNs.

Table 4.6.: Validate errors comparing scan coordinate systems with proxy data of 1000 train & 100 validate scans using CNNs and scan-wise normalisation (blue = the lowest parameter error).

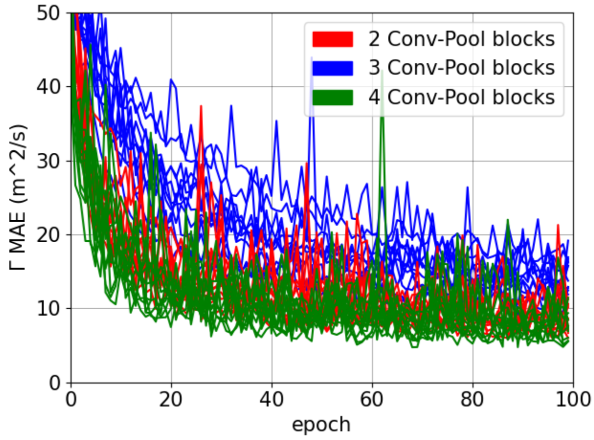
Coordinate system	Γ MAE (m^2/s)		Mean real ADE (m)	
	CW	CCW	CW	CCW
Polar	5.58	3.66	1.51	0.85
Cartesian	3.72	4.36	1.86	2.80

Table 4.7 summarises the hyperparameters investigated in this section. Moreover, to prevent differences in parameter count due to the flatten layer, the different CNN architectures all include an appropriate number of max pooling layers prior to the flatten layer. This ensures that the two-dimensional input shape to the flatten layer is equivalent in all cases, the best performing data shape is found to be $(2, 4)$.

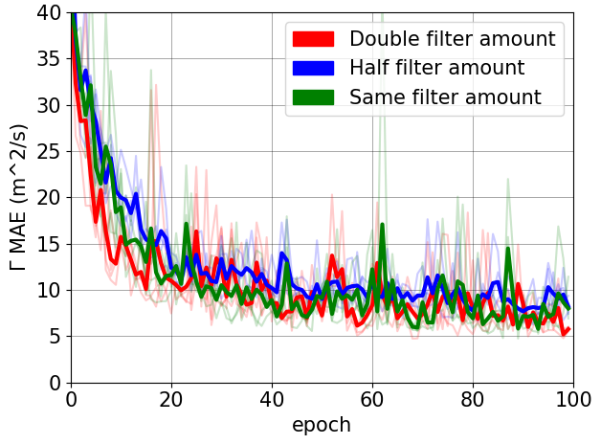
Table 4.7.: Investigated CNN hyperparameters with their possible and selected settings.

Hyperparameter	Possible settings	Chosen setting
Convolution-Pooling Blocks	2, 3 and 4	4
Convolutional layer filter amount pattern	Same, double and half	Double
Filter in first layer	16, 32 and 64	32
Dense layers	1 and 2	1

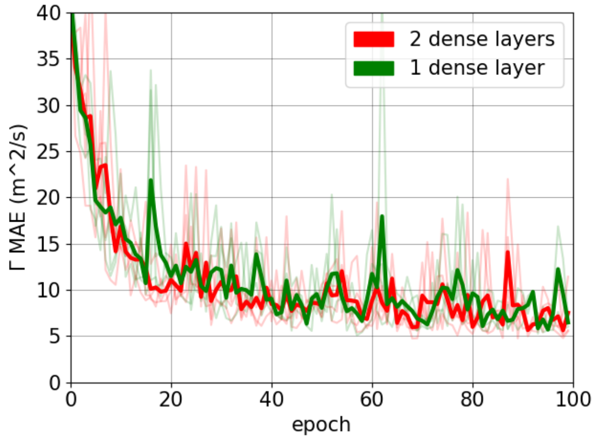
The histories of the various CNNs are plotted in Figure 4.13. Once again the figures are in a logical order, with later figures only investigating relevant architectures based on decisions made within previous hyperparameter studies. Moreover, where appropriate moving averages are used in the foreground.



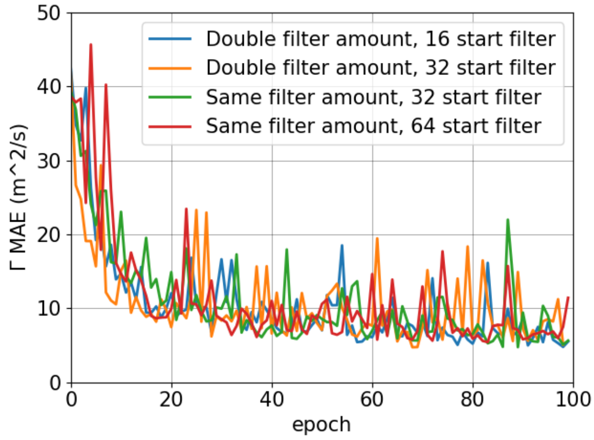
(a) Effect of CPBs.



(b) Effect of convolutional filter pattern.



(c) Effect of dense layers.



(d) Effect of first convolutional filter amount.

Figure 4.13.: CNN hyperparameter comparison with validate proxy data & the clockwise vortex circulation.

First, the number of CPBs is selected: the more blocks are employed, the better complex patterns can be identified in the scans, leading to the lowest Γ MAE with 4 blocks as depicted in Figure 4.13a. Additional CPBs are unlikely to provide a significant performance boost, given the convergence trend in the plot.

Next, the focus is on the convolutional filter pattern. According to literature, the convolutional filter amount needs to be adapted to the network depth. It is investigated whether the amount of filters should increase (‘double’) or decrease (‘half’) with rising network depth. Both approaches have theoretical explanations: on the one hand, doubling filters is supposed to counteract the decreasing parameter count of pooling layers, giving enough learnable weights to find input-output relations. On the other hand, decreasing the amount of filters could allow to focus on salient features in a LiDAR scan and therefore generalise better. As a baseline, constant filter numbers (‘same’) are also evaluated. Figure 4.13b reveals that using half the filter amount with increasing network depth leads to higher Γ MAEs. Despite that, the figure lacks significant performance differences. As a result, the two patterns ‘double’ and ‘same’ are kept in the other hyperparameter studies to see whether they perform superior with a specific pattern.

Thereafter, the number of dense layers following the flatten layer is chosen. In Figure 4.13c both 1 and 2 dense layers deliver similar errors. For computational efficiency reasons, 1 hidden dense layer is chosen.

The final hyperparameter study considers the amount of filters used in the first convolutional layer, given a specific filter pattern from above. Circulation error progression with epochs does not reveal a distinctive trend in Figure 4.13d. Nonetheless, the two best networks can be identified to be the ‘double’ filter case with 32 filters and the ‘same’ filter pattern with 64 filters. These two settings are chosen given the fewer epochs they require to reach low Γ MAEs.

In order to select the leading CNN architecture for all characterisation parameters, the remaining parameters are also trained and the network architecture with lower MAE is noted for each parameter separately. A complied tally reveals that the pattern ‘double’ with 32 convolution starting filters is the superior architecture for twice as many parameters as the pattern ‘same’ with 64 convolution starting filters. The resulting architecture for CNNs is modelled as shown in Figure 4.14. For clarity of layer input and output shapes, the discrete case with 91 elevation angles and 151 ranges is chosen - the scan size of proxy scans (the same as used for counting the trainable parameters of MLPs). The total number of trainable parameters is 519 041 in this case (computed using (4.9) and (4.10)) - nearly half that of the MLP, making overfitting less probable.

The CNN architecture developed with the train and validate data sets is verified with a test data set. This set is equivalent to the one used in Section 4.4.1, the results are tabulated in Table 4.8. The same trend as for MLPs is seen - the circulation estimation is slightly worse, while the localisation is superior. Therefore, the same stochastic fluctuations as discussed previously are assumed, and from now on solely the validate data set is used for the CNN architecture. Verification via the test data set is no longer required.

Table 4.8.: Validate and test errors confirming the CNN architecture with proxy data of 100 validate & 100 test scans and scan-wise normalisation (blue = the lowest parameter error).

Data set	Γ MAE (m^2/s)		φ_O MAE (degrees)		R_O MAE (m)	
	CW	CCW	CW	CCW	CW	CCW
Validate	5.72	5.14	0.19	0.18	2.11	1.27
Test	7.21	6.68	0.22	0.14	1.69	0.81

4.5. Handling of absent vortices

While the proxy data used for developing the ANN architectures always has two vortices for every scan, the measurement data in some cases contains only one vortex - the other vortex has already decayed or was transported out of the measurement region prior to the LiDAR measurement. In reality, LiDAR scans may also contain no wake vortex at all. However for the purpose of this work, only scans with at least one wake vortex are utilised, as the currently used methodology for selecting wake vortex scans is already automated, unlike their characterisation [46].

LAYER 1: CONVOLUTIONAL LAYER
Activation: ReLu | Input Shape: (91, 151, 1)
Output Shape: (91, 151, 32)

LAYER 2: MAX POOLING LAYER
Output Shape: (45, 75, 32)

LAYER 3: CONVOLUTIONAL LAYER
Activation: ReLu | Output Shape: (45, 75, 64)

LAYER 4: MAX POOLING LAYER
Output Shape: (22, 37, 64)

LAYER 5: CONVOLUTIONAL LAYER
Activation: ReLu | Output Shape: (22, 37, 128)

LAYER 6: MAX POOLING LAYER
Output Shape: (11, 18, 128)

LAYER 7: CONVOLUTIONAL LAYER
Activation: ReLu | Output Shape: (11, 18, 256)

LAYER 8: MAX POOLING LAYER
Output Shape: (5, 9, 256)

LAYER 9: MAX POOLING LAYER
Output Shape: (2, 4, 256)

LAYER 10: FLATTEN LAYER
Output Shape: 2048

LAYER 11: DENSE LAYER
Activation: ReLu | Output Shape: 64

LAYER 12: DENSE LAYER
Activation: None | Output Shape: 1

Figure 4.14.: Scalar regression CNN architecture.

The flow pattern of a single vortex is quite different to that of a pair [19], but at the same time it is often the case that the radial velocities in the measured scans are not significantly different, as the missing vortex might just be slightly outside the detection window. Consequently, its existence still affects the detectable velocity field by the LiDAR. Nevertheless, to evaluate the handling of absent vortices, velocity fields with only one vortex are initialised. This is considered acceptable, as outside a vortex core velocities decay with r^{-1} (see (2.4) and (2.5)).

Circulation and location targets of absent wake vortices are set to zero - a common approach for unavailable data with ANNs, as they have the ability to learn that this value is associated with vortices indistinguishable from atmospheric turbulence [55]. An unavailable vortex has no strength, however setting localisation parameters to the origin could create a bias in the data, shifting all predictions towards the origin of the coordinate system. The study is conducted with 1000 train and 100 validate scans, of which half initialise both wake vortices and the other half only initialises the counterclockwise wake vortex. Table 4.9 summarises the results for the one/two vortices case in comparison to always having two vortices using MLPs.

Table 4.9.: Validate errors evaluating how to handle absent vortices with proxy data of 1000 train & 100 validate scans using MLPs and feature-wise normalisation (blue = the lowest parameter error, yellow = mixed results).

Vortices	Γ MAE (m^2/s)		φ_O MAE (degrees)		R_O MAE (m)		Mean/median real ADE (m)	
	CW	CCW	CW	CCW	CW	CCW	CW	CCW
Two	11.92	15.30	0.70	0.42	13.94	3.45	15.38/13.60	3.83/2.39
One/two	7.17	14.02	0.84	0.43	18.60	3.80	18.82/2.41	4.17/2.39

As expected, the addition of another variable - the amount of wake vortices - increases the difficulty of prediction. Several behaviours can be recognised:

- The non-existing wake vortex is often predicted to be located at slightly negative ranges and elevation angles.
- It is common for networks with mixed vortex amounts to halt training after fewer epochs, indicating heavy error fluctuation.
- The circulation prediction is superior with mixed scans, while the localisation accuracy degrades.
- The median ADE sees an improvement of one order of magnitude for the clockwise vortex, which is not present in half of the scans. The median operation orders error magnitudes, potentially hiding higher errors, which could be caused by absent vortices.

Given that only a slight performance degradation can be observed, it is considered a suitable approach to set the parameters of an absent vortex to zero. Alternative strategies would be to set the parameters of an absent vortex closer to the middle of the scan to avoid a heavy bias, however in that case it would be challenging to distinguish an absent vortex from a present one.

4.6. Precision limits

This section uses proxy data results and theoretical characterisation errors from the RV method, which produces the targets, to set precision limits for the measurement data. Any result achieved with the measurement data is unable to be more precise than the targets it is trained with. Furthermore, the ideal nature of the proxy scans makes it unlikely for the measurement data to obtain smaller errors. As a consequence, the errors from the proxy data set with one or two vortices per scan, tabulated in Table 4.10, and the target errors from Section 3.2 are summed, giving the precision limits as shown in Table 4.11. To recapitulate, targets with a conservative CNR of $-10\,\mathrm{dB}$ give errors of $10.3\,\mathrm{m}^2/\mathrm{s}$, $0.21°$ and $1.8\,\mathrm{m}$ for Γ, φ_O and R_O, respectively [51]. In case prediction errors of the measurement data are nearly as low as the precision limits, it is clear that no further feature engineering or data optimisation is required.

Table 4.10.: Validate errors for proxy data with one/two vortices of 1000 train & 100 validate scans using MLPs & CNNs and feature/scan-wise normalisation, MLP values are from Table 4.9.

Network	Γ MAE (m^2/s)		φ_O MAE (degrees)		R_O MAE (m)	
	CW	CCW	CW	CCW	CW	CCW
MLP	7.17	14.02	0.84	0.43	18.60	3.80
CNN	5.46	6.75	0.11	0.11	2.47	1.23

Table 4.11.: Precision limits for the measurement LiDAR data using the target errors and proxy data of 1000 train & 100 validate scans using MLPs & CNNs.

Network	Γ MAE (m^2/s)		φ_O MAE (degrees)		R_O MAE (m)	
	CW	CCW	CW	CCW	CW	CCW
MLP	17.47	24.32	1.05	0.64	20.40	5.60
CNN	15.76	17.05	0.32	0.32	4.27	3.03

5. Results

5.1. Feature engineering

Data pre-processing can be investigated, it may improve the prediction accuracy when using measured LiDAR scans. Accordingly, the present section focuses on the data from the Vienna measurement campaign. The first studies are performed using overflights with flat plate lines, LiDAR position L3 and MLPs, since they are faster to train than CNNs. Some of the below studies are completed using other LiDAR positions, plate settings and ANN types - the reader is made aware of these cases in the relevant sections.

5.1.1. Crosswind effects

Besides the primary wake vortices, the velocity fields captured by LiDAR may also entail a crosswind. It is expected that the characterisation of wake vortices is improved by focusing only on the velocity fields induced by the primary counter-rotating wake vortices. This gives the following question:

- How and to what extent can a crosswind in captured velocity fields be accounted for? -

This thesis takes an approach from [50]: for each overflight, a LiDAR scan - with no wake vortices - just before the aircraft crosses the measurement plane is selected. This so-called background scan models the crosswind likely to appear in the following wake vortex scans. Since there is a time delay between the two scans, this approach holds for constant crosswind. Its suitability is justified by noting that only small changes of wind were experienced throughout the campaign. The background scan is subsequently subtracted from the wake vortex scan by computing the background scan's mean radial velocity of each LOS. This value is then subtracted from the respective LOS of the wake vortex scan. To complete this, inconsistent discrete elevation angles of LiDAR require nearest-neighbour interpolation to be conducted, mapping the background scan onto the wake vortex scan grid.

Resulting average errors for the new scans are shown in Table 5.1. For fluctuation reasons, the model is separately trained multiple times and averaged thereafter - this is also done for the next feature engineering studies, until stated otherwise. Improvements with wind removal are marginal. Nonetheless, the majority of parameters reduce their errors. While the circulation is not conclusive, nearly all localisation parameters gain in precision. As a result, the conclusion of this study is to remove the wind from the wake vortex LiDAR scans via the proposed method.

Table 5.1.: Average validate errors evaluating wind removal with measurement L3 data of 1000 train & 100 validate scans using MLPs and feature-wise normalisation (blue = the lowest parameter errors).

Wind	Γ MAE (m^2/s)		φ_O MAE (degrees)		R_O MAE (m)		Mean/median virtual ADE (m)	
	CW	CCW	CW	CCW	CW	CCW	CW	CCW
Yes	41.99	36.08	1.74	1.13	49.01	60.82	55.63/37.58	65.25/46.73
No	36.60	43.16	1.68	1.28	47.37	56.96	48.85/35.46	59.80/40.76

5.1.2. Usage of ideal scan grids

The problem of each LiDAR scan not featuring the same elevation angles has been mentioned in Section 4.1.1. Furthermore, it is known that ANNs have some context understanding, giving the question:

- What is the effect of having scans with different LOS beam elevation angles? -

It is hypothesised that the context understanding of ANNs benefits from equivalent measurement points in all scans - an ideal grid. It removes variability from the data set. If improvements are achieved for MLPs, then the performance of CNNs should benefit even more, given their context understanding capabilities. As in Section 5.1.1, nearest-neighbour interpolation is utilised. The new discrete elevation angles correspond to the exact values from Table 4.1. This adds an additional LOS to each scan, as for some cases only the upper bound and for other cases only the lower elevation angle bound is included (depending on whether the LiDAR started scanning from the higher or the lower elevation angle).

Results are tabulated in Table 5.2, where the errors with no ideal grid are from Table 5.1. The advantage of ideal grids is unmistakable. For every parameter, the error reduces compared to using the individual grid of each scan. The counterclockwise wake vortex especially profits from this change. Since the counterclockwise vortex is further away from the LiDAR, the grid is more coarse here. Adapting the grid has a larger location influence in this region. Up until this implementation, this vortex performed worse than the clockwise one, however now it outperforms the clockwise vortex for almost all parameters. Given that MLPs only have a one-dimensional context understanding, the improvements are expected to be even greater for CNNs. As a consequence, the interpolation of LiDAR scans onto ideal grids is used for the remainder of this thesis.

5.1.3. Influences of plates and Secondary Vortex Structures

The description of the measurement data and the measurement campaign in Section 4.1.1 revealed that some overflights were conducted with erected and others with flat plate lines. In rare occasions not both, but only one plate line was in use. These cases are not considered in the following; they provide a special scenario. Instead, the focus is put on the effect of plates for the wake vortex characterisation with ANNs. The arising research question is:

- Does the ANN prediction performance vary for different plate states? -

Table 5.2.: Average validate errors evaluating ideal grids with measurement L3 data of 1000 train & 100 validate scans using MLPs and feature-wise normalisation (blue = the lowest parameter errors).

Ideal grid	Γ MAE (m^2/s) CW	Γ MAE (m^2/s) CCW	φ_O MAE (degrees) CW	φ_O MAE (degrees) CCW	R_O MAE (m) CW	R_O MAE (m) CCW	Mean/median virtual ADE (m) CW	Mean/median virtual ADE (m) CCW
No	36.60	43.16	1.68	1.28	47.37	56.96	48.85/35.46	59.80/40.76
Yes	30.53	32.91	1.56	0.96	44.31	39.03	48.58/28.49	42.09/29.03

It is expected that the increased number of flow perturbations due to erected plates, cause significantly more SVSs (as discussed in Section 2.1.5) and therefore problems for the ANNs to identify the primary vortices. Furthermore, circulation mitigation is primarily significant above the plate lines [13]. Consequently, although end effects travel back and forth the vortex filaments, the influence of plates is hypothesised to be larger for LiDAR in-line with the plates. Figure 4.1 showed that LiDAR position L3 is not in-line with a plate line. Accordingly, in addition to L3, this section also considers scans from L2. For both LiDAR positions, the plates down and up models are separately trained with appropriate scans. Note, models with position L3 only use 450 train scans and 80 validate scans, as not enough plates up overflights exist in the data set and for comparison the number of plates down overflights must be equivalent. Investigations with LiDAR position L2 employ the standard 1000 train and 100 validate scans.

Results in Tables 5.3 and 5.4 confirm that vortex detection is affected more heavily in regions close to the plates. Moreover, Table 5.3 shows large deviations between the plates up and down model for LiDAR position L2. While L3 models perform superior in the plates up case, L2 models prefer when the plates are down - potentially because hard targets block part of the measurement region. The former is against the hypothesis from above and raises a future research question: how does the LiDAR position and plate status impact the performance of ANNs? Regardless, given the differing performance depending on the plate state, it is investigated with LiDAR position L3, whether it is possible to scans regardless of the plate state and train solely a single model. The last row in Table 5.4 denies this approach; the worst performance is obtained with mixed scans. Intuitively this makes sense, the option of plates up or down leads to another big variance in possible scan velocity fields and hence makes it harder to generalise features. As a consequence, for the ongoing investigations, separate plates up and down ANN models are trained.

The above study for the two different plate states raises another question:

- Can SVSs and plate effects be reduced for primary wake vortex characteristics prediction? -

It is hypothesised that disregarding radial velocity measurement points close to the plates or the ground could improve predictions. It is known from ANN theory that common patterns recognised are usually a type of gradient [55]. Within the boundary layers, SVSs or also the plates can result in highly turbulent flow and more extreme gradients than within the primary vortices themselves. This is confirmed when inspecting

Table 5.3.: Average validate errors evaluating plate effects with measurement L2 data of 1000 train & 100 validate scans using MLPs and feature-wise normalisation (blue = the lowest parameter errors).

Plates	Γ MAE (m^2/s)		φ_O MAE (degrees)		R_O MAE (m)		Mean/median virtual ADE (m)	
	CW	CCW	CW	CCW	CW	CCW	CW	CCW
Down	33.61	34.80	1.71	1.10	40.82	64.45	43.22/29.03	66.60/44.83
Up	37.73	42.99	1.75	1.22	50.05	86.28	63.22/45.98	94.14/74.02

Table 5.4.: Average validate errors evaluating plate effects with measurement L3 data of 450 train & 80 validate scans using MLPs and feature-wise normalisation (blue = the lowest parameter errors).

Plates	Γ MAE (m^2/s)		φ_O MAE (degrees)		R_O MAE (m)		Mean/median virtual ADE (m)	
	CW	CCW	CW	CCW	CW	CCW	CW	CCW
Down	40.85	43.98	1.67	1.20	57.45	61.40	62.57/38.25	64.52/46.42
Up	32.34	36.01	1.36	0.98	27.15	46.69	34.09/26.25	50.73/35.09
Mix	40.55	47.78	1.93	1.09	60.45	76.73	64.79/39.86	77.62/51.63

LiDAR scans from all five positions - many faulty measurements are seen at low LOS. Consequently, with ANNs performing dimensional reduction and selecting salient features only, the activity at the lower LOS - where the boundary layers and plates are - can cause significant confusion for the ANNs. Disregarding these low LOS could allow ANNs to focus on the prediction of the primary wake vortices. Clearly, it is a crucial step to find an appropriate threshold LOS underneath which measurements are disregarded. The approach of this feature engineering is split into two steps, considering the plates and SVSs separately. A compromising threshold between the two is established thereafter.

First, the investigation focuses on solely the plate line geometry blocking the lower LOS. For clarification, the approach is sketched in Figure 5.1. In the case of the Vienna measurement campaign, only two LiDAR devices are in the risk of geometry blocking, L2 and L4. From Section 4.1.1 it is known that each upright plate has a height of $4.5\,\mathrm{m}$. Moreover, Figure 4.1 shows that the LiDAR positions L2 and L4 are $210\,\mathrm{m}$ and $164\,\mathrm{m}$ from the first plate of their plate line, respectively. Assuming that all LiDAR are at ground level, trigonometry reveals that the minimum LiDAR elevation angles to not hit the plates for positions L2 and L4 are $1.23°$ and $1.57°$, respectively. With these angles, the vertical heights at the runway centre are 6.01m and 6.69m. Following a conservative approach, $7\,\mathrm{m}$ is chosen as the critical height above the ground, underneath which the laser beam from a LiDAR may encounter a plate.

Second, the focus is on SVSs from the boundary layer. It is crucial to understand the typically experienced boundary layer thickness and core radius size of the primary vortices. The latter is required for not disregarding vortex cores. Unfortunately, the boundary layer thickness highly depends on the individual cir-

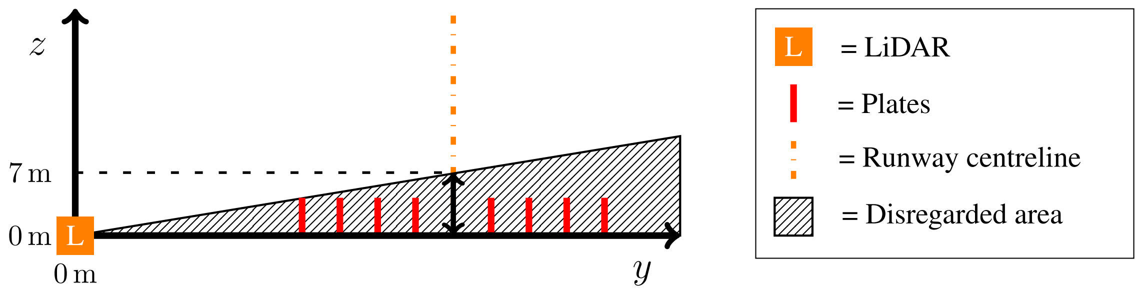


Figure 5.1.: Sketch of the disregarded LOS area.

cumstance. Nonetheless, an attempt is made to generalise it by considering the usual primary vortex altitude of $b_0/2$ (see Section 2.1.5), which is required for SVSs to detach from the boundary layer, and assuming an Airbus A320 to have an elliptical wing loading. The initial wake vortex separation is found from (2.3), which with the $34.1\,\mathrm{m}$ wingspan of an Airbus A320, is $26.8\,\mathrm{m}$ [15]. Given the above, SVSs detach when the primary vortices are at an altitude of $13.4\,\mathrm{m}$. Next, the focus is on the core radius. According to [14], the core radius can be obtained from $0.197(b/2)$. With the computed initial vortex separation, this gives $2.6\,\mathrm{m}$. Consequently, when the primary vortices are at an altitude of $13.4\,\mathrm{m}$ and one does not want to disregard the core radius, the threshold altitude should be $10.8\,\mathrm{m}$.

Combining the results of both the plates and SVSs, leads to using a $7\,\mathrm{m}$ threshold. At this altitude, the plates are ignored, but the vortex cores are not cut off. Primary vortices may hover at lower altitudes, however the rebound and stalling effects from Section 2.1.5 indicate that this is rare. Moreover, SVSs may reach altitudes higher than the $7\,\mathrm{m}$ threshold, but any higher threshold risks altering the vortex cores.

Two LOS disregarding possibilities are considered: ‘deleting’ the LOS or setting the radial velocities at those LOS to zero. The former approach is considered unsuitable. With LiDAR positioned at different positions, each LiDAR position disregards different LOS. Consequently, deleting LOS would result in scans of different sizes - problematic for ANNs. By replacing disregarded measurements with zero, this problem is avoided. Moreover, feature-wise normalisation is no longer applied and the remainder of this thesis uses scan-wise normalisation, given that these models are compared to global models and CNNs. Obtained performance differences have been confirmed to not result from the changed normalisation. The results of the study for LiDAR position L2 are shown in Table 5.5.

The tabulated errors confirm that despite using inferior scan normalisation, the disregarding of LOS is an effective tool for improving the performance of ANNs, not only for the plates up case, but also for the plates down one. Although some parameters - specifically the circulation with plates down - perform superior without disregarding LOS, the differences are negligible, while localisation parameters can result in nearly half of the previous error. Furthermore, the difference between plates up and down networks has shrunk in comparison to not disregarding LOS. Therefore, both the plate effects and the SVSs influences can be mitigated via this technique. The optimisation for the ideal LOS limit should be performed in the future and is beyond the scope of the present work.

Table 5.5.: Average validate errors evaluating LOS disregarding below 7 m at the centreline with measurement L2 data of 1000 train & 100 validate scans using MLPs and scan-wise normalisation (green and red highlight lower and higher errors with respect to not disregarding LOS from Table 5.3).

Plates	Γ MAE (m^2/s)		φ_O MAE (degrees)		R_O MAE (m)		Mean/median virtual ADE (m)	
	CW	CCW	CW	CCW	CW	CCW	CW	CCW
Down	35.06	36.07	1.76	0.94	32.98	36.23	33.95/17.43	42.54/24.74
Up	35.39	37.29	1.85	1.07	42.75	52.54	49.22/27.10	57.80/29.34

5.1.4. Network generalisability

Generalisability is defined as the ability of ANNs to be used with different data. Thus far this thesis has looked at individual LiDAR positions, however these are specific to the measurement campaign. The aim is to train more universal ANNs, such that they could be implemented into WVASs. If this is not done, each airport or new LiDAR position would require another expensive and time-consuming measurement campaign, target generation and model training. This leads to another research question:

- Is it feasible to train global networks based on all LIDAR positions? -

It is speculated that not only the trained models would be more widely applicable, at different airports, runways or LiDAR positions, but also the performance should increase compared to the individual networks. Despite the increase in variability in the scans and arising difficulties of generalisation, when combining the scans from all LiDAR devices, the data set grows substantially. Larger data sets allow more universal patterns to be recognised. This section evaluates whether ANNs simultaneously trained with LiDAR scans from all position, L1 to L5, can achieve similarly low errors as for the individual cases.

The settings for the global network are analogous to those selected for the local networks. However, because the campaign's LiDAR positions have different elevation angle spectra (see Table 4.1) and ANNs usually require scans to always be of the same size, the minimum and maximum elevation angles of any LiDAR position from Table 4.1 are used to make a global ideal scan grid in a similar manner to Section 5.1.2. Unused LOS of a scan are set to zero. In this way, each scan remains relative to the LiDAR that originally measured the velocity field, avoiding the implementation of spatial scan normalisation and related vortex distortions. Additionally, this relative approach makes sure that these models can also be used at different airports, runways or LiDAR positions. Nonetheless, LiDAR positions in the data set are not equally represented. LiDAR position L3 is the most common and therefore most confidently predicted. Moreover, as stated in Section 4.1.1, the campaign data includes more overflights with flat than erected plates, therefore the plates down models have more scans to work and train with, potentially leading to lower errors.

Results for the MLPs are given in Table 5.6. Here, the models are no longer trained three individual times and averaged thereafter, as the number of available scans is high enough to eliminate most performance variations (equally for the following studies).

Table 5.6.: Validate errors evaluating the use of global scans with measurement L1-L5 data of 9187/6708 train (plates down/up) & 1000 validate scans for each using MLPs and scan-wise normalisation (green and red highlight lower and higher errors with respect to L2 results from Table 5.5).

Plates	Γ MAE (m^2/s)		φ_O MAE (degrees)		R_O MAE (m)		Mean/median virtual ADE (m)	
	CW	CCW	CW	CCW	CW	CCW	CW	CCW
Down	34.58	35.39	2.10	1.07	29.62	35.24	32.15/16.01	36.99/19.34
Up	39.14	43.66	1.69	1.27	40.31	47.73	43.24/23.67	50.54/25.34

The majority of errors obtained with globally trained MLPs are at least as small as for the individual models, likely due to the higher number of scans available. Although the colour coding compares only to models trained on scans from LiDAR position L2, the same behaviour is seen for nearly all other positions. Consequently, it is feasible to train global ANNs - the performance is nearly equivalent and more widely applicable. This allows usage of the trained models at other LiDAR positions within the range and elevation angle limits, as neural networks are far superior at interpolating than extrapolating knowledge [52].

5.1.5. Grouping overflights

So far, ANNs have been trained with random scans across all overflights, when in reality scans from one overflight can track the evolution of a vortex in time. Scans from one overflight can be grouped together, creating sets of 20 to 40 scans depending on the vortex lifetime. The scenario obtained from this is closer to new LiDAR measurements from overflights at airports. Although a monitoring system inspects only the most recently recorded scan, with the previous models it was possible for different scans of the same overflight to be used in both the train and validate data sets, hence previous knowledge of an overflight was available. Results in Table 5.7 are obtained when randomly selecting overflights for the train and validate data sets while keeping all scans from one overflight either in the train or validate data set. Note, the sizes of the individual sets differ slightly from the previous analysis, since overflights have a fixed number of scans.

Table 5.7.: Validate errors evaluating full overflights with measurement L1-L5 data of 9128/6683 train & 1059/1025 validate scans (plates down/up) using MLPs and scan-wise normalisation (red highlight higher errors with respect to models with no grouped overflights from Table 5.6).

Plates	Γ MAE (m^2/s)		φ_O MAE (degrees)		R_O MAE (m)		Mean/median real ADE (m)	
	CW	CCW	CW	CCW	CW	CCW	CW	CCW
Down	43.97	43.52	2.55	1.59	48.30	48.17	49.36/28.75	49.88/32.71
Up	52.06	58.41	2.37	1.59	58.09	74.30	59.75/35.97	75.43/44.49

Results degrade in all cases when grouping scans by overflights. While this is unfortunate, it also reveals features of the flow behind an aircraft: even if scans are not grouped by overflights - covering a variety of different velocity fields - each overflight always has a unique set of velocity fields in time. If this was not the case, the results should not differ with overflight dependent data sets. The advantage of grouping by overflights is that results of one entire overflight can be observed - similar to how a LiDAR at an airport would see it in a WVAS - by utilising all associated LiDAR scans arranged in time. Since each of these scans are individually evaluated, the networks are not aware that scans are grouped by overflights.

An example overflight in Figure 5.2 is of an Airbus A320 at LiDAR position L2 with flat plates. Four main observations can be made about the displayed scans:

– Predictions are more reliable and accurate at earlier times, when the wake vortices are still coherent and distinct from the atmospheric turbulence.
– Hence, towards later times predictions deteriorate, especially for the clockwise wake vortex as it transports out of the LiDAR's sight.
– Consider Figures 5.2c and 5.2d. The target first indicates that no counterclockwise wake vortex is present and then later it reappears. Clearly, this is unphysical and reveals existing inconsistencies in the target data from the RV method. In other cases, the RV method captures the vortices more accurately. For instance, in Figure 5.2f the clockwise vortex has either decayed (there were over $3\,\mathrm{min}$ for the vortex to decay) or transported out of the measurement region, but the ANN still predicts it to be present in the frame. Furthermore, some cases are judgement calls. Nonetheless, this exposes that the target data used in this work includes wrong targets, complicating ANN training. It is impossible to know how many wrong targets exist, as minor mistakes may also be present and no absolute truth is available. These faulty targets limit the capabilities of the ANNs.
– It can be speculated whether setting absent vortices to zero adds a bias to the predictions, most of which are to the left of the respective targets in Figure 5.2.

Besides the listed observations, the predictions deliver satisfactory results. Especially at earlier times, when the wake vortex strengths are highest and therefore the hazard is substantial, the predictions show a high accuracy. On the other hand, a following aircraft is unlikely to follow closer than $60\,\mathrm{s}$ after the generator aircraft - the relevant time frames start after $60\,\mathrm{s}$ of an overflight. However, also when the key-point localisation is less accurate, the circulation is correctly predicted as low - a wrong hazard is not predicted. An example of such a case is found in Figure 5.2f. This hints at the possibility of how in the future scans without vortices could be treated, overcoming the limiting factor mentioned in Section 4.5.

5.1.6. Detection and removal of erroneous targets

The current section aims to reduce the number of faulty targets, this gives the following research question:

- How can wrong target scans be identified and eliminated? -

The reason for wake vortices to jump towards the origin originates from how absent wake vortices are handled in this work. Generally, the results in Section 4.5 have shown success with this approach, however

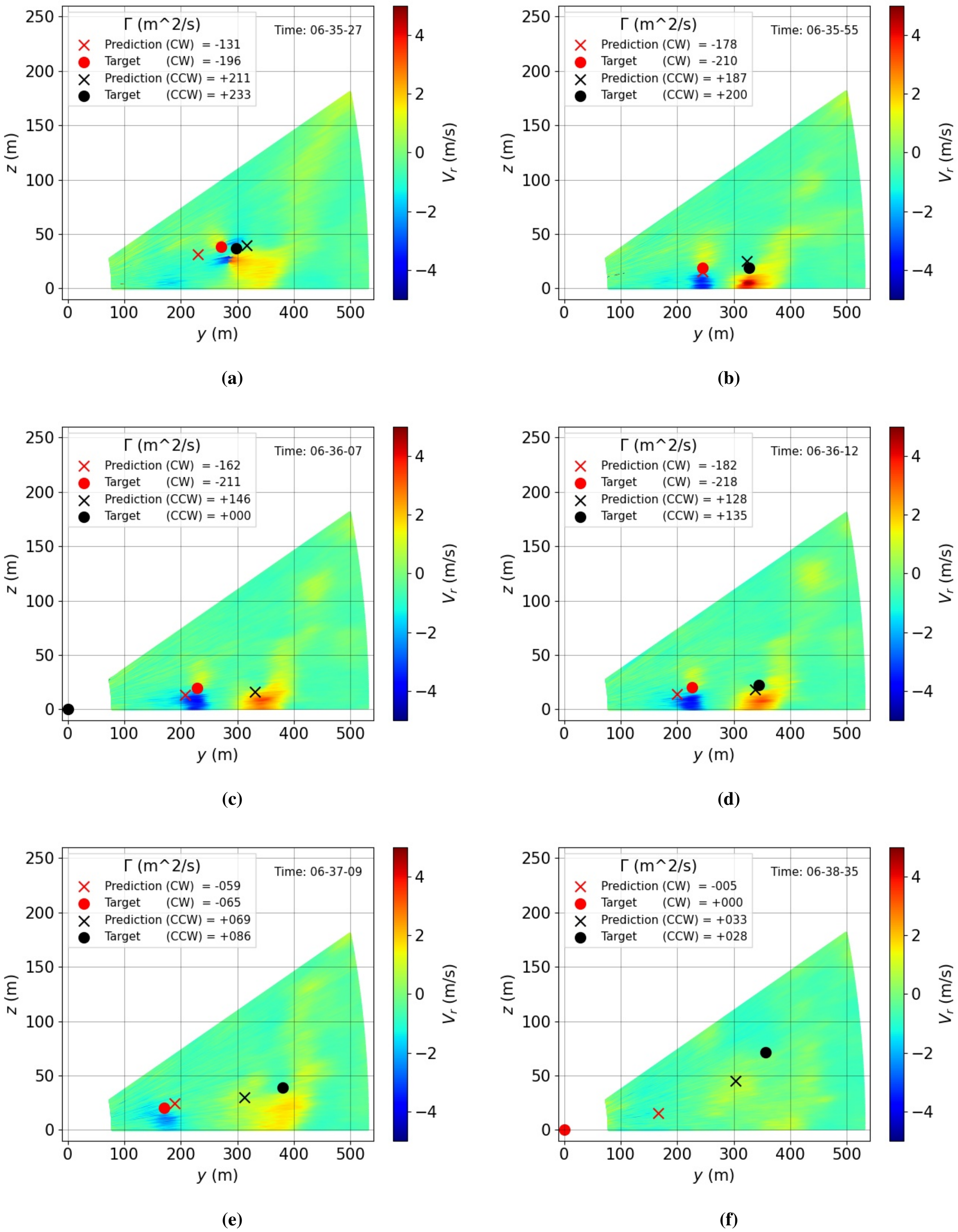


Figure 5.2.: Time frames with key-point and circulation predictions and targets of MLPs trained with 9128 train and 1059 validate scans of an A320 overflight at LiDAR position L2 with plates down.

rather large errors can develop when the targets and predictions are far apart - such as when vortices are missed and thus falsely located at the origin. The objective is to reduce the number of these cases, such that networks can train more reliably, find better minima, and give more realistic errors with the validate data.

The time frames of Figure 5.2 reveal two situations when targets are set to the origin. First, towards the end of an overflight a vortex may become of low strength, making it a challenge to differentiate it from the background turbulence. Second, when the RV method is not able to recognise the wake vortex, although in reality the velocity field infers that it is still strong enough to cause harm (this normally occurs when no peak can be identified in Figure 4.3b). In the latter case, the targets can be outperformed by the prediction of ANNs, as they might still be able to characterise the vortex in a sensible location, even if the RV method fails to do so and wrongly labels the scan characterisation parameters - it sets wrong targets.

It is desirable to dismiss scans with these wrongly labelled targets. This can be done by looking at the individual vortices of complete overflights, with their scans and targets chronologically arranged. Scans are dismissed when the targets of one vortex are set to zero, but any following scan has non-zero targets for that same vortex. Such approach avoids the disregarding of scans which simply have vortices of low strength - the first situation described previously. Note that this approach cannot detect wrongly labelled final scans of an overflight, as there are no following scans to check their validity. Application of the above described method results in 420 scans no longer being considered, nevertheless the number of overflights remains unchanged. The removed scans are quasi-equally distributed among overflights with plates up and down. Table 5.8 clearly shows the lowered errors when no longer considering wrongly labelled scans.

Table 5.8.: Validate errors evaluating the effect of removing wrongly labelled scans with measurement L1-L5 data of 8925/6520 train & 1041/988 validate scans (plates down/up) using MLPs and scan-wise normalisation (green and red highlight lower and higher errors with respect to not removing wrongly labelled scans from Table 5.7, yellow = mixed results).

Plates	Γ MAE (m^2/s)		φ_O MAE (degrees)		R_O MAE (m)		Mean/median real ADE (m)	
	CW	CCW	CW	CCW	CW	CCW	CW	CCW
Down	41.63	40.30	2.53	1.55	45.75	45.59	46.98/27.46	47.22/35.21
Up	53.68	55.47	2.20	1.50	52.65	67.59	54.44/29.72	68.79/43.40

5.1.7. Carrier-to-Noise ratio filter

There are two clashing philosophies with the application of AI to LiDAR scans. On the one end, it is common practice for LiDAR scans to make use of CNR filters, neglecting erroneous radial velocities [35]. On the other end, AI is often praised for its capability to gather information from seemingly faulty data points [98]. This section therefore investigates the following:

- Does a CNR filter aid or hinder the characterisation of wake vortices in LiDAR scans? -

The effect of a filter depends on its type - a wide variety exists [36]. Here, simple CNR thresholds are considered. If error reductions are significant, more sophisticated filters promise even better results.

Two thresholds are required, an upper and a lower limit. CNR values below the lower limit are found when a signal experiences significant atmospheric noise, while too high CNR values occur when the laser beam of a LiDAR hits a hard target, such as a plate in the campaign's case [35]. Additionally, CNR values differ with the range from a LiDAR, the focal range giving the highest performance. Appendix E suggests a focal range of slightly above $500\,\mathrm{m}$ for the campaign LiDAR. Therefore, smaller ranges accommodate a far lower measurement accuracy. Typical thresholds are $-27\,\mathrm{dB}$ to $5\,\mathrm{dB}$ or slightly more conservative $-24\,\mathrm{dB}$ to $0\,\mathrm{dB}$ [35]. However, LiDAR at the campaign applied a de-noise signal to the received backscatter, influencing the aforementioned thresholds.

CNR values of mid-range LOS lead to the usage of $-15\,\mathrm{dB}$ to $9.25\,\mathrm{dB}$ as a CNR threshold range. Entire scans are disregarded if they entail more than $50\,\%$ faulty values within a region made up of range $150\,\mathrm{m}$ to $500\,\mathrm{m}$ and elevation angles above the threshold LOS from Figure 5.1. These restrictions account for the focal range as well as plates, where more erroneous measurements are expected. Measurements outside the threshold CNR range are linearly interpolated from neighbouring scan measurements. With this criteria, no scan of the Vienna data set is neglected. The results with the designed CNR filter are tabulated in Table 5.9.

Table 5.9.: Validate errors evaluating the effect of a CNR filter with measurement L1-L5 data of $8925/6520$ train & $1041/988$ validate scans (plates down/up) using MLPs and scan-wise normalisation (green and red highlight lower and higher errors with respect to no CNR filter used from Table 5.8, yellow = mixed results).

Plates	Γ MAE (m^2/s)		φ_O MAE (degrees)		R_O MAE (m)		Mean/median real ADE (m)	
	CW	CCW	CW	CCW	CW	CCW	CW	CCW
Down	44.62	42.54	2.17	1.54	48.77	40.00	49.73/29.54	41.82/25.81
Up	56.59	55.52	2.09	1.50	52.39	68.33	53.78/29.66	69.54/44.37

It is shown that CNR filters do not necessarily improve the prediction capabilities - in fact they may actually degrade them. There are two possible causes for this, either the chosen CNR filter is not appropriate and perhaps only a more sophisticated filter would improve the results, or as the big data community often suggests, slightly faulty measurements may still include valuable information [98]. Designing an optimised filter is beyond the scope of this thesis. In the light of aiming to have fast-time vortex characterisations, this CNR filter is not a necessity and is therefore not applied further.

5.2. Comparing Multilayer Perceptrons & Convolutional Neural Networks

This section aims to compare the results of the two ANNs analysed in this thesis. For this, Table 5.10 displays the final MLP and CNN prediction accuracies. For the CNNs the architecture from Section 4.4.2 is used along with removing wind, interpolating scans onto ideal global grids, disregarding low LOS, removing faulty targets and grouping overflights. Five comparisons are covered: localisation prediction, circulation prediction, characterisation and training times, as well as interpretability. A discussion summarising feature engineering follows. Lastly, there is a comparison to other automatic characterisation techniques discussed in Section 3.2, precision limits from Table 4.11 and the overall applicability of the algorithm.

Table 5.10.: Validate errors comparing MLPs & CNNs with measurement L1-L5 data of $9128/6683$ train & $1059/1025$ validate scans (plates down/up) using CNNs and scan-wise normalisation (blue = the lowest error per parameter and plate state), MLP values are from Table 5.8.

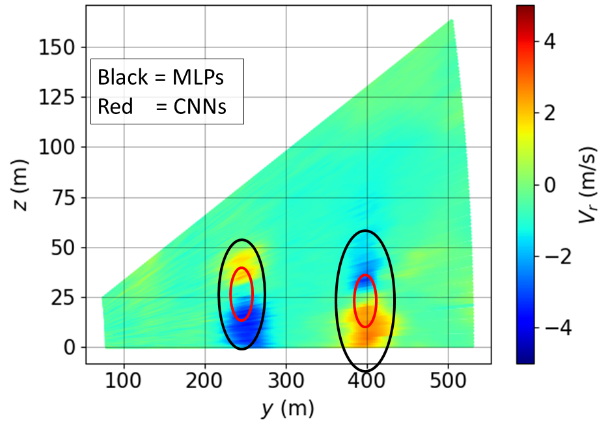
Plates	Γ MAE (m^2/s)		φ_O MAE (degrees)		R_O MAE (m)		Mean/median real ADE (m)	
	CW	CCW	CW	CCW	CW	CCW	CW	CCW
Down								
CNN	32.31	26.44	1.54	1.06	31.89	26.82	32.72/13.27	13.27/13.13
MLP	41.63	40.30	2.53	1.55	45.75	45.59	46.98/27.46	47.22/35.21
Up								
CNN	36.14	32.91	1.55	1.07	56.58	60.39	57.26/45.24	61.05/35.99
MLP	53.68	55.47	2.20	1.50	52.65	67.59	54.44/29.72	68.79/43.40

5.2.1. Localisation

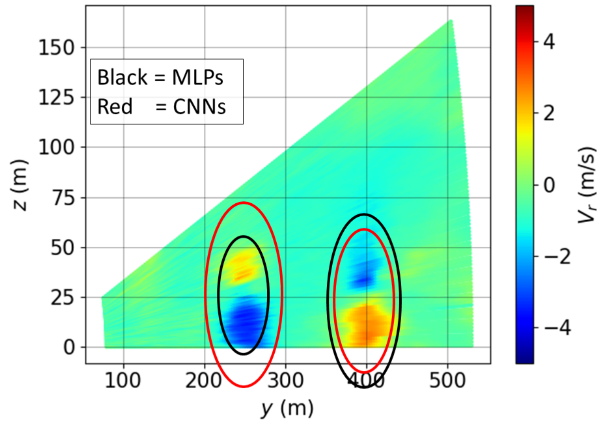
Comparing the errors in Table 5.10 indicates superior performance with CNNs for most location parameters, except the plates up clockwise vortex's R_O MAE and ADE. In this work, it has become clear that the ADE is primarily influenced by the R_O MAE, less by the φ_O MAE. Thus, the two worst performing CNN parameters originate from the same source. Similarly, in comparison to MLPs, CNN R_O MAEs reduce by up to $75\,\%$ and ADEs reduce by up to $60\,\%$. Elevation angle precisions increase by up to $1.0°$ with CNNs.

To further interpret these results, consider Figure 5.3 illustrating the median ADEs of all scans on a single arbitrary scan. The ellipses and the scan should not be compared, instead the localisation capabilities of the models are put into perspective. Two points can be noticed: first, the localisation with plates down is superior to plates up. It is unknown whether this is due to a larger available data set for the plates down case, or whether the plates lead to a more complex flow behind aircraft, making generalisations more rigorous. Second, in most cases the clockwise vortex is predicted with a higher accuracy - possibly because the counterclockwise vortex is located in a coarser grid region. Although the difference is marginal for the CNNs, indicating the reaching of prediction limits, higher differences are observed for the MLPs. The large clockwise vortex ellipse for CNNs cannot be explained, raising the interpretability issue of ANNs.

Figure 5.3 impresses with a high precision for detecting the location of wake vortex centres, especially with CNNs. Since the median ADEs are shown, the distributions should reveal whether the sketched ellipses give a realistic picture of their performance. These are depicted for the validate data set in Figure 5.4.

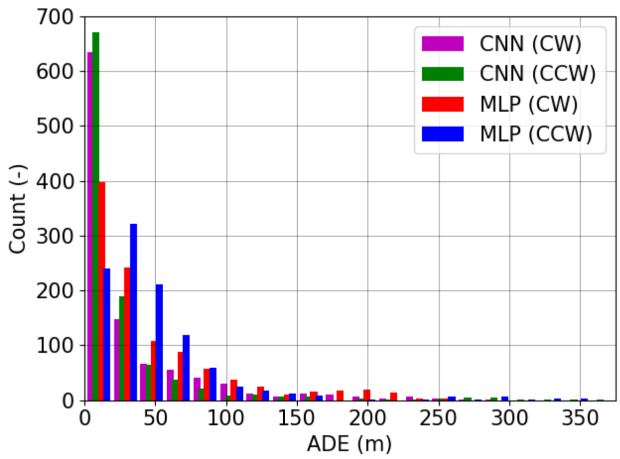


(a) Plates down.

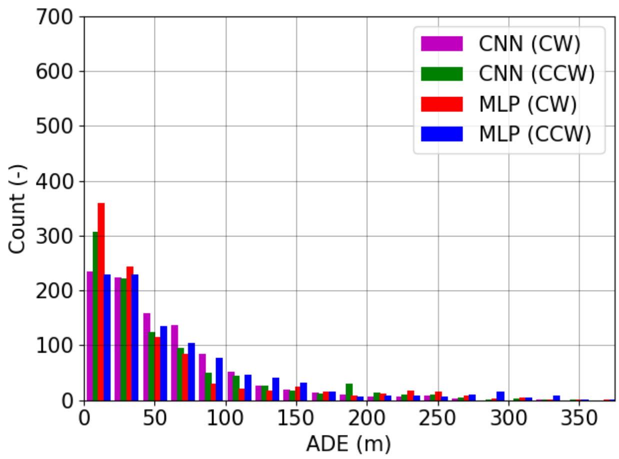


(b) Plates up.

Figure 5.3.: Median ADE ellipses on an arbitrary global LiDAR scan with grouped overflights and all available measured scans, comparing key-point localisation using MLPs & CNNs.



(a) Plates down.



(b) Plates up.

Figure 5.4.: ADE distributions of the global networks with grouped overflights and all available measured scans, comparing key-point localisation using MLPs & CNNs.

The accuracy difference between the plates up and down models is emphasised once again with the count scale. Additionally, a significant difference in ADE distributions between the two ANN types can be seen. CNNs frequently predict with small errors, while the MLP distributions are more spread out and have a greater number of high magnitude ADE outliers. Notably, the difference in plates up distributions is nearly negligible compared to the plates down case. Overall, CNNs have greater precision and are more reliable in their usage due to fewer outliers.

The above analysis is substantiated using statistical tools and a practical scenario. One may claim that the exact vortex location is less critical than the insight into whether it is hovering above the runway and posing a hazard. Thus, a $\pm 50\,\mathrm{m}$ runway corridor is defined within which WVEs could occur [2]. Regardless of its strength, a vortex located within this corridor is considered a hazard. The corridor size originates from a Frankfurt airport campaign [99]. Recapitulating from [2]: an aircraft's lateral drift from the runway centreline is assumed to not exceed $\pm 15\,\mathrm{m}$. Also, a lateral deviation of no more than $30\,\mathrm{m}$, between the wake vortex cores of the generator and the follower aircraft, is assumed. Lastly, a factor of safety is applied.

Table 5.11 summarises statistically relevant metrics, the majority of which confirm the superiority in localisation accuracy with CNNs, especially for the plates down case. Significant performance improvements are observed with CNNs for the plates down case, far more vortices are located in the correct region - in the runway corridor or not (provided the targets represent the truth). CNNs also predict more conservatively compared to MLPs - less vortices are labelled as 'no hazard', when in reality they are a hazard (second row in Table 5.11). This is also confirmed via the False-Negatives rate. Although the plates up case cannot show improvements with CNNs for every metric, the positive predictive rate - the likelihood of predicting a hazard when there actually is one - is also greater for this plate state. This last metric is considered as the most crucial one to an end-user, as it describes the model's reliability.

Table 5.11.: Statistical *location* comparison of MLPs & CNNs performance and reliability for plates down & up - no distinction between vortex senses is made. Positive/negative are vortices within/outside the corridor (blue = the superior network per plate status).

Location metric	Plates down		Plates up	
	MLP	CNN	MLP	CNN
Vortex predicted in correct region (%)	71.9	87.1	65.4	68.2
Dismissed hazards from all errors (%)	70.1	51.5	51.3	65.5
False - Positives rate (%)	15.4	11.5	36.0	23.4
False - Negatives rate (%)	43.2	14.6	33.4	39.2
Positive predictive rate (%)	75.5	86.2	67.7	74.6

5.2.2. Circulation

The circulation of a wake vortex is regarded as the most crucial hazard indicator (see Section 3.2). When comparing Γ MAEs from Table 5.10, the Γ MAEs decrease by around $13\,\mathrm{m}^2/\mathrm{s}$ when using CNNs. Conversely to the localisation parameters, the counterclockwise vortex circulation is usually predicted with a higher accuracy. It may be speculated that this is a consequence of the focal range and more accurate radial velocity measurements at higher ranges (see Section 5.1.7), where the counterclockwise vortex is located. Models show that the absolute magnitude of radial velocities is a crucial indicator for ANN circulation predictions, unlike for the localisations, where the relative magnitudes - the gradients - matter.

The statistical analysis of a hazard is less straightforward when solely using the circulation, the danger largely depends on the encountering aircraft and its path through the vortex (see Section 3.1). Nevertheless,

[100] takes $100\,\mathrm{m}^2/\mathrm{s}$ as a general threshold above which WVEs could be critical. In contrast to Table 5.11, Table 5.12 highlights major statistical results for circulation hazards. First of all, it can be noticed that the circulation prediction differentiates less between models trained with plates up and down, than the localisation prediction. Furthermore, the reliability of hazard prediction is even higher with the circulation compared to the localisation.

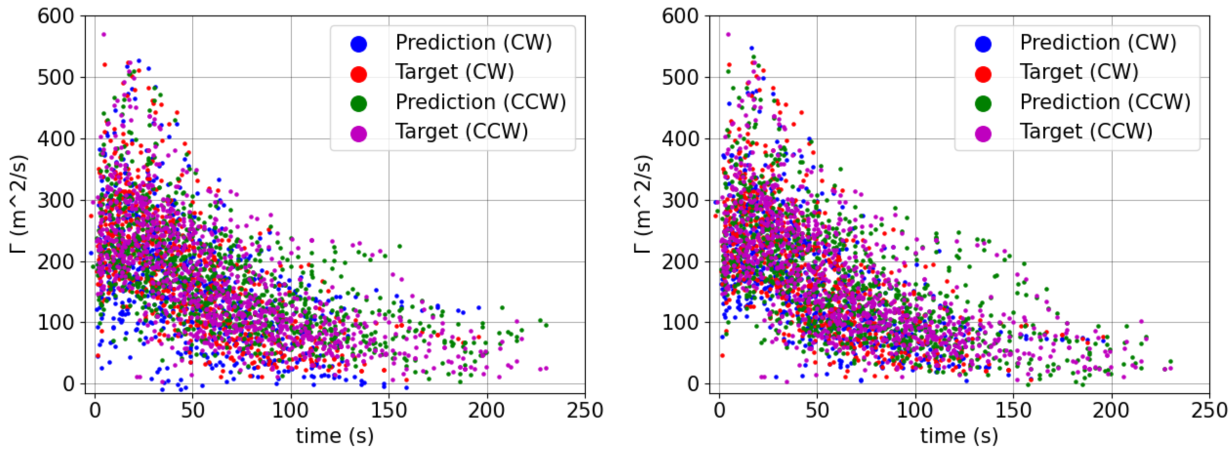
Although in some categories the difference between the two network types is small, overall CNNs obtain smaller errors and higher reliability. On the contrary, row one of Table 5.12 reveals that MLP predictions are more conservative - more errors are false hazards than extenuated hazards. Nonetheless, because CNNs have less fatal errors in absolute terms, their overall reliability is still higher. This can be seen in Figure 5.5, where the temporal evolution of all vortices are depicted in comparison to their vortex circulation. For visualisation purposes, vortices with a target strength of zero are not shown. This has the advantage of directly highlighting vortices predicted to have a low circulation, when in reality they are higher - a problematic prediction scenario. For instance, Figure 5.5a illustrates that MLPs predict many weak vortices at early times, but at that time fewer targets with this strength exist. As seen by the reduced amount of blue dots at early times and low circulation in Figure 5.5b, CNNs can handle early vortex characterisation in a superior manner. Another advantage of the CNN predictions, although not as easy to spot, is that they feature a lower spread of circulation values, again speaking for the higher accuracy of CNN models.

Table 5.12.: Statistical *circulation* comparison of MLPs & CNNs performance and reliability for plates down & up - no distinction between the vortex senses is made. Positive/negative are vortices above/below $100\,\mathrm{m}^2/\mathrm{s}$ (blue = the superior network per plate status).

Circulation metric	Plates down		Plates up	
	MLP	CNN	MLP	CNN
Dismissed hazards from all errors (%)	33.7	44.7	41.6	50.4
False - Positives rate (%)	26.1	13.9	36.3	17.1
False - Negatives rate (%)	8.8	7.4	14.4	9.7
Positive predictive rate (%)	84.1	91.0	80.8	90.4

5.2.3. Characterisation and training times

The goal of this thesis is to complete the characterisation of the wake vortices automatically and in fast-time, such that dynamic aircraft separations can be monitored within WVASs. Recapitulating, the current algorithm used for the characterisation of micro-PCDL is manual and has a processing period of a couple of seconds. Both ANN architectures investigated herein are considerably faster in their characterisation, even with rather low-level hardware. Typically, graphics processing units are employed for ANN training and computation, instead the herein trained models rely on an Intel® Core™ i7-5600U central processing unit at $2.60\,\mathrm{GHz}$. With this hardware, the computation time for a single LiDAR scan with MLPs is around $0.10\,\mathrm{s}$. A



(a) MLPs. **(b)** CNNs.

Figure 5.5.: Temporal vortex circulation evolution of the global network with grouped overflights and all available measured scans with plates down, comparing circulation predictions (time initiates when the overflight crosses the LiDAR scanning plane).

CNN takes slightly longer, $0.16\,\mathrm{s}$. The stated times are not considering data importing, processing or similar, but solely the process of feeding a scan into an already trained model and obtaining the characterisation. The MLPs are faster, however it is questionable whether this marginal difference can justify characterisations with worse precision and a higher number of outliers as discussed in Sections 5.2.1 and 5.2.2.

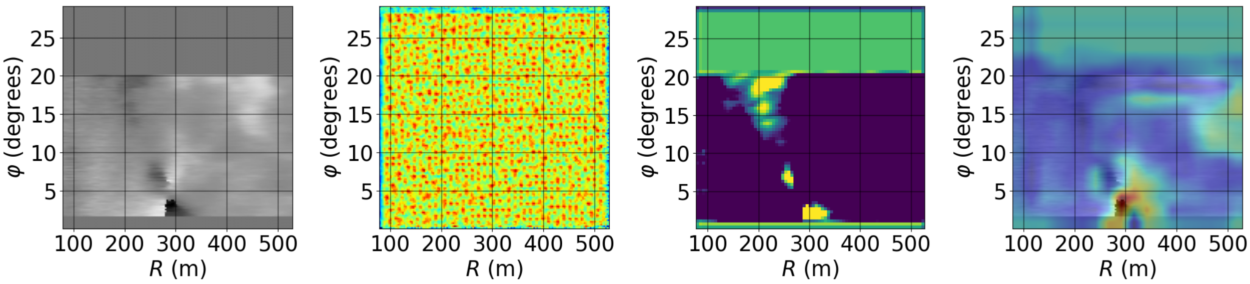
ANN training could also be accelerated with the use of a graphics processing unit. With the used hardware, MLPs and CNNs train for around $35\,\mathrm{min}$ and $810\,\mathrm{min}$, respectively, for the plates down case of Table 5.10, if all 100 epochs are utilised. Data set sizes have a major impact on these durations, however it should be noted that training is performed only once and has no relevance for the operation of a WVAS.

5.2.4. Interpretability

One reason for not using ANNs as a scientific evaluation method is their black box behaviour - inputs are put into models and outputs are delivered without a real explanation for these outcomes. If such trained models were to be used in WVASs, scientists would first want to understand how the characterisation takes place. Studies show that if humans can follow the decision making of algorithms and understand what leads them to arrive at their conclusions, they feel more comfortable in using them [101]. Furthermore, by understanding the thinking process of ANNs, one can validate whether the learned patterns are of relevance, such as high gradients of radial velocity, or whether the predictions are based on background wind which feature engineering could not alleviate - this is highly undesirable.

Unlike with MLPs, the perceptual nature of CNNs allows demystifying operations, some of which are introduced below. These lead to scientifically more satisfying results, reinforcing the use of CNNs. The potential of interpreting CNNs is exemplified with an arbitrary LiDAR scan from position L2 of an Airbus

A320 overflight with flat plates. The visualisations in Figure 5.6 shows the original scan, an input for which a sample filter maximises its activation, the corresponding activation map and the decision making activation for the counterclockwise vortex elevation angle model. These figures only represent a fraction of possibilities, there are far more filters and activations available in the characterisation networks. Shallow layer filters typically contain general shapes and frequencies (to capture velocity gradients), deep layer filters include more input data specific patterns [55]. For instance, Figure 5.6b illustrates the ideal input scan for a filter of the third layer of the CNN architecture. The filter searches for patches of high radial velocity, similar to a vortex. It also detects scan boundaries (blue). The activation maximising scan shows that high radial velocities contribute significantly more to network learning than negative radial velocities. Figure 5.6c shows the resulting activation map from the filter and the measured scan. It confirms the observed, only the black patches from Figure 5.6a are highlighted - the high positive radial velocities. Certainly, the disregarding of negative radial velocities, as done with the ReLu activation function, should be scrutinised and questioned for alternatives, such that also negative velocities are taken into account. Moreover, noise or SVSs, as well as the disregarded LOS from converting local scans to global ones are highlighted in the upper left and vertical boundaries of the scan, respectively.



(a) LiDAR scan as seen by the trained model (dark/bright are higher/lower radial velocities).

(b) Input for which a certain filter maximises its activation (red/blue are higher/lower radial velocities).

(c) Intermediate activation of the filter from the CNN architecture (yellow represents high activations).

(d) Decision making activation of Layer 7 of the CNN architecture (red marks recognised features).

Figure 5.6.: CNN visualisations for the counterclockwise vortex elevation angle model of a measured scan.

The visualisation of the final decision making heatmap is prepared using the Grad-CAM method [101] and presented in Figure 5.6d by superimposing this heatmap onto the original scan. The red patch highlights the counterclockwise vortex slightly above the black patch, which it recognises from the intermediate activations. This does not only indicate that the model can differentiate between the two vortices, but also demonstrates its understanding that not the highest radial velocity patch is searched for, but the centre of the vortex. For a counterclockwise vortex, this is above the high positive radial velocities patch.

Not all filters, activation maps or heatmaps of the trained models deliver such confident results. Nevertheless, the ability of CNNs to allow such insight is a major advantage to obtain scientifically more confident wake vortex characterisations and allow struggles to be addressed.

5.2.5. Discussion

The current section aims to discuss and summarise the main results covered in the present chapter, which is made up of two main parts, feature engineering and comparison of the two ANN types covered in this thesis. Without going into the architectural details of Section 4.4, it is crucial to recapitulate the benefit of using scalar and individual ANNs for each characterisation parameter. In operation, it would be more convenient for a WVAS to incorporate only a single ANN, which outputs all characterisations at once. However, although its speed is comparable to scalar networks, feature sharing amongst the different parameters does not occur, which leads to not only higher errors, but also more outliers.

The first investigation in the results chapter deals with the background wind. The subtraction of the radial velocities of a LiDAR scan prior to each overflight is primarily seen as advantageous for the prediction of the scan location under constant crosswind conditions. Thus, the proposed approach is a useful tool to focus pattern recognition and feature extraction on the primary wake vortices.

The usage of ideal grids highlights the capabilities of ANNs to comprehend the context of individual radial velocity measurements. Small elevation angle variations in the measured LiDAR scans can result in significant performance deterioration. The incorporation of ideal grid interpolation is straightforward, crucial and reinforces the importance of ordering radial velocities when fed into an ANN.

The success of plate lines, reducing the lifetime and strength of wake vortices, may lead to their usage at an increasing number of airports. It is pivotal to understand how ANNs react to flow features generated by plate lines. Networks trained on a mixed data set, containing overflights where plates were erected and other overflights where they were flat, obtain disappointing results. This report therefore recommends the training of two separate models, one for plates erected and one for plates flat on the ground. Without a doubt, this approach is not particularly elegant, as ideally the same performance can be reached with or without plates employed. However, for research purposes it is more reasonable to present the limits of each possibility.

In global networks, erected plates result in more significant characterisation errors. As a direct consequence, it is investigated whether radial velocities below a threshold LOS can be omitted or set to zero. With this feature engineering applied, MAE and ADE values are not only decreased for most parameters, but the difference between the performance of plates up and down models also lessens. This could indicate that plate effects are mainly significant at the boundary layer altitude and separated SVSs, travelling to the altitude of the primary vortices, have a smaller impact.

Interpolating LiDAR scans onto a global grid causes all scans to have the same size, but still be relative to their LiDAR position. This avoids the need for new measurement campaigns and model training for airports of similar geometry and atmospheric conditions as Vienna International Airport.

A perhaps obvious, but often underestimated risk of supervised learning is the usage of targets which do not represent reality. The evaluation of complete overflights determines faulty targets and allows their removal from the data set. This leads to a more robust and precise characterisation of previously unseen vortices. Data sets should always be physically understood and their truth challenged, before accepting them as valid - all data sets have inaccuracies.

The application of CNR filters is not pursued. Either the suggested threshold values are not applicable, or ANNs can make use of faulty measurements. Ultimately, associating the outcome with an unsuitable CNR filter or the network's ability to use faulty data is guesswork. Thus, the application of the built CNR filter is not suggested for the Vienna data, but it is not precluded that alternatives do not work either.

Comparison of the two ANN types follows the feature engineering. Both the localisation and circulation predictions are superior with CNNs. Also, far less outliers are found with them. The localisation in the critical corridor is above $75\,\%$ reliable and the hazard prediction reliability via the circulation is above $90\,\%$, compared to $68\,\%$ and $81\,\%$ for MLPs, respectively. Thus, whether MLPs are sufficient for WVASs is questionable. Lastly, although MLPs allow faster prediction, the processing speed of CNNs is adequate for the foreseen application and evaluation of similar, large data sets. On top of this, CNNs are more trustworthy, as their reasoning can be visualised and tracked. Achievements of this work's simple CNN models are promising to lead to their usage in WVASs and routine measurements. A limiting factor could be the model extrapolation capabilities to other airports and the underrepresented aircraft weight categories.

5.2.6. Comparison with the state-of-the-art

There are three comparisons in this section: comparison to the state-of-the-art automatic vortex characterisation using the VE method [49], comparison to the errors of the manual RV method [51] and comparison to the proxy data errors. The latter two are combined with the precision limits as defined in Table 4.11. Both comparisons are conducted for the CNNs only - the higher performing ANN type.

Since the targets originate from the RV method, it is very unlikely for the ANN predictions to be more accurate. As expected, the errors from Table 5.10 are not more precise than the limits from Table 4.11. The φ_O and R_O MAEs are around 3 to 5 and 7 to 20 times worse than the precision limits, respectively. Considering this, the runway corridor area within each LiDAR scanning window is $18\,\%$ to $26\,\%$. The statistical results from the CNNs are remarkable in comparison to what randomised predictions using the corridor area would give. The circulation prediction accuracies are much closer to the precision limits than the localisation ones, the precision is around two times worse. Hence, significant accuracy improvements cannot be expected when using targets from the RV method and are most likely not required.

Comparison to the state-of-the-art characterisation precision of the automatic VE method, which is not applicable to all LiDAR, elucidates that the circulation errors are at least twice as high with CNNs [33]. Furthermore, while ADEs of the VE method are $7.9\,\mathrm{m}$ when unifying vertical ($4.5\,\mathrm{m}$) and horizontal ($6.5\,\mathrm{m}$) localisation inaccuracies (see Section 3.2), median ADEs from the trained CNNs for the plates down case are nearly twice as high (see Table 5.10) [33]. The plates up models' precision is 5 to 7 times worse [33].

Overall, there is room for improvement with the method of this thesis, but the precision cannot outperform that of the RV method, unless the targets are created in another manner. The clear advantage of ANNs is their swift and automatic Leosphere Windcube 200S LiDAR scan processing. CNNs are superior in nearly all aspects, however they do not yet reach the accuracy of the RV method. However unlike MLPs, they may prove sufficient with further tuning.

6. Conclusions

6.1. Thesis accomplishments

Throughout this thesis, the feasibility of ANNs for the fast-time characterisation of wake vortices in LiDAR scans of wake vortices, via their strength and location, has been investigated and evaluated. Proxy data based on the Lamb-Oseen vortex model proves to be helpful in the design of ANN architectures, given the ideal nature of these artificial LiDAR scans. The development of the ANN architectures showed that overfitting to train data limits their complexity. Regularisation methods did not prove successful in overcoming this, but instead degraded the performance further. Additionally, scalar regression - an individual ANN for each characterisation parameter - is preferred over vector regression.

The following feature engineering proved to be advantageous when applied to LiDAR wake vortex scans: using a LiDAR scan before an overflight to remove crosswinds, and interpolation onto an ideal global elevation-range grid. It was also recognised that plate lines add too much variety to the velocity fields behind an aircraft, therefore separate ANNs must be trained for each plate state. In order to alleviate the performance disparities between different plate state models, disregarding low LOS was found to be advantageous. At the same time, it resulted in improved characterisation performance. This work also reinforced the importance of verifying the target data used for ANNs and removing any erroneous scans. CNR filters, which normally assist this, did not prove profitable with the current data set.

CNNs clearly outperformed the traditional MLPs in the localisation of key-points - the wake vortex centres/origins - and circulation predictions. Despite that, the processing speed of both types is fast, up to $0.1\,\mathrm{s}$ for a single scan, both allowing fast-time wake vortex characterisation.

The objectives for proving the suitability of AI, particularly of ANNs, were fulfilled in this work. Wake vortices are recognised in LiDAR scans, similar to previous literature. However, on top of this, the vortices are automatically characterised with an acceptable accuracy. Depending on the hazard definition, a reliability of up to $91\,\%$ is achievable. Although limited so far, proxy scans accelerate architecture development and coordinate system decision making, they also show the potential for further implementation into the ANNs. By utilising a fundamental ANN type, this work has had the ability to indicate trade-offs of different feature engineering and normalisation, enabling better decision making.

This thesis has shown a variety of potential for AI in fluid dynamic applications through integrating vortex models, flow measurements and ANNs. The models' accuracy and reliability are outstanding using simple software and hardware, pushing the development of dynamic aircraft separations forward by reliably providing safety monitoring nets for prediction systems at airports. Additionally, the accuracy and ability to process large data sets as provided by the Vienna campaign could lead to ANN usage for other campaigns or routine measurements required for the development of the highly discussed temporal aircraft separations.

6.2. Future works

Discussions in the previous chapters reveal areas for improvement. Furthermore, the achieved results also recommend the continuation of combining wake vortex characterisation and ANNs. The following list suggests some visions and possible future steps to be carried out within this subject:

- The network architectures should undergo optimisation via Bayesian optimisation (see Section 4.4). However, also settings such as the activation function, threshold LOS for dealing with plates and SVSs, as well as the application of different CNR filters should be investigated.
- More sophisticated architectures, which include custom layers or loss functions should be consulted. This could allow the inclusion of physical constraints and remove the obstacle of setting absent wake vortices to zero, causing large localisation outliers.
- The capabilities of pre-trained ANN models should be investigated. These are based on enormous, different data sets such as ImageNet and can be applied to a multitude of applications by using trained parameters in shallow layers and only modifying deeper layers to the specific problem, performing so-called transfer learning. For this, a graphics processing unit is required.
- Given the weak extrapolation capabilities of ANNs, airports aiming to use the herein trained models should have similar geometries, as well as elevation angle and range spectra to the Vienna campaign. Further scans from other LiDAR positions, runways and also airports should be used to either verify the success of the current models, or to train new ones. Additionally, it should be aimed to include a higher variation of aircraft types to eliminate bias. This could result in a higher generalisability.
- The success of the simple proxy data and the expense of measurement data suggests the generation of more realistic proxy data, which includes computational simulations, noise, atmospheric as well as boundary layer effects and even shows the mutual induction of the wake vortex pairs. This would allow independence of specific airport geometry and LiDAR positions.
- Currently, models are trained only on LiDAR scans that knowingly contain at least one vortex. Thus, it could be desired to first detect which LiDAR scans within a data set contain vortices. This may be realised with a mix of classification and regression. A bounding box approach such as YOLO could be trained to detect whether a scan entails vortices by introducing circulation thresholds (see Section 5.1.5), and within these boxes, vortices could be characterised by their location and strength.
- The trained models should be incorporated into WVASs, combining them with prediction models.
- Obtained knowledge from this work could be extended to track and predict the path of wake vortices given a single scan. This is achieved by including a temporal component in the analysis.

Bibliography

[1] Eurocontrol, 'Challenges of growth', Eurocontrol, Tech. Rep., 2018, European aviation in 2040. [Online]. Available: `https://www.eurocontrol.int/publication/challenges-growth-2018` (visited on 27/11/2020).

[2] F. Holzäpfel, A. Stephan, G. Rotshteyn, S. Körner, N. Wildmann, L. Oswald, T. Gerz, G. Borek, A. Floh, C. Kern *et al.*, 'Mitigating wake turbulence risk during final approach via plate lines', in *American Institute of Aeronautics and Astronautics AVIATION 2020 FORUM*, 2020. DOI: `10.2514/6.2020-2835`.

[3] T. Gerz, F. Holzäpfel and D. Darracq, 'Commercial aircraft wake vortices', *Progress in Aerospace Sciences*, vol. 38, no. 3, pp. 181–208, 2002. DOI: `10.1016/S0376-0421(02)00004-0`.

[4] F. Holzäpfel, T. Gerz, F. Köpp, E. Stumpf, M. Harris, R. I. Young and A. Dolfi-Bouteyre, 'Strategies for circulation evaluation of aircraft wake vortices measured by lidar', *Journal of Atmospheric and Oceanic Technology*, vol. 20, no. 8, pp. 1183–1195, 2003. DOI: `10.1175/1520-0426(2003)020<1183:SFCEOA>2.0.CO;2`.

[5] F. Holzäpfel and M. Steen, 'Aircraft wake-vortex evolution in ground proximity: Analysis and parameterization', *American Institute of Aeronautics and Astronautics journal*, vol. 45, no. 1, pp. 218–227, 2007. DOI: `10.2514/1.23917`.

[6] F. Holzäpfel, 'Analysis of potential wake vortex encounters at a major european airport', *Aircraft Engineering and Aerospace Technology*, vol. 89, no. 5, pp. 634–643, 2017. DOI: `10.1108/AEAT-01-2017-0043`.

[7] J. Cheng, A. Hoff, J. Tittsworth and W. A. Gallo, 'The development of wake turbulence re-categorization in the united states', in *8th American Institute of Aeronautics and Astronautics Atmospheric and Space Environments Conference*, 2016. DOI: `10.2514/6.2016-3434`.

[8] J. D. Anderson, *Fundamentals of aerodynamics*, 5th. New York: McGraw-Hill, 2011, ISBN: 978-0-07-339810-5.

[9] F. W. Lanchester, *Aerodynamics: Constituting the First Volume of a Complete Work on Aerial Flight*, 1. A. Constable & Company, Limited, 1907, LCCN:08026416. Aerial Flight.

[10] L. Prandtl, *Theory of Lifting Surfaces - Part 1*. National Advisory Committee for Aeronautics, 1920, Report no. NACA-TN-10-Pt-2.

[11] L. Prandtl, *Theory of Lifting Surfaces - Part 2*. National Advisory Committee for Aeronautics, 1920, Report no. NACA-TN-9.

[12] P. K. Kundu and I. M. Cohen, *Fluid Mechanics*, 2nd. Boston: Academic Press, 2002, pp. 629–660, ISBN: 978-0-12-178251-1. DOI: 10.1016/B978-0-08-054558-5.50023-1.

[13] A. Stephan, 'Wake vortices of landing aircraft', DLR Forschungsbericht 2014-10, Ph.D. dissertation, Ludwig-Maximilians-Universität München, 2014.

[14] J. R. Spreiter and A. H. Sacks, 'The rolling up of the trailing vortex sheet and its effect on the downwash behind wings', *Journal of the Aeronautical Sciences*, vol. 18, no. 1, pp. 21–32, 1951. DOI: 10.2514/8.1830.

[15] A. Stephan, D. Rohlmann, F. Holzäpfel and R. Rudnik, 'Effects of detailed aircraft geometry on wake vortex dynamics during landing', *Journal of Aircraft*, vol. 56, no. 3, pp. 974–989, 2019. DOI: 10.2514/1.C034961.

[16] H. Lamb, *Hydrodynamics*, 6th. New York: Dover, 1923, ISBN: 978-0-486-60256-1.

[17] D. C. Burnham and J. N. Hallock, *Chicago monostatic acoustic vortex sensing system*, Report no. DOT-TSC-FAA-79-103.IV, Jul. 1982.

[18] N. N. Ahmad and F. Proctor, 'Review of idealized aircraft wake vortex models', in *52nd Aerospace Sciences Meeting*, 2014. DOI: 10.2514/6.2014-0927.

[19] T. Leweke, S. Le Dizès and C. H. Williamson, 'Dynamics and instabilities of vortex pairs', *Annual Review of Fluid Mechanics*, vol. 48, pp. 507–541, 2016. DOI: 10.1146/annurev-fluid-122414-034558.

[20] M. V. Melander, N. J. Zabusky and J. C. Mcwilliams, 'Symmetric vortex merger in two dimensions: Causes and conditions', *Journal of Fluid Mechanics*, vol. 195, pp. 303–340, 1988. DOI: 10.1017/S0022112088002435.

[21] S. C. Crow, 'Stability theory for a pair of trailing vortices', *American Institute of Aeronautics and Astronautics journal*, vol. 8, no. 12, pp. 2172–2179, 1970. DOI: 10.2514/3.6083.

[22] F. Holzäpfel, T. Hofbauer, D. Darracq, H. Moet, F. Garnier and C. F. Gago, 'Analysis of wake vortex decay mechanisms in the atmosphere', *Aerospace Science and Technology*, vol. 7, no. 4, pp. 263–275, 2003. DOI: 10.1016/S1270-9638(03)00026-9.

[23] J. N. Hallock and F. Holzäpfel, 'A review of recent wake vortex research for increasing airport capacity', *Progress in Aerospace Sciences*, vol. 98, pp. 27–36, 2018. DOI: 10.1016/j.paerosci.2018.03.003.

[24] A. Stephan, T. Misaka and F. Holzäpfel, 'Aircraft wake vortex decay in ground proximity - physical mechanisms and artificial enhancement', in *4th American Institute of Aeronautics and Astronautics Atmospheric and Space Environments Conference*, 2012. DOI: 10.2514/6.2012-2672.

[25] B. Etkin, *Dynamics of atmospheric flight*. John Wiley and Sons, New York, 1972, ISBN: 0-486-44522-4.

[26] A. Stephan, F. Holzäpfel and T. Misaka, 'Aircraft wake-vortex decay in ground proximity—physical mechanisms and artificial enhancement', *Journal of Aircraft*, vol. 50, no. 4, pp. 1250–1260, 2013. DOI: 10.2514/1.C032179.

[27] F. Köpp, 'Doppler lidar investigation of wake vortex transport between closely spaced parallel runways', *American Institute of Aeronautics and Astronautics journal*, vol. 32, no. 4, pp. 805–810, 1994. DOI: 10.2514/3.12057.

[28] A. Stephan, F. Holzäpfel and T. Misaka, 'Hybrid simulation of wake-vortex evolution during landing on flat terrain and with plate line', *International Journal of Heat and Fluid Flow*, vol. 49, pp. 18–27, 2014. DOI: 10.1016/j.ijheatfluidflow.2014.05.004.

[29] F. Holzäpfel, *Oberflächenstruktur einer erdbodenoberfläche zur beschleunigung des zerfalls von wirbelschleppen im endteil eines anflugs auf eine landebahn*, Patent: DE 102011010147 B3, Jul. 2012.

[30] F. Holzäpfel, A. Stephan and G. Rotshteyn, 'Plate lines reduce lifetime of wake vortices during final approach to vienna airport', in *American Institute of Aeronautics and Astronautics Scitech 2020 Forum*, 2020. DOI: 10.2514/6.2020-0050.

[31] S. Schönhals, M. Steen and P. Hecker, 'Wake vortex prediction and detection utilising advanced fusion filter technologies', *Aeronautical Journal*, vol. 115, no. 1166, pp. 221–228, 2011. DOI: 10.1017/S0001924000005674.

[32] F. Köpp, S. Rahm and I. N. Smalikho, 'Characterization of aircraft wake vortices by 2-μ m pulsed doppler lidar', *Journal of Atmospheric and Oceanic Technology*, vol. 21, no. 2, pp. 194–206, 2004. DOI: 10.1175/1520-0426(2004)021<0194:COAWVB>2.0.CO;2.

[33] F. Köpp, S. Rahm, I. N. Smalikho, A. Dolfi, J.-P. Cariou, M. Harris and R. I. Young, 'Comparison of wake-vortex parameters measured by pulsed and continuous-wave lidars', *Journal of Aircraft*, vol. 42, no. 4, pp. 916–923, 2005. DOI: 10.2514/1.8177.

[34] N. Wildmann, *Wind and turbulence measurements with multiple doppler wind lidars*, Innsbruck University, May 2019.

[35] N. Wildmann, Personal communication, Oct. 2020.

[36] L. Alcayaga, 'Filtering of pulsed lidars data using spatial information and a clustering algorithm', *Atmospheric Measurement Techniques Discussions*, 2020. DOI: 10.5194/amt-2019-450.

[37] M. Hoogstraten, H. G. Visser, D. Hart, V. Treve and F. Rooseleer, ‘Improved understanding of en route wake-vortex encounters’, *Journal of Aircraft*, vol. 52, no. 3, pp. 981–989, 2015. DOI: 10.2514/1.C032858.

[38] F. Rooseleer, V. Treve, R. Graham and I. De Visscher, ‘Wake turbulence re-categorisation on approach and departure for safe and more efficient air traffic management’, in *30th Congress of the International Council of the Aeronautical Sciences*, ICAS, 2016. [Online]. Available: http://www.icas.org/ICAS_ARCHIVE/ICAS2016/data/papers/2016_0230_paper.pdf (visited on 27/11/2020).

[39] F. Holzapfel, T. Gerz, M. Frech and A. Dornbrack, ‘Wake vortices in convective boundary layer and their influence on following aircraft’, *Journal of Aircraft*, vol. 37, no. 6, pp. 1001–1007, 2000. DOI: 10.2514/2.2727.

[40] R. Nelson, ‘Trailing vortex wake encounters at altitude - a potential flight safety issue?’, in *American Institute of Aeronautics and Astronautics Atmospheric Flight Mechanics Conference and Exhibit*, 2006. DOI: 10.2514/6.2006-6268.

[41] Transportation Safety Board of Canada, *Encounter with wake turbulence - aviation investigation report a08w000*, Gatineau, QC, Canada, Jan. 2008. [Online]. Available: https://www.tsb.gc.ca/eng/rapports-reports/aviation/2008/a08w0007/A08W0007.pdf (visited on 27/11/2020).

[42] J. Critchley and P. Foot, *United Kingdom civil aviation authority wake vortex database: analysis of incidents reported between 1982 and 1990, Civil Aviation Authority Paper 91*. Civil Aviation Authority, 1991, ISBN: 0-86039-489-1.

[43] V. Rossow and K. James, ‘Overview of wake-vortex hazards during cruise’, *Journal of Aircraft*, vol. 37, no. 6, pp. 960–975, 2000. DOI: 10.2514/2.2723.

[44] F. Holzäpfel, T. Gerz, M. Frech, A. Tafferner, F. Köpp, I. N. Smalikho, S. Rahm, K.-U. Hahn and C. Schwarz, ‘The wake vortex prediction and monitoring system wsvbs part i: Design’, *Air Traffic Control Quarterly*, vol. 17, no. 4, pp. 301–322, 2009. DOI: 10.2514/atcq.17.4.301.

[45] C. W. Schwarz and K.-U. Hahn, ‘Full-flight simulator study for wake vortex hazard area investigation’, *Aerospace Science and Technology*, vol. 10, no. 2, pp. 136–143, 2006. DOI: https://doi.org/10.1016/j.ast.2005.09.005.

[46] A. Stephan, Personal communication, Sep. 2020.

[47] D. A. Hinton and C. R. Tatnall, *A candidate wake vortex strength definition for application to the NASA aircraft vortex spacing system (AVOSS)*. National Aeronautics and Space Administration, Langley Research Center, 1997, vol. 110343, NASA Technical Memorandum 110343.

[48] F. Holzäpfel, ‘Probabilistic two-phase aircraft wake-vortex model: Further development and assessment’, *Journal of Aircraft*, vol. 43, no. 3, pp. 700–708, 2006. DOI: 10.2514/1.16798.

[49] V. Banakh and I. N. Smalikho, *Coherent Doppler Wind Lidars in a Turbulent Atmosphere*. Artech House, 2013, Radar, ISBN: 978-1-60807-667-3.

[50] I. N. Smalikho and V. Banakh, 'Estimation of aircraft wake vortex parameters from data measured with a 1.5-μ m coherent doppler lidar', *Optics letters*, vol. 40, no. 14, pp. 3408–3411, 2015. DOI: 10.1364/OL.40.003408.

[51] I. N. Smalikho, V. Banakh, F. Holzäpfel and S. Rahm, 'Method of radial velocities for the estimation of aircraft wake vortex parameters from data measured by coherent doppler lidar', *Optics Express*, vol. 23, no. 19, A1194–A1207, 2015. DOI: 10.1364/OE.23.0A1194.

[52] S. L. Brunton, B. R. Noack and P. Koumoutsakos, 'Machine learning for fluid mechanics', *Annual Review of Fluid Mechanics*, vol. 52, pp. 477–508, 2020. DOI: 10.1146/annurev-fluid-010719-060214.

[53] S. Zholtovski, A. Stephan and F. Holzäpfel, 'Towards virtual flight in realistic environments: A hybrid coupled simulation method', *American Institute of Aeronautics and Astronautics journal*, vol. 58, no. 3, pp. 1266–1277, 2020. DOI: 10.2514/1.J058390.

[54] I. Arel, D. C. Rose and T. P. Karnowski, 'Deep machine learning - a new frontier in artificial intelligence research - research frontier', *IEEE computational intelligence magazine*, vol. 5, no. 4, pp. 13–18, 2010. DOI: 10.1109/MCI.2010.938364.

[55] F. Chollet, *Deep Learning with Python*. Birmingham: Manning Publications Company, 2017, ISBN: 978-1-617-29443-3.

[56] C.-F. Teng, C.-C. Liao, C.-H. Chen and A.-Y. A. Wu, 'Polar feature based deep architectures for automatic modulation classification considering channel fading', in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, IEEE, 2018, pp. 554–558. DOI: 10.1109/GlobalSIP.2018.8646375.

[57] M. D. Zeiler and R. Fergus, 'Visualizing and understanding convolutional networks', in *European conference on computer vision*, Springer, 2014, pp. 818–833. DOI: 10.1007/978-3-319-10590-1_53.

[58] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel, 'Backpropagation applied to handwritten zip code recognition', *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989. DOI: 10.1162/neco.1989.1.4.541.

[59] M. Milano and P. Koumoutsakos, 'Neural network modeling for near wall turbulent flow', *Journal of Computational Physics*, vol. 182, no. 1, pp. 1–26, 2002. DOI: 10.1006/jcph.2002.7146.

[60] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, 'Imagenet: A large-scale hierarchical image database', in *2009 IEEE conference on computer vision and pattern recognition*, IEEE, 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[61] A. Krizhevsky, I. Sutskever and G. E. Hinton, ‘Imagenet classification with deep convolutional neural networks’, in *Advances in neural information processing systems*, 2012, pp. 1097–1105. DOI: 10.1145/3065386.

[62] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, ‘You only look once: Unified, real-time object detection’, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.91.

[63] A. Toshev and C. Szegedy, ‘Deeppose: Human pose estimation via deep neural networks’, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1653–1660. DOI: 10.1109/CVPR.2014.214.

[64] J. Ribera, D. Guera, Y. Chen and E. J. Delp, ‘Locating objects without bounding boxes’, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6479–6489.

[65] W. de Winter, *Hacker noon*, Key-point detection in flower images using deep learning, 2018. [Online]. Available: https://hackernoon.com/key-point-detection-in-flower-images-using-deep-learning-66a06aadc765 (visited on 18/07/2020).

[66] A. Nibali, Z. He, S. Morgan and L. Prendergast, ‘Numerical coordinate regression with convolutional neural networks’, *arXiv preprint arXiv:1801.07372*, 2018.

[67] Y. Sun, X. Wang and X. Tang, ‘Deep convolutional network cascade for facial point detection’, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 3476–3483. DOI: 10.1109/CVPR.2013.446.

[68] B. Shi, X. Bai, W. Liu and J. Wang, ‘Face alignment with deep regression’, *IEEE transactions on neural networks and learning systems*, vol. 29, no. 1, pp. 183–194, 2016. DOI: 10.1109/TNNLS.2016.2618340.

[69] J. J. Tompson, A. Jain, Y. LeCun and C. Bregler, ‘Joint training of a convolutional network and a graphical model for human pose estimation’, in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence and K. Q. Weinberger, Eds., vol. 27, Curran Associates, Inc., 2014, pp. 1799–1807. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/e744f91c29ec99f0e662c9177946c627-Paper.pdf (visited on 27/11/2020).

[70] B. Colvert, M. Alsalman and E. Kanso, ‘Classifying vortex wakes using neural networks’, *Bioinspiration & biomimetics*, vol. 13, no. 2, 2018. DOI: 10.1088/1748-3190/aaa787.

[71] W. Pan, D. Yingjie, Z. Qiang, T. Jiahao and Z. Jun, ‘Deep learning for aircraft wake vortex identification’, in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, vol. 685, 2019. DOI: 10.1088/1757-899x/685/1/012015.

[72] W. Pan, Z. Wu and X. Zhang, 'Identification of aircraft wake vortex based on svm', *Mathematical Problems in Engineering*, vol. 2020, 2020. DOI: 10.1155/2020/9314164.

[73] M. Szarvas, U. Sakai and J. Ogata, 'Real-time pedestrian detection using lidar and convolutional neural networks', in *2006 IEEE Intelligent Vehicles Symposium*, IEEE, 2006, pp. 213–218. DOI: 10.1109/IVS.2006.1689630.

[74] A. Stephan, *Sicherheit und effizienz des luftverkehrs*, Institutsüberprüfung – Institut für Physik der Atmosphäre 2020, Jun. 2020.

[75] A. Stephan, Personal communication, Jul. 2020.

[76] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer New York, 2006, ISBN: 978-0-387-31073-2.

[77] S. Savarese, 'Introduction to convolutional neural networks', in *CS231A: Computer Vision, From 3D Reconstruction to Recognition*, Stanford: California, US, 2018. [Online]. Available: https://web.stanford.edu/class/cs231a/lectures/intro_cnn.pdf (visited on 22/09/2020).

[78] S. S. Cross, R. F. Harrison and R. L. Kennedy, 'Introduction to neural networks', *The Lancet*, vol. 346, no. 8982, pp. 1075–1079, 1995. DOI: 10.1016/S0140-6736(95)91746-2.

[79] X. Glorot and Y. Bengio, 'Understanding the difficulty of training deep feedforward neural networks', in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Y. W. Teh and M. Titterington, Eds., Proceedings of Machine Learning Research, vol. 9, Chia Laguna Resort, Sardinia, Italy: JMLR Workshop and Conference Proceedings, May 2010, pp. 249–256. [Online]. Available: http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf (visited on 01/12/2020).

[80] J. Wu, 'Introduction to convolutional neural networks', *National Key Lab for Novel Software Technology. Nanjing University. China*, vol. 5, pp. 1–23, 2017. [Online]. Available: https://cs.nju.edu.cn/wujx/paper/CNN.pdf (visited on 27/11/2020).

[81] S. Ruder, 'An overview of gradient descent optimization algorithms', *arXiv preprint arXiv:1609.04747*, 2016.

[82] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy and P. T. P. Tang, 'On large-batch training for deep learning: Generalization gap and sharp minima', *arXiv preprint arXiv:1609.04836*, 2016.

[83] M. Thoma, 'Analysis and optimization of convolutional neural network architectures', *arXiv preprint arXiv:1707.09725*, 2017.

[84] S. Lathuilière, P. Mesejo, X. Alameda-Pineda and R. Horaud, 'A comprehensive analysis of deep regression', *IEEE transactions on pattern analysis and machine intelligence*, 2019. DOI: 10.1109/TPAMI.2019.2910523.

[85] K. O'Shea and R. Nash, 'An introduction to convolutional neural networks', *arXiv preprint arXiv:1-511.08458*, 2015.

[86] W. Hou, D. Darakananda and J. D. Eldredge, 'Machine-learning-based detection of aerodynamic disturbances using surface pressure measurements', *American Institute of Aeronautics and Astronautics journal*, vol. 57, no. 12, pp. 5079–5093, 2019. DOI: 10.2514/1.J058486.

[87] R. Jiang and S. Mei, 'Polar coordinate convolutional neural network: From rotation-invariance to translation-invariance', in *2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2019, pp. 355–359. DOI: 10.1109/ICIP.2019.8802940.

[88] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow - Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd. Sebastopol: "O'Reilly Media, Inc.", 2019, ISBN: 978-1-492-03261-8.

[89] F. Chollet *et al.*, *Keras*, 2015. [Online]. Available: https://keras.io (visited on 08/07/2020).

[90] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Y. Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu and Xiaoqiang Zheng, *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015. [Online]. Available: https://www.tensorflow.org/ (visited on 20/07/2020).

[91] W. E. Faller and S. J. Schreck, 'Neural networks: Applications and opportunities in aeronautics', *Progress in Aerospace Sciences*, vol. 32, no. 5, pp. 433–456, 1996. DOI: 10.1016/0376-0421(95)00011-9.

[92] J. Ling, A. Kurzawski and J. Templeton, 'Reynolds averaged turbulence modelling using deep neural networks with embedded invariance', *Journal of Fluid Mechanics*, vol. 807, pp. 155–166, 2016. DOI: 10.1017/jfm.2016.615.

[93] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. R. Salakhutdinov, 'Improving neural networks by preventing co-adaptation of feature detectors', *arXiv preprint arXiv:1207.0580*, 2012.

[94] S. Ioffe and C. Szegedy, 'Batch normalization: Accelerating deep network training by reducing internal covariate shift', *arXiv preprint arXiv:1502.03167*, 2015.

[95] J. L. Ba, J. R. Kiros and G. E. Hinton, 'Layer normalization', *arXiv preprint arXiv:1607.06450*, 2016.

[96] D. P. Kingma and J. Ba, 'Adam: A method for stochastic optimization', *arXiv preprint arXiv:1412.-6980*, 2014.

[97] T. Tieleman and G. Hinton, *Lecture 6e rmsprop: Divide the gradient by a running average of its recent magnitude*, 2014. [Online]. Available: `https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf` (visited on 30/11/2020).

[98] S. Unterstrasser, Personal communication, Sep. 2020.

[99] T. Gurke and H. Lafferton, 'The development of the wake vortices warning system for frankfurt airport: Theory and implementation', *Air Traffic Control Quarterly*, vol. 5, no. 1, pp. 3–29, 1997. DOI: `10.2514/atcq.5.1.3`.

[100] F. Holzäpfel and J. Kladetzke, 'Assessment of wake-vortex encounter probabilities for crosswind departure scenarios', *Journal of Aircraft*, vol. 48, no. 3, pp. 812–822, 2011. DOI: `10.2514/1.C000236`.

[101] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, 'Grad-cam: Visual explanations from deep networks via gradient-based localization', in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626. DOI: `10.1109/ICCV.2017.74`.

[102] S. Lang and C. Lunsford, 'Recat I: Lessons learned from mem', in *WakeNet - Europe Workshop*, 2013. [Online]. Available: `http://www.wakenet.eu/fileadmin/user_upload/Workshop2013/Presentations/WakeNet2013-04-Lang.pdf` (visited on 05/11/2020).

Appendices

A. RECAT phase one: wake categorisation

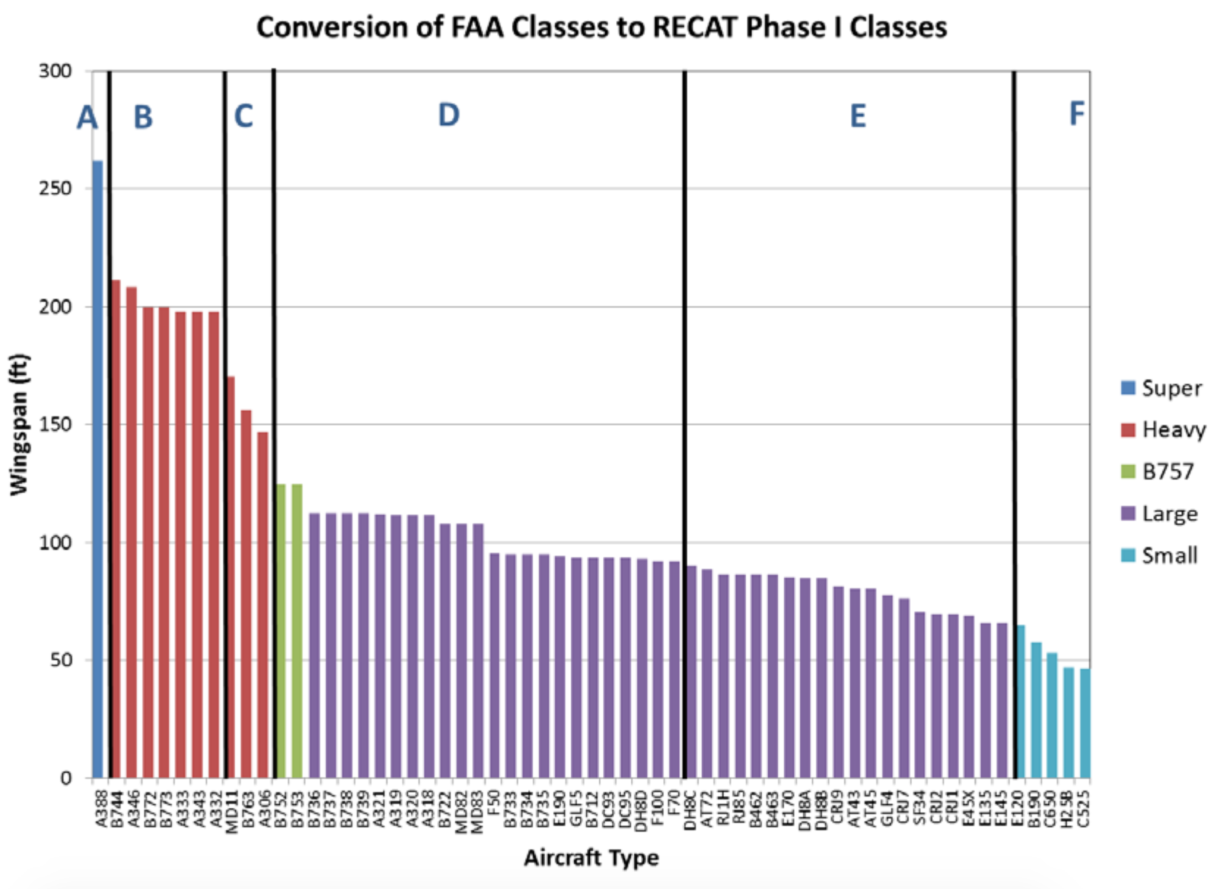


Figure A.1.: RECAT phase one aircraft separations: common aircraft split into categories (taken from [102], slide 5).

RECAT Separation Matrix

		Follower					
		A	**B**	**C**	**D**	**E**	**F**
Leader	**A**	MRS	5.0	6.0	7.0	7.0	8.0
	B	MRS	3.0	4.0	5.0	5.0	7.0
	C	MRS	MRS	MRS	3.5	3.5	6.0
	D	MRS	MRS	MRS	MRS	MRS	5.0
	E	MRS	MRS	MRS	MRS	MRS	4.0
	F	MRS	MRS	MRS	MRS	MRS	MRS

Figure A.2.: RECAT phase one aircraft separations: separation distances in nautical miles (comparison to conventional ICAO separations: green = increased, blue = decreased, white = no change) (MRS = Minimum Radar Separation) (taken from [102], slide 6).

B. Aircraft reaction to a Wake Vortex Encounter

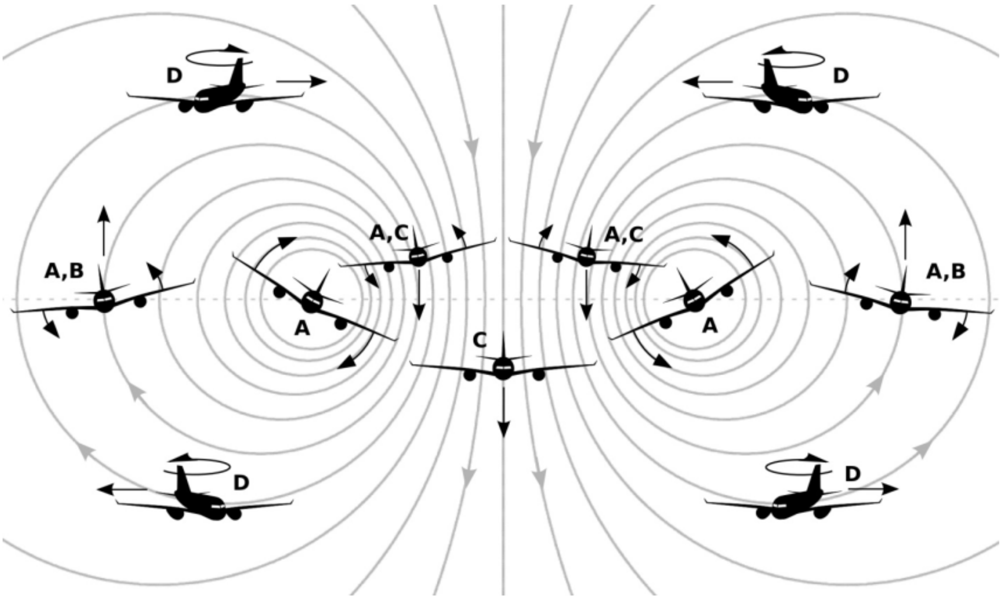


Figure B.3.: Aircraft behaviour with a vortex pair encounter. (A) induced roll, (B) upward motion, (C) downward motion, (D) induced yaw (taken from [23], p.33).

C. Multi-layered Artificial Neural Network backward pass

The process for reaching (4.8) is detailed in this section. The derivation is recapitulated from [76]. MLPs with multiple layers have interim-inputs and interim-outputs to and from each neuron, respectively. Two partial derivatives have to be calculated, the loss function with respect to the weight belonging to an arbitrary neuron (in order to update the layer which that neuron is part of) and the partial derivative of the loss function with respect to the input to that layer - the output of the previous layer (in order to update the parameters backwards) [80]. As seen previously, each neuron computes the weighted sum of its inputs. For the general neuron in a large network, observe (C.1). Here, the output of a neuron j is denoted a_j.

$$a_j = \sum_{i=1}^{I} z_i w_{ji} \tag{C.1}$$

An activation function is then applied to a_j, to give the activated z_j as shown with (C.2) [76].

$$z_j = A(a_j) \tag{C.2}$$

The chain rule is then used to compute the derivative of the overall loss function with respect to a weight w_{ji} from a hidden layer. The contribution of w_{ji} to L_n (the loss of one scan) is only dependent on a_j of this neuron, resulting in (C.3) [76]. For the mini-batch method, the individual gradients from each scan are summed, giving the mini-batch gradient.

$$\frac{\partial L_n}{\partial w_{ji}} = \delta_j \frac{\partial a_j}{\partial w_{ji}} \quad \text{where} \quad \delta_j \equiv \frac{\partial L_n}{\partial a_j} = \sum_{k=1}^{K} \frac{\partial L_n}{\partial a_k} \frac{\partial a_k}{\partial a_j} \tag{C.3}$$

As L_n does not directly depend on a_j, the chain rule is used in (C.3) (δ is referred to as error), consisting of the outputs of neurons in the output layer a_k, which directly influence the loss. The summation indicates that the interim-output influences each output of the output layer (K total outputs). By differentiating (C.1) with respect to w_{ji}, (C.4) is obtained.

$$\frac{\partial a_j}{\partial w_{ji}} = z_i \tag{C.4}$$

Therefore, upon combining (C.3) and (C.4) one obtains (C.5), showing that each neuron only requires δ_j to be computed once [76].

$$\frac{\partial L_n}{\partial w_{ji}} = \delta_j z_i \tag{C.5}$$

By combining the previous equations, one ends up with the backward pass equation, which gives the error coming from a single neuron by scaling the error of all output layer neurons ($'$ = derivative) [76]:

$$\delta_j = A'(a_j) \sum_{k=1}^{K} w_{kj} \delta_k \quad \text{where} \quad \delta_k = y_k - t_k \tag{C.6}$$

D. Vector regression architecture for Multilayer Perceptrons

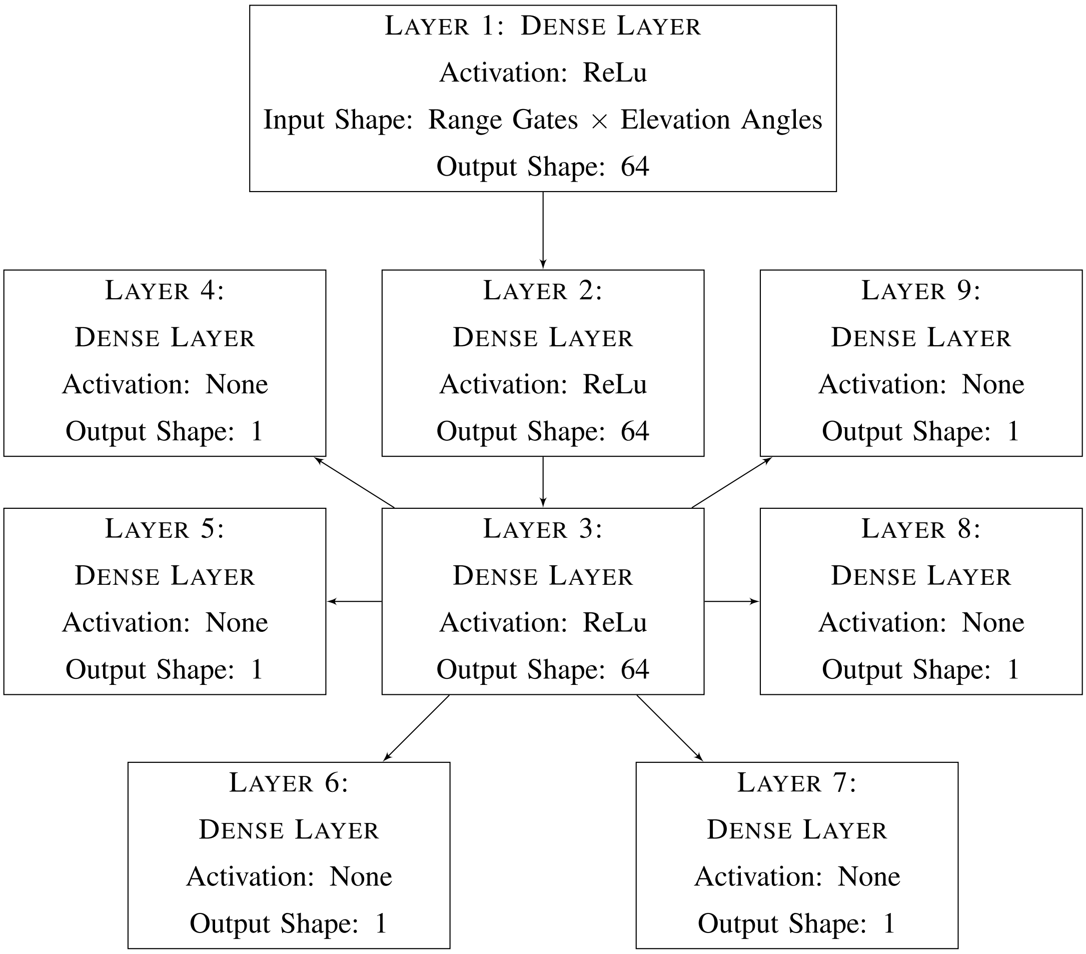


Figure D.4.: Vector regression MLP architecture.

E. Focal range of LiDAR used in the Vienna measurement campaign

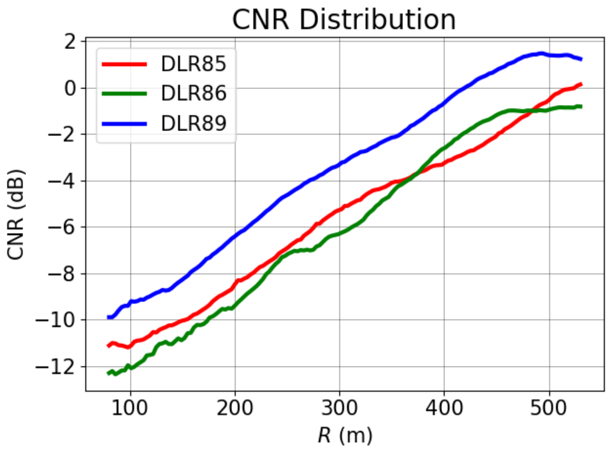


Figure E.5.: Median CNR for ranges of arbitrarily measured scans for all LOS and each LiDAR used in the Vienna campaign, indicating a plateau above 500 m (DLR85, DLR86 and DLR89 are DLR LiDAR serial numbers).

F. Python scripts

Table F.1.: Developed Python scripts for obtaining the final presented results (in order).

Script/Module name	Inputs	Outputs	Purpose
compare_detect-ions.py	Flight list, manual detections (targets), scan data (rad_vels).	Spreadsheet matching scan file names and targets for each individual wake vortex.	Script for the matching of LiDAR scans and targets for each vortex from the RV method. The campaign's flight list is used and for each target a corresponding scan is found by comparing recording times. Matches may be disregarded if the time of the target and scan are too far apart or if the corresponding scan for a target has not yet been processed.

Continued on the next page

Table F.1 – *Continued from the previous page*

Script/Module name	Inputs	Outputs	Purpose
matching_target.py	Output of above.	List of spreadsheets including scan names and targets for both vortices.	Script for the matching of vortex pairs for each scan. If for a certain scan, only one wake vortex exists, the parameters of the absent vortex are set to zero.
order_matches_for-_over-flight_position.py	Directory of output from matching_target.py.	New Directory with matches ordered for each overflight and LiDAR position.	The current script aims to order the match files for each overflight and LiDAR position to allow a chronological understanding of wake vortex development.
remove_erroneous-_series_data from modules_niklas.py	Directory with all overflights.	New overflights directory with no faulty scans.	The purpose of this function is to identify erroneous targets. This is done by looking at each overflight and LiDAR position individually. The targets of each match are inspected. If they are zero and following matches of that overflight are not, the scan is disregarded.
wind_subtraction-_from_radial_velo-cities.py	Flight list, directory with matches, scans directory.	Set of scans with removed wind.	This script's purpose is to load the pure radial velocity files, subtract the background wind for the specific flight and obtain radial velocity files with no wind distraction. First, the scans are converted to the data shape used by the LiDAR. Then, the scan before an overflight is selected and subtracted.
map_scans_onto-_overall_grid.py	Directory to matches and scans with no wind, global grid csv file.	Directory with newly interpolated scans.	This script uses a module from modules_niklas which ultimately interpolates each scan onto the global grid defined using a csv file.

Continued on the next page

Table F.1 – *Continued from the previous page*

Script/Module name	Inputs	Outputs	Purpose
remove_irrelevant-_SVS_LOS.py	Match directories, height threshold, LiDAR positions.	New scans.	This script automates disregarding relevant LOS from the scans of a directory using the module delete_SVS_LOS from modules_niklas.
ANN scripts.	Amount of scans, scan directory, match directory, potential orders.	Trained model, validate data set predictions, training metric history, metric figures.	This script is unique to each characterisation parameter and is detailed in Appendix G. Essentially, pre-processing, model training and evaluation is performed.

G. Typical Artificial Neural Network script

The following nested list provides an overview of the steps undertaken of a typical ANN script. In this case, a CNN is considered.

1. Load the required Python packages: Pandas, Numpy, OS, TensorFlow, Keras, Matplotlib, Scitkit-learn and Pathlib.
2. Using Pathlib, store directories for: the LiDAR scans, match files, overflight orders for the both train and validate data sets.
3. Using a single scan, obtain the unique number of elevation angles and ranges to later reshape all scans. These are stored in one-dimensional arrays in the LiDAR scan files.
4. Separately for the train and validate data set, for each match:
    a) Load the scan, its name and the LiDAR which recorded it.
    b) For this scan, jointly re-arrange the elevation-range measurements in descending elevation angles and for each LOS in ascending ranges. This is comparable to how images are stored.
    c) Convert the radial velocity array of the scan to a numpy array and replace all ‘Not a Number’ (NaN) by 0.
    d) Obtain the target for the parameter from the match file.
5. Data pre-processing:

a) Independently shuffle train and validate data sets, where scans and targets remain pairs.
b) Save the mapping of the scans.
c) Perform scan-wise normalisation:
   i. Obtain the mean radial velocity of all radial velocities in the train data set.
   ii. Subtract this mean radial velocity from all train and validate data set measurements.
   iii. Obtain the standard deviation of all train radial velocities.
   iv. Divide this standard deviation from all train and validate data set measurements.
   v. Store the mean and standard deviation for later usage.
d) Reshape each LiDAR scan to a tensor with rows equivalent to the elevation angle count and columns equivalent to the range count. The depth is one.

6. Initialise Keras callbacks to enable network training to stop early and save the best model based on the validate loss.
7. Implement the CNN architecture:
   a) Define a function build_model: build the model using Keras tools in sequential mode. Add the layers from Figure 4.14 using the function: model.add().
   b) Use the above function to implement the model into the computer's memory.
   c) Use the fit function to train the model using selected metrics, epoch count, batch size, callbacks as well as the train and validate data sets.
   d) Store the results of the function as a history for post-processing.
8. Evaluation:
   a) Load the saved model.
   b) Use the predict and evaluate functions to obtain characterisation estimations by the model.
   c) Using the saved history, find the lowest validate loss and the other metrics for that same epoch.
   d) Plot the history if desired.