UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

Field Calibration Development and Comparative Evaluation of Machine Learning and Physics-based Wind Estimation Methods for the Weather-sensing CopterSonde UAS

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

Master of Science

By

ABDULLAH AL TASIM
Norman, Oklahoma
2025

Field Calibration Development and Comparative Evaluation of Machine Learning and Physics-based Wind Estimation Methods for the Weather-sensing CopterSonde UAS

A THESIS APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. David Schvartzman, Chair

Dr. Robert D. Palmer

Dr. Antonio R. Segalés

Dedicated to Abu Reza Mohammad Hasnatul Islam and my wife, Hafsa Mehrin Mahi

Acknowledgments

To begin, I would like to extend my deepest gratitude to my advisor, Dr. Antonio R. Segalés, for providing me the opportunity to pursue my master's degree as part of an interdisciplinary team of inspiring scientists and researchers. His guidance, motivation, and expertise helped shape the foundation of my research and allowed me to envision impactful contributions to the atmospheric science community. I am especially thankful for his mentorship, not only for the scientific insights that enriched my work but also for the personal support he offered during challenging times. His encouragement was instrumental in the timely completion of this thesis.

I would also like to sincerely thank the other members of my master's committee for their invaluable time, constructive feedback, and continuous support throughout the development and evaluation of this work.

I gratefully acknowledge the financial and institutional support provided by the University of Oklahoma (OU), the National Science Foundation, and the Cooperative Institute for Severe and High-Impact Weather Research and Operation (CIWRO). Special thanks to the exceptional teams at CIWRO and the National Severe Storms Laboratory (NSSL) for granting access to their facilities and instrumentation, without which this research would not have been possible.

I am particularly thankful to my other mentors, especially Dr. Tyler Bell, Dr. Joshua G. Gebauer, and Dr. Elizabeth Smith, for their collaboration, encouragement, and insightful discussions throughout this journey. It was truly an honor to work alongside them, and the experience has been both professionally rewarding and personally memorable.

I am especially grateful to my elder brother, Abu Reza Mohammad Hasnatul Islam, without whose unwavering support and guidance it would not have been possible for me to come to the United States and pursue higher education. I am equally thankful to my wife, Hafsa Mehrin Mahi, whose love, patience, and encouragement have been constant sources of strength throughout my academic journey. I also wish to express my sincere gratitude to my parents, parents-in-law, and my brothers for their continued support and prayers, without which completing this work would not have been possible. Finally, I extend heartfelt appreciation to my close friends at the University of Oklahoma, who became my second family and provided steadfast emotional support during my time here.

Finally, I would like to thank my friends back home for their unconditional support, encouragement, and belief in my aspirations. Their faith in me made it possible to pursue and complete this chapter of my life, even from afar.

Table of Contents

List of Figures

Abstract

The precise measurement of wind within the planetary boundary layer (PBL) is essential for enhancing weather forecasts, comprehending atmospheric phenomena, and facilitating various applications,. Traditionally, researchers have depended on radiosondes, tower-mounted instruments, and remote sensing technologies like Doppler lidar to get kinematic and thermodynamic characteristics of the lower atmosphere. While these methodologies yield significant data, they are frequently limited by cost, mobility, or temporal resolution, resulting in limitations in our knowledge of the PBL's intricate and swiftly changing circumstances. Uncrewed Aircraft System (UAS) present a viable option owing to their adaptability, cost-effectiveness, and capacity for precise and targeted in situ measurements. Nevertheless, affixing wind sensors to small drones may result in weight and aerodynamic drawbacks, in addition to being vulnerable to disturbances or flow distortion around the aircraft. The CopterSonde, a drone developed by meteorologists and engineers at the University of Oklahoma, has been designed and optimized for kinematic and thermodynamic measurements of the PBL. Unlike conventional multicopter platforms, the CopterSonde employs a model-based 3D wind estimation technique that relies on mathematical models. This method uses aerodynamic principles to infer wind vectors from the drone's own response against wind and control inputs. Early phases of the presented study, building on the foundation of previous works, focused on refining parameters of the model-based methods by calibration through the use of Differential Evolution Optimizer (DEO), a cost function optimization technique.

Calibration was performed using a combination of Doppler wind lidars, providing a robust reference for validation. These efforts highlighted the DEO method's ability to optimize the CopteSonde's dynamic model, yielding wind estimates with accuracy comparable to Doppler wind lidars. by minimizing the root mean square error (RMSE) between the ground truth and the CopterSonde's estimations. The integration of a Linear Extended State Observer (LESO) into the model helped to reduce the influence of autopilot-induced perturbations and enhance the overall quality of the wind estimates.

Extending upon this previous research, the presented work introduces advanced machine learning (ML) approaches to avoid reliance on complex mathematical models, offering a platform-agnostic and data-driven characterization of UAS for wind estimation. A selection process to determine an adequate machine learning method was described and implemented. After evaluating several candidate techniques based on predictive accuracy and computational efficiency, machine learning techniques were employed to analyze and learn patterns and relationships in an extensive pre-collected dataset that model-based methods fail to capture. ML methods may offer a better way to distinguish wind perturbations from coupled UAS dynamics. The resulting ML models not only refined the CopterSonde's 3D wind estimation capabilities but also provided a new perspective on how to integrate data-driven methods with mathematical models.

The combined efforts in this study present a comprehensive evaluation of various wind estimation techniques, from initial drag coefficient refinement using DE to advanced ML implementations. This holistic approach not only highlights the progress made in enhancing wind-sensing accuracy but also provides valuable insights and guidelines for future applications to avoid reliance on complex mathematical models, offering a platform-agnostic and data-driven characterization of UAS for wind estimations. Through a comparison of different methodologies, the work establishes a solid foundation for ongoing research in wind estimation and sets the stage for future advancements in UAS technology.

Chapter 1

Introduction

1.1 Motivation

The study of air mass kinematics and dynamics is essential for advancing understanding of the planetary boundary layer (PBL) [1]. These are essential for predicting meteorological processes [2], optimizing wind energy performance [3], understanding wake interactions within the planetary boundary layer (PBL) in large wind farms [4–6], and defining boundary conditions for modeling gas dispersion in the PBL [7]. Commonly used systems for obtaining wind velocity data at different heights within the PBL include traditional mast-mounted anemometers (primarily cup and sonic types), weather balloons (radiosondes), sonic detection and ranging (SODAR), and light detection and ranging (lidar) [1]. These methods are suitable for measuring wind velocities across various temporal and spatial scales, producing complementary datasets that are frequently compared [8].

Despite the wide range of existing measurement techniques, observational gaps remain in wind data collection within the planetary boundary layer (PBL), driving the development of small UASs equipped with sensors for measuring wind velocities [2, 9–13]. UASs offer three-dimensional mobility, ease of deployment, cost-effectiveness, and operational flexibility while providing time- and space-resolved wind measurements [14]. Moreover, advancements in fabrication and prototyping methods, such as the accessibility of 3D printing technologies and open-source autopilot codes, have made UAS development more affordable, user-friendly, and rapid. They are also highly versatile and can be equipped with additional sensors to map various atmospheric properties in three-dimensional space. UASs have become essential tools for addressing observational gaps in the PBL by providing high-resolution, in-situ measurements with greater flexibility than traditional methods. Unlike meteorological towers or weather balloons, UASs enable continuous profiling across various altitudes and terrains, including inaccessible or hazardous regions [12, 13]. Their capability of carrying payload like sensor packages to measure key atmospheric parameters, such as temperature, humidity, pressure, and wind speed, has been demonstrated in field campaigns such as LAPSE-RATE, where multiple UAS systems effectively captured fine-scale meteorological variations [13]. Small UASs, such as the Multi-purpose Airborne Sensor Carrier (MASC), have proven particularly valuable for studying turbulence, wind shear, and wake structures, making them integral to wind energy research and boundary-layer studies [15]. Advancements in sensor integration and wind estimation algorithms have improved the accuracy of UAS-based observations, enabling precise assessments of PBL dynamics [12]. Additionally, UASs provide a cost-effective and rapidly deployable alternative to conventional monitoring systems. Despite challenges related to sensor calibration and flight optimization, recent studies has demonstrated UASs to be a promising bridge to fill up critical gaps in PBL observations and enhance the understanding of boundary-layer processes [12, 13, 15]. Consequently, UASs are expected to contribute valuable data to PBL research [11–13, 15, 16].

Atmospheric wind kinematics can be determined using two different approaches in UASs. Direct methods rely on dedicated wind sensors mounted on the UAS [1]. Various types of sensors have been employed to measure wind speeds using UASs.

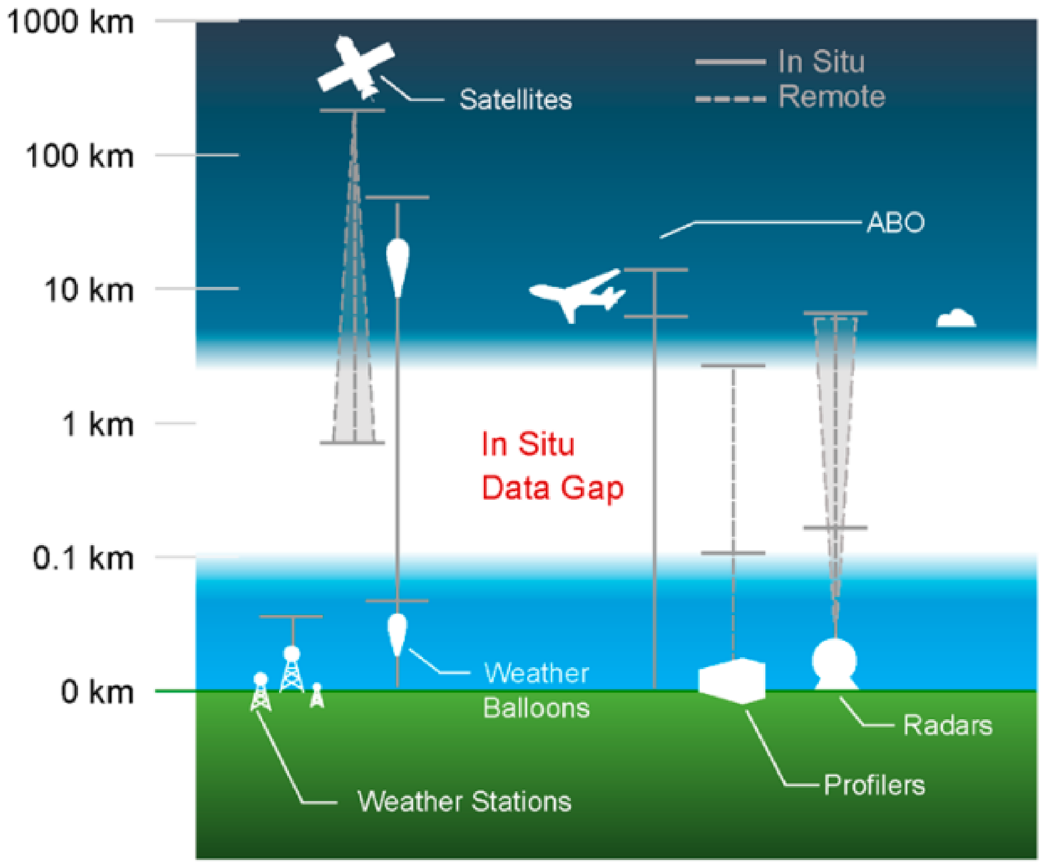

Figure 1.1: Schematic showing the in situ data gap present in the lower atmosphere. Conventional meteorological instruments are illustrated with their respective measurement type and range. ABO stands for Airborne Observations using manned aircraft. Image was taken from [17].

Differential pressure sensors, such as Pitot tubes and multi-hole pressure probes, are particularly suitable for wind measurements covering large areas, especially when the UAS maintains a constant forward motion at elevated speeds. However, strong turbulence has been found to reduce measurement accuracy, as Pitot tube readings depend on dynamic pressure management, introducing additional complexity to UAS operation [12]. Mechanical anemometers (e.g., cup anemometers) are rarely used on UASs.

Most mechanical anemometers cannot be used for accurate measurements of a 3D wind vector [1]. Cup anemometers, for instance, are highly susceptible to interference from airflow disturbances caused by the UAS's nearby rotors, which can significantly alter the local wind field and introduce measurement errors. Additionally, mechanical anemometers generally have a low response time, making them less suitable for capturing rapid fluctuations in wind velocity, particularly in turbulent conditions [1].

Arrays of multiple sensors in different orientations would need to be analyzed, resulting in a bulky setup and a large measurement volume. Additionally, the low temporal resolution of mechanical anemometers makes them unsuitable for capturing rapid wind velocity fluctuations, particularly in the dynamic environment of a flying UAS [18]. Hotwire anemometers can be used for high-frequency wind speed measurements; however, their accuracy deteriorates beyond 7 kHz, and the fragility of these sensors increases the handling difficulty of rapidly deployable UASs in harsh environments [19]. Sonic anemometers mounted on rotary-wing UASs have demonstrated accurate wind measurements; however, time series data collected along a horizontal circular trajectory introduce variability in wind speed measurements due to the continually changing orientation of the aircraft during each orbit and potential inhomogeneities in the wind field [20]. Historically, full-sized sonic anemometers were rarely used on UASs due to size and weight constraints [10]. An exception is the deployment of a full-sized 1.7 kg sonic sensor (R.M. Young 81000 % 3D ultrasonic anemometer) on a large, 12 kg commercial multirotor (DJI M600). However, data from this setup indicated significant inaccuracies under challenging conditions, with wind speed measurements deviating by up to 50 % and azimuth errors reaching 15 ° compared to a lidar reference [21]. Wind measurements obtained from sensor-equipped UASs are further compromised by the high contribution of vertical flow, even when the sensor is mounted on a stabilizing gimbal [1]. Indirect methods estimate wind characteristics by analyzing the UAS's response to airflow, determining wind speed, azimuth, and elevation based on onboard sensors primarily used for flight control. However, these methods require precise knowledge of the UAS's inertia and drag coefficients across various flight conditions, which are challenging to determine but significantly impact measurement accuracy. A key advantage of this approach is the potential to extend flight duration and increase the maximum sustainable wind speed, as no additional wind payload is required [2, 22–25].

Given the complexities associated with wind measurement using physical wind sensors, engineers and meteorologists at the University of Oklahoma developed the CopterSonde, a weather-sensing UAS that employs a dynamic model-based 3D wind estimation approach without relying on physical wind sensors. By eliminating additional payloads, this design enabled the development of a more compact UAS with an improved power-to-weight ratio, enhancing its tolerance to higher wind speeds. Instead of physical wind sensors, the CopterSonde utilizes mathematical models of its body and propellers to estimate external forces acting on the CopterSonde [26]. A Linear Extended State Observer (LESO) incorporating a modified ideal Rayleigh equation was employed to compute external force estimates using motion data from the autopilot, which is subsequently translated into wind measurements. While previous studies have demonstrated the effectiveness of LESO at lower wind speeds, its performance declines as wind speed increases due to nonlinearities in the relationship between aerodynamic drag and wind velocity, as well as extrapolation errors resulting from the limited wind speed range used for calibration. Additionally, as a mathematical model, the LESO relies on inherent assumptions, which may further contribute to estimation errors [26].

In recent years, machine learning techniques have increasingly been adopted as powerful tools in environmental sciences. Numerous applications demonstrate the use of these methods for forecasting meteorological parameters and their derivative products. Machine learning has become widely employed not only to correct biases arising from the high non-linearity of wind dynamics but also as stand-alone predictive models [27]. Several examples highlight the application of machine learning in this field. For instance, neural networks were used in 2020 to generate monthly wildfire predictions, providing valuable insights for long-term fire planning and management [28]. Additionally, incorporating principal component analysis (PCA) improved visibility prediction accuracy in Sichuan [29]. Demonstrating the potential of machine learning, a classification approach for wind gust occurrences successfully enhanced prediction accuracy [30]. Despite these advancements, predicting wind speed remains challenging due to its inherently unpredictable nature. A study conducted in 2024 found that Long Short-Term Memory (LSTM) networks outperformed other machine learning algorithms in wind estimation. However, LSTM was observed to be limited in its effectiveness at higher wind speeds and altitudes [27].

In order to find the estimation approach that is best suited for this particular UAS, the CopterSonde, the primary purpose of this work is to continue evaluating alternative estimation methods. The Doppler wind lidars, an Oklahoma Mesonet tower, and radiosondes, provided by the National Severe Storms Laboratory and supported by the National Oceanic and Atmospheric Administration at the National Weather Center of the University of Oklahoma, are the components that make up the calibration and validation dataset. The innovative method that is introduced in this work is an Artificial Neural Network (ANN), which has the ability to learn detailed patterns in wind dynamics that traditional models might not be able to capture. This approach leverages a variety of machine learning techniques to develop a new method for wind estimation without relying on wind sensors or dynamic models. The goal is to create a calibration and implementation process that is more straightforward while maintaining or improving wind measurement performance and quality. Neural network models have the ability to capture the extremely non-linear dynamics of the CopterSonde UAS and produce a more accurate prediction of the wind flow around the platform. In this work, a comprehensive comparison is made between all of the approaches that have been described, and suggestions for putting them into practice are provided.

1.2 Literature Review

UASs have been increasingly utilized in several civil applications due to their ease of deployment, low maintenance costs, high mobility, and hovering capability [31]. As a result, new opportunities have emerged for exploring novel solutions in fields such as agriculture, land surveying, surveillance, and package delivery [32].

In meteorology, severe weather events—including thunderstorms with hail and strong winds, tornadoes, extreme rainfall, floods, tropical storms, ice storms, heavy snowstorms, and blizzards—cause billions of dollars in economic losses annually in the United States alone [33–35]. The impact of severe weather, exacerbated by climate change [36, 37], extends to biodiversity [38] and public health [39], particularly affecting vulnerable populations [40]. To mitigate societal and infrastructural damage, weather researchers have emphasized the need for developing novel atmospheric monitoring techniques [41].

Among these advancements, the integration of UASs into weather research has been widely recognized as a promising approach to address the existing in situ observational gap in the lower atmosphere. Recent technological advancements have facilitated controlled and targeted atmospheric soundings, generating momentum within the scientific community to further enhance observational capabilities [42]. Furthermore, in 2019 %, the National Weather Service (NWS) released a strategic plan advocating the adoption of cutting-edge science, technology, and engineering to enhance meteorological observations, forecasting, and warning systems [43]. Similarly, the National Severe Storms Laboratory (NSSL) has outlined a vision for integrating large volumes of weather data from multiple sources while ensuring scalability to accommodate future measurement techniques. This initiative, known as Multiple Radar/Multiple Sensor (MRMS), aims to improve atmospheric monitoring and forecasting capabilities [44, 45].

With the increasing use of UASs for atmospheric data collection and the evolving challenges in modern meteorological research, several recent collaborative field experiments have encouraged researchers and engineers to assess the effectiveness of UASs in weather measurements while identifying challenges to improving measurement accuracy [20, 46–48]. This initiative has driven the development of innovative UAS designs for atmospheric sampling [49, 50] and has fostered discussions on future operational concepts [51] and research collaborations [52, 53], particularly concerning the application of UASs in meteorological studies.

The earliest recorded attempts at utilizing UASs for atmospheric research date back to the 1970s, when Konrad proposed the use of hobby-level radio-controlled aircraft to measure weather parameters on convective days [54]. However, progress in this field was limited at the time due to the bulky and heavy meteorological sensors, as well as the absence of autopilot-assisted flights. Despite these technological constraints, this early work represented a significant milestone in demonstrating the feasibility of the concept, highlighting the unique advantages of UASs over conventional instruments, and establishing preliminary operational procedures for their use in atmospheric research.

One of the first UASs designed for planetary boundary layer (PBL) measurements was the Small Unmanned Meteorological Observer (SUMO), an autonomous fixed-wing system developed at the University of Bergen as a cost-effective atmospheric measurement platform. The SUMO was first deployed in 2007 during the FLOHOF field campaign in Iceland, demonstrating atmospheric measurement results comparable to those obtained from conventional radiosondes. Additionally, SUMO is recognized as one of the first open-source UASs capable of recording both weather and flight data through its autopilot system.

SUMO integrates inertial measurement unit (IMU) readings (accelerations and angular rates) with GPS velocity and multi-hole probe measurements within a filtering framework, typically an extended or unscented Kalman filter, to estimate the 3D wind vector. However, its primary drawback is the demanding operational requirements, including the necessity for a large, open, and unobstructed space for takeoff and landing, as well as the need for skilled operators to pilot the system effectively [50, 55].

Commercially available UAS-based solutions exist today. Meteomatics AG, a company based in Switzerland, has been developing and offering the weather UAS "Meteodrone" since 2012. The onboard sensor suite integrates IMU data, GPS, and pressure readings in real time to infer wind speed and direction. The collected data can be assimilated into numerical weather prediction models, such as the Weather Research and Forecasting (WRF) model, or specialized short-range forecast models. However, its functionalities are restricted to those provided by the manufacturer, and access to raw data may be unavailable due to trade secrets, potentially limiting its applicability for research [56].

The Coyote drone, developed by Raytheon and deployed by NOAA, is an air-launched small UAS (sUAS) designed for hurricane observations. It provides high-resolution wind, pressure, temperature, and turbulence data at low altitudes ($<$150 m), where manned aircraft cannot safely operate. Wind measurements are obtained using GPS-derived velocity, inertial motion sensors, and a pitot-static system, enabling turbulence and momentum flux calculations. The Coyote has been successfully deployed in Hurricanes Maria (2017) and Michael (2018). However, its short flight duration ($\sim$1 hour), limited payload (1.8 kg), reliance on a NOAA P-3 aircraft for deployment, and expendable nature pose significant challenges [57].

The NASA Global Hawk estimates wind using a combination of in situ and remote-sensing instruments, including Meteorological Measurement Systems (MMS) for GPS-and inertial-based wind tracking, Microwave Temperature Profilers (MTP) for temperature gradient analysis, Cloud Physics LIDAR (CPL) for aerosol and cloud motion characterization, and dropsondes for vertical wind profiling. With its long-endurance capability (31+ hours) and high-altitude range (up to 65,000 ft), it facilitates extensive atmospheric observations over remote regions, making it highly valuable for hurricane monitoring, climate studies, and upper-atmosphere research. Advantages include autonomous operation, a large payload capacity (1,500 lbs), and the ability to access extreme weather conditions without endangering human pilots. However, challenges include high operational costs, lengthy mission planning (up to six months for FAA approval), limited low-altitude wind measurements, and dependence on ground control for real-time adjustments [58, 59].

The CopterSonde estimates wind using an integrated sensor fusion approach, combining inertial measurement units (IMUs), GPS, onboard pressure sensors, and a wind vane mode algorithm. This system enables the UAS to maintain orientation into the wind while calculating wind speed based on pitch and roll angles, eliminating the need for dedicated anemometers. The wind estimation algorithm employs a kinematic model, utilizing the UAS's tilt to derive wind velocity, with post-processing refining these measurements through calibration against reference meteorological stations. The CopterSonde offers several advantages, including high-resolution vertical profiling capability, adaptability to real-time atmospheric conditions, modular payload configurations for sensor flexibility, and efficient low-altitude wind measurements, making it particularly suitable for boundary layer research. However, limitations include battery constraints leading to short flight durations ($\sim$ 60 min), susceptibility to turbulence and rotor wash effects, limited spatial coverage per flight, and operational restrictions due to visual-line-of-sight regulations. Despite these challenges, the CopterSonde's precise wind estimation, cost-effective deployment, and autonomous adaptability make it a valuable tool for meteorological studies and atmospheric profiling [26, 60].

Limited research has been conducted in this domain to estimate wind using machine learning. In 2020, scholars applied a long short-term memory (LSTM) recurrent neural network, leveraging Doppler lidar observational datasets from three sites over complex terrain. The model was trained using wind speed and direction data from 40 m and 60 m, along with turbulence kinetic energy (TKE) and diurnal cycle information, to predict wind profiles up to 230 m. The LSTM approach outperformed the traditional power-law method, particularly at greater heights, achieving an R^2 above 90 % up to 100 m when using only the 40 m input data, with further improvements observed when incorporating 60 m wind speed data. However, limitations included the model's sensitivity to site-specific training data, necessitating retraining for new locations, and its reduced accuracy in capturing abrupt wind profile changes. Additionally, complex terrain effects, local wind patterns such as low-level jets, and data availability constraints posed challenges for generalizing the model across diverse environments [27]. Another study in 2020 applied an LSTM neural network to estimate wind velocity using quadcopter state measurements without a wind sensor, relying on roll and pitch angles and position data as inputs. However, this study was based on simulated quadcopter flights using the Dryden gust model, a stochastic wind model, and Large Eddy Simulation (LES) of the planetary boundary layer. The approach was constrained by its dependence on specific flight paths, systematic biases in certain cases, high computational costs, and the lack of real-world validation data [61].

Research on estimating three-dimensional (3D) wind fields using machine learning remains limited, particularly in conjunction with UASs for vertical profiling in the planetary boundary layer (PBL) and traditional weather measurement instruments such as radiosondes, mesonets, and lidars in open-field environments. Most existing studies focus either on wind estimation in the PBL or rely on a single measurement method, often lacking an integrated approach that combines remote sensing and in situ observations. This thesis addresses this gap by integrating machine learning techniques with UAS-based wind profiling alongside conventional meteorological instruments in real-world conditions. Significant efforts have been made to develop a comprehensive framework that leverages both observational data sources and machine learning models to enhance wind estimation accuracy.

1.3 Research Scope

After establishing the motivation and reviewing the relevant literature, it was determined that significant potential exists to improve both horizontal and vertical wind estimation using the CopterSonde weather-sensing UAS. The current approach for wind estimation relies on the Linear Extended State Observer (LESO), a mathematical model based on various assumptions that may constrain its applicability. Leveraging machine learning presents an opportunity to enhance wind estimation accuracy by reducing dependence on these assumptions. Consequently, the primary objectives of this research are outlined as follows:

1. Enhance and refine the calibration of existing wind estimation methods across two distinct platforms: CS 3D and CSWX WxUAS.

2. Conduct a performance analysis of current methodologies, considering platform-specific differences and operational constraints.

3. Investigate machine learning techniques capable of capturing the complex interactions between flight control dynamics and wind effects.

4. Perform a comprehensive evaluation of all algorithms, including machine learning approaches, to assess performance trade-offs and operational feasibility.

The UASs utilized in this thesis are designed for weather research. The first stage of this study focused on collecting wind data using CopterSondes. Given that a wind tunnel was not available at the university, the work relied on alternative meteorological instruments, including Doppler wind lidar, a 3D mesonet tower, and radiosondes, provided by the Cooperative Institute for Severe and High-Impact Weather Research and Operations (CIWRO) at the NSSL, located at the National Weather Center. As data collection was conducted in an open-field environment, it was dependent on weather conditions and the calibration of the CopterSondes.

The instruments were assumed not to be colocated. Consequently, it was presumed that large-scale wind motions were equivalent across measurement locations, while small-scale features and turbulence were considered negligible, as the objective was to capture the mean structure of horizontal winds and small-scale variations in vertical winds. This dataset facilitated the development of a Differential Evolution Optimizer (DEO) function, a cost function optimizer that determined the optimal aerodynamic force coefficients acting on the UAS during ascent and descent. These coefficients were then integrated into the LESO algorithm to compute aerodynamic forces on the CopterSonde and convert them into wind velocity estimates. Additionally, other perturbation-based approaches, specifically the Linear and Quadratic algorithms, were evaluated.

In the second phase, the collected dataset was utilized alongside DE and ground-truth data to optimize coefficients for the Linear and Quadratic models. After deriving the optimal coefficients for the LESO, Linear, and Quadratic approaches, a comprehensive RMSE comparison against the ground truth was conducted to determine which model most accurately predicted wind velocity.

Once the most effective algorithm was identified, the next step was to develop a machine learning model capable of estimating both horizontal and vertical wind using only the UAS's dynamic information (i.e., flight telemetry). The objective was to surpass the accuracy of existing wind estimation methods while reducing dependence on complex or rigid mathematical assumptions.

As the proposed solution is based on an experimental prototype, some minor limitations may remain unresolved. Therefore, the final stage of this research outlines a roadmap for future work, including discussions on potential improvements, existing limitations, and additional strategies to further refine the current weather UAS concept and enhance its operational capabilities.

1.4 Chapter Summary

This chapter underscores the importance of accurately characterizing wind kinematics and dynamics within the PBL for applications such as meteorological forecasting, air-quality assessment, and wind estimation. While traditional measurement platforms—including mast-mounted anemometers, radiosondes, SODAR, and lidar—offer valuable observational datasets, they often leave spatial and temporal gaps in the lower atmosphere. To address these limitations, small UASs have emerged as promising tools for high-resolution, three-dimensional wind profiling.

Following a review of direct and indirect sensing techniques, the CopterSonde platform, developed at the University of Oklahoma, is introduced. This system avoids the use of dedicated wind sensors by employing a dynamic model–based estimator that infers external wind forces from the UAS's own motion. Although such perturbation-based algorithms perform adequately at low wind speeds, they rely on simplifying assumptions and platform-specific calibration. These limitations reduce their accuracy under more challenging atmospheric conditions, motivating the development of alternative estimation strategies

A comprehensive literature review traces the evolution of UAS-based atmospheric profiling, from early hobby-grade aircraft experiments to advanced autonomous platforms such as SUMO, commercial systems like Meteodrone, and NOAA's deployment of the Coyote UAS. Recent efforts to integrate machine learning—particularly using neural networks—have shown potential for wind estimation in lidar-assisted and simulated environments. However, these approaches remain limited by site-specific training, lack of validation against real-world datasets, and insufficient integration with conventional remote sensing platforms such as lidar, mesonet towers, and radiosondes.

The chapter concludes by outlining the objectives of this thesis: to refine the calibration of the CopterSonde platform, evaluate traditional dynamic and kinematic estimation algorithms using Differential Evolution optimizer, and develop artificial neural networks that capture the platform's nonlinear dynamics for improved horizontal and vertical wind estimation.

The structure of this thesis is as follows. Chapter 2 provides a detailed description of the instrumentation used in this study, including the UAS platforms and ground-based remote sensing systems. Chapter 3 presents a comparative analysis of established baseline algorithms for wind estimation. Chapter 4 introduces the proposed machine learning framework for UAS-based wind estimation, highlighting model development, training strategies, and evaluation. Chapter 5 concludes the thesis by summarizing the key findings and outlining potential directions for future research.

Chapter 2

Description of Supporting Instruments

This chapter provides a comprehensive overview of the instruments utilized in this study. The discussion begins with the earlier generation of the CopterSonde, designated as CS3D, and subsequently introduces the latest version, CSWX, which was specifically engineered for deployment in severe weather conditions. The chapter then details the DWL, which served as the primary ground truth instrument for wind profile validation as the experiments were carried out in an open field environment. Two DWL systems were deployed which were provided by the National Weather Center (NWC) at the University of Oklahoma (OU): one system was operated by the Cooperative Institute for Severe and High-Impact Weather Research and Operations (CIWRO) and the other by the National Severe Storms Laboratory (NSSL) [62].

2.1 Evolution of the CopterSonde Platform

The CopterSonde UAS was developed to enable high-resolution vertical profiling of thermodynamic and kinematic atmospheric properties, with a particular focus on the planetary boundary layer (PBL). Since its initial deployment in 2017, the platform has undergone substantial development, resulting in two primary iterations: the CopterSonde-3D (CS3D) [63] and the CopterSonde-SWX (CSWX), the latter introduced as a prototype in 2024 and currently in the testing phase. While both platforms are designed for meteorological observations, the CSWX incorporates significant improvements in aerodynamic design, propulsion systems, and operational robustness, making it better suited for deployment in severe weather environments.

2.1.1 CopterSonde-3D UAS

The CS3D platform was developed through iterative design improvements upon learning from field campaign deployments where observations allowed for aerodynamic evaluations of sensor placement [26, 64]. The system features a rotary-wing quadcopter architecture with a modular weather sensor payload , referred to as the "scoop," which houses thermodynamic sensors shielded from direct solar radiation and actively aspirated using a fan. The sensor array is mounted at the front of the airframe and actively oriented into the incoming relative wind using a custom wind vane flight mode (WVFM), thereby minimizing platform-induced thermal biases [63].

Figure 2.1 illustrates the CS3D design concept developed in early 2022, along with a sectional view of the front shell from the computer-aided design (CAD) model. The scoop structure and integrated sensor package are visibly incorporated into a streamlined airframe to promote consistent airflow over the measurement elements.

The CS3D platform is equipped with a Wind Vane Flight Mode (WVFM) that passively aligns the airframe with the prevailing wind direction during flight. This alignment ensures that the sensor array remains exposed to undisturbed ambient airflow, which is critical for accurate retrieval of horizontal wind speed and direction, as well as precise thermodynamic measurements such as temperature and humidity. By minimizing aerodynamic disturbances around the sensing elements, the WVFM substantially improves the fidelity and reliability of the CS3D's atmospheric observations during

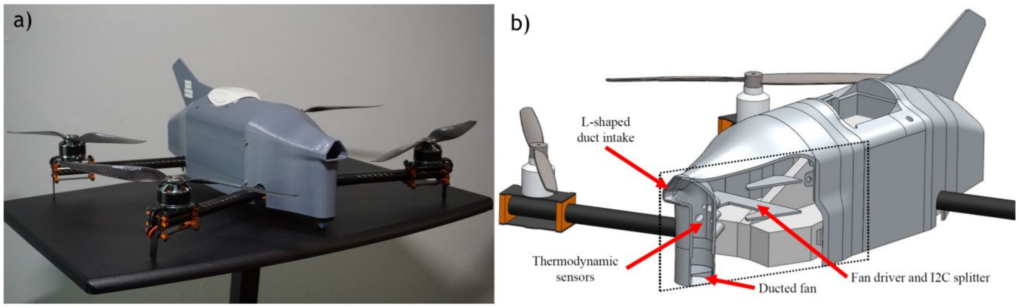

Figure 2.1: (a) Conceptual design of the CopterSonde-3D quadcopter UAS from early 2022, configured for in situ thermodynamic measurements and vertical profiling of the lower atmosphere. (b) Cross-sectional view of the front fuselage from the CopterSonde-3D CAD model, showing the solar radiation shield and internal sensor bay. Positions of thermodynamic instruments and other essential components are indicated, with the dashed box marking the section plane. Image is adapted from [63].

vertical profiling missions.

Table 2.1 presents the complete set of CS3D specifications, including its airframe properties, propulsion components, and operational performance metrics as estimated and stated by [63].

Although the aerodynamic design of the CopterSonde-3D represented a substantial advancement over its earlier iterations, several operational limitations were observed during its deployment in the PERiLS 2022 field campaign. The most critical constraint emerged under persistently elevated wind conditions, which frequently activated the platform's automatic return-to-launch (RTL) failsafe mechanism. This safety protocol was triggered when the aircraft's control stability or GPS accuracy became compromised due to external wind forces exceeding its design thresholds. As a result, the CS3D was unable to ascend beyond altitudes of approximately 1000 m above ground level (AGL), thereby restricting the vertical extent of atmospheric profiles during several critical observation windows [63].

Despite demonstrating limited success in light precipitation events—where the elec-

Table 2.1: Airframe, propulsion components, and operational specifications of the CS3D platform.

Category	**Specification**
Operational Performance	
All-up Weight	2.36 kg
Hover Time	18 min
Maximum Forward Speed	28.4 $\mathrm{m\,s^{-1}}$
Maximum Sideways Speed	22.3 $\mathrm{m\,s^{-1}}$
Maximum Sustained Wind Speed	22 $\mathrm{m\,s^{-1}}$
Maximum Wind Gust Tolerance	26 $\mathrm{m\,s^{-1}}$
Maximum Climb Rate	12.2 $\mathrm{m\,s^{-1}}$
Maximum Descent Rate	6.5 $\mathrm{m\,s^{-1}}$
Maximum Input Voltage	17.5 V
Minimum Input Voltage	12.5 V
Maximum Peak Current	94.4 A
Maximum Continuous Current	61.5 A
Maximum Mechanical Vibration	27 $\mathrm{m\,s^{-2}}$
Maximum Motor Temperature	52°C
Maximum Operating Altitude (AMSL)	3210 m
Maximum Altitude Above Ground Level (AGL)	1800 m
Operating Temperature Range	-20°C to 40°C
Maximum Operating Humidity	100% (light rain)

trical components remained functional without failure—the structural design of the CS3D proved insufficient for prolonged exposure to the dynamically evolving and turbulent conditions characteristic of deep convective environments. Specifically, the use of PLA in the airframe shell and the absence of enhanced motor redundancy or aerodynamic shaping contributed to the platform's vulnerability to both mechanical stress and flight instability. Moreover, the propulsion system lacked the necessary thrust-to-weight ratio to maintain controlled ascent in wind regimes exceeding 20–25 $\mathrm{m\,s^{-1}}$. These cumulative factors ultimately underscored the need for a more resilient and power-efficient platform, motivating the development of the CopterSonde-SWX as a next-generation solution tailored for severe weather deployment [63].

To address these limitations, the CSWX platform was developed with several key enhancements. These included refined aerodynamic shaping to reduce drag, reinforced structural components for improved survivability, increased propulsion capacity to maintain altitude in strong winds, and modular sensor integration designed to minimize flow disturbances. Together, these upgrades aimed to improve operational robustness and ensure reliable performance in severe weather conditions.

2.1.2 CopterSonde-SWX UAS

The CSWX represents a significant aerodynamic and structural advancement over the CS3D. Developed specifically for high-resolution vertical profiling tailored to severe weather environments, the CSWX integrates a high-power propulsion system and refined aerodynamic elements to improve resilience and measurement accuracy under extreme wind conditions [65].

One of the most notable aerodynamic improvements is the incorporation of a fixed upward body tilt of $15°$, which serves to reduce drag during forward flight and enhances efficiency in strong wind environments. This configuration was informed by rotor-tilting studies, which demonstrated enhanced flight stability and speed with moderate tilting angles [66]. The airframe was further elongated and narrowed to minimize frontal area, allowing improved penetration through turbulent atmospheric layers.

The CSWX platform retained the quadcopter configuration of the CS3D, but incorporated key aerodynamic and propulsion enhancements. The airframe is constructed from carbon-fiber plates joined by aluminum standoffs, while the external shell was refined and 3D-printed in polycarbonate to reduce aerodynamic drag and increase structural stiffness [65]. Each arm integrates a 3110-size-10 900 KV brushless motor paired with a high-performance electronic speed controller, enabling a higher thrust-to-weight

Figure 2.2: The CopterSonde-SWX (CSWX), developed in 2024, represents the next evolution of weather-sensing UAS technology, extending the capabilities of the earlier CopterSonde-3D platform. Developed for deployment in severe weather environments, the CSWX incorporates an advanced aerodynamic design tailored to enhance flight stability and improve measurement accuracy under high-wind atmospheric conditions.

ratio [65].

In its high-performance configuration, the platform is powered by a 6000 mAh 6S LiPo battery rated for a continuous discharge of 168 A, which provides sufficient thrust to maintain flight stability under high-wind conditions. For extended-duration missions, a secondary 9000 mAh 6S Li-ion battery, rated for a continuous discharge of 90 A, is employed [65]. Comparative testing against DWL confirmed the improved wind estimation accuracy of the CSWX across both low- and high-wind regimes.

Figure 2.3 shows the side view of the CSWX platform following a night flight, highlighting its streamlined aerodynamic design optimized for high-wind performance.

Table 2.2 presents the full technical specifications of the CSWX platform, highlighting both measured and theoretical performance capabilities [65].

Figure 2.3: Side view of the CSWX on the launchpad after a night flight, illustrating the platform's streamlined airframe designed to enhance aerodynamic efficiency and operational stability in high-wind environments.

Table 2.3 summarizes the key design and performance differences between the CS3D and CSWX CopterSonde platforms, highlighting the enhancements made to enable reliable deployment in severe weather and high-wind environments.

2.1.3 CLAMPS Doppler Wind Lidar System

The Collaborative Lower Atmospheric Mobile Profiling System (CLAMPS) is a ground-based, mobile observational platform designed for high-temporal-resolution profiling of the PBL . Developed collaboratively by the University of Oklahoma and the National Severe Storms Laboratory (NSSL), CLAMPS provides critical thermodynamic and kinematic measurements essential for mesoscale meteorological studies and severe weather monitoring [62].

CLAMPS consists of a compact 4.9 m trailer that supports rapid deployment and autonomous operation. Equipped with onboard diesel generators, the system can

Table 2.2: Measured and theoretical specifications defining the flight envelope of the CopterSonde-SWX. Values in parentheses indicate theoretical maximums.

Specification	**Value**
Maximum All-up weight	2.8 kg
Maximum Hover time (10% batt.)	15.6 min
Maximum Forward speed	35.4 $\mathrm{m\,s^{-1}}$
Maximum Mean wind tolerance	25 (30) $\mathrm{m\,s^{-1}}$
Maximum Gust tolerance	28 (35) $\mathrm{m\,s^{-1}}$
Maximum Climb rate	5 (12.5) $\mathrm{m\,s^{-1}}$
Maximum Descent rate	7 (10) $\mathrm{m\,s^{-1}}$
Maximum Input voltage	26 V
Maximum Peak current	110 (264) A
Maximum Continuous current	65 (140) A
Maximum Mechanical vibration	30 $\mathrm{m\,s^{-2}}$
Maximum Motor temperature	65 (80)°C
Maximum Min. air density	0.95 (0.85) $\mathrm{kg\,m^{-3}}$
Maximum Altitude above ground	1500 (3000) m
Maximum Temperature range	-20 to 45°C
Maximum Relative humidity range	0 – 100%
Maximum Payload weight range	300 – 450 g

Table 2.3: Comparison of CS3D and CSWX CopterSonde Platforms.

Feature	**CS3D**	**CSWX**
Primary Application	Moderate weather profiling	Severe weather profiling
Tilt Configuration	No tilt	Fixed 15° upward tilt
Airframe Material	Aluminum/carbon fiber with PLA shell	Carbon-fiber frame with polycarbonate shell
Maximum Gust Tolerance	26 $\mathrm{m\,s^{-1}}$	35 $\mathrm{m\,s^{-1}}$
Operating Temperature Range	-20°C to 40°C	-20°C to 45°C
Sensor Integration	Modular "scoop" with WVFM	Refined scoop with WVFM
Propulsion and Power	T-Motor MN3110-780 with 5870 mAh LiPo battery	900 KV motors with 6000 mAh (LiPo) and 9000 mAh (Li-ion) dual batteries
All-Up Weight	2.36 kg	2.80 kg

function independently and requires no commercial licenses for transport or setup. Two nearly identical units, CLAMPS-1 and CLAMPS-2, facilitate either simultaneous multi-location observations or redundancy within a single experiment.

Instrumentation. CLAMPS integrates several remote sensing and in situ instruments, including an Atmospheric Emitted Radiance Interferometer (AERI), a Doppler Wind Lidar (DWL), a microwave radiometer (MWR), surface weather sensors, and a radiosonde launch system.

Of these, the Doppler Wind Lidar and radiosonde systems were primarily utilized in this study. The DWL operates by transmitting laser pulses into the atmosphere and analyzing the Doppler shift in the backscattered signal from aerosols and particulates. Through the application of Velocity–Azimuth Display (VAD) techniques, the DWL provides high-temporal-resolution vertical profiles of horizontal and vertical wind components, which are essential for resolving fine-scale boundary layer dynamics and validating UAS-based wind retrievals.

The radiosonde system complements the DWL by delivering in situ measurements of pressure, temperature, humidity, and wind throughout the troposphere. Radiosondes are carried aloft by weather balloons and yield high-vertical-resolution atmospheric profiles, which serve as calibration references and ground truth for both remote sensing instruments and UAS-based observations.

Operational Role. CLAMPS has been deployed in several major meteorological field campaigns, including VORTEX-SE and PECAN, to provide high-frequency, near-storm environmental observations. Its capability for continuous monitoring enables detailed analysis of boundary layer processes that are often under-resolved by traditional radiosonde launches or radar systems. Due to its well-validated vertical wind profiling capability—particularly through DWL—CLAMPS was selected as the ground truth for this study. Its ability to deliver high-temporal-resolution wind profiles in open-field environments made it an ideal benchmark for evaluating the accuracy and robustness of the wind estimation algorithms developed for the CopterSonde platforms.

Table 2.4: CLAMPS instrumentation summary [62].

Component	**CLAMPS-1**	**CLAMPS-2**
Thermodynamic Profiling	AERI v4, HATPRO	AERI v4, MP-3000A
Wind Profiling	Halo Stream Line	Halo Stream Line XR
Aerosol Profiling	–	–
Surface Weather	Vaisala WXT530 (3 m)	Vaisala WXT530 (5 m)
Radiosonde System	iMet	Vaisala RS-41 compatible
Power Source	Diesel Generator / Grid	Diesel Generator / Grid
MWR Support	Yes	Yes

In this study, CLAMPS-provided Doppler Wind Lidar data served as the primary ground truth during the initial phase of wind estimation model development.

In the final phase of this study, WINDoe [67]—an advanced post-processing wind estimation system developed at the Cooperative Institute for Severe and High-Impact Weather Research and Operations (CIWRO)—was used as the ground truth for both horizontal and vertical wind estimation. WINDoe incorporates DWL data collected by the CLAMPS system along with additional meteorological observations from radiosondes, Mesonet towers, and UAS platforms to generate reliable wind profiles through multi-source data fusion.

The primary phase of this thesis focused on estimation of horizontal wind. However, restricting wind estimation research solely to horizontal wind is insufficient for comprehensive atmospheric studies or practical UAS-based applications. Accurate vertical wind estimation is equally important for turbulence characterization, boundary layer analysis, and numerical weather prediction. Traditional approaches to vertical wind estimation typically rely on dedicated DWL, but the logistical and resource limitations associated with deploying multiple DWLs in the field posed a significant challenge in this research.

To address this, WINDoe was employed as a surrogate reference system. Using optimal estimation theory, WINDoe integrates data from diverse platforms and sensors to retrieve consistent and physically constrained estimates of the full three-dimensional (3D) wind field. Unlike single-instrument systems, which are often susceptible to measurement noise or spatial gaps, WINDoe offers smoother and more stable wind profiles suitable for training data-driven models.

2.2 Chapter Summary

This chapter presented a detailed overview of the instruments employed in this study for atmospheric wind profiling and validation. The discussion began with the evolution of the CopterSonde platform, highlighting the development and capabilities of two CopterSonde UASs: the CS3D and the CSWX. The CS3D, although effective under moderate conditions, demonstrated limitations in high-wind environments, prompting the design of the CSWX. The CSWX incorporated significant improvements in aerodynamic shaping, propulsion capacity, and structural resilience, making it suitable for deployment in severe weather scenarios.

The chapter then introduced the CLAMPS platform, a ground-based mobile profiling system developed by the the National Severe Storms Laboratory. CLAMPS integrates a suite of passive and active remote sensing instruments, among which the DWLs and radiosonde systems were of central relevance to this study. These systems provided high-temporal-resolution wind profiles and in situ atmospheric soundings, respectively, and served as the primary ground truth for evaluating CopterSonde-based wind estimation.

CLAMPS's validated performance in prior field campaigns and its ability to operate autonomously in open-field conditions established it as the ideal reference system for benchmarking UAS-derived wind estimates. In the latter phase of this research, the WINDoe system—developed by CIWRO—was used to further enhance ground truth fidelity through multi-sensor data fusion. Overall, this chapter established the foundational instrumentation and reference systems used to assess and validate the performance of the proposed wind retrieval methodologies.

Chapter 3

Comparative Analysis of Established Baseline Wind Estimation Algorithms

In the last decade, several UAS-based wind estimation algorithms have been proposed in the literature, wherein wind vector is derived from onboard sensor measurements in a direct way [68] and kinematic/dynamic modeling in an indirect way [69]. Among these, the Linear, Quadratic [70], and Linear Extended State Observer (LESO)-based [26, 65] approaches—comprising both kinematic and dynamic models—have been developed and tailored to the CopterSonde platform to infer wind vectors from external aerodynamic forces acting on the airframe.

The objective of this chapter is to fine-tune or calibrate these current wind estimation methods of the CopterSonde UAS against a known ground-truth observation. Consequently, the results of the calibration was used as a metric to compare the performance between the methods.

Since this study did not have access to controlled wind tunnel facilities, all calibration and validation activities were conducted under open-field conditions. To ensure the reliability of the results, field experiments were strategically performed on windy days and co-located with high-quality remote sensing instrumentation—including DWLs, Oklahoma Mesonet towers, and radiosonde launches. These instruments are proven technology in the atmospheric science community and, thus, were considered as the observational ground truth for this study necessary for evaluating the performance of the UAS-based wind estimation algorithms under realistic atmospheric conditions.

In addition to these co-located measurements, the WINDoe (Wind Information from Optimally-merged Datasets) system—maintained by CIWRO at the University of Oklahoma—was also available for use in this study [67]. WINDoe [67] is a post processing technique that was developed to fuse together data from DWLs, radiosondes, Oklahoma 3D Mesonet towers, and UAS-based platforms, providing high-resolution wind retrievals with lower uncertainty that could served as a reference dataset for this chapter's UAS-based wind calibration and validation purposes and also to train the machine learning models.

Unlike DWL measurements alone, which may be affected by noise or spurious spikes, WINDoe provides smoother and more stable wind estimates. By synthesizing these diverse datasets, WINDoe produces continuous wind profiles that are deeper, more complete, and include uncertainty estimates at each vertical level. This blended retrieval significantly reduces data gaps and enhances vertical resolution, particularly in challenging conditions where individual instruments alone may fail to provide complete measurements. As such, WINDoe could be adopted as the primary reference for calibrating and validating the LESO, Linear, and Quadratic wind estimation algorithms on both the CS3D and CSWX platforms. Its ability to integrate diverse observational datasets made it especially suitable for serving as ground truth in the development and evaluation of machine learning models for three-dimensional wind estimation.

The co-located DWLs, Mesonet towers, and radiosondes provided high-quality wind vector measurements, which were treated as the most accurate representations of the atmospheric wind field information available and thus served as ground truth for calibrating and validating the CopterSonde's onboard wind estimation algorithms.

These reference observations were used as targets against which the performance of the estimation methods could be quantitatively assessed.

Moreover, procedures for calibration and validation of UAS-based wind estimation algorithms were established and described in the following sections. In short, a series of coordinated field experiments were conducted in which the CopterSonde platforms performed vertical profiling of the PBL under windy conditions. These flights were co-located with ground-based and remote sensing instruments, including a DWL, an Oklahoma Mesonet tower [51], and radiosonde launches provided by CIWRO. The vertical profiles typically extended from the surface to several hundred meters above the ground level (AGL) to capture a representative range of wind conditions. CopterSonde observations were time-synchronized and geospatially aligned with the co-located atmospheric instruments to ensure data consistency across platforms.

In the case of the LESO-based approach, wind speed estimation is derived indirectly through aerodynamic modeling. The LESO reconstructs external forces acting on the CopterSonde—such as aerodynamic drag and lift—based on motion data recorded by the autopilot. These reconstructed force vectors are then converted into wind velocities using simplified aerodynamic formulations [26, 65]. In contrast, the Linear and Quadratic models bypass force reconstruction altogether and empirically relate the platform's tilt angle to horizontal wind speed through regression-based relationships [70].

Once the experimental datasets were collected and synchronized with the ground-truth wind profiles, a Differential Evolution Optimizer (DEO) scheme was implemented to fine-tune the coefficients in each estimation algorithm. DEO was selected for its robustness in solving noisy, nonlinear, and multi-dimensional optimization problems where gradient information is either unavailable or unreliable [71, 72]. The objective of the optimization was to minimize the root mean square error (RMSE) between the algorithm-estimated wind vectors and the reference wind measurements.

The primary advantages of DEO include its simplicity, minimal parameter tuning, and effective global search capabilities. However, it is not without limitations; DE can be sensitive to population size and may exhibit reduced efficiency in high-dimensional or constrained optimization landscapes [71]. By introducing DEO as a general-purpose optimization method, this study evaluates whether both empirical and model-based wind estimation techniques can be enhanced to match or exceed the accuracy of physically derived models under realistic operational conditions.

The structure of this chapter is as follows: first, the LESO, Quadratic, and Linear algorithms are reviewed to provide background on traditional wind estimation approaches. This is followed by an introduction to the proposed DEo based optimization method, which is used to find the optimal method parameters that best approach the output to the reference. Finally, a quantitative comparison is presented across all methods, evaluating both computational efficiency and wind estimation performance.

3.1 Description of the UAS-based Wind Methods

This section presents an overview of three wind estimation algorithms that have been used with the CopterSonde platform. Each method utilizes the tilt angle of the aircraft—typically the pitch in wind-vane mode—to estimate horizontal wind speed. The algorithms considered here include the historically used Linear method, a more recent Quadratic approach, and the Linear Extended State Observer (LESO)-based method discussed separately.

This study examined three baseline algorithms—Linear, Quadratic, and LESO—chosen for their prominence in current UAS-based wind estimation studies and their particular relevance to the CopterSonde platform. The Linear and Quadratic methods are kinematic methods that estimate horizontal wind speed based on the tilt of the UAS, and they have gained widespread acceptance because to their simplicity and limited dependence on dynamic system modeling [69, 70]. These methods are computationally efficient and ideally suited for small UASs functioning in real-time. The LESO algorithm is a dynamic force-based observer that predicts horizontal and vertical wind components by modeling the external aerodynamic forces acting on the vehicle [26]. LESO has exhibited enhanced performance in turbulent conditions by integrating motor RPM, air density, and attitude data. These three methods collectively embody a range of complexity and capacity in wind estimation, therefore serving as a solid comparison basis for assessing performance enhancements facilitated by optimization approaches [26].

3.1.1 Linear Extended State Observer

The CopterSonde is equipped with a wind vane flight mode (WVFM), enabling it to estimate wind direction in real-time and continuously reposition itself pointing into the wind. The elongated shape of the CopterSonde body enhances aerodynamic stability, particularly under active WVFM operation. The use of WFM simplifies the formulation of wind estimation by maintaining the CopterSonde's orientation into the prevailing wind. This orientation ensures that the platform presents a pseudo-constant cross-sectional area to the incoming airflow, thereby stabilizing the aerodynamic configuration and enhancing the predictability of the resulting drag forces. This operational assumption improves the consistency and reliability of force-to-wind relationships utilized in model-based estimators such as LESO [26].

The LESO method, adapted to the CopterSonde configuration, estimates wind by leveraging from the aircraft's response to external forces while maintaining position. It uses the computed position and velocity from the autopilot in the North-East-Down

(NED) frame, using fused GNSS and IMU (Inertial Measurement Unit) data. The LESO-based method relies on multiple onboard measurements to estimate the external aerodynamic forces acting on the CopterSonde. Key parameter inputs to the LESO include rotor's angular velocity (RPMs), air density, platform position and velocity in the NED frame, and attitude information (i.e., roll, pitch, and yaw angles). These inputs are used to solve the system's dynamic equations of motion, allowing the built-in observer system to isolate and estimate the aerodynamic drag and lift forces. The resulting force estimates are then mapped to wind vectors using a Rayleigh's aerodynamic friction equation [73], under the assumption of steady-state equilibrium between thrust and aerodynamic drag.

In the hovering or slow ascent/descent regime, when no horizontal motion is commanded, any tilt compensation produced by the aircraft is primarily induced by wind drag. This drag force D is balanced by a thrust component as a result of the tilt. Therefore, if the thrust component is known, then the drag force can be calculated using Newtonian physics's free-body diagram vector math. The thrust vector can then be translated to wind vector using Rayleigh's aerodynamic friction equation. Under these conditions, the aircraft's dynamic response serves as a proxy for estimating three-dimensional wind vector, including the vertical component [63].

The underlying dynamic model in the LESO assumes that the CopterSonde's position $\vec{P}$ and velocity $\vec{V}$ are directly measurable, and that the total external disturbance force $\vec{D}$—consisting mainly of wind-induced forces—can be represented as an additional state variable. After applying Observer theory to the system of equations that defined the CopterSonde's dynamic model, the matrix formulation is given by [74–76]:

$$\dot{x} = (A - LC)x + B \begin{bmatrix} u \\ h \end{bmatrix} + Ly$$
$$y = Cx \tag{3.1}$$

with system matrices:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{1}{m} \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

where $x = [x_1\ x_2\ x_3]^T \equiv [\vec{P}\ \vec{V}\ \vec{D}]^T$ is the state vector, u is the control input derived from thrust and gravity in the inertial frame, and $h(t)$ is the unknown rate of change of the disturbance which is assumed to be zero. The matrix L is the observer's gain that weighs how reliable the dynamic model's prediction is compared to the direct measurements. By adjusting this parameter, the system may drive the error to zero and converge to an estimate [63]:

$$L = \begin{bmatrix} \beta_{p1} & \beta_{v1} \\ \beta_{p2} & \beta_{v2} \\ \beta_{p3} & \beta_{v3} \end{bmatrix} \tag{3.2}$$

The elements of L are first-guessed and defined based on observer's bandwidths for position and velocity [63, 77]. However, this parameter is then fine-tuned in this study with the use of cost-function-based optimizers which is described in a later section.

To convert the estimated aerodynamic disturbance $\vec{D}$ into wind velocity $\vec{U}$, a modified Rayleigh drag model was applied:

$$D = \rho C_d U^{1/b} \tag{3.3}$$

where C_d is generalized drag coefficient , ρ is the humid air density, and b is a fitted exponent. This formulation generalizes the classic quadratic Rayleigh model to account for surface roughness and geometric irregularities of the CopterSonde [26, 63, 73].

Drag coefficients C_d and exponents b were obtained by fitting LESO-estimated drag against reference measurements from DWL, with the optimal coefficients identified using the DEO algorithm. The results indicated near-linear behavior in the vertical component and a more quadratic response in the horizontal plane, consistent with the classical Rayleigh drag formulation.

Figure 3.1 illustrates the core working principle of the LESO algorithm. In WVFM, the CopterSonde aligns itself with the wind. Using motion data from the autopilot, external forces acting on the airframe are estimated through a mathematical model of the vehicle dynamics. A LESO processes these estimates to isolate external force contributions. Since these forces are primarily due to wind, the LESO output is then translated into wind estimates using an aerodynamic drag equation.

3.1.2 Linear-Regression Method

The linear-regression approach was originally proposed by Neumann and Bartholmai [68], where a wind tunnel was used to determine a relationship between pitch and roll angles—measured by the onboard IMU of a given platform—and the corresponding known wind speed in the tunnel. An empirical relationship between wind speed and tilt angle was later refined to facilitate field-based calibration without the need for wind tunnel access [69].

This method is derived from the force balance acting on the aircraft during steady-state or constant-velocity ascent. Under the assumption of negligible acceleration, the net force on the system can be approximated by:

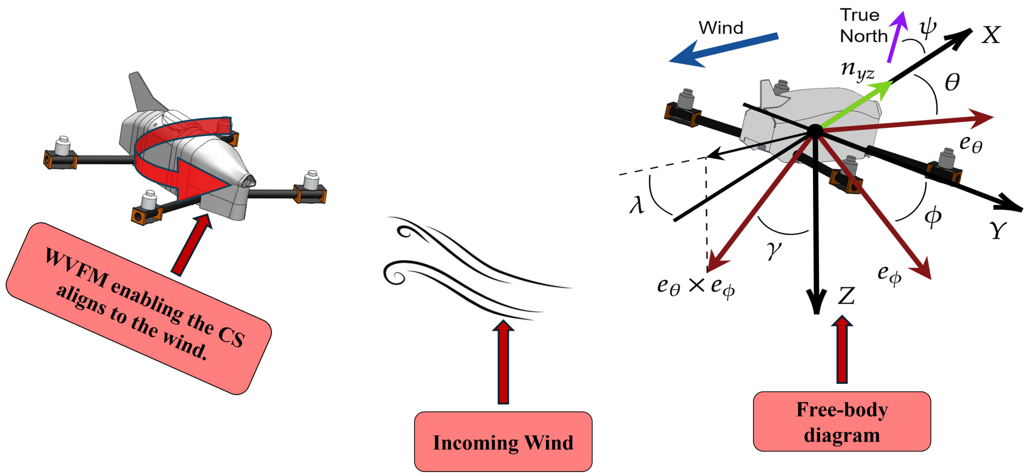

Figure 3.1: Visual representation of the WVFM in action and free-body diagram used for the LESO calculations. The CopterSonde aligns into the wind using WVFM, and motion data from the autopilot are fed into a UAS dynamic model. A state observer, defined in Eqns. (3.1), estimates the net external force, which is then translated into wind speed using an aerodynamic drag relationship as shown in Eq. (3.3). Image taken from [26]

$$0 = \mathbf{G} + \mathbf{D} + \mathbf{T} \tag{3.4}$$

where $\mathbf{G}$ is the gravitational force, $\mathbf{D}$ is the aerodynamic drag, and $\mathbf{T}$ is the thrust [70]. Expanding each of these terms:

$$\mathbf{G} = mg\,\mathbf{e}_D \tag{3.5}$$

$$\mathbf{D} = -D\,\mathbf{e}_v \tag{3.6}$$

$$\mathbf{T} = T_v\,\mathbf{e}_v + T_D\,\mathbf{e}_D = |T|\sin(\psi)\,\mathbf{e}_v + |T|\cos(\psi)\,\mathbf{e}_D \tag{3.7}$$

Here, ψ is the tilt angle, $\mathbf{e}_D$ is the unit vector in the downward (gravitational) direction, and $\mathbf{e}_v$ is the unit vector in the direction of the relative wind velocity. Solving for

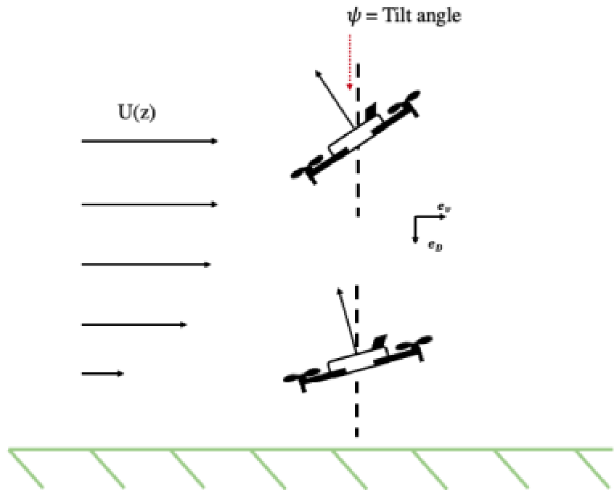

Figure 3.2: Illustration of the coordinate system used to derive wind speed from the tilt angle of the CopterSonde. When wind vane mode is active, the tilt angle ψ corresponds to the pitch of the aircraft. The unit vectors $\mathbf{e}_v$ and $\mathbf{e}_D$ represent the directions of the relative wind (assumed horizontal) and gravitational force, respectively. Image taken from [70]

the drag force D:

$$D = mg\tan(\psi) \tag{3.8}$$

Subsequently, applying the quadratic Rayleigh drag equation from Eq. 3.3 using $b = 2$. The wind speed U can then be expressed as:

$$U = \sqrt{\frac{2mg\tan(\psi)}{C_D\rho A}} \tag{3.9}$$

For practical implementation, the constant parameters are grouped into a calibration coefficient C_1 [69, 78] while also adding a free term C_0 to correct any offset, yielding the linear expression that linearly approximates the physical behavior of the

UAS caused by wind effects:

$$U = C_1\sqrt{\tan(\psi)} + C_0 \tag{3.10}$$

The linear method has been shown to achieve wind speed accuracy within approximately $\sim$0.5 m s^{-1} in calm wind conditions (0–5 m s^{-1}), making it relatively robust, although accuracy tends to degrade at higher wind speeds [69, 70].

3.1.3 Quadratic-Regression Method

The quadratic method was introduced to enhance wind estimation accuracy by fitting a second-degree polynomial to observed relationships between tilt angle and wind speed. In contrast to the linear regression method, this approach is entirely empirical and does not rely on any underlying physical force balance assumptions [70]. However, these observations were obtained under very low wind speed conditions (less than 5 m s^{-1}), and it remains unclear whether the method is broadly applicable to higher wind speeds [79].

The method is mathematically simple, as it does not incorporate physical modeling of the aircraft's dynamics [70]. Instead, the Quadratic methods is an expansion of the Linear method. In the Linear method, the $\sqrt{tan(\psi)}$ is seen as the independent parameter. Therefore, the Quadratic method simply adds a quadratic term to Eq. (3.10) as a function of this independent parameter.

The wind speed is modeled as:

$$U = B_1\sqrt{\tan^2(\psi)} + B_0\sqrt{\tan(\psi)} \tag{3.11}$$

where B_0 and B_1 are empirically derived coefficients obtained through regression against reference wind measurements, similar to the Linear method. Although the Quadratic method has shown improved accuracy over the linear model, particularly under moderate wind conditions, the absence of a physical basis limits its adaptability and robustness across different platforms [70].

3.2 Wind Calibration and Performance Evaluation using Differential Evolution Optimizer

DEO, a population-based stochastic optimization algorithm, was employed to minimize the root mean square error (RMSE) between CopterSonde-estimated winds and reference observations by iteratively adjusting the method's unknown parameters, generally dependent on aerodynamic properties, to meet a predefined error threshold. The DEO is used to dial the coefficients for each of the CopterSonde-based wind algorithms for both CopterSonde-3D and CopterSonde-SWX platform editions. A quantitative comparison was performed across different wind speed regimes presented on a windy day to assess the performance of each combination of algorithm and platform. This evaluation aims to identify the most accurate and robust method suitable for operational deployment.

To perform calibration using the DEO framework, both CopterSonde platforms—CS3D and CSWX—were deployed simultaneously to ensure that each experienced similar atmospheric conditions. These profiling flights were conducted during windy periods and co-located with DWLs, the Oklahoma Mesonet tower in Washington, and a few radiosonde launches, enabling synchronized data acquisition between onboard sensors and ground-based reference systems. The collected datasets were subsequently aligned in time and space to extract valid vertical profiles suitable for calibration. The DEO algorithm was applied in post-processing to optimize the empirical coefficients specific to each wind estimation algorithm—LESO, Linear, and Quadratic. For each extracted profile, DEO iteratively minimized the RMSE between the wind vectors estimated by the algorithm and those measured by the DWL. This optimization process adjusted the model-dependent coefficients until a predefined convergence criterion was satisfied. The outcome of this calibration procedure was a set of optimized coefficients tailored for each algorithm–platform combination. These coefficients were tuned to reduce bias and minimize estimation error under the observed atmospheric conditions. The resulting calibrated models are expected to enhance wind retrieval accuracy in future deployments by aligning algorithmic predictions more closely with reference measurements, both in similar and in diverse environmental scenarios.

3.2.1 Differential Evolution Optimizer

DEO is a stochastic, population-based cost function optimization method that is well-known for its ability to solve difficult, non-differentiable, and multimodal problems. It does not require gradient information and can effectively explore high-dimensional parameter spaces, making it especially helpful for optimizing algorithmic coefficients in dynamic systems such as wind estimates for UASs. DEO was used to optimize the adjustable parameters of three baseline wind estimation techniques. By minimizing the RMSE between algorithm outputs and ground truth reference data, DEO allows for empirical model calibration for enhanced accuracy throughout flight regimes.

DEO is a widely used evolutionary algorithm inspired by Darwin's theory of natural selection. It has been extensively studied and applied to a variety of optimization problems and engineering applications [71]. The algorithm was first introduced by Storn and Price in 1997 [80]. Given the ubiquitous nature of optimization, most real-world engineering systems can be formulated as complex models characterized by nonlinearities, multimodality, discontinuities, and non-differentiability [71, 80]. To effectively address these challenges, an optimization process is typically employed to identify the optimal combination of decision variables that maximizes or minimizes a given cost function, while satisfying both technical and non-technical constraints within a reasonable timeframe [71, 80].

Existing optimization techniques such as linear programming, dynamic programming, and Newton's methods have proven effective only for certain problem types and often require prior knowledge of the problem structure [81]. These methods are limited by poor global search capabilities, sensitivity to initial conditions, and their reliance on gradient information.

Literature shows that DEO has been used in several complex applications and reviewed in the context of artificial intelligence [82], swarm and evolutionary computation [83], dynamic economic dispatch problems [84], and intelligent, software-intensive systems [71]. A comprehensive state-of-the-art survey is also available in [72]. In practice, DEO has been successfully applied to structural optimization, parameter tuning in control systems, image processing, and deep learning, demonstrating its versatility and robustness across domains [85]. Despite its widespread success, DEO is not without limitations. It can suffer from premature convergence into local maxima or minima, including sensitivity to parameter settings, particularly in complex, noisy, or constrained environments [71, 85].

As CopterSonde-based wind estimation algorithms rely on empirical coefficients that are sensitive to both platform-specific aerodynamics and environmental variability, selecting an effective optimization strategy is essential for achieving accurate calibration. DEO was selected for this task due to its proven capability in handling nonlinear, non-convex objective functions without requiring gradient information or explicit prior knowledge of the parameter space. These characteristics make DEO particularly well-suited for tuning UAS-based wind estimation models, where the underlying aerodynamic relationships are often complex or poorly defined, and the observational data are subject to measurement uncertainty [71].

The DEO algorithm proceeds through four main stages: initialization, mutation, crossover, and selection [80]. In DEO, each candidate solution represents a set of algorithm-specific coefficients being optimized, and the population refers to a group of these solutions—each encoded as a D-dimensional vector—evaluated in parallel to minimize wind estimation error for the CopterSonde.

Initialization: The initial population is generated uniformly at random within predefined bounds:

$$x_{i,j}^{(0)} = x_j^{\text{min}} + \text{rand}_{i,j} \cdot (x_j^{\text{max}} - x_j^{\text{min}}) \tag{3.12}$$

where $x_{i,j}^{(0)}$ is the j-th component of the i-th vector, and $\text{rand}_{i,j}$ is a uniformly distributed random number in $[0, 1]$.

Mutation: For each target vector x_i, a donor vector v_i is created by randomly selecting three distinct vectors from the population and adding the scaled difference of two of them to the third:

$$v_i = x_{r1} + F \cdot (x_{r2} - x_{r3}) \tag{3.13}$$

where $r1$, $r2$, and $r3$ are distinct indices different from i, and $F \in (0, 2]$ is the scaling factor.

Crossover: The trial vector u_i is formed by combining components from the target and donor vectors, commonly using binomial crossover:

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } \text{rand}_j \leq CR \text{ or } j = j_{\text{rand}} \\ x_{i,j} & \text{otherwise} \end{cases} \tag{3.14}$$

Here, $CR \in [0, 1]$ is the crossover rate.

Selection: The trial vector competes with the target vector, and the one with better fitness proceeds to the next generation:

$$x_i^{(g+1)} = \begin{cases} u_i & \text{if } f(u_i) \leq f(x_i) \\ x_i & \text{otherwise} \end{cases} \tag{3.15}$$

This evolutionary process continues until a stopping criterion is met, such as a maximum number of generations or a satisfactory error threshold, such as a maximum number of generations or achieving a user-defined RMSE threshold criteria. If this threshold is unattainable, the optimization aims to minimize the RMSE below another secondary user-defined threshold. Given that this study is related to wind-sensing atmospheric instruments, the DEO's threshold targets were set in compliance with the World Meteorological Organization's (WMO) recommended accuracy standards for horizontal wind measurement. WMO defines the wind speed accuracy standard as Goal ($<$1ms^{-1}), Breakthrough ($<$2ms^{-1}), and Threshold ($<$8ms^{-1}) as listed in [86].

3.2.2 Implementation of Differential Evolution Optimizer

In this study, DEO was implemented as an initial step toward integrating machine learning methodologies into the wind estimation framework of the CopterSonde UAS. As described in previous sections, the LESO, Linear, and Quadratic algorithms each rely on empirical coefficients that link kinematic and dynamic responses of the platform to estimate wind.

The objective of the DEO algorithm was to optimize these coefficients by minimizing the discrepancy between estimated and observed wind measurements. Specifically, DEO was employed to fit each presented model to the DWL's wind observations collected in open-field environments. The DWL data, captured during programmed CIWRO and NSSL field deployments, served as the reference or ground truth for both horizontal and vertical wind in the first part of this work, allowing for the fine-tuning of the established UAS-based wind estimation models.

To ensure robustness during optimization and mitigate the influence of measurement noise, the DWL data were filtered using a quality mask labeled `good_lidar`. This excluded anomalous spikes typically present in the DWL observations when the concentration of aerosols in the atmosphere is low and, hence, backscattering is poor. The DEO algorithm was configured to minimize a cost function defined by the RMSE between model-estimated and DWL-measured wind vectors. The optimization proceeded iteratively until convergence was achieved as per the WMO standard, yielding the coefficient set that minimizes the estimation error across the dataset.

To keep this document concise, only the implementation of DEO with the LESO wind estimation is described below. The Linear and Quadratic methods follow the same procedures but are presented only through their results. Therefore, to evaluate the performance of the LESO-based wind estimation and guide the DEO optimization process, the RMSE was computed between LESO-estimated and DWL-measured horizontal and vertical wind.

The RMSE calculations for the wind estimation algorithms were systematically conducted by comparing wind estimates from the CopterSonde against reference measurements obtained from the DWL. RMSE was computed separately for the horizontal and vertical wind components and then aggregated into a global RMSE metric to assess the overall performance of each algorithm.

For each vertical profile, the estimated horizontal wind speed from the CopterSonde was compared against the corresponding DWL reference measurements. The horizontal wind RMSE, denoted as $\mathrm{RMSE}_{\mathrm{U,wspd}}$, was computed iteratively using a weighted formula that accounts for both the cumulative and current profile errors. The RMSE update rule for horizontal wind U ($\text{RMSE}_{\text{U,wspd}}$) was defined as:

$$\text{RMSE}_{\text{U,wspd}} = \sqrt{\frac{N_U \, * \text{RMSE}^2_{\text{U,wspd}} \; + \; \sum_{i=1}^{n} \left(U_{\text{leso},i} - U_{\text{lidar},i}\right)^2}{N_U + n}}, \tag{3.16}$$

where,

$$\begin{aligned}
\text{RMSE}_{\text{U,wspd}} &= \text{RMSE of } U_{\text{leso}} \text{ relative to } U_{\text{lidar}}, \\
N_U &= \text{Cumulative data for horizontal wind from previous profiles,} \\
U_{\text{leso},i} &= \text{horizontal-wind estimate from LESO (sample } i\text{),} \\
U_{\text{lidar},i} &= \text{horizontal-wind measurement from LiDAR (sample } i\text{),} \\
n &= \text{valid LESO horizontal-wind samples in the current profile.}
\end{aligned}$$

A similar approach was applied to update the RMSE for vertical wind W ($\text{RMSE}_{\text{W,wspd}}$):

$$\text{RMSE}_{\text{W,wspd}} = \sqrt{\frac{N_W \, * \, \text{RMSE}^2_{\text{W,wspd}} \; + \; \sum_{i=1}^{m} \left(W_{\text{leso},i} - W_{\text{lidar},i}\right)^2}{N_W + m}}, \tag{3.17}$$

where,

$$
\begin{aligned}
\mathrm{RMSE_{W,wspd}} &= \text{RMSE of } W_{\mathrm{leso}} \text{ relative to } W_{\mathrm{lidar}},\\
N_W &= \text{Cumulative data for vertical wind from previous profiles},\\
W_{\mathrm{leso},i} &= \text{vertical-wind estimate from LESO (sample } i\text{)},\\
W_{\mathrm{lidar},i} &= \text{vertical-wind measurement from LiDAR (sample } i\text{)},\\
m &= \text{valid LESO vertical-wind samples in the current profile.}
\end{aligned}
$$

The global RMSE, which served as the overall cost function for DEO, was computed as a weighted aggregate of the horizontal and vertical components, which is updated at every iteration of the DEO and compared to the defined threshold. Using the global RMSE as the cost function in the DEO framework provides a comprehensive metric for evaluating and minimizing discrepancies between CopterSonde-estimated wind vectors and ground-truth measurements. By aggregating errors from both horizontal and vertical wind components, the global RMSE ensures a balanced optimization process that does not disproportionately prioritize one component over the other. This balanced formulation is essential for achieving accurate 3D wind estimations, as neglecting either component can result in significant inaccuracies, particularly in applications requiring high-fidelity wind estimation. The update rule for the global RMSE:

$$
\mathrm{RMSE_{G,wspd}} = \sqrt{\frac{(N_U + N_W)\,\mathrm{RMSE^2_{G,wspd}} + (N_U + n)\,\mathrm{RMSE^2_{U,wspd}} + (N_W + m)\,\mathrm{RMSE^2_{W,hspd}}}{2N_U + 2N_W + n + m}} \tag{3.18}
$$

where,

$$\mathrm{RMSE}_{\mathrm{G,wspd}} = \text{Combined RMSE of both horizontal and vertical wind,}$$

Similar to the LESO-based approach, the Linear and Quadratic wind estimation algorithms rely on empirical coefficients—denoted as A and B—that relate the tilt angle of the UAS to estimate wind. In both methods, DEO was employed to optimize these coefficients by minimizing the discrepancy between model-estimated wind speeds and reference values obtained from DWL measurements. The cost function used for these two methods is similar to Eq. (3.16) where U_{leso} is replaced with the model's respective horizontal wind estimates. There's not vertical wind estimation for the Linear and Quadratic methods and, thus, ignored for the calculations of the global RMSE.

This work also took the opportunity to compare two CopterSonde designs using the methods presented in this chapter under similar atmospheric conditions. Both CopterSonde platforms (CS3D and CSWX) were deployed simultaneously during multiple flight operations. These coordinated flights were conducted to observe similar atmospheric states, thereby allowing for a fair comparison of performance and parameter tuning. Table 3.1 summarizes the number of Intensive Observation Periods (IOPs) carried out during this study, along with their corresponding dates.

Table 3.1: Number of IOPs conducted for DEO-based optimization under comparable atmospheric conditions.

Date	**Number of IOPs**
22 May 2024	6
07 August 2024	4
08 September 2024	20
22 October 2024	5
20 November 2024	8
Total IOPs	43

The objective of the DEO algorithm was to determine the optimal values unknown coefficients for all presnted wind-estimation methods that minimize $\mathrm{RMSE}_{\mathrm{G,\,wspd}}$, thereby yielding the best agreement between model estimation and observed wind speeds.

The optimization was performed independently for each model, with DEO iterating until convergence, defined by the minimum achievable RMSE for the selected DWL dataset. This process was applied to all three algorithms on both CopterSonde platforms (CS3D and CSWX) to enable comparative performance analysis. The resulting optimized coefficients were then used to compute RMSE values, providing an objective assessment of each algorithm's wind estimation accuracy under open field, windy conditions.

In regards of the LESO-based method, Table 3.2 summarizes the optimized coefficients for both horizontal and vertical wind estimation, as well as the observer gains applied along the longitudinal and vertical axes, as defined in Eq. 3.3 and Eq. 3.2. The aerodynamic coefficients from Equation 3.3, denoted as C and b, are represented here as C_d, b_d for vertical wind estimation, and C_h, b_h for horizontal wind estimation. The LESO observer gains corresponding to the longitudinal and vertical dynamics are denoted as l_{xy} and l_{xz}, respectively. The table also includes the resultant RMSE values associated with each CopterSonde platform, reflecting the performance of the LESO model under the optimized parameter configuration.

Table 3.2: Optimized drag coefficients for horizontal wind (C_h, b_h) and vertical wind (C_d, b_d), observer gains (l_{xy}, l_{xz}), and RMSE values for each CopterSonde platform.

Platform	C_d	b_d	C_h	b_h	l_{xy}	l_{xz}	**RMSE**
CS3D	6.72	0.40	6.62	0.54	1.8	4.2	1.1077
CSWX	6.50	0.4228	4.51	0.94	2.0	1.5	0.8841

The parameters in Table 3.2 can be interpreted by recalling that drag is generally proportional to the square of velocity, as seen when setting the b coefficient in Eq. (3.3) to 0.5. Most of the b values in the table approximate the 0.5 value, indicating minor non-linearities that do not warrant adopting a different dynamic model. An exception is observed in the CSWX's horizontal b_h coefficient, which approaches a linear relationship ($b_h \rightarrow 1$) in addition to the lower C_h or C_d compared to the CS3D. This is attributed to the CSWX's narrower fuselage, which reduces air resistance from the body. Additionally, induced drag from fast-spinning propellers tends to behave linearly, as shown in previous wind tunnel studies [87]. Thus, the CSWX design may effectively minimize fuselage drag, with the rotor disk area dominating the aerodynamic forces.

Although DEO-based optimization was carried out independently for both the CS3D and CSWX platforms, the overall convergence behavior and distribution of evaluated parameters were qualitatively similar across the two cases. To avoid redundancy from here, the results from the CSWX are presented as representative examples. Figures 3.3 to 3.5 illustrate the search space exploration and convergence characteristics of the DEO process, emphasizing the regions where drag coefficients and observer gains converge to values that minimize the RMSE.

Figure 3.3 depicts the distribution of evaluated drag coefficients for the horizontal wind (C_h, b_h), while Fig. 3.4 corresponds to the vertical wind (C_d, b_d). Lastly, Fig. 3.5 shows the distribution of LESO's observer gains in the longitudinal and vertical axes (l_{xy}, l_{xz}). In all cases, darker regions indicate parameter combinations associated with lower estimation error, reflecting convergence behavior and the sensitivity of the optimization to initial conditions.

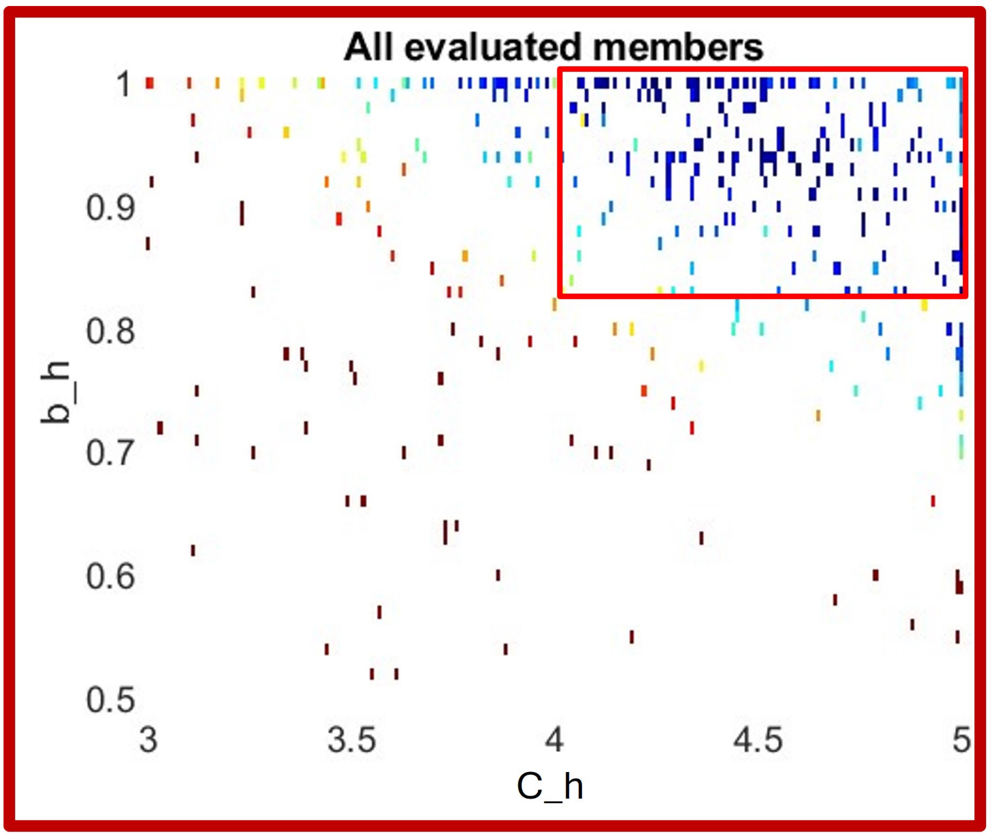

Figure 3.3: Evaluated distribution of drag coefficients for horizontal wind (C_h, b_h) obtained through DEO for the CSWX platform. Convergence is observed near $(C_h, b_h) \approx (4.5, 0.9)$, with the minimum RMSE achieved at $C_h = 4.51$ and $b_h = 0.9448$. These optimized coefficients were subsequently incorporated into the LESO framework for horizontal and vertical wind estimation. The coeffient, $b_h = 0.9448$ suggest a shift to a linear drag regime for the CSWX due to its improved design.

Switching to the Linear and Quadratic algorithms case. These methods can only

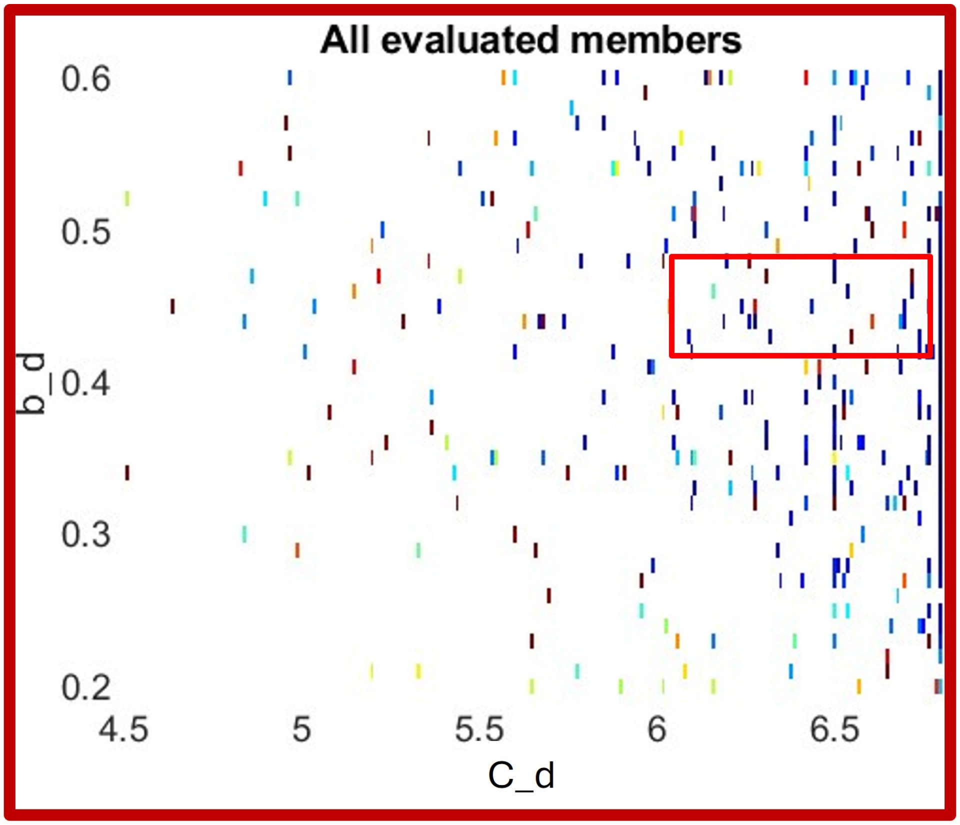

Figure 3.4: Evaluated distribution of drag coefficients for vertical wind (C_d, b_d) obtained through DEO for the CSWX platform. Convergence is observed near $(C_d, b_d) \approx (6.50, 0.42)$, with the minimum RMSE achieved at $C_d = 6.60$ and $b_d = 0.4224$. These optimized coefficients were subsequently incorporated into the LESO framework for vertical wind estimation. The coefficients meet the theoretical threshold that drag is proportional to the square of the wind.

output horizontal wind speed estimates based on the tilt angle of the platform. Similarly, the objective was to determine the optimal coefficients C_1 , C_0 , B_1 and B_0 for each deduced regression-curves in Eq. (3.10) and Eq. (3.11), respectively, that minimize the RMSE relative to DWL observations. Tables 3.3 and 3.4 summarize the optimized coefficients for the Linear and Quadratic methods with their corresponding RMSE values for both the CS3D and CSWX platforms. To avoid redundancy from here, the results from the C3D are presented as representative examples. Figures 3.6 and 3.7 illustrate the search space exploration and convergence characteristics of the DEO process, em-

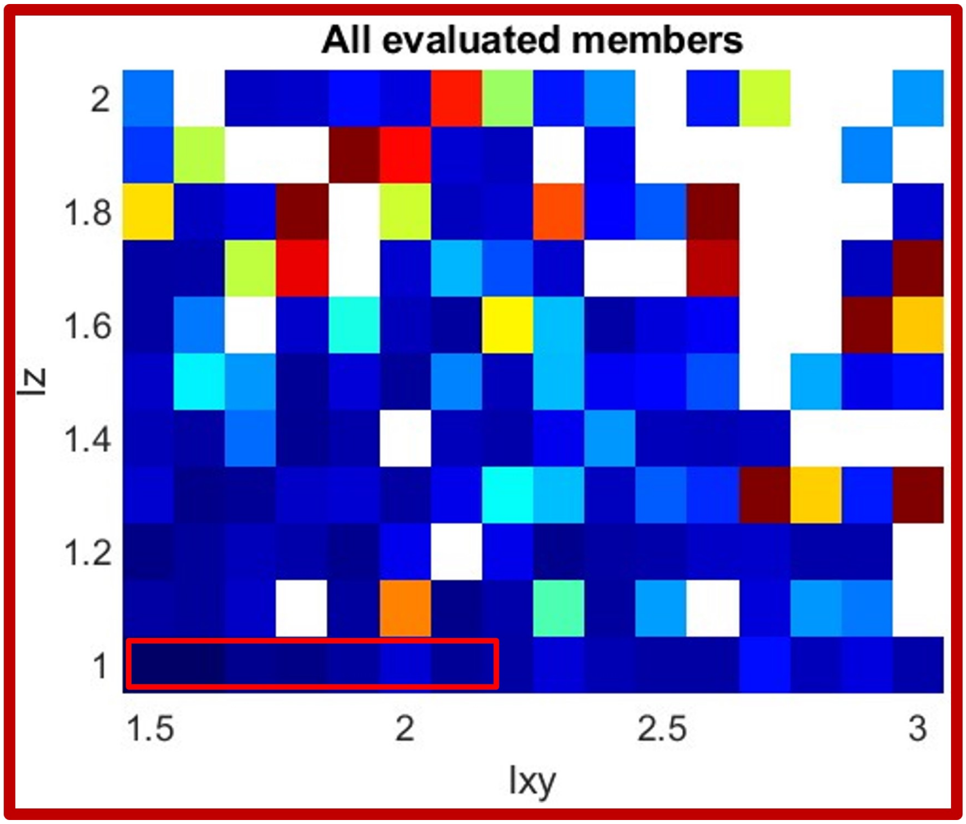

Figure 3.5: Evaluated distribution of the gain of the observer (lxy, lz) obtained through DEO for the CSWX platform. Convergence is observed near $(lxy, lz) \approx (2, 1.5)$, with the minimum RMSE achieved at $lxy = 2$ and $lz = 1.5$. These optimized coefficients were subsequently incorporated into the LESO framework for vertical wind estimation.

phasizing the regions where regression coefficients converge to values that minimize the RMSE.

Table 3.3: Optimized coefficients for the Linear algorithm applied to CS3D and CSWX platforms. The model approximates horizontal wind speed using the tilt angle of the UAS.

Platform	C_1	C_0	**RMSE**
CS3D	39	4.5	1.650
CSWX	37.6	6.8	1.106

Similar to the LESO-based approach, both the Linear and Quadratic algorithms generated parameter space visualizations that illustrate how DEO explored and evaluated

Table 3.4: Optimized coefficients for the Quadratic algorithm applied to CS3D and CSWX platforms. Like the Linear model, this model also approximates horizontal wind speed using the tilt of the UAS , which is caused due to wind.

Platform	B_1	B_0	**RMSE**
CS3D	36.3	10.0	1.520
CSWX	37.1	3.8	1.106

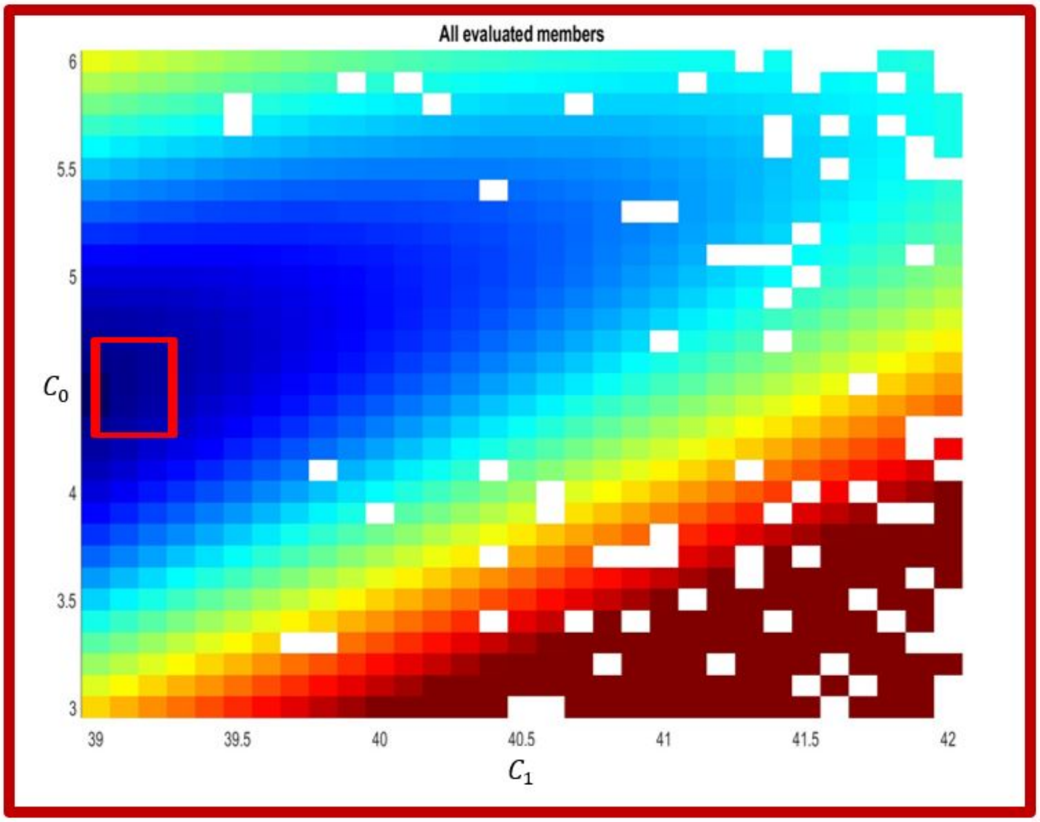

Figure 3.6: Evaluated distribution of coefficients (C_1, C_0) for the Linear algorithm obtained through DEO on the CS3D platform. Convergence is observed near $(C_1, C_0) \approx (39, 4.5)$, with the minimum RMSE achieved at $C_1 = 39$ and $C_0 = 4.5$. These optimized coefficients were subsequently incorporated into the Linear algorithm for horizontal wind estimation.

combinations of coefficients A and B to minimize RMSE. Figures 3.6 and 3.7 show the evaluated distributions for the CS3D platform. A well-defined convergence region is evident in both cases, indicating that optimal coefficient values reliably yielded the lowest estimation errors during the DEO process.

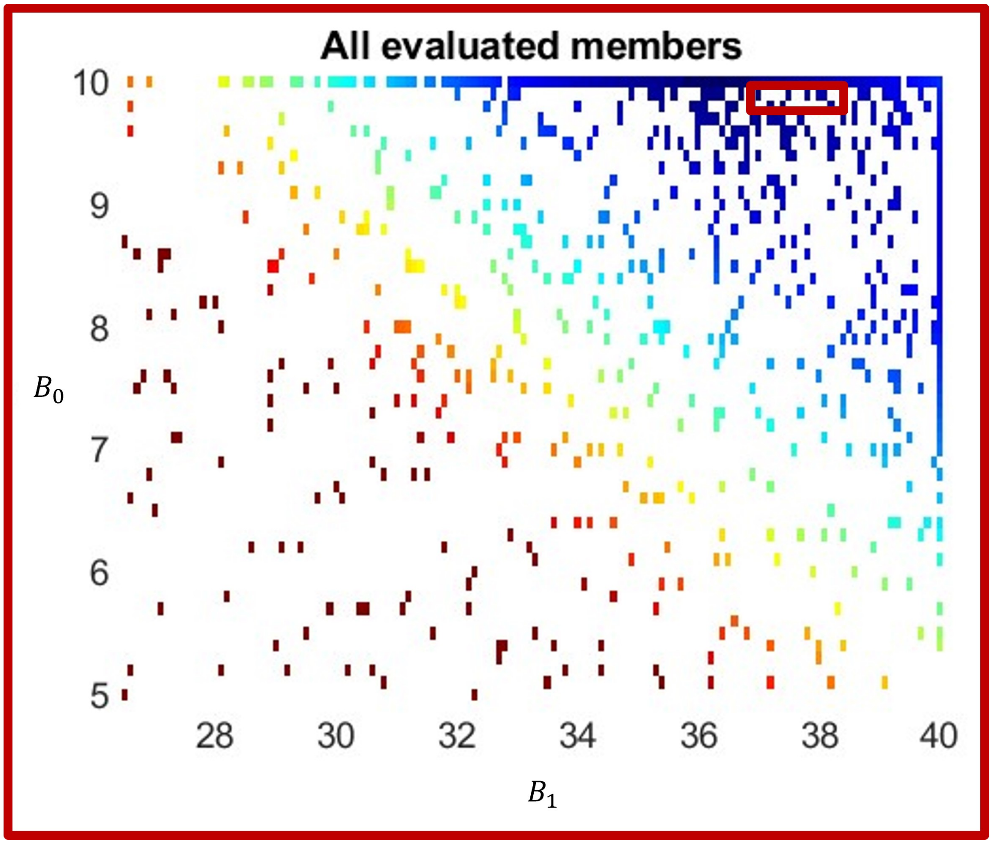

Figure 3.7: Evaluated distribution of coefficients (B_1, B_0) for the Quadratic algorithm obtained through DEO on the CS3D platform. Convergence is observed near $(B_1, B_0) \approx (36, 10)$, with the minimum RMSE achieved at $B_1 = 36.3$ and $B_0 = 10.0$. These optimized coefficients were subsequently incorporated into the Quadratic algorithm for horizontal wind estimation.

3.2.3 Quantitative Comparison between Kinematic and Dynamic Wind Estimation Methods

Following the optimization of coefficients for all presented wind estimation algorithms, a quantitative analysis was conducted to evaluate their performance on both CopterSonde platforms—CS3D and CSWX. The optimized coefficients of the DEO were applied to real-time wind estimation, and the resulting outputs were assessed against reference DWL observations.

The primary objective of this comparison was twofold: (i) to determine which CopterSonde platform delivered superior estimation accuracy across the full range of observed wind speeds, and (ii) to assess the relative performance of the LESO, Linear, and Quadratic algorithms under real-world operational conditions.

This evaluation establishes a performance benchmark that will inform subsequent phases of this research, particularly the integration of machine learning models. These future models aim to improve upon the current best-performing algorithm-platform combination by leveraging data-driven techniques for wind estimation. As such, the findings presented here serve as a critical foundation for the transition toward a machine learning solution.

The RMSE-based performance comparison across platforms and algorithms was conducted under two wind regimes: low wind (up to 10 m s^{-1}) and high wind (up to 20 m s^{-1}). For the low wind regime, data from all platforms and instrumentation were available and a complete evaluation was performed for all three algorithms—LESO, Linear, and Quadratic—across both CopterSonde platforms (CS3D and CSWX). However, in the high wind regime, a key limitation emerged: technical difficulties made the CS3D platform lack rotor RPM data, which is essential for LESO-based wind estimation. Consequently, LESO could only be evaluated using the CSWX platform under high wind conditions. Despite this limitation, LESO had already been shown to outperform the other methods in earlier analyses and was, therefore, used as a reference for evaluating the performance of the Linear and Quadratic models on both platforms in the high-wind regime.

Figure 3.8 presents a histogram of the number of valid DWL data points available for the eight vertical profiles collected during the IOPs on August 7 and October 22, 2024. Given the limited number of available IOPs, wind speed binning was performed using 1 m s^{-1} intervals to enable a more granular and statistically meaningful comparison across algorithms. The analysis was constrained to wind speeds up to 10 m s^{-1}, due to the limitations described above. For each wind speed bin, the RMSE was computed along with the their standard deviation, providing a measure of both average algorithm performance and variability within each wind speed bin.

Table 3.5 and Table 3.6 and Fig. 3.9 summarize the average RMSE and corresponding standard deviation for each 1 m s^{-1} wind speed bin for the CS3D and CSWX platforms, respectively, using the low-wind profile dataset.Although the majority of valid data points were concentrated within the 3–8 m s^{-1} wind speed range, as illustrated in Figure 3.8, bins outside this range were retained in the analysis despite their reduced statistical robustness. The inclusion of these bins enabled a broader evaluation of algorithmic performance, providing insight into the resilience and stability of each method under data-sparse conditions. This approach facilitated a more comprehensive assessment of estimation robustness across diverse wind regimes.

In addition to the dataset from August 7, vertical profile data collected during the IOP on May 22, 2024, were also incorporated into the high wind speed analysis. Therefore, Fig. 3.10 shows the number of valid data points available across the eight vertical profiles collected during the IOPs on August 7 and May 22, 2024. Due to the limited number of available IOPs, wind speed binning was performed in 2 m s^{-1} intervals, as

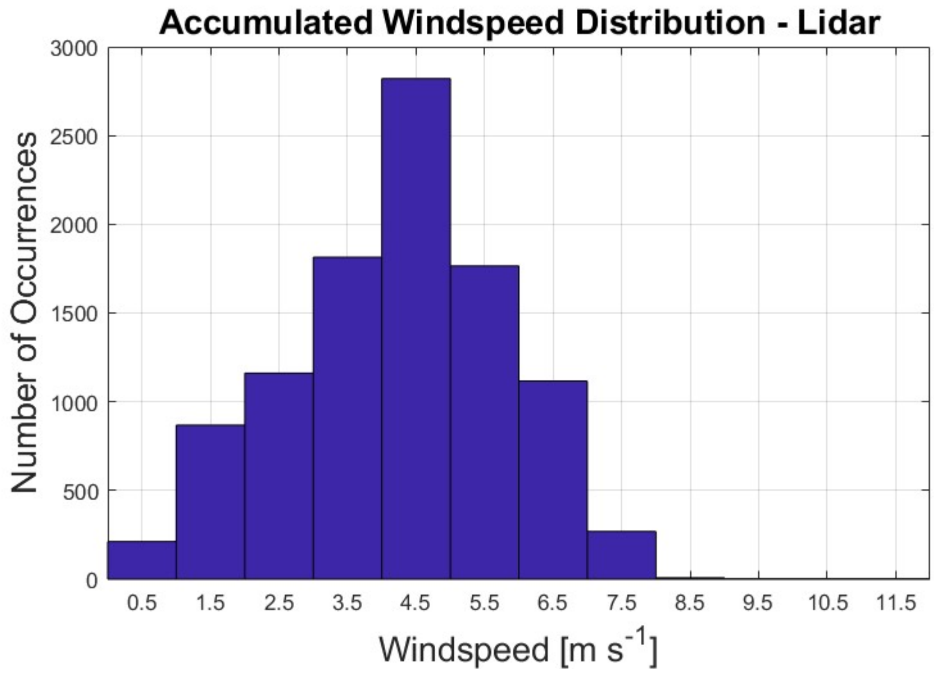

Figure 3.8: Number of valid data points per wind speed bin (1 m s^{-1} intervals) for vertical profiles collected during IOPs on August 7 and October 22, 2024. These bins were used to compute RMSE and standard deviation across different wind regimes for the evaluation of each UAS-based wind estimation method. Most of the data collected were in between 3-8 m s^{-1}, whereas the extremities may have low data; this could be useful to show how robust each algorithm is when low data is present.

Table 3.5: RMSE and standard deviation split into wind-speed bins for each algorithm on the CS3D platform in low wind.

Bin m s^{-1})	**LESO**		**Linear**		**Quadratic**	
	RMSE	Std	RMSE	Std	RMSE	Std
1–2	0.2808	0.0002	3.1950	1.7984	1.0992	0.1502
2–3	0.6591	0.0118	2.1183	1.0628	0.5806	0.0365
3–4	1.7365	0.0028	1.1308	0.9171	0.8526	0.3749
4–5	2.1184	0.7218	1.4902	1.6024	1.6594	0.7114
5–6	2.8527	0.2611	1.7906	0.5577	2.0889	1.0602
6–7	3.7751	0.3562	1.8850	0.7557	2.4741	2.3075
7–8	4.2042	0.6328	2.2293	1.9637	3.9118	2.5959
8–9	5.1396	0.0983	4.2571	0.0760	5.8101	0.3431
9–10	No Data	No Data	5.2512	0.0000	6.0547	0.0000

Table 3.6: RMSE and standard deviation split into wind-speed bins for each algorithm on the CSWX platform.

Bin ($\mathrm{m\,s^{-1}}$)	**LESO**		**Linear**		**Quadratic**	
	RMSE	Std	RMSE	Std	RMSE	Std
1–2	1.1517	0.0000	1.1261	0.0000	1.7788	0.0000
2–3	0.7957	0.0758	0.9387	0.0560	0.8912	0.0679
3–4	0.7633	0.0021	0.9687	0.0324	0.8334	0.0239
4–5	0.6199	0.0372	0.8232	0.0891	0.6737	0.0472
5–6	0.7130	0.0489	0.7360	0.0665	0.5203	0.0191
6–7	1.0887	0.3099	1.1665	0.3680	0.8951	0.2246
7–8	1.0095	0.5217	0.9794	0.4489	0.8526	0.4471
8–9	1.5000	1.7891	1.4443	1.9022	1.5317	1.3952
9–10	3.1186	1.1379	3.1513	1.5062	3.1693	0.7921
10–11	2.8213	0.0000	2.6979	0.0000	3.0570	0.0000

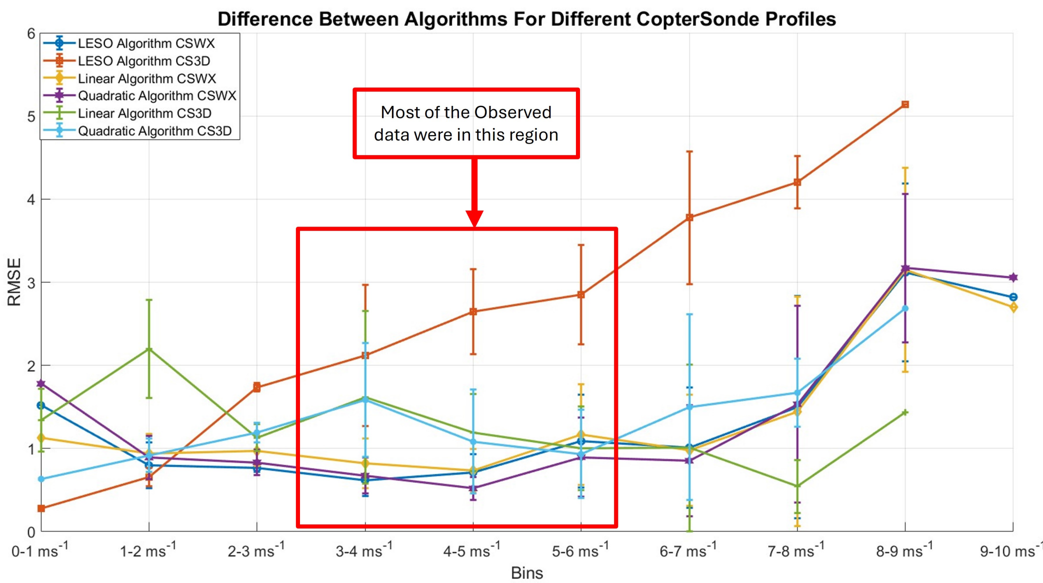

Figure 3.9: RMSE comparison of LESO, Linear, and Quadratic algorithms across CS3D and CSWX platforms under low wind speed conditions. The CSWX platform consistently exhibited lower RMSE values than CS3D, particularly under elevated wind regimes. Among all algorithms, LESO demonstrated the best overall performance in both low wind conditions.

the number of data available was higher this time, to enable a more granular and statistically meaningful algorithm comparison. The analysis was restricted to wind speeds up to 22 m s^{-1}. Similar to the low-wind case, the RMSE and its corresponding standard deviation were calculated for each wind-speed bin, providing insights into both the average performance and variability of each algorithm which are showed in Table 3.7 and Table 3.8.

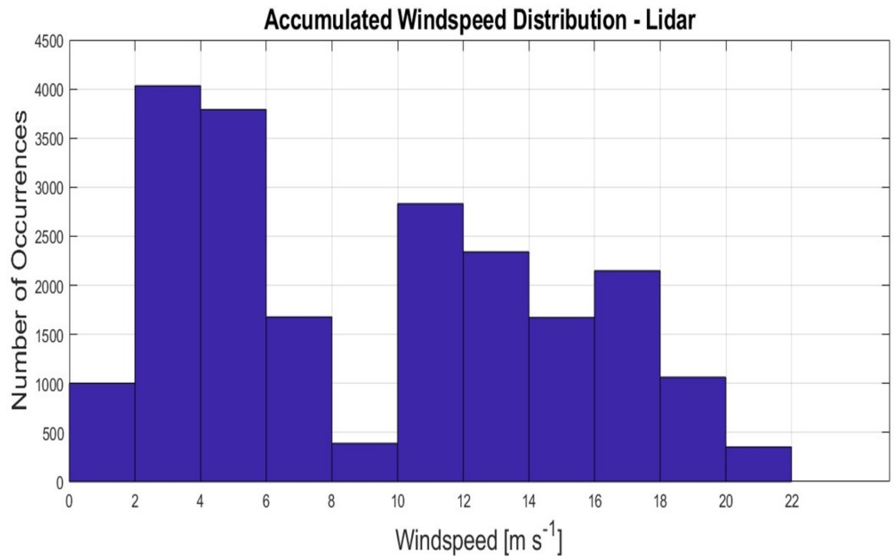

Figure 3.10: Number of valid data points per wind speed bin (2 m s^{-1} intervals) for vertical profiles collected during the IOPs on May 22 and August 7, 2024. These bins were used to compute RMSE and standard deviation across wind regimes for algorithm comparison. Most data points were concentrated in the 3–8 m s^{-1} and 10–18 m s^{-1} ranges, resulting in a bimodal distribution that represents both low and high wind speed regimes.

Figure 3.11 illustrates the RMSE under wide range of wind speed conditions for all three algorithms—LESO, Linear, and Quadratic—across both CS3D and CSWX platforms. The most reliable comparisons were derived from the 3–8 m s^{-1} and 10–18 m s^{-1} bins, as other bins contained significantly lower data for robust evaluation of the absolute accuracy comparison. As a result, the CSWX platform consistently outperformed CS3D in the low and high wind conditions, likely due to its enhanced aerodynamic characteristics, improved propulsion system, and enhanced flight stability in strong winds.

Table 3.7: RMSE and standard deviation split into wind-speed bins (high and low) for each algorithm on the CS3D platform.

Bin ($\mathrm{m\,s^{-1}}$)	**Linear**		**Quadratic**	
	RMSE	Std	RMSE	Std
0–2	1.1517	0.0000	1.1261	0.0000
2–4	0.7957	0.0758	0.9387	0.0560
4–6	0.7633	0.0021	0.9687	0.0324
6–8	0.6199	0.0372	0.8232	0.0891
8–10	0.7130	0.0489	0.7360	0.0665
10–12	1.0887	0.3099	1.1665	0.3680
12–14	1.0095	0.5217	0.9794	0.4489
14–16	1.5000	1.7891	1.4443	1.9022
16–18	3.1186	1.1379	3.1513	1.5062
18–20	2.8213	0.0000	2.6979	0.0000
20–22	2.8213	0.0000	2.6979	0.0000

Table 3.8: RMSE and standard deviation per wind speed bin (high and low) for each algorithm on the CSWX platform.

Bin ($\mathrm{m\,s^{-1}}$)	**LESO**		**Linear**		**Quadratic**	
	RMSE	Std	RMSE	Std	RMSE	Std
0–2	0.8143	0.0888	0.9459	0.0000	0.9142	0.0877
2–4	0.7109	0.0070	0.9567	0.0560	0.7144	0.0549
4–6	0.7795	0.3469	0.8685	0.0324	0.6676	0.1975
6–8	0.6322	0.0334	0.7652	0.0891	0.6459	0.0745
8–10	1.1768	0.4243	1.1558	0.0665	1.1857	0.6507
10–12	1.3352	0.4443	1.5078	0.3680	1.7564	1.2926
12–14	1.1644	0.0898	1.1869	0.4489	1.4670	0.3405
14–16	1.2665	0.3176	1.2064	1.9022	1.5061	0.2499
16–18	1.2698	0.5523	1.1995	1.5062	1.4970	0.4244
18–20	1.2833	0.5748	1.4970	0.0000	1.3149	0.5863
20–22	0.5547	0.0000	1.4870	0.0000	0.5111	0.0000

While the Quadratic method initially demonstrated comparable performance to LESO under low wind, LESO's advantage became increasingly evident as wind speed increased. LESO ultimately provided the most accurate estimates across the entire wind spectrum relative to the DWL observations, establishing it as the most reliable

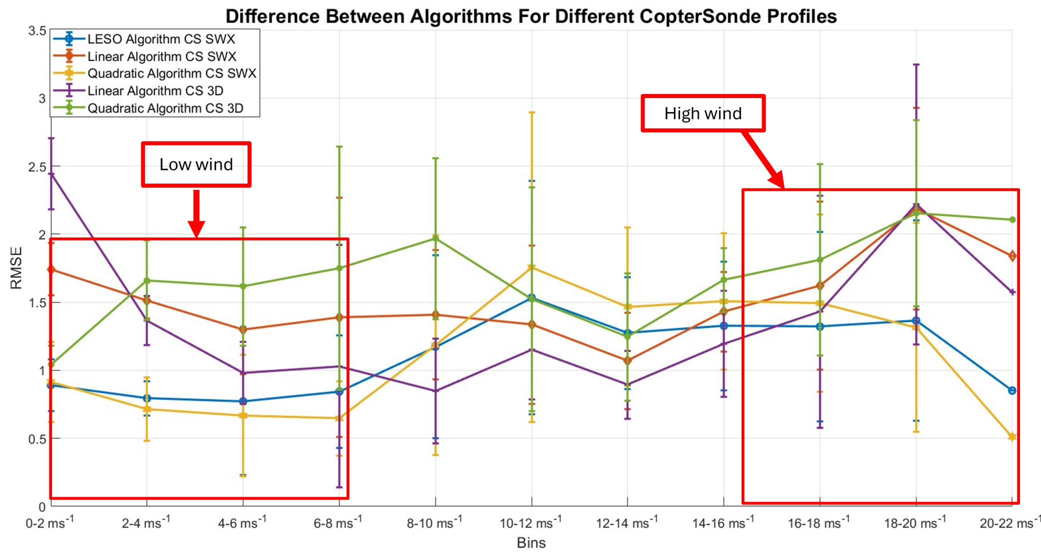

Figure 3.11: RMSE comparison of LESO, Linear, and Quadratic algorithms across CS3D and CSWX platforms under high wind speed conditions. The CSWX platform consistently exhibited lower RMSE values than CS3D, particularly under elevated wind regimes. Among all algorithms, LESO demonstrated the best overall performance in both low and high wind conditions.

algorithm tested in this study . Although limited data were available in the 6–10 $\mathrm{m\,s^{-1}}$ and 16–22 $\mathrm{m\,s^{-1}}$ wind speed bins, these regimes were still included in the analysis to evaluate the resilience of each algorithm under sparse observational conditions. In the high-wind regime (16–18 $\mathrm{m\,s^{-1}}$), the LESO algorithm exhibited relatively low RMSE, indicating robust performance despite the reduced sample size. However, its accuracy began to degrade beyond this range, where the Quadratic method demonstrated improved results. In the lower wind-speed range (6–10 $\mathrm{m\,s^{-1}}$), LESO on the CSWX platform and the Linear method on CS3D performed comparably, suggesting that certain algorithms retain calibration effectiveness even with limited data. Nonetheless, when considering overall performance across all wind regimes, LESO implemented on the CSWX platform consistently delivered the most stable and accurate results among all algorithm–platform combinations.

Figure 3.12 presents a comparative evaluation of wind estimation performance be-

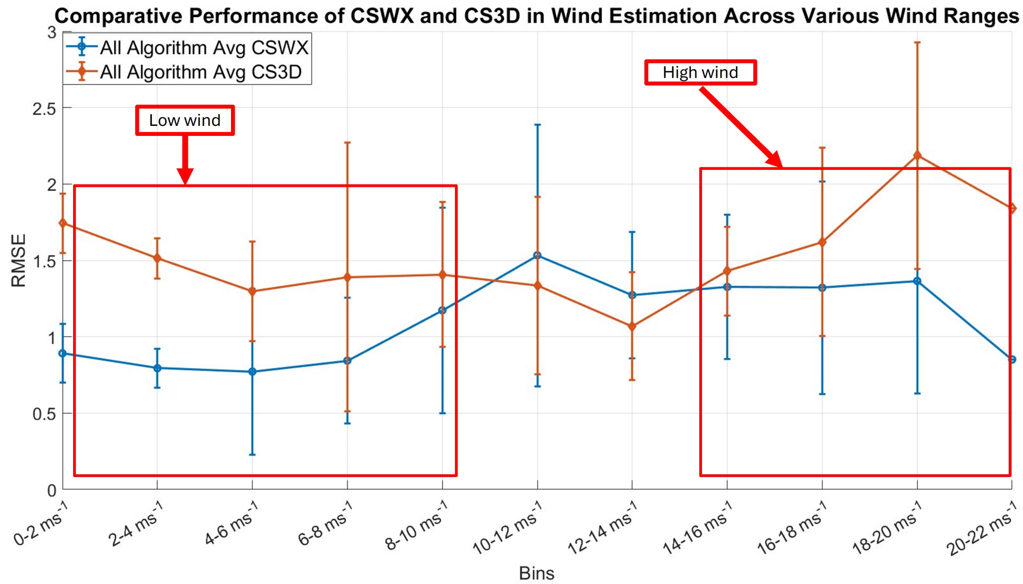

Figure 3.12: Comparison of the CSWX and CS3D platforms in wind estimation performance, averaged across all existing algorithms (LESO, Linear, and Quadratic). The analysis was based on 10 co-located vertical profiles with DWL reference measurements and employed 2 m s^{-1} binning. The CSWX shows lower RMSE in both low and high wind regimes, particularly beyond 16ms^{-1}, indicating better aerodynamic performance and stability.

tween the CSWX and CS3D platforms across various wind speed regimes. Binning was performed in 2 m s^{-1} intervals from 2 to 22 m s^{-1} to ensure statistical robustness. By averaging across all estimation methods, the influence of algorithm-specific biases was minimized, enabling a fair assessment of each platform's aerodynamic sensitivity and response. The results indicate that CSWX consistently outperformed CS3D in both low- and high-wind conditions, with comparable performance observed only in the mid-range wind regime.

3.3 Chapter Summary

This chapter presented a quantitative evaluation of three wind estimation algorithms—LESO, Linear, and Quadratic—implemented on the two different editions of the CopterSonde platform. Each method was examined with respect to its theoretical basis and computational structure, emphasizing how aerodynamic forces and platform attitude and position behaviors are used to infer horizontal and vertical wind components.

The DEO framework was employed in this study due to its effectiveness in tuning parameters within complex, non-linear, and non-differentiable optimization landscapes—conditions commonly encountered in aerodynamic modeling. By iteratively adjusting algorithm-specific coefficients to minimize the RMSE relative to DWL-derived wind measurements, DEO facilitated the calibration of the LESO, Linear, and Quadratic wind estimation methods for both the CS3D and CSWX platforms.

Beyond its calibration utility, interpretation of the DEO-optimized parameters offered additional physical insights into each platform's aerodynamic characteristics. Most of the drag model coefficients converged near the theoretical threshold of 0.5, consistent with the assumption that drag is proportional to the square of velocity. This observation supports the adequacy of a quadratic model in representing aerodynamic forces. However, the horizontal drag coefficient for the CSWX platform notably trended toward a value of 1.0, suggesting a shift toward a linear drag regime. This deviation is attributed to the CSWX's refined structural design—particularly its narrower fuselage—which likely reduces form drag and amplifies the contribution of rotor-induced aerodynamic forces. In this context, DEO not only functioned as a calibration tool but also provided diagnostic value by revealing platform-specific aerodynamic behavior.

Using DEO-optimized coefficients, algorithm performance was assessed through RMSE and standard deviation relative to colocated DWL observations across different windy days. The evaluation revealed that the LESO algorithm—particularly when implemented on the CSWX platform—yielded the highest estimation accuracy, especially under high wind conditions. This performance advantage is attributed to both the LESO algorithm's capability to estimate three-dimensional wind vectors and better predict platform behavior using a more complex dynamic model and the aerodynamic refinements of the CSWX airframe.

These results establish a performance benchmark for future wind estimation strategies. Building upon this foundation, the next chapter will introduce machine learning–based models designed to estimate wind velocity using flight telemetry data and trained using historical datasets. The objective will be to explore a new approach to estimate wind without relying on complex mathematical models tailored to the UAS platform that requires extensive characterization. Opposed to the presented methods in this chapter, machine learning techniques offer platform-agnostic techniques without relying on mathematical models that describe the behavior of the aircraft under assumptions. It only requires abundant datasets to reliably learn and adjust its internal parameters and predictions, potentially making its calibration procedures much more straightforward.

Chapter 4

UAS-based Wind Estimation using Machine Learning

As mentioned in Chapter 1, Machine Learning (ML) techniques have shown strong potential in meteorological applications, particularly for wind estimation. Among these, neural networks offer a flexible and data-driven modeling framework capable of capturing the non-linear interactions between UAS dynamics and the surrounding wind environment that the presented physics-based models, under assumptions, may fail to capture accurately. However, challenges remain in developing models that can effectively adapt to diverse flight conditions, platforms, and altitude regimes.

This chapter studies the use of machine learning (ML) approaches to estimate horizontal and vertical wind profiles from onboard UAS sensors. A series of progressively improved neural network models have been developed with the purpose of overcoming significant drawbacks of existing wind estimating techniques, such as their reliance on aerodynamic assumptions and poor generalization under non-linear and high windy conditions. Three guiding concepts were used to pick the models: interpretability, performance scalability, and domain shift robustness. Simpler models, such as multi-layer perceptrons (MLPs), were initially accepted due to their ease of implementation and low computing demand. More advanced architectures, such as domain-adversarial neural networks (DANNs), were later added to explicitly account for mismatches in training (source) and testing (target) distributions. Other deep learning models, such as convolutional or recurrent networks, were eliminated since the UAS input data was low-dimensional and tabular, lacking the spatial or temporal structures required to justify their usage and to keep the models easier to interpret. DWL was used as the reference instrument to which the neural network models were calibrated, with initial focus on horizontal wind estimation alone. A baseline neural network was implemented using the Levenberg–Marquardt optimization algorithm, with hyperparameters and architecture refined through an iterative trial-and-error process. This baseline configuration was further extended into three variants: a ReLU (Rectified Linear Unit)-activated feedforward network, a batch-trained model, and a domain adversarial neural network (DANN) designed to address domain shifts between source and target datasets.

Subsequently, WINDoe [67]—a multi-source wind estimation framework that integrates DWL, Mesonet tower, radiosonde, and UAS-derived data—was adopted again (like in the previous Chapter 3 as a more reliable and accurate ground truth reference for wind observations. Under this framework, two distinct tasks were defined: (1) three-dimensional (3D) wind estimation using CSWX data to estimate both horizontal and vertical wind , and (2) two-dimensional (2D) horizontal wind estimation using CS3D data.

Model performance was quantitatively evaluated through RMSE analysis and benchmarked against the existing LESO wind estimation algorithm, which resulted in the best existing wind estimation algorithm as demonstrated in Chapter 3 of this thesis. In addition, cross-platform comparisons were conducted under simultaneous and colocated flight conditions to assess model portability and to characterize platform-specific performance for wind vector estimation.

4.1 Field Deployment and Experimental Setup for Model Training

The selection of appropriate input parameters to the model was a critical step in preparing the machine learning pipeline. Emphasis was placed on CopterSonde's flight parameter observations that exhibit high correlation to wind-induced disturbances, thereby enabling the neural networks to learn the implicit relationships between wind effects and UAS platform dynamics. The underlying rationale was to ensure that the input-variable space reflects the fundamental physics governing wind–platform interaction.

The following seven features were selected based on their physical relevance and empirical utility for wind estimation:

1. **Rotor's Angular Velocity:** Rotor's speed (RPM) varies in real time to produce changes in the thrust vector to maintain platform stability controlled by the autopilot. This parameter changes significantly and rapidly at any given wind disturbance.

2. **Air Density:** Air density, which usually decreases with altitude, particularly within the PBL, is a key environmental variable that modulates aerodynamic forces. It is a parameter that cannot be assumed to be constant due to the wide range of weather conditions and altitudes in which the CopterSonde may operate. This is measured by combining temperature, relative humidity, and pressure, which are all measured by the CopterSonde.

3. **Roll and Pitch Angles:** These attitude control variables are adjusted by the flight controller in response to wind-induced aerodynamic drag. By tilting the aircraft, the thrust vector of the rotors does too, effectively redirecting the lifting force to compensate for drag. This keeps the UAS flying through its intended path at the

desired velocity.

4. **Three-Dimensional Velocity Components (North, East, Down):** These components are useful to determine how much the UAS was pushed away from its intended flight path due to wind. It can help to distinguish between intentional autopilot commands from actual wind disturbances.

The presented set of input variables, which involves aerodynamic, environmental, and inertial parameters, ensures that the model is exposed to physically meaningful and interpretable stimuli. Such a configuration not only supports the development of robust and generalizable neural network models for wind estimation across a wide range of atmospheric conditions but also helps the model to identify the underlying physical relationships between the input variables and wind dynamics.

Preparing the dataset for training and evaluating our machine learning models was another key component of this study. Because the University of Oklahoma lacked access to a wind tunnel, we instead collected data in an open-field environment alongside other atmospheric instruments at the flight site. This approach allowed us to capture realistic, operational wind dynamics, but it also introduced several caveats. Taylor's hypothesis of a frozen turbulent wind field was assumed to hold across the experimental area, where all instrument measurements were considered perfectly co-located—even though they were actually spaced up to 20 m apart. As a result, uncertainty increases at smaller spatial scales of the wind field. Consequently, the minimum success criterion set for any wind-estimation method is to reproduce the average wind observations recorded by our reference instruments.

The input parameters were acquired from the flight logs generated by two UAS platforms—CS3D and CSWX—while the ground truth wind measurements were obtained using a DWL system deployed in the field provided by the CIWRO and NSSL.

All flight operations took place at the Oklahoma Mesonet Washington site, a designated research facility affiliated with the University of Oklahoma.

Consistent with the data collection strategy described in Chapter 3, a standardized flight preparation and deployment protocol was followed for all UAS operations. This included battery charging, radio communication checks, structural inspections, and the issuance of NOTAMs Several datasets used for model evaluation in this chapter were also utilized in Chapter 3, thereby ensuring consistency and enabling valid cross-method comparisons. The observed horizontal wind speeds during these operations typically averaged around 10 $\mathrm{m\,s^{-1}}$, with peak values approaching 20 $\mathrm{m\,s^{-1}}$.

Table 4.1 summarizes the flight campaign dates and the number of IOPs conducted on each date. A total of 43 flight profiles were collected, of which 37 were allocated to the training dataset and 6 were retained for independent validation tests.

Table 4.1: Flight dates, number of IOPs conducted, and their partitioning into training and test datasets.

Date	**Number of IOPs**	**Training Set**	**Test Set**
22 May 2024	6	5	1
07 August 2024	4	3	1
08 September 2024	20	18	2
22 October 2024	5	4	1
20 November 2024	8	7	1
Total IOPs	43	37	6

After flight operations, post-processing was performed on the raw flight logs. Both CS3D and CSWX were equipped with onboard SD cards that functioned as black boxes, logging detailed flight information. Custom scripts, adapted from ArduPilot's open-source codebase [88], were used to extract relevant parameters—including position, velocity, orientation, and control inputs—from the binary log files.

The DWL data were recorded in NetCDF format and processed to match the Copter-

Sonde's data refresh rate of 10 Hz. A final processing pipeline was developed to synchronize the flight telemetry with DWL measurements and to export the combined dataset in CSV format for seamless ingestion by machine learning models. Figure 4.1 shows both CS3D and CSWX platforms in action along with the DWL at the Oklahoma Mesonet Washington site.

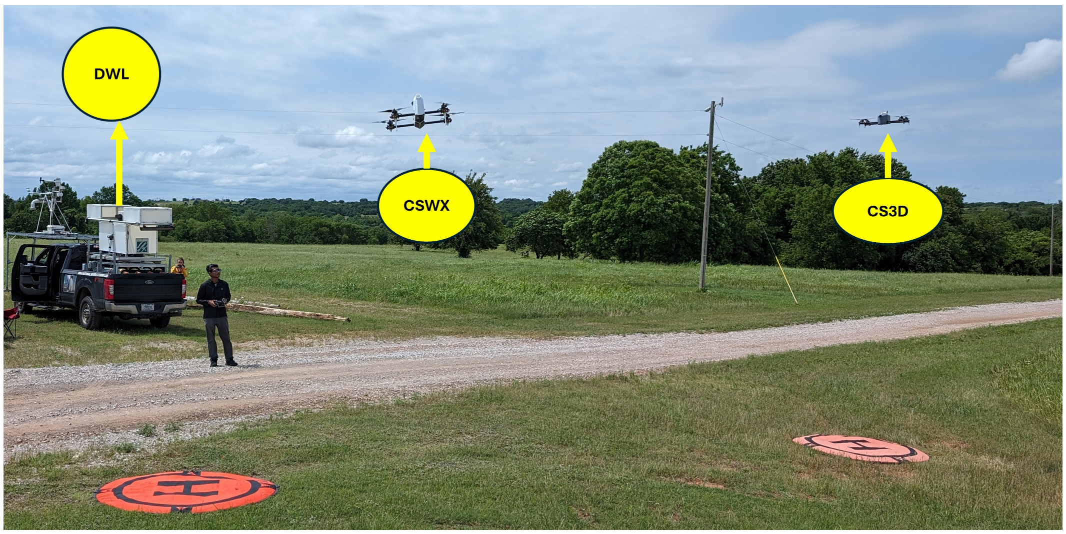

Figure 4.1: Field deployment setup showing the CS3D and CSWX UAS platforms operating in coordination with the DWL system at the KAEFS site. Both CopterSonde platforms ascended to altitudes of up to 1000 m and were flown simultaneously, maintaining an approximate lateral separation of 10 m from each other and 20 m from the DWL unit.

4.2 Machine Learning Algorithm

A fixed algorithmic workflow was followed to ensure consistency and reproducibility in training the neural network models. The process began with data collection, wherein both CS3D and CSWX UAS platforms were flown in coordination with the DWL to ensure synchronized measurements. Following each flight, relevant data were extracted from the CopterSonde binary log files and the DWL NetCDF outputs.

The next step involved identifying usable vertical profiles, as the present work is focused exclusively on vertical wind profiling. Only those flight segments that exhibited vertical ascent of the UAS were retained, while discarding the descent leg due to low measurement reliability. Once selected, input–output datasets were constructed by merging UAS telemetry data with the corresponding wind measurements from the reference instruments, into a CSV file. These structured CSV files formed the historical database for model training.

Before training the machine learning models, the dataset was randomly divided into training and validation subsets. Approximately 80% of the data were used for training, and the remaining 20% were set aside for validation. This separation ensured that the models were evaluated on previously unseen data during training. The validation set was used to monitor the cost function. To improve convergence and reduce computational cost, both input features and output variables were normalized. This preprocessing step was critical to enhance numerical stability, particularly for deep neural architectures.

Model training was performed using Mean Squared Error (MSE) as the primary cost function. A convergence threshold based on validation MSE was defined to determine the optimal stopping point. Once the model meets this criterion, the trained weights are saved.

In the final stage, the trained model was deployed to predict wind profiles from datasets that were not used during the training stage. The reconstructed wind fields were then compared against ground truth measurements to evaluate the model's performance. Figure 4.2 shows the algorithm that was followed in this study to train the ML models.

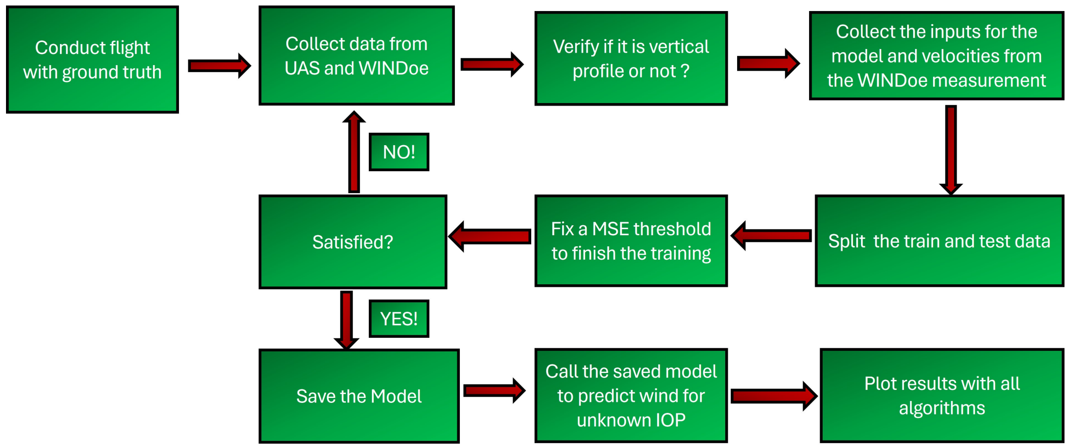

Figure 4.2: General workflow of the machine learning algorithm with CopterSonde datasets, including synchronized data collection, vertical profile selection, CSV construction, normalization, model training using MSE thresholding, and validation using untested IOPs.

4.3 Neural Network Architectures

To evaluate the viability of machine learning for wind estimation using UAS-derived data, four different neural network architectures consisting of several numbers of neurons were developed, each representing a progressively more advanced modeling strategy. In this context, a neuron refers to a computational unit that performs a weighted summation of the input features—collected from the UAS—and applies a non-linear activation function to enable the network to learn complex, non-linear relationships between flight dynamics and wind estimation. The design of these models was guided by three main objectives: improving output estimation accuracy, accelerating convergence during training, and mitigating domain shift effects between training and testing conditions. In this context, *domain shift* refers to the discrepancy between the statistical distributions of the training data (source domain) and the test environment (target domain). This shift can arise due to variations in atmospheric conditions. If not properly addressed, domain shift can degrade the performance of machine learning models, as the learned relationships from the source domain may not generalize well to the target domain.

The first model served as a baseline and employed the Levenberg–Marquardt optimization algorithm—a second-order technique well-suited for nonlinear regression problems. This network utilized a piecewise positive linear transfer function (`poslin`) as the activation function, which is a built-in library in MATLAB [89] and incorporated regularization to reduce overfitting. While effective under controlled conditions, the baseline model exhibited limited generalization across different flight scenarios and atmospheric regimes.

To enhance the model's ability to capture the nonlinear interactions between the UAS dynamics and the ambient wind field, the second architecture adopted the ReLU activation function [90] and used the Adam (Adaptive Moment Estimation) optimizer [91]. This combination provided more stable convergence and better adaptability to diverse hyperparameter configurations. As a result, the model achieved improved generalization under a broader range of meteorological conditions.

Recognizing that performance could still degrade due to distributional shifts of features in between training and validation flight data, the third architecture introduced a simple domain adaptation mechanism. This third model retained the ReLU–Adam configuration but included a fine-tuning stage using a subset of the labeled test data i.e., input samples from the target domain accompanied by their corresponding ground truth wind estimates [92]. This step allowed the model to adapt to new domains with minimal computational overhead, thereby enhancing training robustness.

The fourth and final architecture implemented a DANN [93], specifically designed to learn domain-invariant features through adversarial training. This model consisted of a shared feature extractor, a label predictor, and a domain classifier, trained using a gradient reversal layer. By explicitly accounting for domain shift during the training process, this architecture aimed to provide one of the most consistent generalization performances across varying platforms and environmental conditions.

4.4 Optimizers and Activation Functions

Optimization algorithms play a fundamental role in training deep neural networks by minimizing the cost function through iterative parameter updates, similar to the DEO implemented in Chatper 3. The choice of an optimizer directly influences the convergence rate, computational efficiency, and generalization capability of a neural network model [94]. In this study, several optimization algorithms were tested across the different neural network models. The following section briefly discusses the theoretical foundation and characteristics of the optimizers utilized in this work.

Activation functions also play a crucial role in neural networks by introducing nonlinearity, allowing the network to model complex patterns and relationships within the data. In this study, ReLU [90] activation function was employed extensively across all developed models due to its simplicity, computational efficiency, and proven effectiveness in deep learning applications.

Levenberg–Marquardt

The Levenberg–Marquardt (LM) algorithm is a second-order optimization method specifically designed for solving nonlinear problems through a least-squares method. Initially introduced by Levenberg and later refined by Marquardt [95], the algorithm combines the rapid convergence properties of the Gauss–Newton method with the stability of gradient descent [95].

The LM algorithm is particularly well-suited for applications where the cost function is expressed as the sum of squared errors, as is typically done in supervised regression tasks such as Linear and Quadratic wind estimation from Eqns (3.10) and (3.11).

It operates by solving a local trust-region subproblem at each iteration, minimizing a linearized approximation of the objective function within a constrained step size.

Given a nonlinear real function $F : \mathbb{R}^n \to \mathbb{R}^m$ and its corresponding Jacobian matrix J, the parameter update $\mathbf{p}$ is computed by solving the linear system:

$$\left(J^\top J + \lambda X^\top X\right)\mathbf{p} = -J^\top \mathbf{f} \tag{4.1}$$

where λ is the damping parameter that controls the step size, and X is a diagonal scaling matrix that accounts for the relative sensitivity of each parameter. For small values of λ, the algorithm behaves similarly to the Gauss–Newton method, promoting faster convergence when near the optimal solution. For larger values of λ, it approximates gradient descent, offering increased stability in the early stages of training or when far from the minimum. The damping parameter is dynamically adjusted at each iteration based on the agreement between the estimated and actual reductions in the residual norm, guiding the optimization toward convergence [95].

In the present study, the LM algorithm was applied to the first neural network model due to its ability to provide rapid and stable convergence on moderately sized networks with relatively low-dimensional input features. This baseline model used a (`poslin`) activation function and incorporated regularization to mitigate overfitting. While the LM-based network performed effectively in structured and controlled settings, it exhibited limited generalization in the presence of domain shifts. These limitations motivated the transition to alternative optimizers and architectures in subsequent models.

Stochastic Gradient Descent (SGD)

SGD is one of the earliest and most widely used optimization algorithms for training neural networks [94, 96]. It updates model parameters based on the gradient of the cost function computed over a mini-batch of data. The update rule for SGD is given by

$$\mathbf{Q}_{t+1} = \mathbf{Q}_t - \eta \nabla_{\mathbf{Q}} \mathcal{L}(\mathbf{Q}_t) \tag{4.2}$$

where $\mathbf{Q}_t$ denotes the model parameters at iteration t, η is the learning rate, and $\mathcal{L}$ represents the cost function [94]. While SGD is computationally efficient and robust, its convergence can be slow, particularly in the presence of noisy gradients or non-convex loss landscapes.

RMSprop Optimizer

RMSprop, introduced by Hinton [97], improves upon SGD by using an adaptive learning rate for each parameter. It maintains a decaying average of past squared gradients to normalize the parameter updates. The update rules for RMSprop are given by:

$$P_t = \beta P_{t-1} + (1-\beta) g_t^2 \tag{4.3}$$

$$\mathbf{Q}_{t+1} = \mathbf{Q}_t - \frac{\eta}{\sqrt{P_t} + \epsilon} g_t \tag{4.4}$$

where g_t is the gradient at iteration t, P_t is the moving average of squared gradients, β is the decay rate, and ϵ is a small constant for numerical stability. RMSprop accelerates convergence in non-stationary problems or those with sparse gradients, although it may exhibit limited generalization capability in certain tasks [94].

Adam Optimizer

Adam, proposed by Kingma and Ba [91], combines the advantages of RMSprop and momentum-based methods. It maintains exponential moving averages of both the gradients (m_t) and the squared gradients (P_t). The update rules are given by:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{4.5}$$

$$P_t = \beta_2 P_{t-1} + (1 - \beta_2) g_t^2 \tag{4.6}$$

where β_1 and β_2 represent the decay rates for the first and second moments, respectively. The bias-corrected estimates are computed as:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{4.7}$$

$$\hat{P}_t = \frac{P_t}{1 - \beta_2^t} \tag{4.8}$$

The parameter update is performed as:

$$\mathbf{Q}_{t+1} = \mathbf{Q}_t - \frac{\eta}{\sqrt{\hat{P}_t} + \epsilon} \hat{m}_t \tag{4.9}$$

Adam has become a widely adopted optimizer in deep learning tasks due to its fast convergence and stability. However, its generalization performance may be inferior to SGD in some applications [94].

AdamW Optimizer

AdamW, introduced by Loshchilov and Hutter [98], modifies the original Adam algorithm by decoupling weight decay from the gradient update step. In the standard Adam algorithm, weight decay interacts with the adaptive learning rate, potentially degrading regularization effectiveness. AdamW addresses this by applying weight decay directly to the parameter values:

$$\mathbf{Q}_{t+1} = \mathbf{Q}_t - \eta \left(\frac{\hat{m}_t}{\sqrt{\hat{P}_t} + \epsilon} + \lambda \mathbf{Q}_t \right) \tag{4.10}$$

where λ is the weight decay coefficient. This decoupling improves the generalization capability and regularization performance of the optimizer.

Table 4.2 summarizes the key characteristics and performance considerations of the optimizers employed in this study, as reported in prior research [96].

Table 4.2: Summary of optimization algorithms used in this study.

Optimizer	Key Characteristics
SGD	Simple and robust; sensitive to learning rate; slow convergence without momentum enhancement.
RMSprop	Adaptive learning rate with moving average of squared gradients; effective for non-stationary or sparse gradient problems.
Adam	Combines RMSprop and momentum; fast convergence; requires careful hyperparameter tuning for generalization.
AdamW	Decouples weight decay from gradient update; improved generalization and regularization performance.

ReLU Activation Function

The ReLU activation function, introduced by Hahnloser [99], is mathematically defined as:

$$f(x) = \max(0, x) \tag{4.11}$$

where x denotes the input to a neuron. The function outputs zero for negative inputs and returns the input value unchanged for non-negative inputs. This simple thresholding mechanism promotes sparsity within the network and facilitates efficient gradient propagation during the training phase. Figure 4.3 shows the working procedures for the ReLU function. The derivative of the ReLU function, which is essential for the backpropagation algorithm, is given by:

$$f'(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.12}$$

ReLU has become the default activation function in deep learning due to its ability to mitigate the vanishing gradient problem, which commonly affects traditional activation functions such as sigmoid and tanh (Hyperbolic Tangent) [90] [100]. The vanishing gradient problem occurs when gradients used for updating model parameters during backpropagation become exceedingly small as they are propagated backward through many layers. This often leads to negligible updates in early-layer weights, effectively stalling learning and causing the network to fail in capturing important lower-level features [100]. Unlike those functions, ReLU preserves gradient flow for positive activations, allowing for faster and more stable convergence during optimization.

Moreover, ReLU enables the network to learn piecewise linear functions, providing enhanced capability to approximate complex non-linear mappings between inputs and outputs. This property was particularly advantageous in the present study, given the non-linear nature of the wind estimation problem using UAS telemetry data.

Despite its advantages, ReLU is susceptible to the “dying ReLU” problem, where neurons may become inactive (outputting zero for all inputs) if weight updates lead to predominantly negative activations. However, with proper weight initialization and learning rate tuning, this issue can be effectively mitigated [90].

In this work, the ReLU activation function was employed in all hidden layers of Models 2 (FF-NN), 3 (FT-NN), and 4 (DANN), enabling the neural networks to efficiently learn the complex wind dynamics associated with CopterSonde operations under diverse atmospheric conditions.

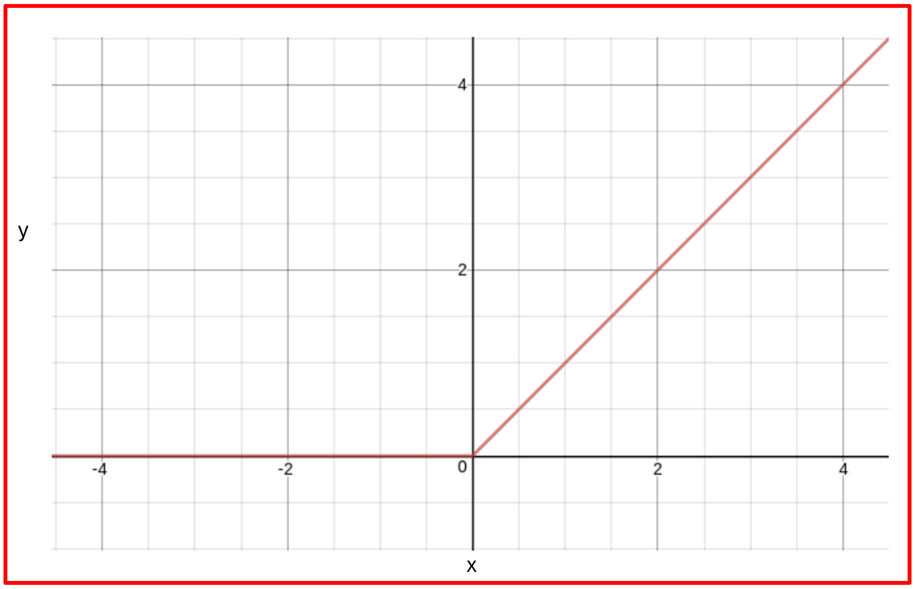

Figure 4.3: The ReLU activation function produces 0 as an output when $x < 0$, and produces a linear output with slope 1 when $x > 0$. Image taken from [90].

Domain-Adversarial Neural Network

The DANN (Domain-Adversarial Neural Network), introduced by Ganin [93], came up with a framework for unsupervised domain adaptation by embedding domain invariance directly into the feature learning process of deep neural networks. The fundamental objective of DANN has been used to learn feature representations that are simultaneously discriminative for source domain classification and invariant across domains, thereby reducing the discrepancy between source and target distributions.

The theoretical foundation of DANN has been established upon the domain adaptation theory proposed by Ben-David [101], which provides a generalization bound on the target domain risk. For a hypothesis class $\eta \in \mathcal{H}$ (set of all possible functions or models that a learning algorithm can explore to find a solution), the risk in the target domain $R_{\mathcal{D}_T}(\eta)$ has been bounded as:

$$R_{\mathcal{D}_T}(\eta) \leq R_{\mathcal{D}_S}(\eta) + \frac{1}{2} d_{\mathcal{H}}(\mathcal{D}_S^X, \mathcal{D}_T^X) + \lambda, \tag{4.13}$$

where $R_{\mathcal{D}_S}(\eta)$ denotes the source risk, $d_{\mathcal{H}}$ represents the $\mathcal{H}$-divergence between the marginal feature distributions of source and target domains, and λ corresponds to the risk of the ideal joint hypothesis across both domains. The $\mathcal{H}$-divergence is defined as:

$$d_{\mathcal{H}}(\mathcal{D}_S^X, \mathcal{D}_T^X) = 2 \sup_{\eta \in \mathcal{H}} \left| \Pr_{\mathcal{D}_S^X}[\eta(x) = 1] - \Pr_{\mathcal{D}_T^X}[\eta(x) = 1] \right|. \tag{4.14}$$

This theoretical result implies that low source risk alone is insufficient to guarantee robust generalization on the target domain. Instead, minimizing both the source risk and the domain discrepancy is essential.

To achieve domain-invariant representation learning, DANN incorporates a domain classifier in parallel with the label predictor and employs a novel component known as the *Gradient Reversal Layer* (GRL) [93]. The model architecture is composed of the following modules:

- Feature extractor $G_f(x; \theta_f)$,
- Label predictor $G_y(G_f(x); \theta_y)$,
- Domain classifier $G_d(G_f(x); \theta_d)$, connected via the GRL.
- (θ_f) , (θ_y) , (θ_d) are the learnable parameters in the feature extractor, label predictor, and domain classifier.

During training, the network simultaneously minimizes the classification loss on the labeled source data and maximizes the domain classification loss to encourage indistinguishability between domains. The GRL functions as the identity operator during the forward pass and multiplies the gradient by -1 during the backward pass. This adversarial mechanism forces the feature extractor to learn representations that confuse the domain classifier, thus promoting domain invariance.

Model parameters are updated using stochastic gradient descent as follows:

$$\theta_f \leftarrow \theta_f - \mu \left(\frac{\partial \mathcal{L}_y}{\partial \theta_f} - \lambda \frac{\partial \mathcal{L}_d}{\partial \theta_f} \right), \tag{4.15}$$

$$\theta_y \leftarrow \theta_y - \mu \frac{\partial \mathcal{L}_y}{\partial \theta_y}, \tag{4.16}$$

$$\theta_d \leftarrow \theta_d - \mu \frac{\partial \mathcal{L}_d}{\partial \theta_d}, \tag{4.17}$$

where μ denotes the learning rate.

Figure 4.4 presents the DANN architecture, illustrating the adversarial interaction between the domain classifier and feature extractor through the GRL.

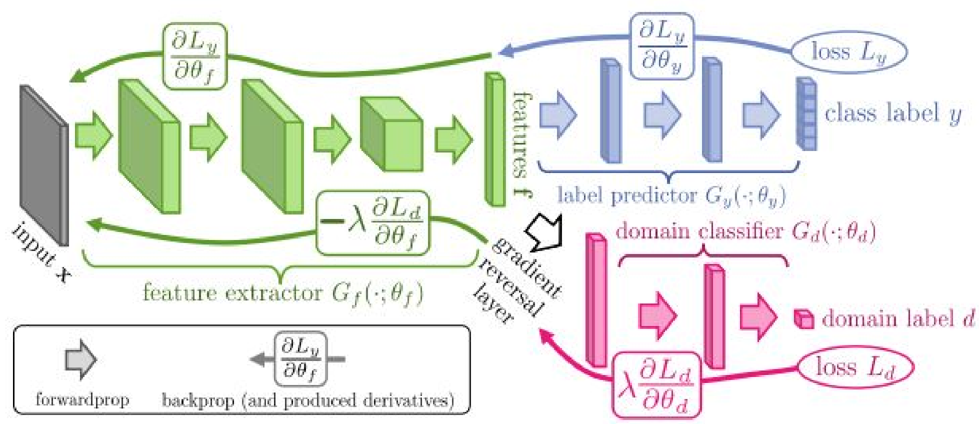

Figure 4.4: The DANN architecture includes a deep feature extractor (green) and label predictor (blue) forming a conventional feedforward model. A domain classifier (red) is connected through a gradient reversal layer that inverts the gradient during backpropagation. This setup enforces domain invariance by encouraging the feature extractor to confuse the domain classifier while optimizing classification performance. Image taken from [93]

In this study, DANN was used to address the limitations imposed by data scarcity across the full range of observed winds in the field. Conventional neural networks trained on specific datasets often exhibit degraded performance when deployed in operational environments that differ from the training conditions—a phenomenon commonly known as domain shift. Wind estimation tasks are particularly susceptible to this issue, as atmospheric conditions encountered during post-training deployment can vary substantially from those present during data collection for the ML training, both in terms of wind intensity and broader meteorological variability.

The DANN framework addresses this challenge by learning domain-invariant feature representations, thereby enabling the model to generalize effectively across different operating conditions. This approach is especially beneficial when training data are disproportionately concentrated in low-wind regimes while high-wind observations remain sparse. Through the integration of adversarial training with a domain classifier, the DANN facilitates the extraction of features that are predictive of wind yet agnostic to the domain origin of the input. As a result, the model demonstrates enhanced adaptability to unseen or underrepresented wind scenarios, offering more resilience to low-data training and more reliable wind estimation under diverse environmental conditions.

4.5 Implementation of Machine Learning for CopterSonde UAS Wind Estimation

This section presents the implementation of ML techniques for estimating wind from CopterSonde UAS telemetry data. The following subsections describe the design, training, and evaluation of multiple neural network architectures, each tailored to estimate wind, address specific challenges s domain variability, and real-time deployment feasibility

4.5.1 Model 1: Levenberg–Marquardt Feed Forward Neural Network

The first neural network model (LM-NN) was developed using the Levenberg–Marquardt (LM) optimization algorithm, which is particularly suited for nonlinear regression tasks where the cost function is formulated as the sum of squared errors [95]. This model served as the baseline framework for wind estimation using CopterSonde UAS data.

The network architecture was configured as a fully connected feedforward neural network consisting of six hidden layers. The number of neurons in each layer followed a pyramidal structure with the configuration [128, 64, 32, 16, 8, 4] from the first to the sixth hidden layer, respectively. This configuration was selected, after performing extensive trial and error to balance model complexity, computational cost, and estimation accuracy.

All hidden layers employed the positive linear transfer function (`poslin`) as an activation function, equivalent to the ReLU, to capture non-linear system dynamics. The output layer utilized a linear activation function (`purelin`), appropriate for continuous regression tasks.

To mitigate overfitting, regularization (a technique to prevent overfitting) was applied using a weight decay parameter of 0.0009. The input data were normalized using z-score normalization based on the statistics of the training and validation datasets. The test data were normalized using the same parameters to ensure consistency.

Model training employed an early stopping strategy based on validation performance, with a maximum limit of 150 training epochs and a patience threshold of 10 consecutive validation failures. The final trained model was stored and subsequently used to infer wind profiles from flight logs not part of the training set.

While the LM-based model demonstrated stable convergence, its generalization capability was limited when applied to unseen flight conditions, particularly under domain shift scenarios. This observation motivated the development of more advanced models utilizing alternative optimization strategies and domain adaptation techniques.

Table 4.3 summarizes the detailed network architecture and configuration parameters of Model 1 (LM-NN). The table outlines the optimizer, activation functions, layer configuration, and regularization techniques and highlights the strengths and limitations associated with this model.

Table 4.3: Configuration and performance summary of Model 1 (LM-NN), including network architecture, training strategies, and key observations.

Parameter	**Details**
Optimizer	Levenberg–Marquardt (LM)
Activation Function (Hidden Layers)	`poslin` (ReLU equivalent)
Activation Function (Output Layer)	`purelin` (Linear)
Hidden Layers	6 layers: [128, 64, 32, 16, 8, 4] neurons
Regularization	Weight decay = 0.0009
Data Normalization	Z-score normalization based on training data statistics
Training Strategy	Early stopping with a maximum of 150 epochs and a patience of 10 validation failures
Used Platform	MATLAB + GPU (NVIDA A100)
Computing Time	4 hrs.
Strength	Stable convergence for moderate-sized datasets; simple architecture
Limitation	Limited generalization capability under domain shift and higher computational cost during training

4.5.2 Model 2: Rectified Linear Unit and Adaptive Moment Estimation Optimized Feed Forward Neural Network

The second neural network model (FF-NN) was developed using the PyTorch framework, incorporating several architectural and optimization improvements over the baseline LM-based model. This model employed the Adam optimizer, which combines the advantages of adaptive learning rates with weight decay regularization, thereby offering increased robustness for training deep neural networks.

Prior to selecting Adam, several other optimization algorithms were evaluated, including SGD, SGD with momentum, RMSprop, and AdamW. The Adam optimizer demonstrated superior performance in capturing the underlying dynamics of the CopterSonde UAS and was therefore adopted for this model.

The network architecture retained a fully connected feedforward structure consisting of five hidden layers. Neurons were configured in a pyramidal pattern with the structure [64, 32, 16, 8, 4]. The ReLU activation function was applied after each hidden layer, chosen for its effectiveness in modeling non-linear relationships and mitigating the vanishing gradient problem commonly encountered in deeper networks. The model employed the MSE as the cost function, defined as the average of the squared differences between the predicted wind values and the corresponding reference measurements. Input parameters and target outputs were normalized using z-score normalization, with scaling parameters derived exclusively from the training dataset to ensure consistency across both training and testing phases.

The optimal model configuration was selected based on the minimum loss achieved on the validation dataset during the training process. Early stopping was applied using a learning rate scheduler, and overfitting was implicitly avoided by saving the model weights and biases corresponding to the epoch yielding the lowest test loss.

Table 4.4 summarizes the detailed network architecture and configuration parameters of Model 2 (FF-NN). The table outlines the optimizer, activation functions, layer configuration, and regularization techniques and highlights the strengths and limitations associated with this model.

Table 4.4: Configuration and performance summary of Model 2 (FF-NN) , including network architecture, training strategies, and key observations.

Parameter	**Details**
Optimizer	Adam optimizer with learning rate =0.0000005
Activation Function (Hidden Layers)	ReLU
Activation Function (Output Layer)	Linear
Hidden Layers	5 layers: [64, 32, 16, 8, 4] neurons
Regularization	Weight decay weight decay = 0.00001
Learrning Rate Scheduler	`ReducedLROnPlateau`
Data Normalization	Z-score normalization (StandardScaler) using training set statistics
Used Platform	Python + CPU (I5-133U)
Computing Time	4 hours of Optuna modeling and 2 hours of model training .
Strength	Improved generalization performance and robustness across unseen data
Limitation	Requires careful tuning of hyperparameters for optimal performance

4.5.3 Model 3: Feed Forward Neural Network Using Pretrained and Fine-Tuning

The third model (FT-NN) developed in this study extends the architecture and training strategy of Model 2 (FF-NN), introducing a fine-tuning approach designed to mitigate domain shift between the source (training) and target (testing) datasets. This strategy could be applicable in UAS-based wind estimation, where a small quantity of labeled target domain data may be available to improve model adaptability and generalization performance.

The architecture of this model remained identical to that of Model 1 (LM-NN), consisting of a fully connected feedforward neural network with six hidden layers configured in a pyramidal structure of [128, 64, 32, 16, 8, 4] neurons. ReLU activation functions were applied after each hidden layer to capture non-linear system dynamics effectively.

The initial training phase was conducted on the source domain data using the RMSprop optimizer with a learning rate of 0.00001. Model selection was based on the lowest validation loss observed during training, thereby avoiding overfitting and ensuring generalization within the source domain.

Following the initial training, a fine-tuning phase was introduced. A small set of labeled target domain samples was fed to the model to adjust the model weights, enabling the network to adapt to the characteristics of the target environment. Fine-tuning was performed using the Adam optimizer at a learning rate of 0.0001 for 5000 iterations. This step preserved the knowledge learned from the source domain while allowing for improved adaptability to the target domain.

Compared to the DANN, this model provides a computationally simpler approach to domain adaptation, particularly when labeled target domain data are available.

Table 4.5 summarizes the detailed network architecture and configuration parameters of Model 3 (FT-NN). The table outlines the optimizer, activation functions, layer configuration, and regularization techniques and highlights the strengths and limitations associated with this model.

Table 4.5: Configuration and performance summary of Model 3 (FT-NN), including network architecture, training strategies, and key observations.

Parameter	**Details**
Optimizer (Initial Training)	RMSprop optimizer with learning rate = 0.00001
Optimizer (Fine-Tuning)	Adam optimizer with learning rate = 0.0001
Architecture Components	Feedforward Neural Network with Fine-Tuning Strategy
Activation Function	ReLU in all hidden layers
Hidden Layers	6 layers: [128, 64, 32, 16, 8, 4] neurons
Training Strategy	Initial training on source domain; fine-tuning on small labeled target domain samples
Used Platform	Python + CPU (I5-I33U)
Computing Time	4 hours for the feedforward network and 2 hours for the batch tuned network.
Strength	Practical domain adaptation with minimal computational cost
Limitation	Requires labeled target data for fine-tuning; adaptation performance is limited by the quantity and quality of target samples

4.5.4 Model 4: Domain-Adversarial Neural Network

The fourth neural network model (DANN) implemented in this study is the DANN, designed to explicitly address do main shift between the source (training) and target (testing) datasets. This challenge is particularly relevant for UAS-based wind estimation, where flight conditions, environmental variability, and sensor characteristics may differ significantly between the data collection and deployment phases.

The network architecture employed ReLU activations in all hidden layers to effectively capture non-linear system dynamics. The feature extractor contained three hidden layers configured with [128, 64, 32] neurons, while the label predictor included three layers with [16, 8, 1] neurons. The domain classifier was designed with two hidden layers of [16, 2] neurons for binary classification of the domain labels.

Model training was performed using two separate cost functions: MSE loss for wind estimation and cross-entropy loss for domain classification. The total loss was computed as the sum of the label loss and the domain loss. The Adam optimizer was employed with a learning rate of 0.0009 and a batch size of 16.

This adversarial training strategy enabled the model to learn robust features that were less sensitive to domain-specific variations, thereby improving generalization performance across diverse flight scenarios.

Table 4.6 summarizes the detailed network architecture and configuration parameters of Model 4 (DANN). The table outlines the optimizer, activation functions, layer configuration, and regularization techniques and highlights the strengths and limitations associated with this model.

Table 4.6: Configuration and performance summary of Model 4 (DANN), including network architecture, training strategies, and key observations.

Parameter	**Details**
Optimizer	Adam optimizer with learning rate = 0.0009
Architecture Components	Feature Extractor, Label Predictor, Domain Classifier
Activation Function	ReLU in all hidden layers
Feature Extractor Layers	[128, 64, 32] neurons
Label Predictor Layers	[16, 8, 1] neurons
Domain Classifier Layers	[16, 2] neurons with Gradient Reversal Layer
cost functions	MSE Loss (Regression) + Cross-Entropy Loss (Domain Classification)
Data Normalization	Z-score normalization (StandardScaler) using source domain statistics
Training Strategy	batch size of 16, best model selected based on wind prediction performance
Strength	Explicit handling of domain shift, improved generalization across varying flight conditions
Platform	Python + GPU (NVIDIA A100)
Computing Time	10 hrs.
Limitation	Higher computational cost and increased model complexity due to adversarial training setup

4.6 Comparison of the Machine Learning Models with the Baseline Wind Estimation Algorithms

Following the training of the neural network models described in the previous sections, the learned model weights and biases were saved for subsequent deployment and evaluation. The final comparison between the machine learning-based wind estimation models and the existing wind estimation algorithms was performed during an independent testing phase. In this section of the study, the saved model weights were utilized to estimate horizontal wind speed from CopterSonde input flight parameters.

In this section, Figure 4.7 to Figure 4.11 show the wind estimation results from multiple test flights conducted on different days. The outputs from all four machine learning models were compared against the Linear, Quadratic, and LESO wind estimation algorithms. Additionally, ground truth wind measurements obtained from the DWL were included as a reference benchmark.

A quantitative assessment of model performance conducted using RMSE analysis is also shown in Figure 4.6. The RMSE was calculated in a bin-wise manner across different wind speed intervals to evaluate the accuracy of each algorithm. This approach facilitated a comprehensive comparison between the traditional wind estimation methods and the machine learning-based models, thereby highlighting the strengths and limitations of each approach under varying wind conditions. The results reveal that among the machine learning models, Model 2 (FF-NN) and Model 4 (DANN) consistently outperformed the traditional methods in low-to-moderate wind regimes (0–12 $\mathrm{m\,s^{-1}}$), achieving RMSE values comparable to or lower than the LESO algorithm. Notably, Model 4 maintained relatively strong performance even in high-wind bins (exceeding 12 $\mathrm{m\,s^{-1}}$), likely due to its domain adaptation capability, whereas the performance of Model 3 showed moderate performance in low wind and deteriorated significantly as wind increased.

The procedure for estimating horizontal wind speed using the saved trained neural network weights and new input data collected from the CopterSonde platform is outlined in Algorithm 1. This algorithm was used for the first three models to retrieve wind from the real time inputs by using the saved weights after training the models.

Algorithm 1 Wind Estimation using Trained Neural Network Weights

1: Load the trained model weights and biases.
2: Load the mean and standard deviation values used during model training for input and output normalization.
3: Extract the required input paramters from the UAS's flight log file (binary format).
4: Normalize the new input data using:

$$X_{\text{norm}} = \frac{X_{\text{new}} - X_{\text{mean}}}{X_{\text{std}}}$$

5: Pass the normalized input through the trained neural network in a sequential, layer-wise manner.
    For each hidden layer:
    Multiply input by weights, add bias, and apply ReLU activation.
6: At the final output layer, apply a linear activation to obtain the predicted normalized wind speed.
7: Denormalize the predicted output using:

$$T_{\text{pred}} = T_{\text{pred\_norm}} \times T_{\text{std}} + T_{\text{mean}}$$

8: Apply bias correction if required based on the model calibration (This is done only for Model 1 (LM-NN)).
9: Output the final estimated horizontal wind speed.

For the DANN model, the wind estimation procedure differs from the previous models due to the presence of two distinct components: a feature extractor and a label predictor. The feature extractor is responsible for learning domain-invariant representations, while the label predictor estimates the horizontal wind speed from these extracted features. Algorithm 2 describes the procedure for estimating horizontal wind speed using the trained DANN model weights.

Algorithm 2 Wind Estimation using Trained DANN Model Weights

1: Load the trained weights and biases of the feature extractor and label predictor networks.
2: Load the mean and standard deviation values used during model training for input and output normalization.
3: Extract the required input features from the drone's onboard binary log file.
4: Replace any missing (NaN) or infinite values in the input data with zero.
5: Normalize the new input data using :

$$X_{\text{norm}} = \frac{X_{\text{new}} - X_{\text{mean}}}{X_{\text{std}}}$$

6: Pass the normalized input through the feature extractor network:
    For each layer:
    Multiply input by weights, add bias, and apply ReLU activation.
7: Pass the extracted features through the label predictor network to obtain the predicted normalized wind speed.
8: Denormalize the predicted wind speed using:

$$T_{\text{pred}} = T_{\text{pred\_norm}} \times T_{\text{std}} + T_{\text{mean}}$$

9: Output the final estimated horizontal wind speed.

An extended RMSE analysis was conducted to compare the performance of the developed machine learning-based wind estimation models with the existing wind estimation algorithms with the CSWX. The RMSE analysis was designed to evaluate model performance across different wind speed regimes, with particular focus on distinguishing between low-wind and high-wind scenarios.

To achieve this, a binning approach—applied similarly to the method used in the DEO-based evaluation (see Chapter 3)—was adopted. In addition to computing the RMSE, the number of data points within each wind speed bin was recorded to assess model performance in data-sparse regimes, particularly under high-wind conditions.

Table 4.7 presents the bin-wise RMSE comparison between the existing wind estimation algorithms (LESO, Linear, and Quadratic) and the four developed machine learning models (ML Model 1 (LM-NN) to Model 4 (DANN)) for the CopteSonde

CSWX. Table 4.8 provides the corresponding standard deviation of errors within each bin to evaluate the consistency and stability of each model's predictions.

The RMSE analysis indicated that the LESO algorithm consistently performed as the most accurate of the existing methods, yielding lower RMSE values across the majority of wind speed bins. Among the machine learning models, Model 2 (FF-NN) (a feedforward neural network using ReLU activation) and Model 4 (DANN) demonstrated performance comparable to the LESO algorithm, particularly under low-wind conditions.

As wind speed increased and the number of available data points within each bin decreased, Model 4 (DANN) exhibited superior performance over the LESO algorithm. This highlights the capability of the DANN architecture to leverage domain adaptation and maintain estimation accuracy in high-wind conditions. Model 2 (FF-NN) also maintained performance close to LESO across different wind speed bins, with indications of improved estimation capability at higher wind speeds.

Table 4.7: Bin-wise RMSE comparison between baseline wind estimation algorithms and the proposed machine learning models (ML) for horizontal wind estimation.

Bin ($\mathbf{m\,s^{-1}}$)	LESO	Linear	Quadratic	ML1 LM-NN	ML2 FF-NN	ML3 FT-NN	ML4 DANN
0–2	1.5768	1.4950	1.8218	2.5602	2.5339	4.2520	2.1466
2–4	0.6929	0.9858	0.8790	1.2157	1.9841	2.7869	1.1843
4–6	0.4907	0.5953	0.4857	0.9671	1.5389	1.4424	0.9003
6–8	0.4734	1.5219	0.8462	0.4885	1.0493	2.5438	0.9886
8–10	0.3671	0.8296	0.3298	0.3305	0.4259	3.5157	0.2329
10–12	0.9486	0.9227	0.9189	0.9638	0.7562	5.1222	0.8145
12–14	0.7494	0.4155	0.6980	1.2601	0.7484	6.2294	0.3553
14–16	2.0746	1.7251	2.2654	2.7431	2.0440	8.1974	1.2866
16–18	3.0096	2.5865	2.8722	3.1665	2.5180	10.007	1.6420

Figure 4.5 presents the bar chart illustrating the distribution of the number of data

Table 4.8: Standard deviation of errors within each wind speed bin for baseline algorithms and proposed machine learning models (ML), illustrating model consistency and stability.

Bin ($\mathbf{m\,s^{-1}}$)	LESO	Linear	Quadratic	ML1 LM-NN	ML2 FF-NN	ML3 FT-NN	ML4 DANN
0–2	0.6206	0.8119	0.5485	0.3430	3.0575	0.0208	1.0053
2–4	0.0294	0.0458	0.0472	0.5874	2.3660	0.0896	0.6938
4–6	0.0630	0.0318	0.0084	0.1624	0.6164	0.1106	0.1002
6–8	0	0	0	0	0	0	0
8–10	0	0	0	0	0	0	0
10–12	0	0	0	0	0	0	0
12–14	0	0	0	0	0	0	0
14–16	0	0	0	0	0	0	0
16–18	0	0	0	0	0	0	0

points available within each wind speed bin. It is evident from the figure that the majority of the collected data correspond to wind speeds within the range of 2–6 $\mathrm{m\,s^{-1}}$. Conversely, the highest wind speed bin (16–18 $\mathrm{m\,s^{-1}}$) contained the fewest data points, emphasizing the scarcity of measurements in higher wind conditions.

Figure 4.6 shows the RMSE comparison between the baseline wind estimation algorithms and the developed machine learning models across various horizontal wind speed bins. It can be observed that Model 2 (FF-NN) and Model 4 (DANN) emerged as the best-performing machine learning models among the four models developed in this study.

Model 3 (FT-NN) demonstrated moderate performance in low wind speed bins where sufficient data were available; however, its accuracy degraded as wind increased due to its failure to adapt to the new domain. This behavior aligns with expectations, as Model 3 (FT-NN) represented the initial attempt to incorporate domain adaptation within this framework.

In contrast, both Model 2 (FF-NN) and Model 4 (DANN) exhibited superior per-

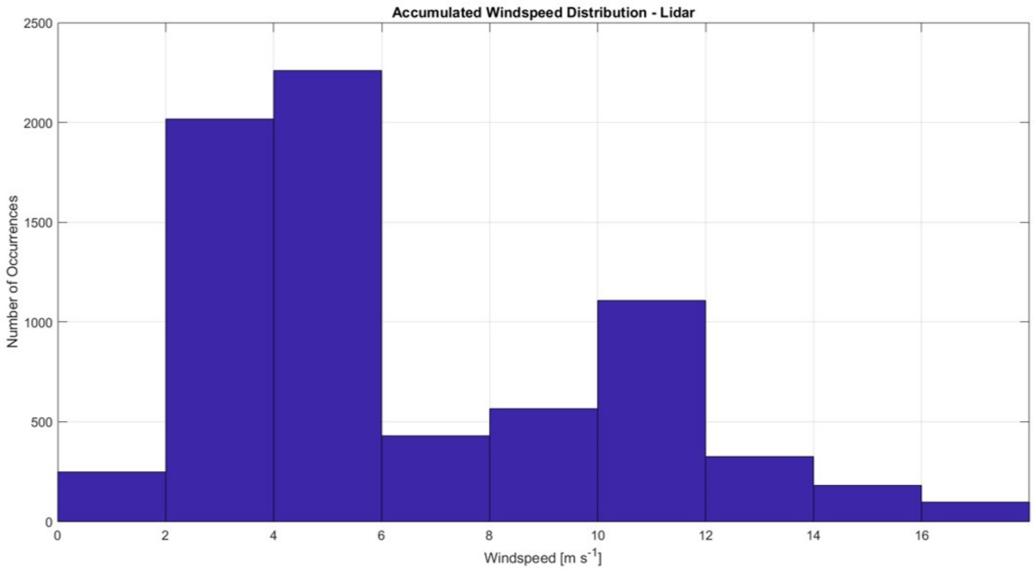

Figure 4.5: Distribution of the number of data points in each wind speed bin. The figure illustrates that the majority of the data were collected within 2–6 $\mathrm{m\,s^{-1}}$, while significantly fewer data points were available at higher wind speeds (16–18 $\mathrm{m\,s^{-1}}$).

formance, maintaining accuracy even at higher wind speeds despite limited data. These models successfully estimated wind speeds with high accuracy and, in several cases, outperformed the LESO algorithm.

Figure 4.7 illustrates the wind estimation performance of the four proposed machine learning models on a low wind speed day (during the SCALES field campaign), where the average wind speed was approximately 5 $\mathrm{m\,s^{-1}}$. In this scenario, Model 1 (LM-NN) and Model 4 (DANN) demonstrated the best performance, achieving the lowest RMSE among all models. Notably, Model 1 (LM-NN) and Model 4 (DANN) even outperformed the LESO algorithm, which has been identified as the most accurate among the existing traditional wind estimation methods, and also showed the potential of the ML models in wind estimation.

Table 4.9 presents a quantitative comparison of the proposed machine learning models and the baseline wind estimation algorithms (LESO, Linear, and Quadratic). The

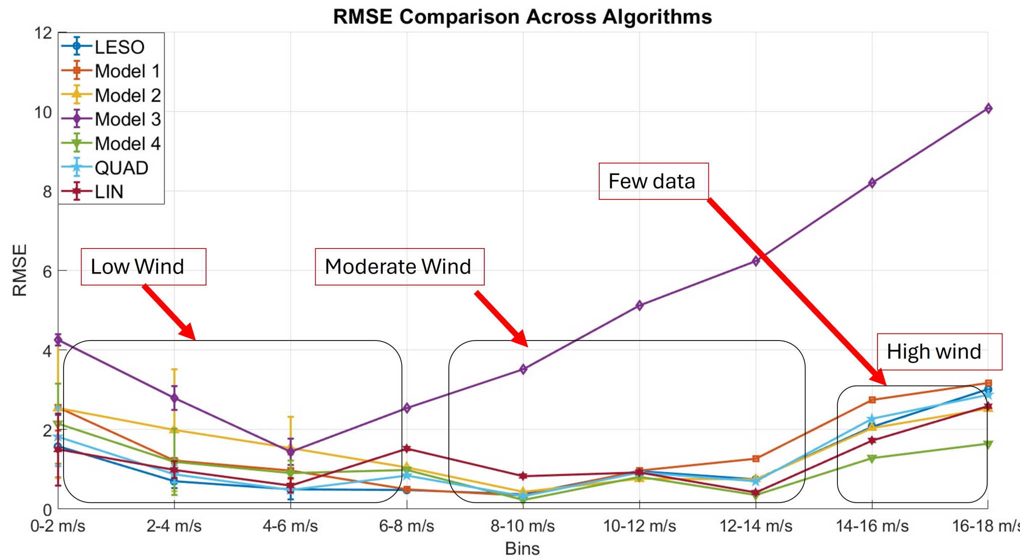

Figure 4.6: RMSE comparison of existing wind estimation algorithms and developed machine learning models (ML). The figure highlights that Model 2 (FF-NN) and Model 4 (DANN) provided the most accurate wind estimation across all wind speed bins, particularly outperforming existing algorithms at higher wind speeds where data availability was limited.

evaluation metrics include the MAE, RMSE, and standard deviation (StdDev) of the estimation errors.

Table 4.9: Performance comparison of the machine learning models with the baseline wind estimation algorithms (LESO, Linear, and Quadratic) for a low wind day in terms of MAE, RMSE, and standard deviation (StdDev) of the error.

Method	**MAE($\mathrm{m\,s^{-1}}$)**	**RMSE($\mathrm{m\,s^{-1}}$)**	**StdDev($\mathrm{m\,s^{-1}}$)**
ML1 LM-NN	0.4863	0.6223	0.6077
ML2 FF-NN	0.7730	1.0440	0.9900
ML3 FT-NN	2.7950	3.0213	1.6004
ML4 DANN	0.5462	0.6975	0.6609
LESO	0.5593	0.6897	0.6516

Figure 4.8 presents the wind estimation performance of the developed machine learning models for another low wind speed day, where the maximum observed wind speed was approximately 4.5 $\mathrm{m\,s^{-1}}$. In this scenario, Model 1 (LM-NN) achieved the

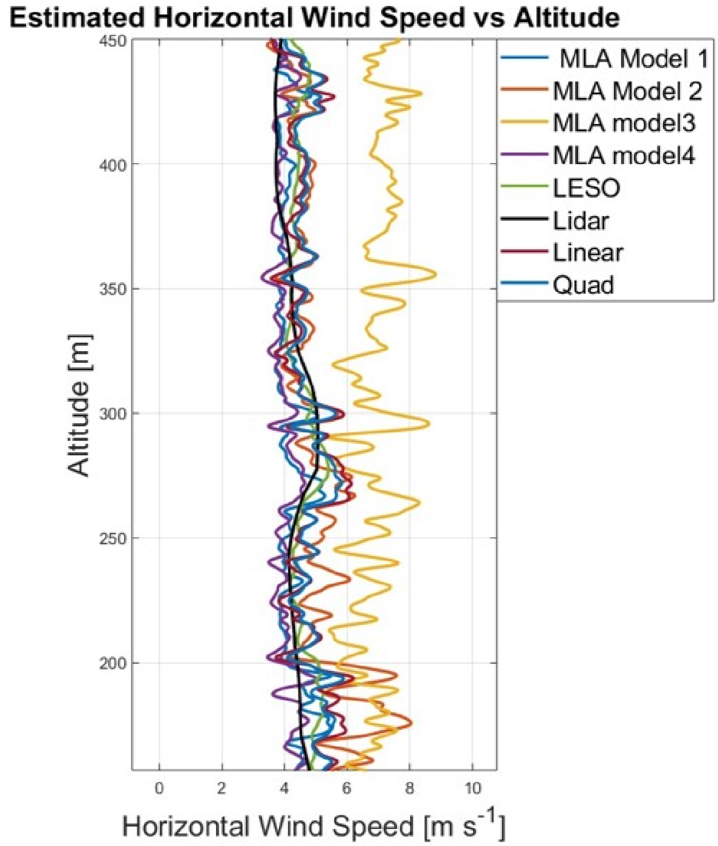

Figure 4.7: The wind estimation performance of the proposed machine learning models and baseline algorithms was evaluated for a representative low-wind case (wind speed $\approx 5\ \mathrm{m\,s^{-1}}$). Under these conditions, the LESO algorithm exhibited reliable performance, serving as a strong baseline. Among the machine learning models, all except Model 3 demonstrated encouraging results. Both Model 1 (LM-NN) and Model 4 (DANN) achieved RMSE lower than those of LESO, indicating their effectiveness in capturing low-wind dynamics. Model 2 (FF-NN) also performed well, although its RMSE was marginally higher than that of LESO. In contrast, Model 3 (FT-NN) failed to represent the wind profile accurately and exhibited poor performance in this low-wind scenario.

best performance among the machine learning models. However, the LESO algorithm outperformed all models, attaining the lowest RMSE and MAE values.

Table 4.10 provides the detailed error metrics, including MAE, RMSE, and standard deviation (StdDev) of the estimation errors for the developed models and the existing wind estimation algorithms.

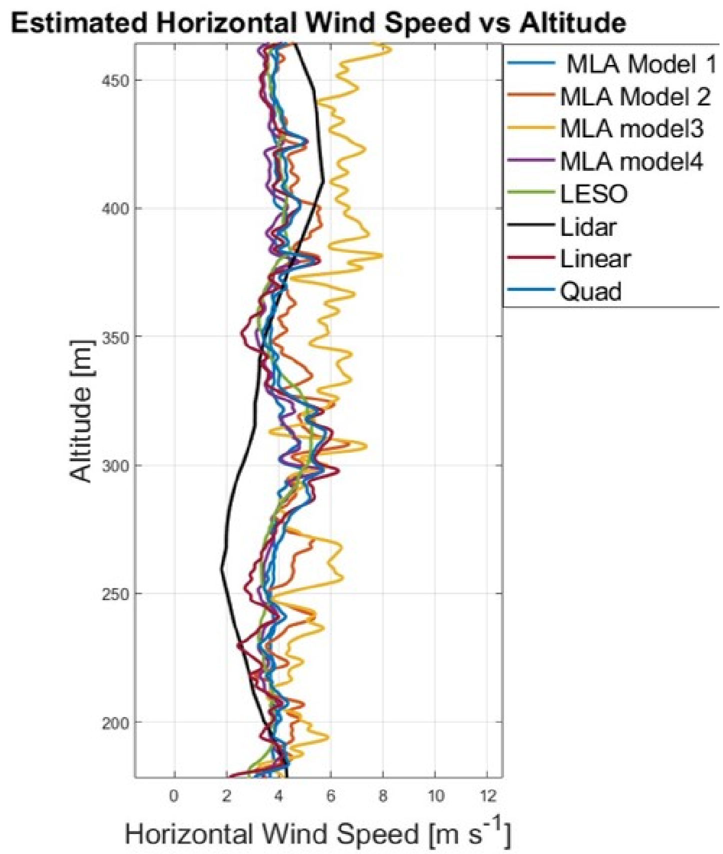

Figure 4.8: The wind estimation performance of the developed machine learning models and existing algorithms was also evaluated for the same low windy day (wind speed $\approx 4.5\ \mathrm{m\,s^{-1}}$). In this scenario, Model 1 (LM-NN) and Model 4 (DANN) again delivered the most accurate results, both outperforming the LESO algorithm. Model 2 (FF-NN) achieved a performance nearly identical to LESO, indicating reliable but non-superior generalization. Model 3 (FT-NN) demonstrated moderate success; however, its higher RMSE suggests an inability to effectively capture domain adaptation under these conditions.

Figure 4.9 presents the wind estimation performance of the developed machine learning models for another low windy day, where the maximum observed wind speed was approximately 6 $\mathrm{m\,s^{-1}}$. In this scenario, Model 1 (LM-NN) achieved the best performance. Model 4 (DANN) and LESO were performing identically, and Model

Table 4.10: Performance comparison of the machine learning models with the existing wind estimation algorithms (LESO, Linear, and Quadratic) for a low wind day in terms of MAE, RMSE, and standard deviation (StdDev) of the error.

Method	**MAE($\mathrm{m\,s^{-1}}$)**	**RMSE($\mathrm{m\,s^{-1}}$)**	**StdDev($\mathrm{m\,s^{-1}}$)**
ML1 LM-NN	1.0520	1.2575	1.2183
ML2 FF-NN	1.2905	1.5841	1.4932
ML3 FT-NN	2.1772	2.6307	2.0696
ML4 DANN	1.0788	1.2924	1.2303
LESO	1.1783	1.4366	1.4336

2 (FF-NN) was having a slightly higher RMSE, likewise on the other low windy day. Once again, the Model 3 (FT-NN) had the highest RMSE.

Table 4.11 provides the detailed error metrics, including MAE, RMSE, and standard deviation (StdDev) of the estimation errors for the developed models and the existing wind estimation algorithms.

Table 4.11: Performance comparison of the machine learning models with the existing wind estimation algorithms (LESO, Linear, and Quadratic) for a low wind day in terms of MAE, RMSE, and standard deviation (StdDev) of the error.

Method	**MAE($\mathrm{m\,s^{-1}}$)**	**RMSE($\mathrm{m\,s^{-1}}$)**	**StdDev($\mathrm{m\,s^{-1}}$)**
ML1 LM-NN	1.5248	1.9240	1.9065
ML2 FF-NN	2.0715	2.4968	2.496($\mathrm{m\,s^{-1}}$)0
ML3 FT-NN	3.4013	3.8825	2.0864
ML4 DANN	1.7872	2.1975	2.1973
LESO	1.7838	2.1920	2.1418

Figure 4.10 shows the wind estimation performance of the developed machine learning models for a moderate wind speed day, where the maximum observed wind speed was approximately 9 $\mathrm{m\,s^{-1}}$. On this day, Model 2 (FF-NN) outperformed all other algorithms, including the existing LESO algorithm, achieving the lowest RMSE and MAE values.

Additionally, Model 1 (LM-NN) and Model 4 (DANN) also demonstrated excellent

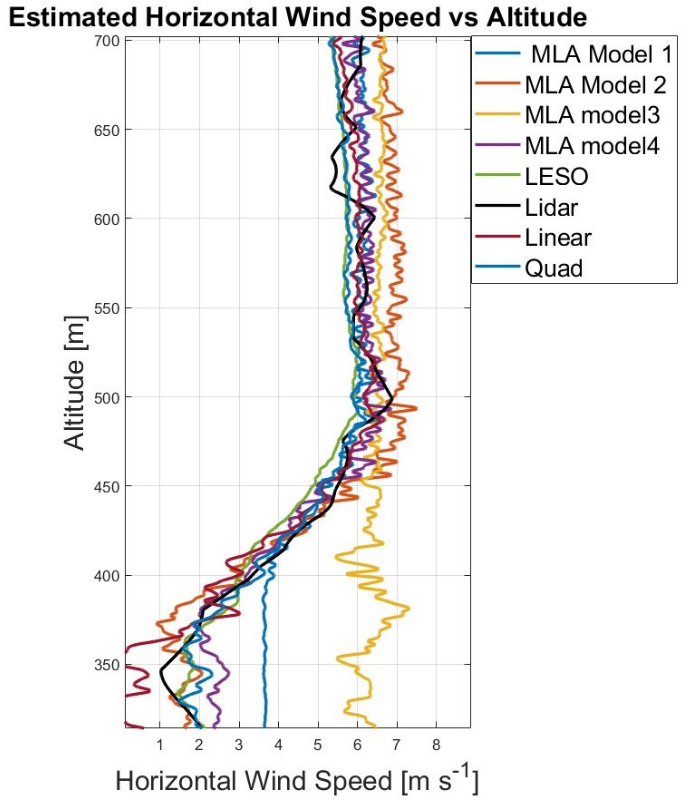

Figure 4.9: The wind estimation performance of the developed machine learning models and existing algorithms was also evaluated for another low-wind case (wind speed $\approx 6\,\mathrm{s}^{-1}$). In this scenario, Model 1 (LM-NN) delivered the most accurate results by outperforming the LESO algorithm. Model 4 (DANN) and LESO were performing identically. Model 2 (FF-NN) achieved a performance nearly identical to LESO likewise on the other windy days, indicating reliable but non-superior generalization. Model 3 (FT-NN) demonstrated poor estimation, and its higher RMSE suggests an inability to effectively capture domain adaptation under these conditions.

performance, producing results comparable to the LESO algorithm. This highlights the robustness of the developed machine learning models, also in moderate wind conditions. However, the highest RMSE was achieved by Model 3 (FT-NN), which showed its failure to domain adaptation for moderate wind also.

Table 4.12 presents the detailed performance metrics, including MAE, RMSE, and standard deviation (StdDev) of the estimation errors for the machine learning models and the existing wind estimation algorithms.

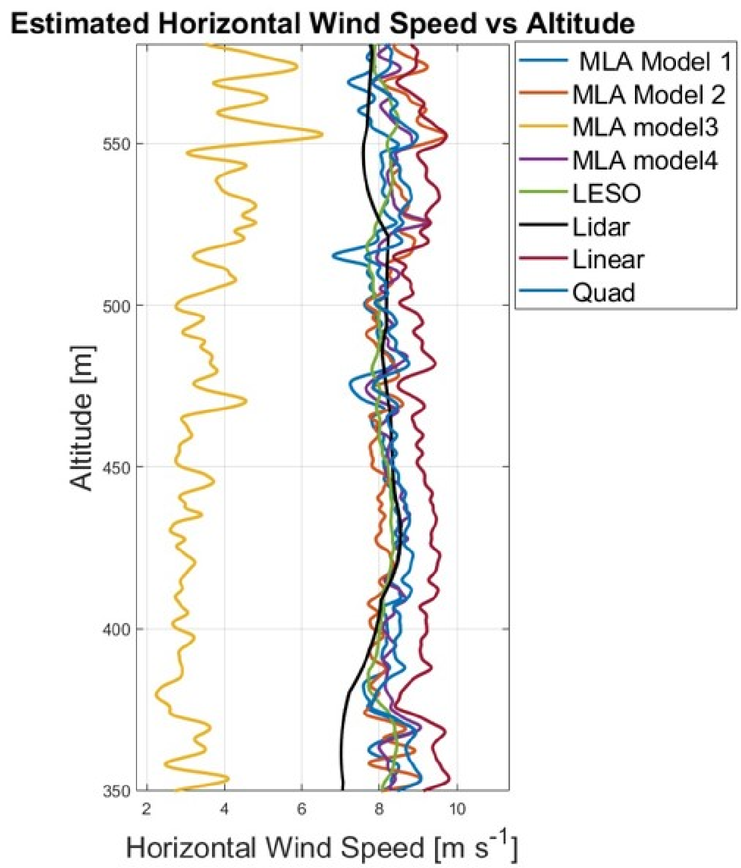

Figure 4.10: The wind estimation performance of the developed machine learning models and existing algorithms was further evaluated for a moderate wind day (wind speed $\approx 9\ \mathrm{m\,s^{-1}}$). Among all models, Model 2 (FF-NN) provided the most accurate wind estimates, outperforming the LESO algorithm. Models 1 (LM-NN) and 4 (DANN) produced results comparable to LESO, indicating consistent performance across varying wind intensities. Model 3 (FT-NN), however, yielded the highest RMSE, confirming its continued inability to capture wind dynamics effectively, even under moderate wind conditions.

Figure 4.11 illustrates the wind estimation performance of the developed machine learning models compared to the existing wind estimation algorithms for a high wind day, where the maximum observed wind speed reached approximately 20 $\mathrm{m\,s^{-1}}$.

In this scenario, the DANN (Model 4 (DANN)) demonstrated its capability to

Table 4.12: Performance comparison of the machine learning models with the existing wind estimation algorithms (LESO, Linear, and Quadratic) for a moderate wind day in terms of MAE, RMSE, and standard deviation (StdDev) of the error.

Method	MAE($\mathrm{m\,s^{-1}}$)	RMSE($\mathrm{m\,s^{-1}}$)	StdDev($\mathrm{m\,s^{-1}}$)
ML1 LM-NN	0.7205	0.7798	0.7486
ML2 FF-NN	0.6931	0.8905	0.8163
ML3 FT-NN	5.4215	5.5924	1.3719
ML4 DANN	0.7057	0.8856	0.7198
LESO	0.6044	0.7988	0.7988

handle domain shift effectively, transitioning from low-wind to high-wind conditions. Though the Model 2 (FF-NN) provided the least RMSE for this case, even lower than the LESO , Model 4 (DANN) also provided high performance in wind estimation, being trained with few high-wind data and achieving almost similar RMSE to LESO. This result highlights the advantage of domain adaptation techniques for real-world wind estimation applications.

Additionally, Model 1 (LM-NN) also performed well compared to the existing LESO algorithm, confirming that a feedforward network can also achieve high performance when trained with a good number of data across varying wind regimes. Conversely, Model 3 (FT-NN) exhibited poor performance in this high-wind scenario, failing to capture the wind dynamics accurately.

Table 4.13 presents the detailed performance metrics for this high-wind day, reporting the MAE, RMSE, and standard deviation (StdDev) of the estimation errors for each model and existing algorithm.

4.7 2D Wind Estimation Using CopterSonde CS3D

The ReLU-activated neural network architecture initially developed for the CSWX platform was adapted for horizontal wind estimation using data from the CS3D platform.

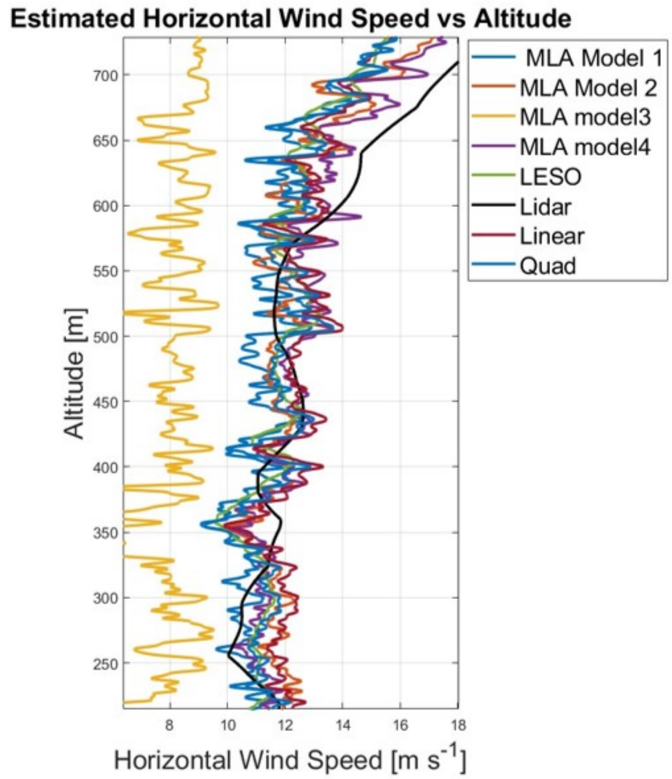

Figure 4.11: The wind estimation performance of the developed machine learning models and existing algorithms was also assessed for a high-wind scenario, with maximum wind speeds reaching approximately 20 $\mathrm{m\,s^{-1}}$. Model 2 (FF-NN) delivered the most accurate wind estimates, demonstrating strong robustness under extreme conditions. Model 4 (DANN) also exhibited high performance by effectively handling domain shift through its adversarial training framework. Model 1 (LM-NN) produced results comparable to the LESO algorithm, maintaining stable performance. In contrast, Model 3 (FT-NN) failed to capture the high-wind dynamics, likely due to its limited capacity for domain adaptation.

WINDoe was utilized as the ground truth reference for this case. This cross-platform adaptation enabled a direct comparison between the legacy CS3D system and the newer CSWX platform under identical atmospheric and operational conditions. During field deployments, both UAS platforms were flown simultaneously for most of the days, ensuring consistency in environmental exposure and flight dynamics.

The model architecture and training methodology were retained from the CSWX

Table 4.13: Performance comparison of the machine learning models with the existing wind estimation algorithms (LESO, Linear, and Quadratic) for a high-wind case in terms of MAE, RMSE, and standard deviation (StdDev) of the error.

Method	**MAE($\mathrm{m\,s^{-1}}$)**	**RMSE($\mathrm{m\,s^{-1}}$)**	**StdDev($\mathrm{m\,s^{-1}}$)**
ML1 LM-NN	1.7268	2.3545	2.0155
ML2 FF-NN	1.3991	1.9654	1.9383
ML3 FT-NN	4.5249	4.9794	2.0784
ML4 DANN	1.6814	2.2111	2.1989
LESO	1.6010	2.1731	2.0478

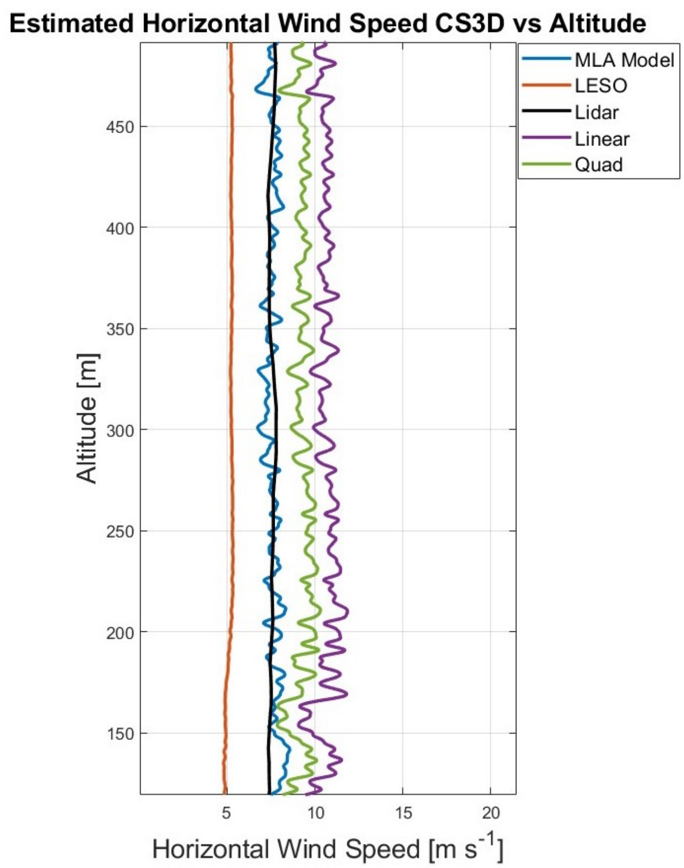

Figure 4.12: Horizontal wind estimation on CS3D for the IOP conducted on November 20, 2024, with a peak wind speed of approximately $9\,\mathrm{m\,s^{-1}}$. The ReLU-activated machine learning model significantly outperforms the LESO, Linear, and Quadratic algorithms in representing wind dynamics.

implementation; the only modification involved substituting the input dataset with flight telemetry from CS3D. The objective of this section is to evaluate whether the machine learning model preserves its performance advantage over traditional algorithms—namely LESO, Linear, and Quadratic—when applied to a different hardware configuration.

Figure 4.12 displays the estimated horizontal wind profile generated by the CS3D-based machine learning model, alongside the outputs of the existing algorithms. The results correspond to an IOP conducted on November 20, 2024, under moderate wind conditions with a peak wind speed of approximately $9\,\mathrm{m\,s^{-1}}$. As shown, the machine learning model markedly outperforms all conventional methods in estimating wind speed.

Table 4.14 presents the statistical evaluation of horizontal wind prediction accuracy using RMSE, MAE, and standard deviation. The ML-based model not only achieved the lowest RMSE but also demonstrated reduced bias and variability compared to LESO.

Table 4.14: Performance metrics for horizontal wind estimation using CS3D-based input data.

Method	**MAE($\mathrm{m\,s^{-1}}$)**	**RMSE($\mathrm{m\,s^{-1}}$)**	**StdDev($\mathrm{m\,s^{-1}}$)**
ML2 FF-NN	0.4103	0.5462	0.5368
LESO	2.2962	2.3151	0.2952

4.7.1 Comparison of CS3D and CSWX with Machine Learning in Horizontal Wind Estimation

A comparative analysis was conducted between the CS3D and CSWX platforms for horizontal wind estimation using machine learning models. To ensure the validity of this comparison, it was essential to select flight profiles with minimal temporal offset, thereby maintaining consistency in environmental conditions and ensuring the reliability of WINDoe-interpolated reference data.

A case from November 20, 2024, was selected in which the time lag between the two flights was approximately 77 seconds. This minimal offset allowed WINDoe to generate consistent wind profiles for both platforms, enabling a fair and accurate cross-platform evaluation.

Figure 4.13 illustrates the horizontal wind estimation results for CS3D and CSWX using both machine learning and LESO algorithms. The CSWX platform exhibited superior performance, likely due to its improved aerodynamic configuration and enhanced flight stability, which contribute to more accurate wind retrieval.

Table 4.15 presents the quantitative evaluation using MAE, RMSE, and standard deviation (StdDev). Both the machine learning model and LESO algorithm produced more accurate estimates on the CSWX platform, reaffirming its design advantages over the legacy CS3D system.

Table 4.15: Performance comparison between CS3D and CSWX for horizontal wind estimation using machine learning and LESO.

Method	**MAE($\mathbf{m\,s^{-1}}$)**	**RMSE($\mathbf{m\,s^{-1}}$)**	**StdDev($\mathbf{m\,s^{-1}}$)**
ML2 FF-NN CS3D	0.6173	0.7949	0.7347
ML2 FF-NN CSWX	0.7056	0.8985	0.7736
LESO_CS3D	1.0125	1.1703	0.6182
LESO_CSWX	0.5887	0.7617	0.7550

Figure 4.14 presents a horizontal wind profile obtained on August 7, 2024, during which a radiosonde was launched concurrently with the DWL. The comparison highlights that both the machine learning-based model and the LESO algorithm, when applied to the CS3D platform, provided wind estimates that closely aligned with the radiosonde and lidar measurements. In contrast, the CSWX platform exhibited higher

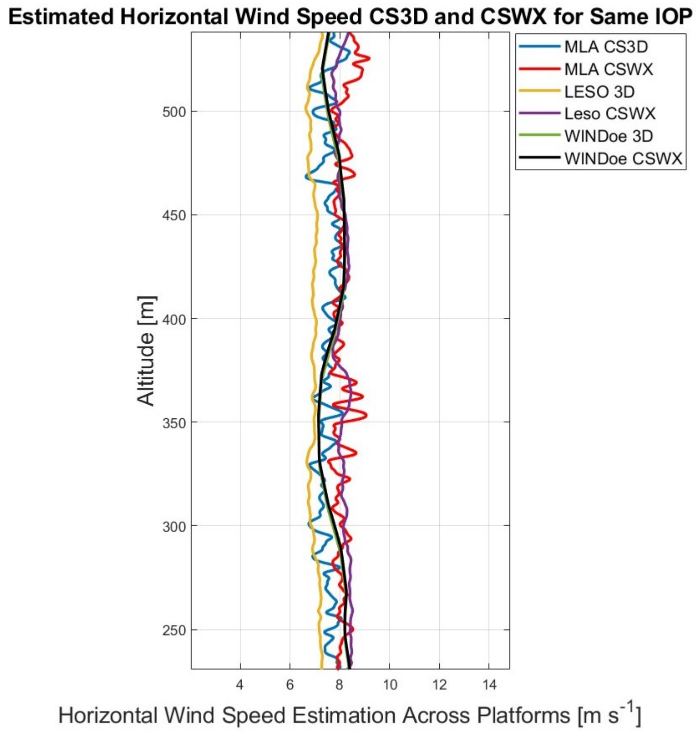

Figure 4.13: Comparison of horizontal wind estimation between CS3D and CSWX using machine learning and LESO algorithms on November 20, 2024. The CSWX platform demonstrates improved performance, attributed to its aerodynamic enhancements and optimized sensor integration.

estimation errors on this particular day, which is likely attributable to the complexity of local atmospheric conditions affecting model accuracy.

4.8 Extension to 3D Wind Vector Estimation

Following the development of machine learning models for horizontal wind estimation and their performance comparison against existing wind estimation algorithms, it was consistently observed that Model 2 (FF-NN) and Model 4 (DANN) outperformed the other two machine learning models. In particular, Model 4 (DANN) demonstrated superior performance under high wind conditions and in data-scarce environments due

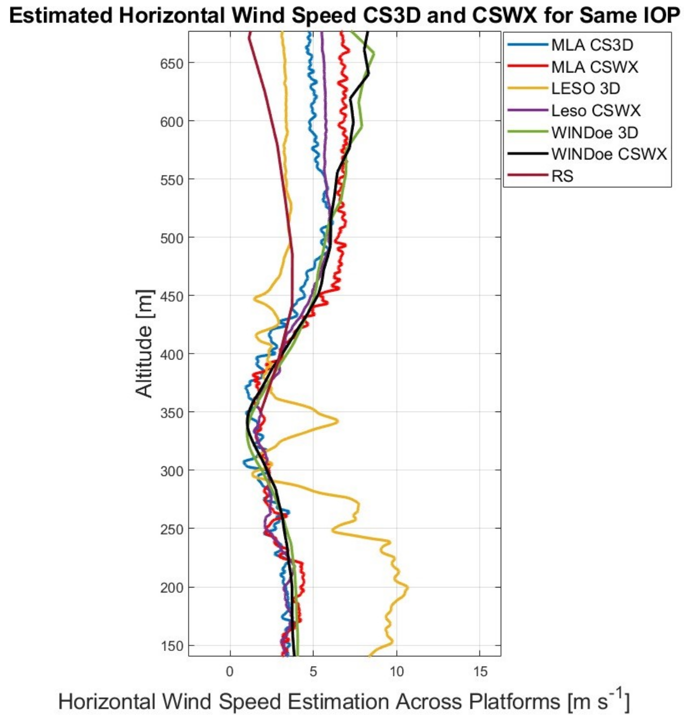

Figure 4.14: Horizontal wind estimation comparison for CS3D and CSWX on August 7, 2024, validated against DWL and radiosonde observations. The CS3D machine learning model showed strong agreement with reference data, while CSWX experienced higher variability, likely due to complex atmospheric conditions.

to its domain adaptation capability.

However, restricting wind estimation studies solely to horizontal wind is insufficient for comprehensive atmospheric studies or practical UAS-based sensing applications. Accurate vertical wind estimation is equally important, especially for tasks such as turbulence characterization, boundary layer analysis, and numerical weather prediction.

Conventionally, vertical wind estimation requires the deployment of an additional DWL to directly measure the vertical wind component. In this research, the availability of multiple DWLs for field deployment posed a significant operational challenge due to resource limitations. Therefore, an alternative solution was necessary.

Due to the unavailability of a second DWL during field operations, acquiring verti-

Table 4.16: Performance metrics for horizontal wind estimation on August 7, 2024, including MAE, RMSE, standard deviation (StdDev), and correlation with reference data.

Method	**MAE($\mathbf{m\,s^{-1}}$)**	**RMSE($\mathbf{m\,s^{-1}}$)**	**StdDev($\mathbf{m\,s^{-1}}$)**
ML FF-NN CS3D	2.2801	2.8071	2.5793
ML FF-NN CSWX	2.3960	2.9306	2.9360
LESO_CS3D	2.7855	3.3567	3.4567
LESO_CSWX	2.2182	2.7211	2.6355

cal wind profiles for model training presented a significant challenge. To address this limitation, WINDoe [67] was employed as a post-processing framework capable of reconstructing both horizontal and vertical wind components. By integrating data from radiosondes, 3D Mesonet towers, DWLs , and UAS observations, WINDoe [67] generated comprehensive reference wind fields. This integration enabled the development and evaluation of machine learning models for 3D wind estimation without the need for an additional lidar system. Using WINDoe as the ground truth, two new neural network models were developed to predict three-dimensional (3D) wind fields encompassing both horizontal and vertical wind components — solely from the dynamics and kinematics of the CopterSonde CSWX platform. These models enabled real-time 3D wind estimation without relying on dedicated wind sensors, using only the telemetry and flight data recorded onboard the UAS.

This advancement demonstrated the potential of machine learning-based models to perform 3D wind estimation from UAS platforms, significantly enhancing their operational utility for future atmospheric sensing missions and boundary-layer research.

4.8.1 Model 1: Feed-Forward ReLU Network for 3D Wind Estimation

Model 1 (3D FF-NN) for three-dimensional wind estimation was developed to simultaneously predict both horizontal and vertical wind components using only onboard measurements from the CopterSonde CSWX platform. The model was trained in a supervised learning setting using WINDoe-retrieved wind profiles as the ground truth, eliminating the need for multiple DWLs during field deployments.

The architecture consists of two dedicated but jointly deployed neural sub-networks, which work in conjunction to estimate horizontal and vertical wind together:

- $\mathcal{M}_h$: A horizontal wind sub-network for estimating the horizontal wind speed component.
- $\mathcal{M}_v$: A vertical wind sub-network for estimating the vertical wind speed component.

Both sub-networks share the same 7-dimensional input vector, which includes dynamic and kinematic parameters derived from the CSWX drone's onboard sensors. The horizontal wind model $\mathcal{M}_h$ comprises five hidden layers with neuron dimensions [64, 32, 16, 8, 1], while the vertical wind model $\mathcal{M}_v$ consists of a shallower four-layer architecture: [32, 16, 8, 1]. All hidden layers utilize the ReLU activation function, while the output layers employ a linear activation appropriate for continuous-valued regression outputs.

Target outputs for horizontal and vertical wind components were normalized separately using z-score normalization based on their respective training distributions. The models were trained independently using the Adam optimizer with a small learning rate and weight decay to ensure stable convergence. A Reduce-on-Plateau scheduler was employed to decrease the learning rate by a factor of 0.5 if validation loss plateaued for 10 consecutive epochs. Early stopping was activated to prevent overfitting.

Both $\mathcal{M}_h$ and $\mathcal{M}_v$ were optimized using the MSE cost function. After training, the models were exported in MATLAB-compatible '.mat' format, allowing them to be jointly loaded for real-time 3D wind prediction tasks. Table 4.17 shows the architecture of the Model 1 (3D FF-NN), which is developed to estimate horizontal and vertical wind simultaneously.

Table 4.17: Configuration of 3D Wind Estimation Model 1 (3D FF-NN) with independent horizontal and vertical sub-networks trained on WINDoe wind profiles.

Attribute	**Details**
Input Features	7-dimensional CopterSonde CSWX dynamics and kinematics
Outputs	Horizontal and vertical wind speeds (referenced to WINDoe)
Model Structure	Two independently trained sub-networks deployed jointly
Hidden Layers (Horizontal)	[64, 32, 16, 8, 1]
Hidden Layers (Vertical)	[32, 16, 8, 1]
Activation Function	ReLU (hidden layers), Linear (output layer)
cost function	MSE
Optimizer (Horizontal)	Adam (learning rate = 0.0038977053188954313, weight decay = 3.265140046530239e-06)
Optimizer (Vertical)	Adam (learning rate =0.016966116172149402, weight decay = 1.2245854568144616e-05)
Scheduler	Reduce-on-Plateau (patience = 10, reduction factor = 0.5)
Early Stopping	Enabled based on validation loss trend
Normalization	Z-score using training statistics (separate for u and w components)
Platform	Python + GPU (NVIDIA A100)
Computing Time	8 hours and 4 hours Optuna search
Deployment Format	MATLAB-compatible `.mat` file for real-time inference

4.8.2 Model 2: Domain-Adversarial Neural Network for 3D Wind Estimation

Model 2 (3D DANN) was developed to address the problem of domain shift between training and deployment conditions, particularly when the majority of training data are concentrated in low-wind environments. This imbalance poses a risk that a model trained exclusively under such conditions may generalize poorly to high-wind scenarios encountered during real-world deployment. To mitigate this limitation, a DANN framework was implemented for three-dimensional (3D) wind estimation similarly to the horizontal wind estimation model of the previous section.

The model was trained to simultaneously predict horizontal and vertical wind components using WINDoe-retrieved wind profiles as ground truth in the source domain. A separate, unlabeled dataset was designated as the target domain, simulating future deployment conditions. Unlike conventional supervised training, DANN exploits both labeled source domain data and unlabeled target domain data to learn domain-invariant representations through adversarial learning.

The architecture consists of three primary components:

- A **feature extractor** that processes the 7-dimensional input vector of drone-derived kinematic and dynamic features into a shared latent representation.
- A **label predictor** that maps the extracted features to two continuous outputs: horizontal and vertical wind speeds.
- A **domain classifier** that attempts to classify whether the extracted features originate from the source or target domain, while the feature extractor is trained adversarially to confuse it.

To enable adversarial learning, a GRL was inserted between the feature extractor and the domain classifier. During training, the model minimizes the regression loss (MSE) on wind predictions from the source domain while maximizing domain confusion through the domain classifier using a cross-entropy cost function. This joint optimization encourages the feature extractor to produce domain-invariant representations.

The model was trained using the Adam optimizer with a learning rate of 4.215540073900586e-05. The source dataset was split into 85% for training and 15% for validation. The same split was applied to the target dataset, although only the validation subset was used for post-training evaluation, as the target domain remains unlabeled during training. Table 4.18 shows the architecture of the DANN model developed to estimate vertical and horizontal wind.

Table 4.18: Configuration of 3D Wind Estimation Model 2 (3D DANN) for learning domain-invariant features.

Component	**Architecture / Details**
Input Features	7-dimensional CSWX drone dynamics and kinematics
Feature Extractor	[128, 64, 32] neurons, ReLU activation
Label Predictor	[16, 8, 2] neurons, ReLU activations; linear output for 2D wind components
Domain Classifier	Gradient Reversal Layer + [16, 2] neurons (binary domain classification)
cost functions	MSE(regression), Cross-Entropy (domain classification)
Optimizer	Adam (learning rate = 1×10^{-4})
Training Strategy	Joint optimization on labeled source and unlabeled target domains
Normalization	Z-score normalization using source training statistics
Platform	Python + GPU (NVIDA A100)
Computing Time	2 hours and 8 hours Optuna search
Deployment Format	MATLAB-compatible `.mat` file

The trained DANN model demonstrated strong generalization in data-scarce regions and under high-wind conditions. Its ability to adapt to new domains without requiring labeled target data makes it particularly well-suited for operational UAS-based wind profiling across diverse environmental regimes.

4.8.3 Performance Analysis of the 3D Wind Estimation Models

The WINDoe dataset used for training the 3D models primarily consisted of low-to-moderate wind profiles. Given this distribution, it was anticipated that Model 1 (3D FF-NN)—a dual-regression feedforward neural network trained directly on WINDoe-labeled data—would perform more effectively under such conditions compared to Model 2 (3D DANN), which incorporates domain adaptation to address distributional shifts. However, from the analysis, it turned out the Model 2 (3D DANN) performed better in 3D wind estimation when averaging profiles from several flights.

Evaluation of the test profiles collected on 20 November 2024 confirmed this expectation. For horizontal wind estimation, Model 1 (3D FF-NN) slightly outperformed the LESO algorithm. More notably, in vertical wind estimation, both Model 1 (3D FF-NN) and Model 2 (3D DANN) exhibited superior accuracy relative to LESO. These machine learning models demonstrated the ability to resolve finer-scale wind features, particularly in the vertical direction, which the LESO framework failed to capture.

Figure 4.15 illustrates the comparative performance of the models and LESO, showing both horizontal and vertical wind profile predictions. It is evident that Model 1 (3D FF-NN) tracks the WINDoe-derived ground truth more closely, especially in representing vertical wind gradients.

The corresponding statistical evaluation is provided in Tables 4.19 and 4.20, with metrics including MAE, RMSE, and standard deviation of error. In horizontal wind estimation (Table 4.19), Model 2 (3D DANN) slightly outperformed LESO across all metrics. In vertical wind estimation (Table 4.20), both machine learning models sig-

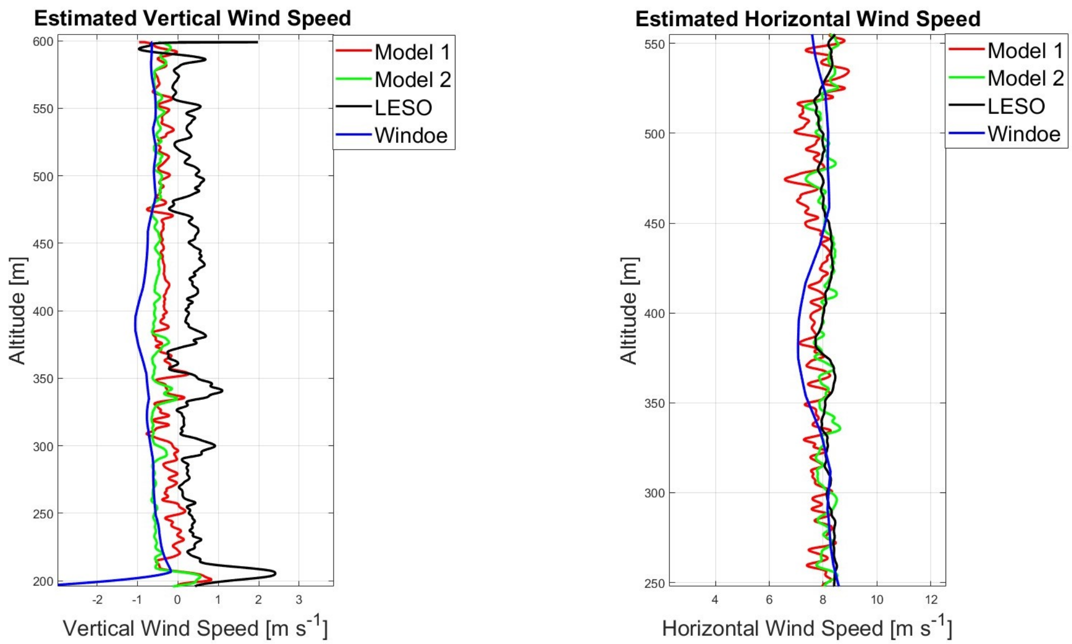

Figure 4.15: Horizontal and vertical wind profile predictions on 20 November 2024 using both machine learning models and the LESO algorithm. Model 1 (3D FF-NN) exhibited superior agreement with WINDoe profiles, particularly in vertical wind estimation.

nificantly reduced estimation error compared to LESO, reinforcing their potential for improved 3D wind reconstruction in UAS-based sensing applications.

Table 4.19: Performance comparison of horizontal wind speed estimation on 20 November 2024.

Method	**MAE($\mathbf{m\,s^{-1}}$)**	**RMSE($\mathbf{m\,s^{-1}}$)**	**StdDev($\mathbf{m\,s^{-1}}$)**
ML1 3D FF-NN	0.7188	0.9271	0.9173
ML2 3D DANN	0.6479	0.8346	0.7664
LESO	0.6580	0.8673	0.8574

To enable a robust statistical assessment, five test profiles from different days were selected—most of which featured low to moderate wind conditions. For each model, the average RMSE, MAE, and standard deviation (StdDev) were computed across these days to evaluate generalization performance.

Table 4.20: Performance comparison of vertical wind speed estimation on 20 November 2024.

Method	**MAE($\mathbf{m\,s^{-1}}$)**	**RMSE($\mathbf{m\,s^{-1}}$)**	**StdDev($\mathbf{m\,s^{-1}}$)**
ML1 3D FF-NN	0.9150	1.7582	1.5698
ML2 3D DANN	0.7637	1.6700	1.5570
LESO	1.5442	2.2416	1.6635

Tables 4.21 and 4.22 summarize the average performance metrics for horizontal and vertical wind estimation, respectively, comparing Model 1 (3D FF-NN), Model 2 (3D DANN), and the LESO algorithm. As shown in Table 4.21, Model 1 (3D FF-NN) achieved horizontal wind prediction accuracy that closely matched LESO, while Model 2 (3D DANN) outperformed the LESO. This degradation in Model 1 (3D FF-NN)'s performance may be attributed to mild overfitting or increased sensitivity to specific test profiles

In contrast, vertical wind estimation—typically a more challenging task due to the lack of reliable direct measurements—was where both machine learning models significantly outperformed LESO. Table 4.22 indicates that both Model 1 (3D FF-NN) and Model 2 (3D DANN) produced lower RMSE and MAE values, with Model 1 (3D FF-NN) demonstrating the most accurate and stable performance.

Figures 4.16 and 4.17 visually illustrate these results. The bar plots reaffirm that Model 2 (3D DANN) consistently delivers reliable predictions in both horizontal and vertical wind components, while Model 1 (3D FF-NN) remains effective but is marginally less accurate.

Table 4.21: Average horizontal wind estimation performance across five test days.

Model	MAE($\mathrm{m\,s^{-1}}$)	RMSE($\mathrm{m\,s^{-1}}$)	StdDev($\mathrm{m\,s^{-1}}$)
ML1 3D FF-NN	1.1213	1.4935	1.3760
ML2 3D DANN	1.0492	1.2818	1.1399
LESO	1.0297	1.2799	1.2149

Table 4.22: Average vertical wind estimation performance across five test days.

Model	MAE($\mathrm{m\,s^{-1}}$)	RMSE($\mathrm{m\,s^{-1}}$)	StdDev($\mathrm{m\,s^{-1}}$)
ML1 3D FF-NN	0.5893	0.8604	0.7176
ML2 3D DANN	0.5440	0.8192	0.6897
LESO	1.0397	1.3503	1.0141

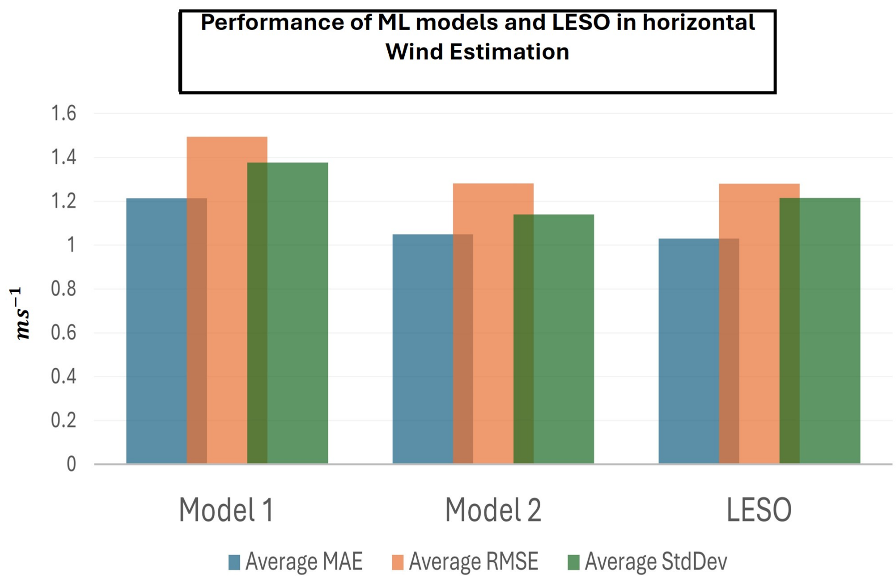

Figure 4.16: Average RMSE, MAE, and standard deviation for horizontal wind estimation across five test days. Model 2 (3D DANN) matches LESO in performance, while Model 1 (3D FF-NN) exhibits slightly higher prediction error.

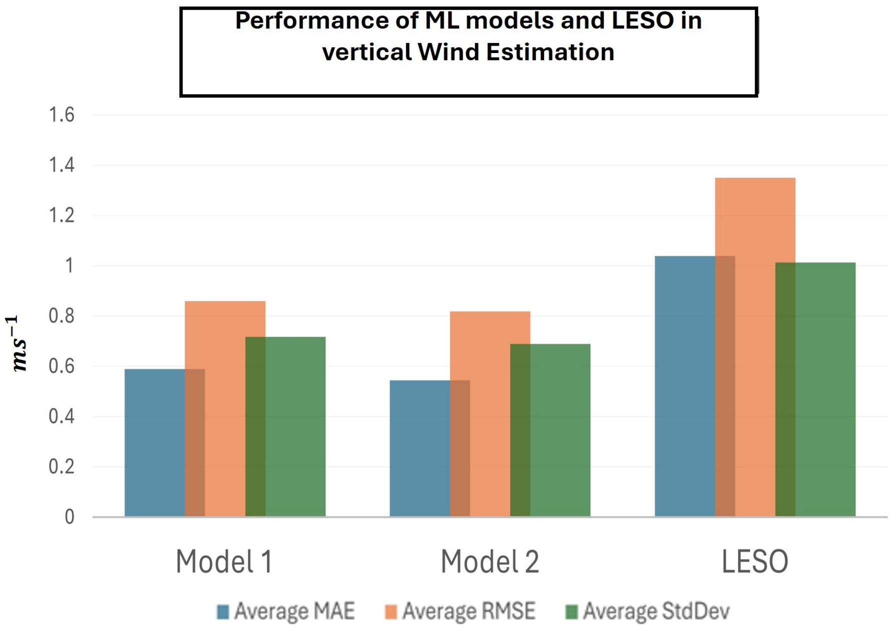

Figure 4.17: Average RMSE, MAE, and standard deviation for vertical wind estimation across five test days. Model 1 (3D FF-NN) outperforms both Model 2 (3D DANN) and LESO, demonstrating its effectiveness in capturing vertical wind dynamics.

4.9 Performance Evaluation of the Proposed Machine Learning Models

Although the developed machine learning models demonstrated strong performance across low and moderate wind regimes, limitations were identified under high-wind conditions—primarily due to the imbalance in the training dataset. The majority of training profiles featured horizontal wind speeds between 2 and 12 $\mathrm{m\,s^{-1}}$, with significantly fewer data points available for speeds exceeding 12 $\mathrm{m\,s^{-1}}$. This limited exposure to high-wind dynamics hindered the models' ability to generalize effectively in such regimes.

To evaluate model performance under high-wind conditions, two test cases were selected. Figure 4.18 presents the first case from an IOP flown on May 19, 2025, where horizontal wind speeds reached approximately 24 $\mathrm{m\,s^{-1}}$. Due to the absence of DWL on this day, the LESO model was used as a reference. The figure shows that both Model 1 (3D FF-NN) and Model 2 (3D DANN) closely followed the LESO estimates and effectively captured the wind dynamics up to 15 $\mathrm{m\,s^{-1}}$. Although a slight offset was observed beyond 15 $\mathrm{m\,s^{-1}}$, this result is encouraging given that the models were trained predominantly on low-wind data.

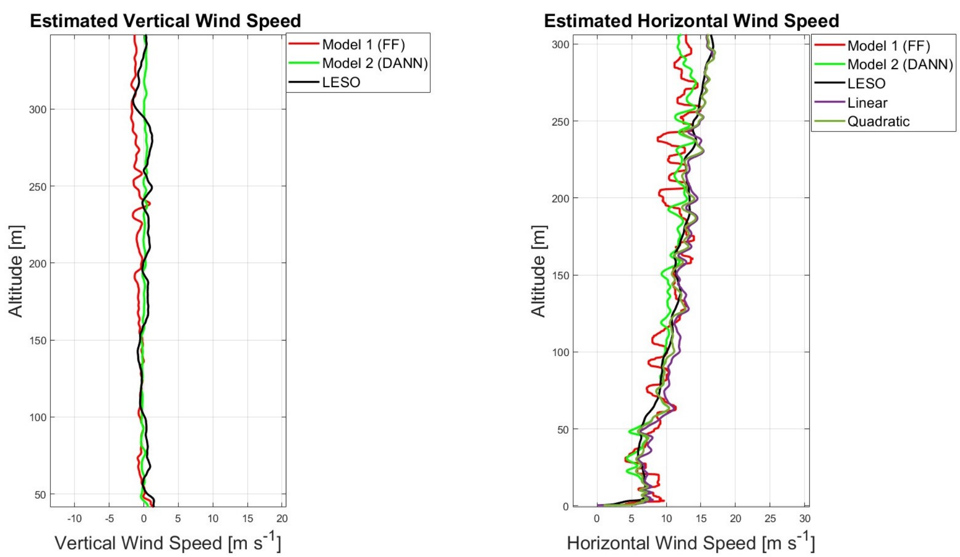

Figure 4.18: Performance of Model 1 (3D FF-NN) and Model 2 on a high-wind case (IOP from May 19, 2025) with a maximum horizontal wind speed of approximately 24 $\mathrm{m\,s^{-1}}$. Despite limited high-wind training data, both models could estimate wind up to 15 $\mathrm{m\,s^{-1}}$.

A second case was analyzed from a fire hazard warning day shown in Figure 4.20, during which CopterSonde flights were restricted to a 250 m altitude ceiling. Wind speeds on this day reached up to 20 $\mathrm{m\,s^{-1}}$; however, both models exhibited performance degradation below 15 $\mathrm{m\,s^{-1}}$—particularly Model 2, which showed a pronounced underestimation. Upon further inspection, it was noted that there were significant variations in atmospheric humidity shown in Figure 4.19, as there was a fire hazard warning on that day, and it is known that fire days are denser, and it contributed to a shifted input distribution compared to the training set. This case underscores the sensitivity of data-driven models to unseen environmental conditions.

To further investigate these limitations, another unseen IOP was evaluated with peak wind speeds approaching 16 $\mathrm{m\,s^{-1}}$. As shown in Figure 4.20, Model 2 (3D DANN) accurately captured wind dynamics up to approximately 12 $\mathrm{m\,s^{-1}}$, but displayed increasing underestimation at higher speeds. This trend reinforces the need for targeted inclusion of high-wind profiles in the training dataset to enhance generalization.

Table 4.23 summarizes the model performance for all test cases with wind speeds exceeding 12 $\mathrm{m\,s^{-1}}$. Both Model 1 (3D FF-NN) and Model 2 exhibited higher RMSE and MAE values relative to LESO, with Model 2 (3D DANN) in particular showing reduced robustness. These results confirm that the limited representation of high-wind scenarios in the training dataset impairs the models' ability to generalize effectively.

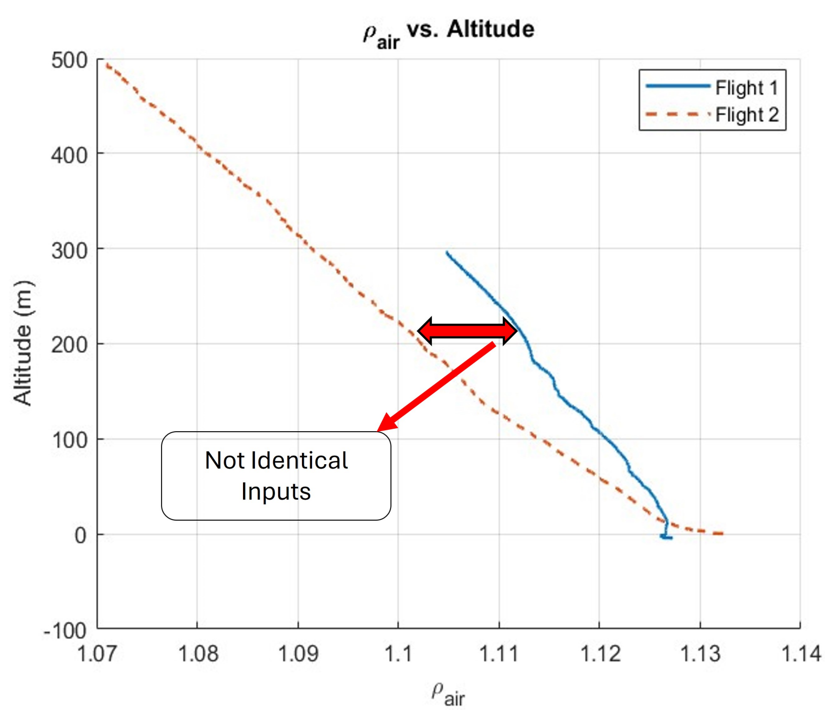

Figure 4.19: Comparison of input feature distributions between two flights: May 19, 2025 (red) and a fire hazard warning day (blue). Substantial differences in humidity that contributed to increased estimation bias in the latter case.

Table 4.23: Performance of the machine learning models and LESO algorithm under high-wind conditions (>12 $\mathrm{m\,s^{-1}}$), highlighting the effect of limited training data.

Method	**MAE($\mathbf{m\,s^{-1}}$)**	**RMSE($\mathbf{m\,s^{-1}}$)**	**StdDev($\mathbf{m\,s^{-1}}$)**
ML1 3D FF-NN	4.1308	5.1615	4.8467
ML2 3D DANN	4.0747	5.2046	3.2575
LESO	3.6297	4.5893	4.0809

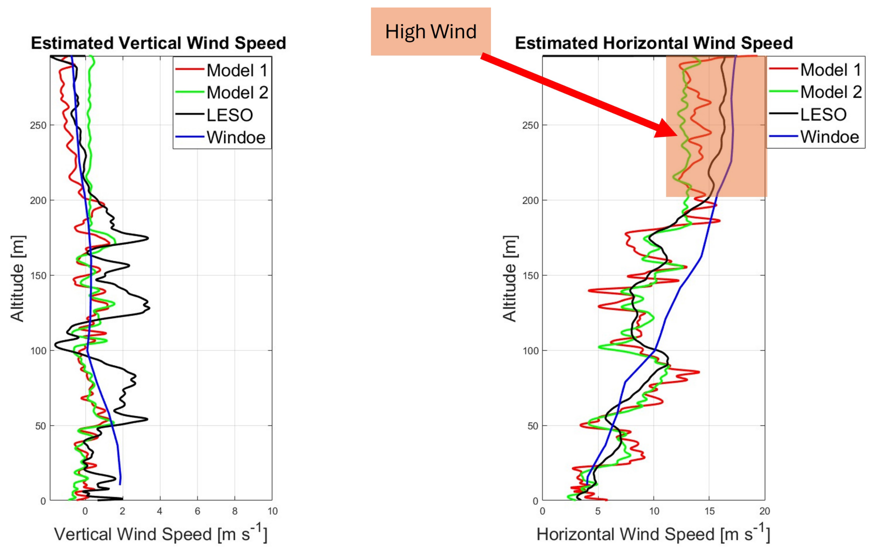

Figure 4.20: Wind estimation for an unseen IOP with peak wind speeds of approximately 16 $\mathrm{m\,s^{-1}}$. Model 2 performs well up to 12 $\mathrm{m\,s^{-1}}$, but increasingly underestimates above this threshold, reflecting the lack of high-wind examples in the training data.

The quality and variety of the training dataset are directly related to how well the machine learning models in this study work. One key limitation is the distribution of wind conditions: the majority of flight profiles captured horizontal wind speeds in the 2–12 $\mathrm{m\,s^{-1}}$ range, while high-wind conditions above 15 $\mathrm{m\,s^{-1}}$ were underrepresented. This imbalance reduced the models' ability to generalize in extreme conditions. Additionally, the vertical wind component had no direct ground truth except from WINDoe reconstructions, which could introduce additional uncertainty.

Other sources of error may include:

- Inherent noise or bias in UAS sensor data (e.g., RPM drift),
- Environmental factors such as turbulence or rapid weather transitions not captured in the training data,
- Temporal mismatch or lag between reference instruments (e.g., WINDoe or Doppler Lidar) and UAS measurements during data collection,
- Limitations of the flight envelope, including altitude constraints during windy conditions.

In summary, while the developed machine learning models demonstrated high fidelity in low-to-moderate wind regimes, their performance under high-wind conditions was constrained by insufficient training data coverage. These results emphasize the importance of dataset representativeness in machine learning applications and suggest that future work should consider augmenting the dataset with targeted high-wind flights and improving synchronization between instruments to reduce observational noise.

4.9.1 Operational Implications and Real-World Feasibility

Each of the machine learning models developed in this study presents specific trade-offs and practical considerations for real-time deployment in field operations. Model 2 (FF-NN) provides a favorable balance between prediction accuracy and computational efficiency, making it suitable for integration with existing uncrewed aircraft system (UAS) flight controllers. Its simplicity supports onboard inference with minimal processing overhead, rendering it a viable choice for real-time applications.

Model 3 (FT-NN) represents a preliminary attempt to incorporate domain adaptation by retraining the model with a limited set of labeled target data. However, this model requires a considerable amount of labeled data and additional architectural refinement. Given its underperformance in practical evaluations, it is not recommended for operational use without further development.

Model 4 (DANN) is designed to address domain shift scenarios, such as seasonal changes or transitions between low- and high-wind environments. Although it is more computationally intensive than FF-NN models, it offers improved generalization in situations where labeled target data are unavailable. This makes it a promising candidate for deployment in regions with poorly characterized meteorological conditions.

For three-dimensional (3D) wind estimation, both Model 1 (3D FF-NN) and Model 2 (3D DANN) demonstrated strong performance, particularly under low-to-moderate wind conditions. Notably, both models outperformed the LESO algorithm in vertical wind estimation. Either model could be utilized for real-time applications; however, Model 1 is better suited for immediate deployment due to its regression-based architecture and stable performance. Model 2, once trained on a more diverse dataset that includes high-wind scenarios, may offer enhanced generalization via its domain adaptation capability.

In summary, the findings of this study confirm that data-driven models can significantly improve wind estimation accuracy for small UAS platforms, particularly under real-world conditions where conventional aerodynamic assumptions or sensor limitations restrict the utility of traditional methods.

4.10 Chapter Summary

This chapter presented the development, implementation, and evaluation of machine learning-based wind estimation models using data collected from the CopterSonde UAS platform. Four distinct neural network architectures were proposed and assessed:

(1) a baseline model trained using the Levenberg–Marquardt algorithm, (2) a ReLU-activated feedforward neural network trained with the Adam optimizer, (3) a fine-tuned variant incorporating domain adaptation, and (4) a DANN explicitly designed to address domain shift.

The initial phase of the study employed DWL as the reference for horizontal wind estimation. Comparative analyses with existing model-based algorithms—LESO, Linear, and Quadratic—revealed that Models 2 and 4 (FF-NN and DANN, respectively) consistently outperformed traditional methods, particularly under high-horizontal wind estimation and in data-sparse regimes. RMSE-based evaluations confirmed their enhanced generalization capability.

To extend the framework to three-dimensional wind estimation, WINDoe—a multi-instrument optimal estimation system—was utilized to generate comprehensive wind profiles serving as ground truth. Using these profiles, two models were developed to estimate both horizontal and vertical wind components from CopterSonde CSWX input features. Among them, Model 2 (3D DANN) achieved comparable accuracy to LESO in horizontal wind estimation and significantly outperformed LESO in vertical wind prediction, particularly in capturing fine-scale dynamics. Model 1 (3D FF-NN) also demonstrated competitive performance in 3D estimation, though it exhibited slightly higher variability across different test profiles.

A cross-platform analysis further indicated that the CSWX platform—owing to its improved aerodynamic design and enhanced flight stability—yielded more accurate wind estimates than the legacy CS3D system under matched environmental conditions. Machine learning models trained on CSWX data consistently produced lower RMSE and MAE values when compared to both CS3D-based models and conventional estimation algorithms.

In summary, this chapter demonstrated that machine learning—particularly domainaware architectures such as DANN—can serve as robust alternatives to traditional wind estimation algorithms for UAS-based sensing. These models achieved high estimation accuracy without relying on mechanical wind sensors or predefined aerodynamic assumptions, thereby enhancing the flexibility and operational performance of the CopterSonde platforms.

A primary advantage of the developed models lies in their use of real-time onboard telemetry. These inputs, comprising dynamic and environmental parameters that are directly influenced by wind disturbances, enable the models to learn representations that reflect the underlying physical interactions between the platform and the atmosphere. Unlike conventional algorithms, which typically depend on simplifying assumptions, the machine learning approach remains assumption-free, allowing for accurate data-driven estimation across a broader spectrum of flight conditions.

However, the analysis also identified a performance limitation under high-wind regimes, primarily due to the scarcity of training data for wind speeds exceeding 15 $\mathrm{m\,s^{-1}}$. While the models performed reliably under low to moderate wind conditions, systematic underestimation and increased estimation error were observed in high wind. To address this gap, future work should focus on collecting additional high-wind training profiles.

Chapter 5

Conclusions

5.1 CopterSonde Accuracy Based on the World Meteorological Organization Standards

A standardized reference for the validation of any meteorological measurement method or instrument were established by the World Meteorological Organization (WMO) [86]. These thresholds are strong considerations in stating the reliability of wind estimation models and potentially produce a positive impact when assimilated into numerical weather models. In this study, a bin-wise RMSE analysis was conducted for 5 vertical test profiles from several moderately windy days, with the results compared against WMO performance categories. WINDoe was assumed as the valid ground truth for the measurement purpose. Table 5.1 outlines the WMO criteria regarding wind observations.

Based on a relatively reasonable dataset size collected during this study, wind speed

Table 5.1: WMO's categorization of horizontal wind speed accuracy criteria.

Category	**RMSE ($\mathrm{m\,s^{-1}}$)**
Goal	≤ 1
Breakthrough	≤ 2
Threshold	≤ 8

Table 5.2: Averaged RMSE within wind-speed bin ranges comparing the proposed ML models with respect to WMO criteria.

Bin ($\mathbf{m\,s^{-1}}$)	**Model 1 (3D FF-NN)**		**Model 2 (3D DANN)**	
	RMSE	WMO Category	RMSE	WMO Category
0–2	0.85	Goal	0.64	Goal
2–4	0.66	Goal	1.23	Breakthrough
4–6	1.58	Breakthrough	1.22	Breakthrough
6–8	1.20	Breakthrough	1.23	Breakthrough
8–10	1.66	Breakthrough	2.26	Breakthrough
10–12	1.59	Breakthrough	1.84	Breakthrough

bins of 2 $\mathrm{m\,s^{-1}}$ were defined, extending up to 12 $\mathrm{m\,s^{-1}}$ as most of the valid measurements were done in this range based on the presented comparisons from Chapter 4. Table 5.2 presents the bin-wise RMSE for the two proposed ML-based horizontal wind estimation models, along with their classification according to the WMO standards [86].

Similarly, the instrument accuracy criteria for vertical wind estimation, as defined by the WMO [102], was also adopted in this study. Table 5.3 presents the WMO performance criteria for vertical wind estimation, which serve as standardized benchmarks for instrument validation. These criteria are particularly stringent and are meant for flux observations, with the goal threshold requiring uncertainties below 1 $\mathrm{cm\,s^{-1}}$. However, CopterSonde is meant for convective-scale observations, and achieving this level of precision remains difficult due to platform-scale turbulence, sensor noise, and flight dynamics.

In this study, vertical wind uncertainties were evaluated over broader spatial scales to reflect practical deployment conditions. Under this framework, the developed machine learning models achieved mean uncertainties ranging between 1 and 2 $\mathrm{m\,s^{-1}}$ which is suitable for updrafts and downdrafts produced in convective environments.

Table 5.3: WMO uncertainty criteria for vertical wind estimation.

Criterion	Goal	Breakthrough	Threshold
RMSE ($\mathrm{cm\,s^{-1}}$)	≤ 1	≤ 2	≤ 5

5.2 Conclusion

This thesis investigated the development and evaluation of machine learning–based wind estimation models for UAS, with a focus on the CopterSonde platforms CS3D and CSWX. Traditional wind estimation techniques—including the Linear, Quadratic, and LESO algorithms—were initially evaluated and finely calibrated using DEO and historical flight logs from past field experiments. Comparative analyses indicated that while the LESO algorithm remained the most robust among the traditional baseline methods, its accuracy diminished under varying wind conditions and complex flight environments. Furthermore, LESO is based on simplifying aerodynamic assumptions that may not be valid across all operational scenarios, thereby imposing additional limitations on its general applicability.

To address these limitations, four neural network architectures were developed and evaluated for horizontal wind estimation: (1) a Levenberg–Marquardt optimized baseline model (Model 1, LM-NN), (2) a ReLU-activated feedforward neural network (Model 2, FF-NN), (3) a fine-tuned variant incorporating limited domain adaptation (Model 3, FT-NN), and (4) a Domain Adversarial Neural Network (Model 4, DANN). All models were initially trained using Doppler Wind Lidar (DWL) data as the reference for supervised learning.

Among these, Model 2 (FF-NN) and Model 4 (DANN) consistently outperformed the LESO algorithm, particularly in high-wind regimes and under conditions with limited training data. RMSE-based evaluations confirmed their superior generalization performance, with Model 2 (FF-NN) showing optimal results in low-to-moderate wind scenarios, while Model 4 (DANN) exhibited strong robustness in the presence of domain shift.

Building on these findings, the two best-performing models were extended to estimate three-dimensional (3D) wind fields: (1) Model 1 (3D FF-NN) and (2) Model 2 (3D DANN). The WINDoe system was utilized as the reference framework for both training and evaluation, enabling accurate prediction of both horizontal and vertical wind components under real-world atmospheric conditions.

5.2.1 Key Findings

This subsection summarizes the key findings from the development, evaluation, and deployment of the machine learning-based wind estimation models. It highlights their practical applicability across different wind regimes and limitations observed during training, offering a concise guide for future real-world implementation and further research.

- Although the LESO algorithm is based on specific aerodynamic assumptions, the machine learning models operate solely through data-driven learning without such assumptions. Despite this distinction, the machine learning models achieved performance levels comparable to LESO in horizontal wind estimation. This finding suggests that the assumptions embedded in LESO do not significantly impair its horizontal wind prediction capabilities.

- For vertical wind speed estimation, machine learning models substantially outperformed LESO, demonstrating their superior capacity to model complex, multi-dimensional wind dynamics. This result highlights the potential of machine learning for more accurate 3D wind estimation in atmospheric sensing applications.

- Based on the findings of this research, practical recommendations can be proposed for future field deployments. If the goal is to estimate the horizontal wind only, assuming that there is no effect from the vertical wind, Model 4 (DANN) should be applied because of its ability to perform domain adaptation from training in the source domain to estimate the wind in the target domain shifted.

- In low-wind conditions, both Model 1 (FF-NN 3D) and Model 2 (DANN 3D) demonstrated comparable performance, particularly in estimating vertical wind more accurately than the existing LESO algorithm. However, Model 2 (DANN 3D) is preferred due to its training on a larger dataset spanning low to moderate wind regimes and its built-in domain adaptation capability, which enhances generalization when field conditions differ from those encountered during training. In contrast, for high-wind scenarios, Model 1 (FF-NN 3D) is recommended, as its regression-based architecture enables more consistent wind estimation. Model 2 (DANN 3D), being a domain-adversarial neural network, exhibited performance degradation in high-wind regimes—likely due to limited high-wind data during training—which may cause deviation from true wind conditions under such extremes.

- Comparisons across platforms revealed that the CSWX system, owing to its enhanced aerodynamic design and flight stability, consistently produced more accurate wind estimates than the CS3D. Machine learning models trained on CSWX data yielded lower RMSE and MAE values, further highlighting the importance of hardware optimization in improving wind estimation outcomes.

- A key challenge faced in this study was the limited number of available DWL systems, which restricted the resolution of high-quality training data, especially in high wind regimes. Collecting adequate datasets for machine learning development was resource-intensive due to the complexity of deploying and coordinating UAS and ground-based DWLs operations when weather was favorable for experiments. Nonetheless, the developed machine learning models still achieved performance within the "breakthrough" category as defined by WMO standards, demonstrating their strong potential. It is expected that with access to larger and more diverse datasets—encompassing a wider range of meteorological conditions—these models could better capture the UAS/wind dynamics which may lead to greater measurement accuracy.

In summary, this thesis demonstrates that machine learning, particularly with the integration of domain adaptation techniques, offers a viable and accurate alternative to conventional wind estimation algorithms. These models eliminate dependency on predefined aerodynamic assumptions and can operate effectively without wind sensors as payload, paving the way for more flexible and scalable atmospheric sensing using UAS platforms. Future work should focus on expanding the dataset to include more varied weather conditions, exploring real-time deployment capabilities, and integrating these models into operational forecasting and decision-support systems.

5.3 Future Work

This thesis demonstrated that machine learning techniques provide a robust alternative to conventional UAS-based wind estimation algorithms. These data-driven models eliminate reliance on predefined aerodynamic assumptions and dedicated wind sensors, thereby offering a flexible and scalable solution for any uncrewed aircraft systems of the multirotor type.

Future work should focus on the development of wind estimation models that are resilient to sparse calibration datasets and capable of supporting rapid deployment workflows, allowing for timely UAS operations following fabrication and system testing. While the acquisition of additional high-wind and complex-flow datasets remains advantageous for improving model generalization, the primary objective should be to develop methodologies that preserve high predictive accuracy under limited-data conditions. Toward this goal, techniques such as transfer learning and attention-based networks could be investigated to enable reliable wind estimation with minimal calibration overhead. These approaches could facilitate model adaptation across different platforms, sensor configurations, or environmental regimes with minimal additional data collection.

Additionally, pathways for real-time deployment should be explored by embedding trained models within onboard or ground-based processing systems, with particular emphasis on minimizing latency, managing computational load, and ensuring operational robustness.

Furthermore, the evaluation framework developed in this study was designed to be compatible with the real-time wind estimation from the Graphical User Interface (GUI) developed at the CIWRO. This GUI facilitates the transfer of UAS telemetry data to a computer program that displays live-data to the user for real-time data verification. This ensures that the presented ML methods remain consistent with ongoing efforts to operationalize weather-sensing UAS and provide reliable data to weather forecast systems.

Finally, future model iterations should incorporate additional features extracted from UAS flight logs. One promising direction involves leveraging the discrepancy between commanded and actual UAS trajectories—induced by wind disturbances—as an indirect measure of ambient wind forcing. Incorporating such control deviation metrics may improve model sensitivity to wind effects and enhance estimation accuracy across a broader range of flight scenarios

References

[1] W. Thielicke, W. Hübert, U. Müller, M. Eggert, and P. Wilhelm, "Towards accurate and practical drone-based wind measurements with an ultrasonic anemometer," *Atmospheric Measurement Techniques*, vol. 14, pp. 1303–1318, Feb. 2021. DOI: 10.5194/amt-14-1303-2021.

[2] J. Lauer and M. Fengler, "Meteodrones-meteorological planetary boundary layer measurements by vertical drone soundings," *EGU General Assembly Conference Abstracts*, 2017.

[3] R. Wagner, I. Antoniou, S. M. Pedersen, M. S. Courtney, and H. E. Jørgensen, "The influence of the wind speed profile on wind turbine performance measurements," *Wind Energy*, vol. 12, no. 4, pp. 348–362, 2009. DOI: https://doi.org/10.1002/we.297.

[4] V.-M. Kumer, J. Reuder, B. Svardal, C. Sætre, and P. Eecen, "Characterisation of single wind turbine wakes with static and scanning wintwex-w lidar data," *Energy Procedia*, vol. 80, pp. 245–254, 2015, 12th Deep Sea Offshore Wind R&D Conference, EERA DeepWind'2015. DOI: https://doi.org/10.1016/j.egypro.2015.11.428.

[5] G. V. Iungo, "Experimental characterization of wind turbine wakes: Wind tunnel tests and wind lidar measurements," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 149, pp. 35–39, 2016. DOI: https://doi.org/10.1016/j.jweia.2015.11.009.

[6] L. Li, L. Gao, Y. Liu, Y. Cui, and B. Wang, "Field measurements of atmospheric boundary layer and the impact of its daily variation on wind turbine wakes," in *5th IET International Conference on Renewable Power Generation (RPG) 2016*, 2016, pp. 1–6.

[7] J. Labovský and L. Jelemenský, "Verification of cfd pollution dispersion modelling based on experimental data," *Journal of Loss Prevention in the Process Industries*, vol. 24, no. 2, pp. 166–177, 2011. DOI: https://doi.org/10.1016/j.jlp.2010.12.005.

[8] R. Barthelmie *et al.*, "3d wind and turbulence characteristics of the atmospheric boundary layer," English (US), *Bulletin of the American Meteorological Society*, vol. 95, no. 5, pp. 743–756, May 2014. DOI: 10.1175/BAMS-D-12-00111.1.

[9] M. D. Ivey, R. Petty, D. Desilets, J. Verlinde, and R. G. Ellingson, in *Polar Research with Unmanned Aircraft and Tethered Balloons*, 2014.

[10] J. Elston, "Overview of small fixed-wing unmanned aircraft for meteorological sampling," *Journal of Atmospheric and Oceanic Technology*, vol. 32, pp. 97–115, Jan. 2015. DOI: 10.1175/JTECH-D-13-00236.1.

[11] S. Prudden, A. Fisher, M. Marino, A. Mohamed, S. Watkins, and G. Wild, "Measuring wind with small unmanned aircraft systems," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 176, pp. 197–210, 2018. DOI: https://doi.org/10.1016/j.jweia.2018.03.029.

[12] A. Rautenberg, M. S. Graf, N. Wildmann, A. Platis, and J. Bange, "Reviewing wind measurement approaches for fixed-wing unmanned aircraft," *Atmosphere*, vol. 9, no. 11, 2018. DOI: 10.3390/atmos9110422.

[13] L. Barbieri *et al.*, "Intercomparison of small unmanned aircraft system (suas) measurements for atmospheric science during the lapse-rate campaign," *Sensors*, vol. 19, no. 9, 2019. DOI: 10.3390/s19092179.

[14] T. W. Nichols, B. Argrow, and D. B. Kingston, "Error sensitivity analysis of small uas wind-sensing systems," in *AIAA Information Systems-AIAA Infotech @ Aerospace*. DOI: 10.2514/6.2017-0647.

[15] N. Wildmann, M. Hofsäß, F. Weimer, A. Joos, and J. Bange, "Masc & dash; a small remotely piloted aircraft (rpa) for wind energy research," *Advances in Science and Research*, vol. 11, no. 1, pp. 55–61, 2014. DOI: 10.5194/asr-11-55-2014.

[16] A. van den Kroonenberg, T. Martin, M. Buschmann, J. Bange, and P. Vörsmann, "Measuring the wind vector using the autonomous mini aerial vehicle m2av," *Journal of Atmospheric and Oceanic Technology*, vol. 25, pp. 1969–1982, 2008.

[17] J. O. Pinto *et al.*, "The status and future of small uncrewed aircraft systems (UAS) in operational meteorology," *Bulletin of the American Meteorological Society*, vol. 102, no. 11, E2121–E2136, 2021. DOI: 10.1175/BAMS-D-20-0138.1.

[18] D. W. Camp, R. E. Turner, and L. P. Gilchrist, "Response tests of cup, vane, and propeller wind sensors," *Journal of Geophysical Research (1896-1977)*, vol. 75, no. 27, pp. 5265–5270, 1970. DOI: https://doi.org/10.1029/JC075i027p05265.

[19] N. Hutchins, J. Monty, M. Hultmark, and A. Smits, "A direct measure of the frequency response of hot-wire anemometers: Temporal resolution issues in wall-bounded turbulence," *Experiments in Fluids*, vol. 56, Jan. 2015. DOI: 10.1007/s00348-014-1856-8.

[20] L. Barbieri *et al.*, "Intercomparison of small unmanned aircraft system (sUAS) measurements for atmospheric science during the LAPSE-RATE campaign," *Sensors*, vol. 19, no. 9, 2019. DOI: 10.3390/s19092179.

[21] V. A. Natalie and J. D. Jacob, "Experimental observations of the boundary layer in varying topography with unmanned aircraft," in *AIAA Aviation 2019 Forum*. DOI: 10.2514/6.2019-3404.

[22] P. Neumann, M. Bartholmai, J. H. Schiller, B. Wiggerich, and M. Manolov, "Micro-drone for the characterization and self-optimizing search of hazardous gaseous substance sources: A new approach to determine wind speed and direction," in *2010 IEEE International Workshop on Robotic and Sensors Environments*, 2010, pp. 1–6. DOI: 10.1109/ROSE.2010.5675265.

[23] T. A. Johansen, A. Cristofaro, K. Sørensen, J. M. Hansen, and T. I. Fossen, "On estimation of wind velocity, angle-of-attack and sideslip angle of small uavs using standard sensors," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015, pp. 510–519. DOI: 10.1109/ICUAS.2015.7152330.

[24] X. Xiang, Z. Wang, Z. Mo, G. Chen, K. Pham, and E. Blasch, "Wind field estimation through autonomous quadcopter avionics," in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, 2016, pp. 1–6. DOI: 10.1109/DASC.2016.7778071.

[25] G. W. Donnell, J. A. Feight, N. Lannan, and J. D. Jacob, "Wind characterization using onboard imu of suas," in *2018 Atmospheric Flight Mechanics Conference*. DOI: 10.2514/6.2018-2986.

[26] A. R. Segales *et al.*, "The coptersonde: An insight into the development of a smart unmanned aircraft system for atmospheric boundary layer research," *Atmospheric Measurement Techniques*, vol. 13, no. 5, pp. 2833–2848, 2020. DOI: 10.5194/amt-13-2833-2020.

[27] C. M. Leme Beu and E. Landulfo, "Machine-learning-based estimate of the wind speed over complex terrain using the long short-term memory (lstm) recurrent neural network," *Wind Energy Science*, vol. 9, no. 6, pp. 1431–1450, 2024. DOI: 10.5194/wes-9-1431-2024.

[28] Y. Song and Y. Wang, "Global wildfire outlook forecast with neural networks," *Remote Sensing*, vol. 12, no. 14, 2020. DOI: 10.3390/rs12142246.

[29] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019. DOI: 10.1162/neco_a_01199.

[30] Y.-K. Wu and J.-S. Hong, "A literature review of wind forecasting technology in the world," in *2007 IEEE Lausanne Power Tech*, 2007, pp. 504–509. DOI: 10.1109/PCT.2007.4538368.

[31] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2624–2661, 2016. DOI: 10.1109/COMST.2016.2560343.

[32] H. Shakhatreh *et al.*, "Unmanned aerial vehicles (uavs): A survey on civil applications and key research challenges," *IEEE Access*, vol. 7, pp. 48 572–48 634, 2019. DOI: 10.1109/ACCESS.2019.2909530.

[33] J. Lazo, M. Lawson, P. Larsen, and D. Waldman, "U.s. economic sensitivity to weather variability," *Bulletin of the American Meteorological Society*, vol. 92, pp. 709–720, Jun. 2011. DOI: 10.1175/2011BAMS2928.1.

[34] I. Goklany, "Deaths and death rates from extreme weather events: 1900-2008," *Journal of American Physicians and Surgeons*, vol. 14, pp. 102–109, Jan. 2009.

[35] P. Webster, "Meteorology: Improve weather forecasts for the developing world," *Nature*, vol. 493, pp. 17–9, Jan. 2013. DOI: 10.1038/493017a.

[36] C.Liu, “Severe weather in a warming climate,” *Nature*, pp. 422–423, 2017. DOI: 10.1038/544422a.

[37] D. J. Frame, ““climate change attribution and the economic costs of extreme weather events: A study on damage from extrem rainfall and drought,” *Climatic change*, vol. 162, no. 2, Sep. 2020. DOI: 10.1007/s10584-020-02729-y.

[38] B. R. Scheffers and G. Pecl, “Persecuting, protecting or ignoring biodiversity under climate change,” *Nature Climate Change*, vol. 9, no. 8, pp. 581–586, Aug. 2019. DOI: 10.1038/s41558-019-0526-5.

[39] S. Curtis, A. Fair, J. Wistow, D. V. Val, and K. Oven, “Impact of extreme weather events and climate change for health and social care systems,” *Environmental Health*, vol. 16, no. 1, p. 128, Dec. 2017. DOI: 10.1186/s12940-017-0324-3.

[40] M. M. Q. Mirza, “Climate change and extreme weather events: Can developing countries adapt?” *Climate Policy*, vol. 3, no. 3, pp. 233–248, Jan. 2003. DOI: 10.3763/cpol.2003.0330.

[41] B. Geerts *et al.*, “Community workshop on developing requirements for in situ and remote sensing capabilities in convective and turbulent environments (C-RITE),” UCAR/NCAR Earth Observing Laboratory, Tech. Rep., 2017.

[42] G. de Boer *et al.*, “Advancing unmanned aerial capabilities for atmospheric research,” English, *Bulletin of the American Meteorological Society*, vol. 100, no. 3, ES105–ES108, Mar. 2019. DOI: 10.1175/BAMS-D-18-0254.1.

[43] NWS. “Building a weather-ready nation. 2019-2022 stratigic plan.” (2019), [Online]. Available: https://www.weather.gov/media/wrn/NWS_Weather-Ready-Nation_Strategic_Plan_2019-2022.pdf.

[44] National Severe Storm Laboratory (NSSL). “MRMS: Multiple Radar/Multiple Sensor.” en. (2015), [Online]. Available: https://www.nssl.noaa.gov/news/factsheets/MRMS_2015.March.16.pdf (visited on 11/08/2022).

[45] J. Zhang *et al.*, “Multi-Radar Multi-Sensor (MRMS) quantitative precipitation estimation: Initial operating capabilities,” *Bulletin of the American Meteorological Society*, vol. 97, no. 4, pp. 621–638, 2016. DOI: 10.1175/BAMS-D-14-00174.1.

[46] S. E. Koch *et al.*, “On the use of unmanned aircraft for sampling mesoscale phenomena in the preconvective boundary layer,” *J. Atmos. Ocean. Tech.*, vol. 35, pp. 2265–2288, 2018. DOI: 10.1175/JTECH-D-18-0101.1.

[47] S. Kral *et al.*, “Innovative strategies for observations in the arctic atmospheric boundary layer (ISOBAR)—the Hailuoto 2017 campaign,” *Atmosphere*, vol. 9, no. 7, p. 268, Jul. 2018. DOI: 10.3390/atmos9070268.

[48] J. D. Jacob, P. B. Chilson, A. L. Houston, and S. W. Smith, “Considerations for atmospheric measurements with small unmanned aircraft systems,” *Atmosphere*, vol. 9, no. 7, 2018. DOI: 10.3390/atmos9070252.

[49] N. Wildmann, M. Hofsäß, F. Weimer, A. Joos, and J. Bange, “MASC – a small remotely piloted aircraft (RPA) for wind energy research,” *Adv. Sci. Res.*, vol. 11, no. 1, pp. 55–61, 2014. DOI: 10.5194/asr-11-55-2014.

[50] J. Reuder, P. Brisset, M. Jonassen, M. Müller, and S. Mayer, "The small unmanned meteorological observer SUMO: A new tool for atmospheric boundary layer research," *Meteorol. Z.*, vol. 18, no. 2, pp. 141–147, 2009.

[51] P. B. Chilson *et al.*, "Moving towards a network of autonomous UAS atmospheric profiling stations for observations in the Earth's lower atmosphere: The 3D mesonet concept," *Sensors*, vol. 19, no. 12, 2019. DOI: 10.3390/s19122720.

[52] G. de Boer *et al.*, "Development of community, capabilities, and understanding through unmanned aircraft-based atmospheric research: The LAPSE-RATE campaign," *Bulletin of the American Meteorological Society*, vol. 101, no. 5, E684–E699, 2020. DOI: 10.1175/BAMS-D-19-0050.1.

[53] G. M. McFarquhar *et al.*, "Current and future uses of UAS for improved forecasts/warnings and scientific studies," *Bulletin of the American Meteorological Society*, vol. 101, no. 8, E1322–E1328, 2020. DOI: 10.1175/BAMS-D-20-0015.1.

[54] T. G. Konrad, M. L. Hill, J. H. Meyer, and J. R. Rowland, "A small, radio-controlled aircraft as a platform for meteorological sensors," *Applied Physics Lab Technical Digest*, vol. 10, pp. 11–19, 1970.

[55] S. Mayer, A. Sandvik, M. Jonassen, and J. Reuder, "Atmospheric profiling with the uas sumo: A new perspective for the evaluation of fine-scale atmospheric models," *Meteorology and Atmospheric Physics*, vol. 116, pp. 15–26, Apr. 2012. DOI: 10.1007/s00703-010-0063-2.

[56] M. Fengler, "How Meteodrones contribute to weather forecasting," Meteomatics AG, St. Gallen, Switzerland, Tech. Rep., Mar. 2020, p. 13.

[57] J. J. Cione *et al.*, "Eye of the storm: Observing hurricanes with a small unmanned aircraft system," *Bulletin of the American Meteorological Society*, vol. 101, no. 2, E186–E205, 2020. DOI: 10.1175/BAMS-D-19-0169.1.

[58] J. C. Naftel, "Nasa global hawk: A new tool for earth science research," NASA Dryden Flight Research Center, , Edwards, CA 93523, USA, Tech. Rep., 2009.

[59] P. Newman and D. Fahey, "Nasa global hawk uas and its application for atmospheric science," Apr. 2009. DOI: 10.2514/6.2009-2007.

[60] P. B. Chilson *et al.*, "Unmanned aerial system for sampling atmospheric data," University of Oklahoma, Tech. Rep. US12084181B2, Sep. 2024, Patent No. US12084181B2, Available at: https://patents.google.com/patent/US12084181B2/en.

[61] S. Allison, H. Bai, and B. Jayaraman, "Wind estimation using quadcopter motion: A machine learning approach," *Aerospace Science and Technology*, vol. 98, p. 105 699, 2020. DOI: https://doi.org/10.1016/j.ast.2020.105699.

[62] T. J. Wagner, P. M. Klein, and D. D. Turner, "A new generation of ground-based mobile platforms for active and passive profiling of the boundary layer," *Bulletin of the American Meteorological Society*, vol. 100, no. 1, pp. 137–154, 2019. DOI: 10.1175/BAMS-D-17-0165.1.

[63] A. R. Segales, "Design and implementation of a novel multicopter unmanned aircraft system for quantitative studies of the atmosphere," Ph.D. Dissertation, Ph.D. dissertation, University of Oklahoma, 2022.

[64] B. R. Greene, A. R. Segales, S. Waugh, S. Duthoit, and P. B. Chilson, "Considerations for temperature sensor placement on rotary-wing unmanned aircraft systems," *Atmos. Meas. Tech.*, vol. 11, no. 10, pp. 5519–5530, 2018. DOI: 10.5194/amt-11-5519-2018.

[65] A. R. Segales *et al.*, "Coptersonde: Uas-based vertical profiler design enhancements," Manuscript submitted to Atmospheric Measurement Techniques(in review), Norman, Oklahoma, May, 2025.

[66] J. Tang, K. P. Jain, and M. W. Mueller, "Quartm: A quadcopter with unactuated rotor tilting mechanism capable of faster, more agile, and more efficient flight," *Frontiers in Robotics and AI*, vol. Volume 9 - 2022, 2022. DOI: 10.3389/frobt.2022.1033715.

[67] J. G. Gebauer and T. M. Bell, "A flexible, multi-instrument optimal estimation retrieval for wind profiles," *Journal of Atmospheric and Oceanic Technology*, vol. 41, no. 6, pp. 605–620, 2024. DOI: 10.1175/JTECH-D-23-0134.1.

[68] P. P. Neumann and M. Bartholmai, "Real-time wind estimation on a micro unmanned aerial vehicle using its inertial measurement unit," *Sensors and Actuators A: Physical*, vol. 235, pp. 300–310, 2015. DOI: https://doi.org/10.1016/j.sna.2015.09.036.

[69] R. T. Palomaki, N. T. Rose, M. van den Bossche, T. J. Sherman, and S. F. J. D. Wekker, "Wind estimation in the lower atmosphere using multirotor aircraft," *Journal of Atmospheric and Oceanic Technology*, vol. 34, no. 5, pp. 1183–1191, 2017. DOI: 10.1175/JTECH-D-16-0177.1.

[70] T. M. Bell, A. R. Segales, J. G. Gebauer, and E. N. Smith, "Considerations for wxuas wind speed estimation," in *Proceedings of the AMS*, Affiliations: a Cooperative Institute for Severe and High-Impact Weather Research and Operations, Norman, OK; b University of Oklahoma, Norman, OK; c NOAA/OAR/-National Severe Storms Laboratory, Norman, OK; d Advanced Radar Research Center, Norman, OK; e School of Meteorology, University of Oklahoma, Norman, OK, Norman, Oklahoma.

[71] M. F. Ahmad, N. A. M. Isa, W. H. Lim, and K. M. Ang, "Differential evolution: A recent review based on state-of-the-art works," *Alexandria Engineering Journal*, vol. 61, no. 5, pp. 3831–3872, 2022. DOI: https://doi.org/10.1016/j.aej.2021.09.013.

[72] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011. DOI: 10.1109/TEVC.2010.2059031.

[73] T. Wetz and N. Wildmann, "Spatially distributed and simultaneous wind measurements with a fleet of small quadrotor uas," *Journal of Physics: Conference Series*, vol. 2265, no. 2, p. 022 086, May 2022. DOI: 10.1088/1742-6596/2265/2/022086.

[74] J. Han, "From pid to active disturbance rejection control," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 3, pp. 900–906, 2009. DOI: 10.1109/TIE.2008.2011621.

[75] G. Herbst, "A simulative study on active disturbance rejection control (adrc) as a control tool for practitioners," *Electronics*, vol. 2, no. 3, pp. 246–279, 2013. DOI: 10.3390/electronics2030246.

[76] "Robustness and multivariable frequency-domain techniques," in *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*. John Wiley & Sons, Ltd, 2015, ch. 6, pp. 500–583. DOI: https://doi.org/10.1002/9781119174882.ch6.

[77] Z. Gao, "Scaling and bandwidth-parameterization based controller tuning," in *Proceedings of the 2003 American Control Conference, 2003.*, vol. 6, 2003, pp. 4989–4996. DOI: 10.1109/ACC.2003.1242516.

[78] J. González-Rocha, S. F. J. De Wekker, S. D. Ross, and C. A. Woolsey, "Wind profiling in the lower atmosphere from wind-induced perturbations to multirotor uas," *Sensors*, vol. 20, no. 5, 2020. DOI: 10.3390/s20051341.

[79] B. R. Greene, S. T. Kral, P. B. Chilson, *et al.*, "Gradient-based turbulence estimates from multicopter profiles in the arctic stable boundary layer," *Boundary-Layer Meteorology*, vol. 183, pp. 321–353, 2022. DOI: 10.1007/s10546-022-00693-x.

[80] R. Storn and K. Price, “Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997. DOI: 10.1023/A:1008202821328.

[81] J. Kim and K.-K. K. Kim, “Dynamic programming for scalable just-in-time economic dispatch with non-convex constraints and anytime participation,” *International Journal of Electrical Power & Energy Systems*, vol. 123, p. 106 217, 2020. DOI: https://doi.org/10.1016/j.ijepes.2020.106217.

[82] Bilal, M. Pant, H. Zaheer, L. Garcia-Hernandez, and A. Abraham, “Differential evolution: A review of more than two decades of research,” *Engineering Applications of Artificial Intelligence*, vol. 90, p. 103 479, 2020. DOI: https://doi.org/10.1016/j.engappai.2020.103479.

[83] S. Das, S. S. Mullick, and P. Suganthan, “Recent advances in differential evolution – an updated survey,” *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016. DOI: https://doi.org/10.1016/j.swevo.2016.01.004.

[84] L. Jebaraj, C. Venkatesan, I. Soubache, and C. C. A. Rajan, “Application of differential evolution algorithm in static and dynamic economic or emission dispatch problem: A review,” *Renewable and Sustainable Energy Reviews*, vol. 77, pp. 1206–1220, 2017. DOI: https://doi.org/10.1016/j.rser.2017.03.097.

[85] Bilal, M. Pant, H. Zaheer, L. Garcia-Hernandez, and A. Abraham, “Differential evolution: A review of more than two decades of research,” *Engineering Applications of Artificial Intelligence*, vol. 90, p. 103 479, 2020. DOI: https://doi.org/10.1016/j.engappai.2020.103479.

[86] World Meteorological Organization. "Oscar requirements - view requirement 451." (2024), [Online]. Available: https://space.oscar.wmo.int/requirements/view/451 (visited on 04/16/2024).

[87] R. Gill and R. D'Andrea, "Computationally efficient force and moment models for propellers in UAV forward flight applications," *Drones*, vol. 3, no. 4, 2019. DOI: 10.3390/drones3040077.

[88] ArduPilot Development Team, *ArduPilot: Open Source Autopilot*, Accessed: 2025-06-09, 2025.

[89] I. The MathWorks, *Matlab*, Natick, MA, USA, 2024.

[90] A. F. M. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[91] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, Dec. 2014.

[92] J. Quinn, *Dive into Deep Learning: Tools for Engagement*. Thousand Oaks, California, 2020, p. 551, Archived from the original on January 10, 2023. Retrieved January 10, 2023.

[93] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 59, pp. 1–35, 2016.

[94] F. Mehmood, S. Ahmad, and T. K. Whangbo, "An efficient optimization technique for training deep neural networks," *Mathematics*, vol. 11, no. 6, 2023. DOI: 10.3390/math11061360.

[95] J. J. Moré, “The levenberg-marquardt algorithm: Implementation and theory,” in *Numerical Analysis*, G. A. Watson, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 1978, pp. 105–116.

[96] D. Choi, C. Shallue, Z. Nado, J. Lee, C. Maddison, and G. Dahl, *On empirical comparisons of optimizers for deep learning*, Oct. 2019. DOI: 10.48550/arXiv.1910.05446.

[97] B. Liu, X. Zhang, Z. Gao, and L. Chen, “Weld defect images classification with vgg16-based neural network,” in *Proceedings of the International Forum on Digital TV and Wireless Multimedia Communications*, Shanghai, China: Springer, Nov. 2017, pp. 215–223.

[98] I. Loshchilov and F. Hutter, “Fixing weight decay regularization in adam,” Nov. 2017. DOI: 10.48550/arXiv.1711.05101.

[99] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, “Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit,” *Nature*, vol. 405, no. 6789, pp. 947–951, 2000.

[100] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *Proceedings of the 30th International Conference on Machine Learning*, S. Dasgupta and D. McAllester, Eds., ser. Proceedings of Machine Learning Research, vol. 28, Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 1310–1318.

[101] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine Learning*, vol. 79, no. 1–2, pp. 151–175, May 2010. DOI: 10.1007/s10994-009-5152-4.

[102] World Meteorological Organization. "Oscar variable requirements - wind (vertical component)." (2024), [Online]. Available: https://space.oscar.wmo.int/variables/view/wind_vertical (visited on 04/16/2024).