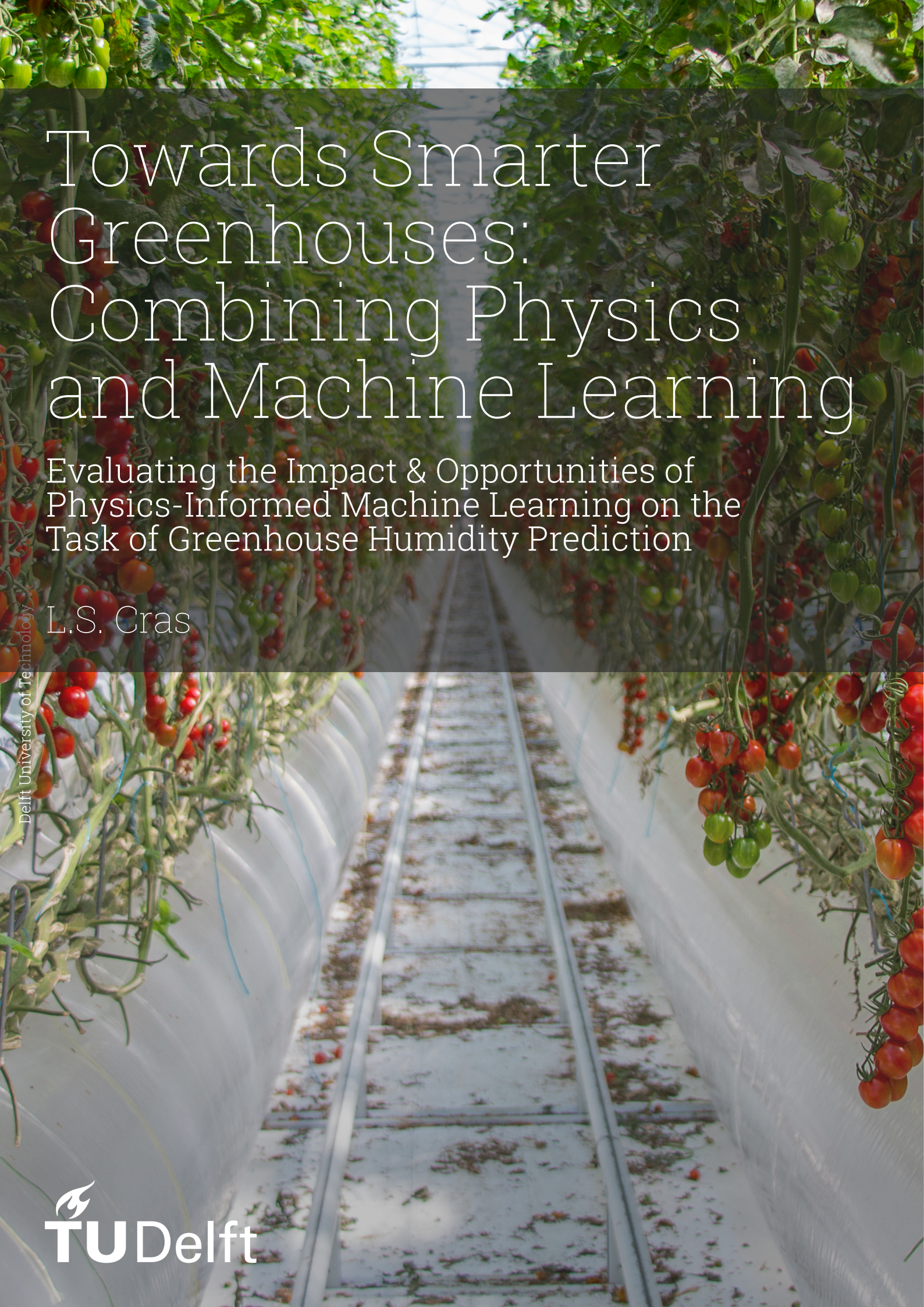




Towards Smarter Greenhouses: Combining Physics and Machine Learning

Evaluating the Impact & Opportunities of
Physics-Informed Machine Learning on the
Task of Greenhouse Humidity Prediction

L.S. Cras

Delft University of Technology

Towards Smarter Greenhouses: Combining Physics and Machine Learning

Evaluating the Impact & Opportunities of
Physics-Informed Machine Learning on the
Task of Greenhouse Humidity Prediction

by

L.S. Cras

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Tuesday July 9, 2024 at 13:00.

Student number: 4660692

Project duration: December 1, 2023 – July 9, 2024

Thesis committee:	dr. N. Yorke-Smith,	TU Delft, supervisor
	prof. dr. ir. M. M. de Weerd, MSc,	TU Delft
	dr. J. Sun,	TU Delft
	dr. ir. B. van Rens,	Blue Radix, daily supervisor

This thesis is confidential and cannot be made public until July 9, 2026.

Cover: Commercially grown tomatoes in a venlo-style greenhouse by
Blue Radix under RF license (Modified)

Style: TU Delft Report Style, with modifications by Daan Zwaneveld

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Preface

The thesis you are reading is the result of months of research and represents the keystone to my master's degree in Computer Science at the Delft University of Technology and the end of my time as a student. I want to extend my deepest gratitude to those who have supported me throughout the course of my thesis. This work would not have been possible without their guidance, assistance, and encouragement.

First and foremost, I wish to express my sincere thanks to my daily supervisor at Blue Radix, *dr. ir. Bram van Rens*, for his continued support and wisdom, and to my supervisor at the Delft University of Technology, *dr. Neil Yorke-Smith*, for his time, advice, and feedback on my work. I would also like to thank *Saheli De* for her help throughout the process and *ir. Dirk van Bolhuis* for helping me wrap my head around the physics involved.

Next, I would like to thank the members of my thesis committee, *prof. dr. ir. Mathijs de Weerd* and *dr. Jing Sun*, for their time and effort in reading and assessing my work.

Furthermore, I want to thank all my colleagues at Blue Radix, who were always ready to answer questions and make the project a joy.

Lastly, I want to thank my family and friends for making my study years the incredible pleasure they have been. I especially want to thank my parents, *Carien & Edwin*, for their unconditional support throughout my life and *Maxime* for her love, patience, and support before, during, and long after this project.

*L.S. Cras
Delft, June 2024*

Abstract

The combination of increasing global food demand with increased food security risks associated with climate change amid a decreasing number of skilled growers necessitates innovative solutions in greenhouse horticulture. Autonomous growing offers a solution based on greenhouse climate forecasting and (optimal) control. However, current theoretical models developed for greenhouse climate forecasting face limitations due to the in-depth physics knowledge required for their use and their dependence on intricate system parameters that are difficult to estimate. Conversely, machine learning models struggle with generalisation to unseen conditions and adherence to physical laws, leading to unrealistic predictions in greenhouse environments. This study addresses these challenges by exploring the use of Physics-Informed Machine Learning (PIML) techniques to enhance greenhouse climate forecasting. A simple differentiable theoretical model for simulating the greenhouse climate is proposed to serve as prior physical knowledge of the system. Subsequently, a novel PIML model is introduced in the form of Controlled Aphynity (CA), which integrates insights from neural controlled differential equations and Aphynity, and is the first such model that allows for the augmentation of incomplete prior knowledge of a dynamical system with data-driven machine learning models *while being adaptable to changing dynamics due to forces acting on the system*. Experimental results show that CA is capable of successfully augmenting incorrect physics under changing dynamics on the task of humidity deficit prediction in the greenhouse. Furthermore, three ensemble methods combining CA with traditional machine learning techniques are explored and demonstrate promising synergies. A detailed case study evaluates CA and the best-performing ensemble approach on the task of humidity deficit prediction under realistic greenhouse scenarios over a complete crop cycle. Both CA and the ensemble methods exhibit superior adherence to physical laws, lower data requirements, and improved performance on outliers compared to conventional machine learning methods. This research contributes to advancing greenhouse climate modelling, underscoring PIML's potential in optimising the greenhouse climate.

Contents

Preface	i
Abstract	ii
1 Introduction	1
1.1 Motivation	1
1.2 Scope	2
1.3 Research Questions & Objectives	3
1.4 Outline & Contributions	3
2 Background and Related Work	5
2.1 The greenhouse	5
2.1.1 Humidity in the greenhouse climate	6
2.1.2 Climate control equipment	7
2.1.3 The greenhouse crop	8
2.2 Theoretical models	9
2.2.1 Overview of theoretical models of greenhouse climate	10
2.2.2 Limitations	11
2.3 Machine learning models	11
2.3.1 Classical methods	12
2.3.2 Deep learning methods	13
2.3.3 Limitations	14
2.4 Physics-informed machine learning models	15
2.4.1 Physics-informed Neural Networks	16
2.4.2 Neural Differential Equations	16
2.4.3 Other Approaches	17
2.5 Discussion	17
3 Theoretical Model for Greenhouse Humidity Deficit Prediction	19
3.1 Model overview	19
3.2 Model implementation	21
3.3 Model validation	21
3.3.1 Data	22
3.3.2 Experimental Setup	22
3.3.3 Determination of model performance	22
3.3.4 Results	23
3.3.5 Discussion	25
3.4 Conclusion	28
4 Controlled Aphynity for Greenhouse Humidity Deficit Prediction	29
4.1 Controlled Aphynity (CA)	29
4.1.1 Implementation	30
4.2 Model validation: Correcting for incomplete physics	31
4.2.1 Data	31
4.2.2 Experimental Setup	31
4.2.3 Determination of model performance	32
4.2.4 A simulation example	32
4.2.5 Results	33
4.2.6 Discussion	34
4.3 Conclusion	35

5	CA-ML Ensemble for Greenhouse Humidity Deficit Prediction	37
5.1	Uniform Average Ensemble	37
5.2	Residual Ensemble	38
5.3	Regression Ensemble	39
5.4	Comparative Analysis	39
5.4.1	Data	39
5.4.2	Experimental setup	39
5.4.3	Determination of model performance	40
5.4.4	Results	40
5.4.5	Discussion	41
5.5	Conclusion	41
6	Case Study: Realistic Greenhouse Scenarios	43
6.1	Data	43
6.2	Experimental set-up	44
6.3	Determination of model performance	45
6.4	Results	46
6.4.1	Overall results on forecasted data	46
6.4.2	Negative humidity deficit prediction rate	46
6.4.3	Data-scarcity	47
6.4.4	Extreme conditions	47
6.5	Discussion	50
6.6	Conclusion	51
7	Conclusion	52
7.1	Summary	52
7.2	Answers to Research Questions	53
7.3	Limitations & Future Work	54
	References	56
A	Supporting formulas for humidity calculations	62
B	Summary of Aphynity (Yin et al., 2021)	63
C	Detailed description of the theoretical greenhouse humidity deficit model	65
D	Derivation of correcting factor for ϵ	69
E	Training data selection procedure	71
F	ML (baseline) model implementation details	72

Introduction

1.1. Motivation

The United Nations (UN) estimate that 691 to 783 million people faced hunger in 2022 (FAO, 2023) and the size of the global population is predicted to grow further to 10 billion by 2050 (DESA, 2022). This underlines the importance of increasing the production of fresh and healthy food. Additionally, climate change increasingly impacts food production worldwide, making food prices more unpredictable and increasing the risk of food insecurity (USGCRP, 2018; van Rijswijk, 2023). Greenhouse cultivation plays a crucial role in providing more and higher-quality fresh food while simultaneously limiting the risk of failed crops and reducing the required area for growing produce. A *greenhouse* is a special structure that is designed to regulate the environment inside, also called its micro-climate. Greenhouses allow for a high crop production density and water-use efficiency per unit of produce (Stanghellini, 2014). Moreover, the controlled micro-climate in the greenhouse provides crops with improved environmental conditions compared to the open field while shielding the crops from extreme weather conditions (Lodovica Gullino et al., 1999; van Rijswijk, 2023).

The greenhouse micro-climate is a complex, non-linear dynamical system characterised by the complex interactions between itself, the outside weather, and the crop. Maintaining an optimal micro-climate for crop growth has historically relied heavily on skilled growers with the knowledge to set the optimal environmental parameters. A grower requires expertise and extensive experience to effectively control crop growth, maximise resource efficiency, and minimise costly errors. However, greenhouse production is increasing worldwide, and the greenhouse industry encounters difficulties finding enough skilled labour to manage crop production (EUROSTAT, 2022; van Rijswijk, 2018). Greenhouses also face challenges related to their energy consumption and carbon footprint. Depending on their surrounding climate, they require significant amounts of energy to regulate the internal micro-climate. Despite incentives to minimise carbon dioxide emissions due to high energy costs, greenhouses still heavily depend on natural gas for heating and electricity to maintain ideal crop growth conditions throughout the year (Kempkes, 2022).

In response to these challenges, there is a rising demand for autonomous growing systems that can effectively manage greenhouse climate and optimise crop production without needing constant human intervention. Advanced autonomous greenhouse control algorithms have emerged as a promising solution but have not yet been widely adopted worldwide for reasons such as lack of transparency, generality, completeness, and continuity (Van Beveren, 2023; van Straten et al., 2000). These autonomous control algorithms typically rely on greenhouse climate models to accurately forecast essential climate factors, such as air temperature and humidity (López-Cruz et al., 2018). These models are increasingly utilised in horticulture (Körner, 2019), and this growing interest has been accompanied by a significant number of published models (Katzin et al., 2022). However, these greenhouse climate models also face issues. Existing greenhouse climate models can be roughly divided into two methodologies: process- or physics-driven models, which we will refer to as “theoretical models” and data-driven machine-learning (ML) models, which we will refer to as “ML models”.

Theoretical models consider the various physical or operational laws operating on the system of interest. By using fundamental laws of physics, such as the law of conservation of energy, the theory can infer hidden information about the system from monitored information and correctly describe how the system might change based on the causal effects between the different components of the system. Theoretical models make it possible to represent the causality between a set of inputs and the corresponding outputs of the system, contrary to the machine learning approach. The expectation is that theoretical models generalise better than ML models to conditions outside the data range on which they have been developed and validated (Katzin et al., 2022). Nevertheless, this approach also has its downsides. Theoretical models rely on intricate prior knowledge about the system of interest. Therefore, creating a satisfactory model depends on an adequate understanding of the system. Additionally, theoretical models often require the calibration of many system parameters, hampering their generality. Moreover, the optimal parameters might be difficult or impossible to estimate or obtain. Finally, some complex theoretical models, such as many computational fluid dynamics models (Zhang et al., 2007), can take longer to simulate than the actual period they are simulating.

Machine learning models take a different approach, aiming to build an empirical model based on statistical correlations found in data. An advantage of this approach is that minimal prior knowledge about the system is required. However, a downside of models based on empirical correlations includes difficulty coping with anomalies and changing circumstances of the system (Kim et al., 2017). Machine learning models are infamous for issues with generalisation and extrapolation. Furthermore, machine learning models' performance heavily depends on the quantity and quality of the available training data from the target system. Both of these are a rarity in many (bio-)physical systems. Moreover, in the context of (bio-)physical systems, any model and its predictions should naturally adhere to the laws of physics to be correct. Data-driven models are oblivious to these laws, and as such, they may make predictions that deviate from physical reality, reducing confidence in the model's correctness.

As a result of these complementary issues, much attention has gone towards combining both approaches in recent years. The emergent field of merging prior physical knowledge with machine learning is commonly referred to as Scientific Machine Learning (SciML) (Shin et al., 2020). Specifically in the context where prior physical knowledge about a system is known, new techniques have emerged to improve the performance of ML models by using this information. These new developments, which fall under the name "Physics-Informed Machine Learning" (PIML), could play an essential role in improving the issues currently experienced when predicting the climate conditions inside the greenhouse. PIML techniques have shown great promise in scientific fields such as chemistry (Anderson et al., 2019), climate science (Beucler et al., 2019; Kashinath et al., 2021), material engineering (D. Liu & Wang, 2019), and biological sciences (L. Liu et al., 2021; Yazdani et al., 2020).

Given the wealth of prior knowledge about the dynamics of the greenhouse micro-climate and the issues experienced with theoretical and ML models, greenhouse climate models can also benefit tremendously from PIML techniques. However, research into using PIML techniques in the context of greenhouse climate models is still minimal. Therefore, this research aims to explore the field of SciML and develop a PIML greenhouse climate model to analyse the benefits, drawbacks, and impacts of such a model in the context of greenhouse horticulture.

1.2. Scope

Although the greenhouse climate generally consists of multiple state variables such as temperature, humidity, and CO₂, the model developed for this research will focus on predicting the humidity levels inside the greenhouse to maintain a viable scope. More specifically, the model will be predicting the humidity deficit (subsection 2.1.1) inside the greenhouse over a predefined length of time (72 hours). Humidity was selected as its dynamics are well-understood in literature and simpler than temperature while being more important and impactful than the CO₂ levels in the greenhouse. Humidity deficit also provides a useful surrogate measure for "un-physical" predictions as negative values are physically infeasible. Furthermore, due to time constraints, the experiments carried out in this research will be confined to a single greenhouse in the Netherlands, where a tomato crop is grown. However, the model developed in this research shall be easy to extend to other climate variables in the future and adaptable to different greenhouses. Next, in most theoretical models of the greenhouse climate, including the one that is provided in chapter 3, information about the state of the climate control equipment is required at

the time of simulation to be able to simulate the HD accurately. For this research, these control inputs are assumed to be known in advance (i.e. at simulation time) for all experiments.

1.3. Research Questions & Objectives

This research aims to explore the potential and analyse the impact of PIML in the context of greenhouse horticulture. To achieve this goal, it aims to explore the field and develop a novel PIML model capable of integrating prior physical knowledge about the greenhouse climate with machine-learning techniques. The developed model can then be experimentally analysed to gain insight into the impact of PIML in greenhouse horticulture. To reach the objective above, this research will be steered by the following main research question:

What is the impact of combining prior physical knowledge of the greenhouse climate and machine learning to forecast the humidity inside a greenhouse, on the general performance, the performance on outliers, the rate of physically infeasible predictions, and the amount of training data required?

To guide the research and help answer the main research question, the following sub-questions are defined:

1. What are the specific limitations of current models in the context of greenhouse climate forecasting, and what are the primary reasons causing these limitations?
2. What prior physical knowledge about the dynamics of the greenhouse climate, specifically humidity, is available and can be used in a PIML model?
3. How can this prior knowledge best be combined with machine learning to form a PIML humidity deficit forecasting model?
4. Compared to a baseline ML model, what implications does the PIML model have for the training procedures, the adherence to physics, the amount of training data required, and the performance generally and on outliers in the form of extreme weather conditions?

The answers to the main research question and the sub-questions can be found in section 7.2.

1.4. Outline & Contributions

This section outlines how the rest of this thesis is structured and mentions the notable contributions made in each part. **Chapter 2** provides the reader with the necessary background to understand the rest of the thesis. Additionally, it disseminates and discusses work that is closely related and important to the contributions made in this thesis. **Chapter 3** presents a differentiable theoretical model that can be used as the prior knowledge in a PIML algorithm for simulating the humidity deficit inside the greenhouse composed of different sources in the related work. The model is implemented and validated to analyse its behaviour, strengths, and weaknesses under varying conditions. **Chapter 4** proposes Controlled Aphynity (CA), a novel PIML model based on the Aphynity framework (Yin et al., 2021) and insights from neural controlled differential equations (Kidger et al., 2020). The newly proposed CA model allows for the augmentation of incomplete prior knowledge of a dynamical system with data-driven machine learning models for complex forecasting tasks *while being adaptable to changing dynamics due to forces acting on the system*. As a practical use-case for analysing the impact of the physics-informed CA model, it was implemented to forecast the humidity deficit inside the greenhouse, allowing for estimating physical parameters and augmenting missing or incorrect physics specifications in the theoretical model. Different settings for correcting incomplete physics and CA's general forecasting performance over a complete crop-growing cycle are experimentally explored and analysed. **Chapter 5** explores the possible synergies of theoretical and machine learning models by testing different methods of using the proposed CA model in an ensemble with a ML model. A comparative experiment among the different methods is performed to analyse each method's strengths and weaknesses and learn more about the underlying interaction between theoretical and ML models. **Chapter 6** compares the proposed CA model, the best-performing ensemble model, and a ML baseline model in a case study with realistic greenhouse scenarios across the following three categories:

1. adherence to laws of physics

2. performance under different amounts of available training data
3. performance for outliers/extreme conditions

Chapter 7 concludes this thesis with a summary, answers to the research questions, and recommendations for future work.

2

Background and Related Work

Understanding the intricate dynamics of greenhouse climates is essential for optimising plant growth and ensuring sustainable agricultural practices. The control and manipulation of humidity within these environments form a critical aspect of this endeavour, necessitating a comprehensive exploration of theoretical and practical approaches. This chapter provides the reader with the necessary background for understanding the rest of this thesis and outlines the related work in the field. The chapter is structured as follows. Section 2.1 begins with a detailed examination of humidity's role in the greenhouse climate. Following this, section 2.2 transitions to a discussion of the theoretical models developed to describe and simulate the humidity under varying outside conditions, providing the reader with the necessary background on the physics and an overview of existing models. Next, the discussion shifts towards machine learning models in section 2.3, exploring their application in predicting time series data through classical methods and progressing into the domain of deep learning techniques. The limitations set out in these sections set the stage for introducing physics-informed machine learning (PIML), an emergent paradigm that seeks to incorporate prior physical knowledge into modern machine learning models. Section 2.4 dissects the mechanics of various innovative approaches in the field of PIML, and section 2.5 finishes the chapter with a discussion of the research gap and the prospects and difficulties related to applying PIML to the context of predicting the humidity inside the greenhouse climate.

2.1. The greenhouse

The greenhouse climate is a complex non-linear dynamical system characterised roughly by two main subsystems: the micro-climate and the crop (López-Cruz et al., 2018). Adding to the complexity, the greenhouse climate is also heavily influenced by outside forces, namely the weather. The indoor climate is commonly described by the following attributes: temperature, humidity, CO₂ concentration, and light. The state of the crop is often characterised by attributes such as the leaf area index (LAI), photosynthetic rate, and biomass. The weather affects the conditions within the greenhouse through factors such as wind, solar radiation, temperature, and humidity. The indoor climate is further influenced by climate control equipment such as windows, different types of curtains, and artificial lighting as a result of the grower trying to steer towards a desired indoor climate. Moreover, the dynamics of the greenhouse climate depend on factors such as the size and geometry of the greenhouse structure, the type of soil, and the amount of insulation. Lastly, the indoor climate affects the growth of the crop, which in turn has a substantial influence on the indoor climate. A schematic overview of the greenhouse system is given in Figure 2.1. Further complicating matters, many different designs of greenhouses exist, and every greenhouse is different in shape, size, and location. The Venlo-style greenhouse is one of the most common greenhouse designs in the Netherlands. Venlo-style greenhouses generally have an aluminium or steel frame and a glass cover, providing a high light transmission (Nemali, 2022). Their distinctive roof shape is characterised by a series of parallel roof sections with slopes (of around 26°) meeting at a central ridge. Ventilation windows are located on the sloped roof sections alternating on each side. An example of a Venlo-style greenhouse can be seen in Figure 2.2. Most greenhouse climate modelling and optimisation research has been performed using Venlo-style greenhouses.

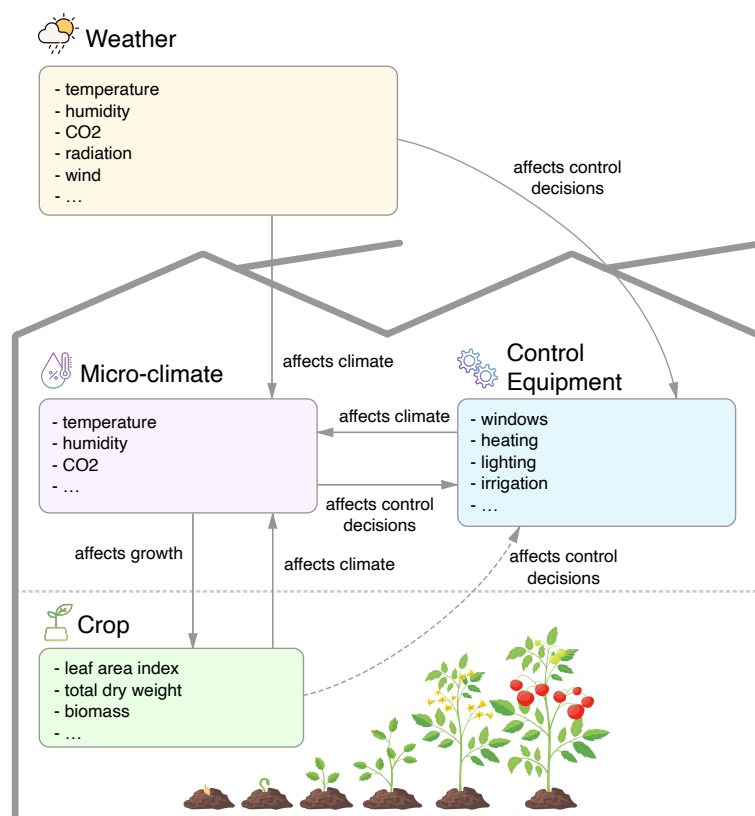


Figure 2.1: A schematic overview of the greenhouse system. Control decisions are based on outside weather, indoor climate, and the crop. The influence of the crop on the control decisions is important but less direct (indicated by the dashed arrow) as it is relatively slowly varying. The outside weather, the control equipment, and the crop influence the indoor climate. The indoor climate subsequently influences the crop. Long-term outputs such as yield, costs, and energy use have been left out as they are not the focus of this study.¹

2.1.1. Humidity in the greenhouse climate

As set out in the scope of this research (section 1.2), this work will focus on the humidity inside the greenhouse. Due to the many different ways in which literature refers to humidity, this subsection will try to explain the most important ones and discuss their interrelationships. Common ways of expressing the humidity inside the greenhouse include absolute humidity, relative humidity, humidity deficit, and vapour pressure. The formulas used in this work for interchanging between these different expressions of humidity can be found in Appendix A. The concepts are briefly explained below:

- **Absolute Humidity (AH)**, also referred to as *vapour concentration*, denotes the amount of water vapour that is present in the air. It is usually expressed in grams per cubic meter of air (g m^{-3}).
- **Relative Humidity (RH)** refers to the amount of water vapour in the air as a fraction of the saturation humidity, the maximum amount of water vapour that the air can contain. RH is calculated by dividing the absolute humidity by the saturation humidity. The result is usually multiplied by 100 to express it as a percentage (%).
- **Humidity Deficit (HD)** refers to the amount of water vapour that can be added to the air before reaching the saturation point. HD is strictly positive, as the water vapour amount can not exceed the saturation point. HD is calculated by subtracting the absolute humidity in the air from the saturation humidity and is subsequently also expressed in grams per cubic meter of air (g m^{-3}).
- **Vapour Pressure (VP)** refers to the pressure exerted by water vapour in a closed container that

¹This figure was created using assets designed by Freepik.com



Figure 2.2: An example of a typical Venlo-style greenhouse.²

is in equilibrium with its liquid water. It indicates the tendency of liquid water, i.e. from a leaf, to evaporate into a gas. Vapour pressure can be obtained using the Clausius–Clapeyron equation (Wark & Richards, 1983) and is usually expressed in Pascals (Pa).

Due to plant transpiration, irrigation, and insulation of the greenhouse structure, the humidity inside the greenhouse tends to be high. Plants exposed to critically high humidity levels may encounter various negative consequences such as a loss of crop quality due to the occurrence and increased severity of fungal diseases and calcium or water deficiencies (Comstock & Ehleringer, 1993; Triguí, 1999). Furthermore, plant growth and development can be disturbed or delayed because of photosynthesis slowing (Mortensen, 2000). High humidity conditions can further hamper plant flowering and pollination in fruit vegetables (Bakker, 1991), and vase life in ornamental plants may be shortened (Mortensen & Gislerød, 2005). On the other hand, low humidity levels can trigger plant water stress. Critically low humidity levels can further lead to a cascade of subsequent impacts on plant health and fruit production, including reduced photosynthesis and growth, higher risks of carbon starvation, and hydraulic failure (Grossiord et al., 2020).

During times of large temperature differences between the greenhouse air and the outside air (i.e. winter in the Netherlands), and accompanying large differences in (absolute) humidity levels, excess water vapour is more easily removed through condensation and ventilation. As a result, humidity generally poses less of an issue during such times (Goldammer, 2019). On the other hand, humidity becomes important for optimal greenhouse control during the warmer months. Because of the smaller temperature differences inside and outside the greenhouse, getting rid of excess humidity becomes more difficult. As a result, the risk for condensation on leaves is higher (especially at night), and thus, the risk of fungal diseases increases (Goldammer, 2019). To counteract this, ventilation can be used more readily to manage the climate inside the greenhouse and allows for steering the humidity inside the greenhouse with much lower energy costs related to heat loss compared to colder weather.

2.1.2. Climate control equipment

Modern greenhouses are often fitted with numerous devices for measuring and manipulating the state of the inside micro-climate. This equipment is generally connected to a climate computer to orchestrate its use in obtaining a desired climate. Most equipment affects the climate in several ways, necessitating an intricate balance of objectives to obtain the desired results.

One influential method for controlling the greenhouse climate is using ventilation windows situated on the roof or side walls of the greenhouse. Properly positioned windows allow for the regulation of temperature and humidity by enabling the exchange of air between the interior and exterior environments. Greenhouses also use heating systems to replace energy lost when outside temperatures are lower than desired. Artificial lighting is crucial in supplementing natural sunlight to sustain plant growth, especially during periods of low amounts of solar radiation. Blackout curtains block out light during specific periods of the day or night, and energy curtains regulate temperature and humidity by trapping air and

²“Westland kassen” by Quistnix is licensed under CC BY 1.0 (source: <https://commons.wikimedia.org/w/index.php?curid=130422>)

heat inside the lower compartment of the greenhouse, reducing the need for heating and potentially saving on cost.

A schematic overview of the greenhouse, the relevant climate control equipment, and the humidity fluxes they influence can be found in Figure 2.3.

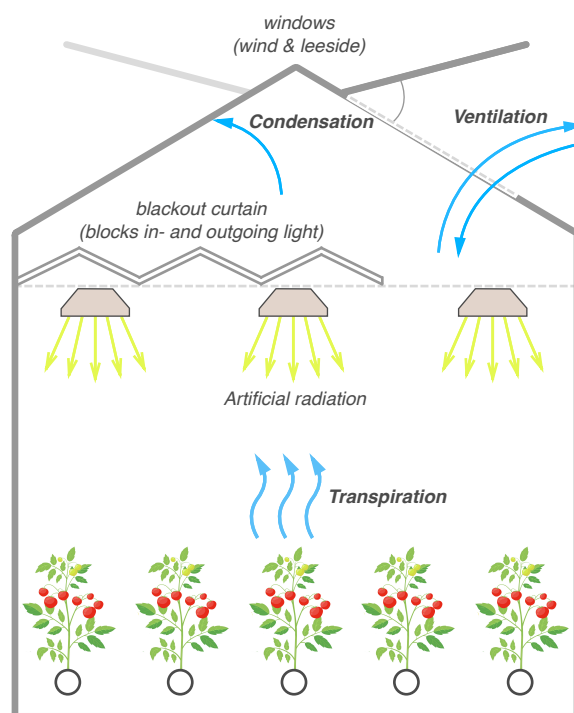


Figure 2.3: A schematic overview of the greenhouse studied in this work. The relevant climate control equipment for the humidity balance is shown, together with the humidity fluxes that are influenced by the equipment (shown in **bold**). Although important in the operation of the greenhouse, irrigation and heating equipment are not shown here as these are not included in the humidity balance considered here.³

2.1.3. The greenhouse crop

The developing crop plays an essential role in the greenhouse. Not only is the crop the driving factor of revenue, but the crop also influences the dynamics of the greenhouse climate and, in turn, impacts climate control decision-making. In this work, crop refers to the commercially grown plant(s) inside the greenhouse.

Crop development encompasses the formation and maturation of various plant organs, such as fruits and flowers. Prior to the development of fruits and flowers, a crop must undergo a stage of vegetative growth to increase its stem and leaf sizes for increased energy production. Although growth and development sometimes require different conditions, we use *growth* to refer to the combination of growth and development in the next part. The most important processes for crop growth are photosynthesis and transpiration (van Weel et al., 2018). Crops use these processes to balance three key areas: internal energy, water, and assimilates (sugars used for growth and internal energy) (van Weel et al., 2018). Photosynthesis, the primary driver of crop growth, utilises light energy to convert water and carbon dioxide (CO₂) into assimilates and oxygen, ultimately contributing to dry matter production in plant organs (van Weel et al., 2018). As specified in section 1.2, this work focuses on the humidity of the greenhouse climate. Therefore, we will focus on the crops' water balance, i.e. the input and output of water from the crop.

The output of water from the crop is mainly driven by transpiration, the process of releasing water vapour through leaves. A schematic overview of this process is shown in Figure 2.4. A crop must evaporate for three key reasons: uptake of nutrients, growth, and cooling. To keep the water balance, the crop

³This figure was created using assets designed by Freepik.com

must uptake water from the roots to equal the amount of water that is consumed by photosynthesis and that evaporates from the leaves. Additionally, the crop might temporarily extract water from any of the fruits, which can act as a buffer in this sense. Through this water uptake, the crops also uptake the necessary nutrients, such as Calcium, to allow for growth.

The crop keeps the water balance in equilibrium through the use of its stomata. Stomata are microscopic pores in the leaves that let water vapour out and can absorb CO_2 from the air. The opening of stomata is mainly driven by increasing water pressure, also called turgor, under the influence of light (van Weel et al., 2018). The stomata will close when the crop faces water stress and depending on CO_2 availability. By opening or closing the stomata, the crop decreases or increases its resistance to evaporate water, which we will refer to as stomatal resistance.

Due to the stomatal mechanisms described above, the amount of transpiration occurring in the greenhouse depends on the number of plants in the greenhouse and the amount of leaf area per plant. The leaf area index (LAI), which is the ratio of leaf area to ground area of the greenhouse in $\text{m}^2 \text{m}^{-2}$, is therefore commonly included in models that simulate the greenhouse climate. However, the LAI is challenging to measure accurately (Fang et al., 2019). As a result, data on the LAI of the crop is often limited or unavailable. This makes predicting the climate inside the greenhouse more difficult.

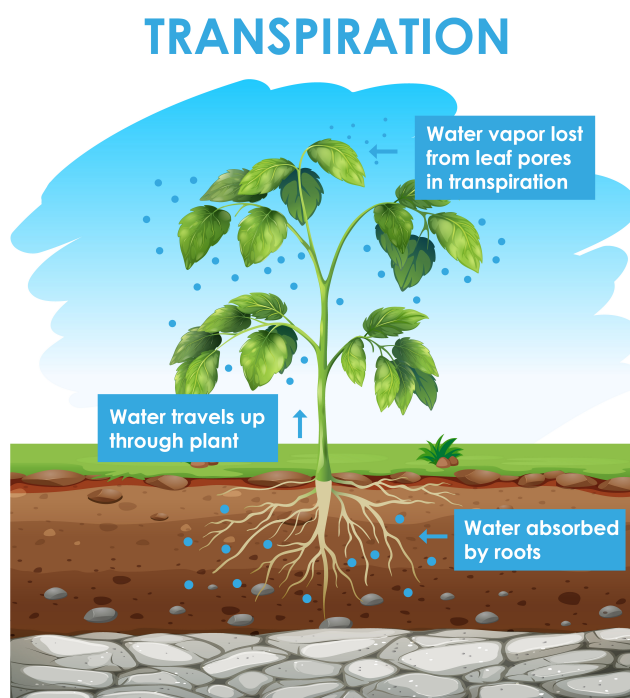


Figure 2.4: Schematic overview of transpiration in a plant. Designed by brgfx / Freepik.

2.2. Theoretical models

This section delves into theoretical models and provides the reader with an introduction to mathematical/theoretical models using differential equations. After this introduction, subsection 2.2.1 provides an overview of existing theoretical/mathematical greenhouse climate models. Lastly, subsection 2.2.2 discusses the limitations of these models.

Theoretical or mathematical models constitute a mathematical representation that describes how a physical system changes over time. It is a framework that involves differential equations or rules that govern the behaviour of the system's variables, illustrating how these variables evolve and interact with each other. The differential equations used to describe systems are typically subdivided into ordinary differential equations (ODE) and partial differential equations (PDE). ODEs are differential equations dependent on a single independent variable, usually time. PDEs are differential equations that depend on partial derivatives over several independent variables, such as time and space. While some models have been developed that describe the greenhouse climate by PDEs (Boulard et al., 2017), the vast majority of existing literature on the subject has modelled the greenhouse climate with systems of ODEs. Another important distinction that should be made about these models is the method by which they have been derived. One such method involves using laws of physics such as energy and mass balances. We refer to these kinds of models as process-based or physics-based. Another common method uses empirical findings to derive simplified relationships between different variables; we refer to these models as empirical models. Because empirical models are no longer directly based on physical laws, they must be calibrated using observational data, which might result in different relationships

under different settings.

Most theoretical models that model the greenhouse climate describe it as a system of ODEs under the 'perfectly stirred tank' assumption; namely, the composition and temperature of air are uniform everywhere in the greenhouse. Due to the exogenous forces on the greenhouse climate, we add a corresponding forcing function, $x(t)$, to the differential equation, which denotes the particular state of these exogenous forces at time t . The task of predicting the future state of the greenhouse can then be considered as solving an initial value problem (IVP) defined by the following first-order ODE and an initial value:

$$\begin{cases} \frac{dy}{dt} = f(t, y(t), x(t)), & t \in [0, T] \\ y(0) = y_0 \end{cases} \quad (2.1)$$

The state of the system at time t is found by calculating the integral of the ODE:

$$y(t) = y(0) + \int_0^t \frac{dy(s)}{ds} ds \quad (2.2)$$

Simple differential equations can sometimes be solved analytically using calculus. However, complex non-linear dynamics, as found in many practical problems, require numerical approximation methods instead.

2.2.1. Overview of theoretical models of greenhouse climate

To predict and control the humidity in a greenhouse, it is necessary to simulate its water balance. In this subsection, we survey the landscape of existing process-based greenhouse climate models, focusing on the water balances. While these models often share similarities, they capture the greenhouse dynamics with differing levels of complexity and with different equipment taken into account. Greenhouse dynamics are often subdivided into fast and slow dynamics. The fast dynamics are typically related to the climate variables and can change in the order of seconds, minutes, or hours (Katzin et al., 2022). The slow dynamics, on the other hand, are related to the crop processes and take much longer to change, specifically in the order of days, weeks, or months (Katzin et al., 2022). An often-used strategy for dealing with these diverging time scales is to separate the climate model from the crop model and solve these separately or assume that the state of the crop does not change for the duration of the simulation. The latter strategy is chosen in this work, so crop models will not be discussed.

Stanghellini et al. (1987) conducted one of the first studies that analysed the interrelationships between the micro-climate and humidity surrounding the crop. The author showed that the energy balance of a greenhouse crop provides a solid basis for quantifying the impact of the micro-climate on the humidity. These findings were supported by Jolliet and Bailey (1992), who analysed the effect of some meteorological variables (wind speed, vapour pressure deficit, solar radiation, and CO₂ concentration) on tomato transpiration and studied climate control in a greenhouse by evaluating five transpiration models. Later, Jolliet (1994) proposed the HORTITRANS model to predict air humidity and crop transpiration as functions of outside weather conditions to develop optimal control strategies for humidity in greenhouses. This model includes the processes of transpiration, condensation, ventilation, and humidification. Stanghellini and De Jong (1995) developed a model that combines a thorough model for natural ventilation of large-span greenhouses initially developed by Bot (1983) and De Jong (1990) with the transpiration model from Stanghellini et al. (1987) and the condensation model from Monteith and Unsworth (1990) to obtain a model that uses transfer functions of which the time-constant of the system could be determined for simulating and controlling the humidity. Another important feature of the model is the ability to simulate greenhouse humidity, given the availability of information such as meteorological and climate data. Additional simplifications and additions to the model from Stanghellini and De Jong (1995) were made by Bontsema et al. (2008) and Van Beveren et al. (2015). Alternatively, Tap (2000) discussed and implemented a greenhouse climate model using empirical relationships based on the process-based methods from De Jong (1990) and Stanghellini et al. (1987) with a large number of calibration parameters.

Another line of research into process-based theoretical models has sought to develop much more elaborate climate models for the greenhouse. De Zwart (1996) developed a complex model for simulating the greenhouse climate using a total of 14 state variables compared to the usual 3 (temperature,

humidity, CO₂) of the simpler models, including the vapour pressure below and above a curtain if available and temperatures of different parts of the greenhouse such as the soil, air, curtain, and more. Vanthoor (2011) builds upon this work and proposes a more general version that can be applied to various greenhouse designs and locations. Boulard et al. (2017) explores modelling the greenhouse climate using complex computational fluid dynamics (CFD). The practical benefits of using these more complex models for modelling and controlling the climate inside the greenhouse are limited, as simpler models perform well enough for most use-cases (Katzin et al., 2022). Complex models also typically require estimating and calibrating many more parameters and settings, making their use much more complicated (Katzin et al., 2022).

2.2.2. Limitations

While theoretical models provide valuable insights into the complex dynamics of greenhouse environments, they are not without limitations. Some of the challenges associated with using such models to understand and simulate real-world physical systems such as a greenhouse climate are:

1. **Detailed Understanding of Physics:** Theoretical models necessitate a profound comprehension of the underlying physics governing the physical system under study. Especially in a complex bio-physical system such as a greenhouse, understanding the heat transfer, fluid dynamics, and plant physiology within the enclosed environment is a difficult feat. Without a thorough understanding of these physical phenomena, constructing reliable models becomes challenging.
2. **Parameter Estimation:** To accurately simulate the behaviour of a greenhouse, numerous parameters need to be estimated/calibrated within the model, as this information is often difficult to measure accurately. These parameters might encompass physical properties like the thermal conductivity of materials (i.e. leaves), heat capacities of various components, and characteristics of the crops being cultivated in process-based models. Empirical models might contain many non-physical parameters that need to be calibrated using experimental measurements or data from existing literature, which may introduce uncertainties into the model. However, process-based models might also contain parameters for which measurements are unavailable, such as leaf temperature, resulting in a similar need for calibration and its accompanying issues. Moreover, experimental measurements of parameters are generally specific to the system used in the experiments, and changing to different environments (i.e. different greenhouses, different crops, or different locations) often necessitates re-calibration.
3. **Challenges in Calibration and Validation:** An extra difficulty that comes with the need for parameter estimation is that the calibration and validation process of theoretical models is a difficult problem and might require extensive data of various fluxes that are often unavailable and can be costly to obtain. Additionally, the calibration process is often made more difficult by the existence of many local minima (Villaverde et al., 2022). Furthermore, discrepancies between simulated and observed data may arise due to uncertainties in parameter estimation or inaccuracies in model assumptions, complicating the validation process.

In summary, while theoretical models offer valuable insights into greenhouse dynamics and are a promising way to perform simulations in an interpretable manner, they are often difficult to adopt and use due to the necessity for in-depth knowledge of the physics involved, the challenge of parameter estimation, specificity to individual setups, and the difficulties in calibration and validation. Addressing these limitations is crucial for enhancing the accuracy and applicability of such models in the greenhouse.

2.3. Machine learning models

Given the difficulties and limitations associated with theoretical models, machine learning has emerged as a promising alternative. Machine learning is a computational approach that empowers systems to automatically learn and improve their performance on a specific task through experience gained from data. Machine learning leverages algorithms that enable systems to recognise patterns, make decisions, and adapt without explicit programming. This section explores different machine learning methodologies, delving into their application for analysing and predicting time series data. The section is structured to encompass both classical methods and contemporary deep learning approaches. Classical methods, rooted in statistical algorithms, play a foundational role in time series forecasting, while deep learn-

ing methods, characterised by deep neural network architectures, represent modern advancements in extracting complex patterns from temporal data.

2.3.1. Classical methods

Classical methods in machine learning presuppose a statistical framework and make some initial assumptions about the task being learned. Due to their explicit nature, classical methods are generally more interpretable than deep learning methods.

Regression stands at the foundation of many state-of-the-art methods within the domain of classical machine learning methods for time series data. Regression analysis entails modelling the relationship between a dependent variable ($y \in \mathbb{R}^{d_y}$), often representing the target for prediction, and one or more independent variables ($x \in \mathbb{R}^{d_x}$) serving as features or predictors. In the context of time series forecasting, the dependent variable typically represents a numerical value associated with a specific time. Linear regression, the most basic approach, presupposes a linear relationship between the variables, seeking to identify the optimal fit line to minimise the discrepancy between predicted and observed values:

$$y_t = ax_t + b \quad (2.3)$$

Polynomial regression further extends this by adapting to non-linear patterns through the utilisation of higher-degree polynomials. There are many more advanced and dynamic regression methods, such as support vector regression, gaussian process regression, decision tree regression, or ensemble techniques like random forests and gradient boosting. These methods offer more versatility in capturing intricate patterns within time series data.

Exponential smoothing was proposed in the late 1950s (Brown, 1956) and has been the foundation of some of the most successful forecasting methods (Hyndman & Athanasopoulos, 2021). Exponential smoothing methods use exponentially decaying weighted averages of past observations to make forecasts, placing more weight on the more recent observations. Exponential smoothing (ETS) generates reliable forecasts quickly and for a wide range of time series, which is a great advantage and of major importance to applications in the industry.

The simplest form of ETS is called Single Exponential Smoothing (SES). SES is useful for univariate data with no trend and no seasonality. SES assigns a single parameter α , also known as the smoothing factor, to control the rate at which the influence of past observations decreases exponentially. The formula for SES is:

$$\begin{aligned} s_0 &= y_0 \\ s_t &= \alpha y_t + (1 - \alpha) s_{t-1} \\ \hat{y}_{t+1} &= s_t \end{aligned}$$

Several extensions exist to allow for seasonality, trends, and multivariate data to be modelled through exponential smoothing (Hyndman & Athanasopoulos, 2021).

Another common assumption used in forecasting time series data using classical methods is the concept of auto-regression (AR), assuming a dependency of future values in a time series on its past values. An AR model of order p can be written as follows:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \epsilon_t$$

where ϵ_t is white noise. AR models, due to their concise and explicit nature, offer interpretable results and are quite flexible (Hyndman & Athanasopoulos, 2021; Triebe et al., 2019).

Instead of using p past values of the target variable in a regression, one can also use q past forecast errors in a regression-like model. This is known as a moving average (MA) model of order q (Hyndman & Athanasopoulos, 2021).

$$y_t = c + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q}$$

where ϵ_t is white noise that models the prediction error at time t .

Stationarity, meaning that the time series' statistical properties do not depend on the time at which the series is observed, is a requirement for the AR and MA models above. For this reason, both the AR and MA models are often combined with differencing to ensure the stationarity requirement in the time series; we obtain an Auto-Regressive Integrated (read: inverse of differencing) Moving-Average (ARIMA) model.

$$y'_t = c + \phi_1 y'_{t-1} + \dots + \phi_n y'_{t-p} + \theta_1 \epsilon_{t-1} + \dots + \theta_q \epsilon_{t-q} + \epsilon_t$$

where y'_t is the differenced series. Several extensions to the ARIMA model exist, such as a variant for handling seasonality (SARIMA), a variant for handling exogenous variables (ARIMAX), or both (SARIMAX). Exponential smoothing and ARIMA models are two of the most widely used approaches to time series forecasting (Hyndman & Athanasopoulos, 2021). On the other hand, these methods require specific parameters, such as the orders p and q for ARIMA models, to be set. Setting these parameters requires analysis of the time series and a deeper understanding of the underlying processes.

To enable easier out-of-the-box forecasts to be created, Taylor and Letham (2017) proposed the Prophet model. Prophet has three main model components: trend, seasonality, and holidays. Prophet combines these components to form the model according to Equation 2.4. Additionally, support for exogenous variables was later added to the model. The advantages of this approach include interpretable parameters, the possibility to incorporate domain knowledge, and out-of-the-box performance with little hyperparameter tuning. However, some works have been critical of the out-of-the-box performance of Prophet compared to other classical and deep learning methods (Papastefanopoulos et al., 2020; Shah & Dimitrov, 2022). Nevertheless, Prophet is one of the most popular time series forecasting models at the time of writing.

$$y(t) = g(t) + s(t) + h(t) + \epsilon(t) \quad (2.4)$$

where:

- $g(t)$ is the trend-modelling function that can be specified as a linear or logistic function.
- $s(t)$ represents the seasonality function that can be daily, weekly, and yearly. Fourier terms handle this function.
- $h(t)$ is the holiday function that considers the effect of holidays, which occur irregularly.
- $\epsilon(t)$ represents the error that the model does not fit.

2.3.2. Deep learning methods

Given the explosive growth of available data and computing resources, recent advances in deep learning have produced many novel and transformative results across multiple tasks and disciplines, such as protein structure prediction (Jumper et al., 2021), image recognition (Krizhevsky et al., 2012), and algorithm discovery (Fawzi et al., 2022). Deep learning seeks to implicitly learn a model from the data through multi-layer feed-forward networks, so-called deep neural networks. Deep learning relies on the ability of these networks to approximate any function arbitrarily well under the universal approximation theorem (Hornik et al., 1989).

In machine learning, a neural network (NN) is a model inspired by the neurons in the brain. NNs learn and memorise patterns by strengthening or weakening the connections between neurons. The simplest form of a neural network is called a Perceptron, developed by Rosenblatt (1958). Later, Ivakhnenko and Lapa (1965) successfully trained the first real deep learning model, the multi-layer perceptron (MLP), with learnable "hidden layers" added in between the input- and output layers. A schematic overview of an MLP with a single hidden layer has been provided in Figure 2.5. The output of a single layer in an MLP can be calculated according to Equation 2.5:

$$f(x) = \sigma(Wx + b) \quad (2.5)$$

$W : \mathbb{R}^{m \times n}$ represents the weights of a layer with m neurons in the layer and n inputs, $x : \mathbb{R}^n$ represents the input to the layer, and $b : \mathbb{R}^m$ represents the biases of each neuron in the layer. σ can be any function but is usually a non-linearity such as the sigmoid- or tanh function. A NN propagates any given output through all the layers to eventually obtain a final output that can be used for either regression or classification purposes. In a supervised setting, the ground-truth value of each training sample is known and compared to the NN prediction. Consequently, the gradient of the loss with respect to

the NN parameters (weights & biases) is calculated through back-propagation, and the parameters are updated to reflect the new knowledge gained from the training example. For a more thorough explanation of the training, optimisation, and back-propagation algorithms of neural networks, we refer the reader to the open-source book by Prince (2023).

By running the training loop described above with large amounts of training data, neural networks are well suited to learning complex patterns in high-dimensional spaces. MLPs (or “vanilla neural networks”) make few assumptions about the structure of the problem at hand which often leaves the search space of all possible solutions too large for practical use cases. As such, many more specialised architectures have been devised for deep learning to add inductive bias and allow these architectures to learn more effectively.

In recent years, specific architectures for sequence-to-sequence problems have also gained popularity for forecasting time series. These deep learning methods include Recurrent Neural Networks (RNN) (Schmidt, 2019), Transformers (Vaswani et al., 2023), and (Temporal) Convolutional Neural Networks (CNN) (Lea et al., 2016; O’Shea & Nash, 2015). Initially designed for natural language and image processing, RNNs, Transformers, and CNNs might still be too complex for most time-series applications (Triebe et al., 2019). Their complexity and difficulty in explaining the models to decision stakeholders has further limited their adoption. However, two attributes make neural networks attractive for time-series modelling. First, as discussed before, neural networks can approximate any continuous function. Consequently, it is capable of solving many complex problems, given adequate data. Second, neural networks do not require restrictive assumptions on the underlying process from which the data is generated, lowering susceptibility to model misspecification problems compared to most parametric non-linear methods (Triebe et al., 2019).

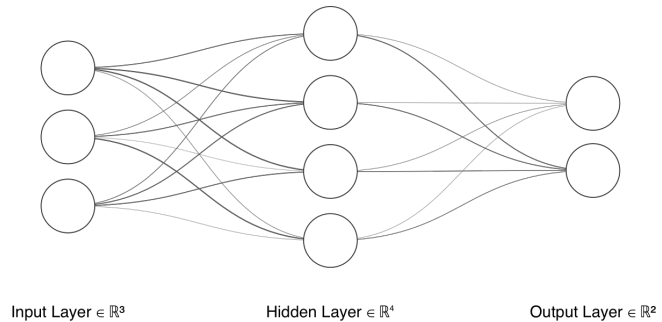


Figure 2.5: An example of a multi-layer perceptron with 3 input neurons, 4 hidden neurons, and 2 output neurons. The edges represent the weights of the network, with the thickness indicating the strength of the connection. This figure does not show the addition of biases or the application of non-linearities for clarity's sake.⁴

2.3.3. Limitations

Although increasingly popular in many fields, machine learning methods have limitations that limit their adoption in the simulation of physical processes and the greenhouse specifically. Some of the most important limitations are the following:

1. **Performance on Out-of-Distribution Samples:** Machine learning models often perform poorly on outliers, i.e. out-of-distribution (OOD) samples. This often stems from the models fitting the training data too closely, a concept called overfitting. This means that the model's performance may degrade significantly when presented with data that differs from the training distribution. Consequently, the reliability of machine learning models for extrapolating beyond the training data may be limited.
2. **Interpretability Issues:** Deep learning models are often referred to as “black boxes” due to their complex, non-linear nature. Classical machine learning models offer more interpretability, but in the context of physical processes, these models still lack the ability to explain predictions through physical causes. This lack of interpretability poses challenges in understanding how the model arrives at its predictions. Understanding the underlying mechanisms driving the model's decision-making process is crucial for trust and acceptance in the greenhouse (van Straten et al., 2000). This lack of trust is a primary reason that might explain the absence of deep learning models being used commercially for greenhouse climate prediction and control.
3. **Data Dependency and Scarcity:** Machine learning models typically require large amounts of

⁴This figure was created using <https://alexlenail.me/NN-SVG/index.html>

data for training to generalise well. However, if the available data is scarce or insufficiently representative of the system's dynamics, the performance of the machine learning model may degrade. Additionally, collecting and annotating large datasets can be time-consuming, expensive, and infeasible for specific domains, further limiting the applicability of deep learning methods.

In summary, while machine learning methods offer powerful tools for simulating complex dynamics, they are accompanied by limitations such as poor performance on out-of-distribution samples, interpretability issues, large data dependency, and computational complexity. Addressing these limitations is essential for harnessing the full potential of machine learning in dynamic system simulation and ensuring the reliability and trustworthiness of the generated models for increasing adoption.

2.4. Physics-informed machine learning models

Given that both theoretical models and ML models come with their drawbacks, the research community has begun to explore the space of hybrid models, where both prior scientific knowledge and ML are combined in a synergistic manner. Two main research directions emerged from this exploration. The first direction, which we will refer to as Data-Driven Physics (DDP), attempts to use data to build, identify, fit, or improve dynamical systems that can then be used for tasks such as simulation, control, and optimisation. Examples include but are not limited to Sparse Identification of Non-Linear Dynamics (SINDy) (Brunton et al., 2016) to identify non-linear dynamical systems using observation data, and Data-Enabled Predictive Control (DeePC) (Coulson et al., 2019); a method for deriving controllers directly from raw data without the need for system modelling or identification. In summary, these methods try to infer or improve physics knowledge using observation data.

The second direction attempts to incorporate prior physical knowledge into machine learning for better performance, interpretability, and data efficiency under the name of physics-informed machine learning (PIML). As this direction stands at the centre of this work, several important developments in PIML will be discussed below. Jointly, this new field is commonly referred to as Scientific Machine Learning (SciML) (Shin et al., 2020). A schematic overview of how PIML relates to Data-Driven Physics is provided in Figure 2.6.

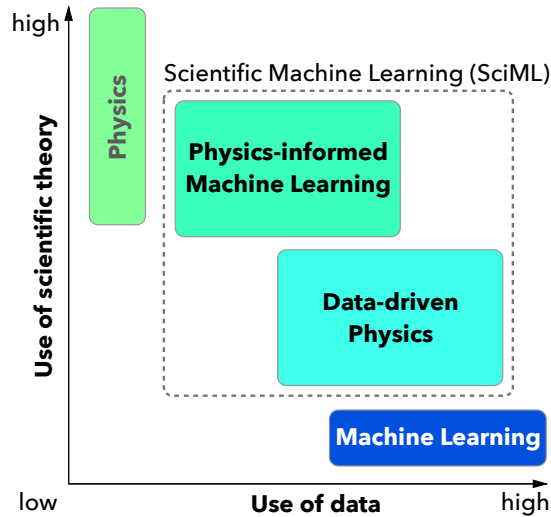


Figure 2.6: A schematic overview of how SciML and its subfields PIML and DDP relate to pure physics and pure data-driven methods in terms of the use of scientific theory and the use of data.

2.4.1. Physics-informed Neural Networks

Raissi et al. (2019) exploit the concept of automatic differentiation (Baydin et al., 2018) to integrate ordinary and partial differential equations as regularisation terms in the loss function of deep neural networks to obtain physics-informed neural networks (PINN). The method has demonstrated a series of promising results for a diverse collection of problems in computational science, such as data-driven solving of PDEs and data-driven discovery of PDEs. A series of subsequent works have further demonstrated the usefulness of PINNs in one-, two-, and three-dimensional problems (Han et al., 2018; Pang et al., 2019; Raissi et al., 2020), and a theoretical justification for PINNs was provided by Shin et al. (2020). The most useful area of PINNs and their many variations are in solving complex PDEs in a tractable amount of time, improving the generalisation capabilities of the neural networks, and reducing the amount of data needed to learn a good solution. Although PINNs can also be applied to solving simpler ODEs, the practical benefits are limited due to the fact that state-of-the-art ODE solvers are already sufficiently capable of solving these kinds of problems in a reasonable amount of time whilst being able to provide practical guarantees on the error tolerances. Furthermore, PINNs do not tackle the interpretability issues that are inherent to deep learning, and they do not provide hard constraints on the learned solution.

2.4.2. Neural Differential Equations

In the context of time series forecasting, the previously discussed deep learning methods tackle the problem by modelling the solution trajectory $y(t)$ using some neural network. Chen et al. (2019) take a different approach and use a neural network to parameterise the vector field of the solution instead (Equation 2.6).

$$y(0) = y_0 \quad \frac{dy(t)}{dt} = f_\theta(t, y(t)). \quad (2.6)$$

where θ represents some vector of learned parameters, $f_\theta: \mathbb{R} \times \mathbb{R}^{d_1 \times \dots \times d_k} \rightarrow \mathbb{R}^{d_1 \times \dots \times d_k}$ is any standard neural architecture, such as a feed-forward network, and $y: [0, T] \rightarrow \mathbb{R}^{d_1 \times \dots \times d_k}$ is the solution.

The central idea of neural (ordinary) differential equations is to use a differential equation solver as part of the learned differentiable computation graph. In their seminal paper, the authors propose a method to allow back-propagation through the differential equation solver in the form of the adjoint sensitivity method; for more information on this topic, the reader is directed to the paper (Chen et al., 2019). The approach has several benefits compared to alternative approaches. The first benefit is that neural ODEs provide a convenient way to add an *inductive bias* to the model in the form of (partially) known differential equations. Second, treating discrete data in a continuous-time regime offers a way to treat irregular data (a common feature of time series) on the same footing as ‘regular’ data (Kidger et al., 2020). Neural ODEs are known to take longer to train and predict than other deep learning architectures due to the need to integrate the solutions over time, a process that is hard to parallelise as each step relies on the output of the previous step. Another shortcoming of neural ODEs is that once θ has been learned, then the solution to Equation 2.6 is determined by the initial condition y_0 , and there is no direct mechanism for incorporating data that arrives later. To address this, subsequent work also introduced neural controlled differential equations (CDEs) as a continuous analogue of recurrent neural networks (Kidger et al., 2020). A neural CDE allows for changing dynamics due to the inclusion of a varying driving signal $x(t): [0, T] \rightarrow \mathbb{R}^{d_x}$ in the vector field specification. This driving signal might describe exogenous forces on the system, such as meteorological conditions over time. Rewritten to follow the form of the neural ODE vector field description from Equation 2.6, we obtain the following:

$$y(0) = y_0, \quad \frac{dy(t)}{dt} = f_\theta(t, y(t)) \frac{dx(t)}{dt} \quad (2.7)$$

Despite the obvious inspiration of the architecture from calculus and theoretical models, neural differential equations are still considered black-box models, given their complete reliance on neural networks. A generalised notion of neural differential equations was coined by Rackauckas et al. (2021) in the form of Universal Differential Equations (UDE), i.e. differential equations defined in whole or part by a universal approximator. UDEs described in part by a theoretical model can be considered grey-box models. In this sense, UDEs can elegantly incorporate prior physical knowledge about the dynamics of a system in the model definition. Yin et al. (2021) propose Aphynity, a theoretical framework that allows for stable training of such grey-box UDEs for forecasting complex time series with prior physical knowledge

that is incomplete or incorrect. The authors show the efficacy of their method on benchmark problems such as a reaction-diffusion system and a damped wave system. Nonetheless, it should be noted that the approach was not tested on more complex real-world physical systems with noisy data, such as a greenhouse. Moreover, Aphynity has no mechanism for adapting to dynamics that change over the course of a simulation, mirroring the limitation experienced by neural ODEs that was described above. This limits the use of Aphynity to systems with fixed dynamics, which is seldom the case in real-world dynamical systems, such as the greenhouse, that are often influenced, either actively or passively, by exogenous forces.

2.4.3. Other Approaches

Raissi et al. (2018) introduced Numerical Gaussian Processes (NGP) as a method to accurately infer solutions and provide uncertainty estimates for mathematical physics problems. However, encoding prior knowledge in the form of theoretical models and differential equations in the GP kernel is challenging, and the method is limited to discrete-time domains with further limitations for strongly non-linear regimes.

Another line of research tries to imbue principles of physics into the architecture of different deep learning architectures. Examples of this are amending an LSTM architecture to make sure that the predicted quantity, water density in this case, is monotonically increasing with depth (Daw et al., 2019), and augmenting a standard NN with n conservation layers to enforce the conservation laws in the context of climate modelling (Beucler et al., 2019).

2.5. Discussion

Due to the complex non-linear dynamics that govern the climate inside the greenhouse, forecasting the future state of the climate is not trivial. Much research has gone into developing theoretical models to describe these dynamics and use them for optimal climate control. The advantages of theoretical models are clear: interpretable results and the ability to causally relate the system's input and output. In reality, theoretical models are difficult to use. Since each model has a different goal and complexity and includes different equipment, choosing a model is not trivial and typically requires adaptation. Due to the commonly shared global structure of the dynamics, the composition of different sub-models to fit one's specific needs is a common strategy used in related work. However, another important question that arises is how complex a model for simulating the greenhouse climate should be. Katzin et al. (2022) mention the evidence that speaks to the power of simpler models and specify the need for a balance between simplicity and complexity. However, practical advice for finding this balance in complexity is hard to come by. In the context of this work, the data imposes a constraint on the model complexity as it is limited to meteorological measurements and forecasts, greenhouse climate measurements, and the states of various pieces of control equipment present in the greenhouse under study. Models that require detailed information about the crop, such as crop temperature, are therefore hard to use. Many models require the LAI of the crop to be known to simulate the humidity inside the greenhouse. Generally, this information is unavailable and changes over time, making calibration more difficult. Furthermore, a model should be able to simulate the climate across different greenhouses and in different locations. Given that many simpler theoretical models require greenhouse-specific parameters to be estimated for optimal functioning, one would need a way to reliably calibrate these parameters, even though measurements for the various fluxes found in the models are often hard to come by.

Machine learning, on the other hand, does not suffer from these issues. Its flexibility lends itself to easy reuse across different greenhouses. In theory, a ML model could adapt to the specific situation of any greenhouse by learning. In practice, however, several issues prevent ML from being widely used in greenhouse climate simulation and control. Several studies have explored the use of deep learning models for greenhouse climate prediction (G. Liu et al., 2024), but these models lack the interpretability that the theoretical models offer. Moreover, these models typically require a lot of data to learn an adequate model. Although simpler and more easily interpreted than deep learning models, classical methods might not be able to capture the complex non-linear behaviour of the greenhouse climate (G. Liu et al., 2024; Taylor & Letham, 2017). Furthermore, classical methods often require stationarity of the time series, something which is generally not the case for humidity in the greenhouse. This problem

might be solved by differencing, but this has the drawback of losing some information contained within the data. To summarise, machine learning models suffer to different degrees from the issues of lack of interpretability, data dependency, and extrapolation to out-of-distribution scenarios.

PIML methods have shown great promise in many different scientific fields such as chemistry (Anderson et al., 2019), climate science (Beucler et al., 2019; Kashinath et al., 2021), material engineering (D. Liu & Wang, 2019), and biological sciences (L. Liu et al., 2021; Yazdani et al., 2020). However, at the time of writing, no research has investigated applying PIML methods to the context of greenhouse climate simulation. While PINNs have shown great promise in many important areas, they are not a great fit for the problem of predicting the greenhouse climate, as they suffer from the same interpretability issues experienced by other deep learning methods. Other techniques, such as NGPs (Raissi et al., 2018) are also not a great fit for this use case, as it would be incredibly difficult to encode the prior knowledge about the greenhouse climate in the GP kernel. Furthermore, Raissi et al. (2018) have noted the difficulties of applying NGPs to a strongly non-linear regime under which the greenhouse climate falls.

Neural differential equations and the Aphynity framework (Yin et al., 2021), on the other hand, provide a promising avenue for addressing the issues experienced using a theoretical model for simulating the greenhouse climate. Aphynity offers a way to conveniently integrate a theoretical model of the greenhouse climate with a data-driven part that can correct for incorrect and missing physics or adapt the model to different greenhouses and conditions while allowing parameters to be learned through commonly available data. Following this approach, adherence to physics, out-of-distribution performance, and data needs might be improved. However, Aphynity is not directly applicable to the context of greenhouse climate simulation as the greenhouse climate dynamics strongly depend on the changing meteorological conditions, control equipment, and crop state. Aphynity lacks any mechanism for dealing with these changing dynamics, and no such mechanism has yet been developed for Aphynity, so a new method should be developed to deal with this issue.

Kidger et al. (2020) discusses adding such a mechanism to purely data-driven neural ODEs in the form of Neural CDEs. However, Neural CDEs are still black-box models and don't directly allow for integration with prior knowledge. Nevertheless, the insights of Kidger et al. (2020) might be combined with the Aphynity framework to build a novel model that can handle dynamics that change over time such that it can be used for greenhouse climate simulation.

Theoretical Model for Greenhouse Humidity Deficit Prediction

Inside the greenhouse, the temperature, humidity, and CO₂ concentration are modelled for simulation and control by considering balances: an energy balance, a water vapour balance, and a CO₂ balance. Not all greenhouse climate models describe all three balances: some focus only on energy, whereas others focus more on humidity. For each of these balances, incoming and outgoing flows are identified, and the difference between the incoming and outgoing flows is the net change in each attribute. Many different theoretical models of the greenhouse climate have been proposed over the years with a wide variety of complexity and focus (Katzin et al., 2022). Though most models share commonalities, many differences exist in calculating the fluxes. Section 3.1 presents a high-level overview of a theoretical model for describing the humidity deficit over time. Implementation details are presented in section 3.2, and the model is tentatively validated in section 3.3 through several experiments to gauge its performance, characteristics, and limitations under different conditions. Finally, section 3.4 concludes the findings in this chapter.

3.1. Model overview

The following provides a general overview of the theoretical model with the necessary information for understanding the rest of this work. Many different sub-models exist for each of the transpiration, condensation, and ventilation fluxes. The presented theoretical model has been composed of these different sources to obtain a differentiable model that suits the specific needs of this work and can be used in a PIML model. The sub-models used in this work are largely based on the models from Bontsema et al. (2008), De Zwart (1996), Monteith and Unsworth (1990), Stanghellini and De Jong (1995), Van Beveren et al. (2015), and Vanthoor (2011). A schematic overview of the greenhouse, the humidity fluxes, and the climate control equipment employed in the greenhouse used in this study can be found in Figure 2.3. A more detailed description of the sub-models used in this work is provided in Appendix C.

The humidity balance for a greenhouse without any fogging or dehumidifying equipment is generally given by (Henten, 1994; Stanghellini & De Jong, 1995; Stanghellini et al., 1987):

$$\frac{d\mathcal{X}_{air}}{dt} = \frac{1}{h}(E - V - C) \quad (\text{g m}^{-3} \text{ s}^{-1}) \quad (3.1)$$

where:

- h (m) is the average height of the greenhouse.
- $\mathcal{X}_{air}$ (g m⁻³) is the absolute humidity of the greenhouse air
- E , V , C (g m⁻² s⁻¹) represent the changes in absolute humidity caused by crop transpiration, ventilation, and condensation respectively.

It is important to note that the model presented in this chapter aims to simulate the humidity deficit (HD) instead (as defined in section 2.1). As a change in absolute humidity is equal to an opposite but equally large change in humidity deficit, no large changes to the model are required. The only change pertains to multiplying the result of Equation (3.1) by -1 to obtain the respective change in humidity deficit:

$$\frac{dHD}{dt} = -\frac{d\mathcal{X}_{air}}{dt} \quad (\text{g m}^{-3} \text{ s}^{-1}) \quad (3.2)$$

The next sections provide a general overview of each sub-model for the transpiration, condensation, and ventilation fluxes. Each flux is written as a transfer function to make clear the relationship between the current humidity and the resulting vapour flux. The conductance, or the capacity for vapour to flow, through each sub-model at any time is useful for visualising the impact of each sub-model on the resulting absolute humidity in the greenhouse. An example of such a visualisation of the conductances for the sub-models can be found in section 3.3.

Transpiration

Transpiration is the process used by the crop to evaporate water from the surface of its leaves and is the main source of water vapour entering the greenhouse air. A wide range of models have been developed to model transpiration for big-leaf crops such as the tomato crop. Generally, the equations for transpiration are derived from the Penman-Monteith equation (Monteith & Unsworth, 1990). The transpiration sub-model used here is based on the work from De Zwart (1996), Stanghellini et al. (1987), Van Beveren et al. (2015), and Vanthoor (2011), which all use an adapted version of the Penman-Monteith equation to model transpiration/evaporation based on meteorological and climate conditions. The transpiration vapour flux depends on the difference between the absolute humidity in the leaves of the crop and the humidity in the air surrounding the crop. Furthermore, the amount of actual transpiration is influenced by the stomata of the plant and the leaf area from which the transpiration can take place.

The change in absolute humidity due to transpiration, E , can be described by the following transfer function:

$$E = g_e(\mathcal{X}_{crop} - \mathcal{X}_{air}) \quad (\text{g m}^{-2} \text{ s}^{-1}) \quad (3.3)$$

where:

- g_e (m s^{-1}) is the transpiration conductance, which depends on factors such as the leaf area, stomatal resistance of the crop, and the leaf boundary layer resistance of the crop.
- $\mathcal{X}_{crop}$ (g m^{-3}) is the absolute humidity inside the crop, which is dependent on factors such as the leaf area and the net radiation absorbed.

Condensation

When the temperature of the greenhouse cover is below the dew point temperature of the air in the greenhouse, condensation starts to form on the cover. Condensation mostly occurs in winter when the windows are closed, and the temperature difference between the inside and outside air is large (Stanghellini, 2024). Greenhouses are generally equipped with scrubbers or gutters to remove this condense from the cover (Stanghellini et al., 1987), meaning that water vapour does not evaporate from the cover into the air. Similar to transpiration, Monteith and Unsworth (1990) also provide a widely used model for condensation on horizontal plates that has been shown to be applicable to Venlo-style greenhouses (Papadakis et al., 1992). The condensation sub-model used here is based on the work by Bontsema et al. (2008), Monteith and Unsworth (1990), Stanghellini and De Jong (1995), and Van Beveren et al. (2015).

The change in absolute humidity due to condensation, C , on the greenhouse cover can be described by the following transfer function:

$$C = \begin{cases} g_c(\mathcal{X}_{air} - \mathcal{X}_{cov}^*) & \text{if } \mathcal{X}_{air} \geq \mathcal{X}_{cov}^* \\ 0 & \text{otherwise} \end{cases} \quad (\text{g m}^{-2} \text{ s}^{-1}) \quad (3.4)$$

where:

- g_c (m s^{-1}) is the condensation conductance, which is dependent on the temperature difference between the air and the cover and the geometry of the condensation surface.
- $\mathcal{X}_{cov}^*$ (g m^{-3}) is the saturation humidity at the temperature of the greenhouse cover.

Ventilation

Natural ventilation occurs by the exchange of air inside the greenhouse and the outside air when the windows are opened. Depending on the wind speed, window opening, and temperatures of the inside and outside air, ventilation can play a large role in the humidity levels inside the greenhouse. Ventilation can be both a source of water vapour entering the greenhouse air or a sink of water vapour exiting the greenhouse air, depending on the difference in the humidity levels inside and outside the greenhouse. The ventilation sub-model used in this work is based on the work by Bot (1983), De Jong (1990), De Zwart (1996), Stanghellini and De Jong (1995), and Vanthoor (2011).

The change in absolute humidity due to ventilation, V , can be described by the following transfer function:

$$V = g_v(\mathcal{X}_{air} - \mathcal{X}_{out}) \quad (\text{g m}^{-2} \text{s}^{-1}) \quad (3.5)$$

where:

- g_v (m s^{-1}) is the ventilation rate or conductance, dependent on factors such as the number of windows, their opening angles, wind speed, and temperature differences between the inside and outside air.
- $\mathcal{X}_{out}$ (g m^{-3}) is the absolute humidity outside the greenhouse.

3.2. Model implementation

To integrate Equation (3.1), the model requires knowledge or estimates of several climate and meteorological conditions, as well as the state of the climate control equipment. To make this dependency explicit, we rewrite Equation (3.1) as a function of these inputs. First, we denote the (hidden) state of the humidity deficit by $y \in \mathbb{R}$. Next, we denote the climate and meteorological conditions by $x_c \in \mathbb{R}^{d_{x_c}}$, $d_{x_c} = 5$. Finally, the state of the control signals is denoted by $u \in \mathbb{R}^{d_u}$, $d_u = 3$:

$$y = [-\mathcal{X}_{air}], \quad x_c = \begin{bmatrix} T_{out} \\ Q_{sun} \\ \mathcal{X}_{out} \\ T_{air} \\ w \end{bmatrix}, \quad u = \begin{bmatrix} \alpha_{lee} \\ \alpha_{wind} \\ Q_{lights} \end{bmatrix} \quad (3.6)$$

Because we are not performing control optimisation (i.e. we consider u to be specified ahead of time), we let $x = (x_c, u)$. Furthermore, to allow for the use of an adaptive step-size solver, x needs to be interpolated. We assume that we have access to a dataset $\mathcal{D} = (y_{k\Delta t}, x_{k\Delta t})_{0 \leq k \leq \lfloor T/\Delta t \rfloor}$. Subsequently, we define $x_{\mathcal{D}} : [0, T] \rightarrow \mathbb{R}^{d_x}$ to be an interpolation of x . In our case, we let $x_{\mathcal{D}}$ be a natural cubic spline. This interpolation scheme ensures smooth gradients, which will be useful for the implementation in the PIML model from chapter 4. Combining the above, we obtain the following model definition:

$$\frac{dy(t)}{dt} = f(y(t), x_{\mathcal{D}}(t)) \quad (3.7)$$

The theoretical model was subsequently implemented in Python.

3.3. Model validation

It is important that the model is able to predict the humidity deficit inside the greenhouse well enough for it to be useful when implemented in a PIML model. Therefore, this section examines the performance of the model presented in this chapter for a specific Venlo-style greenhouse in the Netherlands growing a tomato crop. To account for the changing impact of the growing crop on indoor humidity, as well as variations in climate control equipment usage and weather conditions throughout the seasons, our experiments are conducted at different stages of the crop cycle and across different seasons. This allows us to better understand the impact of these factors on the validity of the model. In subsection 3.3.5, a comparison between the simulations is made, and the strengths and weaknesses are discussed.

3.3.1. Data

For the experiments below, data from a commercial Venlo-style greenhouse located in the Netherlands with a tomato crop was used to calibrate and validate the model. The greenhouse consists of multiple compartments with a climate measurement installation for each compartment. The greenhouse climate data for the entire greenhouse was obtained by averaging the measurements from each compartment. The meteorological data that has further been used as input to the model was also collected by local measurement equipment located at the greenhouse. From this installation, the temperature ($^{\circ}\text{C}$), global radiation (W m^{-2}), wind speed (m s^{-1}), and relative humidity outside (%) were gathered. Finally, data on the window openings (%) (wind and lee side), artificial radiation (W m^{-2}), and blackout curtain deployment (%) over time was gathered from the climate computer. All data was collected at 5-minute intervals (Δt).

3.3.2. Experimental Setup

Four experiments were performed to assess the validity of the model and gain insights into its performance. The data used in these experiments was selected to cover each season and different phases of the crop cycle, which lasted from 28-11-2022 until 03-10-2023. Each simulation takes place on the 1st day of the month and was simulated for 72 hours (T) starting at 7 AM. For each experiment, the model takes the first state y_0 and the interpolation function $x_{\mathcal{D}}$ from section 3.2 of the *measured* data as input and predicts the remaining time-steps at 5-minute intervals by integrating Equation (3.7) through a differentiable, generic, and common adaptive step-size ODE solver (Explicit Runge-Kutta of order 5(4)). This ODE solver was found to provide a good balance between accuracy and speed.

The information that was provided about the greenhouse under study was limited to the country of origin, the type of greenhouse, and the type of crop. Importantly, this means that some of the necessary parameters for the model have been estimated based on best estimates, related work, and hand calibration. With the notable exception of the Leaf Area Index, all parameters were held constant across the different experiments performed in this chapter. An overview of these parameters can be found in Table 3.1. The calibrated LAI is denoted for each experiment individually and has been estimated according to a rough estimate of plant growth over time, i.e. the LAI increases as it matures, coupled with calibration by trial-and-error. It is logical to question the validity of the model parameters. We admit that these are quite likely wrong, and the results should thus be taken with a fair grain of salt, but the results of the simulation can still provide interesting insights into the model. The conductances of the three sub-models are also shown in a similar fashion to Stanghellini and De Jong (1995). This provides insights into the contribution of the different fluxes acting upon the greenhouse humidity. Finally, the meteorological conditions, climate measurements, and control decisions are plotted for the duration of the simulation to provide some background on the simulation result.

Table 3.1: An overview of the parameter values used in the experiments of subsection 3.3.4, and how they were chosen. The average height of the greenhouse was found to be fairly inconsequential to simulation results when kept within reasonable bounds. As such, an average of 10 m was assumed, in line with modern Dutch Venlo-style greenhouses.

Symbol	Description	Value	Origin
h	Avg. height of greenhouse	10 m	Assumption
$A_o (L_o * H_o)$	Area of a single window	1.168 m^2	Calibration
(l/h)	Aspect ratio of a window	1.825 m	Taken from (De Zwart, 1996)
fr_{window}	Number of windows / greenhouse area	0.22 m^{-2}	Calibration

3.3.3. Determination of model performance

Model performance was quantitatively evaluated using the Root Mean Squared Error (RMSE), the Relative Root Mean Squared Error (rRMSE), and the coefficient of determination or R^2 -score.

The RMSE shows the error in absolute terms in the same unit as the quantity being predicted, humidity deficit (g m^{-3}). This allows experts to interpret the information more easily based on their experience.

The RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_{model, i} - y_{data, i})^2} \quad (\text{g m}^{-3}) \quad (3.8)$$

where n is the number of measurements in the simulation, and $\hat{y}_{model, i}$ and $y_{data, i}$ are the simulated and measured HD, at time i , respectively.

The rRMSE converts the RMSE to a standardised scale, allowing for easier comparison with different methods for simulating the greenhouse climate. We calculate the rRMSE similar to Vanthoor (2011):

$$rRMSE = \frac{1}{\bar{y}_{data}} RMSE \quad (3.9)$$

where $\bar{y}_{data}$ represents the mean of the measured humidity deficit over the total simulation time span.

In statistics and regression, the coefficient of determination (R^2) indicates the proportion of variance in the dependent variable (measured humidity deficit) that is explained by the independent variable(s) (simulated humidity deficit). It offers valuable insight into the model's overall goodness of fit and complements RMSE and rRMSE by quantifying the model's predictive power relative to the total variance. An R^2 -score of 1 indicates a perfect fit, whereas an R^2 -score of 0 indicates that the model performs no better than a baseline prediction of the data mean ($\bar{y}_{data}$). A model can also be arbitrarily worse, resulting in negative scores. The R^2 -score is calculated according to:

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_{model, i} - y_{data, i})^2}{\sum_{i=1}^n (\bar{y}_{data} - y_{data, i})^2} \quad (3.10)$$

For simulating the humidity deficit inside the greenhouse, it is assumed that an RMSE of 1.5 g m^{-3} is sufficient for the purpose of optimal control based on previous results by Su et al. (2018) and Van Beveren et al. (2015).

As noted by Katzin et al. (2022), rRMSE provides a convenient comparison between predictions in different units, but it can also be misleading when comparing the same units under different settings. Combined with a lack of a justified standard (Katzin et al., 2022), no fixed permitted rRMSE is specified in this work; instead, the rRMSE is used for obtaining additional insights in comparison to different models operating on different unit measurements.

Over the course of a year, indoor and outdoor conditions change drastically. This also greatly influences the variance of the humidity deficit, resulting in hard-to-compare R^2 -scores. For this reason, no reasonably low benchmark for the R^2 -score is used in this work; namely, an R^2 -score of higher than 0 is deemed required for the model to be useful in a PIML model, with higher scores indicating better performance.

3.3.4. Results

For each simulation, the RMSE, the rRMSE, and R^2 -score are shown in Table 3.2 to gauge the performance of the model for each simulation. Additionally, for each simulation, the simulated HD is plotted against the measured HD for visual inspection of the results. Additionally, the conductances for the transpiration, condensation, and ventilation are plotted below to analyse the role played by each flux over time. The meteorological, climate, and control conditions are further plotted, from which their impact on the conductances can be inspected. The shaded areas in the figures represent the times when the blackout curtain was deployed. Generally, one can observe that the conductances in autumn and winter are generally lower than in spring and summer. This aligns with the reality that, due to the absence of ventilation, the HD varies less than in spring and summer. As a result, condensation is also seen to play a much larger role in autumn and winter, compared to spring and summer, where ventilation dominates the effect of condensation on the humidity in the greenhouse.

Finally, a more thorough analysis of the performance of the theoretical model with learned parameters over a larger dataset can be found in chapter 4.

Table 3.2: A summary of the validation results of the theoretical humidity deficit model, including the root mean square error (*RMSE*), Relative Root Mean Square Error (*rRMSE*), and Coefficient of Determination (R^2).

Simulation example	<i>RMSE</i>	<i>rRMSE</i>	R^2
Autumn	0.854 g m^{-3}	0.143	-0.44
Winter	1.287 g m^{-3}	0.575	-4.05
Spring	1.199 g m^{-3}	0.427	0.55
Summer	0.766 g m^{-3}	0.298	0.80

Autumn example with very young crop – 2022-12-01

The first simulation was run for a very young crop, about 3 days into the crop cycle. The simulation results can be seen in Figure 3.1, together with the conductances of the three sub-models. Below these, the weather, climate, and control conditions inside and outside the greenhouse during the simulation are plotted. The LAI was calibrated to be approximately $0.7 \text{ m}^2 \text{ m}^{-2}$. The simulation takes place in autumn with cold outside temperatures of around 2°C to 8°C . Ventilation is generally limited to avoid heat losses in these conditions, and condensation is more likely due to the large temperature differences between the inside and outside air. Lower radiation levels are supplemented using artificial lights, and blackout curtains are used throughout the night, trapping air and blocking outgoing light. Due to the absence of blackout curtains in the theoretical model, a larger error is expected during the blackout curtain deployment. Because of the lower variance in the HD during the simulation, the low R^2 -score (-0.44) paints a slightly distorted picture of the performance. From visual inspection, the simulated HD follows the measured HD seemingly decently, with structural errors mostly during the deployment of the blackout curtain on the second and third nights, as was expected. The low RMSE of 0.854 g m^{-3} and rRMSE of 0.143 confirm the good performance of the theoretical model in this scenario.

Winter example with young crop – 2023-02-01

The second simulation was run during winter. The simulation results can be seen in Figure 3.2, together with the conductances of the three sub-models. Below, the weather, climate, and control conditions inside and outside the greenhouse during the simulation are plotted. The LAI for this simulation was estimated to be approximately $2 \text{ m}^2 \text{ m}^{-2}$. During the winter, the greenhouse acts similarly to a closed system due to the limited ventilation, and the humidity deficit inside the greenhouse tends to be very low. As in the autumn simulation, blackout curtains are used with the addition of artificial radiation to supplement the limited amount of solar radiation during the winter. The shaded areas represent the times when the blackout curtain was deployed, and a noticeable error during the deployment of the blackout curtain can be seen. This error is less at the end of each night, when artificial lighting is activated, indicating that this lighting possibly offsets the impact of the blackout curtain on the dynamics of the humidity in the greenhouse. The R^2 -score of -4.05 indicates quite poor performance but is largely caused by the structural error during the blackout curtain deployment. Similarly, the RMSE is also higher for this reason, with the rRMSE being even more distorted due to the combination of this error with the low average HD during the simulation period. From visual inspection, the simulated HD accurately follows the measured HD outside the times when the blackout curtain was deployed.

Spring example with semi-mature crop – 2023-05-01

The third simulation was run during spring. The simulation results can be seen in Figure 3.3, together with the conductances of the three sub-models. Below, the weather, climate, and control conditions inside and outside the greenhouse during the simulation are plotted. The LAI for this simulation was estimated to be approximately $2.3 \text{ m}^2 \text{ m}^{-2}$. With spring come increased temperatures and increased solar radiation. Ventilation also starts playing a bigger role in managing the humidity inside the greenhouse. As a result, the greenhouse humidity deficit fluctuates more between day and nighttime. Artificial radiation and blackout curtains are still used, but to a lesser extent. Again, the shaded areas represent the times when the blackout curtain was deployed, and the omission of blackout curtains in the theoretical model can be seen in the result in terms of a structural error. Like the winter simulation, the error during blackout curtain deployment fades when artificial lighting is turned on. Given the shorter lengths of blackout curtain deployment combined with good performance during the daytime, when blackout curtains are not deployed, and higher variance in the HD, the R^2 -score (0.55) improves drasti-

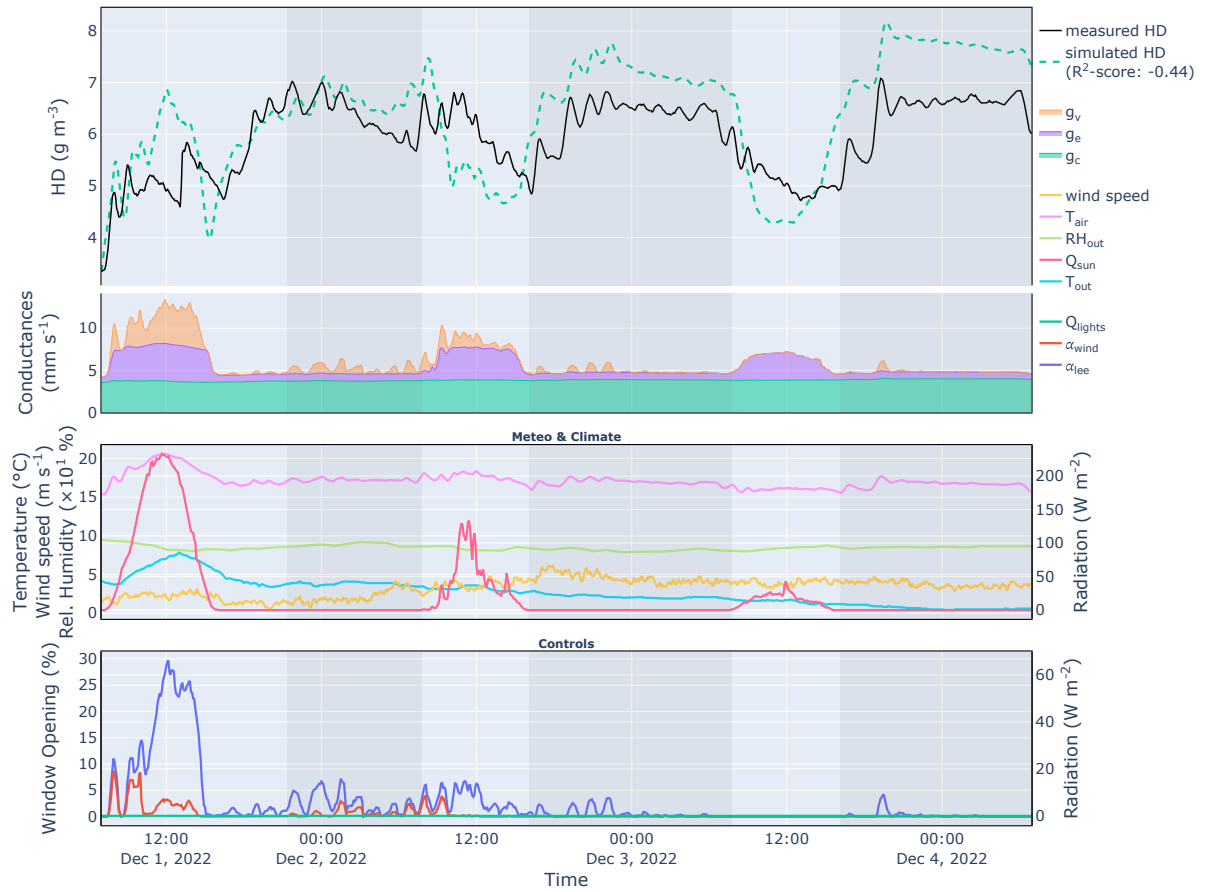


Figure 3.1: A simulation of the humidity deficit plotted against the measured humidity deficit and the conductances of the three sub-models from 2022-12-01 up until 2022-12-04 with a very young crop ($LAI \approx 0.7$), combined with weather, climate, and control conditions during the simulation. The shaded areas in the plot represent when the blackout curtain was deployed.

cally compared to previous examples. The RMSE (1.199 g m^{-3}) and rRMSE (0.427) are, largely due to the present error during the nighttime, still fairly high. Nevertheless, the model accurately follows the measured HD during daytime when visually expected.

Summer example with mature crop — 2023-08-01

The fourth simulation was run during summer. The simulation results can be seen in Figure 3.4, together with the conductances of the three sub-models. Below, the weather, climate, and control conditions inside and outside the greenhouse during the simulation are plotted. The LAI for this simulation was estimated to be approximately $2.5 \text{ m}^2 \text{ m}^{-2}$. During summer, the difference in temperature between the inside and outside is generally the lowest of the year. Solar radiation is also at its peak, meaning supplemental radiation is no longer necessary. Blackout curtains are also generally not deployed during this period. As a result, the structural error during blackout curtain deployment from previous simulations is not present in this simulation. Due to the small differences between inside and outside temperatures, condensation plays a much smaller role than in the other simulations. Ventilation, however, has a much larger impact on the HD inside the greenhouse as it is actively used to steer the greenhouse climate. From visual inspection, the simulated HD accurately follows the fluctuations of the measured humidity deficit in the greenhouse, confirmed by the respectable R^2 -score of 0.80. The RMSE (0.766 g m^{-3}) and rRMSE (0.298) are also lower than in springtime, as a result of the absence of the structural blackout curtain error previously present.

3.3.5. Discussion

Given the limited number of experiments run in this section, a comparison of performance to models of related work is left for chapter 4. However, looking at the results of the four validation experiments,

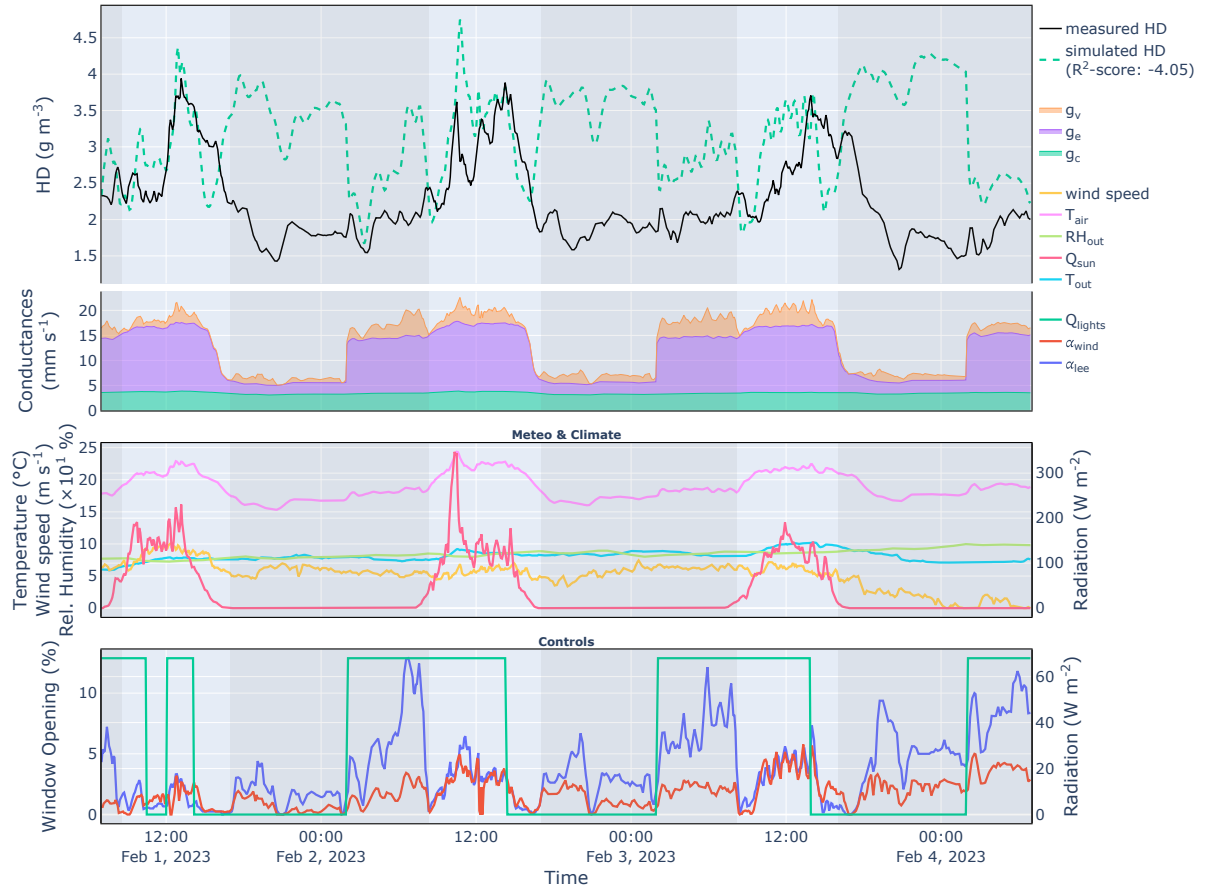


Figure 3.2: A simulation of the theoretical model of the humidity deficit plotted against the measured humidity deficit and the conductances of the three sub-models from 2023-02-01 07:00 up until 2023-02-04 07:00 with a young crop ($LAI \approx 2$), combined with weather, climate, and control conditions during the simulation. The shaded areas in the plot represent when the blackout curtain was deployed.

one can identify a difference in performance over the different seasons and conditions inside the greenhouse, showing particularly poor scores during the winter and much better performance during spring and summer. A significant part of the performance differences between seasons can be attributed to errors caused by the omission of the blackout curtains' effect on the greenhouse climate.

The theoretical model presented in this chapter was specifically designed to be simple to reduce the parameters and data required for simulation. Using this simple theoretical model still requires knowledge of several key aspects of the greenhouse, such as the number of windows, window geometry, greenhouse height, LAI , and more. Most of these quantities have been estimated based on prior work, assumptions, and calibration. This has likely played a role in limiting the performance of the theoretical model. On the other hand, the theoretical model requires several inputs that must be known beforehand, such as the window openings, blackout curtain deployment, meteorological conditions and inside air temperature. These quantities were taken to be known ahead of time during simulation, a feature that is not necessarily true in the real world.

Alternatively, one could use predictions or forecasts of these quantities during simulation. Obtaining forecasts of the inside air temperature should not pose a problem as climate computers can steer towards and maintain a desired temperature inside the greenhouse quite successfully (Stanghellini & De Jong, 1995). Moreover, meteorological forecasting provides a highly accurate estimate of weather conditions in the short-term (i.e. three days) (Ritchie, 2024). The blackout curtain deployment is also fairly predictable, occurring mostly at night in specific conditions and for extended periods. Predicting the openings of the windows at any given time is a much more difficult task, especially during summer, when the opening varies wildly and is continuously used to optimise the greenhouse climate. This

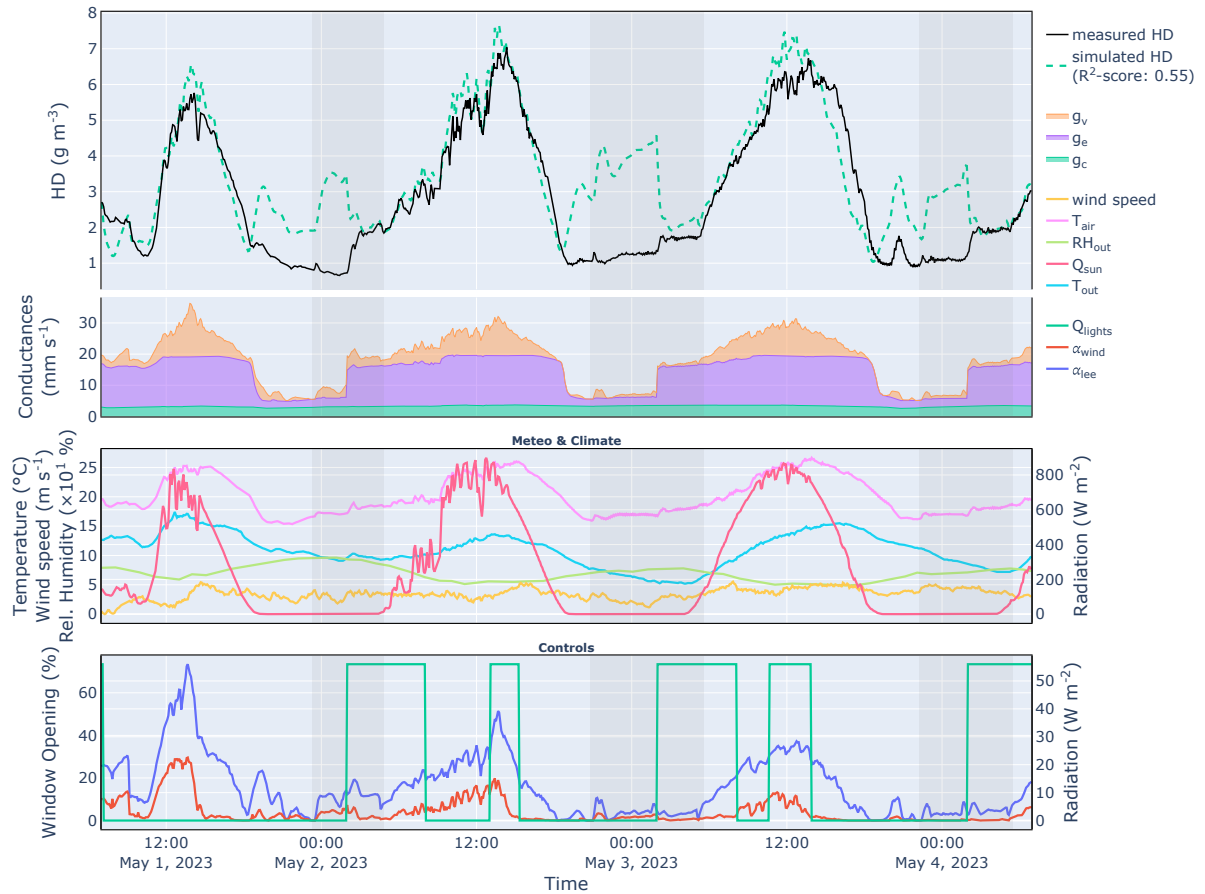


Figure 3.3: A simulation of the theoretical model of the humidity deficit plotted against the measured humidity deficit and the conductances of the three sub-models from 2023-05-01 07:00 up until 2023-05-04 07:00 with a semi-mature crop ($LAI \approx 2.3$), combined with weather, climate, and control conditions during the simulation. The shaded areas in the plot represent when the blackout curtain was deployed.

chapter and chapter 4 consider the optimistic scenario where all the necessary inputs to the theoretical model are measurements or otherwise known at simulation time. In chapter 6, a more realistic scenario is considered instead, making use of weather forecasts and a desired temperature profile inside the greenhouse. The behaviour of the climate controls is still considered to be known in advance, as predicting the window openings is not part of the scope of this work, but this is a notable limitation of the theoretical model and would require more research in the future.

Large temperature differences between the greenhouse air and the outside air and the accompanying large difference in (absolute) humidity levels, make it so that excess water vapour is more easily removed through condensation and ventilation. As a result, humidity generally poses less of an issue during winter (Goldammer, 2019). As a result, the worse performance during the winter season is deemed acceptable for the theoretical model. On the other hand, correct humidity prediction inside a greenhouse is especially important during the warmer months, when the crop transpires more and is more susceptible to water stress if the humidity levels drop too much through ventilation during daytime while being susceptible to condensation forming as a result of the difficulty in removing excess water vapour during nighttime. Furthermore, the humidity can be steered more actively through smart ventilation management if dangerously low or high humidity is anticipated. The theoretical model was found to perform well during the warmer spring and summer months, which agrees with the need for more accurate HD predictions for these months.

Given that the largest errors occur during the blackout curtain deployment, a mechanism for dealing with this issue would be quite important, either through the inclusion of the blackout curtain in the theoretical model or through some correction mechanism. Although the first option might be a viable

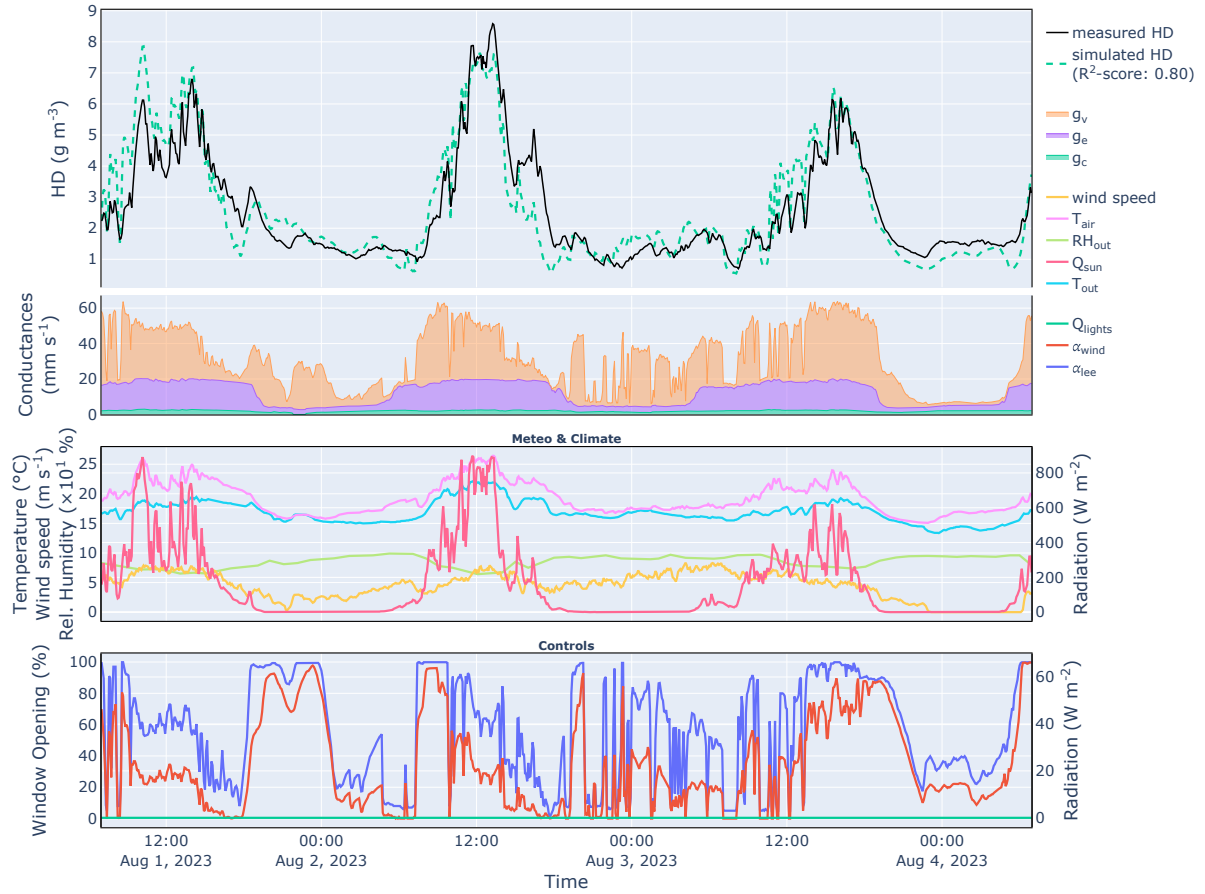


Figure 3.4: A simulation of the theoretical model of the humidity deficit plotted against the measured humidity deficit and the conductances of the three sub-models from 2023-08-01 07:00 up until 2023-08-04 07:00 with a mature crop ($LAI \approx 2.5$), combined with weather, climate, and control conditions during the simulation.

approach to tackling this issue, it introduces a lot of complexity to the theoretical model through extra state variables and fluxes. The latter option of using a correction mechanism is explored in chapter 4.

3.4. Conclusion

This chapter introduced a differentiable theoretical model for simulating the humidity deficit inside the greenhouse. The model is largely based on previous work by Bontsema et al. (2008), De Zwart (1996), Monteith and Unsworth (1990), Stanghellini and De Jong (1995), Van Beveren et al. (2015), and Vanthoor (2011) but required modifications to obtain decent results and to ensure stable and continuous gradients allowing for its use in chapter 4. Through experimental validation, a glimpse of the strengths and weaknesses of the theoretical model was given and discussed. A more in-depth analysis is required and provided in chapter 4. Except for during blackout curtain deployment, the theoretical model can adequately simulate the humidity deficit in the greenhouse even with limited knowledge about the greenhouse. From the experiments, we further conclude that a method for lowering the error induced during blackout curtain deployment is necessary. A particular weakness of using the theoretical model pertains to the need to know the openings of the different windows at any given time to simulate the humidity levels inside the greenhouse correctly; this falls out of the scope of this work.

4

Controlled Aphynity for Greenhouse Humidity Deficit Prediction

Integrating physics-based knowledge with machine learning techniques holds immense promise in enhancing our understanding and predictive capabilities in complex (bio-)physical systems such as greenhouses. In this chapter, we introduce a novel physics-informed model rooted in the Aphynity framework (Yin et al., 2021), which leverages PIML to estimate the parameters of the physical system and augment missing or correct incorrect physics. Section 4.1 introduces Controlled Aphynity (CA), a new method that combines the Aphynity framework (Yin et al., 2021) with the concept of Neural CDEs (Kidger et al., 2020) to incorporate data that arrives later into a universal differential equation, allowing for the model to adapt to changing dynamics, thereby making it suitable for application to the context of the greenhouse climate. Implementation details are provided to apply the CA method to the problem of forecasting the humidity deficit inside the greenhouse. In section 4.2, the newly proposed and implemented CA model is subsequently validated by forecasting humidity deficit levels inside the greenhouse across a complete crop-growing cycle. Finally, the validation results are discussed in subsection 4.2.6

4.1. Controlled Aphynity (CA)

In the following, we adapt the Aphynity method (Yin et al., 2021) for solving the inverse problem of controlled differential equations (Kidger et al., 2020). For a brief introduction to (neural) differential equations and Aphynity, the reader is referred to section 2.4. Additionally, a summary of the Aphynity framework is provided in Appendix B to aid the reader in understanding the next section. For an in-depth background and analysis of neural (controlled) differential equations, we refer the reader to Kidger (2022), which provides an excellent survey of the field.

We define a controlled differential equation similar to Kidger et al. (2020), with the notable exception of using the form $f(y(s), x(s))$ instead of $f(y(s)) dx(s)$. Although both approaches are very similar (Kidger, 2022), the first allows us to share the formulation of Equation (3.7).

First, let $T > 0$ and let $d_x, d_y \in \mathbb{N}$. Let $x : [0, T] \rightarrow \mathbb{R}^{d_x}$ be a continuous function with bounded variation. Let $f : \mathbb{R}^{d_y, d_x} \rightarrow \mathbb{R}^{d_y}$ be Lipschitz continuous. Finally, let $y_0 \in \mathbb{R}^{d_y}$.

A continuous path $y : [0, T] \rightarrow \mathbb{R}^{d_y}$ is said to solve a controlled differential equation, controlled or driven by x , if:

$$y(0) = y_0, \quad y(t) = y(0) + \int_0^t f(y(s), x(s)) \, ds \quad (4.1)$$

In our work, part of the physics is known; therefore, we decompose Equation (4.1) into a known physics part and an augmentation part, obtaining the following:

$$y(0) = y_0, \quad y(t) = y(0) + \int_0^t (f_p + f_a)(y(s), x(s)) \, ds \quad (4.2)$$

Similar to section 3.2, we must first make x continuous to allow for the use of an adaptive step-size solver. We now assume that we have a dataset $\mathcal{D} = \{(y_{k\Delta t}^{(i)}, x_{k\Delta t}^{(i)})_{0 \leq k \leq \lfloor T/\Delta t \rfloor}\}_{1 \leq i \leq B}$ of trajectories y and driving signals x with temporal resolution Δt and batch size B to perform mini-batch gradient descent. Here, we use an altered version of the interpolation function from section 3.2 where $x_{\mathcal{D}}^{(i)} = [0, \lfloor T/\Delta t \rfloor]^B \rightarrow \mathbb{R}^{B \times (d_x+1)}$ such that $x_{\mathcal{D}}^{(i)}(t) = (t, x_t^{(i)})$. This detail of representing the time in the input is important for the expressivity of the model (Kidger, 2022). The function $x_{\mathcal{D}}$, a natural cubic spline in our case, can then be used in Equation (4.2).

Next, both terms of the decomposition are parameterised, indicated by $f_p^{\theta_p}$ and $f_a^{\theta_a}$. The goal then becomes to find the parameters θ_p and θ_a that make up a decomposition $(f_p + f_a)$ that most closely approaches the true dynamics while minimising the role played by f_a . As such, we adapt the objective function from the Aphynity framework (Yin et al., 2021) to fit the controlled differential equation specification from Equation (4.2) and obtain the following:

$$\min_{\theta_p, \theta_a} \|f_a^{\theta_a}\| \quad \text{s.t.} \quad \forall i, \forall k, \quad y_{k\Delta t}^{(i)} = y_0^{(i)} + \int_0^{k\Delta t} (f_p^{\theta_p} + f_a^{\theta_a})(y(s), x_{\mathcal{D}}^{(i)}(s)) ds \quad (4.3)$$

This objective function can be optimised following the same adaptive constrained optimisation procedure as outlined in Yin et al. (2021).

4.1.1. Implementation

The model from section 4.1 was implemented in Python using Pytorch Lightning for optimal organisation of the code. For $f_a^{\theta_a}$, a sufficiently expressive model is necessary to be able to optimise over $\mathbb{R}^{d_y, d_x}$ (Yin et al., 2021). Having shown promising performance in approximating differential equations (Raissi et al., 2019), we represent f_a using a deep neural network, similar to Yin et al. (2021). Specifically, we use a relatively small MLP, allowing quicker training than bigger models with fewer data. Further, we slightly modified the optimisation procedure of Yin et al. (2021) to adapt it for the greenhouse humidity deficit forecasting context:

First, common augmentation losses $\|f_a^{\theta_a}\|$ were found to be in the order of 1×10^{-8} to 1×10^{-6} , whereas the trajectory losses, calculated using the mean squared error (MSE), are typically in the order of 0.1 to 10. Therefore, instead of multiplying the trajectory loss by a small λ , we multiply the augmentation loss by the inverse of that small λ instead. This trick allows us to use standard learning rates (i.e. 1×10^{-3}). Furthermore, λ_j should theoretically only be updated after looping over all batches multiple times to adhere to the exact formulation of the optimisation problem from Yin et al. (2021). This is because the prediction accuracy serves as a constraint on the minimisation of $f_a^{\theta_a}$. In practice, Yin et al. (2021) found that only looping over all batches once already yielded satisfactory results while significantly saving on computational cost (Yin, 2024). For this reason, we adopted this change, aiming to find a compromise between theoretical fidelity and computational efficiency in our implementation. The slightly updated version of the algorithm from Yin et al. (2021) is summarised in Algorithm 1, where τ_1 refers to the learning rate and τ_2 refers to the rate at which the weight of the augmentation loss is being decreased. λ_0 designates the initial weight of the augmentation loss in the total loss. For an in-depth explanation of these hyperparameters, the reader is referred to Appendix B.

Algorithm 1 CA optimisation procedure for Greenhouse Humidity Deficit Prediction

Initialisation: $\lambda_0 \geq 0, \tau_1 > 0, \tau_2 > 0$

```

1: for  $epoch = 1 : N_{epochs}$  do
2:   for  $batch = 1 : B$  do
3:      $\theta_{j+1} \leftarrow \theta_j - \tau_1 \cdot \nabla [\mathcal{L}_{MSE}(\theta_j) + \frac{1}{\lambda_j} \|f_a\|]$  ▷ where  $\theta_j = (\theta_p^j, \theta_a^j)$ 
4:   end for
5:    $\lambda_{j+1} \leftarrow \lambda_j + \tau_2 \cdot \mathcal{L}_{MSE}(\theta_{j+1})$ 
6: end for
```

Next, we make some context-specific alterations to the input vectors x_c and u from Equation (3.6). These alterations are only used by the augmentation network f_a . As stated in section 4.1, we require a representation of time to be included in the input of f_a for the expressivity of the model. For this

purpose, we add an input to x_c that denotes the progress in the crop cycle as a fraction of the total length, $t_{cycle} \in [0, 1]$. Next, to allow the network to correct for the blackout curtains, we add the deployment of the blackout curtain to u as a fraction of its maximum deployment, $\beta_{blackout} \in [0, 1]$. Combining the above, we obtain the following:

$$y = [-\mathcal{X}_{air}], \quad x_c = \begin{bmatrix} t_{cycle} \\ T_{out} \\ Q_{sun} \\ \mathcal{X}_{out} \\ T_{air} \\ w \end{bmatrix}, \quad u = \begin{bmatrix} \alpha_{lee} \\ \alpha_{wind} \\ Q_{lights} \\ \beta_{blackout} \end{bmatrix} \quad (4.4)$$

Same as in section 3.2, we combine the variables x_c and u to form x , but with the addition of using min-max scaling to scale all inputs, including y , to the unit scale. The minimum and maximum of each feature are taken from the entire dataset. We then create a natural cubic spline interpolation of the scaled x to obtain the required x_D , which is used to drive the CDE.

4.2. Model validation: Correcting for incomplete physics

As was already noted in subsection 3.3.5, the theoretical model of the humidity deficit from section 3.1 lacks any specification for blackout curtains and the model might further contain some inherent error due to simplifications and assumptions. Therefore, we expect to be able to improve the results from this theoretical model by using the augmentation network f_a from Equation (4.2). To test this hypothesis, an experiment compares three different uses of this augmentation network.

4.2.1. Data

For the experiment below, the same data as in section 3.3 was used for running the test simulations. The training data consists of a combination of recent data on the *current* (i.e. the crop being grown at the time of the simulation) crop and data on a similar *historical* crop. The detailed procedure used to construct the train set is outlined in Appendix E. The following experiments use *current* crop data from a crop grown from 2022-11-28 to 2023-10-03 and *historical* data from a similar tomato crop grown between 2021-09-06 and 2022-08-27.

For testing, the complete dataset of the current crop cycle was subdivided into 72-hour-long test windows. To ensure that sufficient training data for the *current* crop was available, the first test window starts at 2023-01-01. This results in a total of 89 test windows. 26 test windows take place during winter (due to the simulations starting after winter has begun and the short February month). Because the crop cycle starts and ends in autumn, only 3 test windows take place during autumn, 30 test windows occur during spring, and 30 test windows occur during summer.

4.2.2. Experimental Setup

In the following, we conducted simulations using three different settings. First, we analysed the performance of the theoretical model without any augmentation network f_a . As a result, the model behaved similarly to a standard theoretical model with parameter calibration using mini-batch gradient descent. Next, we added the augmentation network f_a to the model to evaluate its impact on the general performance of the model. Lastly, we included a mask to f_a so that it only augments the dynamics during the deployment of the blackout curtain, a period for which we previously found the theoretical model to be misspecified (section 3.3). This was done to test the effectiveness of f_a in correcting (known) missing physics.

To restrict the search space of the theoretical model parameters, only the parameters for the leaf area index (LAI) and the fraction of windows per greenhouse area (fr_{window}) from the theoretical model were set to be learnable through gradient descent. To aid in training, the initial value for the LAI was calculated according to a simple saturating heuristic function, based on discussion with experts, which assumes that the crop starts out at a low LAI of around $0.5 \text{ m}^2 \text{ m}^{-2}$ and reaches a maximum of around $2.5 \text{ m}^2 \text{ m}^{-2}$:

$$LAI_{initial} = 0.5 + 2 \cdot (1 - e^{-\frac{t}{60}}) \quad (4.5)$$

where:

- t represents the number of days from the start of the crop cycle up until the simulation date

For the other parameters, the (initial) values were set equal to the values from Table 3.1.

CA requires several hyperparameters to be set. For an explanation of these hyperparameters, the reader is referred to Appendix B. A small grid search was performed to find decent values for these hyperparameters. Following this grid search, the learning rate (τ_1) was set to 1×10^{-3} , a common default value that provided stable results. Both λ_0 and τ_2 were set to 1×10^{-6} as this value balanced the augmentation loss nicely with the trajectory losses, resulting in stable learning. Lastly, a cosine annealing learning rate schedule (Loshchilov & Hutter, 2017) was used to help avoid local minima during training.

The augmentation network f_a was represented by an MLP with 1 hidden layer of size 64 and a ReLU activation function applied to each layer except the last. This is similar to the architecture used by Yin et al. (2021), although a smaller hidden layer size was chosen to account for the smaller size of the training data for the experiments here. Nevertheless, the results indicated that the augmentation network of this limited size was already capable of learning. The masked setting of CA is implemented by multiplying the output of f_a with the state of the blackout curtain deployment (i.e. $0.95 * f_a$ for 95 % deployment).

The compute time for numerical integration of the theoretical model exhibited greater-than-linear scaling with respect to the simulation time. Therefore, to lower the computation time, the training data was divided into small samples of 2-hour long windows with a 1-hour stride to provide overlapping context across training samples. The samples were randomly shuffled and split into a train and validation set using a train/validation split of 80/20 %. The training was performed using mini-batches of 128 samples (i.e. $B = 128$) for stable gradient calculations, and a maximum of 30 epochs was used with early stopping after 5 epochs without improvement to the validation loss to keep the compute time manageable. Each simulation, apart from the simulation example in subsection 4.2.4, was performed 3 times with different random seeds to limit the effects of randomness on the results.

4.2.3. Determination of model performance

Model performance was quantitatively evaluated using the metrics described in subsection 3.3.3.

4.2.4. A simulation example

The following demonstrates an example simulation using the CA model over a 72 h time span to visually inspect the impact of the different settings for the augmentation network f_a on the predictions. The simulation runs from 2023-05-01 07:00 AM to 2023-05-04 07:00 AM. The simulation example is run over the same time span as one of the examples of section 3.3 to allow for a comparison of the results. The experiment results and conditions of the same scenario with manual calibration from section 3.3 can be found in Figure 3.3. This scenario was chosen due to the large and distinct error visible during the blackout curtain deployment. The simulation results are shown in Figure 4.1. The shaded areas in the plot represent when the blackout curtain was deployed. Furthermore, the aggregated augmentation losses per hour for the models that use f_a are also shown.

The results show that the no- f_a model performs similarly to the manually calibrated theoretical model, with the large error during blackout curtain deployments still visible. Both masked- f_a and full- f_a improve the predictions considerably during blackout curtain deployment. Moreover, full- f_a also seems to improve the predictions when the blackout curtain is not deployed, likely indicating other, albeit smaller, misspecifications in the theoretical model. These results coincide with the incurred augmentation penalties for both models. Masked- f_a incurs a similar large augmentation loss during the blackout curtain deployment as full- f_a . However, the augmentation loss for the masked- f_a is zero elsewhere, whereas the full- f_a incurs smaller augmentation losses over the entire prediction window, incurring an accompanying interpretability penalty. As a side note, one might gather from this result that the augmentation loss of the full- f_a model could be used to find and improve misspecifications in a theoretical model.

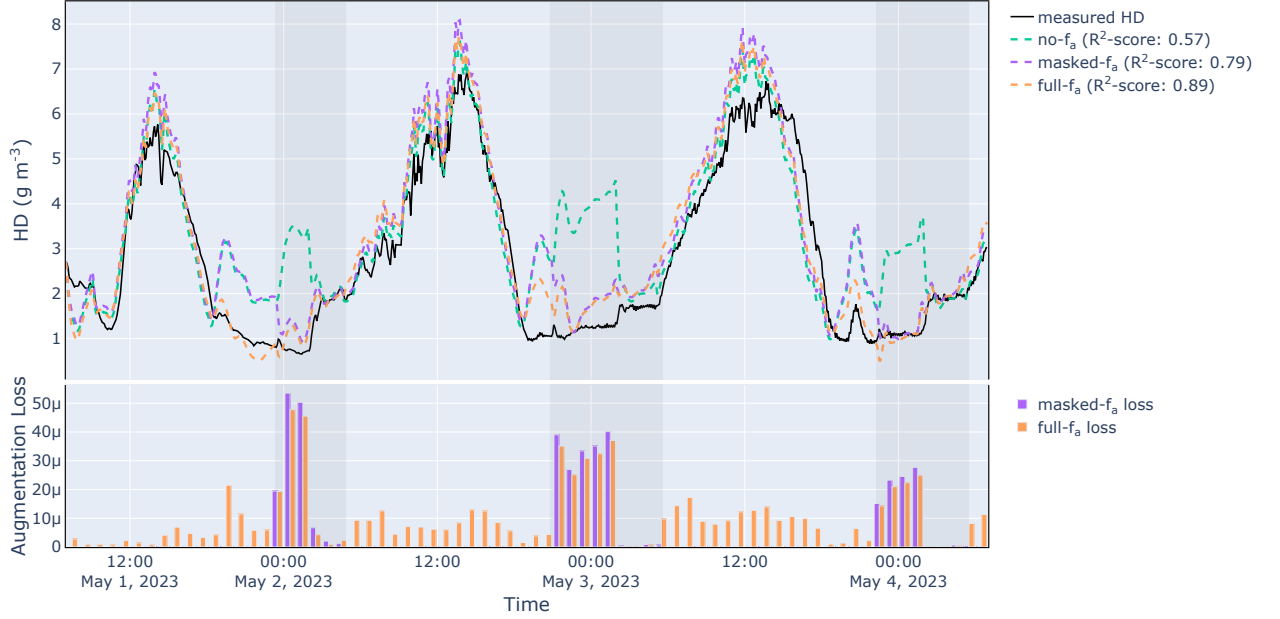


Figure 4.1: The predictions of the HD for the no- f_a , masked- f_a , and full- f_a models vs. the measured HD every 5 minutes from 2023-05-01 07:00 up until 2023-05-04 07:00 together with the sum of augmentation losses of f_a incurred per hour for the models that make use of f_a . The shaded areas in the plot represent the times during which the blackout curtain was deployed.

4.2.5. Results

We analyse the results of running the CA model with the three settings below. The results are summarised in Table 4.1. For all results, the mean and standard deviations are reported. The best scores, calculated using a t-test with the common statistical significance threshold $\alpha = 0.05$, were made bold for each metric and each season. We consecutively analyse the performance of each method according to the metrics from subsection 3.3.3. Due to the difference in performance across different seasons of the theoretical model that was found in section 3.3, the results for these experiments were also split up based on season, with an overall summary of the results added as well. Given the insufficient amount of evidence to gain any useful insights into the performance for the autumn season, these results have been left out of Table 4.1 and the analysis; however, the autumn results are included in the overall metrics.

Although slight variations in performance across metrics can be observed, these all follow a similar trend. Both masked- f_a and full- f_a show significant performance improvements across all metrics compared to the theoretical model without f_a . Full- f_a performs better than or similar to masked- f_a on all occasions across all metrics, but this comes at the cost of a bigger hit on interpretability as a larger proportion of the dynamics is modelled through f_a , as measured through its norm ($\|f_a\|^2$). In the following, the results are reported for each season individually. Afterwards, subsection 4.2.6 discusses the results and section 4.3 concludes our findings.

For winter, both masked- f_a and full- f_a settings offer substantial improvements in performance according to all three metrics, both in mean error/score and variance. Both methods, on average, show improvements of approximately 37 %, 18 %, and 290 % in RMSE, rRMSE, and R^2 -scores respectively. The latter was mainly caused by the extremely poor performance of the theoretical model in winter due to extensive blackout curtain deployment. No significant difference was found between the RMSE of the masked- f_a and full- f_a settings. The rRMSE was found to be slightly lower for full- f_a compared to masked- f_a . However, the R^2 -scores for all three methods are still lower than 0, indicating that the model is worse than predicting the mean of the humidity deficit. As expected, in the full- f_a setting, f_a does have a larger impact on the simulated trajectory of the humidity deficit as measured through its norm than in the masked- f_a .

For spring, the performance of the no- f_a model has the lowest RMSE of any season. Nevertheless, both masked- f_a and full- f_a improve results significantly. For the RMSE, masked- f_a shows an average

Table 4.1: Performance of CA with different settings for the augmentation model (f_a). The 'None' method makes no use of f_a . The 'Masked' method only utilises f_a during blackout curtain deployment. The 'Full' method always utilises f_a . Due to large differences in performance across seasons, these have been split up, showing overall results at the bottom. The sample size n is displayed for each season, and the best-scoring methods (calculated using a t-test with $\alpha = 0.05$) are made bold. The metrics displayed (mean $\pm$ std) are the Root Mean Squared Error (RMSE), Relative Root Mean Squared Error (rRMSE), and coefficient of determination (R^2). For the models that use f_a , the norm $\|f_a\|$ is also displayed to show the impact of f_a . This value is calculated over the periods where f_a is actively used (i.e. only when the blackout curtain is deployed for the masked setting).

Season	Use of f_a	RMSE (g m^{-3})	rRMSE	R^2	$\ f_a\ ^2$
Winter ($n = 78$)	None	1.60 $\pm$ 0.57	0.75 $\pm$ 0.34	-12.90 $\pm$ 28.57	n/a
	Masked	1.01 $\pm$ 0.33	0.46 $\pm$ 0.17	-4.61 $\pm$ 12.41	4.96e-7 $\pm$ 3.26e-7
	Full	0.94 $\pm$ 0.33	0.43 $\pm$ 0.18	-4.19 $\pm$ 11.19	6.48e-7 $\pm$ 2.66e-7
Autumn ¹	—	—	—	—	—
Spring ($n = 90$)	None	1.29 $\pm$ 0.28	0.45 $\pm$ 0.14	0.39 $\pm$ 0.47	n/a
	Masked	0.96 $\pm$ 0.31	0.32 $\pm$ 0.06	0.73 $\pm$ 0.12	3.58e-7 $\pm$ 2.45e-7
	Full	0.87 $\pm$ 0.29	0.29 $\pm$ 0.08	0.77 $\pm$ 0.12	1.02e-6 $\pm$ 4.5e-7
Summer ($n = 90$)	None	1.45 $\pm$ 0.52	0.37 $\pm$ 0.10	0.61 $\pm$ 0.20	n/a
	Masked	1.45 $\pm$ 0.52	0.37 $\pm$ 0.10	0.61 $\pm$ 0.20	4.16e-7 $\pm$ 2.19e-6
	Full	1.26 $\pm$ 0.31	0.33 $\pm$ 0.06	0.70 $\pm$ 0.13	5.33e-6 $\pm$ 4.42e-6
Overall ($n = 267$)	None	1.46 $\pm$ 0.49	0.52 $\pm$ 0.26	-3.41 $\pm$ 16.54	n/a
	Masked	1.17 $\pm$ 0.48	0.39 $\pm$ 0.13	-0.88 $\pm$ 7.10	4.40e-7 $\pm$ 1.31e-6
	Full	1.04 $\pm$ 0.36	0.35 $\pm$ 0.13	-0.70 $\pm$ 6.43	2.77e-6 $\pm$ 3.84e-6

26 % improvement and full- f_a a 33 % improvement. For the rRMSE, masked- f_a shows an average 28 % improvement and full- f_a a 36 % improvement. Finally, masked- f_a and full- f_a show an average 87 % and 97 % improvement in R^2 -scores, respectively. Both methods also reduce the variance, measured through the standard deviation, of the R^2 -score by around 66 %.

In summer, no significant difference in performance was found between no- f_a and masked- f_a . This is to be expected because the blackout curtains are barely used during the summer months. The norm of f_a for the masked- f_a model has a high variance (an order of magnitude larger than the average); this is a product of the limited activation time as a result of the masking, coupled with the fact that the regularisation that tries to minimise the impact of f_a is calculated over the entire trajectory. Because the f_a is masked for most of the time, the loss will inherently be lower, allowing the f_a to play a larger role in the dynamics when it is active. The full- f_a model showed a 13 % improvement with respect to the RMSE, a 10 % improvement in the rRMSE, and a 14 % improvement in the R^2 -score. A relatively large norm of f_a was found for the full- f_a model.

4.2.6. Discussion

From the results, lots of interesting insights can be gathered. We start by comparing the results to related work on simulating humidity levels inside the greenhouse. In general, it is difficult to compare our model to related work on theoretical models for simulating the humidity inside the greenhouse as it requires the same metrics to be used under similar weather conditions and in similar locations for similar lengths of time. Moreover, models to simulate the humidity deficit inside the greenhouse are rare. Instead, most models aim to simulate the absolute humidity or the relative humidity. Nevertheless, we will discuss some results from related work here, including work from which the theoretical model (chapter 3) was composed. Van Beveren et al. (2015) reported an RMSE for the simulation of absolute humidity of 1.76 g m^{-3} , indicating an 18 % improvement from our theoretical model without any augmentation. This comparison is, however, not entirely fair for a couple of reasons:

First, our simulation examples took measured temperature as an input rather than jointly simulating temperature with humidity; an incorrect temperature can have an adverse effect on the predictions due to the interrelationship between humidity and temperature. The differences in the ventilation models

¹Due to the start- and end-date of the crop cycle, not enough data is available

and prediction timescales could also explain the performance differences. Su et al. (2018) reported an RMSE of 1.68 g m^{-3} for their absolute humidity simulation model based on the model from Tap (2000), further observing that in some cases, large simulation errors occur. These results are similar to the ones obtained here. Once again, the timescale for each simulation is shorter in this work; this might impact the resulting comparison.

With the use of the augmentation model f_a , the RMSE drops to well within the benchmark of 1.5 g m^{-3} , across most seasons, as set out in subsection 3.3.3. This leads us to believe that CA is a promising method for correcting model misspecifications, augmenting missing physics, and learning unknown theoretical parameters of complex non-linear theoretical models. When comparing the performance between the masked- f_a model and the full- f_a model, one might find that masked- f_a can already provide sufficient improvement of missing or incorrect physics and provide better interpretability than the full- f_a model.

Simulation performance when using the augmentation model increases for seasons in which the theoretical model performs poorly and for seasons in which it performs well. Overall, it can be observed that the better the performance of the theoretical model itself, the better the final quality of the predictions made by the CA model with augmentation. Therefore, more complex models, such as from De Zwart (1996) and Vanthoor (2011), could and should be studied to test this hypothesis more thoroughly. One of the advantages of the developed CA model is that it is easily extended to accommodate different models.

As a result of the tuning performed for this experiment, some interesting findings were gathered. For example, we found that λ_0 , the initial weight of the augmentation loss, should be chosen such that the weighted augmentation loss is similar to or larger than the expected trajectory loss at the start of training. Otherwise, the augmentation loss would be ignored in training. Moreover, τ_2 , the parameter guiding the rate of change of λ_j , should be at least an order of magnitude lower than λ_0 to avoid the augmentation loss becoming insignificant too quickly. A more thorough ablation study could further result in extra insights regarding the tuning of these hyperparameters and could lead to improved results. This was, unfortunately, not possible within the limited time constraints of this work.

Furthermore, most parameters of the theoretical model were estimated beforehand and not learned by the model to reduce the search space. If more of these parameters, such as the physical dimensions of the greenhouse, are known in advance, model performance is likely to go up. One could then focus on learning the hard-to-measure variables, such as the LAI.

Another important limitation that was already discussed but deserves more attention is the need for information that might not be available about the future states of the climate control equipment, such as the black-out curtains, window apertures, and artificial lighting. Both black-out curtains and artificial lighting are fairly predictable in their deployment, as they are mostly active at night and for extended periods. Window apertures are much more difficult to predict, as these, especially during the hotter months, constantly change in response to outdoor and indoor conditions to maintain the optimal indoor climate. The impact of incorrect predictions of window apertures on the quality of the model predictions should need further investigation. Otherwise, different methods of modelling the ventilation rate in the greenhouse could be investigated, such as through the differences in enthalpy inside and outside the greenhouse, although this method comes with its own downsides (van Weel et al., 2018).

4.3. Conclusion

This chapter introduced *Controlled Aphynity*, a novel PIML model based on the Aphynity framework (Yin et al., 2021) and insights from neural controlled differential equations (Kidger et al., 2020). The newly proposed CA model allows for the augmentation of incomplete prior knowledge of a dynamical system with data-driven machine learning models for complex forecasting tasks *while being adaptable to changing dynamics due to forces acting on the system*. As a practical use-case for analysing the impact of the CA model, it was implemented to forecast the humidity deficit inside the greenhouse, using the theoretical model from chapter 3 as prior knowledge.

Different settings for correcting the incomplete physics and CA's general forecasting performance over a complete crop-growing cycle were experimentally explored and analysed. The results show that

the CA model can learn the unknown physical parameters and successfully augment the theoretical model's missing or incorrect physics specifications. However, the performance of CA was found to be correlated to the performance of the theoretical model. At the cost of a little performance, only augmenting physics during known incorrect physics specifications, during blackout deployment in our case, is already quite successful and enhances the interpretability. Nevertheless, the full- f_a setting for CA will be used in the rest of this work as this setting obtained the best results and better shows the limits of the model's capabilities.

It is important to have a good estimate of the forces acting on the system over time to adapt to changing dynamics. The experiments in this work analysed an optimistic scenario where measurements are used instead, chapter 6 presents experiments using forecasted meteorological data instead. A potential shortcoming of the CA model arises when forecasts of certain forces are hard to obtain, such as the forces originating from the control equipment in the greenhouse. Although the climate control equipment states were assumed to be known in advance here, this might not be the case in the real world. Obtaining accurate forecasts (i.e. for the window openings in summer) might be a challenge, and inaccurate forecasts could have an adverse effect on the performance of the CA model. Inspecting the impact of incorrect forecasts and tackling this challenge is left open for future work.

5

CA-ML Ensemble for Greenhouse Humidity Deficit Prediction

The CA model from section 4.1 and ML models both offer distinct advantages and limitations due to their different approaches to the problem. As a result, it is worthwhile to explore a combination of both methods through ensemble learning. Ensemble methods work by combining multiple learning algorithms to obtain better predictive performance than could be obtained from any of the individual learning algorithms. In the following, we explore three different ensemble models. Section 5.1 discusses a model that makes predictions by averaging the predictions of the CA and ML models. Section 5.2 examines the possibility of training a ML model on the residuals of the CA model that then combines the CA model and the residual predictions to obtain the final prediction. Section 5.3 explores the use of the predictions from the CA model as a regression variable in a ML model. Next, a comparative analysis is performed in section 5.4 that systematically compares the different ensemble methods to pinpoint their similarities, differences, strengths, and weaknesses. Section 5.4.5 discusses the results and section 5.5 concludes the chapter.

5.1. Uniform Average Ensemble

The first ensemble method that will be explored takes a uniform average over the predictions of the CA model and a ML model. The algorithm is summarised in Algorithm 2. Additionally, a flow diagram of the algorithm is shown in Figure 5.1. The advantages of this approach include the ability to train and predict both models simultaneously and its ease of implementation. A downside of this approach is the lack of any mechanism to weigh the models' predictions based on, for example, the uncertainty of the predictions. Nevertheless, if the uniform average ensemble performs better than any of the constituent models individually, this might demonstrate that the strengths and weaknesses of CA and ML models are complementary.

Algorithm 2 Uniform Average Ensemble

Input: $\mathcal{D}_{train} = \{(y_{k\Delta t}^{(i)}, x_{k\Delta t}^{(i)})_{0 \leq k \leq \lfloor T_{train}/\Delta t \rfloor}\}_{1 \leq i \leq B}$, $\mathcal{D}_{test} = (x_{k\Delta t})_{0 \leq k \leq \lfloor T_{test}/\Delta t \rfloor}$

- 1: Train CA on $\mathcal{D}_{train}$ using Algorithm 1
- 2: Train ML model on $\mathcal{D}_{train}$
- 3: Using the trained CA model, predict the humidity deficit ($\hat{y}_{CA}$) on $\mathcal{D}_{test}$
- 4: Using the trained ML model, predict the humidity deficit ($\hat{y}_{ML}$) on $\mathcal{D}_{test}$

Output: $\frac{1}{2}(\hat{y}_{CA} + \hat{y}_{ML})$

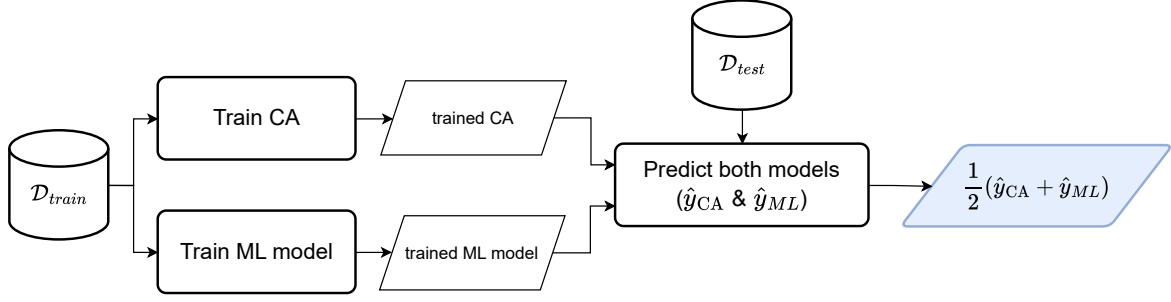


Figure 5.1: Flow diagram of the uniform average ensemble method. The blue-shaded parallelogram represents the final output.

5.2. Residual Ensemble

The next method that will be explored uses a ML model to predict the residuals of the CA model instead. This algorithm is summarised in Algorithm 3. Additionally, a flow diagram of the algorithm is shown in Figure 5.2. By first making predictions using the CA model and subtracting the result from the measured HD, another ML algorithm can use the residual (i.e. error) as the target that is to be predicted. The reasoning for trying this method is as follows: if one considers the augmentation network f_a from CA to be a local correction mechanism for the dynamics at any specific time, the ML model might function as a global correction mechanism, considering the entire prediction window context. If the residual ensemble performs better than its constituent models individually, this might point to some global structural error in the theoretical part of the CA model that could not be corrected locally.

Algorithm 3 Residual Ensemble

Input: $\mathcal{D}_{train} = \{(y_{k\Delta t}^{(i)}, x_{k\Delta t}^{(i)})_{0 \leq k \leq \lfloor T_{train}/\Delta t \rfloor}\}_{1 \leq i \leq B}$, $\mathcal{D}_{test} = (x_{k\Delta t})_{0 \leq k \leq \lfloor T_{test}/\Delta t \rfloor}$

- 1: Train CA on $\mathcal{D}_{train}$ using Algorithm 1
- 2: Using the trained CA model, predict the humidity deficit ($\hat{y}$) on $\mathcal{D}_{train}$
- 3: Create residual train set: $\tilde{\mathcal{D}}_{train} \leftarrow \{(y_{k\Delta t}^{(i)} - \hat{y}_{k\Delta t}^{(i)}, x_{k\Delta t}^{(i)})_{0 \leq k \leq \lfloor T_{train}/\Delta t \rfloor}\}_{1 \leq i \leq B}$
- 4: Train ML model on $\tilde{\mathcal{D}}_{train}$
- 5: Using the trained CA model, predict the humidity deficit ($\hat{y}_{CA}$) on $\mathcal{D}_{test}$
- 6: Using the trained ML model, predict the residual ($\tilde{y}_{Residual}$) on $\mathcal{D}_{test}$

Output: $\hat{y}_{CA} + \tilde{y}_{Residual}$

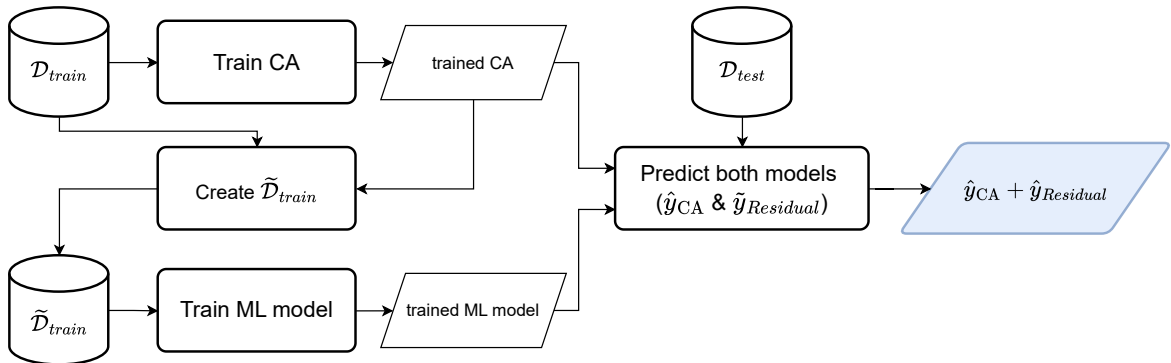


Figure 5.2: Flow diagram of the residual ensemble method. The blue-shaded parallelogram represents the final output.

5.3. Regression Ensemble

The final ensemble method presented here adds the prediction made by the CA model as an input feature to a ML model. This algorithm is summarised in Algorithm 4. Additionally, a flow diagram of the algorithm is shown in Figure 5.3. Letting a ML model learn the importance of the prediction made by the CA model is a straightforward method for allowing improvements to the performance while limiting the risk of adverse impact on performance. If, during training, the performance of the CA model is too low, then the ML model can disregard its information, leading to a score that is no worse than without the extra information. Given the extra information, one might also assume that less training data is required for adequate model performance.

Algorithm 4 Regression Ensemble

Input: $\mathcal{D}_{train} = \{(y_{k\Delta t}^{(i)}, x_{k\Delta t}^{(i)})_{0 \leq k \leq \lfloor T_{train}/\Delta t \rfloor}\}_{1 \leq i \leq B}$, $\mathcal{D}_{test} = (x_{k\Delta t})_{0 \leq k \leq \lfloor T_{test}/\Delta t \rfloor}$

- 1: Train CA on $\mathcal{D}_{train}$ using Algorithm 1
- 2: Using the trained CA model, predict the humidity deficit ($\hat{y}$) on $\mathcal{D}_{train}$
- 3: Using the trained CA model, predict the humidity deficit ($\hat{y}_{CA}$) on $\mathcal{D}_{test}$
- 4: Create augmented train set: $\hat{\mathcal{D}}_{train} \leftarrow \{(y_{k\Delta t}^{(i)}, [x_{k\Delta t}^{(i)}, \hat{y}_{k\Delta t}^{(i)}])_{0 \leq k \leq \lfloor T_{train}/\Delta t \rfloor}\}_{1 \leq i \leq B}$
- 5: Create augmented test set: $\hat{\mathcal{D}}_{test} \leftarrow (x_{k\Delta t}, \hat{y}_{k\Delta t}^{CA})_{0 \leq k \leq \lfloor T_{test}/\Delta t \rfloor}$
- 6: Train ML model on $\hat{\mathcal{D}}_{train}$
- 7: Using the trained ML model, predict the humidity deficit ($\hat{y}_{ML}$) on $\hat{\mathcal{D}}_{test}$

Output: $\hat{y}_{ML}$

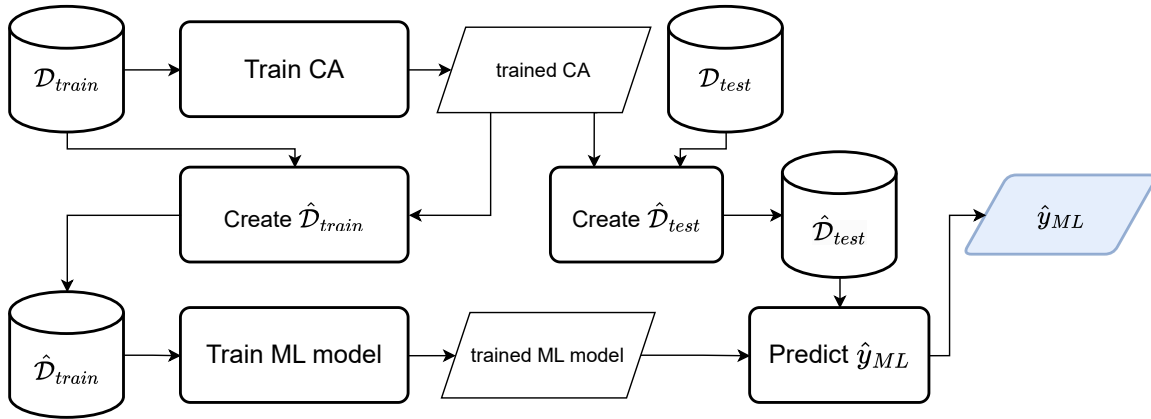


Figure 5.3: Flow diagram of the regression ensemble method. The blue-shaded parallelogram represents the final output.

5.4. Comparative Analysis

5.4.1. Data

The following experiments use the same train and test data as those from section 4.2. We refer the reader to subsection 4.2.1 for more information.

5.4.2. Experimental setup

In the following, the CA model refers to the full- f_a setting of CA. The experimental set-up regarding the training and prediction of the CA model is the same as described in subsection 4.2.2. Similarly, all experiments are run three times with different seeds to account for random variations. Each ensemble method uses the CA model and a well-known classical ML model. The implementation details regarding the ML model are outlined in Appendix F. The same ML model is also used to provide a baseline in the experiments; these results are referred to as “ML baseline”.

5.4.3. Determination of model performance

Model performance was quantitatively evaluated using the metrics described in subsection 3.3.3.

5.4.4. Results

The results of the comparative analysis are summarised in Table 5.1. Similar to before, a breakdown of the results between seasons is also provided. Overall, when using measurement data, the performance of the ensemble methods does not outperform the baseline model in a statistically significant matter. Only the regressor ensemble does slightly better or equally well across all seasons and metrics. Further, the residual method performs worst overall, whereas the average model performs similarly to the baseline across all metrics except for the R^2 -score. However, the lower R^2 -score can be explained by the average ensemble's poor performance during the winter.

Similar to subsection 4.2.5, all models perform best during spring. Moreover, the performance increase from the regressor ensemble compared to the ML baseline is also the largest for spring, with a more modest improvement in summertime. Regarding the RMSE, the regressor ensemble method achieves a 19 % improvement in spring, a 9 % improvement in summer, and no significant improvement in winter. Regarding rRMSE, the regressor ensemble shows a 15 % improvement over the ML baseline in spring while showing similar results for the other seasons. Finally, the R^2 -score for the regressor ensemble improves slightly during spring- and summertime compared to the ML baseline with 3 % and 5 %, respectively. Notably, the average ensemble has a similar performance increase compared to the ML baseline for all three metrics as the regressor ensemble in the summertime.

Overall, the regressor ensemble performs best in terms of RMSE and rRMSE with a 9 % and 13 % improvement over the ML baseline, respectively. R^2 -scores are similar for both the regressor ensemble and the ML baseline, with the average and residual ensembles performing worse, as stated before.

Table 5.1: Performance comparison of a ML baseline model and three different methods of CA Ensembles: Average, Residual, and Regressor. The average ensemble method predicts the uniform average of the CA and the ML baseline models. The residual ensemble predicts the residuals between the CA predictions and the targets. The regressor ensemble adds the CA predictions as a regression variable in the ML baseline model and uses the latter's predictions. The table has been split up to show the performance across seasons, with the overall results at the bottom. The sample size n is displayed for each season, and the best-scoring methods (calculated using a t-test with $\alpha = 0.05$) are made bold. The metrics displayed (mean $\pm$ std) are the Root Mean Squared Error (RMSE), Relative Root Mean Squared Error (rRMSE), and coefficient of determination (R^2).

Season	Model	Ensemble method	$RMSE$ (g m^{-3})	$rRMSE$	R^2
Winter ($n = 78$)	CA-ML	Average	0.61 ± 0.24	0.28 ± 0.11	-1.26 ± 5.25
		Residual	0.65 ± 0.17	0.29 ± 0.06	-0.73 ± 2.95
		Regressor	0.51 ± 0.20	0.23 ± 0.07	-0.20 ± 2.21
	ML baseline	—	0.53 ± 0.20	0.23 ± 0.07	-0.20 ± 2.02
Autumn ¹	—	—	—	—	—
Spring ($n = 90$)	CA-ML	Average	0.60 ± 0.24	0.20 ± 0.05	0.90 ± 0.05
		Residual	0.71 ± 0.37	0.23 ± 0.07	0.86 ± 0.08
		Regressor	0.50 ± 0.23	0.16 ± 0.05	0.93 ± 0.03
	ML baseline	—	0.62 ± 0.47	0.19 ± 0.05	0.90 ± 0.06
Summer ($n = 90$)	CA-ML	Average	0.89 ± 0.20	0.24 ± 0.06	0.84 ± 0.08
		Residual	1.10 ± 0.29	0.28 ± 0.06	0.77 ± 0.12
		Regressor	0.90 ± 0.21	0.24 ± 0.07	0.83 ± 0.15
	ML baseline	—	0.99 ± 0.30	0.26 ± 0.09	0.79 ± 0.19
Overall ($n = 267$)	CA-ML	Average	0.73 ± 0.31	0.24 ± 0.08	0.24 ± 2.99
		Residual	0.84 ± 0.36	0.27 ± 0.07	0.28 ± 1.96
		Regressor	0.67 ± 0.32	0.21 ± 0.08	0.56 ± 1.29
	ML baseline	—	0.74 ± 0.44	0.24 ± 0.09	0.53 ± 1.20

¹Due to the start- and end-date of the crop cycle, not enough data is available

5.4.5. Discussion

When the results of the different ensemble methods are compared to the CA model results from subsection 4.2.5 and the ML baseline, a few notable observations arise. First, the ML baseline, and by extension, the ensemble methods, significantly outperform the results from the CA model across each season. The largest performance differences are found in winter, whereas the smallest are in summer. This closely aligns with the CA model's performance trend over these seasons. Nevertheless, using the CA model in the regressor ensemble shows a performance increase compared to the ML baseline in every season except for winter. The CA model's performance was particularly poor during winter, so, logically, the information from the CA model predictions for winter is of little help in improving results within an ensemble setting. This explains why the regressor ensemble's score was no better than the baseline, as it can disregard the unhelpful information. Meanwhile, the average ensemble's score was poorer, as it uses 50 % of the poor CA predictions regardless. As stated before, the regressor ensemble seems to strike a good balance between improving results during periods where the CA model performs well while demonstrating no visible performance decrease in periods where the CA model performs poorly.

Additionally, from the results in the summertime, we might observe that the average ensemble also outperforms the ML baseline. Although more experimentation would be required for confirmation, this might hint towards the complementary nature between theoretical or PIML models and pure ML models. However, the performance of the average ensemble varies greatly across seasons, with a larger risk of worsening performance. As a result, the average ensemble might not be the best method for increasing the speed of adoption. A way to mitigate this risk is using Bayesian Model Averaging (BMA) (Hoeting et al., 1999). Instead of taking the uniform average over the results, BMA uses a weighted average based on the uncertainty in each model's predictions. Given that many ML models already allow for uncertainty quantification out-of-the-box, one would only need to add some form of uncertainty quantification to the CA model, a feature that is out-of-scope for this work.

Lastly, the results show that the residual ensemble method performs worse across every season and all metrics. From this, we might conclude that the hypothesis from Algorithm 3 is not likely to be true; namely, there is not likely to be some underlying structure, i.e. seasonality, in the residuals of the CA model predictions that could be effectively modelled by a ML model such as SARIMAX or Prophet.

Again, an improved theoretical model of the greenhouse dynamics would be expected to increase the performance of the CA model predictions and, by extension, the ensemble methods as well. Similarly, more advanced ensemble methods could be pursued to improve the results further. Nevertheless, adding the CA model predictions as a regression variable was effective and the most reliable method of incorporating prior physics knowledge into a pre-existing ML model while minimising the risk of model performance deterioration during periods where the prior physics knowledge is insufficient or incorrect.

5.5. Conclusion

This chapter explored three methods for using CA in an ensemble setting with a ML model. First, a uniform average ensemble method was proposed and implemented, which predicts the average of the CA and ML models to analyse the possible synergies or complementary behaviour of PIML and ML models. Second, a residual ensemble method was proposed and implemented, which used the ML model to predict the residual of the CA model predictions. This method allowed us to explore whether there is some global structural error that is not easily corrected by the augmentation network f_a , which functions at the local level, i.e. per time step. Third, a regression ensemble method was proposed and implemented, which uses the CA prediction outputs as inputs to the ML model, allowing it to decide the usefulness of the CA predictions. This third method was explored to limit the risk of adverse impact on performance.

Next, a comparative analysis of the three methods and a ML baseline showed that the uniform average ensemble performed better than the ML baseline at times when the CA model performed well and poor if not. An interesting opportunity for future work would be to explore what would happen to the synergies and performance of the uniform average ensemble method when varying the quality of the prior knowledge used in the CA model. Furthermore, investigating BMA to allow for weighing the results of each model based on uncertainty would be interesting to explore and could likely reduce the variance

in the results obtained here.

The residual ensemble's performance was poorer than the ML baseline across every season and metric. This provided strong evidence for the lack of any structural residual that is easier to predict than the humidity deficit itself for the ML model.

The regressor ensemble was found to perform evenly or better than the ML model across seasons and metrics, confirming that this method is a good way to limit the risk of adverse performance impacts while still allowing for performance improvements where the CA model performed well. As the regressor ensemble was the best-performing ensemble method, chapter 6 explores this method further by comparing it against the ML baseline and plain CA models under more realistic greenhouse scenarios and across three important performance categories: physical adherence, performance under varying amounts of available training data, and performance on outliers in the form of extreme conditions.

6

Case Study: Realistic Greenhouse Scenarios

This chapter provides a case study that experimentally compares the physics-informed CA model and the regressor ensemble against a ML baseline for multiple realistic greenhouse scenarios. Each model will be evaluated according to its performance in these scenarios.

In the previous chapters, each model had access to the measured climate and meteorological data for simulation. In practice, this information is not available at the time of simulation. Instead, forecasts of the meteorological data are often used. Importantly, temperatures inside the greenhouse are also not known ahead of time. However, the climate computer actively steers the temperature in the greenhouse towards a desired temperature profile that is often decided upon ahead of time. As such, the models will still be trained on measurements, but they will only have access to meteorological forecasts and the desired temperature profile during simulation.

Given this more challenging setting, an overall comparison between the performance of the CA, CA-ML regressor ensemble, and ML baseline models is made in subsection 6.4.1. Next, subsection 6.4.2 compares the models' physical adherence in the form of the rate of "un-physical" negative humidity deficit predictions. Each model's performance is then tested with varying amounts of available training data in subsection 6.4.3 to compare each model's capability of dealing with scarce data. The performance of the models for outliers and extreme meteorological/climatic conditions is compared in subsection 6.4.4 to gain insight into each model's ability to deal with barely seen circumstances. The results are discussed in section 6.5, while section 6.6 concludes the chapter.

6.1. Data

For all the experiments performed in this chapter, the same 89 test windows of 72 hours were used, except for the experiments regarding the performance for outliers. Similarly, the same training data selection procedure described in subsection 4.2.1 and summarised in Figure E.1 is used for the following experiments. However, some differences are worth mentioning. The first difference relates to the historical crop being used; a tomato crop grown between 2019-10-07 and 2020-09-12 is used here instead to mimic the setting used in a past production environment for which negative humidity deficit predictions and performance on outliers were known issues of using a ML model. Next, the training data no longer uses measurements for the meteorological conditions from the local meteorological equipment from the greenhouse. Instead, the measurements from a nearby weather station are used as this weather station is also the location from which hourly meteorological forecasts are available. These hourly forecasts are used in the test set to provide meteorological conditions. Each meteorological variable is linearly interpolated to obtain the values at a 5-minute interval, except for the global radiation. Global radiation is supplied as the expected sum of global radiation per hour per area (J cm^{-2}) and subsequently converted to a unit of power over each 5-minute interval (W m^{-2}). Moreover, due to the absence of wind speed data in the available forecast data, we resorted to the measured values for wind

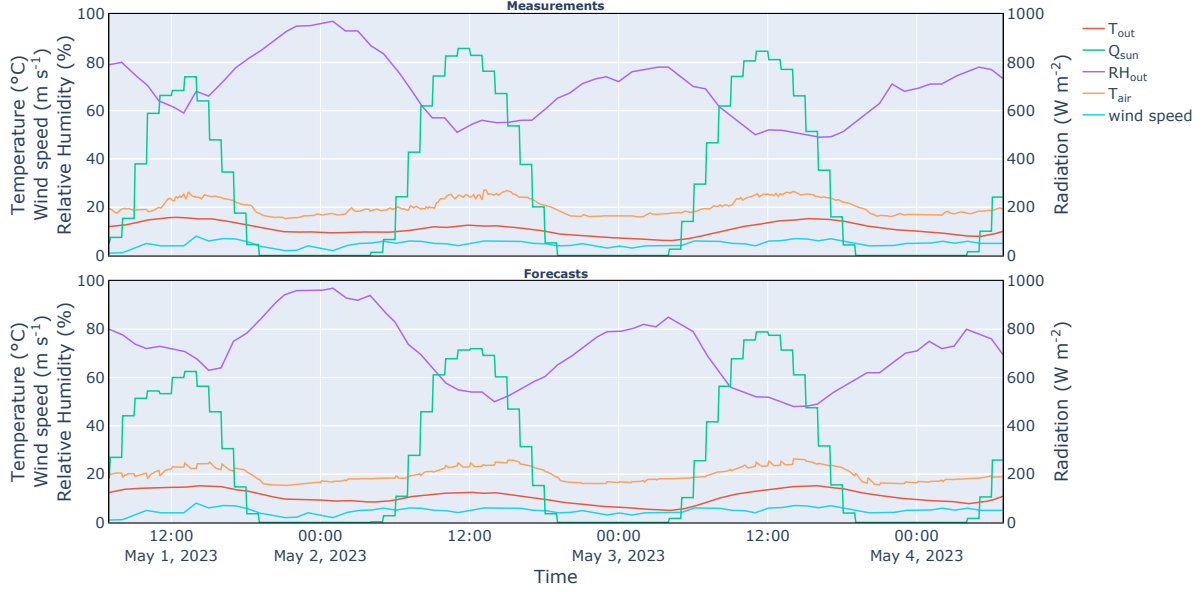


Figure 6.1: A comparison of the greenhouse climate measurements and meteorological measurements from a nearby weather station (top) to the interpolated forecasts for the same period from the same nearby weather station (bottom).

speed instead. The temperature of the air inside the greenhouse (T_{air}) is still gathered from the local climate measurement equipment of the greenhouse for training, whereas the desired future temperature is used during testing alongside the forecasted meteorological data as modern climate computers have long been able to deliver the desired temperatures successfully (Stanghellini & De Jong, 1995). As in the previous chapters, the climate control equipment states and actions are assumed to be known beforehand; thus, the measured values are used. An example of the meteorological and climate measurements compared to the (interpolated) forecasts for the period of 2023-05-01 07:00 to 2023-05-04 07:00 is shown in Figure 6.1. The measurements are shown at the top, and the forecasts are shown at the bottom. The comparison shows that forecasts are relatively accurate for temperature (T_{out}) and relative humidity outside (RH_{out}). Furthermore, forecasts for radiation (Q_{sun}) show more inconsistency, possibly related to local cloud covers being more difficult to forecast. Moreover, one can observe that the desired temperature profile (T_{air}) closely aligns with the measured temperature in the greenhouse.

6.2. Experimental set-up

General

For each experiment, the set-up of the CA and the ML baseline models is the same as outlined in subsection 5.4.2. Specifically, the full- f_a setting of CA and the regressor ensemble are used. The CA model is trained with a maximum of 100 epochs instead of 30 to decrease the risk of stopping before reaching, at least, a local minimum. The regressor ensemble method used in the experiments refers to the regressor ensemble from section 5.3; this method was found to be the most reliable in performance out of the three tested methods. All experiments are run three times with different seeds to limit the effect of randomness on the results. The experiment regarding the training times has been run on a 14-inch M2 Pro Macbook Pro with a 10-core 3.49 GHz 64-bit CPU (6 high-power and 4 efficiency) with 16GB of RAM.

Negative humidity deficit prediction rate

Generally, it is very difficult to ascertain whether a machine learning model has learned a correct model that aligns with the laws of physics. Measuring the alignment of a learned model with the laws of physics is, therefore, non-trivial. In the following, we use the rate of negative humidity deficit predictions as a surrogate measure for this quality instead. As explained in subsection 2.1.1, negative humidity deficits are not physically feasible as this would suggest that the air contains more water vapour than its saturation capacity. The negative humidity deficit prediction rate is calculated by counting the number

of negative predictions (made at a 5-minute interval) and dividing the result by the total number of predictions.

Data-scarcity

A well-known limitation of machine learning models is their reliance on large amounts of training data to achieve adequate performance. However, using prior knowledge should, in theory, allow for better performance with less available data. To test this hypothesis, the CA, regressor ensemble and ML baseline models are each trained using different amounts of available training data. Specifically, for each test window, all three models are trained once with 1, 7, 14, or 28 days of current crop training data for the scenarios where historical crop training data is available and where it is not. The ML baseline and ensemble predictions would not make any sense without at least a little training data, so no experiment is run for the zero-shot (i.e. no training data) scenario.

Outliers / Extreme conditions

Outliers are usually a major issue for ML models that have not yet seen enough similar data in their training set (Kim et al., 2017). It is interesting to examine the performance of the CA, regressor ensemble, and ML baseline models on such outliers. Given that the theoretical model is expected to be better at extrapolating to new situations (Katzin et al., 2022), we expect that the models that use the theoretical model will perform better in these situations. To test this hypothesis, three relevant categories of outliers were identified in collaboration with experts:

- Extremely low & high humidity deficit levels inside the greenhouse
- Extremely high temperatures outside the greenhouse
- Extremely low relative humidity levels outside the greenhouse

The extremely low & high humidity deficit levels are important as they can negatively impact crop health (subsection 2.1.1). Extremely high outside temperatures are important as they are more likely to cause higher humidity deficits and lead to water stress in the crop. Low relative humidity levels outside are similarly important, as ventilation in such conditions can cause excessive humidity loss to the outside air, leading to plant water stress.

For each category, outlier samples were identified using the *interquartile range* (IQR), the difference between the 75th and 25th percentiles of the data. Low outliers are declared as those samples with values less than the 25th percentile $- 1.5\text{IQR}$. High outliers are subsequently declared as those samples with values greater than the 75th percentile $+ 1.5\text{IQR}$. The resulting thresholds were then verified with experts. Following this procedure, the following outlier thresholds were defined:

- $HD \leq 1 \text{ g m}^{-3}$ and $HD \geq 10 \text{ g m}^{-3}$
- $T_{out} \geq 29^\circ\text{C}$
- $RH_{out} \leq 46.5\%$

All models were trained as normal, and predictions were made for the same 89 test windows of 72 hours, with each simulation being run three times with different seeds to limit the effect of randomness. Next, the predictions for the outliers were filtered using the thresholds above. The RMSE and rRMSE of the CA, regressor ensemble, and ML baseline model predictions over the outlier samples are reported. Furthermore, to act as a comparison of performance on similar but non-outlier samples, the mean and standard deviations of the RMSE and rRMSE for the simulations (i.e. the entire 72 hour windows) in which outliers occur are also reported under the name "Overall".

6.3. Determination of model performance

Model performance was quantitatively evaluated using the metrics described in subsection 3.3.3.

6.4. Results

6.4.1. Overall results on forecasted data

First, a general comparison of the CA, regressor ensemble, and ML baseline models was performed to investigate each model's performance and analyse the time needed to train each model. The comparison results between the CA, CA-ML regressor ensemble, and ML baseline models are summarised in Table 6.1. The time to train each model, including the maximum observed, is specified in Table 6.2.

Overall, the performance comparison results show that the obtained errors/scores for both the ML baseline and the regressor ensemble models have deteriorated compared to the results in subsection 5.4.4 from previous experiments using measured data. On the other hand, the performance of the CA model is similar to the results from subsection 4.2.5 from previous experiments using the same measured data.

Table 6.1: Comparison of the performance of the CA, regressor ensemble, and ML baseline models using forecasts of meteorological data instead of measured data. Due to the crop cycle's start- and end date, not enough data is available for the Autumn season.

Season	Model	$RMSE$ (g m^{-3})	$rRMSE$	R^2
Winter ($n = 78$)	CA	1.00 ± 0.47	0.45 ± 0.19	-4.77 ± 18.30
	Ensemble	1.07 ± 0.58	0.47 ± 0.21	-4.60 ± 12.62
	ML baseline	1.13 ± 0.62	0.45 ± 0.19	-5.62 ± 15.06
Autumn ¹	—	—	—	—
Spring ($n = 90$)	CA	0.81 ± 0.35	0.27 ± 0.07	0.79 ± 0.13
	Ensemble	0.84 ± 0.30	0.29 ± 0.08	0.76 ± 0.16
	ML baseline	0.98 ± 0.49	0.32 ± 0.09	0.70 ± 0.18
Summer ($n = 90$)	CA	1.08 ± 0.37	0.28 ± 0.07	0.78 ± 0.13
	Ensemble	1.27 ± 0.49	0.33 ± 0.11	0.65 ± 0.29
	ML baseline	1.40 ± 0.63	0.37 ± 0.15	0.57 ± 0.40
Overall ($n = 267$)	CA	0.98 ± 0.43	0.33 ± 0.15	-0.84 ± 10.17
	Ensemble	1.11 ± 0.55	0.37 ± 0.17	-0.86 ± 7.21
	ML baseline	1.23 ± 0.65	0.40 ± 0.19	-1.23 ± 8.59

Table 6.2: Comparison of the training times (mean + std) and the maximum training time for the CA, regressor ensemble, and ML baseline model. Experiments were run on a 14-inch M2 Pro Macbook Pro with a 10-core 3.49 GHz 64-bit CPU and 16GB of RAM. The best scores were made bold. The sample size for each model is $n = 267$.

Model	Training time (s)	Max training time (s)
CA	508.1 ± 186.5	866.4
Ensemble	529.2 ± 184.9	890.9
ML baseline	8.6 ± 1.9	18.5

6.4.2. Negative humidity deficit prediction rate

The negative humidity deficit prediction rate experiment results are summarised in Table 6.3. A higher rate of negative humidity deficit predictions might indicate that the learned model is "less adherent" to the underlying physics principles at work. From these results, one can see a large difference in the rates of negative, and thus un-physical, predictions for the different models. The worst-performing model is ML baseline, which predicts negative humidity deficit levels 3.41 % of the time. The regressor ensemble model performs better than ML baseline, with 2.28 % of its predictions being negative, a 33 % reduction compared to the ML baseline. The best-performing model is the CA model, which only predicts negative humidity deficit levels 0.44 % of the time, an 87 % reduction compared to the ML baseline.

¹Due to the start- and end-date of the crop cycle, not enough data is available

Table 6.3: Comparison of the rate of negative predictions for the physics-informed CA model, the data-driven ML baseline model, and the regressor ensemble model. Physically, negative humidity deficit values are impossible, so the rate of negative predictions should be as low as possible. The rate of negative predictions was calculated by counting the number of negative predictions per model and dividing by the total number of predictions made by each model.

Model	Rate of Negative Predictions (%)
CA	0.44
Ensemble	2.28
ML baseline	3.41

6.4.3. Data-scarcity

The results for the data-scarcity experiment are summarised in Figure 6.2. In this figure, the average RMSE, rRMSE, and R^2 -score are shown for 1, 7, 14, and 28 days without any historical crop data (a), and with historical crop data (b). From these results, a couple of things become clear. We start by analysing the results of (a) without any historical data.

First, the CA model drastically outperforms the ML baseline and regressor ensemble models with just a single day of current crop training data. This confirms the hypothesis that prior knowledge improves performance with very little training data. The regressor ensemble model does not appear to have enough training data to learn to use the prior knowledge, only outperforming the ML baseline model in R^2 -score. The picture changes when each model has access to 7 days of current crop training; the ML baseline model and the regressor ensemble quickly improve and outperform the CA model. The regressor ensemble outperforms the ML baseline and CA models when given access to 7 or more days of training data. Lastly, the R^2 -score of the CA model does not look to improve much with more data.

The addition of historical crop data has a large impact on the results. With only a single day of current crop training data, we observe that each model performs better in terms of RMSE and rRMSE, whereas the CA model also improves in terms of R^2 -Score, in contrast with the ML baseline and regressor ensemble models, which both deteriorate slightly. Although the same trend of quick improvement can be observed for the ML baseline and regressor ensemble models with more current crop training data, both models perform worse than without historical data. On the other hand, the CA model improves across every metric when given access to the historical-crop training data.

6.4.4. Extreme conditions

The results for the outliers / extreme conditions experiments are summarised in Table 6.4 for the extreme humidity deficit levels, Table 6.5 for the extremely low relative humidity levels, and Table 6.6 for the extremely high temperatures. The results are further analysed below.

High and low humidity deficit (HD) inside the greenhouse

Table 6.4 shows the performance of the CA, regressor ensemble, and ML baseline models on outliers with particularly low and high HD levels inside the greenhouse. Using the thresholds of $HD \leq 1 \text{ g m}^{-3}$ or $HD \geq 10 \text{ g m}^{-3}$, a total of 5500 and 8566 samples spread over 150 and 69 simulations, respectively, or 2.52 % and 3.93 % of the dataset were considered outliers in this sense.

The results show that every method scores better on RMSE on the outliers than the simulations containing the outliers for the extremely low HD levels. However, when looking at the rRMSE, we notice a very high rRMSE. Given the low HD levels for these outliers, the data mean will also be low; thus, obtaining a good rRMSE proves difficult. Nonetheless, the ML baseline scores best for the extremely low HD outliers. The CA model scores best overall on the simulations containing the outliers.

Inversely, RMSE values are significantly higher for extremely high HD levels than low humidity deficit levels and higher than the overall scores on the simulations containing the outliers. The regressor ensemble outperforms the CA and ML baseline models on the outliers while being close to the CA model regarding overall performance on the simulations containing the outliers. Both the CA and ML baseline models perform similarly on the outliers specifically, although the ML baseline performs worse on the simulations containing the outliers overall. Due to the high HD levels for these outliers, the rRMSE values are now lower.

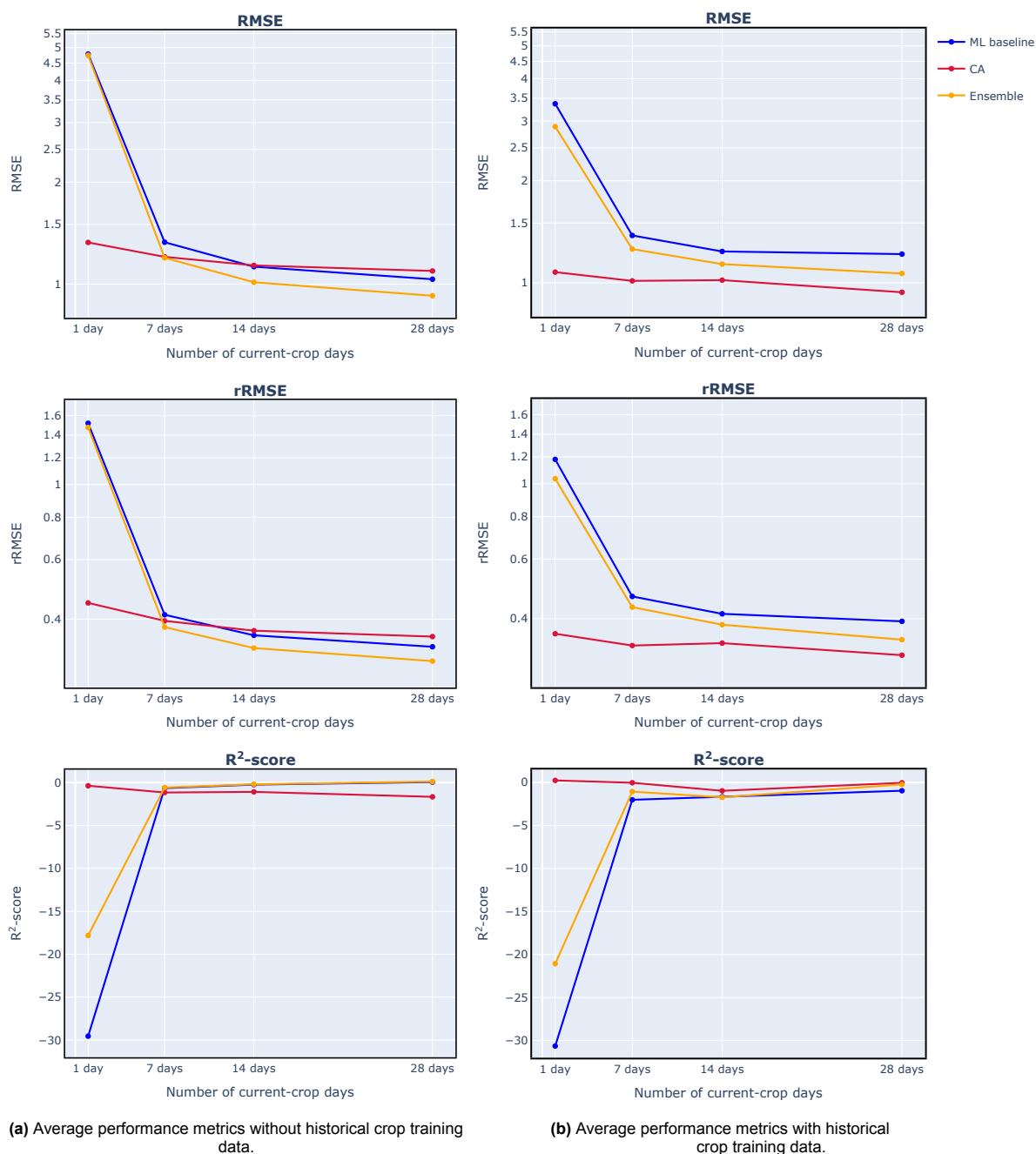


Figure 6.2: Comparison of the CA, regressor ensemble, and ML baseline models using the metrics from subsection 3.3.3. The average values are shown for each model with the number of available training days of the current crop on the x-axis. The RMSE and rRMSE are plotted on log scales to highlight the differences at the lower ends. Due to negative values, the R²-score is plotted on a linear scale. (a) shows the average performance metrics for the models trained without any historical data in the training set. (b) shows the average performance metrics for the models trained with historical data in the training set.

Table 6.4: Performance comparison for outliers with particularly low- and high humidity deficit levels inside. For this experiment, a data point was considered an outlier when $HD \leq 1 \text{ g m}^{-3}$ or $HD \geq 10 \text{ g m}^{-3}$. This resulted in 5500 and 8566 samples spread over 150 and 69 simulations, respectively, or 2.52 % and 3.93 % of the dataset. For each model, the $RMSE$ and the $rRMSE$ are reported solely on the outliers and overall on the simulations containing the outliers (mean $\pm$ std).

Outlier threshold	Model	$RMSE \text{ (g m}^{-3}\text{)}$		$rRMSE$	
		Outliers	Overall	Outliers	Overall
$HD \leq 1 \text{ g m}^{-3}$	CA	0.80	0.83 ± 0.33	0.93	0.31 ± 0.09
	Ensemble	0.77	0.89 ± 0.33	0.90	0.33 ± 0.10
	ML baseline	0.73	0.95 ± 0.36	0.85	0.35 ± 0.11
$HD \geq 10 \text{ g m}^{-3}$	CA	2.70	1.36 ± 0.41	0.22	0.29 ± 0.08
	Ensemble	2.39	1.47 ± 0.56	0.20	0.32 ± 0.14
	ML baseline	2.71	1.70 ± 0.75	0.22	0.37 ± 0.18

Low relative humidity (RH) outside the greenhouse

Table 6.5 shows the results for the performance of the CA, regressor ensemble, and ML baseline models on outliers with particularly low RH levels outside the greenhouse. Using the threshold of $RH_{out} \leq 46.5\%$, a total of 6082 samples spread over 81 simulations, or 2.79 % of the dataset were considered outliers in this sense.

The results show that the regressor ensemble significantly outperforms the CA and ML baseline models on the outliers with low RH levels. It does so in terms of RMSE and rRMSE. Overall, the CA model performs best on the simulations containing the outliers. Nonetheless, overall, the RMSE for the outliers is still worse than that of the simulations containing the outliers, indicating room for improvement. On the other hand, the rRMSE is lower for the outliers than for the simulations containing the outliers. However, the rRMSE does not allow for an entirely fair comparison in this case, as the average HD levels inside the greenhouse during the low RH outliers are higher, resulting in a lower rRMSE, even with similar performance in terms of RMSE.

Table 6.5: Performance comparison for outliers with particularly low relative humidity levels outside. For this experiment, a data point was considered an outlier when $RH_{out} \leq 46.5\%$. This resulted in 6082 samples spread over 81 simulations, or 2.79 % of the dataset. For each model, the $RMSE$ and the $rRMSE$ are reported solely on the outliers and overall on the simulations containing the outliers (mean $\pm$ std).

Model	$RMSE \text{ (g m}^{-3}\text{)}$		$rRMSE$	
	Outliers	Overall	Outliers	Overall
CA	1.81	1.03 ± 0.47	0.19	0.25 ± 0.07
Ensemble	1.52	1.11 ± 0.45	0.16	0.28 ± 0.09
ML baseline	1.96	1.19 ± 0.54	0.21	0.29 ± 0.08

High temperatures outside the greenhouse

Table 6.6 shows the results for the performance of the CA, regressor ensemble, and ML baseline models on outliers with particularly high temperatures outside. Using the threshold of $T_{out} \geq 29^\circ\text{C}$, a total of 1851 samples spread over 24 simulations, or 0.85 % of the dataset were considered outliers in this sense.

From these results, the regressor ensemble performs best on the outliers and the simulations containing the outliers across both RMSE and rRMSE metrics. The air can contain much more water vapour at these high temperatures until saturation. As a result, the HD levels are likely to be higher, partly explaining the larger RMSE scores and lower rRMSE scores. Nevertheless, there is room for improvement as the performance in terms of RMSE is still worse on the outliers than on the simulations containing the outliers.

Table 6.6: Performance comparison for outliers with particularly high temperatures outside. For this experiment, a data point was considered an outlier when $T_{out} \geq 29^\circ\text{C}$. This resulted in 1851 samples spread over 24 simulations, or 0.85 % of the dataset. For each model, the $RMSE$ and the $rRMSE$ are reported solely on the outliers and overall on the simulations containing the outliers (mean $\pm$ std).

Model	$RMSE$ (g m^{-3})		$rRMSE$	
	Outliers	Overall	Outliers	Overall
CA	2.93	1.56 ± 0.45	0.23	0.27 ± 0.06
Ensemble	2.24	1.31 ± 0.43	0.17	0.23 ± 0.06
ML baseline	2.56	1.53 ± 0.52	0.20	0.27 ± 0.07

6.5. Discussion

In this chapter, forecasted meteorological data was used in the simulation process to mimic a more realistic scenario and to explore the impact on the performance of the different models. Each model deteriorated slightly compared to when using measured data, as is to be expected. However, all three models still demonstrated good overall performance, falling well below the 1.5 g m^{-3} $RMSE$ that was posed in subsection 3.3.3 and outperforming theoretical modes from related work (Katzin et al., 2022).

From the experiments with varying amounts of training data, our findings indicate that adding historical data enhances the performance of the CA model. This result confirms our hypothesis that similar and helpful dynamics present in historical data benefit the CA model. However, this enhancement did not extend to the ML baseline or ensemble models. This suggests that while the dynamics might be similar, differences in humidity distribution and meteorological conditions can negatively affect the performance of the ML baseline and ensemble models. Taylor and Letham (2017) also mention: “forecasting methods can perform better or worse with more data. A longer history can lead to worse forecasts when the model is misspecified, and we are overfitting the past”. In any case, further investigation into this discrepancy is warranted, though it falls outside the scope of this study.

Another finding that might point to the ML baseline model being “more” misspecified than the CA model is the notable distinction observed in the rate of negative predictions, where the CA model outperformed the ML baseline model by a significant margin. This supports the hypothesis that prior physical knowledge embedded in the CA model helps prevent un-physical predictions. However, the CA model’s negative prediction rate, though lower, was not zero. This non-zero rate may stem from the augmentation network introducing some un-physical behaviour alongside imperfections in the theoretical model. Furthermore, the regressor ensemble model performs worse than the CA model in terms of negative prediction rate; however, it is still better than the ML baseline. As the regressor ensemble only uses the CA predictions as a regression variable in combination with additional data-driven features, such as auto-regression, it is unsurprising that the regressor ensemble model produces more negative predictions than the CA model.

When evaluating performance on outlier data, the regressor ensemble method excelled, outperforming both the CA and ML baseline models. This indicates a synergistic effect between the two constituent models, enabling more accurate predictions when faced with underrepresented conditions in the training data. Contrary to what was expected, this also shows that the CA model alone does not significantly outperform a pure ML model on outliers despite the prior knowledge incorporated in the model. The regressor ensemble method strikes a good balance, consistently performing well across all metrics and consistently outperforming the ML baseline.

One downside of the CA and ensemble methods is their longer training times, necessitated by the repeated integration of the theoretical model. Nevertheless, given the short prediction times and length of the prediction windows, this longer training duration is unlikely to pose a significant issue in optimal control settings. Furthermore, the CA model was trained on a CPU for this work. From initial testing, GPU training did not speed up the training processes. Due to the limited training data and batch sizes, the overhead of moving memory to and from GPU combined with the large number of complex operations in the theoretical model evaluations was likely too high to see a noticeable improvement in training times. If these issues are overcome, training on the GPU might decrease the training time considerably.

Lastly, it is worth mentioning again that the ML baseline model did not have access to the same data inputs as the CA model, such as the states of climate control equipment, to mimic a realistic setting. This might somewhat skew the comparison's results. Nevertheless, the comparison should still allow us to analyse the impacts of a PIML model in the context of greenhouse horticulture. Additionally, as building a state-of-the-art model has never been the goal of this work, this comparison is justified in the author's view.

6.6. Conclusion

This chapter compared the CA model, a ML baseline model, and the regressor ensemble model (combining the CA and ML baseline models) in a case study to understand their implications on training time, adherence to physical laws, training data requirements, and performance on outliers and in general.

The findings suggest several key insights. First, regarding training procedures and duration, the CA and regressor ensemble models required longer training times due to the repeated integration of the theoretical model. Despite this, the extended training durations of around 5 minutes should not pose a significant issue in our optimal control setting, given that predictions are made for 72 hours in the future. While GPU training did not provide significant speed improvements due to overhead issues, addressing these might reduce training times considerably.

Regarding adherence to physics, the CA model outperformed the ML baseline in producing fewer unphysical negative predictions, indicating that embedding prior physical knowledge helps the CA model prevent unrealistic predictions. However, the CA model still produced some negative predictions, likely due to imperfections in the augmentation network and the theoretical model. The regressor ensemble model performed better than the ML baseline but worse than the CA model, aligning with its reliance on both a data-driven and a physics-informed component.

Regarding training data requirements, adding historical data improved the performance of the CA model. This enhancement was not observed in the ML baseline or regressor ensemble models, suggesting that differences in humidity deficit distribution and meteorological conditions may negatively impact these models' performance, while the underlying dynamics are still a favourable source of information for the CA model. This finding warrants further investigation.

In terms of general performance and handling outlier data, all models performed well overall, staying below the 1.5 g/m^3 RMSE threshold. The regressor ensemble model excelled in handling outlier data, outperforming both the CA and ML baseline models. This indicates a synergistic effect between the constituent models, allowing for more accurate predictions in underrepresented conditions. Contrary to expectations, the CA model alone did not significantly outperform the pure ML model on outliers, highlighting the regressor ensemble model's balanced performance across all metrics.

In conclusion, the physics-informed CA model and the regressor ensemble model show significant advantages over the ML baseline model in terms of adherence to physical laws and performance on outliers. The CA model benefits from historical data, while the regressor ensemble model offers robust performance across various conditions, but both methods require longer training times. These findings underscore the potential of PIML approaches in improving prediction accuracy and reliability in complex environmental modelling scenarios like greenhouse horticulture.

7

Conclusion

7.1. Summary

This thesis explored the impacts of combining prior physics knowledge about the greenhouse climate with machine learning in the form of a PIML model for the task of greenhouse climate forecasting. First, a differentiable theoretical model was developed that aimed at simulating greenhouse humidity deficits, drawing upon established frameworks while enhancing gradient stability for practical applications. Despite its effectiveness under normal conditions, challenges arose during blackout curtain deployments, highlighting the need for error mitigation strategies in such scenarios. Additionally, the model's dependency on precise window opening data posed a notable limitation.

Next, the Controlled Aphynity (CA) model was introduced, leveraging insights from neural controlled differential equations and the Aphynity framework. This novel approach allows for augmenting incomplete prior knowledge with data-driven insights for dynamic forecasting within greenhouse environments while being able to adapt to changing dynamics due to exogenous forces, such as changing weather. *This adaptability to changing dynamics allows for the CA model to be applied to a much wider set of real-world problems than prior models such as Aphynity.* Application of the CA model to predict humidity deficits demonstrated promising results, showing improved results and the ability to correctly augment the misspecification's in the theoretical model during blackout curtain deployments.

Further advancements involved exploring ensemble methods that integrated the CA model with traditional machine learning techniques. Comparative analyses revealed that a uniform average ensemble showed potential synergies during favourable CA model performance but negatively impacted results when CA performed poorly. In contrast, a regressor ensemble method emerged as an effective method for combining PIML and ML models, balancing risk and performance by allowing the ML model to learn from the CA model. This approach proved robust across various greenhouse scenarios and under extreme conditions.

In a detailed, realistic case study, the thesis evaluated the CA model, regressor ensemble, and conventional machine learning baseline across multiple metrics, including training time, adherence to physical laws, data requirements, and performance on outlier data. Results indicated that while the CA and regressor ensemble models required longer training times due to theoretical model integration, they demonstrated superior adherence to physical laws and exhibited fewer unrealistic predictions compared to the baseline. Notably, the CA model benefited significantly from historical data, enhancing its predictive accuracy under varying environmental conditions.

In conclusion, this thesis underscores the potential of PIML approaches in advancing greenhouse climate forecasting. By integrating prior physics knowledge with machine learning techniques, particularly through models like the CA and its use in an ensemble, PIML was made applicable to the context of greenhouse climate simulation and notable improvements in general performance, performance on outliers, performance under scarce training data, and in physical adherence were achieved. Although more research is required, these findings pave the way for future research and practical applications

of PIML to simulate and optimise the greenhouse climate and other real-world dynamical systems influenced by exogenous forces.

7.2. Answers to Research Questions

The research questions from chapter 1 are answered in this subsection. First, the research sub-questions are listed and answered below:

What are the specific limitations of current models in the context of greenhouse climate forecasting, and what are the primary reasons causing these limitations?

- Current models for greenhouse climate forecasting face significant limitations. Theoretical models require in-depth physics knowledge and often difficult parameter estimation, making them hard to use and adapt to different greenhouses. On the other hand, machine learning models require extensive, high-quality data to build empirical models, making them prone to overfitting and often lacking generalisability and adaptability to data they haven't seen before. Furthermore, the lack of any mechanism to base learned models on the laws of physics often leads to misspecified models and infeasible predictions, which further hinders trust in the model and, as a result, its usefulness.

What prior physical knowledge about the dynamics of the greenhouse climate, specifically humidity, is available and can be used in a PIML model?

- Over the years, many theoretical models have been developed to predict the humidity inside a greenhouse based on meteorological and climate control equipment data. These theoretical models provide useful prior knowledge about the dynamics of greenhouse humidity levels that can be used in a physics-informed machine-learning model. A theoretical model derived from the work of Bontsema et al. (2008), De Zwart (1996), Monteith and Unsworth (1990), Stanghellini and De Jong (1995), Van Beveren et al. (2015), and Vanthoor (2011) was adapted to be differentiable for its use in a PIML model. The model was found to provide good results in predicting the humidity deficit, although a more complex model is expected to improve results at the cost of simplicity.

How can this prior knowledge best be combined with machine learning to form a PIML humidity deficit forecasting model?

- Prior knowledge in the form of differential equations was found to be best combined with ML through the concept of Neural Differential Equations (Kidger, 2022). Specifically, ML can be used in this setting to estimate the physical parameters and augment any incorrect physics in the theoretical model. Controlled Aphynity (CA) was proposed in this work as a method to achieve this goal while trying to minimise the role played by the augmentation network. Specifically, Controlled Aphynity is the first model to achieve this goal while being able to adapt to dynamics that change over time due to exogenous forces, such as changing weather. The results show that CA can learn physical parameters, correctly augment incorrect physics, and improve the results of a misspecified theoretical model in the context of greenhouse humidity forecasting. Additionally, the CA model can be used in conjunction with a ML model through an ensemble. Specifically, using the predictions from CA as input to a ML model was found to generally improve results while never lowering them. Through this mechanism, the use of CA in an ensemble was found to be a low-barrier method of implementing PIML into existing ML forecasting models.

Compared to a baseline ML model, what implications does the PIML model have for the training procedures, the adherence to physics, the amount of training data required, and the performance generally and on outliers in the form of extreme weather conditions?

- The CA and ensemble models developed in this work, compared to a pure ML model, have distinct implications for training procedures, adherence to physics, training data requirements, and performance on outliers. Training the CA and ensemble models takes longer due to the repeated integration of theoretical models, although this extended training time is manageable in optimal control settings. The CA model better adheres to physical laws, producing far fewer un-physical negative humidity deficit predictions as a result of the prior physical knowledge, while

the regressor ensemble model also shows improved adherence over the ML model, albeit to a lesser extent. The CA model provides adequate results with very limited data, unlike the ML and regressor ensemble models, which require more data before adequate results are achieved. The regressor ensemble model outperforms the CA and ML models in handling extreme weather conditions, indicating its balanced performance across different metrics and robust handling of outliers.

Given the answers to the questions above, the main research question can now be answered:

What is the impact of combining prior physical knowledge of the greenhouse climate and machine learning to forecast the humidity inside a greenhouse, on the general performance, the performance on outliers, the rate of physically infeasible predictions, and the amount of training data required?

- Combining prior physical knowledge of the greenhouse climate with machine learning to forecast humidity inside a greenhouse improves general performance by leveraging the strengths of both approaches. The physics-informed CA and ensemble model further show enhanced performance on outliers, with the ensemble model particularly excelling in extreme weather conditions. Both the CA and, to a lesser extent, the ensemble model significantly improve the rate of physically infeasible predictions. Additionally, the CA model requires less training data to achieve adequate results than pure ML and ensemble models, making it more data-efficient while maintaining high accuracy and robustness in greenhouse climate forecasting. At the cost of computation time, this combination of physical knowledge with machine learning offers a best-of-both-worlds approach to greenhouse climate forecasting, positively impacting the important areas in this context.

7.3. Limitations & Future Work

Reflecting on this thesis's findings and achievements, several limitations and avenues for future research have emerged, highlighting both the scope of the current work and opportunities for further exploration. In total, we note four important limitations and directions for future work here (in no particular order):

1. **Theoretical model complexity:** While the developed theoretical model and the CA model have shown promising results in greenhouse humidity forecasting, there remains a scope to experiment with larger and more complex theoretical models. This could involve incorporating additional states, variables, control equipment (notably blackout curtains), or refining existing parameters to better capture the intricacies of greenhouse dynamics across different environmental conditions and operational scenarios. Such advancements would likely enhance the model's usefulness, predictive accuracy and robustness, addressing some of the current limitations related to specific operational contexts or greenhouse configurations not fully captured in the current theoretical model. Additionally, the quality of the parameter estimation of CA could not be assessed due to the lack of ground truth for these parameters in this study. Future work could investigate the parameter estimation quality of CA, which might also reveal further optimisations that can be made to CA.
2. **Generalisability across greenhouses:** Although CA promises to be able to generalise across different (types of) greenhouses, this thesis has primarily focused on validating the models in controlled settings and specific case studies. Future research should extend these evaluations to analyse model performance across a broader range of greenhouses with varying structural designs, geographical locations, and crop types. This approach would provide more comprehensive insights into the CA's applicability and effectiveness under diverse real-world conditions, akin to the comprehensive studies conducted by Vanthoor (2011) and others in the field.
3. **Training optimisation:** Although some basic hyperparameter tuning was performed in this thesis, an in-depth exploration of hyperparameters and their impact on model interpretability and performance is warranted. While the current study has made strides in demonstrating the effectiveness of the CA model and ensemble methods, further optimisation of the hyperparameters concerning CA could enhance the results and robustness further and increase understanding of the CA model. Furthermore, more advanced training techniques, such as increasing the sequence length as training progresses, could be tested and explored for increased performance.

Lastly, one could address the overhead issues encountered in this study to speed up training using GPUs. A simple strategy to tackle these issues is to increase the training data and batch size, but this would require some augmentations to the model to adapt to a continuously growing crop, i.e. by including a crop model in the CA prior knowledge.

4. **More advanced ensemble techniques:** Three relatively basic ensemble techniques were explored in this thesis. Although these already provided interesting insights, exploring more advanced ensemble techniques beyond the ones studied here could significantly enhance their usefulness. One promising avenue pertains to using uncertainty quantification in the form of Bayesian model averaging (BMA) to weigh the results of the different models. We might expect that a larger augmentation loss would correspond to a larger uncertainty quantification, but investigating methods to perform uncertainty quantification for the CA model could provide interesting new insights and test this expectation.

References

- Anderson, B., Hy, T.-S., & Kondor, R. (2019, November). Cormorant: Covariant Molecular Neural Networks [arXiv:1906.04015 [physics, stat]]. <https://doi.org/10.48550/arXiv.1906.04015>
- Bakker, J. (1991). Analysis of humidity effects on growth and production of glasshouse fruit vegetables.
- Baydin, A. G., Pearlmutter, B. A., Radul, A. A., & Siskind, J. M. (2018). Automatic differentiation in machine learning: A survey [Publisher: Microtome Publishing]. *Journal of Machine Learning Research*, 18, 1–43.
- Beucler, T., Rasp, S., Pritchard, M., & Gentine, P. (2019, June). Achieving Conservation of Energy in Neural Network Emulators for Climate Modeling [arXiv:1906.06622 [physics]]. <https://doi.org/10.48550/arXiv.1906.06622>
- Bontsema, J., Hemming, J., Stanghellini, C., Visser, P. H. B. d., Henten, E. v., Budding, J., Rieswijk, T., & Nieboer, S. (2008). On-line estimation of the transpiration in greenhouse horticulture. *Proceedings Agricontrol 2007. 2nd IFAC International Conference on Modeling and Design of Control Systems in Agriculture, Osijek, Croatia, 3 - 5 September, 2007*, 29–34. Retrieved January 9, 2024, from <https://research.wur.nl/en/publications/on-line-estimation-of-the-transpiration-in-greenhouse-horticultur>
- Bot, G. P. A. (1983). *Greenhouse climate : From physical processes to a dynamic model* [phd]. Landbouwhogeschool [Pages: -]. Retrieved January 22, 2024, from <https://library.wur.nl/WebQuery/wurpubs/77514>
- Boulard, T., Roy, J. C., Pouillard, J. B., Fatnassi, H., & Grisey, A. (2017). Modelling of micrometeorology, canopy transpiration and photosynthesis in a closed greenhouse using computational fluid dynamics [Publisher: Academic Press]. *Biosystems Engineering*, 158, 110–133. <https://doi.org/10.1016/J.BIOSYSTEMSENG.2017.04.001>
- Brown, R. G. (1956). *Exponential Smoothing for Predicting Demand*. Little. https://books.google.nl/books/about/Exponential_Smoothing_for_Predicting_Dem.html?id=Eo_rMgEACAAJ
- Brunton, S. L., Proctor, J. L., & Kutz, J. N. (2016). Sparse Identification of Nonlinear Dynamics with Control (SINDYc)**SLB acknowledges support from the U.S. Air Force Center of Excellence on Nature Inspired Flight Technologies and Ideas (FA9550-14-1-0398). JLP thanks Bill and Melinda Gates for their active support of the Institute of Disease Modeling and their sponsorship through the Global Good Fund. JNK acknowledges support from the U.S. Air Force Office of Scientific Research (FA9550-09-0174). *IFAC-PapersOnLine*, 49(18), 710–715. <https://doi.org/10.1016/j.ifacol.2016.10.249>
- Chen, R. T. Q., Rubanova, Y., Bettencourt, J., & Duvenaud, D. (2019, December). Neural Ordinary Differential Equations [arXiv:1806.07366 [cs, stat]]. <https://doi.org/10.48550/arXiv.1806.07366>
- Comstock, J., & Ehleringer, J. (1993). Stomatal response to humidity in common bean (*Phaseolus vulgaris*): Implications for maximum transpiration rate, water-use efficiency, and productivity. *Australian Journal of Plant Physiology*, (20), 669–691.
- Coulson, J., Lygeros, J., & Dörfler, F. (2019, March). Data-Enabled Predictive Control: In the Shallows of the DeePC [arXiv:1811.05890 [math]]. <https://doi.org/10.48550/arXiv.1811.05890>
- Daw, A., Thomas, R. Q., Carey, C. C., Read, J. S., Appling, A. P., & Karpatne, A. (2019, November). Physics-Guided Architecture (PGA) of Neural Networks for Quantifying Uncertainty in Lake Temperature Modeling [arXiv:1911.02682 [physics, stat]]. <https://doi.org/10.48550/arXiv.1911.02682>
- De Jong, T. (1990). *Natural ventilation of large multi-span greenhouses* [phd]. Landbouwniversiteit Wageningen [Pages: -]. Retrieved January 22, 2024, from <https://library.wur.nl/WebQuery/wurpubs/11638>
- De Zwart, H. F. (1996). Analyzing energy-saving options in greenhouse cultivation using a simulation model. *Landbouwniversiteit Wageningen*. Retrieved January 4, 2024, from <https://research.wur.nl/en/publications/analyzing-energy-saving-options-in-greenhouse-cultivation-using-a-2>

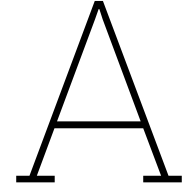
- DESA, U. (2022). *World Population Prospects - Summary of results* (tech. rep.). U.N. Department of Economic and Social Affairs. Retrieved December 7, 2023, from <https://population.un.org/wpp/>
- EUROSTAT. (2022, November). Farmers and the agricultural labour force - statistics. Retrieved December 6, 2023, from https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Farmers_and_the_agricultural_labour_force_-_statistics
- Fang, H., Baret, F., Plummer, S., & Schaepman-Strub, G. (2019). An Overview of Global Leaf Area Index (LAI): Methods, Products, Validation, and Applications. *Reviews of Geophysics*, 57(3), 739–799. <https://doi.org/10.1029/2018RG000608>
- FAO, I. (2023). *The State of Food Security and Nutrition in the World 2023: Urbanization, agrifood systems transformation and healthy diets across the rural–urban continuum*. <https://doi.org/10.4060/cc3017en>
- Fawzi, A., Balog, M., Huang, A., Hubert, T., Romera-Paredes, B., Barekatin, M., Novikov, A., R. Ruiz, F. J., Schrittwieser, J., Swirszcz, G., Silver, D., Hassabis, D., & Kohli, P. (2022). Discovering faster matrix multiplication algorithms with reinforcement learning [Number: 7930 Publisher: Nature Publishing Group]. *Nature*, 610(7930), 47–53. <https://doi.org/10.1038/s41586-022-05172-4>
- Goldammer, T. (2019, April). *Greenhouse Management: A Guide to Operations and Technology* (1st ed.). Retrieved May 8, 2024, from <https://www.greenhouse-management.com/>
- Grossiord, C., Buckley, T. N., Cernusak, L. A., Novick, K. A., Poulter, B., Siegwolf, R. T. W., Sperry, J. S., & McDowell, N. G. (2020). Plant responses to rising vapor pressure deficit [eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/nph.16485>]. *New Phytologist*, 226(6), 1550–1566. <https://doi.org/10.1111/nph.16485>
- Han, J., Jentzen, A., & E, W. (2018). Solving high-dimensional partial differential equations using deep learning [Publisher: Proceedings of the National Academy of Sciences]. *Proceedings of the National Academy of Sciences*, 115(34), 8505–8510. <https://doi.org/10.1073/pnas.1718942115>
- Henten, E. J. v. (1994). *Greenhouse climate management : An optimal control approach* [phd]. Landbouwniversiteit Wageningen [ISBN: 9789054853213 Pages: -]. Retrieved January 9, 2024, from <https://library.wur.nl/WebQuery/wurpubs/25055>
- Hoeting, J. A., Madigan, D., Raftery, A. E., & Volinsky, C. T. (1999). Bayesian Model Averaging: A Tutorial [Publisher: Institute of Mathematical Statistics]. *Statistical Science*, 14(4), 382–401. Retrieved May 27, 2024, from <https://www.jstor.org/stable/2676803>
- Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359–366. [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8)
- Hyndman, R., & Athanasopoulos, G. (2021). *Forecasting: Principles and Practice* (3rd ed). OTexts. Retrieved February 23, 2024, from <https://otexts.com/fpp3/>
- Ivakhnenko, A. G., & Lapa, V. G. (1965). *Cybernetic Predicting Devices*. Joint Publications Research Service [available from the Clearinghouse for Federal Scientific; Technical Information]. https://books.google.nl/books/about/Cybernetic_Predicting_Devices.html?id=I38DHQAACAAJ&redir_esc=y
- Jolliet, O. (1994). HORTITRANS, a Model for Predicting and Optimizing Humidity and Transpiration in Greenhouses. *Journal of Agricultural Engineering Research*, 57(1), 23–37. <https://doi.org/10.1006/jaer.1994.1003>
- Jolliet, O., & Bailey, B. J. (1992). The effect of climate on tomato transpiration in greenhouses: Measurements and models comparison. *Agricultural and Forest Meteorology*, 58(1), 43–62. [https://doi.org/10.1016/0168-1923\(92\)90110-P](https://doi.org/10.1016/0168-1923(92)90110-P)
- Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Židek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., ... Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold [Number: 7873 Publisher: Nature Publishing Group]. *Nature*, 596(7873), 583–589. <https://doi.org/10.1038/s41586-021-03819-2>
- Kashinath, K., Mustafa, M., Albert, A., Wu, J., Jiang, C., Esmaeilzadeh, S., Azizzadenesheli, K., Wang, R., Chattopadhyay, A., Singh, A., Manepalli, A., Chirila, D., Yu, R., Walters, R., White, B., Xiao, H., Tchelepi, H., Marcus, P., Anandkumar, A., & Prabhat, M. (2021). Physics-informed machine learning: Case studies for weather and climate modelling. *Philosophical transactions. Series*

- A, *Mathematical, physical, and engineering sciences*, 379, 20200093. <https://doi.org/10.1098/rsta.2020.0093>
- Katzin, D., van Henten, E. J., & van Mourik, S. (2022). Process-based greenhouse climate models: Genealogy, current status, and future directions. *Agricultural Systems*, 198, 103388. <https://doi.org/10.1016/j.agsy.2022.103388>
- Kempkes, F. (2022). Greenhouses can't do without gas yet. *Wageningen World*, (02). Retrieved May 29, 2024, from <https://www.wur.nl/en/show-longread/greenhouses-cant-do-without-gas-yet.htm#:~:text=Greenhouse%20horticulture%20is%20a%20major,of%20the%20country's%20electricity%20needs>.
- Kidger, P. (2022, February). On Neural Differential Equations [arXiv:2202.02435 [cs, math, stat]]. <https://doi.org/10.48550/arXiv.2202.02435>
- Kidger, P., Morrill, J., Foster, J., & Lyons, T. (2020, November). Neural Controlled Differential Equations for Irregular Time Series [arXiv:2005.08926 [cs, stat]]. <https://doi.org/10.48550/arXiv.2005.08926>
- Kim, B. S., Kang, B. G., Choi, S. H., & Kim, T. G. (2017). Data modeling versus simulation modeling in the big data era: Case study of a greenhouse control system [Publisher: SAGE Publications Ltd]. *Simulation*, 93(7), 579–594. <https://doi.org/10.1177/0037549717692866>
- Körner, O. (2019, September). Models, sensors and decision support systems in greenhouse cultivation. In *Achieving sustainable greenhouse cultivation* (pp. 379–412). <https://doi.org/10.19103/AS.2019.0052.15>
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 25. Retrieved December 14, 2023, from https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
- Lea, C., Vidal, R., Reiter, A., & Hager, G. D. (2016, August). Temporal Convolutional Networks: A Unified Approach to Action Segmentation [arXiv:1608.08242 [cs]]. <https://doi.org/10.48550/arXiv.1608.08242>
- Liu, D., & Wang, Y. (2019). Multi-Fidelity Physics-Constrained Neural Network and Its Application in Materials Modeling. *Journal of Mechanical Design*, 141(121403). <https://doi.org/10.1115/1.4044400>
- Liu, G., Zhong, K., Li, H., Chen, T., & Wang, Y. (2024). A state of art review on time series forecasting with machine learning for environmental parameters in agricultural greenhouses. *Information Processing in Agriculture*, 11(2), 143–162. <https://doi.org/10.1016/j.inpa.2022.10.005>
- Liu, L., Wolterink, J. M., Brune, C., & Veldhuis, R. N. J. (2021). Anatomy-aided deep learning for medical image segmentation: A review. *Physics in Medicine and Biology*, 66(11). <https://doi.org/10.1088/1361-6560/abfbf4>
- Lodovica Gullino, M., Albajes, R., & van Lenteren, J. C. (1999). Setting the Stage: Characteristics of Protected Cultivation and Tools for Sustainable Crop Protection. In R. Albajes, M. Lodovica Gullino, J. C. van Lenteren, & Y. Elad (Eds.), *Integrated Pest and Disease Management in Greenhouse Crops* (pp. 1–15). Springer Netherlands. https://doi.org/10.1007/0-306-47585-5_1
- López-Cruz, I. L., Fitz-Rodríguez, E., Salazar-Moreno, R., Rojano-Aguilar, A., & Kacira, M. (2018). Development and analysis of dynamical mathematical models of greenhouse climate: A review. *European Journal of Horticultural Science*, 83(5), 269–279. <https://doi.org/10.17660/eJHS.2018/83.5.1>
- Loshchilov, I., & Hutter, F. (2017, May). SGDR: Stochastic Gradient Descent with Warm Restarts [arXiv:1608.03983 [cs, math]]. <https://doi.org/10.48550/arXiv.1608.03983>
- Monteith, J. L., & Unsworth, M. H. (1990, February). *Principles of Environmental Physics* [Google-Books-ID: WWbloBL9SmgC]. Butterworth-Heinemann.
- Monteith, J. L., & Unsworth, M. H. (Eds.). (2013). *Principles of Environmental Physics*. Academic Press. <https://doi.org/10.1016/B978-0-12-386910-4.00018-4>
- Mortensen, L. M. (2000). Effects of air humidity on growth, flowering, keeping quality and water relations of four short-day greenhouse species. *Scientia Horticulturae*, 86(4), 299–310. [https://doi.org/10.1016/S0304-4238\(00\)00155-2](https://doi.org/10.1016/S0304-4238(00)00155-2)

- Mortensen, L. M., & Gislerød, H. R. (2005). Effect of air humidity variation on powdery mildew and keeping quality of cut roses. *Scientia Horticulturae*, 104(1), 49–55. <https://doi.org/10.1016/j.scienta.2004.08.002>
- Nemali, K. (2022). History of Controlled Environment Horticulture: Greenhouses [Publisher: American Society for Horticultural Science Section: HortScience]. *HortScience*, 57(2), 239–246. <https://doi.org/10.21273/HORTSCI16160-21>
- O'Shea, K., & Nash, R. (2015, December). An Introduction to Convolutional Neural Networks. <https://doi.org/10.48550/arXiv.1511.08458>
- Pang, G., Lu, L., & Karniadakis, G. E. (2019). fPINNs: Fractional Physics-Informed Neural Networks [arXiv:1811.08967 [physics]]. *SIAM Journal on Scientific Computing*, 41(4), A2603–A2626. <https://doi.org/10.1137/18M1229845>
- Papadakis, G., Frangoudakis, A., & Kyritsis, S. (1992). Mixed, forced and free convection heat transfer at the greenhouse cover. *Journal of Agricultural Engineering Research*, 51, 191–205. [https://doi.org/10.1016/0021-8634\(92\)80037-S](https://doi.org/10.1016/0021-8634(92)80037-S)
- Papastefanopoulos, V., Linardatos, P., & Kotsiantis, S. (2020). COVID-19: A Comparison of Time Series Methods to Forecast Percentage of Active Cases per Population [Number: 11 Publisher: Multidisciplinary Digital Publishing Institute]. *Applied Sciences*, 10(11), 3880. <https://doi.org/10.3390/app10113880>
- Prince, S. J. (2023, May). *Understanding Deep Learning*. MIT Press. Retrieved March 13, 2024, from <http://udlbook.com>
- Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A., & Edelman, A. (2021, November). Universal Differential Equations for Scientific Machine Learning [arXiv:2001.04385 [cs, math, q-bio, stat]]. <https://doi.org/10.48550/arXiv.2001.04385>
- Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378, 686–707. <https://doi.org/10.1016/j.jcp.2018.10.045>
- Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2018). Numerical Gaussian Processes for Time-Dependent and Nonlinear Partial Differential Equations [Publisher: Society for Industrial and Applied Mathematics]. *SIAM Journal on Scientific Computing*, 40(1), A172–A198. <https://doi.org/10.1137/17M1120762>
- Raissi, M., Yazdani, A., & Karniadakis, G. E. (2020). Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science (New York, N.Y.)*, 367(6481), 1026–1030. <https://doi.org/10.1126/science.aaw4741>
- Ritchie, H. (2024). Weather forecasts have become much more accurate; we now need to make them available to everyone. *Our World in Data*. Retrieved June 4, 2024, from <https://ourworldindata.org/weather-forecasts>
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain [Place: US Publisher: American Psychological Association]. *Psychological Review*, 65(6), 386–408. <https://doi.org/10.1037/h0042519>
- Schmidt, R. M. (2019, November). Recurrent Neural Networks (RNNs): A gentle Introduction and Overview [arXiv:1912.05911 [cs, stat]]. <https://doi.org/10.48550/arXiv.1912.05911>
- Shah, V., & Dimitrov, S. (2022). A comparative study of univariate time-series methods for sales forecasting. *International Journal of Business and Data Analytics*, 2(2), 187. <https://doi.org/10.1504/IJBDA.2022.126806>
- Shin, Y., Darbon, J., & Karniadakis, G. E. (2020). On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs [arXiv:2004.01806 [cs, math]]. *Communications in Computational Physics*, 28(5), 2042–2074. <https://doi.org/10.4208/cicp.OA-2020-0193>
- Stanghellini, C. (2014). Horticultural production in greenhouses: Efficient use of water [Publisher: International Society for Horticultural Science ISBN: 9789462610217]. *Acta Horticulturae*, 1034, 25–32. <https://doi.org/10.17660/ACTAHORTIC.2014.1034.1>
- Stanghellini, C., & De Jong, T. (1995). A model of humidity and its application in a greenhouse [Publisher: Elsevier]. *Agricultural and Forest Meteorology*, 76(2), 129–148. [https://doi.org/10.1016/0168-1923\(95\)02220-R](https://doi.org/10.1016/0168-1923(95)02220-R)
- Stanghellini, C. (2024, March). Personal correspondence.

- Stanghellini, C., 9fJy, I., & Vitae, C. (1987). Transpiration of greenhouse crops : An aid to climate management. Retrieved January 9, 2024, from <https://api.semanticscholar.org/CorpusID:83424473>
- Su, Y., Xu, L., & Goodman, E. D. (2018). Nearly dynamic programming NN-approximation-based optimal control for greenhouse climate: A simulation study. *Optimal Control Applications and Methods*, 39(2), 638–662. <https://doi.org/10.1002/oca.2370>
- Tap, F. (2000). *Economics-based optimal control of greenhouse tomato crop production* [phd] [ISBN: 9789058082367 Pages: -]. Retrieved May 15, 2024, from <https://library.wur.nl/WebQuery/wurpubs/66290>
- Taylor, S., & Letham, B. (2017, September). *Forecasting at scale*. <https://doi.org/10.7287/peerj.preprints.3190v2>
- Triebe, O., Laptev, N., & Rajagopal, R. (2019, November). AR-Net: A simple Auto-Regressive Neural Network for time-series [arXiv:1911.12436 [cs, stat]]. <https://doi.org/10.48550/arXiv.1911.12436>
- Triguilii, M. (1999). Effects of humidity on tomato. *Canadian Biosystems Engineering / Le Genie des biosystems au Canada*, 41(3), 135–140.
- USGCRP. (2018). *Fourth National Climate Assessment* (tech. rep.). U.S. Global Change Research Program, Washington, DC. Retrieved December 7, 2023, from <https://nca2018.globalchange.govhttps://nca2018.globalchange.gov/chapter/10>
- Van Beveren, P. J. M. (2023). Optimal management of energy resources in greenhouse crop production systems. <https://doi.org/10.18174/583595>
- Van Beveren, P. J. M., Bontsema, J., Van Straten, G., & Van Henten, E. J. (2015). Minimal heating and cooling in a modern rose greenhouse. *Applied Energy*, 137, 97–109. <https://doi.org/10.1016/j.apenergy.2014.09.083>
- Van Beveren, P., Bontsema, J., Van Straten, G., & Van Henten, E. (2015). Optimal control of greenhouse climate using minimal energy and grower defined bounds. *Applied Energy*, 159, 509–519. <https://doi.org/10.1016/j.apenergy.2015.09.012>
- van Rijswijk, C. (2018, January). World Vegetable Map 2018. Retrieved December 5, 2023, from https://research.rabobank.com/far/en/sectors/regional-food-agri/world_vegetable_map_2018.html
- van Rijswijk, C. (2023, November). The state of the global fresh produce industry. Retrieved December 5, 2023, from <https://research.rabobank.com/far/en/sectors/fresh-produce/the-state-of-the-global-fresh-produce-industry.html>
- van Straten, G., Challa, H., & Buwalda, F. (2000). Towards user accepted optimal control of greenhouse climate. *Computers and Electronics in Agriculture*, 26(3), 221–238. [https://doi.org/10.1016/S0168-1699\(00\)00077-6](https://doi.org/10.1016/S0168-1699(00)00077-6)
- Vanthoor, B. (2011). A model-based greenhouse design method [ISBN: 9789085859192 - 3]. *WUR Repository*.
- van Weel, P., Geelen, P., & Voogt, J. (2018). *Plant Empowerment: The basic principles*.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023, August). Attention Is All You Need [arXiv:1706.03762 [cs]]. <https://doi.org/10.48550/arXiv.1706.03762>
- Villaverde, A. F., Pathirana, D., Fröhlich, F., Hasenauer, J., & Banga, J. R. (2022). A protocol for dynamic model calibration. *Briefings in Bioinformatics*, 23(1), bbab387. <https://doi.org/10.1093/bib/bbab387>
- Wark, K., & Richards, D. (1983). *Thermodynamics* (4th ed.) [Open Library ID: OL3490273M]. McGraw-Hill.
- Yazdani, A., Lu, L., Raissi, M., & Karniadakis, G. E. (2020). Systems biology informed deep learning for inferring parameters and hidden dynamics [Publisher: Public Library of Science]. *PLOS Computational Biology*, 16(11), e1007575. <https://doi.org/10.1371/journal.pcbi.1007575>
- Yin, Y. (2024, February). Personal correspondence on APHYNITY paper.
- Yin, Y., Guen, V. L., Dona, J., de Bézenac, E., Ayed, I., Thome, N., & Gallinari, P. (2021). Augmenting Physical Models with Deep Networks for Complex Dynamics Forecasting [arXiv:2010.04456 [cs, stat]]. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12), 124012. <https://doi.org/10.1088/1742-5468/ac3ae5>
- Zhang, Z., Zhang, W., Zhai, Z. J., & Chen, Q. Y. (2007). Evaluation of Various Turbulence Models in Predicting Airflow and Turbulence in Enclosed Environments by CFD: Part 2—Comparison

with Experimental Data from Literature. *HVAC&R Research*, 13(6), 871–886. <https://doi.org/10.1080/10789669.2007.10391460>



Supporting formulas for humidity calculations

In the following, the formulas are provided that can be used to switch between the different methods of expressing humidity. First, the saturated (absolute) humidity $\mathcal{X}_{air}^*$ at a given temperature T_{air} is calculated according to (Monteith & Unsworth, 2013):

$$\mathcal{X}_{air}^* = \frac{1000 M_w VP_{air}^*}{R(T_{air} + 273.15)} = \frac{2165 VP_{air}^*}{(T_{air} + 273.15)} \quad (\text{g m}^{-3}) \quad (\text{A.1})$$

where:

- M_w (g mol^{-1}) is the molar mass of water.
- R ($\text{J mol}^{-1} \text{K}^{-1}$) is the molar gas constant.
- T_{air} ($^{\circ}\text{C}$) is the temperature of the air.

The saturated vapour pressure VP_{air}^* at T_{air} is calculated according to a slight variation of Tetens' formula (Monteith & Unsworth, 2013):

$$VP_{air}^* = 0.61078 \exp\left(\frac{17.08085 T_{air}}{234.175 + T_{air}}\right) \quad (\text{kPa}) \quad (\text{A.2})$$

Using the above, the following formula can be used to calculate the absolute humidity $\mathcal{X}_{air}$ from the relative humidity RH_{air} :

$$\mathcal{X}_{air} = \frac{RH_{air}}{100} \mathcal{X}_{air}^* \quad (\text{g m}^{-3}) \quad (\text{A.3})$$

where:

- RH_{air} (%) is the relative humidity of the air.

The humidity deficit HD_{air} is calculated from the absolute humidity $\mathcal{X}_{air}$ according to:

$$HD_{air} = \mathcal{X}_{air}^* - \mathcal{X}_{air} \quad (\text{g m}^{-3}) \quad (\text{A.4})$$

Table A.1: The symbol, description and value for the constants used in the supporting formulas for the humidity calculations.

symbol	description	value
M_w	Molar mass of water	18 g mol^{-1}
R	Molar gas constant	$8.314 \text{ J mol}^{-1} \text{K}^{-1}$

B

Summary of Aphynity (Yin et al., 2021)

As the method proposed in 4.1 directly builds upon the work by Yin et al. (2021), we will use this section to provide the reader with a summary of the relevant parts of the Aphynity method to aid in understanding this work.

Yin et al. (2021) study a differential equation over a finite time interval $[0, T]$, which they decompose into a known physics part and an augmentation part of the following form:

$$\frac{dy(t)}{dt} = f(y(t)) = (f_p + f_a)(y(t)) \quad (\text{B.1})$$

where $f_p : \mathbb{R}^{d_y} \rightarrow \mathbb{R}^{d_y}$ encodes the known (incomplete) physics, $f_a : \mathbb{R}^{d_y} \rightarrow \mathbb{R}^{d_y}$ is a data-driven term encompassing the unknown or augmented physics, and $y : [0, T] \rightarrow \mathbb{R}^{d_y}$ describes the state of the system. Notably, the decomposition $f_p + f_a$ is not unique. For example, all the dynamics could be represented solely by the f_a term, vice-versa by the f_p term, or by some combinations of the two. Ideally, one would wish to capture as much of the underlying dynamics of the system in the interpretable and causal f_p term, whereas f_a should model only the information that *cannot* be captured by the physical prior. The general idea of Aphynity is to minimise the impact of f_a , as measured through its norm. This idea is embedded in the following optimisation problem:

$$\min_{f_p, f_a} \|f_a\| \quad \text{s.t.} \quad \forall y \in \mathcal{D}, \forall t, \quad \frac{dy(t)}{dt} = (f_p + f_a)(y(t)) \quad (\text{B.2})$$

To solve this optimisation problem, f_p and f_a are parameterised, denoted by $f_p^{\theta_p}$ and $f_a^{\theta_a}$. Given access to a dataset $\mathcal{D} = \{(y_{k\Delta t}^{(i)})_{0 \leq k \leq \lfloor T/\Delta t \rfloor}\}_{1 \leq i \leq N}$ of observed trajectories with a temporal resolution Δt , the optimisation problem can be reformulated as follows:

$$\min_{\theta_p, \theta_a} \|f_a^{\theta_a}\| \quad \text{s.t.} \quad \forall i, \forall k, \quad y_{k\Delta t}^{(i)} = y_0^{(i)} + \int_0^{k\Delta t} (f_p^{\theta_p} + f_a^{\theta_a})(y(s)) ds \quad (\text{B.3})$$

Due to hard-to-strictly-enforce constraints in Equation (B.3), Yin et al. (2021) use an adaptive constrained optimisation procedure, based on a sequence of Lagrangian relaxations $\mathcal{L}_{\lambda_j}(\theta_p, \theta_a)$:

$$\mathcal{L}_{\lambda_j}(\theta_p, \theta_a) = \|f_a^{\theta_a}\| + \lambda_j \cdot \mathcal{L}_{MSE}(\theta_p, \theta_a) \quad (\text{B.4})$$

where $\mathcal{L}_{MSE}(\theta_p, \theta_a)$ denotes the mean squared error between the simulated trajectories $\tilde{y}_{\theta_p, \theta_a}$ and the target trajectories y . An increasing sequence $(\lambda_j)_j$ is used to maintain a balance between the minimisation of $\|f_a^{\theta_a}\|$ and prediction accuracy. This sequence is selected using the following iterative strategy (Yin et al., 2021):

$$\lambda_{j+1} = \lambda_j + \tau_2 \cdot \mathcal{L}_{MSE}(\theta_{j+1}) \quad (\text{B.5})$$

where λ_0 and τ_2 are hyper-parameters that influence the balance between optimising the trajectory loss and the augmented loss, and $\theta = (\theta_p, \theta_a)$. The algorithm is summarised in Algorithm 5.

Algorithm 5 Aphynity optimisation procedure (Yin et al., 2021)

Initialisation: $\lambda_0 \geq 0, \tau_1 > 0, \tau_2 > 0$

```

1: for  $epoch = 1 : N_{epochs}$  do
2:   for  $iter = 1 : N_{iter}$  do
3:     for  $batch = 1 : B$  do
4:        $\theta_{j+1} \leftarrow \theta_j - \tau_1 \cdot \nabla[\lambda_j \mathcal{L}_{MSE}(\theta_j) + \|f_a\|]$  ▷ where  $\theta_j = (\theta_p^j, \theta_a^j)$ 
5:     end for
6:   end for
7:    $\lambda_{j+1} \leftarrow \lambda_j + \tau_2 \cdot \mathcal{L}_{MSE}(\theta_{j+1})$ 
8: end for

```
