Identification of thunderstorm occurrence in convection-permitting ensemble forecasts using deep neural networks

Kianusch Vahid Yousefnia

München 2025

Identification of thunderstorm occurrence in convection-permitting ensemble forecasts using deep neural networks

Kianusch Vahid Yousefnia

Dissertation
der Fakultät für Physik
der Ludwig-Maximilians-Universität
München

vorgelegt von
Kianusch Vahid Yousefnia
aus Freiburg im Breisgau

München, den 2. Juni 2025

Erstgutachter: Prof. Dr. George Craig
Zweitgutachter: Prof. Dr. Markus Rapp
Tag der mündlichen Prüfung: 21. Juli 2025

ABSTRACT

Thunderstorms have potentially hazardous impacts on society and the economy due to accompanying phenomena, such as lightning, strong winds, and intense precipitation, creating a demand for accurate and timely thunderstorm forecasts. Thunderstorm forecasts several hours in advance are based on simulations of the future atmosphere via numerical weather prediction (NWP). However, as none of the NWP state variables, such as temperature, pressure, or specific humidity, directly indicates thunderstorm occurrence, surrogate variables like convective available potential energy or synthetic radar reflectivity are used as proxies instead.

Surrogate variables of thunderstorm occurrence are typically derived from NWP state variables through the consideration of physical principles and empirical knowledge. In this thesis, however, we present a machine learning (ML) model based on deep learning which bypasses the use of such surrogate variables; instead, the model directly processes the vertical variation of the NWP state variables with height to infer the corresponding probability of thunderstorm occurrence. In addition, this thesis makes use of a convection-permitting ensemble NWP model, i.e., an NWP model which (1) allows for resolving atmospheric convection without parameterizations, and (2) generates multiple possible forecasts consistent with forecast uncertainty. While these two properties have individually shown promise for improving thunderstorm forecasts, their combined potential for this task has so far been less explored. Specifically, we train our model on forecasts of ICON-D2-EPS, a limited-area model for Central Europe run operationally by the German Meteorological Service (DWD), with observations from the lightning detection network LINET serving as the ground truth. With regard to model architecture, we employ considerations based on physics and symmetries to keep model size and inference times computationally efficient. For instance, a sparse layer encourages interactions at similar height levels, whereas a shuffling mechanism forces the model to learn pressure coordinates instead of non-physical patterns tied to the vertical NWP grid.

Evaluating our model for lead times up to 11 h, we find that it outperforms a baseline model relying on traditional thunderstorm surrogate variables, which shows the capability of deep learning methods to discover—on their own—skillful representations of thunderstorm occurrence in NWP data. A linear sensitivity analysis (saliency map) suggests that these patterns found in the data are to a considerable extent physically interpretable: our model has learned the climatological propagation direction of thunderstorms in the study region and

relies on fine-grained structures, such as ice-particle content near the tropopause and cloud cover, as well as mesoscale structures related to atmospheric instability and moisture. As additional results, we quantitatively explain skill gains resulting from our use of ensemble data. Finally, we demonstrate how neural network models like ours help keeping thunderstorm occurrence predictable for longer lead times compared to models which do not rely on ML.

This thesis primarily contributes to improving the skill of thunderstorm forecasts by combining high-resolution NWP and ensemble systems with deep learning. On the other hand, many concepts and methods derived here apply to general binary classification problems, especially when high class imbalance is involved. More generally, our results exemplify the usefulness of incorporating physical considerations and symmetry principles into ML architectures to achieve lightweight models.

ZUSAMMENFASSUNG

Gewitter haben potenziell gefährliche Auswirkungen auf Gesellschaft und Wirtschaft, da sie mit Begleiterscheinungen wie Blitzschlag, starken Winden und intensiven Niederschlägen einhergehen. Dies führt zu einem Bedarf an präzisen Gewittervorhersagen. Vorhersagen mehrere Stunden im Voraus basieren auf Simulationen der zukünftigen Atmosphäre mittels numerischer Wettervorhersage (NWP). Da jedoch keine der NWP-Zustandsgrößen wie Temperatur, Druck oder spezifische Feuchte direkt auf das Auftreten von Gewittern hinweist, werden stattdessen Ersatzgrößen wie die für Konvektion verfügbare potentielle Energie (CAPE) oder synthetische Radarreflektivität als Indikatoren für Gewitter herangezogen.

Solche Ersatzgrößen werden üblicherweise unter Anwendung physikalischer Prinzipien und Empirie aus den NWP-Zustandsgrößen abgeleitet. In der vorliegenden Arbeit wird hingegen ein tiefes neuronales Netzwerkmodell des maschinellen Lernens (ML) vorgestellt, das ohne Verwendung dieser Ersatzgrößen auskommt. Stattdessen verarbeitet das Modell direkt die vertikale Variation der NWP-Zustandsgrößen mit der Höhe, um die entsprechende Wahrscheinlichkeit eines Gewitterauftretens zu bestimmen. Ergänzend kommt ein konvektionsauflösendes Ensemble-NWP-Modell zum Einsatz, das also (1) atmosphärische Konvektion ohne Parametrisierung auflöst und (2) mehrere mögliche Vorhersagen im Rahmen der Unsicherheit generiert. Obwohl beide Eigenschaften bereits einzeln vielversprechende Ergebnisse hinsichtlich der Verbesserung der Vorhersagegüte von Gewittern zeigen konnten, wurde deren kombinierter Nutzen bisher weniger untersucht. Das trainierte ML-Modell basiert auf Vorhersagen des operationellen Lokalmodells ICON-D2-EPS für Mitteleuropa des

Deutschen Wetterdienstes (DWD); als Referenzdaten dienen Beobachtungen des Blitzortungssystems LINET.

Bei der Modellarchitektur wird besonderer Wert auf physikalische und symmetriebezogene Überlegungen gelegt, um Modellgröße und Inferenzzeiten effizient zu halten. So fördert etwa eine spärliche Schicht Interaktionen auf ähnlichen Höhenniveaus, während ein Shuffling-Mechanismus das Erlernen von Druckkoordinaten anstelle nicht-physikalischer Muster innerhalb des vertikalen NWP-Gitters erzwingt.

Die Modellbewertung für Vorhersagehorizonte von bis zu 11 Stunden zeigt eine überlegene Leistung im Vergleich zu einem Referenzmodell, das auf traditionellen Ersatzgrößen basiert. Dies unterstreicht das Potenzial von Deep-Learning-Methoden, selbstständig aussagekräftige Repräsentationen für Gewitterereignisse in NWP-Daten zu erlernen. Eine lineare Sensitivitätsanalyse (Salienzkarte) weist darauf hin, dass viele der identifizierten Muster physikalisch interpretierbar sind: So wird etwa die klimatologische Zugrichtung von Gewittern im Untersuchungsgebiet erfasst, wobei zusätzlich feinskalige Strukturen wie der Eispartikelgehalt nahe der Tropopause, Wolkenbedeckung sowie mesoskalige Merkmale atmosphärischer Instabilität und Feuchtigkeit berücksichtigt werden. Darüber hinaus wird in dieser Arbeit die Verbesserung der Vorhersagegüte, die aus dem Einsatz von Ensemble-Daten resultiert, quantitativ erklärt. Abschließend wird gezeigt, dass neuronale Netzwerke wie das vorgestellte Modell zur Erhaltung der Vorhersagbarkeit von Gewittern auch bei längeren Vorhersagezeiten beitragen können – im Gegensatz zu Verfahren ohne maschinelles Lernen.

Diese Arbeit leistet einen Beitrag zur Verbesserung der Gewittervorhersagegüte durch die Kombination hochauflösender numerischer Wettervorhersagemodelle mit Ensemble-Ansätzen und Deep Learning. Die entwickelten Konzepte und Methoden sind zudem auf allgemeine binäre Klassifikationsprobleme übertragbar, insbesondere bei stark unausgewogenen Klassenverhältnissen. Darüber hinaus wird die Relevanz physikalischer Überlegungen und Symmetrieprinzipien für die Entwicklung effizienter ML-Modelle exemplarisch aufgezeigt.

PUBLICATIONS

All major results of this thesis have been reported in one of the following first-author publications (abbreviated P1-3).

- **P1:** Vahid Yousefnia, K., T. Bölle, I. Zöbisch, T. Gerz (2024). "A machine-learning approach to thunderstorm forecasting through post-processing of simulation data." In: *Quarterly Journal of the Royal Meteorological Society* 150.763, pp. 3495–3510. DOI: 10.1002/qj.4777.
- **P2 [under review]:** Vahid Yousefnia, K., C. Metzl, T. Bölle (2025). "Inferring Thunderstorm Occurrence from Vertical Profiles of Convection-Permitting Simulations: Physical Insights from a Physical Deep Learning Model." In: *Artificial Intelligence for the Earth Systems*. Preprint available from: https://arxiv.org/abs/2409.20087.
- **P3:** Vahid Yousefnia, K., T. Bölle, C. Metzl (2025). "Increasing NWP Thunderstorm Predictability Using Ensemble Data and Machine Learning". arXiv: 2502.13316 [physics.ao-ph]. URL: https://arxiv.org/abs/2502.13316.

Disclaimer: *Some sentences or parts of sentences in this thesis have been taken verbatim from P1-3, or have been only slightly modified.*

ACKNOWLEDGMENTS

This PhD thesis would not exist without the continuous support of many people over the past years. Before delving into the topic of thunderstorm forecasting, I would like to express my heartfelt gratitude to those who accompanied me on this journey.

When I graduated in 2021 with a Master's degree in Theoretical and Mathematical Physics (TMP), my focus had been on heavy-ion collisions. While I deeply enjoyed my studies, I began to wonder whether the theoretical tools I had acquired could be applied to a new domain—one equally significant in its own right, but whose societal relevance is more immediately recognizable to a broader audience. In my search for such a topic, I came across the Institute of Atmospheric Physics (IPA) at the German Aerospace Center (DLR) in Oberpfaffenhofen, near Munich. By fortunate coincidence, the Applied Meteorology (MET) Department was in the process of reshaping its thunderstorm nowcasting and forecasting efforts, with a renewed focus on machine learning methods. At that time, the only two members in the new Thunderstorm Group were the postdoctoral researchers Isabella Zöbisch and Tobias Bölle, who became my supervisors at DLR. Coming all three from different scientific disciplines, we had the exciting opportunity to build research methodologies, frameworks, and visions for our group from the ground up. Thank you both for always being available, for encouraging me to explore detours—but not to get lost in them—and for our stimulating Tuesday discussions and joyful evenings at the *Ungewitter* pub. I would also like to thank Thomas Gerz, who led the MET Department when I began, for entrusting me with a meteorology-focused PhD project despite my background, and for granting me so much freedom in shaping my research. To his successor, Norman Wildmann, I am grateful for your continued support, trust, and genuine interest in my work, as well as the financial stability that allowed me to focus fully on my project and attend insightful conferences. From the university side, I am deeply thankful to George Craig for agreeing to supervise my PhD. Your vast theoretical knowledge and meteorological intuition, as well as your inspired research suggestions, have been invaluable.

Beyond my supervisors, I owe the completion of this thesis to many others. Christoph Metzl, fellow TMP-PhD student in the Thunderstorm Group—thank you for countless stimulating discussions on physics and beyond. I am also grateful to you and Tobias for the in-depth discussions surrounding the results of this thesis and for reviewing my paper drafts. Many thanks to Martin Dameris, my mentor, for his guidance, and to Brigitte Ziegele for her kind and reliable

support with all administrative matters. I am grateful to Björn Brötz for introducing me to the computational and data resources of the high-performance data analytics project *terrabyte*—instead of fulfilling my naive original request for "lots of external hard drives". My PhD project would simply not have been possible without the ability to store over 100 terabytes of data and access it seamlessly from a powerful CPU/GPU cluster—all within a unified infrastructure. I also want to thank those who internally reviewed the papers or code pertaining to my PhD: George Craig, Gerard Kilroy, Jeffrey Thayer, Hessel Juliust, Björn Brötz, and Klaus Gierens. To my former interns Dorothea Schwärzel and Eoin Commins—much of the intuition I developed around thunderstorm forecasting stems from the amazing work you did during your time in the Thunderstorm Group. Thank you! A big thank you also to Antonia Winter for proofreading the final PhD thesis manuscript, and to Guido Schröder and Manuel Baumgartner from DWD for ongoing helpful discussions. Niklas Wartha—thank you for the coffee and tea breaks, and the (more or less) productive late nights at IPA. I am also grateful to all my colleagues in the MET Department for fostering such a respectful and welcoming work environment.

In addition to the scientific community, I want to acknowledge the many dear friends who offered encouragement, perspective, and kindness throughout this time—you know who you are. Your presence, near or far, made a world of difference. A special shoutout to my eleven housemates—thank you! Together, we truly made the most of a global pandemic. To Benita—I am so glad that our paths crossed during my PhD. Thank you for celebrating the highs with me and helping me weather the (thunder)stormy times. Finally, my deepest gratitude goes to my parents, Minna and Amir, and my sister, Jasmin. Thank you for your unwavering belief in me and for your loving support throughout my academic journey.

CONTENTS

ACRONYMS

AUC area under the curve

BS Brier score

BSS Brier skill score

CAPE convective available potential energy

CIN convective inhibition

CSI critical-success index

DWD German Meteorological Service

EL equilibrium level

ETS equitable threat score

FAR false-alarm ratio

ICON Icosahedral Nonhydrostatic

KENDA Kilometer-scale Ensemble Data Assimilation

LAM limited-area model

LCL lifting condensation level

LFC level of free convection

LINET Lightning Detection Network

MCS mesoscale convective system

ML machine learning

MPI Message-Passing Interface

NWP numerical weather prediction

POD probability of detection

PR precision-recall

RQ research question

SALAMA signature-based approach of identifying lightning activity using machine learning

SGD stochastic gradient descent

UTC Coordinated Universal Time

WMO World Meteorological Organization

Part I

INTRODUCTION AND BACKGROUND

1 INTRODUCTION

1.1 THE AIM OF THIS WORK

Thunderstorms have likely inspired awe, fear, and reverence, in humankind since the earliest days of our existence. For instance, in Ancient Rome, lightning and thunder were associated with Jupiter, the most powerful god in Roman mythology.[1] In 1505, the 21-year-old law student Martin Luther was caught in a violent thunderstorm and, fearing for his life, vowed to become a monk if he survived—an event that ultimately set him on the path to initiating the Protestant Reformation (Brecht, 1983, p. 57). In 1725, the Italian composer Antonio Vivaldi expressed the fury and raw energy of thunderstorms in the dramatic third movement of *Summer* from *The Four Seasons* (Lockey, 2017).

While thunderstorms are no less inspiring nowadays, their impact in the form of lightning, strong winds, and intense precipitation (including graupel and hail) is nevertheless hazardous to society and the economy. For instance, there is the small but real chance of being struck by lightning (Holle, 2016), which is why people are advised to "go indoors when thunder roars". Thunderstorms also threaten crops and livestock (Holle, 2014), and may trigger wild fires (Veraverbeke et al., 2017). Additionally, they constitute a major safety concern for aviation (Gerz et al., 2012; Borsky et al., 2019). Furthermore, thunderstorms and lightning damage electrical electrical infrastructure such as wind turbines (Yasuda et al., 2012), decelerating the transition to sustainable energy production. Finally, although the global impact of climate change on thunderstorm frequency is still uncertain and varies regionally, studies suggest that thunderstorms will become more frequent in many European countries (Diffenbaugh et al., 2013; Rädler et al., 2019; Taszarek et al., 2021). This trend underscores the growing importance of accurate and timely forecasts of thunderstorm occurrence in the future.

Methods for forecasting thunderstorm occurrence fall into two categories based on how far in advance one aims to predict. For short-term forecasts with lead times up to approximately 2 h, the highest skill is achieved with *nowcasting* methods based on the extrapolation of remote sensing data (e.g., James et al., 2018; Pulkkinen et al., 2019; Y. Zhang et al., 2023). On the other hand, nowcasting skill quickly

1 See, e.g., "tum pater omnipotens misso perfregit Olympum fulmine" (translation from Latin: "then the all-mighty father shattered Olympus by emitting a bolt of lightning"), from Ovid: *Metamorphoses*, book 1, lines 154–155.

deteriorates in the course of even 1 h (Pulkkinen et al., 2020; Leinonen et al., 2023), which is why thunderstorm forecasts several hours ahead rely on numerical weather prediction (NWP). This method essentially consists of simulating the future atmospheric state by numerically solving equations derived from the laws of physics. NWP skill has improved in the past decades due to more powerful high-performance computing, a more accurate representation of physical processes, the increased availability of observational data through satellite imagery, and advancements in how these observations are assimilated into physically consistent initial conditions (Bauer et al., 2015; Yano et al., 2018). While the atmospheric state in NWP is encoded in terms of certain three-dimensional state variables, such as temperature, pressure, or specific humidity, none of these variables alone directly indicates thunderstorm occurrence. As a matter of fact, individual strokes of lightning, the defining aspect of thunderstorms, are not even part of the governing equations in NWP. Instead, forecasting thunderstorm occurrence using NWP boils down to first computing the atmospheric state at the target time, and then to identifying thunderstorm occurrence in the NWP output via surrogate variables.

Surrogates of thunderstorm occurrence and convective environments in NWP output have traditionally been derived from the state variables via a combination of physical considerations and empirical knowledge. Examples include simulated radar reflectivity (e.g., Kain et al., 2008; Kober et al., 2012; Kerr et al., 2025), updraft helicity (Sobash et al., 2011; Loken et al., 2017), or convective available potential energy (CAPE; Kaltenböck et al., 2009; Taszarek et al., 2021). In addition, forecasters consider possible sources of lift via, e.g., orography, or solar radiation. A simultaneous consideration of *multiple* surrogates was made possible by fuzzy logic expert systems, which rely on decision rules based on domain knowledge to automatically identify thunderstorm occurrence (e.g., P.-F. Lin et al., 2012; J. Li et al., 2021). Lately, works have been concentrating on machine learning (ML) methods based on artificial neural network models, which generalize the fuzzy-logic approach in the sense that decision rules are constructed by solving a data-driven optimization problem. These methods are powerful because they enable the systematic development of models tailored to specific applications. While the developer supplies input samples and a ground truth of the application of interest, the ML framework provides a systematic "recipe" for model development. In the use-case of identifying thunderstorm occurrence in NWP data, the ground truth is often provided by observational data, such as satellite imagery (Jardines et al., 2021, 2024a), radar data (Gagne et al., 2017; Burke et al., 2020; Leinonen et al., 2022), storm reports (Loken et al., 2020; Sobash et al., 2020), or lightning (Ukkonen et al., 2019; Geng et al., 2021). Previous studies include neural networks with relatively few neurons (Jardines et al., 2021; Kamangir et al., 2020; Sobash

et al., 2020; Ukkonen et al., 2019), as well as neural networks with hundreds of thousand trainable parameters (Geng et al., 2021; Jardines et al., 2024b). Findings suggest that neural network models are more skillful at predicting thunderstorm occurrence than comparable ML approaches like random forests (Herman et al., 2018; Ukkonen et al., 2019).

As many surrogate variables are derived from the NWP state variables, a natural progression from the state of the art is to process the NWP state variables directly, as shown in Fig. 1.1. The corresponding ML architectures, deep neural networks, would learn the representations needed to infer thunderstorm occurrence automatically, eliminating the need for "human-designed" surrogates altogether. Deep learning models have demonstrated remarkable success in computer vision, outperforming traditional ML approaches (Krizhevsky et al., 2012; LeCun et al., 2015). On the other hand, if applying deep learning directly to NWP state variables is a logical next step, one may wonder why this has not yet become standard for the identification of thunderstorm occurrence in NWP forecasts. The primary limitations have likely been the computational demands associated with handling large, complex, datasets. These challenges are increasingly surmountable, as evidenced by recent work incorporating multiple NWP state variables at several vertical levels alongside surrogates (Zhou et al., 2019; Jardines et al., 2024a). An example from a related topic includes a deep learning model which infers precipitation rates directly from six state variables, outperforming the NWP model's quantitative precipitation forecasts (Zhou et al., 2022). Yet, to the best of our knowledge, no study has directly inferred thunderstorm occurrence solely from the NWP state variables. With this thesis, we aim to contribute to closing this research gap.

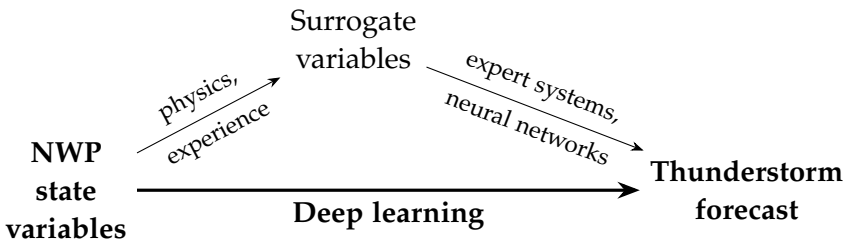

Figure 1.1: Deep learning as a means of processing NWP state variables directly, bypassing the use of surrogate variables.

While the development of a deep neural network model processing NWP state variables directly constitutes a crucial means by which we aim to improve the skill of NWP-based thunderstorm forecasts, it is not the only research path which we will follow in this thesis. Indeed, one crucial reason why the provision of accurate thunderstorm forecasts several hours ahead remains difficult is that the underlying NWP model's skill at producing the target atmospheric state is simply not

perfect. NWP forecast uncertainty results from model grid spacings too coarse to resolve single-cell convection, insufficient sub-gridscale parametrization of model physics and uncertain initial and boundary conditions, and is exacerbated by rapid error growth in time due to the fundamentally chaotic nature of the equations of motion of the atmospheric state (Lorenz, 1969; Palmer et al., 2014; Craig et al., 2021). As a consequence, the skill of thunderstorm forecasts based on NWP decreases with lead time, which ultimately limits the predictability of thunderstorm occurrence in practice. Therefore, thunderstorm forecast skill may not only be increased by improvements of the NWP output processing but also by accounting for advancements related to the underlying NWP model itself.

We now motivate two advancements in NWP towards more skillful thunderstorm forecasts, namely convection-permitting models and ensemble systems. The former concept refers to NWP models with a sufficiently small grid spacing so that convective parameterizations, which account for the net effect of subgrid-scale atmospheric convection on the NWP state variables, can be switched off. Studies have shown convection-permitting models to be beneficial for forecasting thunderstorms and extreme precipitation (Done et al., 2004; A. J. Clark et al., 2009; P. Clark et al., 2016). However, so far, only few studies have deployed ML for identifying thunderstorm occurrence in forecasts of convection-permitting NWP models (Sobash et al., 2020; Burke et al., 2020). The latter NWP-based advancement refers to ensemble models. In contrast to deterministic models, which compute a single future atmospheric state from the initial conditions, ensemble models produce multiple physically consistent forecasts. The variability between the ensemble members aims to reflect the NWP uncertainty in the initial conditions, NWP model parameters, and boundary conditions, and can be harvested to improve on deterministic forecasts (Richardson, 2000; Zhu et al., 2002; Schwartz et al., 2017). In fact, previous studies (Schwartz et al., 2015; Loken et al., 2017) show that combining severe weather forecasts from multiple members increases forecast skill, especially on mesoscale length scales relevant for thunderstorms (Sobash et al., 2016). Although Jardines et al. (2024a) explicitly consider the ensemble spread of predictors in their ML model, most studies already achieve improved forecast skill simply by ensemble-averaging member-wise severe-weather forecasts (Schwartz et al., 2015; Sobash et al., 2016; Loken et al., 2017).

Based on the above considerations, we identify three research paths towards more skillful NWP-based thunderstorm forecasts:

1. Direct processing of the NWP state variables via a deep learning model instead of relying on thunderstorm surrogates which are derived from the state variables.

2. Consideration of convection-permitting NWP forecasts.

3. Consideration of ensemble NWP forecasts.

Pursuing the incorporation of all three aspects, we formulate the following overarching aim of this thesis:

Aim of this thesis: *Development of a deep neural network model for the identification of thunderstorm occurrence in convection-permitting NWP ensemble forecasts.*

Addressing the aim in several intermediate steps, we develop SALAMA (signature-based approach of identifying lightning activity using machine learning), a series of ML models for identifying thunderstorm occurrence in NWP data. All SALAMA models are trained on operational forecasts of ICON-D2-EPS, a convection-permitting ensemble NWP model for Central Europe run operational by the German Meteorological Service (DWD). Lightning observations from the lightning detection network LINET serve as the ground truth for training. We evaluate model performance for lead times up to 11 h.

1.2 INTERMEDIATE STEPS AND RESEARCH QUESTIONS

Instead of building the target model for identifying thunderstorm occurrence in one go, we take several intermediate steps towards the aim of this thesis. Each step gives rise to one first-author publication, and helps addressing certain research questions which we will formulate below.

As we shall see, the large files sizes associated with individual convection-permitting ensemble NWP forecast samples limit training set sizes and, ultimately, the complexity of ML models trained on these data sets. Especially since thunderstorms are climatologically rare events, it is all the more difficult to adequately represent them in the already limited training set. Therefore, it is key to develop an ML framework which addresses problem-related data scarcity. On the other hand, an ML model processing the NWP state variables directly requires a certain amount of model complexity for skillful representations indicative of thunderstorm occurrence. Our first research question (RQ) explores how to address this issue:

RQ 1: *How can an ML framework account for the rare occurrence of thunderstorms and for practical limits on training data size due to computational costs?*

One outcome of this will be to rely on physics-based ideas and symmetry considerations to constrain model complexity. In particular, instead of processing forecasts from all ensemble members simultaneously, we first develop a simpler neural network model which identifies thunderstorm occurrence in forecasts of each member separately; hence, the model acts as a single-member model. Furthermore, building on successful work in the literature, our first model prototype still relies

on surrogate variables which are known to be associated with thunderstorm occurrence. This initial model is referred to as SALAMA 0D and gives rise to the publication P1.

In a second step, we replace the surrogate variables of our initial model by vertical profiles of the NWP state variables, which yields the deep neural network SALAMA 1D and the publication P2. This work allows us to address RQ 2:

RQ 2: *Can a deep neural network model, which is given the flexibility to discover—on its own—the representations needed to infer thunderstorm occurrence, outperform a conventional ML model relying on human-engineered predictors, despite constraints on training set size and high computational resource requirements?*

A follow-up issue which immediately arises concerns the interpretability of ML output (Flora et al., 2024; Yang et al., 2024). Arguably, insight into how ML models arrive at their predictions is crucial for end users to put trust into them (Dramsch et al., 2025). Therefore, we address RQ 3:

RQ 3: *To what extent are the patterns identified by our deep neural network model physically interpretable?*

SALAMA 1D is still a single-member model. Therefore, in a third step, we process all members of the NWP ensemble simultaneously by applying SALAMA 1D to all members and computing the ensemble mean. This evaluation mode, to which we refer as SALAMA 1D-EPS, gives rise to the publication P3, in which we address RQ 4:

RQ 4: *By how much and why does skill increase when averaging over an NWP ensemble of thunderstorm forecasts?*

Finally, with our trained ML model in place, we explore the net benefit of ML-based thunderstorm forecasts compared to raw NWP output. Specifically:

RQ 5: *Which factors affect the decay of ML model skill with lead time? To what extent can ML counteract skill decays resulting from the increase of NWP uncertainty?*

This thesis is structured as follows. Chapter 2 summarizes the relevant background on thunderstorms, NWP, and ML. We present our ML framework in Chapter 3 and our data preprocessing pipeline in Chapter 4. The results are divided into three chapters. Namely, the findings related to SALAMA 0D, SALAMA 1D, and SALAMA 1D-EPS, are presented and discussed in Chapters 5, 6, and 7, respectively. Chapter 8 summarizes our responses to the RQs raised in this thesis, concludes our work, and proposes further research avenues.

2 FOUNDATIONS

This chapter summarizes the main foundational concepts at a level required to follow this thesis. We start by providing essential meteorological background on thunderstorms. We then introduce numerical weather models, which produce the main data source of this work. Finally, we close with an outline of binary classification using neural networks, which will later form the basis of our own machine-learning framework.

2.1 THUNDERSTORM FUNDAMENTALS

The World Meteorological Organization (WMO) defines a thunderstorm as "[o]ne or more sudden electrical discharges, manifested by a flash of light (lightning) and a sharp or rumbling sound (thunder)."[1] How do these electrical discharges come about? Why are thunderstorms so often accompanied by heavy rain or even hail? And how does warm and sunny summer-day weather suddenly turn into dark skies with lightning in the first place? In this chapter, we discuss these questions by explaining the basic physics of how thunderstorms are formed in Earth's atmosphere and how the associated hazards are produced.

The atmosphere—a fluid

We begin by introducing the physical framework used to describe the state of the atmosphere throughout this thesis. Apart from occasional liquid and solid particles, such as raindrops, hail, or aerosols, the atmosphere essentially consists of what is commonly referred to as *air*, a mixture of many different gases, the two most abundant of which are nitrogen and oxygen (Wallace et al., 2006, p. 8). The atmosphere also contains water vapor, the concentration of which depends on location, height, and time. The thermodynamic conditions in the atmosphere are such that water vapor can undergo phase transitions. What may sound trivial turns out to be of paramount importance for thunderstorm formation, as we will discuss shortly.

The gaseous nature of the atmosphere lends itself to a two-component (dry air + water vapor) fluid-mechanic treatment, in which the atmospheric state is described by certain scalar functions and vector quantities of space and time (e.g., Vallis, 2017). For dry air, these atmospheric variables are pressure p, temperature T, density

1 https://cloudatlas.wmo.int/en/thunderstorm.html, last access: 2025/05/20

westerly: eastward
southerly: northward

ρ, as well as velocity $\boldsymbol{v}$. In cartesian coordinates on a tangent plane on the sphere, $\boldsymbol{v}$ decomposes into the vertical velocity w, and two horizontal components, namely the *zonal*, or *westerly*, velocity u, and the *meridional*, or *southerly*, velocity v. These six variables are coupled to each other by three momentum equations, two equations reflecting the conservation of energy and mass, and one equation of state, such as the ideal gas law. Additional constituents, like water vapor, liquid water, or ice, require separate variables (and equations of state). For instance, to account for water vapor, one can introduce the *water vapor mixing ratio* $r_v \equiv \rho_v/\rho_d$ (throughout this chapter, the subscript "v" denotes water vapor quantities while "d" is used for dry-air quantities). In fact, there are several often-used variables to describe moisture. Closely related to r_v is *specific humidity* $q \equiv \rho_v/(\rho_d + \rho_v)$. Alternatively, *dew-point temperature* T_d refers to the temperature to which air needs to be cooled (at constant pressure and water vapor mixing ratio) to become saturated. If we denote saturation water vapor mixing ratio by r_{vs}, then the quotient r_v/r_{vs} is called *relative humidity* (Markowski et al., 2010, p. 12).

Air density decreases with height, causing 80 % of the atmospheric mass to lie within the lowest $10-15\,\mathrm{km}$ above the ground. This lowest atmospheric layer, referred to as the *troposphere*, exhibits a temperature decrease γ with height (*lapse rate*), of

$$\gamma \equiv -\frac{dT}{dz} \approx 6.5\,\mathrm{K\,km^{-1}} \tag{2.1}$$

on average (Wallace et al., 2006, p. 11), and hosts most of the atmospheric processes related to thunderstorm activity. The upper limit of the troposphere, referred to as *tropopause*, is marked by a temperature inversion caused by the absorption of ultraviolet radiation by ozone molecules. Tropopause height varies with location and time and is defined in practice by the lowest level where the lapse rate drops to $\leqslant 2\,\mathrm{K/km}$, with the average lapse rate within 2 km above remaining $\leqslant 2\,\mathrm{K/km}$, (World Meteorological Organization, 1957).

Buoyancy and parcel theory

With all atmospheric variables in place, we start approaching the issue of how thunderstorms form. Thunderstorms are associated with the atmospheric process of *deep (moist) convection* (Stevens, 2005). Generally, convection refers to the motion of fluids due to density changes. For instance, in the atmosphere, air parcels which are warmer than their environment will rise due to their reduced density. As we shall see, the fact that water vapor can undergo a phase transition under atmospheric conditions is key in releasing a density-driven instability capable of lifting air to great heights.

Next, we derive an equation of motion for the vertical displacement of parcels, relying on Markowski et al. (2010, pp. 19–21, 41–43). This

equation will later allow us to perform a stability analysis, which will in turn uncover the mechanism releasing deep convection. We begin by recognizing that, since Earth rotates, the atmosphere is subject to inertial forces, namely the Coriolis force and the centrifugal force. However, we will assume these terms to be negligible for vertical motions in what follows. We will also neglect viscous effects. The momentum equation (i.e., Newton's second law, expressed for a fluid) in the vertical then reads

$$\rho \frac{dw}{dt} = -\frac{\partial p}{\partial z} - \rho g, \tag{2.2}$$

where g denotes gravitational acceleration. Equation (2.2) can be recast into a more intuitive form, highlighting how variations in density are associated with a force, namely buoyancy. To see this, we develop p and ρ around a hydrostatic ($dw/dt = 0$) base state depending only on height z; i.e., we set $p(\mathbf{x}, t) = \overline{p}(z) + p'(\mathbf{x}, t)$, $\rho(\mathbf{x}, t) = \overline{\rho}(z) + \rho'(\mathbf{x}, t)$ with

$$\frac{\partial \overline{p}}{\partial z} = -\overline{\rho} g. \tag{2.3}$$

Subtracting Eq. (2.3) from Eq. (2.2) and rearranging, we obtain

$$\frac{dw}{dt} = -\frac{1}{\rho}\frac{\partial p'}{\partial z} - \frac{\rho' g}{\rho}. \tag{2.4}$$

The first term describes the perturbation pressure gradient force, while the second term denotes buoyancy B. Using the ideal gas law, and neglecting the effect of water vapor (and other hydrometeors, such as cloud water or ice) on buoyancy, one can express B as

$$B \approx \frac{T'}{\overline{T}} g, \tag{2.5}$$

where we expressed temperature, as well, in terms of a perturbation around the hydrostatic base state.

In the following, we consider a location in which the troposphere is in the above hydrostatic base state. At this location, let us consider a parcel of air which is located at the height z_0 and in equilibrium with its environment. We will now study in some detail vertical displacements Δz of this parcel. For this, we assume the parcel to be sufficiently small such that it does not affect the air in its environment during parcel displacement. This allows us to neglect the perturbation pressure gradient force in Eq. (2.4). Therefore, and as $w = dz/dt$, parcel displacement is described by

$$\frac{d^2 \Delta z}{dt^2} = \frac{T - \overline{T}}{\overline{T}} g, \tag{2.6}$$

where we regard T as the parcel temperature and $\overline{T}$ as the environmental temperature. The set of simplifying assumptions which went into deriving Eq. (2.6) is referred to as *parcel theory* in the literature (Emanuel, 1994; Markowski et al., 2010).

Infinitesimal parcel displacements

In order to identify buoyancy-related instabilities which may initiate deep convection, we will first linearize Eq. (2.6) to study under which conditions parcels are unstable with respect to small vertical displacements. For small displacements, we can perform a first-order Taylor series expansion,

$$T(z) = T_0 - \Gamma_p \Delta z, \quad \overline{T}(z) = T_0 - \gamma \Delta z, \tag{2.7}$$

where T_0 denotes the temperature of the parcel and the environment at height z_0 while the vertical temperature gradients Γ_p and γ are referred to as the parcel lapse rate and environmental lapse rate, respectively. We assume particle displacement to occur adiabatically, i.e., without heat exchange with the environment. If the parcel is unsaturated (i.e., its relative humidity is below 100 %), adiabatic ascent occurs at the *dry-adiabatic* lapse rate, which can be shown to be $\Gamma_d \equiv g/c_p \approx 9.8\,\mathrm{K\,km^{-1}}$. Here, c_p denotes the specific heat of air for a constant pressure process. On the other hand, if the rising parcel is saturated, latent heat is released, which is why the *moist-adiabatic* lapse rate Γ_m is smaller than Γ_d. In contrast to its dry-air counterpart, Γ_m varies as a function of r_v; typical values are between $5\,\mathrm{K\,km^{-1}}$ and $9\,\mathrm{K\,km^{-1}}$ (Markowski et al., 2010, p. 14).

Inserting Eq. (2.7) into Eq. (2.6), we obtain

$$\frac{d^2\Delta z}{dt^2} = \frac{\gamma - \Gamma_p}{T_0 - \gamma \Delta z} g \Delta z \approx -g \frac{\Gamma_p - \gamma}{T_0} \Delta z. \tag{2.8}$$

Apparently, the parcel experiences a positive restoring force (i.e., is stable with respect to small vertical displacements) for $\gamma < \Gamma_p$. As there are two types of parcel lapse rates, γ falls into one of three categories:

- $\gamma > \Gamma_d$: the environmental lapse rate is *absolutely unstable*.
- $\Gamma_m < \gamma < \Gamma_d$: the environmental lapse rate is *conditionally unstable*; i.e., it is stable with respect to dry-adiabatic ascent, but unstable with respect to moist-adiabatic ascent.
- $\gamma < \Gamma_m$: the environmental lapse rate is absolutely stable.

Absolutely stable environmental lapse rates allow for stable stratification of air, preventing any convective overturn. As for absolutely unstable lapse rates, these may occur, for instance close to the ground in the boundary layer as a consequence of high solar radiation. In these cases, convection is indeed triggered, restoring stability. However, since absolutely unstable layers of air are redistributed as they form, they cannot be the driving instability of deep convection.

This leaves us with conditionally unstable lapse rates. Unless the environment happens to be saturated at z_0, conditionally unstable

lapse rates are actually not unstable towards infinitesimal displacements either. On the other hand, if a parcel were forced to rise a *finite* distance against the restoring force until the parcel becomes saturated, it may become unstable. Indeed, as it turns out, we need to extend our discussion to finite-amplitude displacements to understand the initiation of deep convection.

Finite-amplitude parcel ascent

While infinitesimal vertical parcel displacements do not suffice to release an instability capable of producing deep convection, finite displacements do. To see this, let us consider the vertical profile of environmental temperature displayed in Fig. 2.1. Since pressure monotonously decreases with height in a hydrostatic environment, we opt for measuring height in terms of pressure, as is common practice in meteorology (e.g., Markowski et al., 2010; Vallis, 2017). Note in particular that the vertical axis is scaled logarithmically with pressure while the horizontal temperature axis is skewed by 45°, giving rise to a *skew*-T *log*-p *diagram*. Consider now a parcel located close to the surface. This parcel is stable with respect to infinitesimal displacements, but we assume now that the particle is forced to ascent (we will discuss possible sources of lift later). The parcel rises dry-adiabatically until it saturates. Parcel temperature as a function of height is equally shown in Fig. 2.1. The height at which saturation occurs is referred to as the parcel's *lifting condensation level (LCL)*. From there, the parcel rises moist-adiabatically. Remarkably, for this environmental temperature profile and parcel, there exists a height at which parcel temperature exceeds the temperature of its environment, at which point the parcel becomes positively buoyant. This height is called *level of free convection (LFC)*. While external forcing was needed to raise the parcel to its LFC, the parcel now continues to rise on its own. It remains positively buoyant until it reaches its *equilibrium level (EL)*. This instability with respect to finite-amplitude parcel displacements, which enables air to be lifted to heights as great as the tropopause, gives rise to deep convection. Henceforth, for simplicity, the term "convection" will be used to refer specifically to deep convection.

The energy (per parcel mass) released at the LFC is referred to as *convective available potential energy (CAPE)*, and is given by

$$\mathrm{CAPE} = \int_{\mathrm{LFC}}^{\mathrm{EL}} B \, \mathrm{d}z = \int_{\mathrm{LFC}}^{\mathrm{EL}} g \frac{T(z) - \overline{T}(z)}{\overline{T}(z)} \, \mathrm{d}z. \tag{2.9}$$

Typically, CAPE values $\lesssim 1000\,\mathrm{J\,kg^{-1}}$ are considered small while values $\gtrsim 2500\,\mathrm{kg}$ are considered large (Markowski et al., 2010, p. 33). One can show that CAPE is proportional to the area enclosed by T and $\overline{T}$ between the LFC and the EL in a skew-T log-p diagram (Emanuel, 1994, p. 171). Similarly, the energy barrier (per parcel mass) which a

parcel at height z_0 needs to overcome to reach its LFC is referred to as convective inhibition (CIN), and is given by

$$\mathrm{CIN} = -\int_{z_0}^{\mathrm{LFC}} B \, dz = -\int_{z_0}^{\mathrm{LFC}} g \frac{T(z) - \overline{T}(z)}{\overline{T}(z)} \, dz. \tag{2.10}$$

CIN values $\lesssim 10\,\mathrm{J\,kg^{-1}}$ are typically considered small, whereas values $\gtrsim 50\,\mathrm{J\,kg^{-1}}$ are considered large.

CAPE and CIN for the parcel described above are annotated in Fig. 2.1. If the two variables are computed for a parcel starting from $25 - 50\,\mathrm{hPa}$ above ground (rather than directly from the ground), we refer to them as *mixed-layer* CAPE and CIN. These aim to account for parcel dilution with dry environmental air (Markowski et al., 2010, pp. 192–193). Throughout this thesis, we assume CAPE and CIN to be associated with mixed-layer parcels starting at 25 hPa above the ground unless stated otherwise.

Ingredients for convection initiation

Forecasters traditionally consider three aspects, or ingredients, when predicting convection initiation. This is sometimes called *ingredient-based methodology* (Doswell et al., 1996). We will put these three ingredients into the context of what we have discussed so far.

The first ingredient for convection initiation is *instability*. As we have seen, this aspect is associated with the presence of CAPE. Generally, the presence of CAPE requires sufficiently high mid-tropospheric lapse rates. The second ingredient is *lower tropospheric moisture*. Abundant low-level moisture allows a parcel to quickly reach its LCL, such that it can continue to rise moist-adiabatically to its LFC. Conversely, if the lower troposphere is too dry, an LFC might not even exist. The third ingredient refers to a source of *lift*. Lift is required to overcome any existing CIN. Lift may be provided by air mass boundaries. For instance, the cold (and, hence, relatively dense) air mass behind a cold front may advance underneath the warmer air mass, causing the warmer air mass to rise. Furthermore, orography (i.e., hills, mountainside) can initiate convection through upward-blowing winds.

There are also processes to consider which act on the *synoptic* scale; i.e., on length scales of multiple thousand kilometers. While synoptic-scale processes are typically too slow to actively trigger convection, they can prepare the environment for convection by modulating CAPE and CIN. For instance, environmental lapse rates and moisture can be advected by the mean wind and, thereby, help destabilizing the middle troposphere and moistening the environment. Synoptic ascent contributes to reducing CIN.

It is important to stress that while high values of CAPE and low values of CIN are certainly favorable for convection to be initiated, they do not guarantee convection initiation, even if sufficient lifting

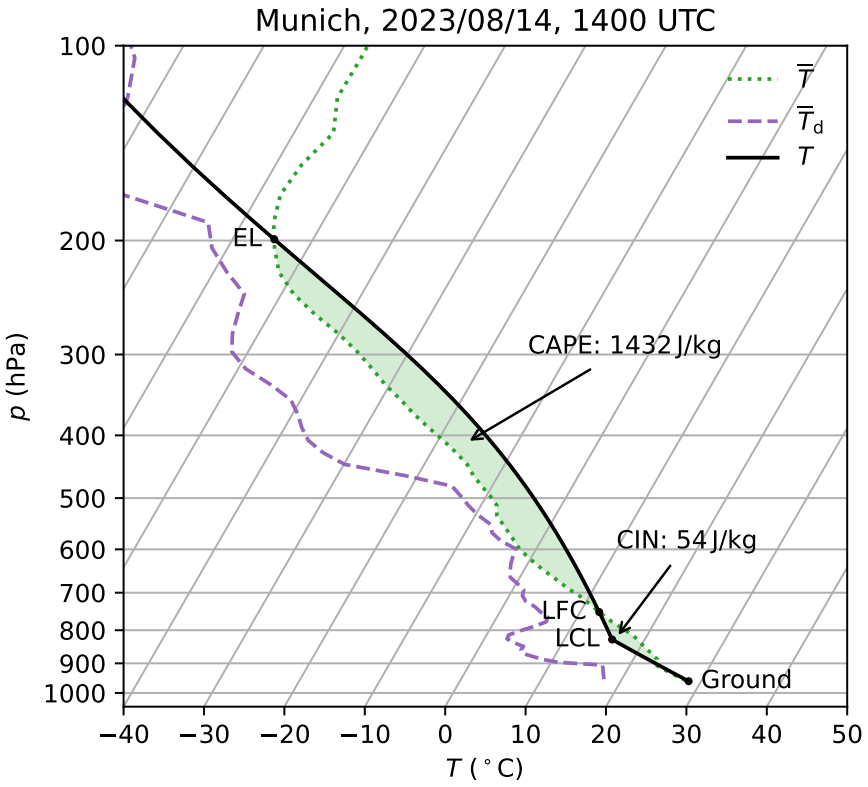

Figure 2.1: Trajectory of a parcel forced to rise from the ground in a skew-T log-p diagram. The parcel rises dry-adiabatically until reaching its LCL, at which point it ascends moist-adiabatically, cooling down more slowly than the environment (we assume a *pseudo-adiabatic* process, in which liquid water is removed as soon as it forms; the corresponding formula is given in, e.g., Emanuel (1994, p. 131)). Parcel temperature T starts to exceed environmental temperature $\overline{T}$ at a height marked by the LFC. The parcel stays positively buoyant until reaching its EL. While enough CAPE is available to sustain convection, a strong lift would be required to overcome the present amount of CIN. As a matter of fact, convection failed to be initiated that day. The vertical profiles of $\overline{T}$ and dew-point temperature $\overline{T}_d$ have been obtained from a 2-hour forecast of the sixth member of the ICON-D2-EPS model (introduced in Chapter 4).

is provided (Markowski et al., 2010, pp. 192–195). This is partly due to the simplifications made in parcel theory, namely the neglect of pressure perturbations, which reduce the energy available for updrafts, as well as the neglect of moisture and condensate loading, which are buoyancy-altering. Moreover, we have not taken into the account the entrainment (i.e., mixing) of dry environmental air into the parcel, which reduces the parcel's buoyancy. CAPE and CIN also depend on the parcel's initial height. Hence, while surface-based convection is prevented by high CIN, *elevated* convection atop stable layers might still occur. This illustrates the complexity associated with forecasting convection initiation.

The convective life cycle

Having understood the basic mechanism for convection initiation, we discuss now ongoing convection and its accompanying phenomena, relying on Doswell (1984). Let us, therefore, assume that convection has been initiated at a given location. As positively buoyant air is accelerated to its EL, condensation occurring in the rising parcels produces a towering cumulus cloud of great vertical extent. If the instability persists, the cloud continues to grow until reaching the tropopause. There, the temperature inversion causes the cloud top to become characteristically wide and flat like an anvil. At this stage, the cloud is referred to as *cumulonimbus*.

The updraft in a cumulonimbus cloud also lifts water droplets, which grow and cool below their freezing point. These supercooled droplets may freeze to ice crystals or graupel, or grow to hail. The abundant condensation causes precipitation to start and then to intensify. Precipitation partly evaporates in unsaturated air, causing it to cool. This results in a downdraft of air, which is intensified by the drag of the precipitating droplets. It is at this *mature* stage, where a steady updraft and downdraft coexist, that lightning activity is maximal. Lightning refers to electrical discharges within a cloud, or between the cloud and the ground. The charging of the cloud is hypothesized to be driven by collisions between ice particles and larger graupel particles (Saunders, 2008; Dwyer et al., 2014). Lightning causes nearby air to heat up and rapidly expand, resulting in a sound perceived as *thunder*.

The air transported in the downdraft spreads out horizontally when reaching the surface, creating a *gust front* of relatively cold wind. As this *cold pool* continues to grow horizontally, it may provide the required lift to trigger convection nearby. The cold pool eventually reaches the updraft region, replacing the warm and moist low-level air. As the updraft can no longer be sustained, convection stops.

What we have outlined above is actually the life cycle of single thunderstorm cells, which are rather short-lived with life times of the order of half an hour (Markowski et al., 2010, pp. 207–208). When gust fronts

repeatedly initiate new cells (*multi-cell convection*), a thunderstorm can persist for multiple hours. A third organization type of thunderstorms is referred to as *supercellular convection*. A supercell is a large isolated cell with a rotating updraft and can equally last for multiple hours.

We mention in closing that convection is not always organized in an isolated manner. A widespread group of thunderstorms spanning a contiguous area of the order of 100 km in at least one horizontal direction is referred to as *mesoscale convective system (MCS)*. MCSs may develop through cold-pool merging of multiple isolated thunderstorms, or arise as a whole after convection initiation (Houze Jr., 2004).

2.2 NUMERICAL WEATHER PREDICTION

Numerical weather prediction (NWP) provides the primary data source of input for our ML models. Therefore, in this section, we give an overview of the NWP aspects which we deem key for following the remainder of this thesis.

The basic idea

We have already stated in Section 2.1 that the atmosphere can be modeled as a (rotating, two-component) fluid characterized by a set of state variables, such as pressure, temperature, and vertical velocity, which are functions of space and time. The state variables are coupled to each other by the laws of physics, namely

- the Navier-Stokes equations (including the effect of Earth's rotation), which describe the effect of pressure gradients, gravity, inertial and viscous forces on fluid velocity,
- the mass continuity equation reflecting mass conservation,
- the first law of thermodynamics reflecting energy conservation,
- equations of state for dry air and for each water phase.

Given appropriate initial and boundary conditions, one may solve this set of partial differential equations, obtaining a prediction of the future atmospheric state. The equations can typically be solved only numerically, giving rise to NWP, which has steadily improved over the past century in terms of forecast skill (Bauer et al., 2015).

Discretization

The governing equations of the atmosphere need to be discretized horizontally, vertically, and in time, to be solved numerically. To this end, NWP models define a spatial grid with a certain horizontal

grid spacing and several vertical levels. While discretization errors decrease at smaller grid spacings, computation times increase, which is why it is currently unfeasible to, e.g., run kilometer-scale *global* NWP models operationally. On the other hand, it is often sufficient to obtain high-resolution forecasts only for a specific region of interest. One possible way of achieving this is to simply restrict the model domain to the region of interest. These *limited-area models (LAMs)* allow for model runs on a higher spatial resolution compared to global models. Boundary conditions are provided by a *driving model*, which is usually a global model (Bauer et al., 2015). An alternative way of obtaining regional high-resolution forecasts is through *nesting*, by which a refined subregion with smaller grid spacings (a *nest*) is added to a coarse global model's grid (Reinert et al., 2020).

Parameterization of physical processes

When discretizing the atmospheric equations, one needs to recognize that any processes occurring at scales smaller than grid-scale cannot be resolved. Examples include turbulent transport near the surface or drag effects from subgrid-scale orography. In addition, there are physical processes which are simply not represented in the fluid equations, such as reflection and absorption of electromagnetic radiation by the atmosphere *(radiative transfer)*, or processes determining cloud cover (Prill et al., 2024). The effect of these processes on the atmosphere's state variables is modeled by *parameterizations* (Bauer et al., 2015; Yano et al., 2018).

If the horizontal grid spacing of an NWP model is of the order of tens of kilometers or more, deep convection has to be parameterized to account for the corresponding bulk transport of mass, momentum, heat, and moisture (Emanuel, 1994, Chapter 16). Examples of operational parameterization schemes for cumulus convection are the Tiedtke-Bechthold scheme (Bechtold et al., 2014), or the Kain-Fritsch scheme (Kain et al., 1990). However, in this thesis, we will work with forecasts of a *convection-permitting* NWP model. The grid spacing of such a model is sufficiently small so that deep convection can be triggered without a parameterization (P. Clark et al., 2016).

Obtaining an initial state from observations

Once the physical equations governing the time evolution of the atmosphere are identified and discretized, the problem at hand boils down to solving an initial-value problem, which might seem conceptually straightforward. However, observational data is too sparse to provide full initial conditions for all atmospheric state variables on the entire spatial grid, horizontally and vertically. This issue has given rise to *data assimilation* techniques, which produce a physically-consistent

atmospheric initial state (called *analysis*) from available observations and a previous model run (Bauer et al., 2015).

Once the analysis is available, future atmospheric states (the *forecasts*) are computed by numerically solving the initial value problem. The time difference between when the analysis is valid and when a forecast is valid is referred to as *lead time*. As the computations associated with determining the analysis are extensive, the valid times of the first few forecasts of an operational NWP model lie in the past by the time at which they become available. In fact, it usually takes $1-2\,\mathrm{h}$ for an operational NWP model to catch up with real-time.

Ensemble systems

We close by discussing the origin and quantification of NWP forecast uncertainty. One source of uncertainty is given by the fact that the time-evolution of a nonlinear system, despite being deterministic, can sensitively depend on the system's initial conditions—a finding which has given rise to *chaos theory* (Lorenz, 1963). The chaotic nature of Earth's atmosphere has been extensively studied (Thompson, 1957; Lorenz, 1969; Palmer et al., 2014; Craig et al., 2021) and has become known to a broader audience as the *butterfly effect*. In our case, this implies that uncertainties in the NWP analysis will grow in time. Further uncertainties may arise from physical parameters in the NWP model parameterizations, or from any boundary conditions. These uncertainties translate into a forecast uncertainty which grows with lead time, ultimately limiting predictability. A major breakthrough in the history of NWP is marked by the advent of methods capable of estimating forecast uncertainty, namely *ensemble methods* (Bauer et al., 2015).

The general idea of ensemble forecasting systems is reminiscent of Monte-Carlo simulations: Instead of just one forecast, ensemble forecasting systems produce multiple physically consistent forecasts (the ensemble *members*). This is done by running the NWP model several times, each time with "slightly" different initial and boundary conditions and model parameters. Importantly, these perturbations are chosen in such a way that they reflect the corresponding uncertainties. In practice, the process of generating an ensemble of perturbed initial conditions has been integrated into data assimilation (Bauer et al., 2015; Reinert et al., 2020). In an operational setting, the number of ensemble members is limited by computational resources, typically to a few tens of members.

2.3 BINARY CLASSIFICATION USING ARTIFICIAL NEURAL NETWORKS

As was pointed out, we will devote ourselves in this thesis to identifying thunderstorms in NWP data. Specifically, given some NWP output, e.g., the 5-hour forecast of the atmospheric state variables at Munich, Germany, our task will be to determine whether that output is associated with thunderstorm occurrence or not. The task of categorizing elements into one of two classes is referred to as *binary classification*, and encompasses a vast number of use cases, for instance,

- is a given e-mail spam or not?
- given a patient's mole, is it benign or skin cancer?
- will a given borrower return their loan?

In this section, we will formally introduce binary classification. While motivated by thunderstorm identification, we keep the setting general so that our discussion remains valid for other use cases. We will also introduce artificial neural network models and explain how they can be employed to construct binary classification models in a data-driven manner. Finally, we discuss verification scores to quantify the skill of binary classification models.

Definition of binary classification models

Given a pair of classes which we denote by "A" and "B", we formally define a binary classification model as a function $C : \mathbb{R}^{N_f} \to [0, 1]$ which maps a sample $\boldsymbol{x}$ of real-valued numbers, the *features*, to a number between zero and one (Goodfellow et al., 2016, p. 98). The sample $\boldsymbol{x}$ characterizes the element to be classified. The output $C(\boldsymbol{x})$ is interpreted as the probability that $\boldsymbol{x}$ belongs to class "A". The probability that $\boldsymbol{x}$ belongs to class "B" is then $1 - C(\boldsymbol{x})$. Hence, rather than to assign a class label (i.e., "A" or "B"), a binary classification model as defined above provides probabilistic scores for each class. This enables C to convey the degree of confidence associated with classifying $\boldsymbol{x}$. If required, a class label could be assigned in this probabilistic setting by establishing a decision threshold (e.g., an e-mail is treated as spam if the binary classifier is at least "60 % certain" that it is spam).

Without loss of generality, we tailor the discussion to class "A" instead of class "B".

Constructing a binary classification model from data

Now that we know what a binary classification model is, we discuss next how to construct a specific model $C : \mathbb{R}^{N_f} \to [0, 1]$ for a given use case from data. To be precise, imagine that we have collected N *examples* $(\boldsymbol{x}^{(j)}, y^{(j)})_{j=1,\dots,N}$, each consisting of a sample $\boldsymbol{x}^{(j)} \in \mathbb{R}^{N_f}$ and a label $y^{(j)} \in \{0, 1\}$ (1: class "A", 0: class "B"). Furthermore, we assume

C to take a certain parameterized form; i.e., $C = C(\cdot\,;\theta)$. The task of setting the functional form of C and adjusting the corresponding parameters θ to optimally describe the given data is addressed by *machine learning (ML)* methods. ML is a subfield of artificial intelligence and exploits optimization methods and data to build statistical models, such as C (Goodfellow et al., 2016, pp. 2–3). The process of adjusting the parameters θ is commonly referred to as *training* an ML model. In our case, the training set $(\xi^{(j)}, y^{(j)})_{j=1,\dots,N}$ is *labeled*; i.e., the ground truth $y^{(i)}$ is provided for each input sample. This puts us into the branch of *supervised learning*. As the label $y^{(j)}$ takes on discrete and finite values, we are dealing with a *classification* task (in contrast to *regression* tasks, which handle continuous labels). In our setting, the possible outcomes of $y^{(j)}$ are restricted to two values, hence the term *binary* classification.

We will later introduce artificial neural network models as a means to systematically parameterize C in terms of θ. For now, however, considering C as given, we explain how to adjust its parameters θ. Consider a sample $\xi^{(j)}$ from the training set, as well as the corresponding label $y^{(j)}$. We assume now that C, with the parameters θ fixed to certain values, is the true statistical model describing the example $(\xi^{(j)}, y^{(j)})$. In this case, we can meaningfully ask: what is the probability of observing the label $y^{(j)}$, given the sample $\xi^{(j)}$? By definition of C, this probability L is given as

$$L\left(\theta|\xi^{(j)}, y^{(j)}\right) \equiv \begin{cases} C(\xi^{(j)};\theta) & \text{if } y^{(j)} = 1, \\ 1 - C(\xi^{(j)};\theta) & \text{if } y^{(j)} = 0. \end{cases} \tag{2.11}$$

Note that we have made explicit in the notation that we regard L as a function of θ, while we consider $\xi^{(j)}$ and $y^{(i)}$ as parameters of L. In a Bayesian framework, the expression L as a function of θ is commonly referred to as *likelihood function* (Gelman et al., 2013, p. 7). It can be seen as a measure of plausibility that the example $(\xi^{(j)}, y^{(j)})$ is described by C with parameters θ.

We can now similarly ask for the probability of observing *all* labels in the training set, given the corresponding input samples and parameters θ. As our statistical model C processes each sample in the training set individually, the likelihood function $\tilde{\mathcal{L}}$ associated with all samples and labels factorizes,

$$\tilde{\mathcal{L}}(\theta) = \prod_{j=1}^{N} L\left(\theta|\xi^{(j)}, y^{(j)}\right). \tag{2.12}$$

The most plausible parameters θ, given the examples in the training set, are those which maximize Eq. (2.12) with respect to θ. Equivalently, one can minimize $\mathcal{L}(\theta) \equiv -\frac{1}{N}\log\tilde{\mathcal{L}}(\theta)$,

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{j=1}^{N} \log L\left(\theta|\xi^{(j)}, y^{(j)}\right). \tag{2.13}$$

We reiterate that the term "training", as adopted by the ML community, simply refers to identifying the parameters θ which minimize Eq. (2.13). The functions to be minimized during training are commonly referred to as *loss functions* in ML. Equation (2.13) is usually called *binary cross-entropy loss* and constitutes, as we have outlined, a natural choice of loss function for binary classification tasks. In particular, this loss function allows us to build the probabilistic interpretation of C (i.e., $C(\xi)$ referring to the probability of ξ being associated with class "A") explicitly into the ML framework.

Training details

We continue by describing how loss functions, such as $\mathcal{L}(\theta)$, are minimized in practice. $\mathcal{L}(\theta)$ is a nonlinear function of its parameters, and parameter size ranges from thousands to millions in many ML applications, which is why minimization is done numerically (LeCun et al., 2015). Minimization algorithms in ML rely on the concept of gradient descent,

$$\theta_{\text{new}} \equiv \theta - \eta \nabla_\theta \mathcal{L}(\theta), \tag{2.14}$$

in which adjustments in parameter space occur towards the direction of steepest descent (if $\mathcal{L}(\theta)$ is considered as a high-dimensional landscape). The step size η is usually called *learning rate*. Gradient descent in this form can become rather expensive as each parameter update involves computing gradients for all examples in the training set. Therefore, in practice, one typically partitions the training set into smaller, random, subsets, which then fit into the memory of the available compute hardware. Each subset, called *minibatch*, is used to estimate the gradient in Eq. (2.14) and update θ. This approach is referred to as *stochastic gradient descent* (SGD; Bishop, 2006, p. 240). After each minibatch has been used once, which defines an *epoch*, the training set is again randomly partitioned and one repeats SGD. Modern SGD algorithms like AdaGrad (Duchi et al., 2011) or Adam (Kingma et al., 2014) keep track of past parameter adjustments, and apply averaging and adaptive learning rates to improve convergence.

Provided that our model C is sufficiently complex in terms of parameter size (and assuming that the predictors ξ contain enough relevant information for the classification task), C will be able to classify the examples in the training set at a certain level of skill (we will discuss measures of skill later in this chapter). However, C is only useful if it is also skillful at classifying *unseen* data. To check for generalization, available data can be split before training. In practice, data is often actually split into three separate data sets, so that training is done using one set while generalization is measured using a test set. A third set, the validation set, can be used to monitor training progress and for *hyperparameter tuning*. The latter refers to the adjustment of

model or training parameters other than θ, such as learning rate, or minibatch size.

If C performs significantly worse on the test set than it does on the training set, *overfitting* has occurred. This issue tends to arise if the amount of training data is not sufficient to constrain the parameters of the ML model. Methods to address overfitting are referred to as *regularizations*. An example is given by *early stopping*, which relies on tracking training skill and validation skill as a function of epoch: if validation skill no longer improves, or even deteriorates, although training skill continues to improve, training is stopped.

Neural networks and deep learning

Having discussed the training of an ML model C, we now discuss one particular means to parameterize C, namely artificial neural network models, or neural networks, in short. These arguably constitute the most ubiquitous type of ML models employed in science and engineering (LeCun et al., 2015). We restrict our discussion to *feedforward* neural networks, as these are conceptually the most fundamental ones.

A feedforward neural network for binary classification is a function $C : \mathbb{R}^{N_f} \to (0, 1)$ modeling the relationship between the predictors $\xi \in \mathbb{R}^{N_f}$ and the corresponding probability of occurrence of class "A". The simplest feedforward neural network consists of a single unit of what is sometimes referred to as *single-layer perceptron*, or *artificial neuron* (Goodfellow et al., 2016, pp. 12–14, 164–165),

$$C(\xi; \theta) = \varphi\left(\theta_0 + \sum_{i=1}^{N_f} \theta_i \xi_i\right). \tag{2.15}$$

This parameterization features a linear combination of the input plus an offset θ_0, as well as a non-linear function $\varphi : \mathbb{R} \to (0, 1)$ called *activation function*, which provides a mapping to probability-like output. The thusly parameterized model has $N_f + 1$ parameters θ_i, $i = 0, ..., N_f$. These parameters are often referred to as *weights*, though the offset θ_0 is sometimes called *bias*.

It is useful to introduce at this point a graphical representation of the model's *architecture*, which we show in Fig. 2.2. The vector-valued input is depicted as vertically stacked nodes, to which we refer as *input layer*. Equivalently, there is one node in the *output layer* representing the final result. Arrows leading to the node indicate which nodes in the input layer contribute.

More complex feedforward neural networks are obtained by combining multiple artificial neurons. An example is given in Fig. 2.3. The input layer connects to five artificial neurons, each of which has its own set of parameters and a custom activation function. Two important activation functions are shown in Fig. 2.4. Each neuron connected to the input layer outputs one number, leading to five nodes in what

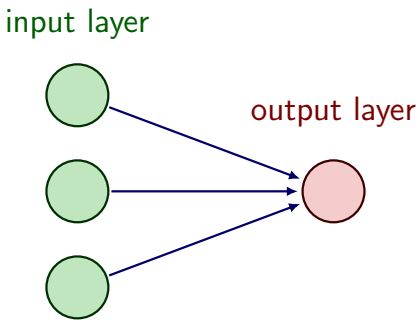

Figure 2.2: Architecture of a simple feedforward neural network consisting of a single artificial neuron.

is called a *hidden layer*. The nodes in the first hidden layer then feed to a subsequent layer of artificial neurons, giving rise to a second hidden layer. Finally, a last neuron maps to the output layer. Importantly, the repeated use of activation functions enables the representation of complex non-linear relationships between input and output.

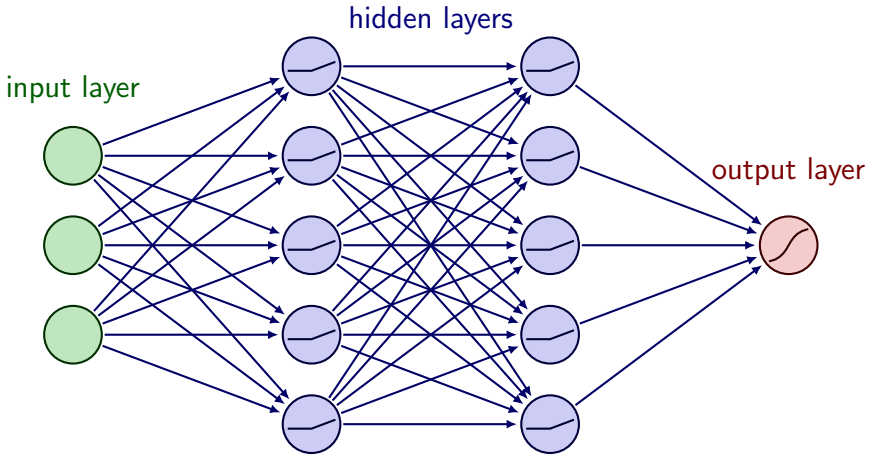

Figure 2.3: Architecture of a feedforward neural network with two hidden layers consisting of five nodes each. The symbols inside the nodes indicate activation functions: the neurons in the hidden layers use rectified linear units (ReLUs) as activation functions while the output neuron uses a sigmoid function (Fig. 2.4).

The term "feedforward" stems from the fact that the information content in one layer is processed and passed on only to subsequent layers (and not to preceding layers). Moreover, a feedforward neural network such as in Fig. 2.3 is referred to as *fully-connected* or *dense*, as the nodes in a given layer are connected to *all* nodes of the subsequent layer. The complexity of a (fully-connected) feedforward neural network is adjustable through the number of hidden layers, and the size of each layer; i.e., the number of nodes. Tasks involving neural networks with multiple hidden layers are referred to as *deep learning* (LeCun et al., 2015). Their hierarchical structure allows deep neural

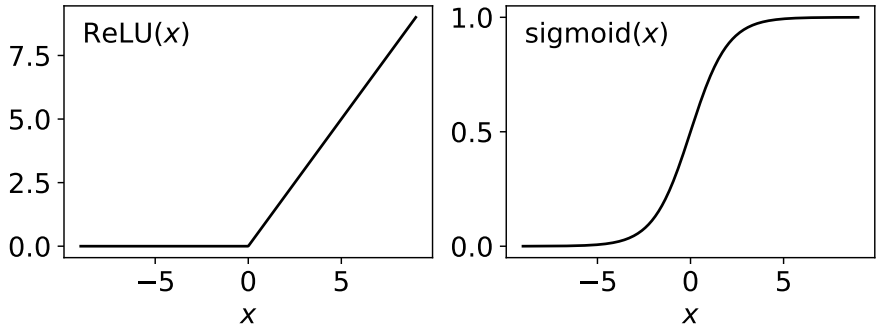

Figure 2.4: Two commonly-used activation functions, $\mathrm{ReLU}(x) \equiv \max(x, 0)$ and $\mathrm{sigmoid}(x) \equiv 1/(1 + e^{-x})$. A sigmoid activation function is typically used for the output layer to obtain probability-like output within the open interval $(0, 1)$. Conversely, the ReLU activation function is often chosen for hidden layers as it is less prone to numerical instability during training (Glorot et al., 2011).

networks to learn useful representations of the input data on their own, rather than relying on manual feature engineering using domain knowledge.

Skill evaluation metrics

Next, we discuss how to quantify a binary classification model's classification skill using appropriate metrics which are called skill scores. Skill scores are useful for multiple reasons. In many applications, the difference between an ML model's skill on the training set and on unseen data constitutes a measure of what is called *generalization error* (Goodfellow et al., 2016, p. 110). In this thesis, however, we use skill scores mostly for comparing the skill of multiple ML models which are evaluated on the same test set. Skill evaluation metrics form a vast subject with considerable research contributions by the meteorological community, who has gathered abundant forecast verification expertise (e.g., Wilks, 2019, Chapter 9). We will focus on skill scores used in severe weather forecasting and briefly motivate their respective purpose.

Often deployed metrics for evaluating classification skill include the Brier score (BS) (Brier, 1950),

$$\mathrm{BS} = \frac{1}{N} \sum_{j=1}^{N} (p^{(j)} - y^{(j)})^2, \qquad p^{(j)} = f(\xi^{(j)}). \tag{2.16}$$

The BS is negatively-oriented ("the lower, the better"). The popularity of the BS stems from it being *strictly proper* (Bröcker et al., 2007b). For this, one needs to understand that for some scores it is possible to strategically alter ("hedge") the probabilities $p^{(j)}$ to obtain a higher

score on average (Wilks, 2019, pp. 418–419). Strictly proper scores, however, penalize hedging.

Normalization with a reference Brier score BS_{ref} yields the positively-oriented Brier skill score (BSS),

$$BSS = 1 - \frac{BS}{BS_{ref}}. \tag{2.17}$$

Positive values correspond to higher-than-reference skill, with $BSS = 1$ indicating perfect skill. As a reference score, one often chooses the BS obtained for a model classifying samples randomly with a probability g, such that $BS_{ref} = \frac{1}{N}\sum_{j=1}^{N}(g - y^{(j)})^2$. The probability g is set to the fraction of samples of class "A" in the test set and is called *sample climatology* in meteorological contexts. Throughout this work, the BSS is to be understood with climatology as reference unless stated otherwise.

While the BSS directly acts on the probability outputs $p^{(j)}$ (Eq. (2.16)) of the model, a large class of classification metrics requires the conversion of probabilities to binary output first. This is done by introducing a decision threshold $\tilde{p}$. If $p \geqslant \tilde{p}$, thunderstorm occurrence for the corresponding example is deemed "true", otherwise "false". In combination with the two options from the label, there are four possible outcomes for each example. They are presented as a contingency matrix in Table 2.1.

Table 2.1: Contingency matrix for binary classification.

		Class "A" observed?	
		True	False
Class "A" predicted?	True	Hit	False alarm
	False	Miss	Correct reject

We can now use counts of the four different outcomes (evaluated on the test set) to construct skill scores. While there is an infinite number of options to do so, we will focus in this thesis on scores suitable for classification tasks with a high *class imbalance*. This means that, e.g., class "A" (the *minority* class) occurs considerably less frequently than the other (the *majority* class). In particular, we do not wish to reward the ML model for correctly classifying the majority class. Therefore, we dismiss scores which explicitly involve correct rejects.

Two suitable scores are given by the positively-oriented probability of detection (POD) and the negatively-oriented false-alarm ratio (FAR), defined by

$$POD = \frac{\text{hits}}{\text{hits} + \text{misses}}, \tag{2.18}$$

$$FAR = \frac{\text{false alarms}}{\text{hits} + \text{false alarms}}. \tag{2.19}$$

Here, e.g. "hits" refers to the number of examples in the test set which qualify as "hit" according to Table 2.1. POD is often called *recall* in the ML literature, while $1 - \mathrm{FAR}$ is also known as *precision*.

Precision and recall need to be simultaneously considered when evaluating a classification model. For instance, a model predicting the class "A" at every occasion would have perfect recall—but minimal precision. For problems with class imbalance, a popular choice of combining the two scores consists of taking the harmonic mean, which yields the positively-oriented F_1-score:

$$F_1 = \frac{2}{\mathrm{POD}^{-1} + (1 - \mathrm{FAR})^{-1}} = \frac{2\,\mathrm{hits}}{2\,\mathrm{hits} + \mathrm{misses} + \mathrm{false\ alarms}} \tag{2.20}$$

Another option of combining the contingency matrix elements is given by the positively-oriented critical-success index (CSI):

$$\mathrm{CSI} = \frac{\mathrm{hits}}{\mathrm{hits} + \mathrm{misses} + \mathrm{false\ alarms}} \tag{2.21}$$

A modification of the CSI consists of subtracting from the hit count the hits expected from climatology. The equitable threat score (ETS) reads

$$\mathrm{ETS} = \frac{\mathrm{hits} - \mathrm{hits\ by\ accident}}{\mathrm{hits} - \mathrm{hits\ by\ accident} + \mathrm{misses} + \mathrm{false\ alarms}}, \tag{2.22}$$

with

$$\mathrm{hits\ by\ accident} = \frac{(\mathrm{hits} + \mathrm{misses}) \times (\mathrm{hits} + \mathrm{false\ alarms})}{N}. \tag{2.23}$$

Finally, drawing (POD, $1 - \mathrm{FAR}$) for different decision thresholds into one diagram, one obtains a precision-recall (PR) curve. The area under the PR curve (PR-AUC) is bounded by 0 and 1 and constitutes a positively-oriented measure of skill. An example PR diagram will be given in Fig. 5.6 later.

Reliability diagrams

We close by discussing an additional versatile tool for quantifying the skill of binary classification models, namely *reliability diagrams* (e.g., Bröcker et al., 2007a). In contrast to the skill evaluation metrics introduced before, which assign a single number to a given test set, reliability diagrams offer a comprehensive overview of the full joint distribution of predictions and observations (Wilks, 2019, pp. 404–405). A reliability diagram is constructed as follows. Partitioning the range $(0, 1)$ of possible model probabilities into N_b equidistant bins of width $\Delta p \equiv 1/N_b$, we distribute the test set examples among the bins according to the assigned model probabilities. For each bin $i = 1, 2, ..., N_b$, we extract the observed relative frequency $\overline{o}_i$ of class

"A", the bin-averaged model probability p_i, and the number N_i of examples per bin. A reliability diagram, exemplified in the upper panel of Fig. 5.3 later, consists of a *calibration function* and a *refinement distribution*. The calibration function is a plot of $\overline{o}_i$ against p_i, and measures whether the model probabilities are consistent with observed relative frequencies of the target class, a characteristic known as *reliability*. A well-calibrated model exhibits a calibration function close to the 1:1 diagonal. The refinement distribution corresponds to the distribution of model probabilities. Skillful models are capable of producing well-calibrated model probabilities larger than climatology, which is referred to as *resolution* (Toth et al., 2003). Shaded bands on the calibration function correspond to the symmetric 90 % confidence interval around the median. The confidence interval is estimated by *bootstrap resampling* (e.g., Bröcker et al., 2007a): By drawing with replacement from the test set, one produces multiple resamples of the same size as the original set and considers resample-to-resample fluctuations of the calibration function.

Part II

METHODS AND DATA

3

OUR MACHINE LEARNING FRAMEWORK

In this chapter, we will present our ML approach for identifying thunderstorm occurrence in convection-permitting ensemble numerical weather prediction (NWP) forecasts. We will frame the problem in terms of the concepts introduced in Section 2.3, motivate some crucial simplifying assumptions made on the way and discuss our treatment of issues related to the rarity of thunderstorm occurrence.

3.1 PROBLEM FORMULATION

Recall the overarching aim of this thesis:

Aim of this thesis: *Development of a deep neural network model for the identification of thunderstorm occurrence in convection-permitting NWP ensemble forecasts.*

So, given an N_e-member ensemble forecast of the atmospheric state for a given horizontal NWP grid point and target time, we aim to infer the corresponding probability of thunderstorm occurrence. This defines a binary classification task with the two classes "thunderstorm occurrence" (label: 1) and "no thunderstorm occurrence" (label: 0). We encode the atmospheric state at a given grid point in terms of N_f atmospheric variables given on N_z vertical levels. Their functional relationship with the corresponding probability of thunderstorm occurrence is modeled via a deep neural network model C_{EPS}

$$C_{EPS} : \mathbb{R}^{N_e \times N_f \times N_z} \to (0,1). \tag{3.1}$$

The predictors are denoted as $\xi \equiv (\xi^{(k)})_{k=1,\dots,N_e}$ with $\xi^{(k)} \in \mathbb{R}^{N_f \times N_z}$. We call the deep neural network model SALAMA (signature-based approach of identifying lightning activity using machine learning). Once trained, and given an NWP forecast of a given lead time, SALAMA provides the corresponding thunderstorm forecast (Fig. 3.1).

salama (Finnish): bolt of lightning

We can make certain symmetry considerations to reduce model complexity. In particular, we expect C_{EPS} to be invariant with respect to ensemble member permutations $(\xi^{(1)}, \dots, \xi^{(N_e)})$. One possible way of accounting for exchange symmetry is by considering C_{EPS} as the ensemble mean of a more elementary model C *acting on the individual members*,

$$C_{EPS}(\xi) = \frac{1}{N_e} \sum_{k=1}^{N_e} C\left(\xi^{(k)}\right); \qquad C : \mathbb{R}^{N_f \times N_z} \to (0,1). \tag{3.2}$$

The function C defines a binary classification model for the identification of thunderstorm occurrence in the NWP output of the individual

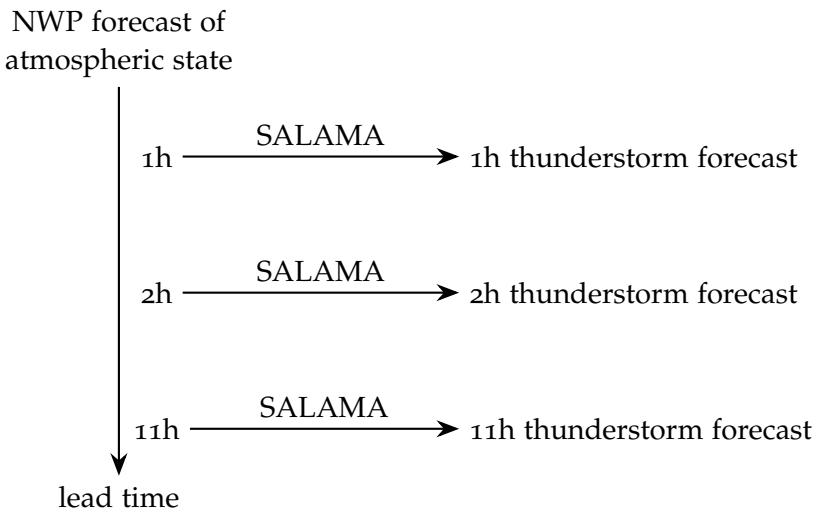

Figure 3.1: Our ML framework within the thunderstorm forecast value chain. The SALAMA model identifies thunderstorm occurrence in NWP forecasts of the atmospheric state. The same SALAMA configuration is used for all lead times.

members. Therefore, we first train a single-member deep neural network model, which we call SALAMA 1D (Chapter 6), and later use Eq. (3.2) to apply it to the entire NWP ensemble (Chapter 7). We will refer to the latter evaluation mode as SALAMA 1D-EPS.

While training a single-member model is conceptually less complex than training on the entire ensemble, processing vertical profiles of atmospheric state variables remains intricate and requires care with regard to both the model architecture design and the data preprocessing. To disentangle these two issues, we first train a simpler model to which we refer as SALAMA 0D. This model also acts on a single-member basis and uses the same preprocessing pipeline (presented in Chapter 4) as SALAMA 1D. The difference lies in the predictors $\boldsymbol{\xi} = (\boldsymbol{\xi}^{(k)})_{k=1,\dots,N_e}$, which are not vertical profiles but N_f' (zero-dimensional) atmospheric variables associated with thunderstorm occurrence in the meteorological literature,

$$C_{0D} : \mathbb{R}^{N_f'} \to (0, 1). \tag{3.3}$$

As a result, SALAMA 0D gets by with a less complex architecture compared to SALAMA 1D, allowing us to build up the entire value chain from NWP input to thunderstorm forecast output in a simpler setting before we address processing vertical profiles. In addition, since the SALAMA 0D predictors are ultimately derived from the state variables fed to SALAMA 1D, a comparison between the two models enables us to test whether a deep neural network can learn NWP input representations which are more useful for identifying thunderstorm occurrence than the derived ones from the literature.

3.2 SIMPLIFYING ASSUMPTIONS

We have—sometimes implicitly—made simplifying assumptions and choices in the presented ML framework, which we discuss now.

The first simplification concerns the horizontal and temporal extent of the predictors and the ML output. Indeed, our single-member models output the probability of thunderstorm occurrence for a single grid point, and infer said probability from predictors $\boldsymbol{x}$ on the same grid point. One could also take neighboring grid points (Zhou et al., 2019), or time series (e.g., Geng et al., 2019; Sobash et al., 2020), into account for classification. One could equally process multi-grid-point input to also produce probabilities for multiple adjacent grid points simultaneously (e.g., T. Lin et al., 2019; Geng et al., 2021). The latter corresponds to framing the task as *semantic segmentation* instead of binary classification. The reason why we opted for binary classification of grid-pointwise input was to limit model complexity.

For the same reason, we do not consider the lead time of the NWP input. Instead, as we will discuss in Chapter 4, we will train our SALAMA models on short-term ($0-2\,\mathrm{h}$) NWP forecasts, for which we expect NWP uncertainty to be minimal. Applying the trained models on forecasts with increasing lead times will then allow us to study the effect of increasing NWP uncertainty on thunderstorm identification skill.

Finally, following Ukkonen et al. (2019), we feed atmospheric variables to our models *without* providing the geographical (lat/lon) location of the corresponding grid point, in contrast to Zhou et al. (2019). Nor do we provide the time of the day, or the time of the year of the input, in contrast to Jardines et al. (2021). This might seem counterintuitive; for instance, thunderstorm occurrence is more likely in the afternoon than in the morning, which is why one might be tempted to leverage the climatological insight associated with the time of the day. However, the NWP forecasts arguably already reflect the diurnal cycle of convection (as well as its seasonal cycle, and geographical distribution). Therefore, we expect our models to be capable of reproducing temporal or spatial climatology variations solely from processing atmospheric variables.

3.3 HANDLING CLASS IMBALANCE

Thunderstorms are relatively rare events. As we shall see in Chapter 4, only a few percent of the training set samples are associated with thunderstorm occurrence if the data set is climatologically consistent. This may complicate the prediction of thunderstorm occurrence, as ML models tend to struggle with learning from unbalanced data sets (Sun et al., 2009). As a matter of fact, we verified in a preliminary training run that, when trained on a climatologically consistent data set,

SALAMA 0D would predict the majority class (i.e., no thunderstorm occurrence) at every occasion. Therefore, we undersample the majority class in the training sets of all our ML models, such that both classes appear equally frequently (*class balance*). However, we keep the data sets for testing and validation climatologically consistent in order to evaluate (and eventually apply) our ML models in a realistic (operational) setting in which thunderstorms rarely occur. Undersampling is common practice and allows us to avoid training with unfeasibly large datasets solely to ensure sufficient representation of the minority class (Hasanin et al., 2018; Mohammed et al., 2020). An issue that arises is that ML binary classifiers trained on balanced data fundamentally become miscalibrated when evaluated on a climatologically consistent test set; i.e., the produced probabilities are inconsistent with observed relative frequencies of thunderstorm occurrence. To address this, we derived a simple analytic correction to adjust the raw model outputs based on the sample climatology of the test set. We later found that this phenomenon is known in the ML literature as *prior probability shift* (Quiñonero-Candela et al., 2008, pp. 16–19). Specifically, if a model trained on balanced data is evaluated on a climatologically consistent data set, one needs to calibrate the raw model output p' using the following formula to obtain a well-calibrated probability p (Elkan, 2001; Pozzolo et al., 2015):

$$p = \frac{gp'}{gp' + (1-g)(1-p')} \tag{3.4}$$

Here, g denotes the fraction of positive examples in the climatologically consistent data set (sample climatology). We provide a derivation of Eq. (3.4) at the end of the section.

While undersampling has been applied in binary classification problems of thunderstorm forecasting, we are not aware of any related works incorporating Eq. (3.4). Reasons why this issue has not arisen in past works include:

- Model output calibration was not investigated (Jardines et al., 2021; J. Li et al., 2021).
- Model output was calibrated via statistical methods like isotonic regression (Niculescu-Mizil et al., 2005) using a validation set (Ukkonen et al., 2019; Burke et al., 2020).
- Training was done without class-balancing (Sobash et al., 2020; He et al., 2020). In these cases, the minority class appears to have been sufficiently represented in the training set. We hypothesize that these models could have been trained faster, using significantly smaller training sets (with fewer majority class samples), without loss of skill.

In contrast, the approach adopted in this thesis does not require any calibration fit after training and allows for an economical use of

computational resources. We will exemplify the validity of Eq. (3.4) for SALAMA 0D in Chapter 5.

We close by formally deriving Eq. (3.4). To this end, it is useful to revisit binary classification from a probabilistic perspective. Let a sample $\xi \in \mathbb{R}^N$ denote a realization of a continuous N-dimensional random variable Ξ, while the label $y \in \{0, 1\}$ denotes a realization of a discrete random variable Y. The joint probability distribution of (Ξ, Y) is entirely fixed by three terms, namely

- $f_{\Xi|Y}(\xi|1)$, the conditional probability density function of Ξ, given $Y = 1$,
- $f_{\Xi|Y}(\xi|0)$, the conditional probability density function of Ξ, given $Y = 0$,
- $g \equiv P(Y = 1)$, the probability of sampling $Y = 1$ with no prior knowledge, the sample climatology.

In particular, the terms can be combined via the law of total probability to express the marginal probability density function of Ξ,

$$f_\Xi(\xi) = f_{\Xi|Y}(\xi|1)g + f_{\Xi|Y}(\xi|0)(1-g). \tag{3.5}$$

Invoking Bayes' theorem, we express the conditional probability of observing $Y = 1$ if given a sample ξ,

$$P(Y = 1|\Xi = \xi) = \frac{f_{\Xi|Y}(\xi|1)g}{f_\Xi(\xi)} = \frac{1}{1 + (1-g)R(\xi)/g}, \tag{3.6}$$

where we introduced the residual function $R(\xi) \equiv f_{\Xi|Y}(\xi|0)/f_{\Xi|Y}(\xi|1)$. Equation (3.6) as a function of ξ is what an ML binary classification model $C : \mathbb{R}^N \to (0, 1)$ learns during training. However, note that Eq. (3.6) depends on sample climatology through the prefactor $(1-g)/g$. As our training set contains an increased sample climatology $\tilde{g}$ (in our work: $\tilde{g} = 1/2$), the ML model learns to output

$$C(\xi, \tilde{g}) = \frac{1}{1 + (1-\tilde{g})R(\xi)/\tilde{g}}, \tag{3.7}$$

where we made the dependence of C on $\tilde{g}$ explicit. Let now ξ be a sample from a data set with a different sample climatology $g \neq \tilde{g}$. Naively applying our model results in the miscalibrated raw model output $p' \equiv C(\xi, \tilde{g})$. The correct probability output $p \equiv C(\xi, g)$ is given by

$$p = \frac{p'}{p' + \frac{1-g}{g}\frac{\tilde{g}}{1-\tilde{g}}(1-p')}, \tag{3.8}$$

which can be derived by solving Eq. (3.7) for $R(\xi)$ and substituting the result into Eq. (3.6). Finally, Eq. (3.4) immediately follows from substituting $\tilde{g} = 1/2$. Note also that if the sample climatologies of the training set and the test set are equal (i.e., $\tilde{g} = g$), Eq. (3.8) yields $p = p'$, so that no probability correction is required.

4 DATA COLLECTION

As the ML model framework presented in Section 3.1 is based on supervised learning, we require input samples, as well as labels, for training. In this chapter, we introduce the NWP model which provides the input to our ML models. Furthermore, we detail how we obtain our ground truth for training from observational data. Finally, we present the general structure of our preprocessing pipeline for producing data sets for training, validation, and testing.

4.1 DATA FROM NUMERICAL WEATHER PREDICTION

According to the aim of this thesis, our objective is to process forecasts of a convection-permitting ensemble model. There are several NWP models available which produce forecasts of the above kind in an operational fashion. Examples include AROME-EPS (France; Bouttier et al., 2012), MOGREPS-UK (United Kingdom; Porson et al., 2020), and ICON-D2-EPS (Germany; Zängl et al., 2015), all with comparable horizontal grid spacings and ensemble sizes. Since this thesis was carried out in Germany, we opt for exemplifying our ML framework on forecasts of the ICON-D2-EPS model, with the aim of strengthening collaborations with DWD, who runs the model operationally.

Icosahedral Nonhydrostatic (ICON)-D2-EPS is an limited-area model (LAM) covering the areas of Germany and parts of its neighboring countries (Fig. 4.1). The driving model is given by the global ICON ensemble model with a nesting area over Europe. The ICON models tesselate Earth's surface (modeled as a perfect sphere with radius 6371.229 km) using spherical triangles. ICON-D2-EPS is convection-permitting with an average horizontal grid spacing of 2.1 km, while the grid spacing of its driving model amounts to 20 km within the nesting area. In the vertical, ICON-D2-EPS adopts a terrain-following coordinate system with 66 *half levels*, on which vertical velocity is defined. The other state variables are computed on *full levels*, which lie halfway between the half levels (Fig. 4.2). Additionally, ICON-D2-EPS provides a number of *single-level* variables, which assign a single value to each horizontal grid point on the NWP model domain. Examples include surface pressure, mixed-layer convective available potential energy (CAPE), or relative humidity at 700 hPa. The data assimilation system KENDA (Kilometer-scale Ensemble Data Assimilation; Schraff et al., 2016) with a latent-heat nudging scheme (Stephan et al., 2008) combines current observations and a short-term forecast from the preceding data assimilation cycle to create a 20-member ensemble

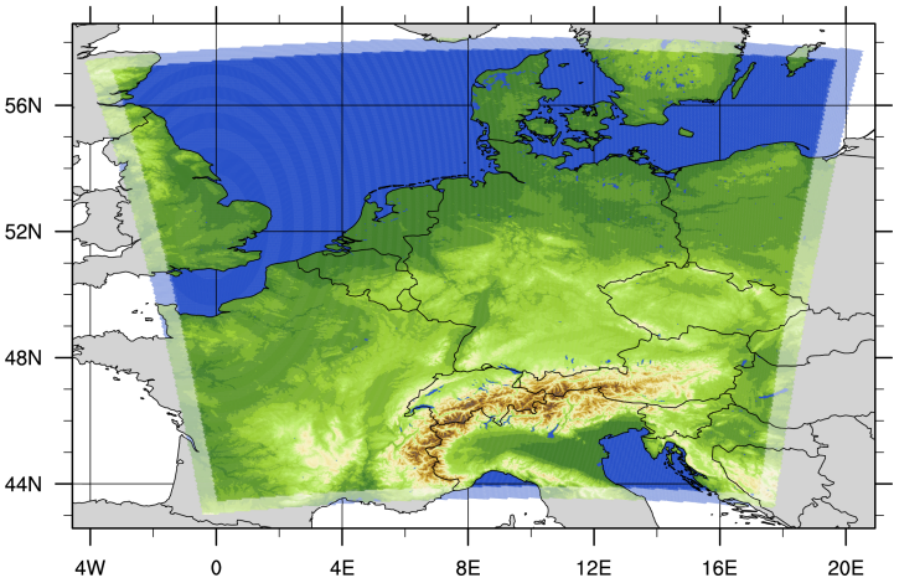

Figure 4.1: ICON-D2-EPS domain, taken from Reinert et al. (2020).

of physically consistent and statistically indistinguishable ensemble members. The operational runs are initialized eight times daily, starting at 0000 UTC (Coordinated Universal Time), and produce forecasts with a time resolution of 1 h.

Cf. Central European Summer Time (CEST): UTC = CEST − 2 h

As DWD keeps a rolling archive of ICON-D2-EPS forecasts from only the last 1.5 years, we gather our own forecast archive by continuously retrieving the latest forecasts from the DWD data base[1]. Our NWP archive comprises several summer months in

- 2021 (June, July, August)
- 2022 (May, June, July)
- 2023 (July, August).

For each full hour of the day, we collect the latest available forecast. According to the outlined NWP model initialization schedule, the collected forecasts are at most 2 h old. This procedure allows us to train our models with minimal NWP forecast uncertainty. We collect the forecasts for all ensemble members. For a limited number of days within the study period, we additionally collect all forecasts with a lead time of $3-11$ h to evaluate ML skill as a function of lead time. The precise days used for evaluating the different ML models are specified in Chapters 5.1 (SALAMA 0D), 6.1 (SALAMA 1D), and 7.1 (SALAMA 1D-EPS), respectively. Each horizontal grid point of an ensemble forecast contributes $N_e = 20$ samples $\boldsymbol{\xi}$ of atmospheric variables for training a single-member SALAMA model. Lists of the atmospheric variables used for training can be found in Section 5.1 (SALAMA 0D) and Section 6.1 (SALAMA 1D), respectively.

1 https://www.dwd.de/EN/ourservices/pamore/pamore, last access: 2025/05/20

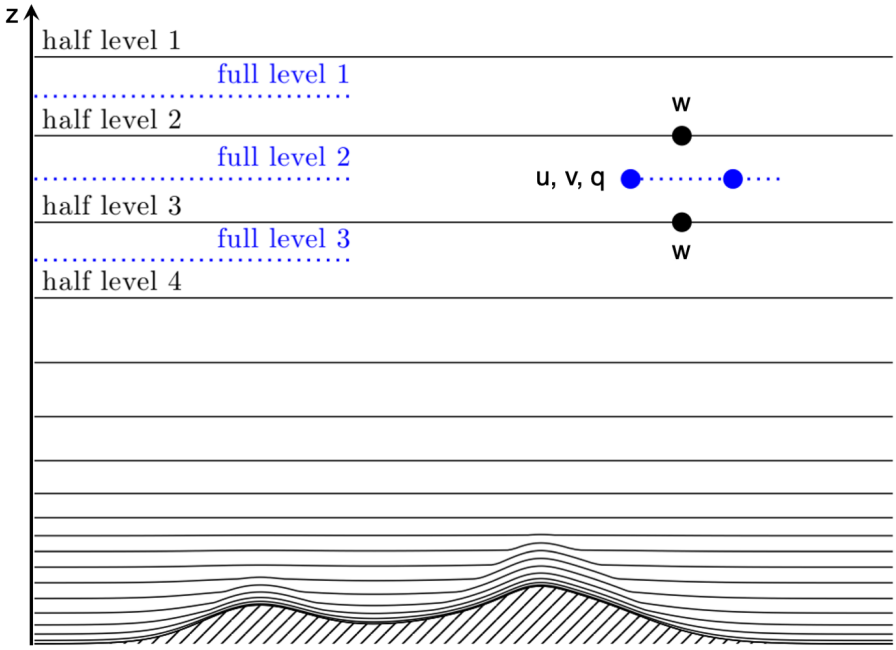

Figure 4.2: ICON-D2-EPS vertical levels and state variables, adapted from Reinert et al. (2020).

4.2 LIGHTNING OBSERVATIONS

As thunderstorms are defined by the occurrence of lightning (Section 2.1), it seems natural to consider lightning observations to establish a ground truth for past thunderstorm occurrence. Possible alternatives are given by radar data and satellite imagery. However, we decided against radar observations because of the heterogeneity of radar systems across the countries within the study region. As for satellite imagery, data quality drops when channels for visible light become unavailable, most notably after sunset. Instead, we choose ground-based lightning observations, which provide high detection efficiency and spatial accuracy across borders and throughout the day. We expect the corresponding ground truth to be associated with mature-stage convection, as lightning activity peaks at this stage of the convective life cycle (Section 2.1). Specifically, we resort to the Lightning Detection Network (LINET) (Betz et al., 2009), which exploits the radio spectrum to continuously measure strokes of lightning over Europe. The technology achieves a detection efficiency of more than 95 % and an average location accuracy of 150 m. While LINET is able to differentiate between cloud-to-ground and intracloud flashes, we have considered all lightning events as we are only interested in the yes/no occurrence of thunderstorm activity.

Given an NWP grid point at horizontal position $\mathbf{x}$ and time t, we consider a thunderstorm to occur at $(\mathbf{x}, t)$ if a flash of lightning is detected at any $(\mathbf{x}_l, t_l)$ with

$$\|\mathbf{x} - \mathbf{x}_l\| < \Delta r, \qquad |t - t_l| < \Delta t, \tag{4.1}$$

where $\|\cdot\|$ denotes the great-circle distance between $\mathbf{x}$ and $\mathbf{x}_l$. Unless stated otherwise, the spatial and temporal thresholds used in this thesis read $\Delta r = 15\,\text{km}$ and $\Delta t = 30\,\text{min}$. The temporal threshold was chosen such that the symmetric time interval around t (of length $2\Delta t$) agrees with the time resolution at which the NWP model produces operational forecasts (1 h). In turn, we set the spatial threshold to $\Delta r \approx c\Delta t$, assuming a typical thunderstorm advection speed of $c \approx 10\,\text{m}\,\text{s}^{-1}$. A similar reasoning can be found in Ukkonen et al. (2019). In Chapter 5, we will study the effect of varying the two thresholds.

4.3 COMPILATION OF DATA SETS FOR MACHINE LEARNING

The gathered archive of NWP forecasts and lightning observations can be considered a set of tuples (ξ, y), where ξ denotes the atmospheric variables for a particular grid point, target time, and ensemble member, whereas y denotes the corresponding ground truth. This constitutes precisely the required setting needed to train our single-member ML models. However, training with the entire archive ($\approx$ 100 terabytes) would not be computationally feasible. Therefore, we compile data sets for training, validation, and testing, by drawing at random a fixed number of tuples from the archive. All grid points, times and members are equally likely to be drawn, which results in climatologically consistent data sets. In the case of the training set, we arrange class balance by keeping a drawn negative example only if the current number of negative examples does not exceed half of the data set target size. In practice, we parallelize the sampling of examples with multiple workers which continuously communicate with each other about the current number of negative examples (*Message-Passing Interface (MPI)*). A summary of the different data sets is provided in Table 4.1.

To reduce correlations between the data sets, we ensure temporal separation (e.g., Ravuri et al., 2021; Geng et al., 2021; Jardines et al., 2024b). In practice, we use separate days for training, testing, and validation. In addition, we let each day begin at 0800 UTC, which we identified as the hour of least thunderstorm occurrence in our observations (Fig. 4.3). The reason for shifting the start of the day is to minimize the risk of correlations caused by thunderstorms which persist after 0000 UTC.

Table 4.1: General structure of the data sets for training, testing, and validation, used for the ML models in this thesis. We additionally compile separate test sets in which the lead time t_{lead} of the forecasts is fixed. There is one such test set for each $t_{lead} = 0\,h, 1\,h, ..., 11\,h$.

	# examples	Lead-time range	Class imbalance
Training	4×10^5	$0 - 2\,h$	1 : 1
Validation	10^5	$0 - 2\,h$	climat. consist.
Test	10^5	$0 - 2\,h$	climat. consist.
Test (t_{lead} fixed)	10^5	t_{lead}	climat. consist.

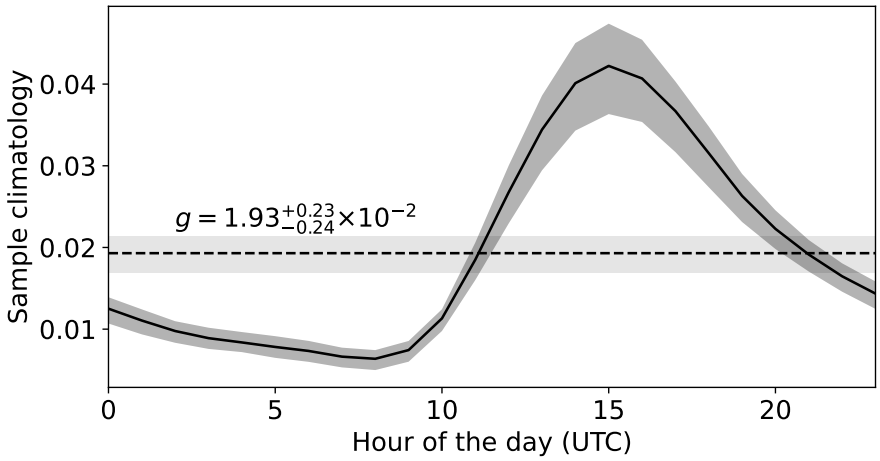

Figure 4.3: Probability of thunderstorm occurrence (sample climatology; $\Delta r = 15\,km$, $\Delta t = 30\,min$) on the ICON-D2-EPS domain from July to August as a function of the hour of the day, estimated via lightning observations between 2018 and 2022. Dashed line corresponds to the average over all hours. Shaded bands denote the symmetric 90 % confidence intervals centered around the median values. Confidence intervals for a given hour are estimated by bootstrap resampling: First, we evaluate average sample climatology for each observation month (10 months in total). Then we generate 200 *bootstrap* resamples of the 10-element set by drawing with replacement 10 elements for each resample.

Part III

RESULTS AND CONCLUSIONS

5

SALAMA 0D: INFERENCE FROM SINGLE-LEVEL PREDICTORS

In this chapter, we present results related to SALAMA 0D, the first in a series of ML models processing numerical weather prediction (NWP) forecasts. SALAMA 0D is a single-member ML model which infers thunderstorm occurrence from a number of single-level variables which are known to be associated with thunderstorms. Setting up the model and training it allows us to exemplify and test our ML framework (Chapter 3) and data preprocessing pipeline (Chapter 4) in a setting of limited compute and data intensity. We show that SALAMA 0D produces well-calibrated probabilities while outperforming a baseline ML classifier which relies on NWP reflectivity only. By varying the spatiotemporal thresholds by which we associate lightning observations with NWP data, we show that the time scale for skillful thunderstorm predictions increases linearly with the spatial scale of the forecast. The findings have been previously reported in the publication P1:

P1: Vahid Yousefnia, K., T. Bölle, I. Zöbisch, T. Gerz (2024). "A machine-learning approach to thunderstorm forecasting through post-processing of simulation data." In: *Quarterly Journal of the Royal Meteorological Society* 150.763, pp. 3495–3510. DOI: 10.1002/qj.4777.

5.1 DATA AND METHODS

In this section, we give details on the data used specifically in P1 and introduce the model architecture of SALAMA 0D. In addition, we propose a method to visualize model skill in terms of reliability and resolution, and introduce a baseline ML model for comparison.

Study region and period

For the studies in P1, we crop the the ICON-D2-EPS model domain at its borders by approximately 100 km to reduce boundary computation errors. In a cylindrical projection, our study region corresponds to a rectangle with the southwest corner located at 45° N, 1° E, the northeast corner located at 56° N, 16° E and all sides being either parallels or meridians (Fig. 5.5). At the time of this study, the NWP forecast archive introduced in Section 4.1 comprised forecasts only from 2021. Therefore, we randomly allocate the available days to training, testing, and validation, in a ratio of 4:1:1 (Fig. 5.1). Other than

that, we abide by the preprocessing pipeline presented in Section 4.3 to produce all ML data sets in Table 4.1. In particular, the days allocated for testing are equally used to generate test sets with fixed lead times. The input samples ξ are comprised by $N_f' = 21$ single-level predictors introduced next.

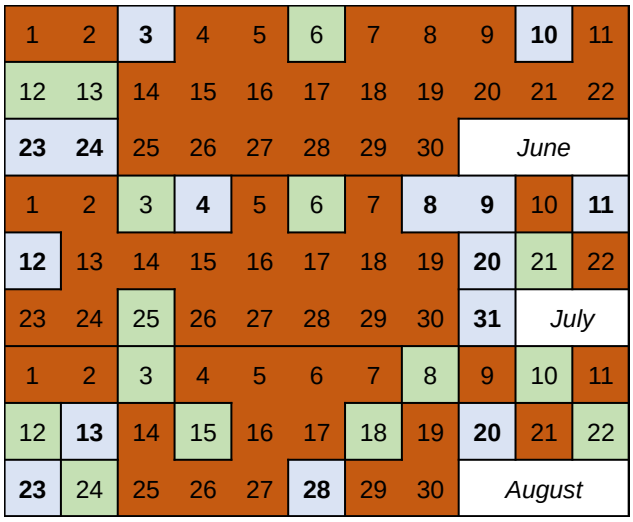

Figure 5.1: Days (from 0800 UTC to 0800 UTC on the following day) during the summer of 2021 which were used for compiling the datasets for training (dark brown), testing (light blue with bold numerals) and validation (light green). The days have been distributed at random among the three sets.

NWP predictors

The atmospheric fields used as predictors of thunderstorm occurrence in this study are given in Table 5.1. They have been selected as follows: We considered as candidate predictors all single-level fields provided in ICON-D2-EPS, as well as two pressure-level fields associated with convection in the literature, namely the relative humidity at 700 hPa and the vertical wind velocity in pressure coordinates at 500 hPa (J. Li et al., 2021). In addition, we stipulated that the predictors be available on the DWD open-data server[1], so that the trained model can eventually be used in real-time. For a given candidate input field, we compared histograms of the field value distribution during and in the absence of thunderstorm occurrence and kept only fields that differed significantly in the two distributions.

As shown in Table 5.1, all predictors can be related to physical phenomena related to convection, such as instability and moisture (Section 2.1). In particular, our selection process has led to predictors which agree with findings in the literature (Ukkonen et al., 2019; Jardines et al., 2021; Leinonen et al., 2022). Conversely, convective

1 https://opendata.dwd.de, last access: 2025/05/20

inhibition (CIN), which is sometimes used as a convective predictor (e.g., Kamangir et al., 2020), has not passed the selection process. This is likely due to the fact that we have checked for predictive power in terms of mature-stage thunderstorms. CIN, however, correlates with the hours leading up to a thunderstorm and has been removed once the storm reaches its mature stage.

Table 5.1: List of the 21 input parameters used in the study ("DIA": including sub-grid scale).

Phys. significance	ICON name	Description
Instability	CAPE_ML	Mixed-layer convective available potential energy
	CEILING	Ceiling height
	OMEGA500	Vertical wind velocity in pressure coordinates at 500 hPa
	PS	Surface pressure
	PMSL	Surface pressure reduced to mean sea level
Cloud cover	CLCH	High level clouds ($0-400$ hPa)
	CLCM	Mid-level clouds ($400-800$ hPa)
	CLCL	Low-level clouds (800 hPa to ground)
	CLCT	Total cloud cover
Precipitation and moisture	DBZ_CMAX	Maximal radar reflectivity
	ECHOTOP	Echotop pressure
	RELHUM700	Relative humidity at 700 hPa
	RELHUM_2M	2 m relative humidity
Column-integrated water quantities	TQC, TQC_DIA	Cloud water
	TQG	Graupel
	TQI, TQI_DIA	Ice
	TQV, TQV_DIA	Water vapor
	TWATER	Total water content

ML architecture and training

The architecture deployed for SALAMA 0D is presented in Fig. 5.2. Evaluating the training set, we scale the predictors to have zero mean

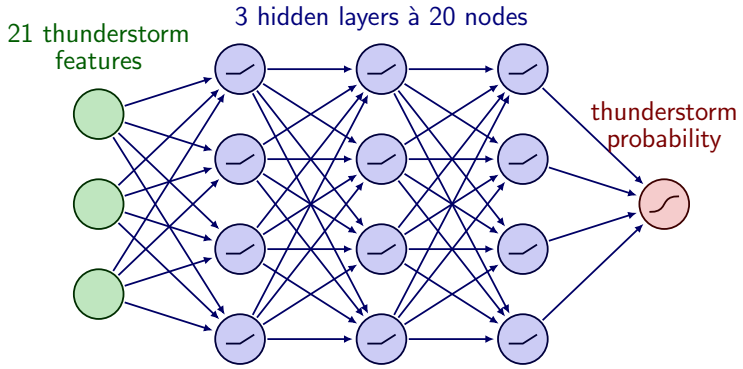

Figure 5.2: The architecture of SALAMA 0D: Input features are scaled to order 1. We use rectified linear units as activation functions in the hidden layers. A sigmoid function maps the output layer to the open interval $(0, 1)$.

and unit variance before minimizing binary cross-entropy loss via the Adam optimizer (Kingma et al., 2014). The analytic calibration formula (3.4) is applied whenever SALAMA 0D is used on the validation or test set.

The architecture presented in Fig. 5.2 actually constitutes the result of a hyperparameter study in which we varied the number of hidden layers, as well as the number of nodes per layer. We found that once a certain complexity was reached in terms of network size, adding new nodes or layers had no effect on the validation loss at the end of training. Figure 5.2 constitutes the smallest network for which this complexity threshold has been exceeded.

Bin-wise reliability and resolution

Figure 5.3a shows the reliability diagram (Section 2.3) obtained for SALAMA 0D. Shown are two calibration functions: The light gray line corresponds to a calibration function *without* any probability correction, whereas the solid black line results from applying our analytic calibration formula (3.4) to the model output. The uncalibrated line severely overestimates the relative frequency of thunderstorm occurrence at all model probabilities. As has been worked out in Section 3.3, this is *not* a result of faulty training but stems from having different sample climatologies in the training and test sets. The calibrated curve, indeed, considerably corrects for this effect, resulting in reliable forecasts for probabilities close to 0 and 1. On the other hand, our model still underestimates the relative frequency of thunderstorm occurrence

for forecast probabilities below 0.6. We shall see in Chapter 6 that model retraining after expanding our NWP forecast archive resolves this issue. For now, we consider our model sufficiently well-calibrated and appreciate that the level of reliability has been attained by means of the analytic correction (3.4) alone without the need of resorting to statistical methods like isotonic regression (Niculescu-Mizil et al., 2005).

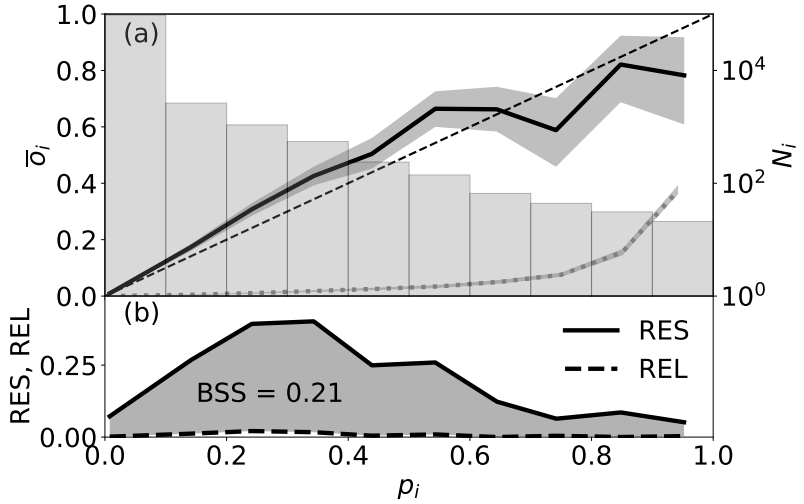

Figure 5.3: Reliability diagram of SALAMA 0D, evaluated for the test set with the label configuration $\Delta r = 15\,\text{km}, \Delta t = 30\,\text{min}$ (Section 4.2). (a) Calibration curve after applying probability correction (3.4) (black solid line), and before (grey light dotted line), and refinement distribution. Shaded band corresponds to the symmetric 90 % confidence interval obtained by 200 bootstrap resamples. (b) Bin-wise resolution and reliability (Eqs. (5.1) to (5.2)) and their relation to the Brier skill score (BSS) (Section 2.3) as a function of model probability.

The refinement distribution in Fig. 5.3a informs about the resolution of our model. It can can be difficult to compare two models solely from their refinement distributions, especially when the two models in question are similarly skillful. To assess resolution (and reliability) more easily, we introduce the following two terms which are defined for each probability bin i of width Δp:

$$\mathrm{RES}_i = \frac{1/\Delta p}{g(1-g)} \frac{N_i}{N} (p_i - g)^2, \tag{5.1}$$

$$\mathrm{REL}_i = \frac{1/\Delta p}{g(1-g)} \frac{N_i}{N} (p_i - \bar{o}_i)^2 \tag{5.2}$$

Up to a factor $g(1-g)$ known as the *uncertainty* term, the sums $\sum_i \Delta p_i \mathrm{RES}_i$ and $\sum_i \Delta p_i \mathrm{REL}_i$ are often used as quantitative definitions of resolution and reliability (Wilks, 2019, pp. 402–404). It follows from Murphy (1973) that the area enclosed by RES_i and REL_i as a func-

tion of p_i is equivalent to the Brier skill score (BSS) (with climatology as reference).

We propose to plot the bin-wise terms defined in Eqs. (5.1) to (5.2) against p_i, as shown in Fig. 5.3b. This visualization offers an overview of how much each probability bin contributes to reliability and resolution, and ultimately to the BSS. For instance, resolution is most impacted by examples with model probabilities of approximately 0.3 and dominates over reliability. This visualization will prove most useful in Chapters 6.2 and 7.2, as well.

Baseline model

To better assess the skill of SALAMA 0D, we introduce a baseline model for comparison. As thunderstorms are accompanied by heavy precipitation (Section 2.1), radar reflectivity as a measure of precipitation rate constitutes a natural surrogate for thunderstorm occurrence in the nowcasting community (Dixon et al., 1993; Wilson et al., 1998; Turner et al., 2004). ICON-D2-EPS outputs the column-maximal radar reflectivity (DBZ_CMAX in Table 5.1), to which we henceforth refer as reflectivity. In order to construct a baseline, we train a new model which uses only reflectivity as input. The architecture of the baseline model is identical to the one presented in Fig. 5.2 (three hidden layers with twenty nodes each) except for the input layer, which has now only a single node. Just like for SALAMA 0D, the baseline model outputs the probability of thunderstorm occurrence. Note that since reflectivity is one of the predictors used for SALAMA 0D, we can use the same data sets as for SALAMA 0D, ignoring predictors other than reflectivity.

Figure 5.4a shows the resulting reliability diagram. The light dotted line corresponds to the uncorrected calibration curve, while the dash-dotted line results from applying probability correction (3.4). The baseline model produces well-calibrated output for small model probabilities while the model displays underconfidence above probabilities of approximately 0.2. As examples with probabilities higher than 0.2 make up less than 1 % of the examples in the test set, we presume that these examples therefore did not contribute sufficiently to the loss function, which instead favored well-calibrated small probabilities. In an effort to construct a competitive baseline model, we use the validation set to fit a linear function to the part of the dash-dotted calibration curve with probabilities higher than 0.15. Specifically, if the output of the baseline model after application of probability correction (3.4) is denoted by p, the calibrated output reads $2.379\,p - 0.206$ for $p > 0.15$, and p otherwise. The resulting well-calibrated calibration curve is given by the solid line in the reliability diagram. The refinement distribution and the lower panel in Fig. 5.4a refer to fully calibrated probabilities. One can see that the BSS is essentially determined by the

baseline resolution, just like for SALAMA 0D (Fig. 5.3). The baseline scores comparably to SALAMA 0D in terms of reliability. On the other hand, the baseline resolution is significantly worse, which results in a lower BSS compared to SALAMA 0D.

Figure 5.4b shows the learned and calibrated relationship between NWP reflectivity and the corresponding probability of thunderstorm occurrence. The herein observed monotonously increasing relationship implies that thunderstorms become more likely as reflectivity increases. A typical reflectivity threshold for defining thunderstorms in nowcasting is 35 dBZ (Dixon et al., 1993; Mueller et al., 2003), for which the probability of thunderstorm occurrence reads 0.22.

dBZ: decibels relative to a reflectivity factor of $1\,\mathrm{mm\,m^{-3}}$*; logarithmic unit for quantifying the returned power of radars (Markowski et al., 2010, p. 369).*

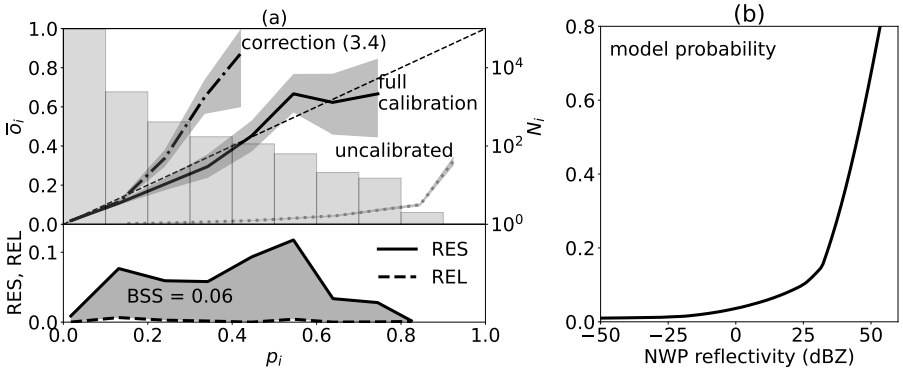

Figure 5.4: Training of the baseline model. (a) Reliability diagram panels as in Fig. 5.3, but for the baseline model. (b) Learned relationship between the baseline input field and the corresponding probability of thunderstorm occurrence.

5.2 RESULTS

In the following, we compare SALAMA 0D to the baseline model based on reflectivity, and investigate how the spatiotemporal thresholds of the lightning label configuration (Section 4.2) influence the classification skill of SALAMA 0D as a function of lead time. Unless stated otherwise, these thresholds are fixed to $\Delta r = 15\,\mathrm{km}, \Delta t = 30\,\mathrm{min}$ in this section.

Comparison to the baseline model

As a first step, we visually compare the performance of SALAMA 0D and the baseline model in a case study. For this purpose, we run SALAMA 0D for three consecutive hours of an evening with thunderstorm occurrence over Southern Germany. This day has not been used for the training of SALAMA 0D. In Fig. 5.5, we plot the probability of thunderstorm occurrence for an arbitrary member of the NWP ensemble for the entire study domain. Observed thunderstorm occurrence

is given by black contours. The corresponding plots for the baseline model are added below the panels of SALAMA 0D. In this particular case study, SALAMA 0D tends to identify more thunderstorm pixels than the baseline model. On the other hand, SALAMA 0D seems to produce more false alarms.

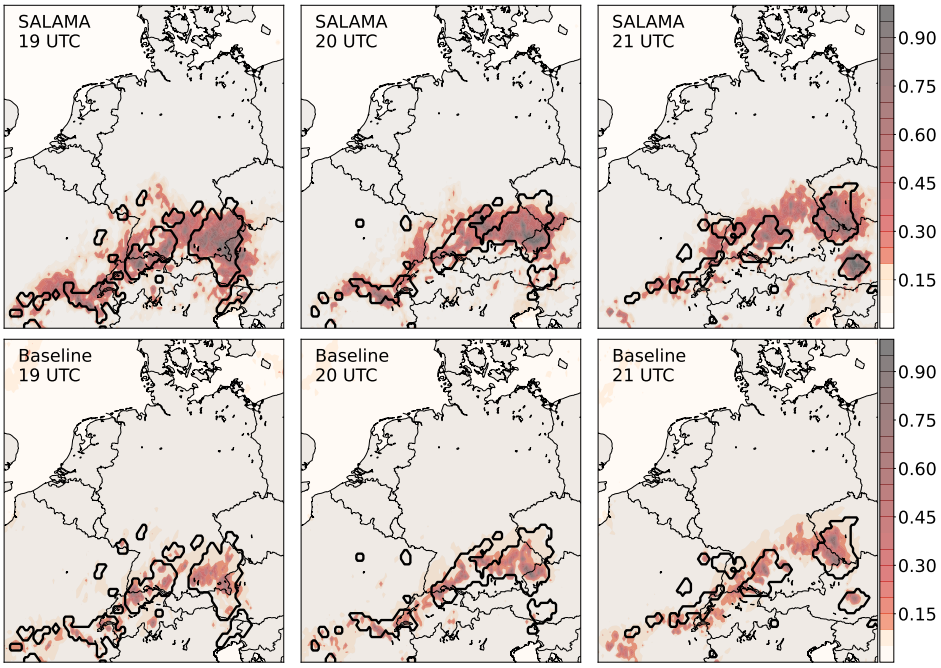

Figure 5.5: Probability of thunderstorm occurrence for June 23, 2021 from 1900 UTC on, for SALAMA 0D (upper row) and the baseline model (lower row). The model lead times for the three hours are 1 h, 2 h, and 0 h, respectively. The color maps display the result for the first ensemble member of ICON-D2-EPS, while lightning labels ($\Delta r = 15\,\text{km}, \Delta t = 30\,\text{min}$, Section 4.2) are shown as black contours. A jump in the color maps indicates the decision thresholds used for the evaluation of the skill scores in Table 5.2.

In order to compare the skill of SALAMA 0D and our baseline quantitatively for the entire study period, we evaluate the skill scores introduced in Section 2.3. We use for this purpose the test introduced in Section 5.1, which consists of examples of the entire summer of 2021. For some of the scores, we need to set a decision threshold. As a criterion, we demand that forecasts be unbiased (average fraction of examples classified as thunderstorms is equal to the observed fraction of thunderstorm examples), yielding thresholds of 0.193 (SALAMA 0D) and 0.119 (baseline). The thresholds are also indicated in the color bars of Fig. 5.5. The threshold found for reflectivity corresponds to 28 dBZ and is slightly below the typical literature threshold cited in Section 5.1.

The performance of SALAMA 0D and the baseline is summarized in Table 5.2. Irrespectively of the skill score under consideration, SALAMA 0D scores better than the baseline model. The uncertain-

ties are computed here, as well as for the subsequent evaluations, by bootstrap resampling. Note that we obtain $\mathrm{POD} = 1 - \mathrm{FAR} = \mathrm{F}_1$ for both models. This results from our choice of decision threshold (hits + misses = hits + false alarms for unbiased forecasts). We also

Table 5.2: Scores for classification skill, evaluated on the test set, for SALAMA 0D and the baseline model. The probability thresholds used for evaluation are chosen such that the forecast is unbiased and amount to 0.193 (SALAMA 0D), 0.119 (baseline). Uncertainties are obtained from 200 bootstrap resamples and show the symmetric 90 % confidence interval.

Skill score	SALAMA 0D	Baseline
PR-AUC	0.358 ± 0.018	0.141 ± 0.012
BSS	0.209 ± 0.010	0.063 ± 0.007
POD	0.403 ± 0.016	0.189 ± 0.012
$1 - \mathrm{FAR}$	0.402 ± 0.017	0.188 ± 0.013
F_1	0.403 ± 0.015	0.189 ± 0.012
CSI	0.252 ± 0.012	0.104 ± 0.007
ETS	0.241 ± 0.012	0.093 ± 0.007

show the precision-recall (PR) diagram for both models in Fig. 5.6. Both models considered in this study display higher skill than a random model following climatology would. SALAMA 0D, however, has higher classification skill than the baseline, as can be seen from the higher area under the curve (AUC) in the PR diagram in Fig. 5.6. The enhanced skill of SALAMA 0D with respect to the baseline model illustrates that a multi-parameter approach to thunderstorm forecasting is superior to employing a single input feature.

Lead time dependence of classification skill

The data sets for training, testing and validation used so far are comprised of NWP forecasts with lead times up to 2 h. We reiterate that the reason for this choice was to train and evaluate our model in a setting of minimal NWP forecast uncertainty. On the other hand, this procedure raises the question whether the thunderstorm signature learned by the model generalizes to NWP data with longer lead times (and higher forecast uncertainty). For this purpose, we involve the test sets with fixed lead times. In Fig. 5.7, we plot the SALAMA 0D classification skill, measured in terms of the skill scores introduced in Section 2.3 as a function of lead time and compare it to the dependence obtained for the baseline model. Figure 5.7 shows that, for SALAMA 0D, classification skill decreases approximately exponentially (note the log-scaling of the y-axis) for lead times longer than

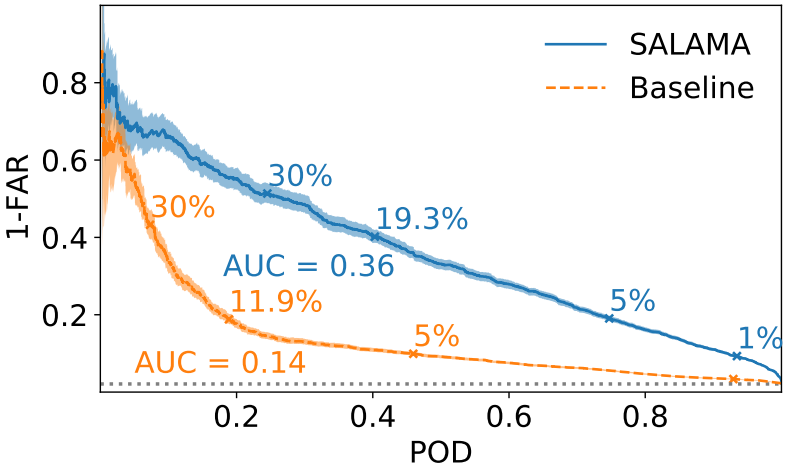

Figure 5.6: PR curve for SALAMA 0D (solid) and the baseline model (dashed), evaluated on the test set. Annotations added to the curves correspond to different decision thresholds (Section 2.3). Grey dotted line denotes models with no identification skill ($1 - \text{FAR} = 0$). Uncertainties are obtained from 200 bootstrap resamples and show the symmetric 90 % confidence interval.

1 h, irrespectively of the skill score under consideration. On the other hand, SALAMA 0D skill is systematically superior to baseline skill for all lead times. In fact, even the 11-hour-lead-time skill of SALAMA 0D is higher than the baseline skill for any of the considered lead times.

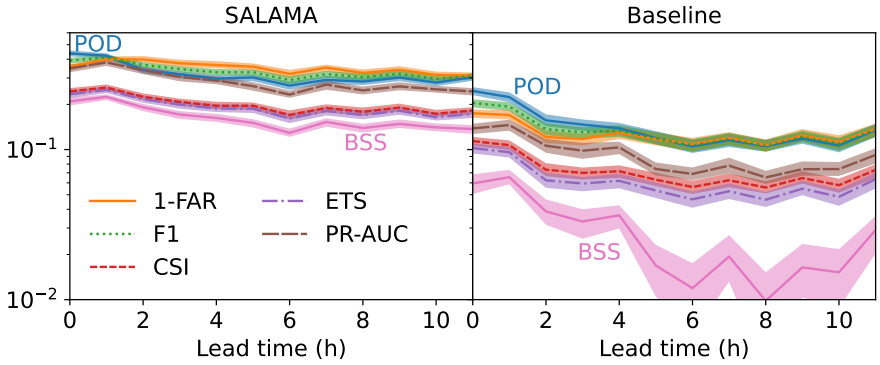

Figure 5.7: Classification skill as a function of lead time for SALAMA 0D (left) and the baseline model (right). The probability thresholds used for evaluation are chosen such that the forecast is unbiased and amount to 0.193 (SALAMA 0D), 0.119 (baseline). Uncertainties are obtained from 200 bootstrap resamples and show the symmetric 90 % confidence interval.

It is tempting to assume that the decrease in skill with lead time originates from an increasing NWP forecast uncertainty for longer lead times. We can use ensemble data to check this hypothesis. Let q be either one of the 21 input features, or the model thunderstorm probability; i.e., a quantity that is given for each ensemble member

and each lead time. Then define the ensemble spread σ'_q of q as the ensemble standard deviation of q,

$$\sigma'_q(t_{\text{lead}}) = \sqrt{\langle q(t_{\text{lead}})^2 \rangle - \langle q(t_{\text{lead}}) \rangle^2}, \tag{5.3}$$

where we make the dependence on the lead time t_{lead} explicit. The brackets $\langle \cdot \rangle$ denote the average over all 20 ensemble members. Denote by $\overline{\sigma'_q}(t_{\text{lead}})$ the expression obtained by performing the average of σ'_q over the entire study region and all times associated with the test set. Lastly, we define the normalized ensemble spread of q,

$$\sigma_q(t_{\text{lead}}) = \frac{\overline{\sigma'_q}(t_{\text{lead}})}{\overline{\sigma'_q}(0\,\text{h})}, \tag{5.4}$$

as a function of lead time. It quantifies ensemble spread in such a way that different input features can be directly compared to each other. In Fig. 5.8, the normalized ensemble spread of each of the 21 input features is shown as thin solid lines and the corresponding curve for the model output of SALAMA 0D is drawn in thick and dashed. One can see that the ensemble spread does indeed increase with lead time for most input features, the increase being approximately linear. The ensemble spread of the SALAMA 0D output increases in line with the majority of the input features and with a similar slope. This suggests that the decrease in classification skill observed in Fig. 5.6 is qualitatively consistent with the increasing variance in the simulation data.

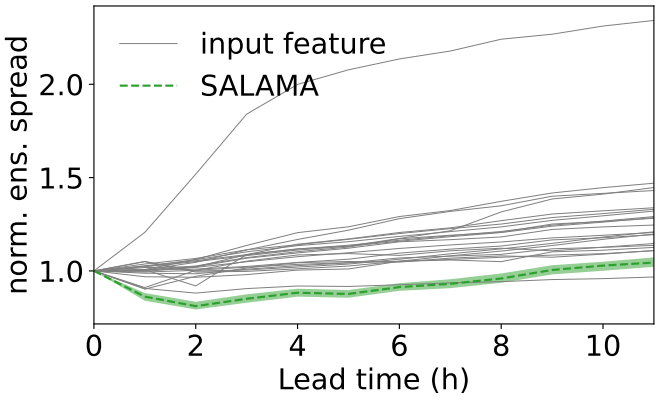

Figure 5.8: Normalized ensemble spread (as defined in Eq. (5.4)) of input features in comparison to spread of model thunderstorm probability as a function of lead time. Each thin solid line refers to one of the 21 input features. The thick dashed green line is associated with SALAMA 0D. A shaded band represents the symmetric 90 % confidence interval of uncertainty, estimated with 200 bootstrap resamples.

Effect of the label size

So far, the temporal and spatial thresholds of the label configuration have been fixed to $\Delta r = 15\,\text{km}$ and $\Delta t = 30\,\text{min}$ (henceforth referred to as "default configuration"). We now study how varying the spatiotemporal thresholds affects the classification skill of SALAMA 0D.

As a first step, we compute reliability diagrams for different label configurations. In Fig. 5.9a, we study a configuration with smaller thresholds than for the default configuration. Figure 5.9b displays a configuration with reduced Δt and increased Δr. In Fig. 5.9c, both thresholds are increased with respect to the default configuration. The exact choice of Δt and Δr for the three panels is somewhat arbitrary but still allows for qualitative insight: Irrespectively of the configuration, forecasts are well-calibrated for small and large model probabilities. In addition, model skill, quantified in terms of the BSS, increases from left to right. The diagrams show that the increase in the BSS is mainly due to enhanced contribution to resolution from probabilities larger than 0.3, though a reliability improvement from probabilities around 0.2 adds to the increase in the BSS as well.

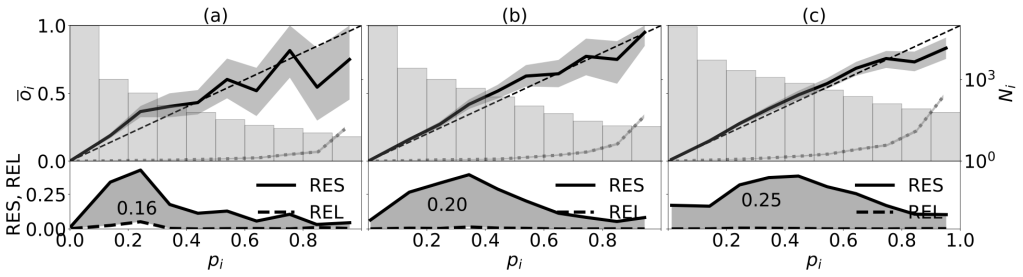

Figure 5.9: Reliability diagrams as in Fig. 5.3, but with label configurations (a) $\Delta t = 15\,\text{min}, \Delta r = 9\,\text{km}$ ($s = 14\,\text{km}$), (b) $\Delta t = 10\,\text{min}, \Delta r = 21\,\text{km}$ ($s = 24\,\text{km}$), (c) $\Delta t = 40\,\text{min}, \Delta r = 24\,\text{km}$ ($s = 36\,\text{km}$). The spatial scale s is introduced in Eq. (5.5).

As we have seen that the qualitative lead time dependence of SALAMA 0D skill does not depend on the skill score, we consider from now on only PR-AUC for further investigations. We start by computing PR-AUC for several label configurations, which is shown in Fig. 5.10. The color pattern in the figure suggests that the two thresholds are not independent variables of classification skill. Instead, one can find a parameter c with a unit of speed such that classification skill is nearly constant along lines

$$s = \Delta r + c\Delta t = \text{const.} \tag{5.5}$$

Indeed, s corresponds to a spatial resolution scale; it determines the minimal spatial accuracy that can be expected from a model trained with a given label configuration. We expect the parameter c to roughly quantify the speed at which regions of thunderstorm

occurrence are advected in the atmosphere. A fit to the data provides $c = (5.6 \pm 0.3)\,\mathrm{m\,s^{-1}}$, the order of magnitude of which is consistent with typical low- to mid-tropospheric wind speeds. We can now motivate the spatiotemporal thresholds for the reliability diagram in Fig. 5.9b: they have been chosen such that s takes on the same value as the default configuration ($s = 24\,\mathrm{km}$).

Lines of constant s appear as dashed lines in Fig. 5.10 and indicate that classification skill increases with s. This is in line with the displayed observation of an increased BSS in the reliability diagrams. This is also consistent with Roberts (2008), which investigates the spatial variation of precipitation forecast skill. Note that sample climatology g increases with s as well. In fact, it amounts to $g = 1.7 \times 10^{-3}$ in the lower left pixel of Fig. 5.10, and to $g = 4.6 \times 10^{-2}$ in the upper right corner. Since a random model with no skill has PR-AUC $= g$, the increase in skill as a function of s is to a slight extent also due to the increase in g.

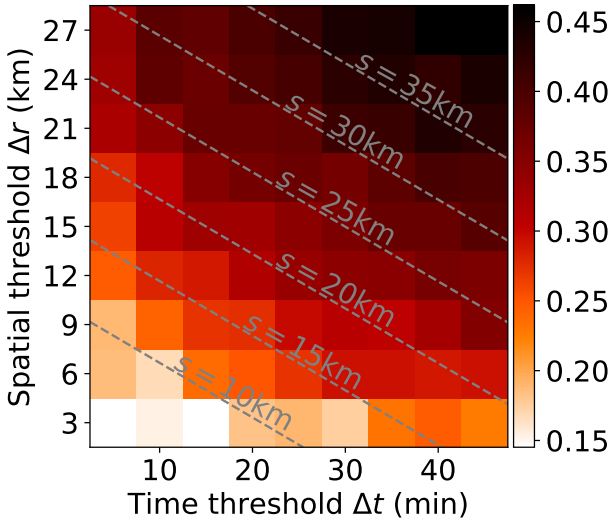

Figure 5.10: Classification skill of SALAMA 0D, expressed in terms of the area under the PR curve, as a function of the label configuration (Section 4.2). The slope of the dashed lines is chosen such that classification skill is approximately constant along the lines. Each line corresponds to a specific spatial scale s (Eq. (5.5)).

Next, we investigate how the decrease of classification skill with lead time depends on the spatial scale. Motivated by the observed decay of classification skill with lead time, we fit an exponential function $\exp(-t_{\mathrm{lead}}/\tau)$ to the lead time dependence of classification skill (measured again by PR-AUC). The skill decay time τ then provides a characteristic time scale for the decrease of classification skill. For each label configuration in Fig. 5.10, we compute the corresponding spatial scale, as well as τ. In Fig. 5.11, we present a scatter plot of τ and s. The figure shows a tight positive linear correlation between the two quantities, which means that classification skill decreases more slowly

for coarser label configurations. This is in agreement with the anticipation (Lorenz, 1969) that resolving smaller scales in NWP models is associated with a more rapid growth of forecast errors. Our finding is complementary to convection studies involving a scale-dependent skill score (Roberts, 2008), and high-resolution simulations (Selz et al., 2015).

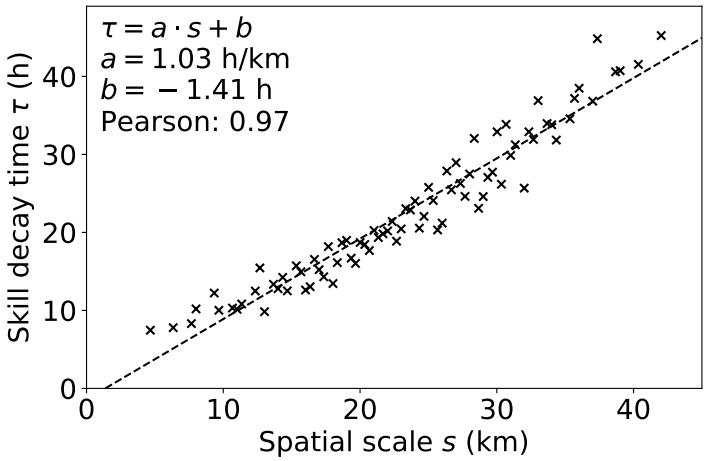

Figure 5.11: Decay time of classification skill (quantified by the area under the PR curve) as a function of the spatial scale. Each data point corresponds to one label configuration in Fig. 5.10. The parameters of a linear fit are also shown, as well as the Pearson coefficient of correlation.

5.3 CONCLUSIONS

In this chapter, we developed SALAMA 0D, a feedforward neural network model which identifies thunderstorm occurrence in NWP forecasts up to 11 h in advance in a grid-point-wise manner. The inference of the probability of thunderstorm occurrence is based on single-level predictors that are physically related to thunderstorm activity. The availability of all predictors in real-time makes SALAMA 0D suitable for operational use.

In response to research question (RQ) 1, we addressed the technical challenge caused by the rarity of thunderstorms and the corresponding small fraction of positive examples by increasing this fraction during training and analytically accounting for the increase when testing. This approach allowed us to ensure reasonable reliability without calibration fits. Furthermore, we proposed a novel visualization of reliability and resolution as a function of bin-wise model probability. The visualization could be beneficial for the comparison of similarly skillful binary classification models.

We studied how the NWP forecast uncertainty depends on the lead time of the forecast and related it to the classification skill decrease

of SALAMA 0D. As a preliminary response to RQ 5, this suggested that the decrease in ML skill is driven by an increasing NWP uncertainty. Additionally, we systematically varied the spatiotemporal criteria by which we associate lightning observations with NWP data. This allowed us to test SALAMA 0D with different spatial scales and estimate the order of magnitude of the speed at which thunderstorms are advected in the atmosphere. We showed that classification skill increases with the spatial scale of the forecast and is higher than for a baseline model based on NWP reflectivity alone. In response to RQ 5, we found that the decay time of classification skill is proportional to the spatial scale. In combination with the result that the SALAMA 0D classification skill is correlated with the NWP forecast uncertainty, our findings indicated that predicting thunderstorm occurrence at a smaller spatiotemporal resolution reduces the predictability of thunderstorm occurrence.

The development of SALAMA 0D has allowed us to build up our single-member ML framework and the corresponding data preprocessing value chain in a setting in which computational demands have been limited. In the next chapter, we replace single-level predictors by vertical profiles of the NWP state variables, building on our successfully tested ML framework.

6

SALAMA 1D: PROCESSING VERTICAL PROFILES OF STATE VARIABLES

In this chapter, we present results associated with SALAMA 1D. In contrast to SALAMA 0D, the new ML model infers the probability of thunderstorm occurrence from the vertical profiles of ten atmospheric variables, bypassing single-level predictors. The model's architecture is physically motivated: sparse connections encourage interactions at similar height levels while keeping model size and inference times computationally efficient, whereas a shuffling mechanism prevents the model from learning non-physical patterns tied to the vertical grid. We show that SALAMA 1D displays superior skill compared to SALAMA 0D. Secondly, expanding the archive of forecasts from which training examples are sampled improves skill, even when training set size remains constant. Finally, a sensitivity analysis using saliency maps indicates that our model relies on physically interpretable patterns consistent with established theoretical understanding. The findings have been previously reported in the publication P2:

P2 [under review]: Vahid Yousefnia, K., C. Metzl, T. Bölle (2025). "Inferring Thunderstorm Occurrence from Vertical Profiles of Convection-Permitting Simulations: Physical Insights from a Physical Deep Learning Model." In: *Artificial Intelligence for the Earth Systems*. Preprint available from: https://arxiv.org/abs/2409.20087.

6.1 DATA AND METHODS

In this section, we provide details on the data sets used specifically in P2. Furthermore, we present in depth the architecture developed for SALAMA 1D.

Study region and input fields

In contrast to Chapter 5, we alter the cropping of the ICON-D2-EPS domain slightly to increase the size of our study region. The resulting crop, which defines our study region for the remainder of this thesis, is shown in Fig. 6.1.

Equally, instead of processing single-level predictors, SALAMA 1D shall infer thunderstorm occurrence from the $N_f = 10$ variables given in Table 6.1. These variables correspond to the fields which are operationally available in ICON-D2-EPS on either full or half levels (Section 4.1). Note that for the remainder of this work, we denote these

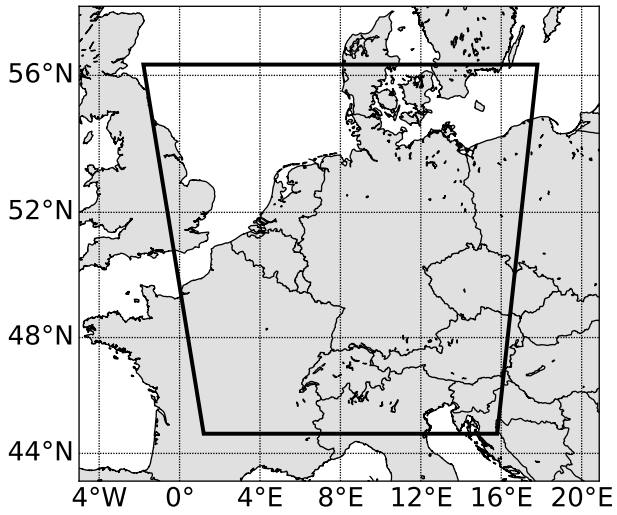

Figure 6.1: Study region for the remainder of this thesis, shown in a parallel projection. The polygon vertices are listed counterclockwise from the bottom-left: (44.7°N, 1.2°E), (44.7°N, 15.8°E), (56.3°N, 17.8°E), (56.3°N, 1.8°W). This region roughly corresponds to the numerical weather prediction (NWP) model domain (Fig. 4.1), which we cropped at the borders by approximately 80 km to reduce boundary computation errors.

variables by their respective ICON names provided in Table 6.1. We keep the fields on their native grid (vertically: $N_z = 65$ non-equidistant levels, horizontally: spherical triangles) to limit interpolation errors.

ML data sets and SALAMA model configurations

For the remainder of this thesis, we resort to the entire numerical weather prediction (NWP) archive with forecasts between 2021 and 2023 (Section 4.1). Table 6.2 provides an overview of how the archive is split into ML data sets. We use the even days of 2023 for testing and the odd days of 2023 for validation. Finally, to examine whether extending the study period from which to gather examples enhances skill, we compile two training sets for SALAMA 1D: one consists of examples from 2021 and yields a model configuration to which we refer as SALAMA 1D-2021. The other training sets comprises examples from 2021 and 2022, resulting in the model configuration SALAMA 1D-2022. Note, however, that both training sets contain an equal number of examples. Furthermore, the two models are tested (validated) on the same test (validation) set.

As a baseline for comparison, we additionally retrain SALAMA 0D to account for changes in the study region and period. In particular, we ensure that the training set for SALAMA 0D is equally made up of examples from only 2021, making it readily comparable to SALAMA 1D-2021. Since SALAMA 0D is trained on a different set

Table 6.1: (Instantaneous) vertical ICON-D2-EPS field profiles used in this study.

ICON variable	Description
U	Zonal velocity
V	Meridional velocity
T	Temperature
P	Pressure
QV	Specific humidity
QC	Cloud water mixing ratio
QI	Cloud ice mixing ratio
QG	Graupel mixing ratio
CLC	Cloud cover
W	Vertical velocity

Table 6.2: Splitting of the NWP archive for the ML data sets used in this chapter. We refer to Table 4.1 for details on the composition of these data sets (such as class imbalance).

Data set	Time period
Training	Jun-Aug 2021 (for SALAMA 0D and SALAMA 1D-2021)
	Jun-Aug 2021, May-Jul 2022 (for SALAMA 1D-2022)
Validation	Jul-Aug 2023 (odd days)
Test	Jul-Aug 2023 (even days)
Test (t_{lead} fixed)	Jul-Aug 2023 (even days)

of atmospheric variables, we cannot directly use the same test set as for the SALAMA 1D models. However, to ensure comparability, we compile the SALAMA 0D test set such that each sample corresponds to the same forecast, retrieved at the same grid point and from the same ensemble member, as its counterpart in the SALAMA 1D test set. The same procedure applies to the test sets with fixed lead time. This guarantees that any differences in model performance can be attributed to the choice of input variables.

SALAMA 1D model architecture and training

The architecture of SALAMA 1D, as illustrated in Fig. 6.2, combines dense layers with a sparse layer strategically designed to reduce the number of parameters. This approach addresses challenges such as

overfitting and the high computational demands typically associated with large ML models. Instead of using the pruning technique (LeCun et al., 1989; Frankle et al., 2019), we incorporate physical aspects and symmetry considerations to achieve a reduction in parameters. Because translational symmetry is broken along the z-direction, weight sharing, as in convolutional layers, cannot be applied effectively. Instead, we implement sparse connections, allowing interactions only between field values at similar height levels. Only further downstream do dense layers then construct dependencies between more distant field values. Additionally, we introduce a shuffling mechanism to ensure that the model does not rely on the vertical grid structure, forcing it to infer vertical orientation from the data itself. This design allows the model to, for instance, associate the formation of ice particles with the height of the tropopausal temperature inversion rather than a fixed height level, such as level 11. It turns out that shuffling also regularizes the model, limiting overfitting issues further.

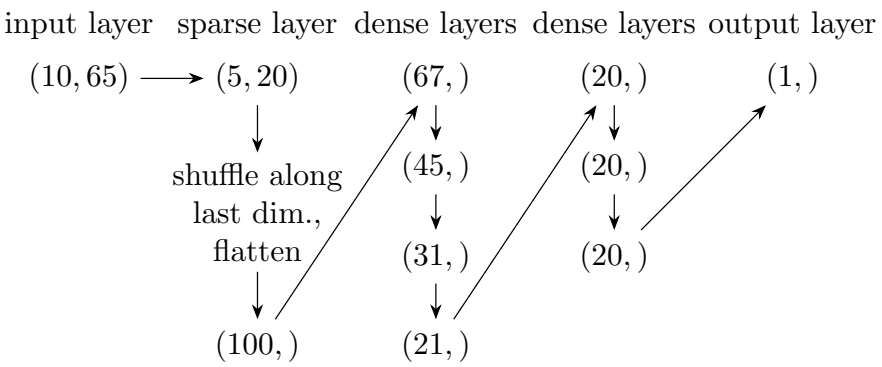

Figure 6.2: Change in input size during a forward pass in SALAMA 1D. The sparse layer reduces dimensionality and shuffles the data to prevent the model from learning dependencies tied to the vertical grid structure. Additionally, the shuffling acts as a regularization technique, helping to limit overfitting. Input fields are scaled to order 1. We use rectified linear units as activation functions after the flattened sparse layer and each dense layer, and a sigmoid function to map the output layer to the open interval $(0, 1)$. The sparse layer has 8100 trainable parameters, the other layers add 13 226 parameters. SALAMA 1D is lightweight with a computational complexity (Sovrasov, 2018) of roughly 22 kMAC (multiply-accumulate operations). SALAMA 0D (Section 5.1) requires 1.3 kMAC.

Training is then performed analogously to Section 5.1: Evaluating the training set, we scale the input fields to have zero mean and unit variance before minimizing binary cross-entropy loss via the Adam optimizer (Kingma et al., 2014). Using minibatches of size 1000, we train for 300 epochs. After training, we inspect the validation loss as a function of epoch and select the smallest epoch for which the loss no longer decreases. The analytic calibration formula (3.4) is applied

with the sample climatology value found in Section 4.3 whenever the model is used on climatologically consistent data sets.

Sparse layer details

Figure 6.3 provides an illustration of our sparse layer. The input layer is given by an array of shape (N_f, N_z) with a field dimension (iterating over the N_f field profiles) and a height dimension (iterating over the N_z vertical levels). Now, we consider a block of shape (N_f, k), where k is the size of the block along the height dimension. We densely connect the nodes within this block to h nodes in the following layer. Next, we slide the block by s nodes along the height dimension and, again, densely connect the corresponding nodes to h subsequent nodes of the following layer. Starting with a block at the bottom of the input layer, we repeat this procedure until reaching the top of the input layer. Provided that $N_z - k$ is divisible by s, this procedure leads to $N_k \equiv (N_z - k + s)/s$ blocks and produces $N_k \times h$ nodes in the following layer. We incorporate a shuffling mechanism that randomly permutes the order of the blocks for each example during training.

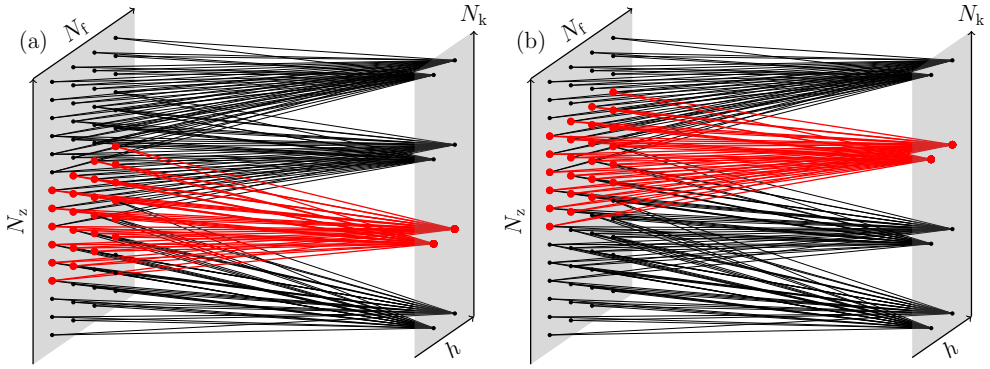

Figure 6.3: Illustration of the connections (lines between bold dots) between the input layer of shape (N_f, N_z) and the following layer of shape (h, N_k). (a) A block of nodes of shape (N_f, k) in the input layer is connected to a row of h nodes in the following layer. (b) Then, the block is shifted upwards by s nodes and rewired with the next row of h nodes of the following layer. The input layer is shown in two dimensions to help visualize each vertical field profile, but there is no spatially extended structure beyond the vertical (z) direction. Equally, we show the following layer in two dimensions to illustrate the yield of each group of connections in a separate row; however, this layer is flattened further downstream (Fig. 6.2). For better readability, the layer sizes and hyperparameter settings used in this illustration do not correspond to those in the actual model.

In contrast to a convolutional layer with a sliding kernel, all N_k blocks in our sparse layer have their own set of free parameters. In total, the sparse layer contributes $N_k \times (N_f \times k + 1) \times h$ parameters to

the model. We have studied a large variety of (k, s, h)-combinations and found that skill depends barely on the particular sliding block configuration as long as a sufficiently large number of parameters is exceeded. Setting $k = 8, s = 3, h = 5$, which corresponds to the smallest model configuration with saturating skill, we obtain $N_k = 20$ blocks, and 8100 trainable parameters. In comparison to a fully connected layer, parameter size is reduced by around 90 %.

6.2 RESULTS

We intend to investigate the following aspects concerning the model skill of SALAMA 1D. First, we compare SALAMA 1D to SALAMA 0D, studying the potential benefit of considering vertical profiles (instead of derived single-level predictors) for the prediction of thunderstorm occurrence. Moreover, we are interested in examining whether extending the study period from which to gather training examples enhances skill. Finally, we study how sensitively SALAMA 1D reacts to small input changes. The results offer insight into how the model infers thunderstorm occurrence from the input.

Model comparison

In order to get a first idea of the skill of the two SALAMA 1D models and SALAMA 0D, we consider two cases with thunderstorm activity in Central Europe, namely July 24, 2023, 1700 UTC (case A), and August 2, 2023, 1200 UTC (case B). These two cases were chosen since they display multiple simultaneous convective regions of varying size. In Fig. 6.4, we show maps of the probability of thunderstorm occurrence for Central Europe as produced by the three models and compare them with lightning observations. The probability maps have been computed by retrieving the latest NWP forecast for each target time (case A: the 2-hour forecast of the 1500 UTC model run, case B: 0-hour forecast of the 1200 UTC run) and applying the SALAMA models to one ensemble member. For both cases, we also show raw NWP output to see where the NWP model produces convection. To this end, we consider the column-maximal radar reflectivity product of ICON-D2-EPS. Specifically, for a given pixel, we compute the fraction of pixels within a radius of 15 km which exceed a threshold of 37 dBZ (e.g., Theis et al., 2005; Roberts et al., 2008). Exceedance probabilities of reflectivity with thresholds between 30 dBZ and 40 dBZ have also been used in previous studies to identify thunderstorm occurrence (e.g., Mueller et al., 2003; Leinonen et al., 2022; Ortland et al., 2023).

Case A is characterized by intense thunderstorm activity from the Alps to Northern Germany, displaying roughly ten convective objects of different sizes. Most lightning contours are predicted by all three models. However, SALAMA 0D produces a significant number of false

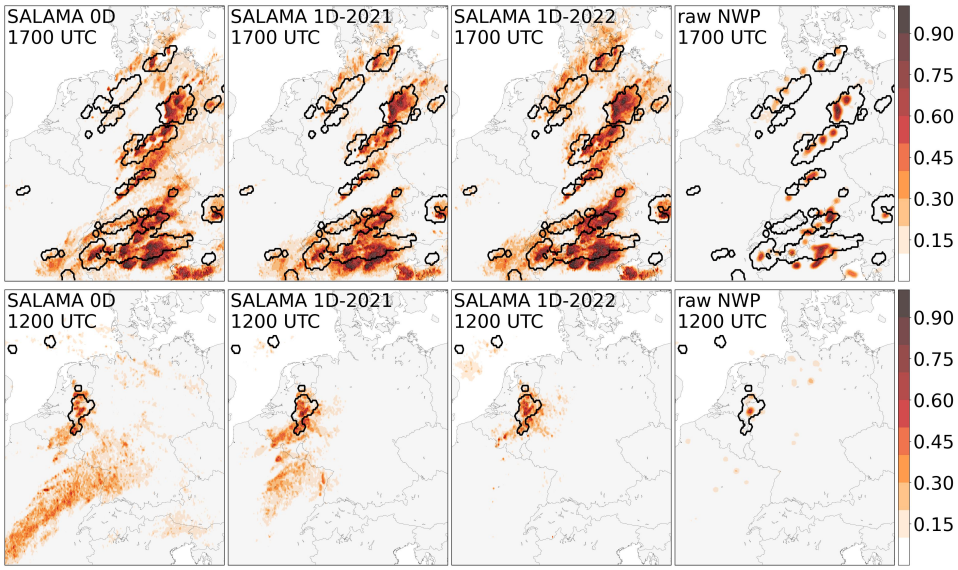

Figure 6.4: Model probability of thunderstorm occurrence for the three models of this study, evaluated for July 24, 2023, 1700 UTC (upper panels), and August 2, 2023, 1200 UTC (lower panels). The filled contours with varying shading display the result for the first ensemble member of ICON-D2-EPS, whereas lightning labels are shown as black contours. None of the dates have been used for training. NWP forecast lead times are 2 h for the upper panels and 0 h for the lower panels (note that 0-hour forecasts have limited operational utility, as they become available only after the valid time has passed). The last column shows the probability of exceeding a reflectivity threshold of 37 dBZ for the first ensemble member of ICON-D2-EPS. To obtain these exceedance probabilities, we computed for each pixel the fraction of pixels within a radius of 15 km at which the column-maximal radar reflectivity product of ICON-D2-EPS exceeds the threshold.

alarms. SALAMA 1D-2021 corrects many of them, especially in Southern Germany. SALAMA 1D-2022 tends to make its predictions more confidently than the other models, resulting in more contours which are filled out with high-probability pixels. On the other hand, the model seems to produce slightly more false alarms than SALAMA 1D-2021.

Thunderstorm activity in case B occurs primarily over the Benelux, while two smaller thunderstorms are observed over the North Sea. The latter two events are missed by the three models, though SALAMA 1D-2022 only misplaces the storms towards the South. The thunderstorm over the Benelux is captured to some extent by all the models. However, the SALAMA 1D models are more confident in their predictions, producing high-probability pixels almost everywhere within the thunderstorm contour. On the other hand, they overestimate the size of the thunderstorm, resulting in false alarms directly outside the contour. SALAMA 0D predicts a wide band of thunderstorm activity over France and Southwestern Germany, which was not confirmed by lightning observations. This region of false alarms is significantly reduced by the two SALAMA 1D models, with SALAMA 1D-2022 reducing the region to essentially zero.

The ML models, overall, align with raw NWP structures, with highest ML probability output being collocated with a high likelihood of exceeding 37 dBZ. On the other hand, the ML models tend to correct the areal size of simulated convection, with SALAMA 1D-2022 producing the least false alarms. Remarkably, the SALAMA 1D models can also produce high-probability output when lightning occurred but no convection has been triggered in the NWP model, as can be seen for the lightning regions over France for case A, which suggests that our ML models, SALAMA 1D-2022 in particular, may be able to correct for NWP model biases.

To compare the models quantitatively, we use the test set from Section 6.1. In the upper panels of Fig. 6.5, we show for each of our models the corresponding reliability diagram with $N_b = 10$ bins. All models display a similar degree of high reliability. We reiterate here that it is important to apply the analytic model calibration formula (3.4), otherwise high reliability could not be expected. The refinement distributions, as well, look similar. However, the model resolutions differ significantly: The lower panels of Fig. 6.5, which show bin-wise reliability and resolution, indicate that the increase in skill—measured by the Brier skill score (BSS)—for the two SALAMA 1D models over SALAMA 0D is primarily due to enhanced resolution. For SALAMA 1D-2021, both low- and high-probability examples enhance resolution, while for SALAMA 1D-2022, all bins with $p_i > 0.3$ contribute additional improvements to resolution.

In Table 6.3, we summarize the performance of the three models using the scores introduced in Section 2.3. These scores are posi-

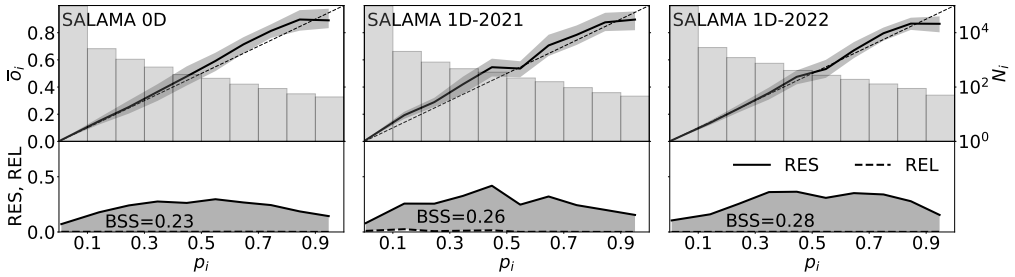

Figure 6.5: Reliability diagram for SALAMA 0D (left), SALAMA 1D-2021 (middle), and SALAMA 1D-2022 (right). Upper panels show the calibration curve and the refinement distribution, while the lower panels display bin-wise resolution and reliability (Eqs. (5.1) and (5.2)). The uncertainty on the calibration curve is obtained from 10^4 bootstrap resamples, using day-wise block resampling, and show the symmetric 90 % confidence interval. The area enclosed by bin-wise resolution and reliability corresponds to the BSS with climatology as reference. The BSS differences between the models are significant on a 90 % confidence level, as we check in Table 6.3.

tively oriented and bounded by unity. Across all skill scores, the SALAMA 1D models consistently outperform SALAMA 0D, with SALAMA 1D-2022 showing higher skill than SALAMA 1D-2021.

So far, we have worked with a test set that consists of examples from NWP forecasts with a lead time of at most 2 h. Next, we examine systematically how model skill depends on NWP forecast lead time. The lead-time dependence of skill is shown in the upper panel of Fig. 6.6. While we measure skill in terms of the BSS, we have checked that the results of this section do not qualitatively change when considering a different skill score. All three models exhibit an approximately exponential decrease in skill. The rate at which skill decreases is very similar for the three models. This suggests that the decrease of skill is not model-specific but results to a significant degree from an increasing NWP forecast uncertainty, which is consistent with previous work (Section 5.2). As a consequence, the SALAMA 1D models' superior skill for low lead times is passed on to longer lead times. In the lower panel of Fig. 6.6, we show the difference in skill as a function of lead time for all model pairs. Again, we find that the SALAMA 1D models consistently outperform SALAMA 0D at a confidence level of 90 %, with SALAMA 1D-2022 showing higher skill than SALAMA 1D-2021.

It is worth noting that the decrease in skill of SALAMA 0D is stronger than reported in Section 5.2. There, initial skill decreased by at most 30 % after 11 h, while the decrease here is approximately twice as high. This may result from using more diverse test sets in this study (we use twice as many days to compile the training set).

Table 6.3: Scores for classification skill (Section 2.3) evaluated on the test set. All scores except BSS and PR-AUC require setting a decision threshold to convert probabilities to binary output. The threshold is chosen for each model such that the average fraction of examples classified as thunderstorms is equal to the observed fraction of thunderstorm examples. For this threshold, recall is equal to precision and the F_1-score, such that only recall (POD) is reported here. Uncertainties are obtained from 10^4 bootstrap resamples, using day-wise block resampling, and show the symmetric 90 % confidence interval. The last five rows evaluate the distribution of difference in skill between the models (score(A) − score(B) for model pair (A, B)), obtained from the bootstrap resamples, and show that all differences are significant on a 90 % confidence level.

Skill score	0D	1D-21	1D-22
BSS	$0.234^{+0.037}_{-0.048}$	$0.261^{+0.035}_{-0.044}$	$0.281^{+0.034}_{-0.044}$
PR-AUC	$0.397^{+0.055}_{-0.071}$	$0.439^{+0.051}_{-0.065}$	$0.452^{+0.047}_{-0.061}$
POD	$0.414^{+0.045}_{-0.059}$	$0.452^{+0.041}_{-0.054}$	$0.465^{+0.039}_{-0.050}$
CSI	$0.261^{+0.037}_{-0.045}$	$0.292^{+0.035}_{-0.043}$	$0.303^{+0.034}_{-0.041}$
ETS	$0.250^{+0.035}_{-0.043}$	$0.281^{+0.033}_{-0.042}$	$0.293^{+0.032}_{-0.039}$
Difference	1D-21, 0D	1D-22, 1D-21	1D-22, 0D
BSS	$0.027^{+0.015}_{-0.013}$	$0.020^{+0.008}_{-0.008}$	$0.047^{+0.014}_{-0.013}$
PR-AUC	$0.043^{+0.021}_{-0.019}$	$0.012^{+0.011}_{-0.011}$	$0.055^{+0.021}_{-0.017}$
POD	$0.038^{+0.017}_{-0.016}$	$0.014^{+0.009}_{-0.009}$	$0.052^{+0.020}_{-0.018}$
CSI	$0.031^{+0.013}_{-0.013}$	$0.012^{+0.007}_{-0.007}$	$0.043^{+0.015}_{-0.014}$
ETS	$0.031^{+0.013}_{-0.013}$	$0.012^{+0.007}_{-0.007}$	$0.043^{+0.015}_{-0.014}$

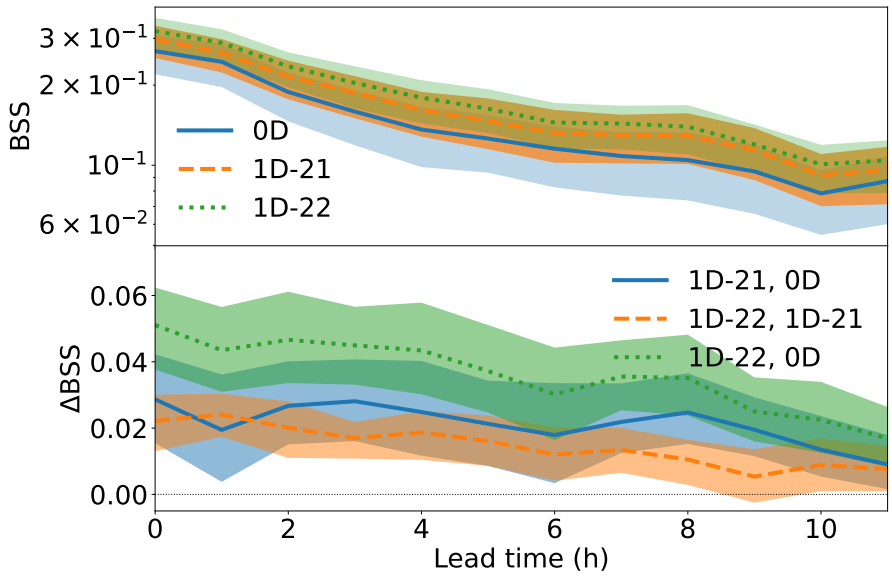

Figure 6.6: Lead-time dependence of model skill, quantified by the BSS. Upper panel shows the BSS for the individual models, while the lower panel shows the skill difference $\Delta\text{BSS} = \text{BSS(A)} - \text{BSS(B)}$ between model pair (A, B). Uncertainties are obtained from 10^4 bootstrap resamples, using day-wise block resampling, and show the symmetric 90 % confidence interval. As the shaded bands in the upper panel overlap, the lower panel is crucial for testing whether skill differences between the models are significant.

Interpretability study

For the remainder of this section, we focus on SALAMA 1D-2022, referring to it simply as SALAMA 1D. Our goal is to gain insight into how our model classifies input. We start by inspecting how the vertical profiles of the SALAMA 1D input fields look like on average for test set samples to which our model assigns a particularly high or low probability of thunderstorm occurrence. Figure 6.7 shows the average vertical profiles for the top and bottom probability percentile of test set samples. Shaded bands denote the symmetric 50 % confidence interval. Note that we converted specific humidity to dew-point temperature T_d for a more straightforward comparison with temperature. For better orientation within the panels, we also plot the average tropopause height (Section 2.1).

The first column of panels in Fig. 6.7 shows temperature and dew-point temperature T_d for the two percentiles. The top percentile troposphere displays more moisture than the bottom percentile, in particular in $2-5$ km height. This is consistent with the documented importance of moisture for thunderstorm development, as the buoyancy of rising air parcels is otherwise reduced by dry-air entrainment (F. Zhang et al., 2003; Peters et al., 2023; Marquis et al., 2023). In the second column of Fig. 6.7, we show average profiles of mixing ratios of cloud

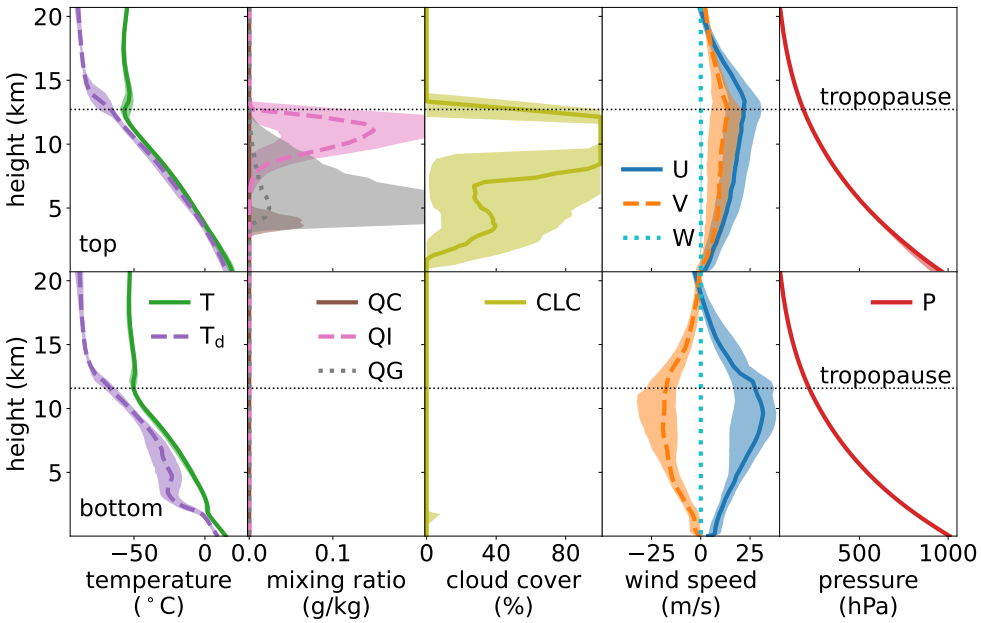

Figure 6.7: Average vertical field profiles for the top probability percentile of test set samples (upper row) and the bottom percentile (lower row), with annotated levels of the tropopause. We convert average specific humidity to dew point temperature T_d for a comparison with T. Shaded bands correspond to the symmetric 50 % confidence interval.

water (QC), cloud ice (QI), and graupel (QG). The column suggests that the model uses non-vanishing profiles of QI and QG to discriminate between the top and the bottom percentile. For high-probability samples, ice particle content peaks close to the tropopause at a height of $10 - 12$ km, consistent with measurements of vertical hydrometeor distributions (e.g., Vivekanandan et al., 1999; Hubbert et al., 2018). The third column of Fig. 6.7 displays the average profiles of cloud cover (CLC). In general, our model associates non-vanishing CLC with a high probability of thunderstorm occurrence. In particular, close to the tropopause, CLC tends to be 100 %. This is consistent with anvil cloud top levels (Markowski et al., 2010, pp. 208–209). In the fourth column of Fig. 6.7, we show vertical profiles of the three wind components U, V, and W. It is noteworthy that high-probability samples tend to have southwesterly wind profiles, whereas samples from the bottom percentile display northwesterly winds. This is consistent with studies on the typical propagation direction of thunderstorms in Central Europe (Hagen et al., 1999). On the other hand, vertical profiles of W vanish for both percentiles. We presume that due to convection displacement errors in the NWP model, the updraft regions within simulated deep convection rarely match observed lightning observations. Therefore, SALAMA 1D may have learned not to rely on W for inferring thunderstorm occurrence. The last column of Fig. 6.7 shows the average

profiles of pressure. Both profiles appear to be essentially hydrostatic. Surface pressure tends to be lower for the top percentile than for the bottom percentile.

Comparing the average profiles of the SALAMA 1D input fields for the two percentiles is useful to get a first idea whether our model separates the thunderstorm class from the majority class in a physically-interpretable manner. On the other hand, this analysis does not inform about the relative importance of the individual atmospheric variables. Therefore, we conduct a linear sensitivity analysis of the conditional probability C of thunderstorm occurrence (SALAMA 1D model output) with respect to the input. The general idea is to consider for a given input sample $\boldsymbol{\xi} = (\xi_{ij}) \in \mathbb{R}^{N_f \times N_z}$ the partial derivatives of C:

$$S_{ij}(\boldsymbol{\xi}) \equiv \frac{\partial C(\boldsymbol{\xi})}{\partial \xi_{ij}} \tag{6.1}$$

The term $S_{ij}(\boldsymbol{\xi})$ constitutes a measure of how much C reacts to changes in ξ_{ij} and, therefore, quantifies the importance of ξ_{ij} to the outcome. $S_{ij}(\boldsymbol{\xi})$ is commonly referred to as *saliency* in the ML literature (Simonyan et al., 2014; W. Li et al., 2022) while meteorologists might know it as adjoint sensitivity (Errico, 1997; Warder et al., 2021).

In order for saliency values $S_{ij}(\boldsymbol{\xi})$ to be comparable across all indices i, j, we need to scale the input fields appropriately. This scaling accounts for the fact that the fields have different units and vary differently from one sample to another. Consider an unscaled input field $\tilde{\xi}(z)$ (e.g., pressure), which we take as a function of height z above ground. We define the corresponding scaled fields as

$$\xi(z) = \frac{\tilde{\xi}(z) - \mu_\xi(z)}{\sigma_\xi(z)}, \tag{6.2}$$

where $\mu_\xi(z)$ and $\sigma_\xi(z)$ encode a characteristic background climatology for $\tilde{\xi}(z)$. We define these terms as

$$\mu_\xi(z) \equiv P_{50}\left(\tilde{\xi}(z)\right) \tag{6.3}$$

$$\sigma_\xi(z) \equiv \frac{P_{99}\left(\tilde{\xi}(z)\right) - P_1\left(\tilde{\xi}(z)\right)}{2}, \tag{6.4}$$

where P_n stands for the nth percentile of the variable in the brackets, evaluated for the training set. We then compute saliency with respect to the scaled fields.

While saliency varies from one sample to another and can be used to interpret individual predictions, we propose averaging over the top probability percentile to obtain more robust insight on the input fields and height levels which contribute most to high-probability output. To this end, we compute $|\langle S_{ij} \rangle|$, where the angle brackets denote averaging over the top percentile samples. We take the absolute value (absolute saliency), as we consider feature importance to be linked to the (sample-averaged) intensity of the ML model's linear

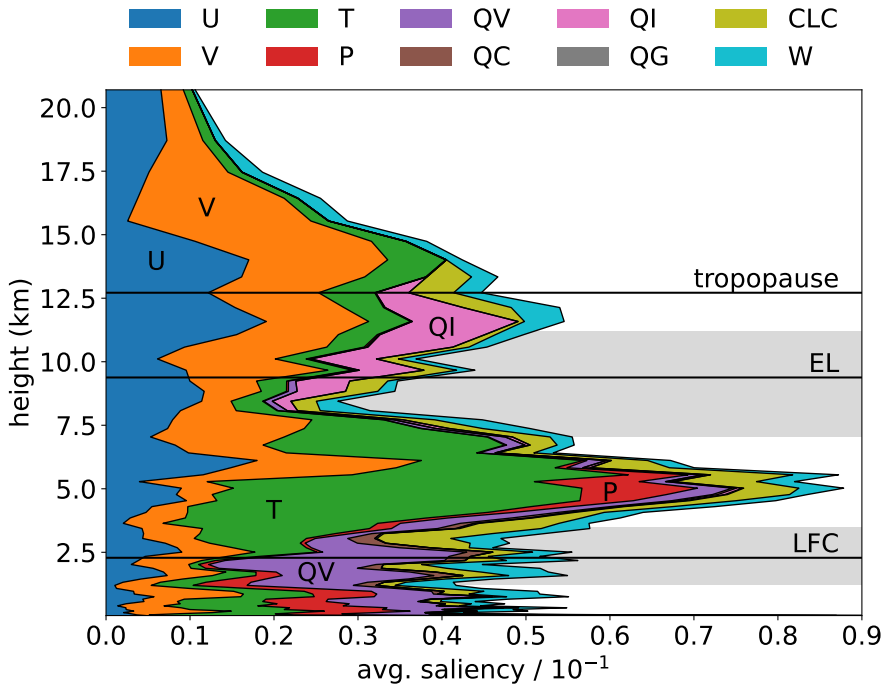

Figure 6.8: Vertical profiles of average saliency for the top percentile of test set samples (based on model probability), with annotated levels of the tropopause, as well as the level of free convection (LFC) and equilibrium level (EL) of a mixed-layer parcel. Saliencies for the different fields are stacked on top of each other in the order given in Table 6.1.

response, irrespective of whether this response is positive or negative. In what follows, we refer to $|\langle S_{ij} \rangle|$ simply as saliency, i.e., with sample-averaging and taking the absolute value being implicitly implied unless stated otherwise. The resulting saliency maps for the two percentiles are shown in Fig. 6.8. Note that average saliencies of the different fields are stacked on top of each other. As a consequence, the saliency envelope quantifies how much individual height levels affect the model outcome.

The saliency envelope displays two distinct peaks, at $z = 12\,\mathrm{km}$ and $z = 5\,\mathrm{km}$, respectively. The upper-level peak receives the most contributions by horizontal velocity. Indeed, the saliencies of U and V are maximal near the tropopause, where the average horizontal velocity difference between the top and the bottom percentile is greatest (Fig. 6.7). This suggests that the model relies to a considerable extent on the learned climatological propagation direction of thunderstorms. Conversely, W saliency is approximately one order of magnitude smaller than the saliencies of U and V, which is consistent with the average vertical profiles of W being identical for the top and bottom percentile (Fig. 6.7). QI saliency is maximal at the top of the troposphere, contributing significantly to the upper-level peak, as well. QI

is present only at this height (Fig. 6.7), which suggests that the model actively takes ice particle content into consideration to infer thunderstorm occurrence. The only other hydrometeor field with significant non-vanishing saliency is specific humidity (QV). QV saliency peaks near the level of free convection (LFC), coinciding with the vicinity of maximal tropospheric moisture (Fig. 6.7). Similarly, CLC saliency is low but non-vanishing at all heights with non-zero CLC (Fig. 6.7).

Next, we turn to the mid-level peak. Apart from horizontal velocity, the mid-level peak receives most contributions from temperature. This may be partly due to convection feeding back to the temperature field. However, since pressure saliency is non-vanishing only at the mid-level peak and near the surface, we hypothesize that the ML model also reconstructs lapse rates. To understand this, note that our model is not informed about the height of individual vertical levels; rather, these levels are randomly shuffled (Section 6.1). Thus, our model can reliably infer heights levels—and in particular level spacings—only from pressure, which monotonously decreases with height (Fig. 6.7). Therefore, we expect pressure saliency to be a proxy for how much the model relies on vertical gradients. Finally, as pressure saliency contributes to the mid-level peak and temperature saliency is high, we conjecture that our model reconstructs mid-level lapse rates. This is supported by the fact that the mid-level peak is bounded by the LFC and the equilibrium level (EL) of parcels lifted from the mixed layer, meaning that such parcels are buoyant for height levels in the vicinity of the mid-level peak. Positive buoyancy occurring in a conditionally unstable troposphere is known to be a crucial ingredient for thunderstorm development (Doswell et al., 1996) and constitutes the basis for several traditional thunderstorm predictors, such as convective available potential energy (CAPE).

To test whether SALAMA 1D considers mid-level lapse rates, we show in Fig. 6.9a the distribution of $500 - 300\,\mathrm{hPa}$ lapse rates for the top and bottom probability percentile. Indeed, essentially all high-probability samples are associated with conditionally unstable mid-level lapse rates, whereas the bottom percentile distribution extends further into absolutely stable mid-level lapse rates. On the other hand, the distributions of the two percentiles show a significant overlap, which implies that considering mid-level lapse rates alone is not sufficient to infer a high probability of thunderstorm occurrence.

Complementarily to mid-level lapse rates, we show the distributions of CAPE for the two percentiles in Fig. 6.9c. Most samples in both percentiles have a low value of CAPE, with around 40 % of the high-probability samples and 90 % of the low-probability samples falling into the lowest bin. Nevertheless, the top percentile distribution has a longer tail, towards higher CAPE values, than the bottom percentile distribution does. The low CAPE values of the bottom percentile samples are consistent with conditional instability failing to develop.

As for the top percentile samples, we expect CAPE to be considerably reduced inside the core of a convective cell, whereas some CAPE is expected to remain in lightning regions of less intense precipitation, producing the long tail of the top percentile distribution.

As pressure saliency is non-vanishing also near the surface, we show in Fig. 6.9b distributions of near-surface ($10 - 1000\,\mathrm{m}$) lapse rates. Indeed, the distributions differ for the two percentiles. In contrast to the mid-level peak, near-surface lapse rates for the top percentile tend to be lower, and mostly absolutely stable, whereas lapse rates for the bottom percentile are mostly conditionally unstable. Figure 6.9d shows the corresponding distributions of convective inhibition (CIN). While most samples of both percentiles fall into the lowest bin, the top percentile distribution has a longer tail towards higher values of CIN. This might seem surprising as this result seems to suggest that stable low-level lapse rates (or: high values of CIN) favor thunderstorm occurrence. However, the reduction of CIN is conducive to thunderstorm development only if sufficient CAPE has been able to build up beforehand. The bottom-percentile samples are essentially from environments with low CAPE and low CIN, which suggests that in these cases, the constantly low CIN (due to, e.g., continuous mixing of low-level air) caused any instabilities aloft to be released prematurely, preventing CAPE from building up, or thunderstorms from forming (Carlson et al., 1983; Tuckman et al., 2023). As far as the top-percentile samples are concerned, CIN is expected to be mostly removed within storm centers, while some CIN can be expected to remain in the less intense regions of convective precipitation or for samples with simulated convection occurring nearby or in the near future ("near misses"). This effect likely produces the longer tail of the top-percentile distribution of CIN.

In summary, SALAMA 1D appears to rely on two categories of patterns. One category consists of patterns related to

- tropopausal ice particle content, or
- cloud cover.

These patterns pertain to regions within ongoing convection, in which precipitation is most intense. In contrast, we identify a category of patterns related to

- horizontal wind direction,
- mid-level and near-surface lapse rates, or
- low-level moisture.

We refer to the latter patterns as mesoscale since the underlying fields, such as temperature and pressure, vary more slowly in the horizontal than the fine-grained hydrometeor variables do. Conversely, we refer to

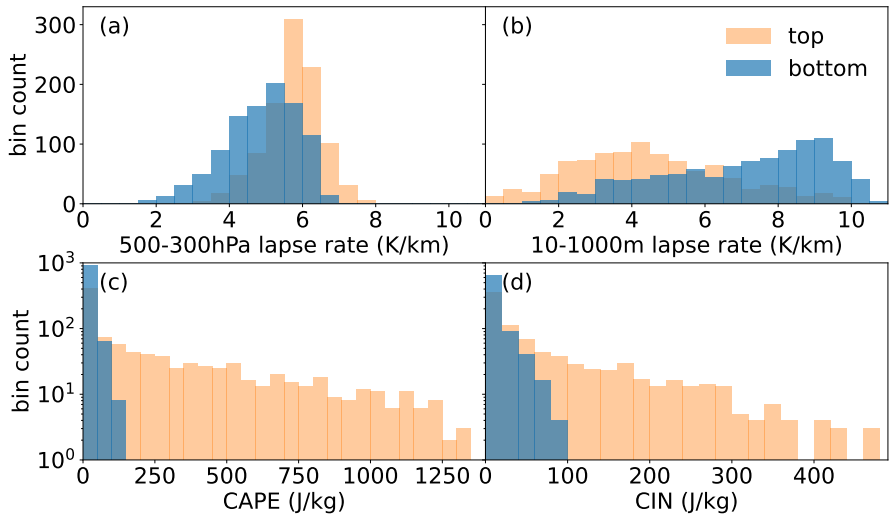

Figure 6.9: Distribution of mid-level lapse rates (a), near-surface lapse rates (b), mixed-layer CAPE (c) and mixed-layer CIN (d) for the top and bottom probability percentile of test set samples.

the former patterns as sub-mesoscale. Mesoscale patterns in the above sense tend to be characteristic of regions with sufficient distance to the convective cores such that precipitation is less intense and some CAPE remains to be released. SALAMA 1D is sensitive to both categories in order to identify thunderstorm occurrence. While sub-mesoscale patterns are useful for the identification of convective storm centers, the sensitivity to mesoscale patterns could explain the ML model's skill at increasing the areas of simulated convection observed in the case study (Fig. 6.3), namely by accounting for modestly-precipitating regions with low-level moisture and left-over CAPE.

6.3 CONCLUSIONS

Bypassing the traditional use of derived single-level predictors from NWP data, we developed SALAMA 1D, an ML model for predicting the probability of thunderstorm occurrence on a pixel-wise basis by processing vertical profiles of three-dimensional variables from convection-permitting NWP forecasts. In response to research question (RQ) 1, the design of our model's architecture was guided by physical considerations. In particular, a sparse layer reduced parameter size by encouraging interactions at similar height levels, while a shuffling mechanism prevented the model from learning patterns tied to the vertical grid structure. The latter also added a form of regularization which limited overfitting.

In comparison to SALAMA 0D, which infers thunderstorm occurrence from derived single-level features, our new model demonstrated higher skill across a wide range of metrics and for lead times up to at least 11 h. In response to RQ 2, this result indicated that informa-

tion relevant to thunderstorm occurrence, while intricately encoded in vertical profiles, can be successfully extracted by ML, resulting in an improved ability to recognize thunderstorm occurrence in NWP forecasts. Notably, our model remained lightweight in terms of computational complexity, making it just as suitable for real-time operational use as SALAMA 1D is. In response to RQ 5, case studies suggested that SALAMA 1D is capable of correcting the raw NWP output when convective areas are of incorrect size or when NWP fails to produce convection at all. Furthermore, doubling the number of days used to compile the training set (while keeping the training set size constant) also increased skill, underscoring the importance of a large and diverse database of NWP data. We anticipate further skill improvements with the collection of more NWP data.

In response to RQ 3, a sensitivity analysis based on saliency maps revealed that many learned patterns are physically interpretable. For instance, our results suggested that SALAMA 1D has learned the climatological propagation direction of thunderstorms in the study region and relies on fine-grained (sub-mesoscale) structures, such as ice particle content near the tropopause, or cloud cover, to identify high-precipitation regions of ongoing convection. Conversely, mesoscale patterns related to atmospheric instability and moisture are used, possibly to account for regions with less intense precipitation and left-over CAPE. We hypothesized that mesoscale patterns are instrumental in correcting the areal size of simulated convection.

To improve the ML model's capability of correcting for NWP-related biases, it may be beneficial to adapt the model in such a way that horizontally extended input, or several forecast times around the target time, are processed. This would allow for improvement on correcting location and timing errors of convection. Furthermore, one could train the model in such a way that it can process all ensemble members simultaneously in one forward pass, which would enable the ML model to account for NWP forecast uncertainty.

In the next chapter, we will make a step towards ensemble processing. While we do not retrain our model, we will investigate in detail the benefit of applying SALAMA 1D to all ensemble members and averaging over the generated predictions.

7

SALAMA 1D-EPS: LEVERAGING ENSEMBLE FORECASTS

This chapter presents the results associated with SALAMA 1D-EPS, our final ML model configuration, in which SALAMA 1D output is averaged across members. We first demonstrate for our ML model that ensemble-averaging significantly improves forecast skill. For one particular skill score, the Brier skill score (BSS), we derive a novel analytic expression linking skill differences to correlations between ensemble members, which aligns with observed performance gains. Additionally, we perform analyses which suggest that ML models like SALAMA 1D can identify patterns of thunderstorm occurrence which remain predictable for longer lead times compared to raw numerical weather prediction (NWP) output. The findings have been previously reported in the publication P3:

P3: Vahid Yousefnia, K., T. Bölle, C. Metzl (2025). "Increasing NWP Thunderstorm Predictability Using Ensemble Data and Machine Learning". arXiv: 2502.13316 [physics.ao-ph]. URL: https://arxiv.org/abs/2502.13316.

7.1 DATA AND METHODS

In this section, we briefly summarize the ML model configurations and data sets used throughout the current chapter.

ML model

So far, we have used SALAMA 1D only as a single-member model; i.e., given the NWP forecast of a single ensemble member, SALAMA 1D infers the corresponding probability of thunderstorm occurrence. In this chapter, we study the benefit of considering the entire forecast ensemble. To this end, we define two evaluation modes of SALAMA 1D:

- Evaluation of SALAMA 1D on single ensemble members. We use the SALAMA 1D-2022 variant throughout this chapter. We refer to this evaluation mode as "SALAMA 1D model".
- Evaluation of SALAMA 1D on all ensemble members (including the analytic calibration (3.4)) and, then, computation of the ensemble mean (Eq. (3.2)). We refer to this evaluation mode as "SALAMA 1D-EPS model".

Data preprocessing

To compare the skill of our two ML models, we again compile test sets with fixed lead times. The study region and time period for the data sets are identical to those chosen for the test sets in Section 6.1. Each data set (one for each lead time between 0 h and 11 h) consists of $N = 10^5$ examples, of which the NWP input is $\xi \in \mathbb{R}^{N_e \times N_f \times N_z} = \mathbb{R}^{20 \times 10 \times 65}$. To assure a fair comparison, both models are evaluated on the same data sets. We deal with the different model input dimensionalities as follows: For SALAMA 1D-EPS, we process each data set directly by applying Eq. (3.2) to N ensemble samples. For SALAMA 1D, we restructure each data set in such a way that each ensemble member is treated as an independent sample. Hence, we apply the single-member model to $N_e \times N = 2 \times 10^6$ samples.

7.2 RESULTS

We first report on an increase in skill of SALAMA 1D-EPS with respect to SALAMA 1D. In particular, we derive an analytic expression for the difference in skill and show that the expression is consistent with the measured difference in skill. Then, we compare the skill decay of our ML model as a function of lead time to a simple benchmark model based on raw NWP output without any ML-based corrections.

Quantitative benefit of ensemble data

To compare SALAMA 1D and SALAMA 1D-EPS quantitatively and as a function of lead time, we first produce reliability diagrams (Section 2.3). In Fig. 7.1, we show the reliability diagrams of SALAMA 1D and SALAMA 1D-EPS for the lead times 0 h, 4 h, and 8 h. We focus first on the upper halves of each panel, which show the corresponding calibration function and refinement distribution. SALAMA 1D is well calibrated for the 0-hour forecasts and even outperforms SALAMA 1D-EPS for high model probabilities. The higher reliability of SALAMA 1D is not surprising since this model was explicitly trained by loss-function optimization to output reliable *single-member* probabilities. The model was not optimized for ensemble-averaged model probabilities to be reliable. Therefore, it is remarkable that SALAMA 1D-EPS produces similarly reliable forecasts anyway. For longer lead times, both models show a similar degree of reliability.

As for resolution, it can be difficult to compare the two models solely from their refinement distributions, especially when the two models in question are similarly skillful. The lower half of each panel in Fig. 7.1 displays the bin-wise reliability and resolution contributions to the BSS (Section 5.1). Inspection of the enclosed areas reveals that even though SALAMA 1D-EPS scores worse than SALAMA 1D in terms of

reliability, its resolution is higher across all lead times, which results in a higher BSS. As lead time increases, both models' skill drops; SALAMA 1D-EPS, however, consistently outperforms SALAMA 1D in terms of the BSS. Note that the higher skill results mainly from larger contributions to resolution for modest probabilities lower than 0.5. Conversely, the contribution to skill from probabilities close to 1 are actually smaller for SALAMA 1D-EPS than they are for SALAMA 1D. This illustrates qualitatively how ensemble-averaging increases skill: The ensemble mean smooths out the rare high-probability predictions of individual members in favor of a less confident but overall more skillful averaged forecast. On a separate note, this finding also exemplifies how useful our visualization method of bin-wise reliability and resolution is for interpreting model output.

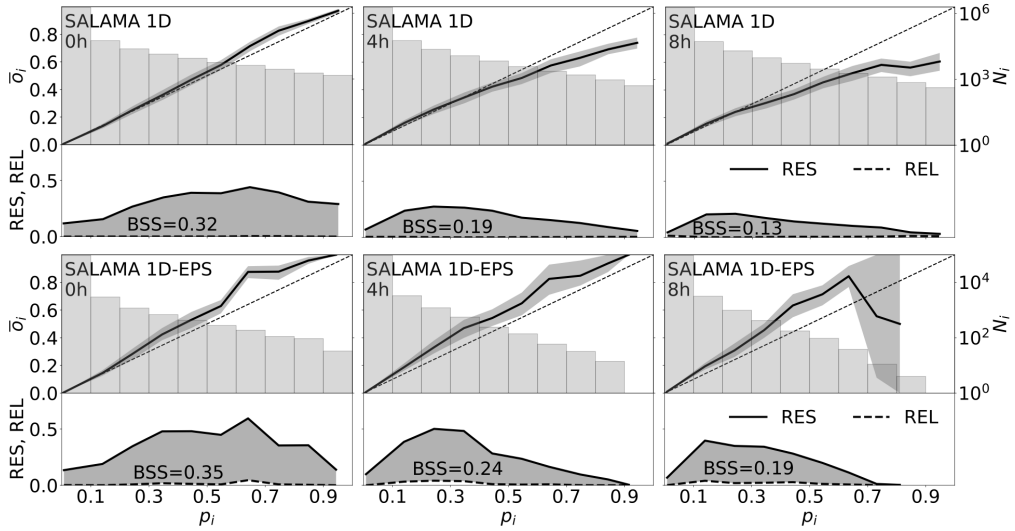

Figure 7.1: Reliability diagrams and bin-wise reliability and resolution for SALAMA 1D (upper panels) and SALAMA 1D-EPS (lower panels) for the lead times 0 h (left), 4 h (middle), 8 h (right). Shaded bands around the calibration functions denote uncertainties on a symmetric 90 % confidence interval. Uncertainties are obtained from 10^4 block bootstrap resamples, with day-wise block resampling.

For the remainder of this section, we study in more detail the lead-time dependence of the BSS for the two models, which is shown in the upper panel of Fig. 7.2. SALAMA 1D-EPS outperforms SALAMA 1D significantly. Indeed, an 11-hour forecast of SALAMA 1D-EPS is as skillful (in terms of the BSS) as the 5-hour forecast of SALAMA 1D. It is well established in the literature that ensemble systems extend the skill of deterministic forecasts (e.g., Richardson, 2000; Zhu et al., 2002; Schwartz et al., 2017). However, in our case (ensemble-averaging over binary-classifier predictions), it is possible to analytically describe the difference between deterministic and ensemble skill—at least in terms of the BSS. We now derive this expression, which, to the best

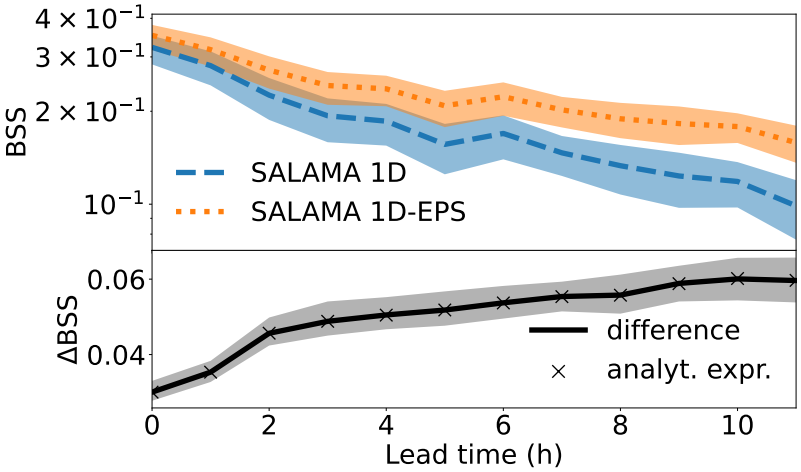

Figure 7.2: Lead-time dependence of skill, quantified by the BSS, of single-member forecasts (SALAMA 1D) and ensemble forecasts (SALAMA 1D-EPS) of thunderstorm occurrence. The lower panel shows the difference in skill, together with the prediction from the analytic expression (7.8), discussed later in the main text. Shaded bands correspond to sampling uncertainty for a symmetric 90 % confidence interval. Uncertainties are obtained from 10^4 block bootstrap resamples, with day-wise block-resampling. The lower panel reveals that the skill difference between SALAMA 1D and SALAMA 1D-EPS is significant even for the lead times at which the shaded bands in the upper panel overlap.

of our knowledge, has not been previously reported in the forecast verification literature.

Intuitively, ensemble-averaging leads to a more skillful forecast because we estimate the probability of thunderstorm occurrence based on a larger sample (of size $N_e = 20$) than in the single-member case (sample size 1). Formalizing this intuition is key to deriving the bespoke analytic expression for the ensemble-versus-deterministic BSS difference. To this end, it is instructive to introduce a probabilistic setting similar to Section 3.3, as is common practice when investigating the statistical properties of verification scores like the BSS (Bröcker et al., 2007b; Bradley et al., 2008). Specifically, we consider (continuous) SALAMA 1D output $p \in (0, 1)$ and the corresponding (discrete) thunderstorm occurrence ground truth $y \in \{0, 1\}$ to follow some unknown joint probability distribution. We denote the expected value and variance of p by $\mathbb{E}[p] \equiv \overline{p}$ and $\mathrm{Var}[p] \equiv \sigma^2$, respectively. In a probabilistic framework, the Brier score (BS) is formally defined as the following expected value (Brier, 1950; Wilks, 2019),

$$\mathrm{BS}_{\text{single-mem}} = \mathbb{E}\left[(p - y)^2\right] \tag{7.1}$$

$$= \sigma^2 + \overline{p}^2 - 2\mathbb{E}[py] + \mathbb{E}\left[y^2\right], \tag{7.2}$$

where we added the subscript "single-mem" to emphasize that this result holds for when evaluating a single ensemble member (the pre-

vious definition of the BS, Eq. (2.16), formally constitutes an estimator of Eq. (7.1) for a finite sample size). The Brier *skill* score (BSS) is still related to the BS by

$$\mathrm{BSS}_{\text{single-mem}} = 1 - \frac{\mathrm{BS}_{\text{single-mem}}}{\mathrm{BS}_{\text{ref}}}, \tag{7.3}$$

where $\mathrm{BS}_{\text{ref}} = \mathbb{E}\left[(g-y)^2\right] \equiv \kappa^2$ is a reference score using sample climatology $g \equiv \mathbb{E}[y]$.

As for SALAMA 1D-EPS, we replace p in the above framework by N_e (possibly correlated) continuous random variables $p^{(k)}$, the arithmetic mean of which yields SALAMA 1D-EPS model output. We now make the crucial assumption that the random variables $p^{(k)}$ are exchangeable; i.e., their joint distribution is invariant under any permutation of the indices $\{1, \ldots, N_e\}$. This assumption is motivated by the fact that we trained SALAMA 1D on all ensemble members without favoring any individual member. Moreover, the ensemble of perturbed initial conditions produced by the Kilometer-scale Ensemble Data Assimilation (KENDA) system in ICON-D2-EPS consists of statistically indistinguishable members (Section 4.1). Now, exchangeability implies that all $p^{(k)}$ have the distribution of p from SALAMA 1D as their marginal distribution. In addition, we have $\mathbb{E}[p^{(k)}] \equiv \overline{p}$ and $\mathbb{E}[p^{(k)}y] = \mathbb{E}[py]$ for all k, and the covariance matrix of the $p^{(k)}$ takes on a particularly simple structure,

$$\mathrm{Cov}\left[p^{(k)}, p^{(l)}\right] = \begin{cases} \sigma^2 & \text{if } k = l \\ \gamma & \text{otherwise,} \end{cases} \tag{7.4}$$

with σ^2 from the single-member case, and some number $\gamma \in \mathbb{R}$. In Fig. 7.3, we exemplify the validity of parametrization (7.4) for two test sets.

The BS for SALAMA 1D-EPS then reads:

$$\mathrm{BS}_{\text{EPS}} = \mathbb{E}\left[\left(\frac{1}{N_e}\sum_{k=1}^{N_e} p^{(k)} - y\right)^2\right] \tag{7.5}$$

$$= \frac{\sigma^2}{N_e} + \overline{p}^2 + \frac{N_e - 1}{N_e}\gamma - 2\mathbb{E}[py] + \mathbb{E}\left[y^2\right] \tag{7.6}$$

By subtracting Eq. (7.2) from Eq. (7.6), we obtain the BS difference $\Delta\mathrm{BS} = \mathrm{BS}_{\text{EPS}} - \mathrm{BS}_{\text{single-mem}}$ between the two SALAMA 1D evaluation modes,

$$\Delta\mathrm{BS} = \frac{N_e - 1}{N_e}\left(\gamma - \sigma^2\right), \tag{7.7}$$

where we note that the case of uncorrelated members ($\gamma = 0$) is known in the ensemble ML community (Abe et al., 2022). The reference score $\mathrm{BS}_{\text{ref}} = \kappa^2$ depends only on the observations; hence, it is independent

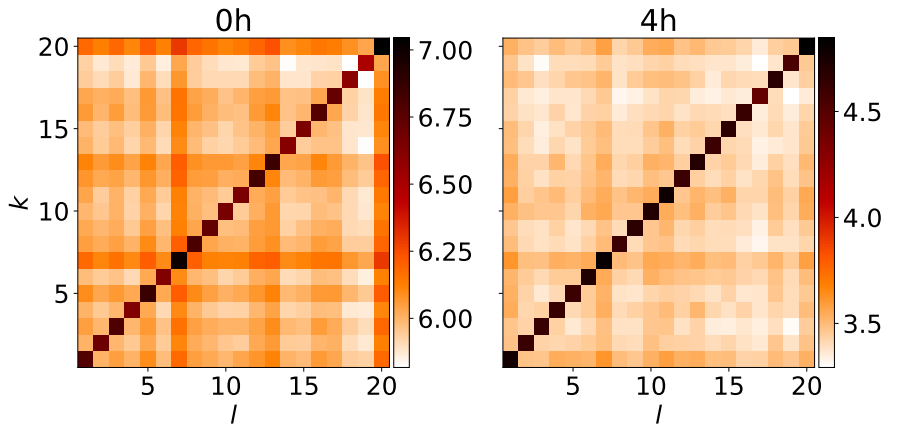

Figure 7.3: Sample covariance matrix $\mathrm{Cov}[p^{(k)}, p^{(l)}]/10^{-3}$ of the member-wise probabilities $p^{(k)}, k = 1, \ldots, N_e$, estimated for the 0-hour test set (left) and the 4-hour test set (right). If the members of the ensemble are exchangeable, the covariance matrix is fully determined by two numbers (one number for the diagonal entries of the matrix, one number for the off-diagonal entries), which is approximately the case.

of the thunderstorm identification model under consideration. Therefore, it follows from Eq. (7.3) that the difference ΔBSS between the two models is given by $-\Delta\mathrm{BS}/\kappa^2$, which yields

$$\Delta\mathrm{BSS} = \frac{N_e - 1}{N_e} \frac{\sigma^2}{\kappa^2} \left(1 - \frac{\gamma}{\sigma^2}\right). \tag{7.8}$$

Before discussing Eq. (7.8) more thoroughly, we compare the lead-time dependence of ΔBSS, as shown in the lower panel of Fig. 7.2, obtained through direct evaluation of the test sets with the values calculated using Eq. (7.8). To evaluate σ^2 and γ for a given lead time, we used the corresponding test set to estimate the ensemble covariance matrix (Fig. 7.3), and averaged over its diagonal entries (for σ^2), or off-diagonal entries (for γ). We find excellent agreement between the direct measurement of ΔBSS and (7.8). Note, however, that since the test sets are used for the direct evaluation of ΔBSS and for computing σ^2 and γ, the two data series in the lower panel of Fig. 7.2 are not independent evaluations of ΔBSS. Nevertheless, their agreement justifies the assumptions which went into deriving Eq. (7.8), in particular, the $p^{(k)}$ being exchangeable.

Having validated Eq. (7.8), we discuss some immediate conclusions. First, notice that ΔBSS decreases when γ/σ^2 approaches 1, which corresponds to ensemble members becoming increasingly correlated. Figure 7.3 shows that correlation between the members is quite high ($\gamma \approx 0.85\sigma^2$ for lead times of 0 h). This suggests that efforts to decrease inter-member correlations in the NWP ensemble (e.g., Anderson, 2016; Necker et al., 2023; Morzfeld et al., 2023) are most promising for improving thunderstorm forecasting skill in terms of the BSS. Fur-

thermore, as expected, ΔBSS increases for larger sample sizes N_e, the prefactor $(N_e - 1)/N_e$ approaching 1. However, according to Eq. (7.8), an ensemble size of $N_e = 20$ already yields a factor of $19/20 = 0.95$, which suggests that there is only little gain to be expected from increasing the size of the NWP ensemble.

Finally, we acknowledge that our analysis focused on only one skill score, namely the BSS. Note, however, that the qualitative trend seen in the upper panel of Fig. 7.2 is equally recovered when using different skill scores. In particular, we checked this for the F_1-score, critical-success index (CSI), equitable threat score (ETS), and the area under the precision-recall curve (PR-AUC). The reason why we concentrated on the BSS here is its mathematical tractability, which allows for a closed-form expression of ΔBSS. Remarkably, the result that $\Delta\text{BSS} \geqslant 0$ can also be obtained from Jensen's inequality, which states that if a function $\varphi : \mathbb{R} \to \mathbb{R}$ is convex, then

$$\varphi\left(\mathbb{E}[x]\right) \leqslant \mathbb{E}[\varphi(x)] \tag{7.9}$$

for a continuous random variable x (Jensen, 1906). In our case, we estimate expected values via averaging over N_e samples of the random variable $p - y$, using a convex function $\varphi(p) = p^2$ for BS. This immediately yields $\Delta\text{BS} < 0$. We conclude that a skill increase through ensemble averaging is guaranteed for all convex skill scores. This has been noted before (Rougier, 2016).

ML skill decay with lead time

By now, we have well understood the difference in skill between our two ML models. However, we have not commented yet on their general decrease in skill as a function of lead time (e.g., Fig. 7.2). We saw in Chapters 5.2, and 6.2, that the decrease of ML-based classification skill with lead time is considerably driven by an increase in NWP forecast uncertainty. We now investigate more precisely whether ML skill decay with lead time can be solely attributed to the increasing forecast uncertainty of the underlying NWP model.

To this end, we define a surrogate variable for thunderstorm occurrence in the raw NWP output (without any ML-based corrections) and compare it to our ML predictions. ICON-D2-EPS generates a column-maximal radar reflectivity product, in which (synthetic) radar reflectivity is computed from simulated liquid and solid water content using a one-moment parametrization scheme (Zängl et al., 2015). To obtain probabilistic reflectivity forecasts which can be directly compared to ML model output, we consider exceedance probabilities of reflectivity (e.g., Theis et al., 2005; Roberts et al., 2008): For a given grid point of an NWP forecast, we first define its neighborhood as the set of grid points within a great-circle distance $\Delta r = 15\,\text{km}$, which yields $N_n = 166$ neighbors. Just as in Fig. 6.4, our surrogate variable

for thunderstorm occurrence in raw (deterministic) NWP data at a given grid point is then defined as the fraction of neighbors exceeding a reflectivity threshold of 37 dBZ. We chose this threshold after consulting previous studies in the literature which identified thunderstorms via thresholds between 30 dBZ and 40 dBZ (Mueller et al., 2003; Leinonen et al., 2022; Ortland et al., 2023), and verifying that the following results do not change qualitatively if a different threshold within this range is chosen. The spatial threshold Δr has been chosen to match the threshold used for the spatial aggregation of the lightning observations (Section 4.2), which serve as the ground truth for evaluating raw NWP skill, as well. The thusly constructed surrogate variable, henceforth called "raw NWP", produces probability-like output between 0 and 1, making a comparison to SALAMA 1D output straightforward. For ensemble forecasts, we compute exceedance probabilities for each member, and then evaluate the ensemble mean, just like for SALAMA 1D-EPS.

When comparing the skill of the ML-based models with those based on raw NWP output, one needs to take the following aspect into account: while we expect higher model reflectivities to be more frequently associated with observed thunderstorm occurrence, the exceedance probabilities are generally not well-calibrated. In turn, a calibration-sensitive skill score (e.g., the BSS) would display low skill even if our surrogate variable were perfectly capable of discriminating between the two classes. Therefore, we measure skill using the resolution term (5.1) of the BSS, defined as $\mathrm{RES} = \sum_{i=1}^{N_b} \mathrm{RES}_i \Delta p$, effectively removing calibration sensitivity from the BSS. Any other calibration-blind score, such as PR-AUC, would be equally suitable for the following analysis.

Figure 7.4 shows the skill of the SALAMA 1D models and raw NWP as a function of lead time. Raw NWP initial skill is lower than ML initial skill, which likely originates from the ML model having access to more atmospheric variables, resulting in more precise patterns of thunderstorm occurrence. Skill decreases with lead time in the case of raw NWP output, as expected from increasing NWP forecast uncertainty. In order to compare the drop in skill quantitatively with the SALAMA 1D models, we fit exponential functions $\propto \exp(-t_{\mathrm{lead}}/\tau)$ to each curve. The fit parameter τ then provides a characteristic time scale of skill decay—and hence, predictability—that can be compared between the individual models. It is worth noting that taking into account the entire ensemble results in longer skill decay times, no matter whether one considers the ML-based models or raw NWP output. On the other hand, the SALAMA 1D models display significantly (on a 90 % confidence level) longer skill decay times than the corresponding models based on raw NWP output. This suggests that while the SALAMA 1D models' skill decay is to a significant extent driven by an increasing NWP forecast uncertainty, the ML models

do manage to slow down the decay. SALAMA 1D may have learned from observations to advantageously combine multi-variable input, resulting in more persistent patterns of thunderstorm occurrence—specifically, longer-term predictable patterns—than if no observation-based postprocessing had occurred. This finding is consistent with the established understanding that the postprocessing of NWP variables with observational data leads to improved forecasts (Vannitsem et al., 2021).

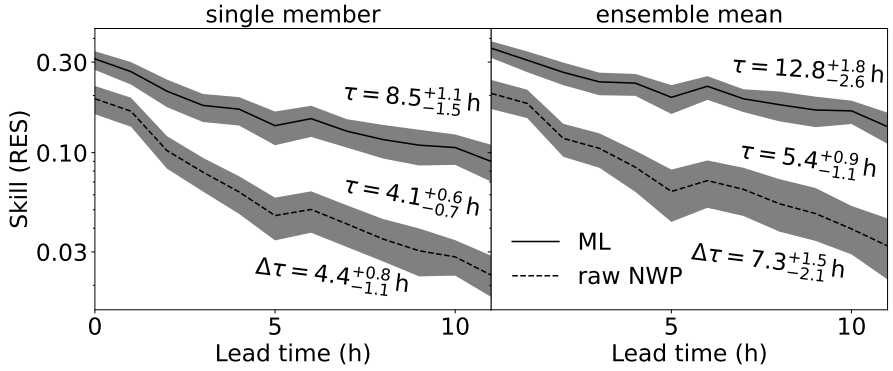

Figure 7.4: Lead-time dependence of skill, quantified by the calibration-blind skill score RES (Eq. (5.1)) for deterministic forecasts (left panel) and ensemble-averaged forecasts (right panel). Each panel displays the results for SALAMA 1D and a simple model based on raw NWP output without any ML corrections. For each line, we fit an exponential function $\propto \exp(-t_{lead}/\tau)$ to introduce a characteristic time scale τ of skill decay. Across all lines, the skill of ML-based forecasts decays more slowly than for raw NWP forecasts, as $\Delta\tau \equiv \tau(\text{ML}) - \tau(\text{raw NWP}) > 0$. Shaded bands correspond to sampling uncertainty for a symmetric 90 % confidence interval. Uncertainties are obtained from 10^4 block bootstrap resamples with day-wise block-resampling.

7.3 CONCLUSIONS

This chapter aimed to contribute to the question of how ensemble NWP models can help improving thunderstorm forecasting. Specifically, we quantitatively investigated the added benefits of ensemble-averaging, and of using an ML model trained on observations instead of using raw NWP output.

We exemplified the benefit of averaging over ensemble predictions using the SALAMA 1D model, which infers the probability of thunderstorm occurrence from vertical profiles of atmospheric variables and has been trained using forecasts of the convection-permitting NWP model ICON-D2-EPS (Chapter 6). In response to research question (RQ) 4, we found that applying SALAMA 1D to each NWP forecast member individually and then evaluating the ensemble mean (the "SALAMA 1D-EPS" evaluation mode) increases skill across lead times

up to (at least) 11 h, with an 11-hour ensemble forecast displaying the same skill as a 5-hour forecast of a single member (effectively a deterministic forecast). Importantly, we were able to quantitatively describe skill improvements resulting from ensemble-averaging by deriving and validating a novel analytic formula for the difference in skill (quantified by the BSS).

In response to RQ 5, a comparison with a simple model based on raw NWP output without any ML-based corrections revealed that skill decreases less quickly with lead time for the ML model than for the model based on raw NWP. This suggested that the decrease in ML skill with lead time is only partially a result of increasing NWP forecast uncertainty. Instead, the ML approach may allow for favorably combining input from multiple atmospheric variables by systematically taking observational data into account, which is consistent with understandings in the postprocessing community.

In closing, we stress that our findings justify applying ensemble-averaging to any binary classification model of ensemble NWP forecasts which processes each member separately. As long as the correlation between the members of the underlying ensemble NWP model is sufficiently small, we expect classification skill to improve as a result. This particularly applies to ML-based classification models whose growing role in severe weather forecasting is strengthened by our findings.

8 CONCLUSION AND PERSPECTIVES

This thesis was targeted at improving numerical weather prediction (NWP)-based thunderstorm forecasts by combining three concepts, namely deep learning, convection-permitting NWP, and ensemble systems. To this end, we developed SALAMA 1D, a deep neural network model for the identification of thunderstorm occurrence in convection-permitting ensemble forecasts. Bypassing the traditional use of single-level thunderstorm surrogates derived from the NWP state variables, our model directly processes the vertical profiles of state variables on a point-wise basis to infer the corresponding, well-calibrated, probability of thunderstorm occurrence. SALAMA 1D produces probability output for each member of an ensemble forecast individually; to consider the entire ensemble system, we proposed applying SALAMA 1D to all members and computing the ensemble mean on the generated predictions. We referred to this evaluation mode as SALAMA 1D-EPS. The single-member model SALAMA 1D was trained on operational forecasts of the ICON-D2-EPS model with lightning observations from the LINET network providing the ground truth. The code for our model is available under an open-source license,[1] as are the data sets used for training and evaluation.[2] We conclude by summarizing our findings in the context of our research questions (RQs), discussing them, and proposing further research.

8.1 SUMMARIZED ANSWERS TO THE RESEARCH QUESTIONS

The research conducted in this thesis was guided by RQs raised in Chapter 1, to which we summarize our answers below.

RQ 1: *How can an ML framework account for the rare occurrence of thunderstorms and for practical limits on training data size due to computational costs?*

The issue at hand consisted in the fact that, on the one hand, training set size was limited in practice due to otherwise unfeasibly long training times of our ML model. On the other hand, since thunderstorms are rare events, climatologically consistent data sets would need to be large to contain sufficiently many thunderstorm samples for training. Our solution to this issue was twofold. Firstly, we adopted the widely adopted approach of undersampling the majority class during training. During inference on climatologically consistent data sets, we

1 https://github.com/kvahidyou/SALAMA
2 https://doi.org/10.5281/zenodo.13981207

analytically corrected the probability shift caused by the undersampling strategy and showed that this procedure yielded well-calibrated probability output without calibration fits. The second part of our solution to the issue at hand consisted in keeping ML model complexity in check to ensure efficient training and avoid overfitting. To this end, we applied considerations based on physics and symmetries. As a first step, we invoked exchange symmetry to evaluate all individual members separately via a single model instead of processing all members simultaneously. We physically motivated the redundancy of providing location or time information as predictors. Within SALAMA 1D's architecture, we introduced a sparse layer, which encouraged interactions at similar height levels, allowing us to reduce model size further in spite of a broken translational symmetry in the vertical. In addition, a shuffling mechanism prevented the model from learning patterns tied to the vertical grid structure, effectively forcing the model to discover pressure-based coordinates, while also adding a form of regularization which limited overfitting.

RQ 2: *Can a deep neural network model, which is given the flexibility to discover—on its own—the representations needed to infer thunderstorm occurrence, outperform a conventional ML model relying on human-engineered predictors, despite constraints on training set size and high computational resource requirements?*

For all considered lead times, our deep neural network approach significantly (on a 90 % confidence level) outperformed our initial model prototype which relied on single-level thunderstorm surrogate variables. This indicated that information relevant to thunderstorm occurrence, while intricately encoded in vertical profiles, can be successfully extracted by ML, resulting in an improved ability to recognize thunderstorm occurrence in NWP forecasts. While our deep learning model constituted a major conceptual leap forward in comparison to our initial model prototype, it is still a lightweight ML model in terms of computational complexity and, hence, suitable for real-time operational use.

RQ 3: *To what extent are the patterns identified by our deep neural network model physically interpretable?*

The patterns learned by our deep neural network are to a considerable extent physically interpretable, as we established using a linear sensitivity analysis based on saliency maps. Our results suggested that our model has learned the climatological propagation direction of thunderstorms in Central Europe and relies on fine-grained structures, such as ice particle content near the tropopause, or cloud cover, to identify highly-precipitating regions of ongoing convection. Conversely, mesoscale patterns related to atmospheric instability and moisture serve to additionally account for areas with less intense precipitation

and left-over CAPE. Models with interpretable patterns consistent with physical understanding arguably foster more trust than a pure "black-box" model, which is important to make ML models like ours likely to be used in practice by severe-weather forecasters.

RQ 4: *By how much and why does skill increase when averaging over an NWP ensemble of thunderstorm forecasts?*

SALAMA 1D-EPS displayed a considerable increase in skill compared to the single-member model SALAMA 1D. As a matter of fact, an 11-hour ensemble forecast turned out to be as skillful as a 5-hour single-member forecast. We argued that skill increases because the ensemble system's larger sample size allows to estimate the probability of thunderstorm occurrence more accurately than when only a single forecast is available. For a particular skill score (the Brier skill score (BSS)), we derived an analytic expression for the difference in skill, which, to the best of our knowledge, has not been previously reported, and encourages efforts to reduce correlations between the NWP ensemble members.

RQ 5: *Which factors affect the decay of ML model skill with lead time? To what extent can ML counteract skill decays resulting from the increase of NWP uncertainty?*

The thunderstorm identification skill of SALAMA 1D-EPS decreases exponentially in the considered forecast range (Chapter 7), which we found to be the case for earlier model prototypes, as well (Chapter 6). We showed for SALAMA 0D that ML initial skill and skill decay time increase with the spatial scale of the forecast (Chapter 5). Furthermore, our ML model's initial skill, as well as skill decay time, are significantly (on a 90 % confidence level) higher than raw NWP skill, which results in an increased practical predictability of thunderstorm occurrence (Chapter 7). Rather than mirroring the increasing NWP forecast uncertainty, this result suggested that our ML model can partially correct for biases in the NWP model. This was consistent with a case study in Chapter 6, in which our ML model corrected the areal size of convection and correctly produced non-vanishing probability output even the NWP model failed to produce convection. We hypothesized that the reliance of our ML model on mesoscale patterns indicative of less intense precipitation outside convective cores is instrumental in correcting the areal size of simulated convection.

8.2 DISCUSSION AND OUTLOOK

In this section, we discuss limitations in our methodology and propose research avenues to address them. Furthermore, we comment on the general significance of this work.

Some conceptual questions went unaddressed in this thesis. For instance, while we investigated the added benefit of ensemble data compared to deterministic forecasts, we did not study the benefit of convection-permitting forecasts. Retraining on NWP data with a coarser horizontal grid resolution can help quantifying how much of the observed skill of our SALAMA models can be attributed to the grid spacing of ICON-D2-EPS. Possible NWP models for this task include the deterministic ICON model over Europe with a resolution of 6.5 km. Such a retraining additionally allows us to study whether the learned patterns of SALAMA 1D generalize to different model domains while addressing the need of the aviation sector for higher spatial coverage than provided by ICON-D2-EPS.

Further limitations are given by simplifying assumptions which went into constructing our ML framework, namely the processing of grid-pointwise input, and the merging of ensemble forecasts by computing the ensemble mean of ML output. These model choices were justified to constrain model complexity and arguably were instrumental in keeping a degree of interpretability in the model, as, e.g., spatially extended input would have complicated a graphical representation of saliency in terms of input fields and height. Nonetheless, future research should investigate the use of NWP state variables at several horizontal locations around the grid point of interest. This could help a deep neural network to better account for convection displacement errors of NWP. Similarly, an ML model trained to process all ensemble members could harvest valuable data from the spread of an ensemble, instead of considering only the ensemble mean. On the other hand, individual training set samples will become large in size when ensemble members and neighboring grid points are considered, which will in practice limit the number of available training samples—even more so than in this thesis. Methods from transfer learning could be used to limit the number of adjustable parameters. For instance, one could first train an autoencoder to learn a low-dimensional embedding of the $(10, 65)$-dimensional SALAMA 1D input. In a second step, a deep neural network could be trained to process the vertical-profile embeddings from N_n neighboring grid points and N_e members simultaneously.

We close by emphasizing that while the primary focus of this thesis was to develop a deep learning model for identifying thunderstorm occurrence in NWP data, many concepts and methods derived here actually apply to a large class of use-cases involving binary classification: Our novel visualization method of reliability and resolution as a function of model output (Chapter 5) provides a useful extension to reliability diagrams, arguably making refinement distributions easier to interpret. Furthermore, while the analytic calibration formula (3.4) used for correcting probability shifts due to undersampling is known in the broader ML literature, its application to binary classification in

severe weather forecasting appears to be novel, allowing for training under optimal use of computational resources without the need for calibration fits afterwards. Finally, while deep learning on increasing amounts of data plays an ever more significant role in research and in our daily life, this thesis demonstrates how the incorporation of physical considerations and symmetry principles can help reducing model complexity and addressing issues like overfitting. In this sense, theoretical physics remains instrumental in our data-driven new era—just as thunderstorms will continue to inspire awe across humankind.

BIBLIOGRAPHY

Abe, Taiga, Estefany Kelly Buchanan, Geoff Pleiss, Richard Zemel, and John P Cunningham (2022). "Deep Ensembles Work, But Are They Necessary?" In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., pp. 33646–33660. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/da18c47118a2d09926346f33bebde9f4-Paper-Conference.pdf.

Anderson, Jeffrey L. (2016). "Reducing Correlation Sampling Error in Ensemble Kalman Filter Data Assimilation." In: *Monthly Weather Review* 144.3, pp. 913–925. DOI: 10.1175/MWR-D-15-0052.1.

Bauer, Peter, Alan Thorpe, and Gilbert Brunet (2015). "The quiet revolution of numerical weather prediction." In: *Nature* 525.7567, pp. 47–55. DOI: 10.1038/nature14956.

Bechtold, Peter et al. (2014). "Representing Equilibrium and Nonequilibrium Convection in Large-Scale Models." In: *Journal of the Atmospheric Sciences* 71.2, pp. 734–753. DOI: 10.1175/JAS-D-13-0163.1.

Betz, Hans D. et al. (2009). "LINET—An international lightning detection network in Europe." In: *Atmospheric Research* 91.2. 13th International Conference on Atmospheric Electricity, pp. 564–573. ISSN: 0169-8095. DOI: 10.1016/j.atmosres.2008.06.012.

Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. New York: Springer. ISBN: 978-0-387-31073-2.

Borsky, Stefan and Christian Unterberger (2019). "Bad weather and flight delays: The impact of sudden and slow onset weather events." In: *Economics of Transportation* 18, pp. 10–26. ISSN: 2212-0122. DOI: 10.1016/j.ecotra.2019.02.002.

Bouttier, François, Benoît Vié, Olivier Nuissier, and Laure Raynaud (2012). "Impact of Stochastic Physics in a Convection-Permitting Ensemble." In: *Monthly Weather Review* 140.11, pp. 3706–3721. DOI: 10.1175/MWR-D-12-00031.1.

Bradley, A. Allen, Stuart S. Schwartz, and Tempei Hashino (2008). "Sampling Uncertainty and Confidence Intervals for the Brier Score and Brier Skill Score." In: *Weather and Forecasting* 23.5, pp. 992–1006. DOI: 10.1175/2007WAF2007049.1.

Brecht, Martin (1983). *Martin Luther. Sein Weg zur Reformation: 1483-1521*. 2nd ed. Calwer Verlag. ISBN: 3-7668-0678-5.

Brier, Gleinn W. (1950). "Verification of forecasts expressed in terms of probability." In: *Monthly Weather Review* 78.1, pp. 1–3. DOI: 10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2.

Bröcker, Jochen and Leonard A. Smith (2007a). "Increasing the Reliability of Reliability Diagrams." In: *Weather and Forecasting* 22.3, pp. 651–661. DOI: 10.1175/WAF993.1.

– (2007b). "Scoring Probabilistic Forecasts: The Importance of Being Proper." In: *Weather and Forecasting* 22.2, pp. 382–388. DOI: 10.1175/WAF966.1.

Burke, Amanda, Nathan Snook, David John Gagne II, Sarah McCorkle, and Amy McGovern (2020). "Calibration of Machine Learning–Based Probabilistic Hail Predictions for Operational Forecasting." In: *Weather and Forecasting* 35.1, pp. 149–168. DOI: 10.1175/WAF-D-19-0105.1.

Carlson, T. N., S. G. Benjamin, G. S. Forbes, and Y-F. Li (1983). "Elevated Mixed Layers in the Regional Severe Storm Environment: Conceptual Model and Case Studies." In: *Monthly Weather Review* 111.7, pp. 1453–1474. DOI: 10.1175/1520-0493(1983)111<1453:EMLITR>2.0.CO;2.

Clark, Adam J., William A. Gallus, Ming Xue, and Fanyou Kong (2009). "A Comparison of Precipitation Forecast Skill between Small Convection-Allowing and Large Convection-Parameterizing Ensembles." In: *Weather and Forecasting* 24.4, pp. 1121–1140. DOI: 10.1175/2009WAF2222222.1.

Clark, Peter, Nigel M. Roberts, Humphrey Lean, Susan P. Ballard, and Cristina Charlton-Perez (2016). "Convection-permitting models: a step-change in rainfall forecasting." In: *Meteorological Applications* 23.2, pp. 165–181. DOI: 10.1002/met.1538.

Craig, George C. et al. (2021). "Waves to Weather: Exploring the Limits of Predictability of Weather." In: *Bulletin of the American Meteorological Society* 102.11, E2151–E2164. DOI: 10.1175/BAMS-D-20-0035.1.

Diffenbaugh, Noah S., Martin Scherer, and Robert J. Trapp (2013). "Robust increases in severe thunderstorm environments in response to greenhouse forcing." In: *Proceedings of the National Academy of Sciences* 110.41, pp. 16361–16366. DOI: 10.1073/pnas.1307758110.

Dixon, Michael and Gerry Wiener (1993). "TITAN: Thunderstorm Identification, Tracking, Analysis, and Nowcasting—A Radar-based Methodology." In: *Journal of Atmospheric and Oceanic Technology* 10.6, pp. 785–797. DOI: 10.1175/1520-0426(1993)010<0785:TTITAA>2.0.CO;2.

Done, James, Christopher A. Davis, and Morris Weisman (2004). "The next generation of NWP: explicit forecasts of convection using the weather research and forecasting (WRF) model." In: *Atmospheric Science Letters* 5.6, pp. 110–117. DOI: 10.1002/asl.72.

Doswell, Charles A. (1984). *The operational meteorology of convective weather, Volume II: Storm scale analysis.* National Oceanic and Atmospheric Administration.

Doswell, Charles A., Harold E. Brooks, and Robert A. Maddox (1996). "Flash Flood Forecasting: An Ingredients-Based Methodology." In:

Weather and Forecasting 11.4, pp. 560–581. DOI: 10.1175/1520-0434(1996)011<0560:FFFAIB>2.0.CO;2.

Dramsch, Jesper Sören et al. (2025). "Explainability can foster trust in artificial intelligence in geoscience." In: *Nature Geoscience*, pp. 1–3. DOI: 10.1038/s41561-025-01639-x.

Duchi, John, Elad Hazan, and Yoram Singer (2011). "Adaptive subgradient methods for online learning and stochastic optimization." In: *Journal of machine learning research* 12.7.

Dwyer, Joseph R. and Martin A. Uman (2014). "The physics of lightning." In: *Physics Reports* 534.4. The Physics of Lightning, pp. 147–241. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2013.09.004.

Elkan, Charles (2001). "The foundations of cost-sensitive learning." In: *International joint conference on artificial intelligence*. Vol. 17. 1. Lawrence Erlbaum Associates Ltd, pp. 973–978.

Emanuel, K.A. (1994). *Atmospheric Convection*. Oxford University Press. ISBN: 978-0-19-506630-2.

Errico, Ronald M. (1997). "What Is an Adjoint Model?" In: *Bulletin of the American Meteorological Society* 78.11, pp. 2577–2592. DOI: 10.1175/1520-0477(1997)078<2577:WIAAM>2.0.CO;2.

Flora, Montgomery L., Corey K. Potvin, Amy McGovern, and Shawn Handler (2024). "A Machine Learning Explainability Tutorial for Atmospheric Sciences." In: *Artificial Intelligence for the Earth Systems* 3.1, e230018. DOI: 10.1175/AIES-D-23-0018.1.

Frankle, Jonathan and Michael Carbin (2019). "The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks." In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=rJl-b3RcF7.

Gagne, David John et al. (2017). "Storm-Based Probabilistic Hail Forecasting with Machine Learning Applied to Convection-Allowing Ensembles." In: *Weather and Forecasting* 32.5, pp. 1819–1840. DOI: 10.1175/WAF-D-17-0010.1.

Gelman, Andrew et al. (2013). *Bayesian Data Analysis*. 3rd. Boca Raton, FL: CRC Press. ISBN: 978-1-4398-4095-5.

Geng, Yangli-ao et al. (2019). "LightNet: A Dual Spatiotemporal Encoder Network Model for Lightning Prediction." In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '19. Anchorage, AK, USA: Association for Computing Machinery, pp. 2439–2447. ISBN: 978-1-4503-6201-6. DOI: 10.1145/3292500.3330717.

Geng, Yangli-ao et al. (2021). "A deep learning framework for lightning forecasting with multi-source spatiotemporal data." In: *Quarterly Journal of the Royal Meteorological Society* 147.741, pp. 4048–4062. DOI: 10.1002/qj.4167.

Gerz, Thomas, Caroline Forster, and Arnold Tafferner (2012). "Mitigating the Impact of Adverse Weather on Aviation." In: *Atmospheric Physics: Background – Methods – Trends*. Berlin, Heidelberg: Springer

Berlin Heidelberg, pp. 645–659. ISBN: 978-3-642-30183-4. DOI: 10.1007/978-3-642-30183-4_39.

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio (Apr. 2011). "Deep Sparse Rectifier Neural Networks." In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Geoffrey Gordon, David Dunson, and Miroslav Dudík. Vol. 15. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, pp. 315–323. URL: https://proceedings.mlr.press/v15/glorot11a.html.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press.

Hagen, Martin and Ullrich Finke (1999). "Motion characteristics of thunderstorms in southern Germany." In: *Meteorological Applications* 6.3, pp. 227–239. DOI: 10.1017/S1350482799001164.

Hasanin, Tawfiq and Taghi Khoshgoftaar (2018). "The Effects of Random Undersampling with Simulated Class Imbalance for Big Data." In: *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, pp. 70–79. DOI: 10.1109/IRI.2018.00018.

He, Jiaying and Tatiana V Loboda (Nov. 2020). "Modeling cloud-to-ground lightning probability in Alaskan tundra through the integration of Weather Research and Forecast (WRF) model and machine learning method." In: *Environmental Research Letters* 15.11, p. 115009. DOI: 10.1088/1748-9326/abbc3b.

Herman, Gregory R. and Russ S. Schumacher (2018). "Money Doesn't Grow on Trees, but Forecasts Do: Forecasting Extreme Precipitation with Random Forests." In: *Monthly Weather Review* 146.5, pp. 1571–1600. DOI: 10.1175/MWR-D-17-0250.1.

Holle, Ronald L. (2014). "Some aspects of global lightning impacts." In: *2014 International Conference on Lightning Protection (ICLP)*, pp. 1390–1395. DOI: 10.1109/ICLP.2014.6973348.

– (2016). "A Summary of Recent National-Scale Lightning Fatality Studies." In: *Weather, Climate, and Society* 8.1, pp. 35–42. DOI: 10.1175/WCAS-D-15-0032.1.

Houze Jr., Robert A. (2004). "Mesoscale convective systems." In: *Reviews of Geophysics* 42.4. DOI: 10.1029/2004RG000150.

Hubbert, John C. et al. (2018). "S-Pol's Polarimetric Data Reveal Detailed Storm Features (and Insect Behavior)." In: *Bulletin of the American Meteorological Society* 99.10, pp. 2045–2060. DOI: 10.1175/BAMS-D-17-0317.1.

James, Paul M., Bernhard K. Reichert, and Dirk Heizenreder (2018). "NowCastMIX: Automatic Integrated Warnings for Severe Convection on Nowcasting Time Scales at the German Weather Service." In: *Weather and Forecasting* 33.5, pp. 1413–1433. DOI: 10.1175/WAF-D-18-0038.1.

Jardines, Aniel, Manuel Soler, Alejandro Cervantes, Javier García-Heras, and Juan Simarro (2021). "Convection indicator for pre-

tactical air traffic flow management using neural networks." In: *Machine Learning with Applications* 5, p. 100053. ISSN: 2666-8270. DOI: 10.1016/j.mlwa.2021.100053.

Jardines, Aniel, Manuel Soler, Javier García-Heras, Matteo Ponzano, and Laure Raynaud (2024a). "Pre-tactical convection prediction for air traffic flow management using LSTM neural network." In: *Meteorological Applications* 31.3, e2215. DOI: 10.1002/met.2215.

Jardines, Aniel et al. (2024b). "Thunderstorm prediction during pre-tactical air-traffic-flow management using convolutional neural networks." In: *Expert Systems with Applications* 241, p. 122466. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2023.122466.

Jensen, J. L. W. V. (1906). "Sur les fonctions convexes et les inégalités entre les valeurs moyennes." In: *Acta Mathematica* 30.none, pp. 175–193. DOI: 10.1007/BF02418571.

Kain, John S. and J. Michael Fritsch (1990). "A One-Dimensional Entraining/Detraining Plume Model and Its Application in Convective Parameterization." In: *Journal of Atmospheric Sciences* 47.23, pp. 2784–2802. DOI: 10.1175/1520-0469(1990)047<2784:AODEPM>2.0.CO;2.

Kain, John S. et al. (2008). "Some Practical Considerations Regarding Horizontal Resolution in the First Generation of Operational Convection-Allowing NWP." In: *Weather and Forecasting* 23.5, pp. 931–952. DOI: 10.1175/WAF2007106.1.

Kaltenböck, Rudolf, Gerhard Diendorfer, and Nikolai Dotzek (2009). "Evaluation of thunderstorm indices from ECMWF analyses, lightning data and severe storm reports." In: *Atmospheric Research* 93.1. 4th European Conference on Severe Storms, pp. 381–396. ISSN: 0169-8095. DOI: 10.1016/j.atmosres.2008.11.005.

Kamangir, Hamid, Waylon Collins, Philippe Tissot, and Scott A. King (2020). "A deep-learning model to predict thunderstorms within 400 km2 South Texas domains." In: *Meteorological Applications* 27.2, e1905. DOI: 10.1002/met.1905.

Kerr, Christopher A. et al. (2025). "Limitations of Short-Term Thunderstorm Forecasts from Convection-Allowing Models with 3-km Horizontal Grid Spacing." In: *Weather and Forecasting* 40.1, pp. 223–234. DOI: 10.1175/WAF-D-24-0100.1.

Kingma, Diederik P. and Jimmy Ba (2014). *Adam: A Method for Stochastic Optimization*. DOI: 10.48550/ARXIV.1412.6980.

Kober, K., G. C. Craig, C. Keil, and A. Dörnbrack (2012). "Blending a probabilistic nowcasting method with a high-resolution numerical weather prediction ensemble for convective precipitation forecasts." In: *Quarterly Journal of the Royal Meteorological Society* 138.664, pp. 755–768. DOI: 10.1002/qj.939.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks." In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger. Vol. 25. Curran Asso-

ciates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning." In: *Nature* 521.7553, pp. 436–444. DOI: 10.1038/nature14539.

LeCun, Yann, John Denker, and Sara Solla (1989). "Optimal Brain Damage." In: *Advances in Neural Information Processing Systems*. Ed. by D. Touretzky. Vol. 2. Morgan-Kaufmann. URL: https://proceedings.neurips.cc/paper_files/paper/1989/file/6c9882bbac1c7093bd25041881277658-Paper.pdf.

Leinonen, J., U. Hamann, U. Germann, and J. R. Mecikalski (2022). "Nowcasting thunderstorm hazards using machine learning: the impact of data sources on performance." In: *Natural Hazards and Earth System Sciences* 22.2, pp. 577–597. DOI: 10.5194/nhess-22-577-2022.

Leinonen, J., U. Hamann, I. V. Sideris, and U. Germann (2023). "Thunderstorm Nowcasting With Deep Learning: A Multi-Hazard Data Fusion Model." In: *Geophysical Research Letters* 50.8. e2022GL101626 2022GL101626, e2022GL101626. DOI: 10.1029/2022GL101626.

Li, Jingmin, Caroline Forster, Johannes Wagner, and Thomas Gerz (2021). "Cb-Fusion–forecasting thunderstorm cells up to 6 hours." In: *Meteorologische Zeitschrift*, pp. 169–184.

Li, Wenyuan, Haonan Chen, Lei Han, and Jianliang Xu (2022). "The Interpretation of Deep Learning for Convective Storm Nowcasting." In: *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, pp. 7922–7925. DOI: 10.1109/IGARSS46834.2022.9883967.

Lin, Pin-Fang, Pao-Liang Chang, Ben Jong-Dao Jou, James W. Wilson, and Rita D. Roberts (2012). "Objective Prediction of Warm Season Afternoon Thunderstorms in Northern Taiwan Using a Fuzzy Logic Approach." In: *Weather and Forecasting* 27.5, pp. 1178–1197. DOI: 10.1175/WAF-D-11-00105.1.

Lin, Tianyang et al. (2019). "Attention-Based Dual-Source Spatiotemporal Neural Network for Lightning Forecast." In: *IEEE Access* 7, pp. 158296–158307. DOI: 10.1109/ACCESS.2019.2950328.

Lockey, Nicholas (2017). "Antonio Vivaldi and the sublime seasons: Sonority and texture as expressive devices in early eighteenth-century Italian music." In: *Eighteenth Century Music* 14.2, pp. 265–283. DOI: 10.1017/S1478570617000070.

Loken, Eric D., Adam J. Clark, and Christopher D. Karstens (2020). "Generating Probabilistic Next-Day Severe Weather Forecasts from Convection-Allowing Ensembles Using Random Forests." In: *Weather and Forecasting* 35.4, pp. 1605–1631. DOI: 10.1175/WAF-D-19-0258.1.

Loken, Eric D., Adam J. Clark, Ming Xue, and Fanyou Kong (2017). "Comparison of Next-Day Probabilistic Severe Weather Forecasts from Coarse- and Fine-Resolution CAMs and a Convection-Allowing

Ensemble." In: *Weather and Forecasting* 32.4, pp. 1403–1421. DOI: 10.1175/WAF-D-16-0200.1.

Lorenz, Edward N. (1963). "Deterministic Nonperiodic Flow." In: *Journal of Atmospheric Sciences* 20.2, pp. 130–141. DOI: 10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2.

– (1969). "The predictability of a flow which possesses many scales of motion." In: *Tellus* 21.3, pp. 289–307. DOI: 10.3402/tellusa.v21i3.10086.

Markowski, Paul and Yvette Richardson (2010). *Mesoscale Meteorology in Midlatitudes*. John Wiley & Sons, Ltd. ISBN: 978-0-470-68210-4. DOI: 10.1002/9780470682104.

Marquis, James N. et al. (2023). "Near-Cloud Atmospheric Ingredients for Deep Convection Initiation." In: *Monthly Weather Review* 151.5, pp. 1247–1267. DOI: 10.1175/MWR-D-22-0243.1.

Mohammed, Roweida, Jumanah Rawashdeh, and Malak Abdullah (2020). "Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results." In: *2020 11th International Conference on Information and Communication Systems (ICICS)*, pp. 243–248. DOI: 10.1109/ICICS49469.2020.239556.

Morzfeld, Matthias and Daniel Hodyss (2023). "A Theory for Why Even Simple Covariance Localization Is So Useful in Ensemble Data Assimilation." In: *Monthly Weather Review* 151.3, pp. 717–736. DOI: 10.1175/MWR-D-22-0255.1.

Mueller, C. et al. (2003). "NCAR Auto-Nowcast System." In: *Weather and Forecasting* 18.4, pp. 545–561. DOI: 10.1175/1520-0434(2003)018<0545:NAS>2.0.CO;2.

Murphy, Allan H. (1973). "A New Vector Partition of the Probability Score." In: *Journal of Applied Meteorology and Climatology* 12.4, pp. 595–600. DOI: 10.1175/1520-0450(1973)012<0595:ANVPOT>2.0.CO;2.

Necker, Tobias, David Hinger, Philipp Johannes Griewank, Takemasa Miyoshi, and Martin Weissmann (2023). "Guidance on how to improve vertical covariance localization based on a 1000-member ensemble." In: *Nonlinear Processes in Geophysics* 30.1, pp. 13–29. DOI: 10.5194/npg-30-13-2023.

Niculescu-Mizil, Alexandru and Rich Caruana (2005). "Predicting Good Probabilities with Supervised Learning." In: *Proceedings of the 22nd International Conference on Machine Learning*. ICML '05. Bonn, Germany: Association for Computing Machinery, pp. 625–632. ISBN: 1-59593-180-5. DOI: 10.1145/1102351.1102430.

Ortland, Stephanie M., Michael J. Pavolonis, and John L. Cintineo (2023). "The Development and Initial Capabilities of ThunderCast, a Deep Learning Model for Thunderstorm Nowcasting in the United States." In: *Artificial Intelligence for the Earth Systems* 2.4, e230044. DOI: 10.1175/AIES-D-23-0044.1.

Palmer, Tim, Andreas Döring, and G Seregin (2014). "The real butterfly effect." In: *Nonlinearity* 27.9, R123. DOI: 10.1088/0951-7715/27/9/R123.

Peters, John M., Daniel R. Chavas, Chun-Yian Su, Hugh Morrison, and Brice E. Coffer (2023). "An Analytic Formula for Entraining CAPE in Midlatitude Storm Environments." In: *Journal of the Atmospheric Sciences* 80.9, pp. 2165–2186. DOI: 10.1175/JAS-D-23-0003.1.

Porson, Aurore N. et al. (2020). "Recent upgrades to the Met Office convective-scale ensemble: An hourly time-lagged 5-day ensemble." In: *Quarterly Journal of the Royal Meteorological Society* 146.732, pp. 3245–3265. DOI: 10.1002/qj.3844.

Pozzolo, Andrea Dal, Olivier Caelen, Reid A. Johnson, and Gianluca Bontempi (2015). "Calibrating Probability with Undersampling for Unbalanced Classification." In: *2015 IEEE Symposium Series on Computational Intelligence*, pp. 159–166. DOI: 10.1109/SSCI.2015.33.

Prill, F, D Reinert, D Rieger, and G Zängl (2024). "ICON Tutorial: Working with the ICON model." In: *Deutscher Wetterdienst*. DOI: 10.5676/DWD_pub/nwv/icon_tutorial2024.

Pulkkinen, S., V. Chandrasekar, A. von Lerber, and A.-M. Harri (2020). "Nowcasting of Convective Rainfall Using Volumetric Radar Observations." In: *IEEE Transactions on Geoscience and Remote Sensing* 58.11, pp. 7845–7859. DOI: 10.1109/TGRS.2020.2984594.

Pulkkinen, S. et al. (2019). "Pysteps: an open-source Python library for probabilistic precipitation nowcasting (v1.0)." In: *Geoscientific Model Development* 12.10, pp. 4185–4219. DOI: 10.5194/gmd-12-4185-2019.

Quiñonero-Candela, Joaquin, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence (Dec. 2008). *Dataset Shift in Machine Learning*. The MIT Press. ISBN: 978-0-262-25510-3. DOI: 10.7551/mitpress/9780262170055.001.0001.

Rädler, Anja T, Pieter H Groenemeijer, Eberhard Faust, Robert Sausen, and Tomáš Púčik (2019). "Frequency of severe thunderstorms across Europe expected to increase in the 21st century due to rising instability." In: *npj Climate and Atmospheric Science* 2.1, p. 30. DOI: 10.1038/s41612-019-0083-7.

Ravuri, Suman et al. (2021). "Skilful precipitation nowcasting using deep generative models of radar." In: *Nature* 597.7878, pp. 672–677. DOI: 10.1038/s41586-021-03854-z.

Reinert, D et al. (2020). "DWD database reference for the global and regional ICON and ICON-EPS forecasting system." In: *Technical report Version 2.1. 8, Deutscher Wetterdienst*. URL: https://www.dwd.de/DWD/forschung/nwv/fepub/icon_database_main.pdf.

Richardson, D. S. (2000). "Skill and relative economic value of the ECMWF ensemble prediction system." In: *Quarterly Journal of the Royal Meteorological Society* 126.563, pp. 649–667. DOI: 10.1002/qj.49712656313.

Roberts, Nigel M. (2008). "Assessing the spatial and temporal variation in the skill of precipitation forecasts from an NWP model." In: *Meteorological Applications* 15.1, pp. 163–169. DOI: 10.1002/met.57.

Roberts, Nigel M. and Humphrey W. Lean (2008). "Scale-Selective Verification of Rainfall Accumulations from High-Resolution Forecasts of Convective Events." In: *Monthly Weather Review* 136.1, pp. 78–97. DOI: 10.1175/2007MWR2123.1.

Rougier, Jonathan (2016). "Ensemble Averaging and Mean Squared Error." In: *Journal of Climate* 29.24, pp. 8865–8870. DOI: 10.1175/JCLI-D-16-0012.1.

Saunders, Clive (2008). "Charge Separation Mechanisms in Clouds." In: *Planetary Atmospheric Electricity*. Ed. by F. Leblanc et al. New York, NY: Springer New York, pp. 335–353. ISBN: 978-0-387-87664-1. DOI: 10.1007/978-0-387-87664-1_22.

Schraff, C. et al. (2016). "Kilometre-scale ensemble data assimilation for the COSMO model (KENDA)." In: *Quarterly Journal of the Royal Meteorological Society* 142.696, pp. 1453–1472. DOI: 10.1002/qj.2748.

Schwartz, Craig S., Glen S. Romine, Kathryn R. Fossell, Ryan A. Sobash, and Morris L. Weisman (2017). "Toward 1-km Ensemble Forecasts over Large Domains." In: *Monthly Weather Review* 145.8, pp. 2943–2969. DOI: 10.1175/MWR-D-16-0410.1.

Schwartz, Craig S. et al. (2015). "A Real-Time Convection-Allowing Ensemble Prediction System Initialized by Mesoscale Ensemble Kalman Filter Analyses." In: *Weather and Forecasting* 30.5, pp. 1158–1181. DOI: 10.1175/WAF-D-15-0013.1.

Selz, Tobias and George C. Craig (2015). "Upscale Error Growth in a High-Resolution Simulation of a Summertime Weather Event over Europe." In: *Monthly Weather Review* 143.3, pp. 813–827. DOI: 10.1175/MWR-D-14-00140.1.

Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman (2014). "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps." In: *Workshop at International Conference on Learning Representations*.

Sobash, Ryan A., Glen S. Romine, and Craig S. Schwartz (2020). "A Comparison of Neural-Network and Surrogate-Severe Probabilistic Convective Hazard Guidance Derived from a Convection-Allowing Model." In: *Weather and Forecasting* 35.5, pp. 1981–2000. DOI: 10.1175/WAF-D-20-0036.1.

Sobash, Ryan A., Craig S. Schwartz, Glen S. Romine, Kathryn R. Fossell, and Morris L. Weisman (2016). "Severe Weather Prediction Using Storm Surrogates from an Ensemble Forecasting System." In: *Weather and Forecasting* 31.1, pp. 255–271. DOI: 10.1175/WAF-D-15-0138.1.

Sobash, Ryan A. et al. (2011). "Probabilistic Forecast Guidance for Severe Thunderstorms Based on the Identification of Extreme Phe-

nomena in Convection-Allowing Model Forecasts." In: *Weather and Forecasting* 26.5, pp. 714–728. DOI: 10.1175/WAF-D-10-05046.1.

Sovrasov, Vladislav (2018). *ptflops: a flops counting tool for neural networks in pytorch framework*. URL: https://github.com/sovrasov/flops-counter.pytorch.

Stephan, K., S. Klink, and C. Schraff (2008). "Assimilation of radar-derived rain rates into the convective-scale model COSMO-DE at DWD." In: *Quarterly Journal of the Royal Meteorological Society* 134.634, pp. 1315–1326. DOI: 10.1002/qj.269.

Stevens, Bjorn (2005). "Atmospheric Moist Convection." In: *Annual Review of Earth and Planetary Sciences* 33.Volume 33, 2005, pp. 605–643. ISSN: 1545-4495. DOI: 10.1146/annurev.earth.33.092203.122658.

Sun, Yanmin, Andrew KC Wong, and Mohamed S Kamel (2009). "Classification of imbalanced data: A review." In: *International journal of pattern recognition and artificial intelligence* 23.04, pp. 687–719.

Taszarek, Mateusz, John T Allen, Mattia Marchio, and Harold E Brooks (2021). "Global climatology and trends in convective environments from ERA5 and rawinsonde data." In: *npj climate and atmospheric science* 4.1, p. 35. DOI: 10.1038/s41612-021-00190-x.

Theis, S. E., A. Hense, and U. Damrath (2005). "Probabilistic precipitation forecasts from a deterministic model: a pragmatic approach." In: *Meteorological Applications* 12.3, pp. 257–268. DOI: 10.1017/S1350482705001763.

Thompson, Philip Duncan (1957). "Uncertainty of Initial State as a Factor in the Predictability of Large Scale Atmospheric Flow Patterns." In: *Tellus* 9.3, pp. 275–295. DOI: 10.3402/tellusa.v9i3.9111.

Toth, Zoltan, Olivier Talagrand, Guillem Candille, and Yuejian Zhu (2003). "Probability and ensemble forecasts." In: *Forecast verification: A practitioner's guide in atmospheric science* 137, p. 163.

Tuckman, Philip, Vince Agard, and Kerry Emanuel (2023). "Evolution of Convective Energy and Inhibition before Instances of Large CAPE." In: *Monthly Weather Review* 151.1, pp. 321–338. DOI: 10.1175/MWR-D-21-0302.1.

Turner, B. J., I. Zawadzki, and U. Germann (2004). "Predictability of Precipitation from Continental Radar Images. Part III: Operational Nowcasting Implementation (MAPLE)." In: *Journal of Applied Meteorology* 43.2, pp. 231–248. DOI: 10.1175/1520-0450(2004)043<0231:POPFCR>2.0.CO;2.

Ukkonen, Peter and Antti Mäkelä (2019). "Evaluation of Machine Learning Classifiers for Predicting Deep Convection." In: *Journal of Advances in Modeling Earth Systems* 11.6, pp. 1784–1802. DOI: 10.1029/2018MS001561.

Vallis, Geoffrey K. (2017). *Atmospheric and Oceanic Fluid Dynamics: Fundamentals and Large-Scale Circulation*. 2nd ed. Cambridge University Press.

Vannitsem, Stéphane et al. (2021). "Statistical Postprocessing for Weather Forecasts: Review, Challenges, and Avenues in a Big Data World." In: *Bulletin of the American Meteorological Society* 102.3, E681–E699. DOI: 10.1175/BAMS-D-19-0308.1.

Veraverbeke, Sander et al. (2017). "Lightning as a major driver of recent large fire years in North American boreal forests." In: *Nature Climate Change* 7.7, pp. 529–534. DOI: 10.1038/nclimate3329.

Vivekanandan, J. et al. (1999). "Cloud Microphysics Retrieval Using S-Band Dual-Polarization Radar Measurements." In: *Bulletin of the American Meteorological Society* 80.3, pp. 381–388. DOI: 10.1175/1520-0477(1999)080<0381:CMRUSB>2.0.CO;2.

Wallace, John M. and Peter V. Hobbs (2006). *Atmospheric Science. An Introductory Survey*. 2nd ed. San Diego: Academic Press. ISBN: 978-0-12-732951-2. DOI: 10.1016/B978-0-12-732951-2.50006-5.

Warder, Simon C., Kevin J. Horsburgh, and Matthew D. Piggott (2021). "Adjoint-based sensitivity analysis for a numerical storm surge model." In: *Ocean Modelling* 160, p. 101766. ISSN: 1463-5003. DOI: 10.1016/j.ocemod.2021.101766.

Wilks, Daniel S. (2019). *Statistical methods in the atmospheric sciences*. eng. 4. ed. Elsevier. ISBN: 978-0-12-815823-4. DOI: 10.1016/C2017-0-03921-6.

Wilson, James W., N. Andrew Crook, Cynthia K. Mueller, Juanzhen Sun, and Michael Dixon (1998). "Nowcasting Thunderstorms: A Status Report." In: *Bulletin of the American Meteorological Society* 79.10, pp. 2079–2100. DOI: 10.1175/1520-0477(1998)079<2079:NTASR>2.0.CO;2.

World Meteorological Organization (1957). "Definition of the tropopause." In: *Bulletin of the World Meteorological Organization* 6, pp. 136–137.

Yang, Ruyi et al. (2024). "Interpretable machine learning for weather and climate prediction: A review." In: *Atmospheric Environment* 338, p. 120797. ISSN: 1352-2310. DOI: 10.1016/j.atmosenv.2024.120797.

Yano, Jun-Ichi et al. (2018). "Scientific Challenges of Convective-Scale Numerical Weather Prediction." In: *Bulletin of the American Meteorological Society* 99.4, pp. 699–710. DOI: 10.1175/BAMS-D-17-0125.1.

Yasuda, Yoh, Shigeru Yokoyama, Masayuki Minowa, and Tomoyuki Satoh (2012). "Classification of lightning damage to wind turbine blades." In: *IEEJ Transactions on Electrical and Electronic Engineering* 7.6, pp. 559–566. DOI: 10.1002/tee.21773.

Zängl, Günther, Daniel Reinert, Pilar Rípodas, and Michael Baldauf (2015). "The ICON (ICOsahedral Non-hydrostatic) modelling framework of DWD and MPI-M: Description of the non-hydrostatic dynamical core." In: *Quarterly Journal of the Royal Meteorological Society* 141.687, pp. 563–579. DOI: 10.1002/qj.2378.

Zhang, F., Chris Snyder, and Richard Rotunno (2003). "Effects of Moist Convection on Mesoscale Predictability." In: *Journal of the*

Atmospheric Sciences 60.9, pp. 1173–1185. DOI: 10.1175/1520-0469(2003)060<1173:EOMCOM>2.0.CO;2.

Zhang, Yuchen et al. (2023). "Skilful nowcasting of extreme precipitation with NowcastNet." In: *Nature* 619.7970, pp. 526–532. DOI: 10.1038/s41586-023-06184-4.

Zhou, Kanghui, Jisong Sun, Yongguang Zheng, and Yutao Zhang (2022). "Quantitative Precipitation Forecast Experiment Based on Basic NWP Variables Using Deep Learning." In: *Advances in Atmospheric Sciences* 39.9, pp. 1472–1486.

Zhou, Kanghui, Yongguang Zheng, Bo Li, Wansheng Dong, and Xiaoling Zhang (2019). "Forecasting different types of convective weather: A deep learning approach." In: *Journal of Meteorological Research* 33, pp. 797–809. DOI: 10.1007/s13351-019-8162-6.

Zhu, Yuejian, Zoltan Toth, Richard Wobus, David Richardson, and Kenneth Mylne (2002). "The economic value of ensemble-based weather forecasts." In: *Bulletin of the American Meteorological Society* 83.1, pp. 73–84. DOI: 10.1175/1520-0477(2002)083<0073:TEVOEB>2.3.CO;2.

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede and Ivo Pletikosić. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

https://bitbucket.org/amiede/classicthesis/

The typographic style of this document is based on Palatino as the main serif font, chosen for its elegance and readability. Mathematical content is set in AMS Euler, lending a distinctive and classic appearance to formulae. The monospaced font used throughout is Bera Mono, selected for its clarity and aesthetic harmony with the other typefaces. Chapter numbers employ Euler fonts, contributing to the traditional and scholarly feel of the layout.

Final Version as of July 23, 2025 (classicthesis v4.8).