ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DOTTORATO DI RICERCA IN

IL FUTURO DELLA TERRA, CAMBIAMENTI CLIMATICI E SFIDE SOCIALI

Ciclo 37

Settore Concorsuale: 04/A4 - GEOFISICA

Settore Scientifico Disciplinare: GEO/12 - OCEANOGRAFIA E FISICA DELL'ATMOSFERA

DEVELOPMENTS IN MACHINE LEARNING DOWNSCALING FOR STORM SURGE IN THE NORTHERN ADRIATIC SEA

Presentata da: Rodrigo Vicente Campos Caba

Coordinatore Dottorato

Silvana Di Sabatino

Supervisore

Lorenzo Mentaschi

Co-supervisore

Jacopo Alessandri

Esame finale anno 2025

Borsa di dottorato del Programma Operativo Nazionale Ricerca e Innovazione 2014-2020 (CCI 2014IT16M2OP005), risorse FSE REACT-EU, Azione IV.4 "Dottorati e contratti di ricerca su tematiche dell'innovazione" e Azione IV.5 "Dottorati su tematiche Green."

Codice CUP borsa: J35F21003110006

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to:

My supervisor, Professor Lorenzo Mentaschi, for his constant support, motivation, and guidance throughout the successful development of this research. His insightful ideas and constructive feedback were of great value and played a key role in shaping the direction and quality of this work.

Carolina Padget, my wife, friend, and partner—the person who encouraged me to pursue this dream from the very beginning. Her love and companionship continue to inspire me every day, regardless of the challenges we face.

My parents, Reina and Vicente, and my siblings, Reina and Cristóbal, for their unwavering support throughout every challenge I have undertaken. Their love and warmth are always with me.

My colleagues and friends from the SINCEM lab: Italo (my companion and friend since day one), Jacopo (also my co-supervisor), Roberto, Veronica, and Enrico. Sharing and learning from their diverse worldviews has added great value and warmth to my doctoral journey.

Professor Paula Camus from the University of Cantabria (Santander, Spain), for her ongoing collaboration and willingness to share ideas. Her experience has undoubtedly enriched the research carried out.

Finally, Massimo Tondello, Silvia Beriotto, and Nicola Sguotti from HS Marine S.r.l. (Padova, Italy), who kindly welcomed me during a valuable and enriching work period. A special mention goes to Pietro Innangi, who became a true friend during my time at HS, a friendship that continues to this day.

ABSTRACT

Accurate storm surge prediction is critical for coastal resilience and disaster management, particularly in the face of intensifying climate change impacts. While Machine Learning (ML) models have shown potential for storm surge downscaling, their application often lacks systematic comparisons with high-resolution dynamical models and sufficient focus on extreme events, which are key for mitigating coastal hazards. This study addresses these gaps by combining advanced dynamical modeling and ML techniques to assess storm surge prediction in the Northern Adriatic Sea.

Using the SHYFEM model, high-resolution storm surge simulations were developed, incorporating optimized physical configurations and high-quality forcing datasets. This benchmark model achieved robust accuracy in capturing storm surge variability and extremes, providing a reliable baseline for evaluating ML models. To advance ML-based downscaling, this study implemented and compared models of varying complexity, from Multivariate Linear Regression (MLR) to advanced Long Short-Term Memory (LSTM) networks. A novel corrected mean absolute deviation (MADc) metric and a customized loss function ($MADc^2$) were introduced to enhance model validation and extreme event prediction.

The results revealed that simpler ML models, such as MLR, provided computationally efficient solutions but struggled with capturing non-linear patterns and extremes. Conversely, more complex algorithms, such as LSTM networks, demonstrated superior skill in capturing temporal dependencies and non-linearities, particularly when trained with $MADc^2$. Using the dynamical downscaling output to train the ML models showed that the $MADc^2$ variants achieved strong consistency with observations, highlighting their potential as efficient, low-cost alternatives to dynamical downscaling in the absence of high-quality forcing data. Furthermore, direct training on observed data in key locations, such as Punta della Salute and Trieste, revealed that several architectures, including LSTM, could outperform dynamical models in key metrics, emphasizing the critical importance of observational data.

These findings demonstrate the competitive performance of ML approaches, particularly when trained with high-quality modeling and observational data. Given the significant computational demands and costs associated with high-resolution numerical models, the results suggest that ML techniques offer a viable and efficient alternative for storm surge prediction. In the long run, the ability of ML to provide accurate predictions with lower resource requirements indicates its potential to surpass traditional numerical modeling, especially as data availability and computational methodologies continue to evolve.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

In the following sections, the physical and oceanographic characteristics of the Adriatic Sea are described, with a focus on the processes involved in storm surge generation and the mathematical tools commonly used for its representation. Additionally, since this study involves the implementation of Machine Learning (ML) algorithms, descriptions of various ML techniques are provided, with an emphasis on those specifically applied in this research. Finally, Section 1.5 outlines the objectives of this study.

As this research focuses on storm surges, the following terminology is adopted in this document, following the approach of Rus et al. (2023a) and the definitions provided by Gregory et al. (2019):

- Sea level: The total, time-varying local water depth.
- Sea surface height: The height of the sea level above (or below) the reference ellipsoid.
- Storm surge: The elevation or depression of the sea surface relative to the predicted tide during a storm.
- Storm tide: The sea surface height, elevated during a storm by the addition of a storm surge.

1.1 THE ADRIATIC SEA

The Adriatic Sea is a sub-basin of the Mediterranean Sea, bordered by Italy to the west, Slovenia, Croatia, Bosnia and Herzegovina, Montenegro, and Albania to the east (Vlachogianni et al., 2018). The basin is traditionally divided into three sub-basins based on its seafloor morphology: the northern, central, and southern sub-basins (Figure 1). The northern sub-basin, with an average depth of 35 m, exhibits distinct coastal characteristics. In the middle Adriatic, depths increase progressively from north to south, featuring two depressions reaching depths of approximately 250 m. The transition to the southern sub-basin is characterized by a sharp bathymetric gradient, with depths steeply increasing from around 200 m to over 1000 m (Oddo et al., 2005).

The coastline of the Adriatic is characterized by diverse geomorphological features. The western coast is relatively smooth and characterized by long sandy beaches and lagoons, while the eastern coast is more irregular, with numerous islands, peninsulas,

and rocky shorelines (Asciutto et al., 2024). This geomorphological difference is a direct result of the contrasting tectonic histories of the two coasts. The dynamic bathymetry and coastal configuration significantly impact the physical oceanographic processes, such as circulation patterns, wave dynamics, and storm surge events (Bellafiore & Umgiesser, 2010).

Moreover, the semi-enclosed nature of the Adriatic Sea, coupled with its unique geomorphological features, predisposes it to experiencing intense storm surge events. These events are primarily driven by cyclogenesis and associated low-pressure systems, which form in the vicinity of the Adriatic, often influenced by the surrounding orographic features, such as the Dinaric Alps and the Apennines (Umgiesser et al., 2021).

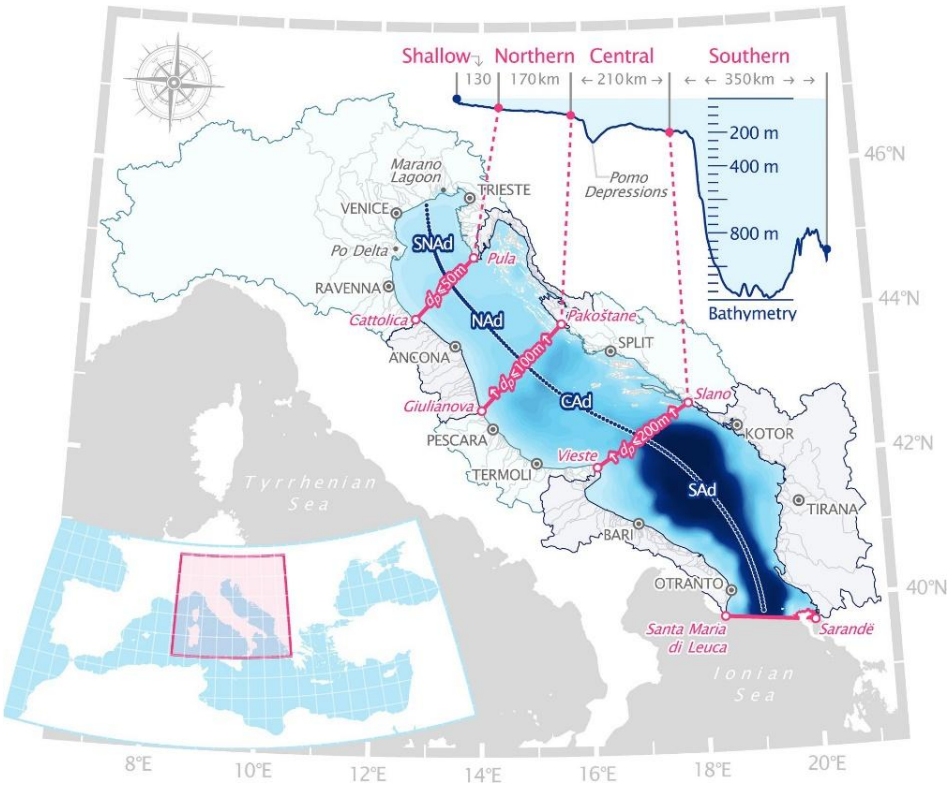

Figure 1: Adriatic Sea subregions defined by bathymetry: Shallow Northern Adriatic Sea (SNAd), Northern Adriatic Sea (NAd), Central Adriatic Sea (CAd), and Southern Adriatic Sea (SAd). The shaded areas represent the hydrological catchments of each subregion, whilst the embedded graphics show the bathymetry along the blue circles' path (top) and the domain position within the Mediterranean Region (bottom). Source: Aragão et al. (2024).

1.1.1 WIND REGIMES AND SEA LEVEL PRESSURE IN THE ADRIATIC SEA

The Adriatic Sea's wind regime is dominated by two principal winds: the Bora and the Sirocco (Ostoich et al., 2018). These winds play a significant role in shaping the region's meteorological and oceanographic conditions, particularly influencing storm surge events.

The Bora is a cold, dry, and gusty northeasterly wind that originates from the Dinaric Alps and flows towards the Adriatic Sea. It is most frequent and intense during the winter months, particularly from November to March, though it can also occur sporadically in autumn and spring (Jeromel et al., 2009). The Bora is a katabatic wind (winds that flow downhill), driven by a steep pressure gradient between a high-pressure system over the Balkans and a low-pressure system over the Adriatic. It often reaches speeds of 55-90 km/h, with gusts sometimes exceeding 185 km/h during severe episodes. The Bora is strongest along the eastern Adriatic coastline, particularly in the Gulf of Trieste and the Kvarner Bay (Jurčec, 1981; Signell et al., 2010).

The Sirocco (Figure 2), in contrast, is a warm, humid southeasterly wind that originates from the Sahara Desert and blows across the Mediterranean Sea towards the Adriatic (Ferrarese et al., 2009). It is most prevalent during the transitional seasons of spring (April to June) and autumn (September to November), but it can also occur during the winter months. The Sirocco is typically associated with cyclonic systems moving from North Africa or the central Mediterranean towards Europe. It can bring heavy rainfall, reduced visibility due to dust, and significant storm surge along the western Adriatic coast. Wind speeds during Sirocco events generally range from 37-74 km/h, but they can exceed 93 km/h during intense storms. Unlike the cold Bora, the Sirocco often raises temperatures and humidity levels, creating challenging conditions for coastal regions (Bertotti et al., 2011).

In addition to the wind regimes, sea level pressure (SLP) patterns play a crucial role in modulating storm surge events in the Adriatic Sea. One of the key processes contributing to it is the inverse barometer effect. This phenomenon occurs when a drop in atmospheric pressure leads to a corresponding rise in sea level. In simple terms, as the atmospheric pressure decreases, there is less force pressing down on the sea surface, allowing the water to rise. In general, a 1 hPa drop in pressure typically results in a sea level rise of approximately 1 cm (Lionello et al., 2012). Specifically, low-pressure centres over the Gulf of Genoa or the northern Adriatic contribute to the generation of storm surge events (Šepić et al., 2022).

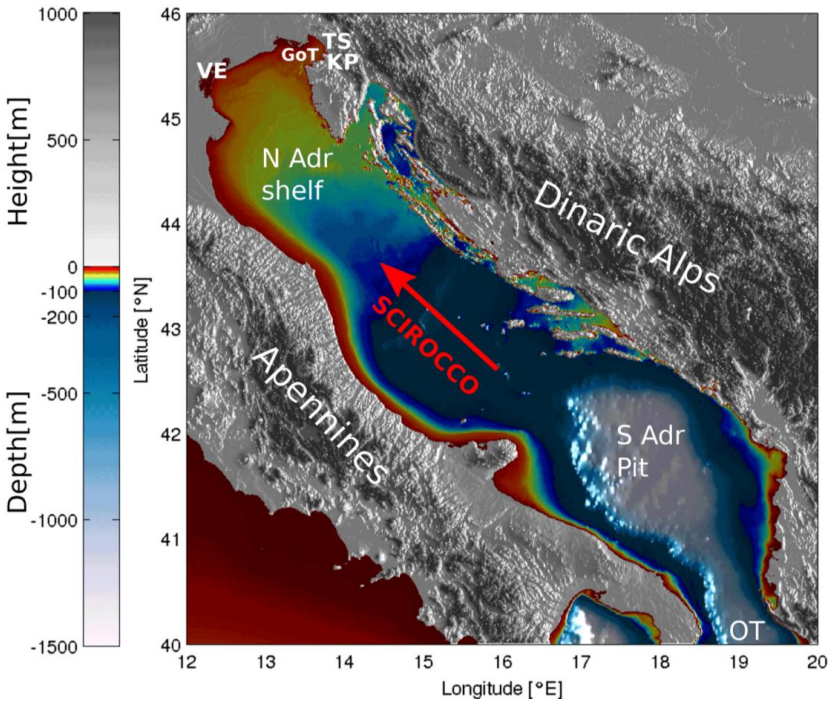

Figure 2: Adriatic orography and bathymetry. TS: Trieste; KP: Koper; GoT: Gulf of Trieste; VE: Venice; N Adr Shelf: Northern Adriatic Shelf; S Adr Pit: Sourthern Adriatic Pit; OT: Otranto Strait. Direction of Scirocco is marked with a red arrow. Source: Žust et al. (2021).

1.1.2 TIDES IN THE ADRIATIC SEA

Tides are the regular rise and fall of sea levels caused by the combined gravitational effects of the Moon, the Sun, and the rotation of the Earth. They result in periodic changes in sea level at any given location, which can vary in amplitude and timing depending on the region's geographical and oceanographic characteristics.

In the Adriatic Sea, the tides are relatively modest compared to other parts of the world, primarily because it is a semi-enclosed basin with a complex bathymetry. The tidal regime in the Adriatic is mixed, with both diurnal (once a day) and semidiurnal (twice a day) components, but semidiurnal tides dominate (Lionello et al., 2021). The most significant contributions come mainly from seven principal constituents (Defant, 1961), four semi-diurnal (M2, S2, N2, and K2) and three diurnal (K1, 01, and P1).

The M2 tidal constituent, which is the lunar semidiurnal tide, is the most significant tidal component in the Adriatic Sea. Its amplitude varies across the basin, generally ranging from 10 to 25 cm, with the highest values observed in the northern part of the Adriatic. The S2 constituent, which is the solar semidiurnal tide, has a smaller amplitude of around 5 to 10 cm. Other notable tidal constituents in the Adriatic include the K1 (lunar-solar diurnal) and O1 (lunar diurnal) tides, with amplitudes typically around 5 to 7 cm (Malačič et al., 2000).

In addition to these primary constituents, there are shallow water tides and harmonic overtides that become more significant in certain areas of the Adriatic, especially in narrow straits and shallow regions where nonlinear interactions between the tide and bathymetry occur (Pugh & Woodworth, 2014). For example, the M4 and M6 overtides can sometimes be observed in the northern Adriatic, where the shallow bathymetry amplifies these higher frequency components, although their amplitudes are relatively small compared to the main constituents.

Another key feature of the tidal dynamics in the Adriatic Sea is the presence of amphidromic points. These are locations where the tidal range is essentially zero, meaning that the sea level at these points does not rise or fall during the tidal cycle (Desplanque & Mossman, 2004). Around these points, the tide rotates, with increasing amplitude as you move away from the center. In the Adriatic Sea, there is an amphidromic point located near the central part of the basin, roughly between the coasts of Italy and Croatia, which is observed for M2 constituent and the other semi-diurnal components at the border between northern and middle Adriatic (Figure 3). This amphidromic system contributes to the spatial variation in tidal amplitudes observed across the basin (Defant, 1961).

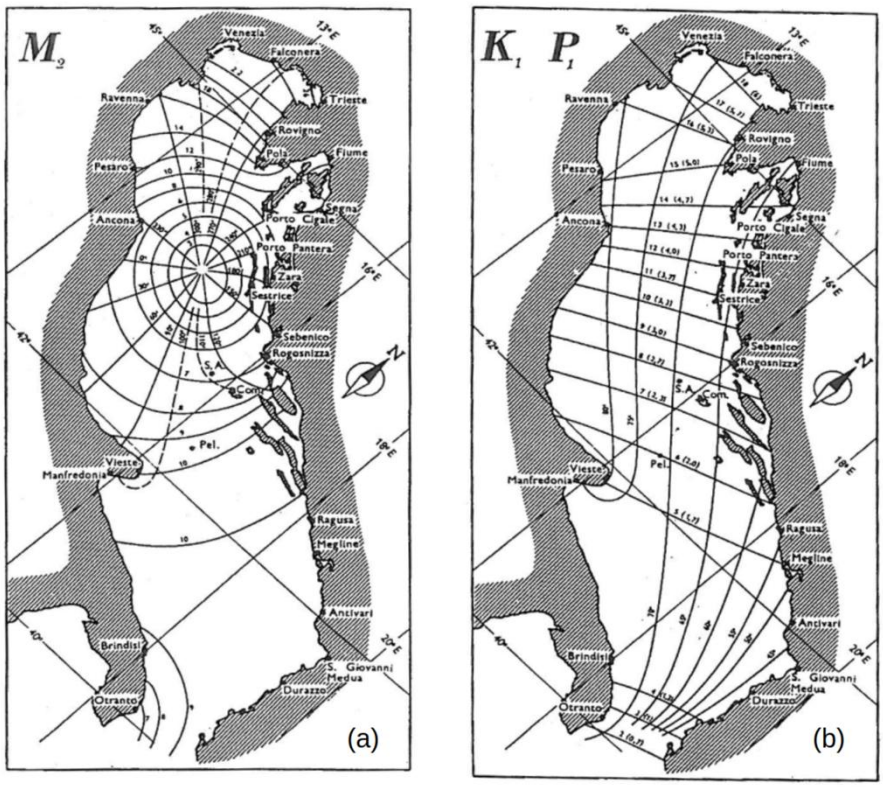

Figure 3: Lines of equal tidal amplitude (in cm) and phase in the Adriatic Sea, for: (a) M2 constituent; (b) K1 and P1 constituents. All the other semi-diurnal and diurnal constituents have similar distribution as in (a) and (b) respectively. Source: Polli (1959).

1.2 STORM SURGE

Storm surges are atmospherically forced oscillations of the water level in a coastal or inland water body, occurring over periods ranging from a few minutes to several days. By this definition, storm surges are distinct from wind waves and swell, which have much shorter periods, typically lasting only a few to several seconds (Murty et al., 1986).

Storm surges belong to the same class of wave as tides and tsunamis, that is, long gravity waves. As can be seen in Figure 4, storm surges are centered at about 10^{-4} cycles per second (cps, or hertz, Hz), which gives a period of approximately three hours.

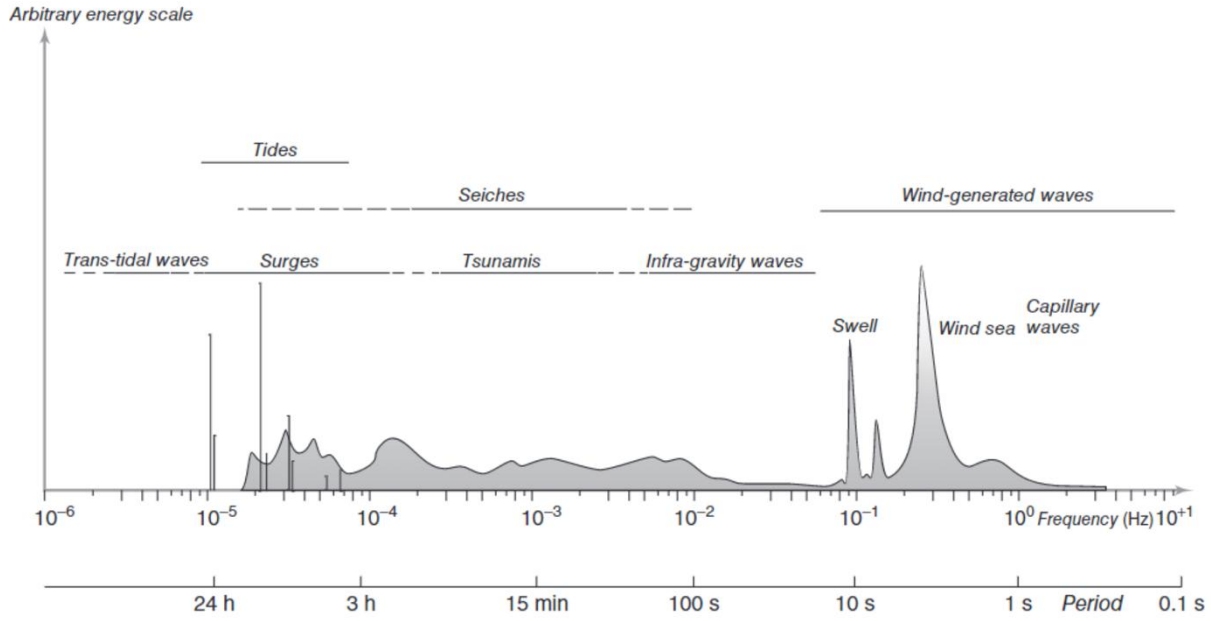

Figure 4: Frequencies and periods of the vertical motions of the ocean surface. Source: Holthuijsen (2007).

The storm surge consists of two main components: the decrease in sea level pressure (inverse barometric effect) and the increase in wind speed. The inverse barometer effect, also known as static amplification, occurs when a drop in atmospheric pressure causes a corresponding rise in sea level, typically accounting for 10 to 15% of the total storm surge magnitude (World Meteorological Organization, 2011). This static amplification is particularly pronounced in semi-enclosed basins like the Adriatic Sea, where limited exchange with the open ocean can lead to the accumulation of water over extended periods.

The more significant contributor to storm surge is the wind setup, also known as dynamic amplification, which occurs when strong winds exert tangential stress on the ocean surface, pushing water towards the coast, thereby causing a pile-up of water at the coast. The storm surge and wind speed follow the relationship show in Equation 1 (World Meteorological Organization, 2011).

$$S = K * \frac{w^2}{D} \quad (1)$$

Where S is the storm surge amplitude, w the wind speed, D the depth of water and K a constant encompassing several other factors, such as bottom stress, stratification of the water body, atmospheric stability, nature of the ocean surface (rough versus smooth) and the angle at which the wind is blowing. As can be seen, the surge amplitude is directly proportional to the square of the wind speed. Hence, if the wind speed doubles, the surge height increases fourfold (Pugh, 1987).

In the Adriatic Sea, this phenomenon is often associated with the Bora and Sirocco winds, which can generate significant sea level anomalies, particularly in the northern part of the basin (Chaumillon et al., 2017). The shallow bathymetry of the Northern Adriatic further amplifies the impact of these winds, leading to severe storm surge events that can cause coastal flooding and damage to infrastructure (Lionello et al., 2012; Umgiesser et al., 2021).

1.2.1 IMPACTS OF STORM SURGE

In coastal areas, accurately depicting storm surge is paramount for effective risk assessment, preparedness, and mitigation strategies, as they can lead to coastal erosion, inundation, and infrastructure damage, and threaten important cultural heritage sites (Reimann et al., 2018; Vousdoukas et al., 2022). The immediate impacts of storm surges include extensive property damage, loss of life, and severe economic disruptions. For instance, the storm surge from Hurricane Ike affected approximately 143,598 people, flooding 55% of the floodplain and demonstrating the extensive reach of such natural disasters (Al-Attabi et al., 2023).

Beyond immediate destruction, storm surges contribute to long-term environmental degradation. They accelerate coastal erosion, damage vital habitats such as wetlands and mangroves, and undermine natural coastal defenses. The loss of these protective ecosystems exacerbates the vulnerability of coastal areas to future storm surges, creating a cycle of increasing risk and damage (Herrera Silveira et al., 2022).

The economic ramifications are profound, encompassing the costs of infrastructure repair, loss of business continuity, and the financial burden of disaster response and recovery. In China, for example, storm surges have been identified as a significant cause of financial damage, highlighting the global economic threat posed by these events (Jin et al., 2018).

Furthermore, storm surges can lead to prolonged power outages by inundating substations and other critical infrastructure, disrupting essential services, and hindering

recovery efforts. Studies have shown that the compounded impact of hurricanes and storm surges can amplify the vulnerability of affected communities, emphasizing the need for comprehensive risk assessments and resilient infrastructure planning (Poudyal et al., 2023).

In the Northern Adriatic Sea, the study of impacts due to storm surge is of great importance, primarily due to its status as a high-risk area with unique cultural and environmental heritage, as well as significant economic activities (Ferrarin et al., 2020). Some relevant storm surge events occurred in the Northern Adriatic Sea are described below:

- November 4, 1966: One of the worst floods in Venice's recorded history. Almost the entire city was submerged, causing extensive damage to infrastructure, homes, and cultural sites. This event raised awareness about Venice's vulnerability to flooding and led to increased efforts to protect the city (De Zolt et al., 2006).

- November 16, 2002: The event was driven by intense winds and atmospheric pressure conditions, causing significant coastal flooding and impacting urban areas. This period helped refine predictive models and underscored the need for reliable monitoring systems and real-time response measures (Zampato et al., 2006).

- October 29, 2018: This event affected the Veneto and Friuli-Venezia Giulia regions. The peak surge reached approximately 1.56 meters above sea level in Venice, leading to flooded squares, streets, and buildings, particularly in the city's historic center. The extreme waves reached up to 6 meters offshore near Venice, causing significant coastal erosion and damaging infrastructure. The Italian government declared a state of emergency due to the storm's widespread destruction (Ferrarin et al., 2020).

- November 12, 2019: Severe flooding affected around 85% of Venice, damaging historic buildings, businesses, and homes. This event prompted emergency measures and underscored the need for the MOSE flood barrier system, which was under development at the time. The event also caused economic losses estimated at hundreds of millions of euros (Umgiesser et al., 2021; Giesen et al., 2021).

- November 22, 2022: Thanks to the operation of the Experimental Electromechanical Module (MOSE, from Italian Modulo Sperimentale Elettromeccanico) barriers, Venice was largely spared from extensive flooding, demonstrating the system's effectiveness. However, this storm surge event highlighted the ongoing threat from rising sea levels and severe weather, as well as the essential role of infrastructure adaptations in protecting vulnerable coastal cities (Mel et al., 2023).

Given the escalating frequency and intensity of storm surges due to climate change, there is an urgent need for accurate predictive models to inform preparedness and mitigation strategies. Developing robust storm surge models is essential for safeguarding lives, protecting property, and preserving the ecological integrity of coastal regions. Such models enable policymakers and stakeholders to implement effective coastal management practices, enhance early warning systems, and design resilient infrastructure capable of withstanding future storm surge events (Melet et al., 2024).

1.3 NUMERICAL SIMULATIONS WITH UNSTRUCTURED GRIDS

Numerical simulations play a pivotal role in unraveling the complexities of physical phenomena, such as storm surge (Park et al., 2022). They offer invaluable insights into various processes and greatly contribute to building extensive databases for further analysis and comprehension. Concerning storm surge, this refers to a complex oceanographic phenomenon that demands accurate oceanic and atmospheric data for precise representation. Due to diverse orographic configurations, atmospheric models often exhibit significant errors, necessitating the utilization of local-scale models with high resolution (Umgiesser et al., 2021). Additionally, the intricate coastal and bathymetric features and interactions pose challenges for existing hydrodynamical models to fully capture the relevant dynamics, partly due to their low resolution (Mentaschi et al., 2015; Toomey et al., 2022).

On the other hand, the utilization of unstructured grid models enables a more accurate portrayal of coastal dynamics, considering the intricacies of bathymetry and shoreline configurations (Federico et al., 2017). This approach offers the advantage of employing higher resolution at the coastlines while maintaining more modest resolution in deeper waters (Ferrarin et al., 2019). Unstructured meshes offer flexibility in resolving basin geometry, allowing for local refinement of computational domains to simulate regional dynamics on a global mesh with coarse resolution. This

flexibility is particularly valuable for coastal applications, where computational domains encompass complex coastlines and varying scales, ranging from basin size to details of river estuaries or riverbeds (Danilov, 2013). Over recent years, unstructured grid models have increasingly emerged as alternatives to regular grids for large-scale simulations (e.g. Mentaschi et al., 2020; Muis et al., 2016; Vousdoukas et al., 2018; Fernández-Montblanc et al., 2020; Saillour et al., 2021; Wang et al., 2022; Zhang et al., 2023; Mentaschi et al., 2023), with established circulation unstructured models like ADCIRC (Luettich et al., 1992; Pringle et al., 2021), the Finite-Volume Coastal Ocean Model (FVCOM, Chen et al., 2003), the Semi-implicit Cross-scale Hydroscience Integrated System Model (SCHISM, Zhang & Baptista, 2008; Zhang et al., 2016), the System of HydrodYnamic Finite Element Modules (SHYFEM, Umgiesser et al., 2004; Bellafiore & Umgiesser, 2010) and its MPI version (SHYFEM-MPI; Micaletto et al., 2022), the model TELEMAC (Hervouet & Bates, 2000), Delft3D-FM (Deltares: Delft, 2024), among others.

1.3.1 STORM SURGE NUMERICAL SIMULATIONS APPLIED TO THE NORTHERN ADRIATIC SEA

Storm surge modeling in semi-enclosed basins like the Adriatic Sea has long been a subject of scientific investigation, due to the combination of complex bathymetry, coastal morphology, and meteorological forcings such as bora and sirocco winds. Over the years, various numerical approaches have been developed and tested to improve surge prediction accuracy and to evaluate the sensitivity of surge response to changing environmental and climatic conditions.

One of the earliest comparative studies by De Vries et al. (1995) examined the performance of five two-dimensional storm surge models across three European seas, including the Adriatic Sea. Their work underscored the importance of using standardized input data such as bathymetry and meteorological forcing. While model discrepancies were generally small when using a consistent setup, significant underestimations of storm surge heights (up to 50 cm) highlighted the need for improving shared components such as surface drag formulations. Their findings also pointed to the persistent nature of seiches in the relatively deep basins of the Mediterranean, emphasizing the role of accurate initial conditions.

Yu et al. (1998) focused specifically on the Adriatic Sea and constructed a storm surge model to simulate historical and hypothetical future events. They found that increasing wind intensity had the most pronounced effect on surge heights, while sea-level rise slightly mitigated the surge magnitude in the northern basin due to bathymetric characteristics. This research emphasized the need to account for evolving topography and climate-driven changes when projecting future coastal hazards.

The importance of high-quality atmospheric forcing was reinforced by Zampato et al. (2006), who analyzed an extreme storm surge event in 2002 using a finite element hydrodynamic model. Their work demonstrated that different sources of wind input significantly affected model performance. Notably, model accuracy improved when wind intensities were empirically adjusted, indicating that even state-of-the-art atmospheric products may underestimate coastal wind forcing during extreme events.

Recognizing the limitations of purely physics-based approaches, Bajo and Umgiesser (2009) introduced a hybrid modeling system combining hydrodynamic simulations with artificial neural networks to enhance surge predictions near Venice. This operational system halved the first-day forecast error compared to the standalone hydrodynamic model, demonstrating the added value of data-driven approaches in coastal forecasting.

In terms of long-term climate projections, Lionello et al. (2010) employed regional climate models and shallow water hydrodynamics to assess changes in extreme storm events at the Venetian littoral under different IPCC scenarios. Although scenario simulations suggested stronger future storms—particularly under the B2 scenario—the study concluded that the evidence for significantly increased storminess was not statistically robust, mainly due to the limited ensemble size and the dominance of multidecadal variability.

Coastal and inlet-scale dynamics have also been explored in three dimensions. Bellafiore and Umgiesser (2010) implemented a 3D finite element model to investigate baroclinic processes around the Venice Lagoon in unprecedented detail. Their simulations identified wind forcing as the primary driver of surface currents, with bora and sirocco producing distinct spatial patterns. The study highlighted the role of advection and baroclinic pressure gradients in shaping persistent coastal vorticity structures, with implications for transport and mixing processes along the lagoon boundary.

To improve operational forecasting, Mel and Lionello (2014) developed and verified an Ensemble Prediction System (EPS) for storm surges in the northern Adriatic. Using the HYPSE model and ECMWF ensemble meteorological inputs, they demonstrated that ensemble mean forecasts consistently outperformed deterministic predictions, particularly for short lead times. The system showed skill in estimating probability distributions of sea level, offering a probabilistic framework for surge forecasting that accounts for forecast uncertainty.

Međugorac et al. (2018) introduced a novel perspective by investigating cross-basin sea-level slopes during surge events. By analyzing long-term tide gauge data from both the eastern and western Adriatic coasts, they revealed that atmospheric conditions can induce significant sea-level gradients across the basin, leading to asymmetrical coastal impacts. Their work underscored the importance of considering cross-basin dynamics when interpreting or forecasting storm surge events.

In a later study, Bajo et al. (2019) examined how the Adriatic's natural seiches, easily excited due to the basin's morphology, interact with storm surges. They used a hybrid modeling setup and Ensemble Kalman Filter data assimilation to show that assimilating tide gauge data significantly improves forecast accuracy even when wind forcing is deficient. Their results highlighted that non-linear interactions between sea level components can influence the seiche decay time, and that persistent oscillations can impact coastal sea level days after a storm event.

More recently, Alessandri et al. (2023) developed an ensemble prediction system tailored for lagoons and transitional environments, which are particularly vulnerable to surge-related flooding. Using a high-resolution finite element model and 45 ensemble members, they evaluated five recent events and showed improved forecast skill, especially by the third day. Their work emphasized that initial and lateral boundary conditions, including tidal components, are the main sources of uncertainty in surge prediction, and demonstrated the value of probabilistic forecasts for coastal management.

In this study, the SHYFEM-MPI model was used to carry out numerical simulations in the Northern Adriatic Sea, which is described in the following Section 1.3.1.

1.3.2 SYSTEM OF HYDRODYNAMIC FINITE ELEMENT MODULES

The SHYFEM-MPI model is an unstructured-grid finite element hydrodynamic open-source code that solves the Navier-Stokes equations with hydrostatic and Boussinesq approximations (Umgiesser et al., 2004; Micaletto et al., 2022), allowing for flexible resolution in areas with complex bathymetry and coastline geometry. This makes it particularly well-suited for simulating the intricate physical processes that occur in the Adriatic Sea, such as the interaction between wind, tides, and storm surge.

SHYFEM-MPI utilizes staggered finite elements in an unstructured Arakawa B horizontal grid, with the vertices of the triangle elements referred to as nodes. Vectors (velocity) are calculated at the center of each element, while scalars (temperature, salinity, and water levels) are determined at nodes (Federico et al., 2017).

The model has been already implemented in operational (Federico et al., 2017) and relocatable forecasting frameworks (Trotta et al., 2016), and for storm surge events (Park et al., 2022; Alessandri et al., 2023).

GOVERNING EQUATIONS

Related to the governing equations on SHYFEM, the horizontal momentum equations integrated over a vertical layer l are defined as is shown on Equations 2 and 3.

$$\frac{\partial U_l}{\partial t} + u_l \frac{\partial U_l}{\partial x} + v_l \frac{\partial U_l}{\partial y} + \int_{z_l}^{z_{l-1}} w \frac{\partial u}{\partial z}\, dz - fV_l = -gh_l \frac{\partial \zeta}{\partial x} - \frac{gh_l}{\rho_0} \int_{H_l}^{0} \frac{\partial \rho'}{\partial x}\, dz - \frac{h_l}{\rho_0} \frac{\partial P_a}{\partial x} + \nabla_h \cdot (A_H \nabla_h U_l) + \int_{z_l}^{z_{l-1}} \frac{\partial \tau_{xz}}{\partial z}\, dz \quad (2)$$

$$\frac{\partial V_l}{\partial t} + u_l \frac{\partial V_l}{\partial x} + v_l \frac{\partial V_l}{\partial y} + \int_{z_l}^{z_{l-1}} w \frac{\partial v}{\partial z}\, dz + fU_l = -gh_l \frac{\partial \zeta}{\partial y} - \frac{gh_l}{\rho_0} \int_{H_l}^{0} \frac{\partial \rho'}{\partial y}\, dz - \frac{h_l}{\rho_0} \frac{\partial P_a}{\partial y} + \nabla_h \cdot (A_H \nabla_h V_l) + \int_{z_l}^{z_{l-1}} \frac{\partial \tau_{yz}}{\partial z}\, dz \quad (3)$$

Here, $\zeta = \zeta(x, y, t)$ represents the free surface; $l = 1 \ldots N$ is the vertical layer index, starting with $l = 1$ for the surface layer and increasing with depth, with $l = N$ corresponding to the bottom layer; $z_l = 0 \ldots N$ denotes the layer interfaces, where $z_0 = 0$ represents the surface and $z_N = N$ the bottom interface. H_l is the layer

thickness, P_a represents the atmospheric pressure at the sea surface, g is the gravitational acceleration, ρ_0 is the reference density of seawater, $\rho = \rho_0 + \rho'$ is the water density with ρ' denoting the density perturbation from the reference value ρ_0, H_l is the depth of the bottom of layer l, A_H is the horizontal eddy viscosity calculated using the Smagorinsky formulation (Smagorinsky, 1963; Blumberg & Mellor, 1987), and w is the vertical velocity within the layer.

u_l and v_l are the horizontal velocities, while U_l and V_l are the horizontal velocities integrated over layer l (transport), as defined by Equations 4 and 5.

$$U_l = \int_{z_l}^{z_{l-1}} u_l \, dz \tag{4}$$

$$V_l = \int_{z_l}^{z_{l-1}} v_l \, dz \tag{5}$$

τ_{xz} and τ_{yz} are the turbulent Reynolds stresses at the top and bottom of each layer, defined by Equations 6 and 7.

$$\int_{z_l}^{z_{l-1}} \frac{\partial \tau_{xz}}{\partial z} \, dz = \tau_{xz}^{z_{l-1}} - \tau_{xz}^{z_l} = Av \frac{\partial u_l}{\partial z} \tag{6}$$

$$\int_{z_l}^{z_{l-1}} \frac{\partial \tau_{yz}}{\partial z} \, dz = \tau_{yz}^{z_{l-1}} - \tau_{yz}^{z_l} = Av \frac{\partial v_l}{\partial z} \tag{7}$$

For the first layer ($l = 1$), the stress terms $\tau_{xz}^{z_0}$ and $\tau_{yz}^{z_0}$ are defined by the momentum surface boundary condition (Equations 21 and 22). For the last layer ($l = N$), the terms $\tau_{xz}^{z_N}$ and $\tau_{yz}^{z_N}$ are determined by the bottom boundary condition (Equations 23 and 24).

The continuity equation integrated over a vertical layer l is shown in Equation 8.

$$\frac{\partial U_l}{\partial x} + \frac{\partial V_l}{\partial y} = w_{z_l} - w_{z_{l-1}} \tag{8}$$

The layer integrated salinity and temperature equations are shown in Equations 9 and 10, respectively.

$$\frac{\partial(h_l S_l)}{\partial t} + U_l \frac{\partial S_l}{\partial x} + V_l \frac{\partial S_l}{\partial y} + \int_{z_l}^{z_{l-1}} w \frac{\partial S}{\partial z}\, dz = \nabla_h \cdot (K_H \nabla_h h_l S_l) + \int_{z_l}^{z_{l-1}} \frac{\partial}{\partial z}\left(K_V \frac{\partial S}{\partial z}\right) dz \tag{9}$$

$$\frac{\partial(h_l \theta_l)}{\partial t} + U_l \frac{\partial \theta_l}{\partial x} + V_l \frac{\partial \theta_l}{\partial y} + \int_{z_l}^{z_{l-1}} w \frac{\partial \theta}{\partial z}\, dz = \nabla_h \cdot (K_H \nabla_h h_l \theta_l) + \int_{z_l}^{z_{l-1}} \frac{\partial}{\partial z}\left(K_V \frac{\partial \theta}{\partial z}\right) dz + \int_{z_l}^{z_{l-1}} \frac{I}{\rho_0 C_p} dz \tag{10}$$

Where K_H and K_V are the horizontal and vertical turbulent diffusion coefficients, respectively, and S_l and θ_l are the salinity and temperature in layer l. For both Equation 9 and Equation 10, in the first layer ($l = 1$) and the last layer ($l = N$), the surface and bottom boundary conditions are defined for the last term on the left-hand side with the vertical velocity boundary condition presented on Equation 19, and for the second term on the right-hand side with the flux boundary conditions (Equations 26 and 27). The term I in Equation 10 represents the solar irradiance at depth z, parameterized using a double exponential according to Paulson & Simpson (1977), as defined in Equation 11.

$$\frac{I}{I_0} = Re^{-z/\xi_1} + (1 - R)e^{-z/\xi_2} \tag{11}$$

Where I_0 is the irradiance at the surface, and ξ_1 and ξ_2 are the attenuation lengths for the portion of the surface radiation in the visible spectrum.

The set of governing equations is completed with the in-situ density ρ, computed from the salinity, temperature and pressure according to the United Nations Educational, Scientificc and Cultural Organization (UNESCO) equation of state (Fofonoff & Millard, 1983), shown in Equation 12.

$$\rho_l(x, y, l, t) = \rho_l(s_l, \theta_l, p_l) \tag{12}$$

TURBULENCE MODEL

For the turbulence model, the vertical eddy viscosity (A_V) and diffusivity (K_V) are computed using a two-equation model based on a $k-\varepsilon$ scheme for turbulence closure, implemented in the General Ocean Turbulence Model (GOTM) (Burchard & Petersen, 1999), which is integrated into the SHYFEM-MPI code. A_V and K_V are obtained using the relations of Kolmogorov (1941) and Prandtl (1945), which relate the turbulent coefficients to a velocity and a turbulent length scale. These expressions are shown in Equations 13 and 14.

$$A_V = c_\mu \sqrt{k} l + v_v \tag{13}$$

$$K_V = c'_\mu \sqrt{k} l + \gamma_v \tag{14}$$

Where k is the turbulent kinetic energy, l is a turbulent length scale, and v_v and γ_v are the molecular viscosity and diffusivity, respectively, while c_μ and c'_μ are dimensionless stability functions. To determine the vertical turbulent coefficients, the GOTM model solves equations for the turbulent kinetic energy (k) and turbulence dissipation (ε), as defined in Equations 15 and 16.

$$\frac{\partial k}{\partial t} + \vec{U} \cdot \nabla k = \frac{\partial}{\partial z}\left(\frac{A_v}{\sigma_k}\frac{\partial k}{\partial z}\right) + P_s + B - \varepsilon \tag{15}$$

$$\frac{\partial \varepsilon}{\partial t} + \vec{U} \cdot \nabla \varepsilon = \frac{\partial}{\partial z}\left(\frac{A_v}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial z}\right) + \frac{\varepsilon}{k}(c_{\varepsilon 1} P_s + c_{\varepsilon 3} B - c_{\varepsilon 2}\varepsilon) \tag{16}$$

Where σ_k and σ_ε are the turbulent Schmidt numbers for k and ε, respectively. P_s represents the turbulent production due to shear, B is the buoyancy production/destruction term, and $c_{\varepsilon 1}$, $c_{\varepsilon 2}$, and $c_{\varepsilon 3}$ are empirical constants. The classical energy cascade model establishes a relationship between k, ε, and l, as expressed in Equation 17.

$$l = (c_\mu^0)^3 \frac{k^{\frac{3}{2}}}{\varepsilon} \tag{17}$$

Where c_μ^0 is an empirical constant. Once Equations 15 and 16 are numerically solved, the turbulence length scale can be derived from Equation 17, and the vertical eddy viscosity and diffusivity can be computed using Equations 13 and 14.

BOUNDARY CONDITIONS

Finally, the boundary conditions must be specified. By integrating the continuity equation over the water column, Equation 18 is derived.

$$\frac{\partial \hat{U}}{\partial x} + \frac{\partial \hat{V}}{\partial y} = w_B - w_0 \tag{18}$$

Where $\hat{U}$ and $\hat{V}$ are the barotropic velocity components, and w_B and w_0 are the vertical velocities at the bottom and surface, respectively. These velocities are determined by the kinematic boundary conditions, as shown in Equation 19.

$$w_0 = \frac{Dz}{Dt}|_\zeta + E - P \qquad\qquad w_B = 0 \tag{19}$$

Where E is the evaporation, and P is the precipitation. The free surface equation is then expressed as Equation 20.

$$\frac{\partial \zeta}{\partial t} + \frac{\partial \hat{U}}{\partial x} + \frac{\partial \hat{V}}{\partial y} = P - E \tag{20}$$

The wind stress, applied at the first layer interface, is treated following the MFS bulk formulae approach (Pettenuzzo et al., 2010), which is shown on Equations 21 and 22.

$$\tau_{xz}^{z_0} = A_V \frac{\partial u}{\partial z}|_{z(0)} = \frac{\rho_a}{\rho_0} C_D |u_w| u_w \tag{21}$$

$$\tau_{yz}^{z_0} = A_V \frac{\partial v}{\partial z}|_{z(0)} = \frac{\rho_a}{\rho_0} C_D |u_w| v_w \tag{22}$$

Here, ρ_a is the air density, u_w and v_w are the zonal and meridional wind velocity components at 10 m height, and C_D is the wind drag coefficient.

The bottom stress is determined using the quadratic formulations presented in Equations 23 and 24.

$$\tau_{xz}^{z_N} = \frac{C_B}{H_N^2} |\overrightarrow{U_N}| U_N \tag{23}$$

$$\tau_{yz}^{z_N} = \frac{C_B}{H_N^2} |\overrightarrow{U_N}| V_N \quad (24)$$

Where $\tau_{xz}^{z_N}$ and $\tau_{yz}^{z_N}$ are the turbulent shear stresses at the bottom interface of the deepest layer, H_N is bottom layer thickness, U_N and V_N the zonal and meridional transports of the bottom layer. C_B is the bottom drag coefficient defined by Equation 25.

$$C_B = \left(\frac{0.4}{\ln\left(\frac{\lambda_B + 0.5H_N}{\lambda_B}\right)} \right)^2 \quad (25)$$

Where λ_B is the bottom roughness length expressed in m.

The diffusive flux of temperature at the air-sea interface follows the Equation 26 from (Pettenuzzo et al., 2010).

$$K_V \frac{\partial \theta}{\partial z}|_{z(0)} = \theta_{l=1}(E - P) + \frac{Q_{net}}{\rho_0 C_P} \quad (26)$$

Here, $Q_{net} = Q_S - Q_L - Q_H - Q_E$ is the net downward heat flux, with the shortwave radiation flux Q_S, the longwave radiation flux Q_L, the sensible heat flux Q_H, and the latent heat flux Q_E. The C_P coefficient represents the specific heat of seawater.

The diffusive flux of salinity at the surface is expressed by Equation 27.

$$K_V \frac{\partial S}{\partial z}|_{z(0)} = S_{l=1}(E - P) \quad (27)$$

At the bottom for the tracers the adiabatic boundary conditions (no flux), based on Equations 28 and 29.

$$K_V \frac{\partial \theta}{\partial z}|_{z(N)} = 0 \quad (28)$$

$$K_V \frac{\partial S}{\partial z}|_{z(N)} = 0 \quad (29)$$

Further details regarding the model setup, which is primarily based on the work of Alessandri (2022), are provided in Section 2.1.2.

1.4 MACHINE LEARNING FOR STORM SURGE DOWNSCALING

Traditional hydrodynamic models have been widely used to represent meteorological and oceanographic variables, such as storm surge, but recent advances in Artificial Intelligence (AI) have introduced new possibilities for improving the accuracy and efficiency of weather predictions (Chen et al., 2022). AI data-driven systems refer to computational models that leverage large datasets to identify patterns, make predictions, and solve complex problems. These systems use algorithms, such as Machine Learning (ML) and Deep Learning (DL), to "learn" from data by recognizing underlying relationships between input and output variables. AI models are particularly useful in processing and analyzing vast amounts of information, allowing them to handle complex, nonlinear interactions that traditional methods might struggle to model (Fetzer, 1990).

In this work, the algorithms are referred to as "Machine Learning models" (ML models) because the focus is on learning patterns from data and evaluating their performance in the context of downscaling. Before proceeding with the description of ML algorithms, and to support a comprehensive understanding of this study, key concepts are summarized in Table 1.

Table 1: Key concepts in Artificial Intelligence algorithms.

Concept	Definition
Weights	Values that determine the importance of a connection between two neurons in a neural network. They are adjusted during training to help the network learn and improve its predictions or decisions.
Epoch	Refers to one complete pass through the entire training dataset. During an epoch, the model updates its weights based on the gradients calculated from the loss function. The number of epochs determines how many times the model will see the entire dataset during training.
Iteration	An iteration is a single update of the model's parameters, typically after processing a batch of training data. In each iteration, the optimizer updates the model's weights based on the gradients of the loss function. Relation to Epochs: The number of iterations per epoch depends on the batch size. If the batch size is smaller than the total dataset, there will be multiple iterations in a single epoch.
Loss function	A loss function (also called a cost function or objective function) quantifies the difference between the predicted outputs of the model and the actual target values. The goal of training is to minimize this loss, improving the model's accuracy.

	Common Loss Functions: • Mean Squared Error (MSE): Often used for regression tasks, it measures the average of the squares of the differences between predicted and actual values. • Cross-Entropy Loss: Commonly used in classification problems, this measures the difference between the predicted probability distribution and the actual distribution.
Optimization algorithm	Algorithm that looks for a set of model parameters that minimizes the loss function. Usually based on gradient descent, that adjusts the model's parameters by moving in the direction of the negative gradient of the loss function. This helps the model converge toward a minimum of the loss function, which ideally corresponds to the best possible model parameters. Variants: • Batch Gradient Descent: Uses the entire dataset to compute gradients, which can be computationally expensive but more accurate. • Stochastic Gradient Descent (SGD): Uses a single data point to compute the gradient, which introduces more noise but can speed up convergence. • Mini-batch Gradient Descent: Combines both approaches by using small batches of data to calculate gradients, balancing accuracy and efficiency. • Momentum: optimization algorithm that accelerates gradient descent by using a moving average of past gradients to smooth updates, helping to navigate faster through flat or noisy regions. • Root Mean Square Propagation (RMSProp): is an optimization algorithm that adjusts learning rates for each parameter by dividing the gradient by a moving average of its squared values, helping to stabilize and accelerate training. • Adaptive Moment Estimation (Adam): An advanced optimizer that combines the benefits of two other methods (momentum and RMSProp) to adjust learning rates for each parameter adaptively.
Learning rate	Hyperparameter that controls how much to change the model's parameters during each update. A higher learning rate makes larger adjustments, while a lower learning rate makes smaller, more precise updates.
Activation	In neural networks, an activation function determines whether a neuron

function	should be activated (i.e., produce an output). It introduces non-linearity into the model, enabling it to learn more complex patterns. Common Activation Functions: • Rectified Linear Unit (ReLU): Outputs the input directly if it's positive; otherwise, it outputs zero. • Sigmoid: Produces a value between 0 and 1, often used in binary classification tasks. • Hyperbolic tangent (tanh): Produces a value between -1 and 1, used in tasks requiring normalized outputs.
Forward propagation	Forward propagation is the process where input data moves through a neural network layer by layer, with each neuron calculating an output using weights, biases, and an activation function, to produce the final prediction.
Back propagation	Backpropagation efficiently computes the gradient of the loss function with respect to network weights for a single input-output pair. It iterates backward layer by layer, using the chain rule to avoid redundant calculations of intermediate terms.
Overfitting	Occurs when a model learns to perform very well on training data but fails to generalize to new, unseen data. This often happens when the model is too complex and memorizes the training data rather than learning underlying patterns.
Underfitting	Happens when the model is too simple to capture the underlying patterns in the data, resulting in poor performance on both the training and validation datasets.
Hidden layer	Layer of neurons in a neural network between the input and output layers, where computations are performed to learn patterns and representations from the data.
Hidden state	Internal memory of a recurrent neural network (RNN) that stores information about previous time steps to capture dependencies in sequential data.

In environmental modeling, ML data-driven models play a crucial role in phenomena such as storm surge downscaling and prediction (Zhao et al., 2024). To further model and predict, regression algorithms are often applied to establish relationships between the predictors or features (variables selected to represent the target) and predictand (target, for example storm surge responses). Regression algorithms are widely used for predictive modeling, providing a framework for quantifying the

influence of multiple variables on a target outcome, such as storm surge levels (Mi Zhang et al., 2019).

In addition, Neural Networks (NNs), a class of ML algorithms inspired by the structure and function of the human brain, are increasingly used for complex tasks like time series forecasting (Bezuglov et al., 2016). NNs can learn from historical data, capturing both short-term fluctuations and long-term trends (Suradhaniwar et al., 2021), making them valuable tools for predicting storm surge events that evolve over time. By integrating techniques such as regression algorithms and neural networks, ML-driven models can effectively downscale storm surge predictions, enhancing the accuracy of localized forecasts. This approach allows for more precise early warning systems and better coastal management strategies, ultimately reducing the risks associated with extreme weather events.

In this study different ML models were implemented under different configurations for storm surge time-series downscaling. To support the understanding of the configurations and models implemented, in Sections 1.4.1 to 1.4.6 important aspects and descriptions related to ML models are provided.

1.4.1 TRAINING PROCESS

Training is a fundamental step in machine learning, where a model learns to map input predictors to target the predictand by optimizing its internal parameters (e.g., weights and biases) based on a given dataset. The process typically involves the following key steps:

1. Forward Propagation: The model processes input data through its layers to compute predicted outputs. This step is governed by the mathematical structure of the model (e.g., linear equations for regression models, activation functions in neural networks).

2. Loss Computation: The discrepancy between predicted outputs and actual target values is quantified using a loss function. The choice of the loss function depends on the task (e.g., mean squared error for regression, cross-entropy for classification) and defines the optimization objective (Goodfellow et al., 2016).

3. Backward Propagation: The gradients of the loss function with respect to the model parameters are computed using the chain rule of calculus through a

process called "automatic differentiation". This step provides the direction to adjust the parameters to minimize the loss (Lecun et al., 2015).

4. Parameter Optimization: An optimization algorithm, such as stochastic gradient descent (SGD) or Adam, updates the model's parameters iteratively. The magnitude of these updates is controlled by the learning rate, which balances convergence speed and stability (Kingma & Ba, 2015).

The training process typically involves multiple iterations (epochs) through the dataset, where the model progressively improves its predictions by reducing the loss. To ensure robust generalization, techniques such as regularization, early stopping, or cross-validation may be applied (Bishop, 2006).

1.4.2 LOSS FUNCTION

In training ML models, the choice of loss function plays a crucial role in the optimization process, particularly in the context of gradient-based methods like gradient descent. Specifically, differentiable loss functions around zero, such as those with quadratic forms (e.g., Mean Squared Error, MSE), provide distinct advantages over non-differentiable options, particularly when approaching the minimum error region. Understanding the theoretical foundation behind these advantages clarifies why quadratic loss functions are often preferred in deep learning and other data-driven applications (Bishop, 2006; Goodfellow et al., 2016).

Gradient descent relies on iterative updates to model parameters based on the gradient, or slope, of the loss function. For this method to be effective, the loss function should ideally be smooth and differentiable, allowing gradients to be well-defined and continuous at all points. This differentiability is particularly essential around zero error, where the model's aim is to converge to minimal error values. A differentiable function in this region provides a smooth gradient that decreases steadily as the predictions increasingly align with the target values, enabling precise adjustments that guide the model toward stable convergence (Ruder, 2016).

Quadratic loss functions are continuous and differentiable across all error values, including zero. This property ensures that gradients decrease smoothly as errors decrease, providing a natural progression toward convergence. As the model approaches the minimum error region, the gradients reduce in magnitude, leading to finer adjustments that enable stable convergence to the optimal solution. In contrast, absolute loss functions (e.g., Mean Absolute Error) contain a non-smooth

point at zero due to the nature of the absolute value operation. Around zero the gradient presents a discontinuity, which can lead to erratic adjustments and introduce instability in the optimization process. Close to zero error, the gradient from an absolute loss function "jumps" rather than decreases smoothly, making it challenging for the optimizer to converge as accurately as it would with a differentiable function (Boyd & Vandenberghe, 2004; Bishop, 2006).

The continuous, smooth gradients provided by quadratic loss functions contribute not only to faster convergence but also to more stable and reliable parameter updates. As a result, models trained with quadratic loss functions frequently achieve better precision in the final stages of training. In contrast, absolute loss functions, with their inherent non-differentiability, can slow down convergence, causing the optimizer to oscillate or make less precise adjustments near the optimal error region (Nocedal & Wright, 2006).

From a theoretical perspective, the smoothness of quadratic loss functions results in a convex error surface that is straightforward for gradient-based optimizers to navigate. This convex, "bowl-shaped" landscape consistently directs the gradient toward the global minimum, minimizing the risk of becoming trapped in local minima and allowing the optimizer to reach the best solution efficiently. Conversely, the non-smooth transition in absolute loss functions creates an irregular optimization landscape, making it challenging to efficiently locate the global minimum, especially near zero error (Boyd & Vandenberghe, 2004; (Ruder, 2016).

1.4.3 CLASSIFICATION ALGORITHMS AND PRINCIPAL COMPONENT ANALYSIS

Classification algorithms are techniques used in ML to categorize data into predefined classes or categories. These algorithms function by learning patterns from labeled data during training, enabling the model to make predictions on unseen data. The choice of classification algorithm depends on the nature of the data, the number of classes, and the complexity of the relationships among features. Common algorithms include decision trees, support vector machines (SVM), k-nearest neighbors (k-NN), and neural networks, each with distinct mechanisms for learning and decision-making (Hastie et al., 2007).

To enhance the performance of classification algorithms, particularly in high-dimensional datasets, dimensionality reduction techniques such as Principal Component Analysis (PCA) are often employed. PCA is an unsupervised method that

transforms the original set of features into a smaller set of uncorrelated variables, known as Principal Components (PCs), which capture the maximum variance of the data in a reduced-dimensional space (Jolliffe, 2002).

PCA operates by identifying the principal directions (orthogonal axes) that account for the greatest variability in the dataset. This is accomplished by computing the eigenvectors and eigenvalues of the covariance matrix, allowing PCA to project the data onto a lower-dimensional space where the most informative features, or those with the highest variance, are retained. In geophysical and oceanographic applications, these eigenvectors are often referred to as Empirical Orthogonal Functions (EOFs), while the corresponding principal component time series are called Principal Components (PCs). EOFs represent spatial patterns of variability, whereas PCs describe how these patterns evolve over time. By transforming the original dataset into these components, PCA can eliminate noise and redundancy, making the dataset more manageable for classification algorithms while preserving the essential structure of the data (Bishop, 2006).

In the context of classification, PCA's role is especially beneficial when the dataset has a large number of features, which can complicate model training. Reducing the feature space can improve classifier efficiency, decrease computational costs, and potentially enhance generalization to unseen data. PCA also aids interpretability by emphasizing the most significant patterns in the data, which can be particularly helpful for analyzing complex datasets in fields like image recognition or text classification (Goodfellow et al., 2016).

However, it is important to note that PCA is an unsupervised technique, meaning that it does not consider class labels in its dimensionality reduction process. Consequently, PCA may discard features that are essential for distinguishing between classes, especially when those features do not align with the directions of greatest variance. For datasets where maximizing class separability is crucial, supervised methods such as Linear Discriminant Analysis (LDA) may sometimes be more effective. LDA, unlike PCA, optimizes for variance between classes rather than general variance, which can improve classification accuracy (Jollife & Cadima, 2016).

1.4.4 MULTIVARIATE LINEAR REGRESSION

Multivariate linear regression (MLR) is a statistical technique used to model the relationship between a dependent variable (predictand) and multiple independent

variables (predictors). In the context of time-series regression problems, MLR aims to predict the future values of a target variable based on several predictor variables (Uyanik & Guler, 2013). Mathematically, the model is expressed as:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \epsilon$$

Where Y is the predictand, $X_1, X_2, \ldots, X_n$ are the predictors, β_0 is the intercept, $\beta_1, \beta_2, \ldots, \beta_n$ are the coefficients corresponding to each predictor, and ϵ is the error term.

The goal of MLR is to model a linear relationship between predictors and the predictand. To achieve this, the coefficients (β) must be estimated by minimizing the sum of the squared differences between the observed and predicted values of the dependent variable. This is accomplished during the training process.

One key assumption of MLR is that there is a linear relationship between the dependent and independent variables. Additionally, it assumes that the residuals (differences between observed and predicted values) are normally distributed with constant variance and that there is no perfect multicollinearity (Su et al., 2012).

1.4.5 NEURAL NETWORKS

A Neural Network (NN) is a class of ML models inspired by the structure and function of biological neural networks. It consists of layers of interconnected nodes, or neurons, where each neuron performs a simple computational task. The network is typically organized into three main types of layers: an input layer, one or more hidden layers, and an output layer. Each layer is made up of several neurons, and each neuron is connected to others in adjacent layers by weighted connections. These weights are learned during the training process, which is typically done using backpropagation and gradient descent techniques (Bishop, 2006).

NNs are highly flexible and can approximate virtually any continuous function. This is due to their ability to learn non-linear relationships in the data, which distinguishes them from linear models such as linear regression or logistic regression. The learning process involves adjusting the weights through backpropagation, where the error is propagated backward through the network to update the weights and minimize the error. The error is typically computed using a loss function, such as Mean Squared Error (MSE) for regression tasks or Cross-Entropy for classification tasks (Goodfellow et al., 2016).

The neurons in a neural network apply an activation function to the weighted sum of their inputs to introduce non-linearity, which is essential for the network's ability to model complex patterns. Common activation functions include sigmoid, tanh, and ReLU. Each of these functions has different properties that make them suitable for different tasks. For example, ReLU is often preferred because it mitigates the vanishing gradient problem, especially in deep networks, where gradients can become very small and hinder learning (Glorot et al., 2011).

While neural networks are powerful and capable of learning complex patterns, they also come with challenges. The overfitting problem, where the model learns the noise in the training data, is a significant concern. Regularization techniques such as dropout, L2 regularization, and early stopping are used to prevent overfitting and improve generalization. Additionally, NNs require careful tuning of hyperparameters such as the learning rate, the number of hidden layers, and the number of neurons in each layer. The computational cost of training large networks, especially on big datasets, can also be a challenge (Bishop, 2006).

MULTI-LAYER PERCEPTRON

The Multilayer Perceptron (MLP) is a type of NN that has become a cornerstone of modern ML, particularly for tasks involving nonlinear data patterns. It consists of layers of interconnected neurons, each performing computations that allow the network to learn complex relationships in the data (Bishop, 2006).

At its core, the MLP performs its calculations by taking input data and passing it through the network in a feedforward manner. In this process, the input features are multiplied by corresponding weights and summed at each neuron. This sum is then passed through an activation function, which introduces non-linearity to the model. The choice of activation function (such as sigmoid, tanh, or ReLU) plays a critical role in the network's ability to model complex patterns (Goodfellow et al., 2016). The non-linearity introduced by the activation function allows the MLP to approximate any continuous function, a characteristic that distinguishes it from simpler models like linear regression or logistic regression.

An important feature of MLPs is their ability to handle nonlinear relationships in data. Unlike linear models, which can only capture linear decision boundaries, MLPs, with their multiple layers and nonlinear activation functions, can model highly complex and abstract relationships in data. This flexibility makes MLPs suitable for a wide range

of tasks, including function approximation, regression, and pattern recognition in both structured and unstructured data (e.g., images, time series, and text) (Bishop, 2006).

However, the design and training of MLPs come with several challenges. One of the primary concerns is the overfitting problem, particularly when the network is too large relative to the training data. Overfitting occurs when the model learns the noise or irrelevant details in the training data, leading to poor generalization on unseen data. To counter overfitting, regularization techniques such as dropout, L2 regularization (weight decay), and early stopping are commonly employed. These methods help prevent the network from becoming overly complex and encourage it to generalize better to new data (Srivastava et al., 2014).

Another challenge is the vanishing gradient problem, which is particularly significant in deep networks (networks with many hidden layers). When the network has many layers, gradients used for weight updates during backpropagation can become exceedingly small, effectively preventing the network from learning. This problem can be alleviated by using activation functions like ReLU instead of sigmoid or tanh, as ReLU has a constant gradient for positive inputs, making it more suitable for deep networks (Glorot et al., 2011).

MLPs also require careful tuning of their hyperparameters, such as the number of hidden layers, the number of neurons in each layer, the learning rate, and the batch size. The process of selecting the best hyperparameters is typically performed through cross-validation or grid search, and can be computationally expensive, especially for deep networks. Furthermore, training MLPs on large datasets can require significant computational resources, particularly in terms of memory and processing power.

Despite these challenges, MLPs have proven to be highly effective for a broad range of machine learning tasks, especially as computational power has increased, and deep learning frameworks have been optimized. The development of techniques like batch normalization, advanced optimization algorithms, and transfer learning has further enhanced the performance of MLPs, making them more efficient and robust for real-world applications (Goodfellow et al., 2016).

RECURRENT NEURAL NETWORKS

Recurrent Neural Networks (RNNs) are a specialized class of neural networks designed to handle sequential data. Unlike traditional feedforward neural networks, RNNs have connections that form cycles within the network, allowing information to persist from

previous time steps. This cyclic structure enables RNNs to maintain a memory of past inputs, which makes them particularly suitable for tasks involving time series data, natural language processing, and speech recognition (Rumelhart et al., 1986).

At each time step, an RNN takes an input, processes it through its hidden layers, and updates its hidden state based on both the new input and the previous hidden state. This structure allows the network to use the information from prior time steps to influence the current output. The output is then typically passed to the next layer or used to make predictions. The core computation in an RNN involves a hidden state h_t at each time step t, which summarizes past inputs up to that point. The hidden state is updated as Equation 30.

$$h_t = \sigma(W_h h_{t-1} + W_x x_t + b) \tag{30}$$

Where h_{t-1} is the hidden state from the previous time step, x_t is the input at the current time step t, W_h is the weight matrix for the previous hidden state, W_x is the weight matrix for the current input, and b is the bias term. The activation function σ, typically a non-linear function, controls the degree of influence each time step's input has on the network's output. The final output y_t at each time step is obtained by Equation 31.

$$y_t = \emptyset(W_y h_t + b_y) \tag{31}$$

Where $\emptyset$ is the output activation function, and W_y and b_y are the weight and bias of the final output.

While RNNs are powerful for sequential data, they suffer from several limitations. A primary issue is the vanishing gradient problem, where gradients of the error become exceedingly small as they are propagated backward through time. This makes it difficult for standard RNNs to learn long-term dependencies in sequences. To address this issue, Long Short-Term Memory (LSTM) networks were developed, which are a specialized type of RNN that are more effective at capturing long-term dependencies (Hochreiter & Schmidhuber, 1997).

LONG SHORT-TERM MEMORY NETWORK

Long Short-Term Memory (LSTM) networks are an advanced variant of RNNs specifically designed to address the vanishing gradient problem. LSTMs introduce a more complex architecture, which includes special units known as memory cells that regulate the flow of information through the network. These memory cells are equipped with gates that control when information should be updated, remembered, or forgotten. This gating mechanism allows LSTMs to maintain long-term dependencies in sequential data, making them effective for tasks such as machine translation, speech recognition, and time series forecasting.

The LSTM architecture consists of three main gates: the input gate, the forget gate, and the output gate. These gates regulate the memory cell's internal state, and the flow of information between them. The input gate controls how much new information should be added to the memory cell, the forget gate determines how much of the previous memory should be discarded, and the output gate controls how much of the memory cell's contents should influence the output (Yu et al., 2019).

Mathematically, the LSTM cell at time t is updated as is shown in Equations 32 to 37.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (32)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (33)$$

$$\tilde{C}_t = tanh(W_C[h_{t-1}, x_t] + b_C) \quad (34)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (35)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (36)$$

$$h_t = o_t * tanh(C_t) \quad (37)$$

Where f_t, i_t, and o_t represent the forget, input, and output gates, respectively, $\tilde{C}_t$ is the candidate memory, and C_t is the cell state at time t.

LSTMs have several advantages over standard RNNs. Their ability to retain information over long sequences allows them to outperform traditional RNNs on tasks that require modeling long-term dependencies, such as speech recognition and language

modeling. Moreover, LSTMs have been shown to be more resistant to the vanishing gradient problem, making them a preferred choice for many sequential learning tasks (Hochreiter & Schmidhuber, 1997). However, LSTMs are computationally more complex than simple RNNs, and require more memory and processing power, especially when dealing with large datasets.

Despite these challenges, LSTMs have become a popular choice in deep learning, especially for applications involving sequential data. They have been used to achieve state-of-the-art performance in a wide range of domains, including natural language processing, time series forecasting, and financial modeling (Chen et al., 2022).

1.4.6 MACHINE LEARNING MODELS APPLIED TO STORM SURGE

The application of ML algorithms for the representation of storm surge has been focused on three main prediction problems: peak-value forecasting, time-series forecasting, and spatio-temporal forecasting. However, the utilization of ML algorithms faces the challenge of data availability, quantity and quality, data required to carry out the training and testing processes. For this, studies mainly rely on either the proper representation of observational data, synthetic data (generated by numerical simulations) or a combination of both (Qin et al., 2023). The analysis of the spatial and temporal variability of storm surges from observations is a difficult task to accomplish since observations are not homogeneous in time, scarce in space, and moreover, their temporal coverage is limited (Cid et al., 2017). This corresponds to a limitation during the training process since ML algorithm requires homogeneous data on time. The use of synthetic data could compensate for the scarcity of data on space and offer homogenous training data, since the domains and time spans of numerical models can be adjusted based on user requirements.

In the early years of applying ML to storm surge prediction, linear regression models and artificial neural networks (ANNs) were commonly used. Lee (2006) demonstrated the potential of ANNs for storm surge prediction in Taiwan, indicating that, despite their ability to model complex relationships, traditional ANNs were still constrained by the amount of temporal data they could process, motivating the later shift toward more dynamic models.

The increasing availability of large datasets and the need for real-time predictions led to more sophisticated techniques. Bezuglov et al. (2016) used a multiple-output

feedforward ANN to predict storm surges in North Carolina, revealing that a three-layer network provided the optimal balance of accuracy and complexity. However, the model showed occasional underestimation of peak surges, indicating the need for more sophisticated models like recurrent networks to address limitations in capturing temporal dynamics. In the same period, Jia et al. (2016) proposed a surrogate modeling approach for storm surge prediction using a combination of Kriging and PCA applied to the Gulf of Mexico. Their method reduced dimensionality and computational costs while maintaining predictive accuracy.

Cid et al., (2017; 2018) employed Multivariate Linear Regression to reconstruct storm surge levels from reanalysis data for the 20th century. These models successfully predicted daily maximum surges, validated against tide gauge data, and provided valuable historical perspectives on storm surge behavior, essential for understanding long-term trends in Southeast Asia.

Costa et al. (2020) proposed a statistical downscaling model for predicting extreme storm surges at La Rochelle–La Pallice, located in France. Their model, based on weather types, uses sea level pressure, wind magnitude, and geopotential height from CFSR and CFSRv2 reanalysis as predictors. Their fully supervised classification approach, which applies the minimum daily sea level pressure and maximum SLP gradient, yields better performance in predicting daily maximum storm surges compared to other statistical models, including multi-linear regression. Notably, the optimal configuration can identify unusual atmospheric patterns, such as the Xynthia storm, and reduces the maximum error by 50%, enhancing prediction accuracy for extreme storm surges.

As machine learning approaches advanced, more complex models such as Support Vector Regression (SVR), Gaussian Process Regression (GPR), and Tree-based algorithms gained prominence. Al Kajbaf & Bensi (2020) investigated surrogate models for estimating peak storm surges along the East Coast of the United States. Their study compared various algorithms, including ANN, GPR, and SVR, and highlighted the importance of incorporating physically motivated parameter scaling to improve model performance, especially under varying storm surge conditions. Bruneau et al. (2020) proposed a neural network approach to estimate coastal sea level extremes at over 600 tide gauges worldwide, capturing non-linearities in tide-surge interactions and providing probabilistic forecasts of sea level changes. This method has the potential to improve local resilience when combined with more expensive dynamic models. This period also saw the application of Ridge Regression and other ML models, like Support Vector Regressor and Tree-based models, for storm

surge prediction in the New York metropolitan area by Ayyad et al. (2022). Despite working with an imbalanced dataset, their models accurately predicted peak storm surges and return period curves that aligned well with results from high-fidelity hydrodynamic simulations (ADCIRC+SWAN, with 80 m horizontal resolution near the coastline), offering a more computationally efficient alternative to traditional simulation methods.

Ramos-Valle et al. (2021) implemented an artificial neural network (ANN) for storm surge forecasting in the Mid-Atlantic Bight region, utilizing synthetic data generated by coupling the Hybrid Weather Research and Forecasting cyclone model (HWCM, Bruyère et al., 2019) with ADCIRC. The ANN was tested for optimal lead-time configuration and neural network architecture, with the findings demonstrating the ANN's capability in forecasting moderate storm surge levels, while highlighting its limitations in predicting peak surge magnitudes. The addition of a recurrent neural network (RNN) improved the prediction of peak values, offering better performance in capturing the dynamics of storm surges.

In recent years, deep learning techniques, particularly Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks and hybrid models, have become central to storm surge prediction. Igarashi & Tajima (2021) explored the use of recurrent neural networks (RNNs) for predicting time-varying storm surge heights, using synthetic data from 151 typhoons in Japan. Their results demonstrated that RNNs outperformed deep neural networks (DNNs) in predictive accuracy, suggesting that including historical storm surge data can improve the forecasting performance. This insight encouraged the inclusion of temporal dynamics in the models, leading to further developments in hybrid and advanced machine learning techniques. Chen et al. (2022) applied an LSTM network to predict storm surge water levels in the East China Sea using meteorological and tidal data. Their findings demonstrated that LSTM outperformed Bayesian Ridge Regression (BRR), Gradient Boosted Decision Tree (GBDT), Linear Regression (LR), and SVR, providing accurate one-hour predictions and up to 15-hour forecasts with minimal error.

Lockwood et al. (2022) examined the potential of ANNs in predicting hurricane-induced storm surges along the U.S. East and Gulf Coasts, using synthetic hurricanes derived from the National Oceanic and Atmospheric Administration-National Center for Environmental Prediction (NOAA-NCEP) reanalysis data. The study found that ANNs could predict storm surge levels with high accuracy, showcasing the model's efficiency compared to traditional approaches, and aligning with the results of high-fidelity hydrodynamic simulations. This efficiency highlights the potential for machine

learning models to serve as alternatives to more computationally expensive methods in real-time forecasting.

Different studies have also introduced hybrid models that integrate Convolutional Neural Networks (CNNs) with LSTM networks to capture both spatial and temporal patterns in storm surge data. Wang et al. (2021) applied CNNs and LSTMs for multi-step ahead short-term predictions of storm surge levels in China. The study showcased the effectiveness of combining these two models for accurate predictions, contributing to better preparedness and response strategies for coastal communities at risk of storm surges. Tiggeloven et al. (2021) explored deep learning models for predicting surge components in sea-level variability, using ANNs, CNNs, LSTMs, and ConvLSTMs (Convolutional LSTM) at 736 tide stations globally. Their study demonstrated that LSTMs generally outperformed other neural network models, and adding more predictor variables improved performance, though at the cost of increased computation time. This study emphasizes the potential of deep learning models in improving storm surge predictions, although some regions, particularly the tropics, require more complex models to fully capture intra-annual variability.

Adeli et al. (2023) developed an advanced spatio-temporal ConvLSTM for storm surge prediction in Coastal Texas. This model combines the strengths of LSTM and CNN to process both spatial and temporal correlations in storm surge data, offering robust forecasts for new storm tracks based on synthetic storm simulations. Similarly, Dang et al. (2024) applied CNNs for storm surge reconstruction, utilizing data from 160 tide gauges in the Western North Pacific. Their model enabled the identification of regions with significant storm surge level changes and facilitated extreme value analysis, contributing to a better understanding of past storm surge dynamics.

More recently, Giaremis et al. (2024) focused on correcting systemic errors in storm surge forecasting models using LSTM-based deep learning. Their approach aimed to improve forecast accuracy by bias-correcting simulation results from hurricane events, thus enhancing the reliability of real-time storm surge forecasts.

Other ML models have also been recently applied to the storm surge representation. Sun & Pan (2023) developed a novel ensemble model (NEM) that integrates three machine learning algorithms (Random Forest, Gradient Boosting Decision Tree, and XGBoost) for storm surge prediction in Hong Kong. This model, which used a stacking technique for combining the algorithms, demonstrated superior performance over individual models with a coefficient of determination (R^2) of up to 0.95 and low mean absolute error (MAE) values. The study also performed an interpretability analysis

using the SHapley Additive exPlanations (SHAP) method, highlighting gale distance and nearest wind speed as critical features for predicting peak storm surge levels.

Tausía et al. (2023) introduced a methodology to reconstruct storm surge maximum levels across New Zealand using high-resolution regional hydrodynamic hindcast data. This approach employed various statistical models, including Multivariate Linear Regression, k-NN regression, and Gradient Boosting Regression, alongside atmospheric predictors. The results of this study show the Multivariate Linear Regression as the best performing method. Additionally, the findings showed that carefully selecting the atmospheric predictors and statistical models could significantly improve storm surge predictions, with the best model providing a Pearson correlation coefficient of 0.88 and an RMSE of 4.3 cm.

Harter et al. (2024) explored the limitations of statistical models in storm surge reconstruction, particularly their underestimation of extreme surge events. The authors tested Multivariate Linear Regression and NN models at 14 tide gauges across the North-East Atlantic. Their findings suggest that using wind stress rather than wind speed as a predictor reduces bias in extreme surge predictions. Additionally, including significant wave height as a predictor can help reduce errors at some locations. Their analysis also highlights the shortcomings of atmospheric reanalyses in accurately representing historical extreme events, with neural networks showing promise in predicting extreme surges even without wind information, providing new insights into air-sea interactions.

In the Atlantic hurricane region, Feng & Xu (2024) proposed a multi-recursive neural network based on Gated Recurrent Units (GRUs). This model incorporated a distortion loss function to improve accuracy, particularly for longer lead times, by capturing critical physical factors like wind speed and atmospheric pressure. This study represents the growing trend of incorporating physically motivated losses into machine learning models, which could significantly enhance their reliability in storm surge prediction.

APPLICATIONS IN THE ADRIATIC SEA

The application of ML models in the Adriatic Sea region is still relatively limited. Bajo & Umgiesser (2010) developed an operational surge forecast system that combines hydrodynamic simulations with ANNs for the Venice region. The system utilizes a five-day forecast from a Mediterranean Sea hydrodynamic model based on SHYFEM

(highest horizontal resolution of 1.5 km near the coast of the north Adriatic Sea), with the results near Venice being enhanced by the neural network. This integration reduces the average error of the hydrodynamic model by half for the first day forecast and maintains strong performance for longer forecast periods. Furthermore, the neural network approach is successful in reducing the model's bias error, improving the accuracy of storm surge predictions.

Žust et al. (2021) introduced HIDRA 1.0 (HIgh-performance Deep tidal Residual estimation method using Atmospheric data), a deep-learning-based ensemble sea level forecasting method for the Northern Adriatic Sea (Koper, Slovenia). HIDRA outperformed the ocean circulation model ensemble NEMO v3.6 (~1.5 km horizontal resolution) across all forecast lead times while requiring a fraction of the computational cost (order of $2x10^{-6}$). The model integrates atmospheric and sea level features, optimizing predictions by treating sea level as a combination of tidal and residual components. HIDRA demonstrated lower RMSE, especially for short-term forecasts (<24 h), and effectively captured semi-diurnal and basin seiche frequencies, which NEMO underestimated. Trained on a 10-year dataset and tested on a 1-year dataset, HIDRA successfully replicated the timing and amplitude of basin seiches during storm surges, showcasing its potential for accurate, efficient sea level forecasting.

Rus et al. (2023a) presented HIDRA2, a deep-learning ensemble model designed for sea level and storm tide forecasting, developed also for Koper (Slovenia). This model outperforms both, its predecessor (HIDRA1) and two advanced numerical ocean models, NEMOv3.6 and SCHISM (barotropic model with 200 m horizontal resolution at the coastline), in terms of forecast accuracy. The architecture of HIDRA2 incorporates novel atmospheric, tidal, and sea surface height feature encoders and a feature fusion and regression block. HIDRA2's performance is particularly notable in storm events, where it reduces forecast errors by 25%. Additionally, it accurately predicts the amplitudes and temporal phases of Adriatic basin seiches, which is critical due to their influence on storm tide levels and coastal flood forecasting.

Rus et al. (2023b) further developed HIDRA-T, a transformer-based deep learning architecture for sea level and storm tide forecasting. This model improves upon HIDRA2 by employing transformer-based encoders for atmospheric and sea level data, along with a feature fusion and regression block. HIDRA-T outperforms all previous models, including HIDRA2 and HIDRA1, in predicting extreme sea level events. The model excels in the extreme tail of sea level distribution, providing more accurate predictions for high-impact storm surges. The improved accuracy of HIDRA-

T is particularly useful for forecasting coastal flooding, showcasing its potential to address the limitations of traditional numerical ocean models.

To synthesize the key findings from the reviewed literature, Table 2 presents a summary of the different case studies, outlining their methodological approaches, locations, and identified pros and cons. This allows for a clearer comparison between traditional and advanced ML-based forecasting systems and emphasizes recent progress made in the Adriatic Sea region.

Table 2: Overview of ML applications for sea level and storm surge forecasting and downscaling.

Reference	Location	Approach	Pros	Cons
Lee (2006)	Taiwan	Artificial Neural Networks (ANNs)	Demonstrated the potential of ANNs for storm surge prediction	Limited by data availability and temporal data processing capacity
Bezuglov et al. (2016)	North Carolina	Feedforward ANN	Optimal balance of accuracy and complexity	Occasional underestimation of peak surges
Jia et al. (2016)	Gulf of Mexico	Surrogate modeling (Kriging + PCA)	Reduced dimensionality and computational costs	May not capture complex dynamics as well as other models
Cid et al. (2017; 2018)	Southeast Asia	Multivariate Linear Regression	Successfully predicted daily maximum surges using historical data	Limited scope for capturing dynamic, real-time surge variations
Costa et al. (2020)	La Rochelle–La Pallice, France	Statistical downscaling model	Better performance than other statistical models; identifies unusual atmospheric patterns	Dependent on weather types and specific reanalysis data
Al Kajbaf & Bensi (2020)	East Coast, USA	GPR, SVR, ANN	Accurate under varying storm surge conditions	Limited by imbalanced datasets

Bruneau et al. (2020)	Global	Neural Networks for sea level extremes	High accuracy in capturing non-linearities	Computationally intensive
Ayyad et al. (2022)	New York	Ridge Regression, SVR, Tree-based models	Computationally efficient alternative to high-fidelity simulations	May not handle complex or imbalanced data well
Ramos-Valle et al. (2021)	Mid-Atlantic Bight	ANN with synthetic data	Optimized for moderate surge levels, good for real-time forecasting	Less accurate for peak surge predictions
Igarashi & Tajima (2021)	Japan	Recurrent Neural Networks (RNNs)	RNN outperforms DNNs in predictive accuracy	Requires historical data, which may not be available for all regions
Chen et al. (2022)	East China Sea	LSTM	Accurate one-hour predictions, performs well with minimal error	May be sensitive to data quality and quantity
Lockwood et al. (2022)	U.S. East and Gulf Coasts	ANN	High efficiency compared to traditional methods	Efficiency may come at the cost of capturing detailed dynamics
Wang et al. (2021)	China	CNN + LSTM Hybrid	Captures both spatial and temporal patterns	Increased computation time with more variables
Tiggeloven et al. (2021)	Global	LSTM, ConvLSTM	LSTM outperforms other models for sea-level surge prediction	Requires more computing power for more predictors
Adeli et al. (2023)	Coastal Texas	ConvLSTM	Robust for forecasting new storm tracks	Relies on synthetic data, which may not always reflect real-world dynamics
Dang et al.	Western North	CNN	Identifies regions	May not fully

(2024)	Pacific		with significant surge changes	account for extreme surge events in all regions
Giaremis et al. (2024)	Various	LSTM-based deep learning	Corrects systemic errors in surge forecasts	Requires accurate simulation data for bias correction
Sun & Pan (2023)	Hong Kong	Ensemble Model (Random Forest, GBDT, XGBoost)	High R2 and low MAE, interpretable results	May require significant computational resources for ensemble models
Tausía et al. (2023)	New Zealand	Multivariate Linear Regression, k-NN, Gradient Boosting	Best performing method with high Pearson correlation	Limited by atmospheric predictors' quality and selection
Harter et al. (2024)	North-East Atlantic	NN and Multivariate Linear Regression	Reduces bias by using wind stress and adding significant wave height	May not fully account for extreme surge events in all locations
Feng & Xu (2024)	Atlantic Hurricane Region	Gated Recurrent Units (GRUs)	Improves accuracy with physically motivated loss functions	May require fine-tuning for optimal lead times
Bajo & Umgiesser (2010)	Venice, Northern Adriatic Sea	Artificial Neural Networks (ANNs)	Reduces model error by half; improves storm surge forecast accuracy	Limited by hydrodynamic model limitations and forecast lead times
Žust et al. (2021)	Koper, Northern Adriatic Sea	Deep-learning-based ensemble (HIDRA 1.0)	Outperformed NEMO model with lower RMSE and computational cost	May not capture long-term trends as accurately as ocean models
Rus et al. (2023a)	Koper, Northern Adriatic Sea	Deep-learning ensemble (HIDRA2)	25% reduction in forecast errors for storm events; accurately	May require continuous data updates for improved

			predicts Adriatic basin seiches	prediction accuracy
Rus et al. (2023b)	Koper, Northern Adriatic Sea	Transformer-based deep learning (HIDRA-T)	Improved forecast accuracy, especially for extreme sea level events; better storm surge predictions	Transformer-based models may require more computational power for large datasets

DISCUSSION ON GAPS IN CURRENT RESEARCH

While the reviewed studies demonstrate significant advancements in ML techniques for storm surge downscaling and prediction, several gaps remain in the literature that hinder the full potential of these models in accurately forecasting extreme storm surge events, which are summarized in the following points:

- Lack of systematic comparison with high-resolution dynamic models: One of the key limitations in the existing body of work is the absence of a systematic comparison between ML models and state-of-the-art high-resolution dynamic models. While machine learning models, such as ANNs, LSTMs, and CNNs, have shown promising results in predicting storm surge events, many studies rely on surrogate models or simplified statistical techniques. Few studies have directly compared these ML models with high-fidelity hydrodynamic simulations that incorporate complex physical processes (e.g., Ayyad et al., 2022; Žust et al., 2021; Rus et al., 2023a), such as atmospheric conditions, ocean dynamics, and wave interactions. However, none of them developed a high-resolution dynamic model specifically to carry out this comparison. This comparison is crucial as high-resolution dynamic models are the standard baseline for storm surge predictions and can provide more reliable and physically consistent forecasts.

- Acceptance of linear approaches: Another significant gap is the persistence of linear approaches in storm surge prediction, with several studies claiming that linear models, such as Multivariate Linear Regression, are sufficient for capturing storm surge dynamics (e.g., Cid et al., 2017; Tausía et al., 2023; Harter et al., 2024). Despite the growing complexity of storm surge events, especially in the face of climate change, linear models fail to account for the non-linear and highly dynamic nature of storm surge phenomena. Machine

learning techniques, such as NN models, have demonstrated their ability to handle non-linearity, yet many studies continue to use linear models without a thorough exploration of more advanced algorithms. A systematic comparison of linear approaches against non-linear models, including hybrid deep learning architectures, is necessary to determine the true limitations of linear models and to identify the most effective techniques for storm surge prediction.

- Insufficient focus on extreme event validation: The validation of ML models for storm surge prediction has predominantly focused on general forecasting accuracy, but there is a notable lack of attention to the performance of these models in predicting extreme events, which are of particular importance for coastal resilience and disaster management. Many studies assess model performance using mean error metrics or overall predictive accuracy but fail to specifically analyze how well these models predict extreme storm surges, which are critical for understanding and mitigating the impacts of severe weather events.

Addressing these critical gaps in storm surge forecasting research is essential for advancing the reliability and applicability of predictive models in real-world scenarios. By systematically comparing ML techniques with high-resolution dynamic models developed ad-hoc for the purpose, and prioritizing the validation of extreme event predictions, researchers can unlock the full potential of ML in capturing the complexities of storm surge phenomena. Such efforts will not only enhance the scientific understanding of storm surge dynamics but also support the development of robust, actionable tools for coastal resilience and disaster management in the face of a changing climate.

1.5 THESIS OBJECTIVES

Building on the gaps identified in the literature, this thesis aims to address key limitations in storm surge downscaling, particularly their performance in predicting extreme events and their integration with dynamic and machine learning approaches.

The main objective of this research is defined as: **Assessment of machine learning downscaling techniques for storm surge prediction in the Northern Adriatic Sea, emphasizing their performance relative to state-of-the-art dynamical models and observational data, with a particular focus on extreme events**.

The specific objectives to reach the main objective are the following:

1. Development of a state-of-the-art dynamical downscaling for accurate storm surge reproduction: This objective focused on the development of a robust storm surge numerical model capable of accurately simulating storm surge events under various atmospheric conditions, with an emphasis on extreme values. For this task, high-resolution simulations were conducted with the SHYFEM-MPI model, using different configurations and forced by various atmospheric databases, to achieve the best possible representation of extreme values.

The rationale for developing a new hindcast, rather than relying on existing products, is supported by the following considerations:

- Limitations in existing datasets: Available storm surge datasets for the Northern Adriatic Sea often lack the spatial resolution and localized calibration required to simulate storm surge dynamics accurately in shallow, semi-enclosed basins. These limitations are particularly evident in the underestimation of extreme storm surge, which is a key focus of this study.
- Need for a high-quality benchmark for ML comparison: To fairly evaluate the performance of the ML models in this study, it was essential to use a benchmark that offered both physical realism and high fidelity in reproducing observed data. The developed dynamical downscaling ensured internal consistency, spatial coverage, and accuracy in extremes, all of which were necessary for robust ML training, validation, and comparison.

2. Evaluation of dynamical downscaling performance with emphasis on extreme events: This research aims to thoroughly assess the skill of the developed dynamical downscaling, with particular attention to its ability to predict extreme storm surge

events. The model's performance will be assessed against nearshore station data using a variety of statistical metrics. While traditional error indicators like RMSE and Pearson correlation will be considered, new skill indicators will be proposed to better assess the model's ability to reproduce the distribution of observed storm surge data, particularly focusing on extreme events. This objective aims to identify the most accurate model setup for storm surge downscaling and establish a reliable framework for Machine Learning data-driven models.

3. Comparison of ML approaches with varying degree of complexity versus dynamical models: The third specific objective of this research focuses on implementing and evaluating various ML models for storm surge downscaling, with an emphasis on assessing the trade-offs between model complexity, general accuracy, and precision in extreme percentiles. The tasks include:

3.1. Implementation of ML models: Develop and train multiple ML models, including Multivariate Linear Regression, Multilayer Perceptron, Recurrent Neural Networks, and Long Short-Term Memory networks, using storm surge data from the best configuration of dynamical downscaling identified in Objective 1.

3.2. Customization of the loss function: Investigate the impact of a customized loss function on the performance of the ML models, particularly in capturing extreme storm surge events.

3.3. Performance evaluation: Evaluate the models' performance across multiple locations, comparing accuracy metrics, while also assessing the ability of each model to reproduce extreme surge events. Examine the trade-offs in model complexity and computational overhead for different ML model configurations.

3.4. Comparison with dynamical downscaling: Compare the ML models' predictions with the best dynamical downscaling model identified in Objective 1, using observed storm surge data.

3.5. Direct training with observational data: Train and test the ML models directly with observed storm surge data, evaluating their ability to capture temporal patterns and extreme surge events, and compare the results with the best dynamical downscaling model identified in Objective 1.

These tasks will contribute to understanding the potential of ML models as efficient alternatives to traditional dynamical models for storm surge prediction, with a focus on accuracy, computational efficiency, and extreme event representation.

The remainder of the document is structured as follows. Section 2 describes the materials (data) used and the methods followed for the study, including the details of the implementation of the dynamical and ML models. Section 3 presents the results of the performance evaluation of the dynamical and ML models, with a focus on the compliance with the specific objectives previously described. Section 4 includes a discussion of the implications of the findings and identifies avenues for further research to advance the efficiency and accuracy of storm surge prediction using ML algorithms. Finally, Section 5 presents the main conclusions of the study.

2 MATERIALS AND METHODS

This section presents the data and methods used for implementing the dynamic and ML models. Section 2.1 focuses on the methods employed to conduct the numerical simulations with SHYFEM-MPI, including the various forcings, model setup, and considerations for performance evaluation.

Section 2.2 outlines the procedures followed for implementing the ML models. It describes the predictand and predictors, the different configurations considered, and the methods used for the training and testing processes.

2.1 DYNAMICAL DOWNSCALING

2.1.1 ATMOSPHERIC FORCING

In this study, two distinct atmospheric databases were utilized to force the circulation model, incorporating mean sea level pressure and wind fields. The first database is ERA5, the fifth generation of reanalysis data generated by the European Centre for Medium-Range Weather Forecasts (ECMWF). ERA5 builds upon the Integrated Forecasting System (IFS) Cy41r2, which became operational in 2016, providing hourly output with a horizontal resolution of 0.25°x0.25° for atmospheric variables (Hersbach et al., 2020). ERA5 is relatively high resolution and accurate for a global reanalysis, although it is known to be affected by negative biases at high percentiles, particularly when compared with measured wind speed (Pineau-Guillou et al., 2018; Vannucchi et al., 2021; Benetazzo et al., 2022; Gumuscu et al., 2023).

Since ERA5 is relatively coarse for local studies and exhibits significant underestimation of extremes, we employed an alternative approach using a high-resolution (3.3 km) atmospheric downscaling developed by the University of Genoa (UniGe). Wind forcing was derived from 10 m wind fields via the Weather Research and Forecast (WRF-ARW) model v3.8.1, allowing for improved representation of small-scale forcings and physics. The computational domains comprised a 10 km resolution grid covering the Mediterranean, Northern Africa, and Southern Europe (A10), and a 3.3 km grid over the Tyrrhenian Basin and Northern Adriatic basin (A3), nested within A10. Initial conditions were obtained from the Climate Forecast System Reanalysis (CFSR) data, known for reliability but occasionally underestimating extreme events (Saha et al., 2010). WRF simulations were conducted for 24 hours with hourly outputs, employing established physical parameterization schemes to ensure accuracy across

various atmospheric conditions. For further details, readers are referred to Mentaschi et al., 2015).

2.1.2 MODEL SETUP

The unstructured grid for the simulations in this study was generated using the OceanMesh2D tool (K. J. Roberts et al., 2019) with a horizontal resolution of 3 km on the open ocean boundary and 50 m in the coastline (Figure 5b). The General Bathymetric Chart of the Oceans (GEBCO) dataset (Weatherall et al., 2015) was used, incorporating a high-resolution coastline from the European Environmental Agency. However, due to identified overestimations in water depth in the Venice and Marano lagoons from GEBCO bathymetry, adjustments were made based on the contributions from Fagherazzi et al. (2007), Lovato et al. (2010), and Zaggia et al. (2017) for the Venice lagoon and Petti et al. (2019) and Bosa et al. (2021) the for the Marano lagoon.

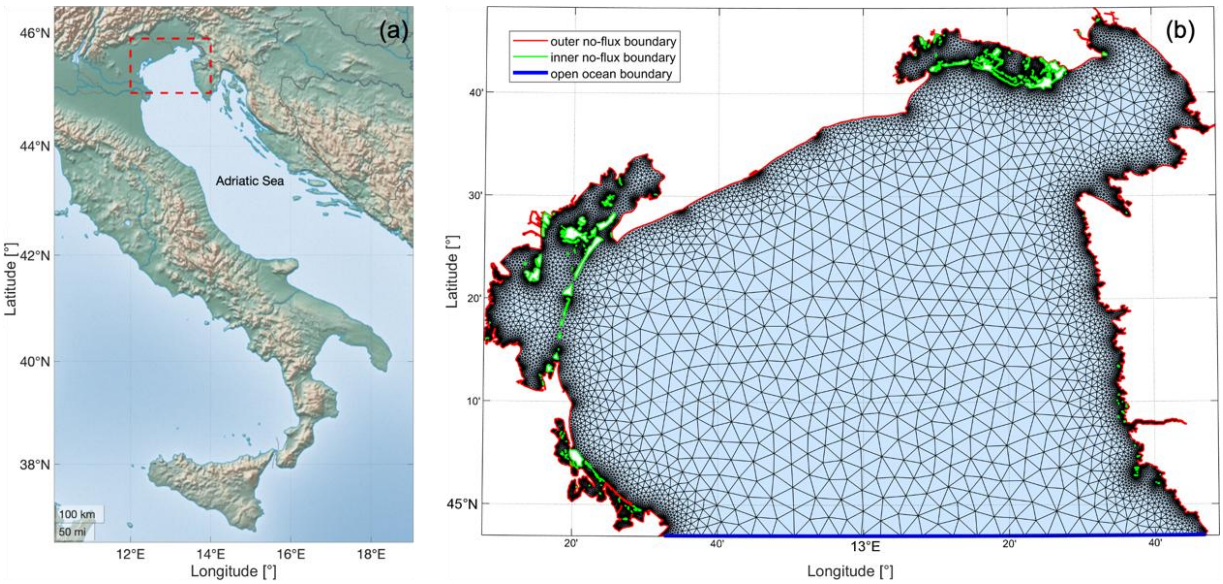

Figure 5: (a) Location of study area, marked with dashed red line; (b) Unstructured grid for study area, in which the blue line represents the location of the open boundary condition, the red line the coastline, and the green lines the coastline formed by islands.

As initial and open ocean boundary conditions, sea surface height, current velocity, temperature, and salinity from the Copernicus Mediterranean Sea Physics reanalysis (Med-MFC) (Escudier et al., 2021) were considered. Tides with hourly resolution from the Finite Element Solution (FES) 2014 (Lyard et al., 2021) were also included to account for the total sea level in the simulations. Specifically, the constituents included for the tide reconstruction are SA, SSA, O1, P1, S1, K1, N2, M2, MKS2, S2, R2, K2, M3, M4, and MS4, which were selected based on preliminary harmonic analysis applied to sea level observation data in the locations specified in Section 2.1.3, using the T-Tide MATLAB package (Pawlowicz et al., 2022). At the closed boundaries, a full-slip condition is applied, as in this study, along the coastline. The velocity component normal to the boundary is set to zero, while the tangential velocity remains a free parameter. At the open boundary, Dirichlet boundary conditions are applied when the flux enters the domain; otherwise, a zero-gradient condition (Neumann boundary condition) is imposed.

The dynamical downscaling period extends from 1987 to 2020, with hourly output. The simulations consider different setups to explore the influence of different atmospheric forcings and model configurations on the model's skill. Two model configurations were considered: (a) barotropic (BT), where the fluid's density depends solely on pressure, resulting in uniform density surfaces and no vertical variations in the horizontal pressure gradient; and (b) baroclinic (BC), where the fluid's density is influenced by both pressure and temperature/salinity, leading to sloping density surfaces and vertical variations in the horizontal pressure gradient. For the baroclinic configuration, 33 vertical levels were employed, with a layer thickness of 1 m up to a depth of 10 m, increasing to 2 m layers up to a maximum depth of 60 m. Then, three combinations of atmospheric forcing and configuration are considered here: 1) barotropic forced by ERA5 (BT-ERA5), 2) baroclinic forced by ERA5 (BC-ERA5), and 3) baroclinic forced by UniGe (BC-UniGe). To determine vertical viscosities and diffusivities, a k-ε turbulence scheme derived from the General Ocean Turbulence Model (GOTM) model (Burchard & Petersen, 1999) was utilized. For wind stress at the air-sea interface a constant wind drag coefficient of $2.5 * 10^{-3}$ was employed, following the works from Orlić et al. (1994) and Zampato et al. (2006). The bottom stress is determined through the quadratic formulation exposed on Equation 25, with a constant bottom roughness of 0.01 m. Table 3 summarizes the key characteristics of the various model setups implemented for dynamical downscaling.

Table 3: Key characteristics of the model setups implemented in SHYFEM for dynamical downscaling.

Acronym	Model type	Atmospheric forcing	Open ocean boundary conditions	Key parameterizations
BT-ERA5	Barotropic	Mean sea level pressure, and zonal and meridional wind fields, from the ERA5 database.	-Sea surface height from Med-MFC. -Tides from FES2014.	-Constant wind drag coefficient of $\mathbf{2.5 * 10^{-3}}$. -Constant bottom roughness of 0.01 m. -Turbulence closure model: k-ε turbulence scheme derived from GOTM model.
BC-ERA5	Baroclinic (33 vertical levels)	Mean sea level pressure, and zonal and meridional wind fields, from the ERA5 database.	-Sea surface height from Med-MFC. -Tides from FES2014. -Current velocity, temperature, and salinity from Med-MFC.	
BC-UniGe		Mean sea level pressure, and zonal and meridional wind fields, from the UniGe database.		

2.1.3 MODEL PERFORMANCE EVALUATION

The model output was compared with observations from tide gauges located in the Northern Adriatic Sea. The observational data were acquired from the Italian National Institute for Environmental Protection and Research (ISPRA), the Civil Protection of the Friuli-Venezia Giulia Region, and Raicich (2023). Table 4 summarizes the locations considered, and the available time spans for comparison that match with the simulation timespan. Figure 6 shows the locations considered for comparison between measured and simulated storm surge, together with the bathymetry used for the simulations.

Table 4: Locations considered for validation, including available start and end dates matching the simulation timespan.

Location	Lon [°]	Lat [°]	Start date	End date
ISMAR-CNR research platform "Aqua Alta" (hereafter CNR platform)	12.53	45.31	01-01-1987	31-12-2020
Punta della Salute	12.33	45.43	01-01-1987	31-12-2020
Caorle	12.86	45.59	01-01-2000	31-12-2020
Grado	13.38	45.68	01-01-1991	31-12-2020
Monfalcone	13.54	45.78	01-01-2008	31-12-2020
Trieste	13.76	45.64	01-01-1987	31-12-2020

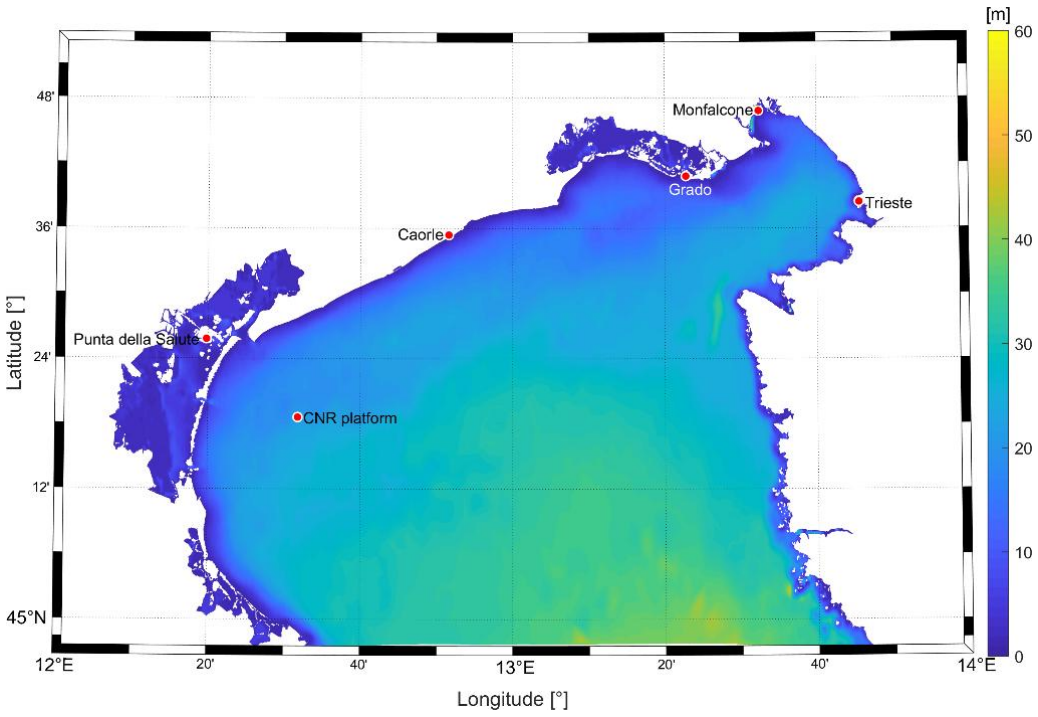

Figure 6: Tide gauges locations and bathymetry (depth values on positive).

Both the model output and the observations were processed as follows to enable their intercomparability. To start, both measurement and simulation were centered with a zero mean and then detrended. This approach mitigates possible effects of unmodulated land motion (Chepurin et al., 2014) and ensures that extreme values across the years can be considered as homogeneous and can be compared despite relative sea level changes (Ferrarin et al., 2022). Harmonic analysis was performed for each calendar year on the detrended sea levels using the T-Tide MATLAB package (Pawlowicz et al., 2022), and the non-tidal residual was obtained as the arithmetic difference between sea level and tides (Tiggeloven et al., 2021). Performing yearly harmonic analysis reduces timing errors that could cause tidal energy to seep into the non-tidal residual (Merrifield et al., 2013).

Finally, to obtain the pure storm surge (hereafter also called "surge"), a low-pass filter is applied to the non-tidal residual, following the work from Park et al. (2022). In this study, we consider a cut-off period of 13 hours for the filter based on the mixed-semidiurnal tidal regime around the Northern Adriatic Sea (Lionello et al., 2021).

The performance evaluation of the simulations relies on the computation of statistical metrics of hourly data, which encompass the entire dataset, as well as values exceeding the 99th percentile from the cumulative distribution of measured data at each location. The following metrics are considered:

Pearson correlation (Corr):

$$\rho = \frac{1}{N-1}\sum_{i=1}^{N}\left(\frac{S_i - \mu_S}{\sigma_S}\right)\left(\frac{O_i - \mu_O}{\sigma_O}\right) \quad (38)$$

Where S_i and O_i are the ith simulated and observed data respectively, N is the sample size, μ and σ are the mean and standard deviations of S and O. A value closer to one identifies a better performance.

Root-Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(S_i - O_i)^2} \quad (39)$$

A value closer to zero indicates a better performance. During the performance evaluation the RMSE double-penalty effect was identified, affecting the performance evaluation of the most sophisticated models. This phenomenon arises because RMSE penalizes errors based on both the magnitude and location of deviations. If a prediction is accurate in magnitude but misaligned in time or space (e.g., shifted slightly from the true value), RMSE penalizes it twice: once for the magnitude difference at the predicted and actual points, and again for failing to match the value at the shifted location. To address the phase error quantification between observations and simulations, peaks in the hourly time series were identified using the "find peaks" function available in MATLAB. This was applied to both the observed and simulated data. The phase error was then calculated by measuring the time difference, in hours, between the occurrence of each peak in the observations and the corresponding peak in the simulations. This allowed for a direct assessment of the model's ability to capture the timing of key events, such as storm surge.

Bias:

$$Bias = \bar{S} - \bar{O} \quad (40)$$

Where $\bar{S}$ and $\bar{O}$ are the average simulation and observation values respectively. A value closer to zero identifies a better performance, negative values indicate underestimation, and positive values indicate overestimation from the simulations. Given that both observed and simulated data were detrended and had their mean removed, bias was solely applied to the analysis of values exceeding the 99th percentile.

Slope of linear fit between observations and simulation (SLF):

$$S = m\,O + b \tag{41}$$

Where the slope is given by the coefficient m. A value closer to one indicates a better performance.

Mean Absolute Deviation (MAD):

$$MAD = \overline{|S - O|} \tag{42}$$

A value closer to one indicates a better performance.

Additionally, with the aim of considering the representation of extremes by the simulations, two new metrics based on customized versions of the Mean Absolute Deviation are introduced:

MAD of the percentiles (MADp):

$$MADp = \overline{|S_{prc} - O_{prc}|} \tag{43}$$

Where S_{prc} and O_{prc} are the simulation and observation percentile values, considered from 0 to 100%, every 1%. The MADp metric provides a comprehensive assessment of simulation model performance by comparing percentile values derived from simulations (S_{prc}) with those observed (O_{prc}). This evaluation encompasses the entire distribution, from the lowest to the highest percentiles, allowing to gauge the model's accuracy across a range of scenarios. MADp is particularly valuable for its sensitivity to systematic errors, such as persistent underestimation of high percentiles, which can significantly impact the reliability of simulation results. By penalizing these systematic errors, MADp highlights areas where improvements in the simulation model are necessary to better align with observed data. Lower MADp values indicate closer agreement between simulations and observations.

Corrected MAD (MADc):

$$MADc = \text{MAD} + MADp \tag{44}$$

In this indicator the ability of the "traditional" MAD to capture the model's skill is exploited but reducing its strong penalization of the phase error or timing error by adding the MAD on the percentiles (MADp) previously defined. MAD measures the average absolute difference between simulated and observed values, while MADp evaluates the average percentage deviation between them. By combining these two components, MADc provides a comprehensive evaluation of the simulation model's performance, considering both the magnitude and percentage deviations. A lower MADc value indicates better agreement between simulated and observed values, reflecting higher accuracy and reliability of the simulation model.

To validate the proposed metrics, a sinusoidal time series was generated to represent an observed parameter. Then, two time series representing simulations were created: one with the same amplitude as the observation but shifted in time (phase error), and a second time series with the same phase as the observation but with half the amplitude. The different metrics were then calculated and plotted on scatter plots (Figure 7). The results show better performance for the simulation that underestimates the observations in terms of Pearson correlation, RMSE, and MAD. Conversely, the time series representing the simulation that accurately captures the amplitude is penalized for the phase error, which negatively impacts its performance with the aforementioned metrics. However, the proposed MADp and MADc metrics identify it as the better model.

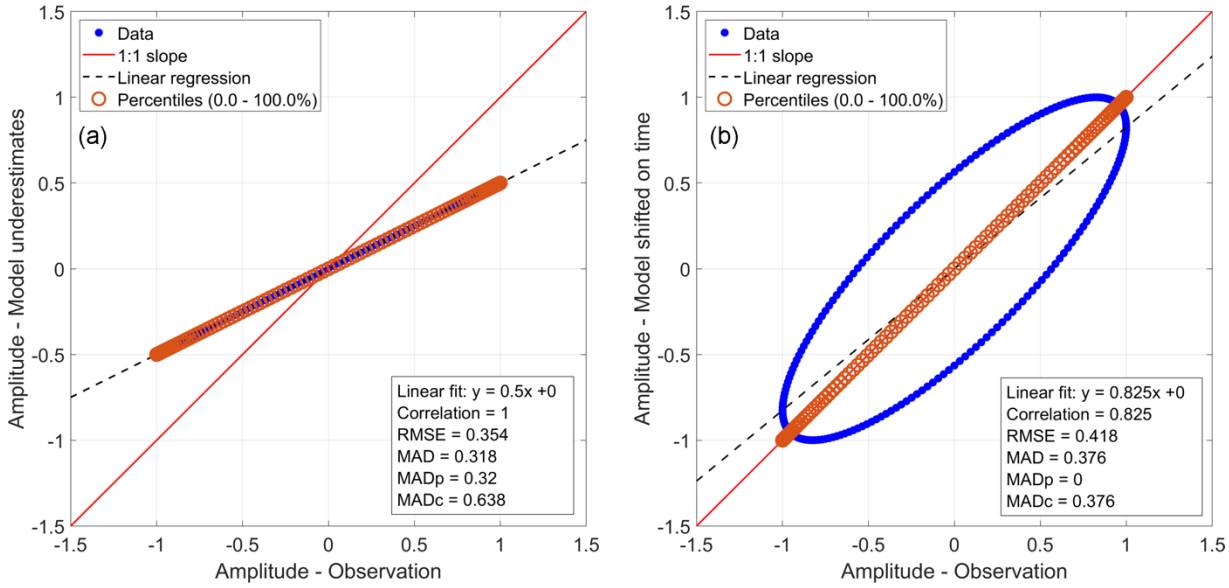

Figure 7: Scatter plots and metrics for the validation of the proposed metrics, MADp and MADc. (a) Observation vs model with half amplitude and same phase; (b) Observation vs model with same amplitude and phase shifted.

2.2 MACHINE LEARNING DOWNSCALING

2.2.1 PREDICTAND

The predictand used for the ML data-driven downscaling is the surge time series generated by the BC-UniGe simulations, validated in Section 3.1. Additionally, observed data were used as a predictand to compare the performance of algorithms trained on dynamically modeled data versus those trained directly on observations. However, due to gaps and inconsistencies in the starting dates of observational data at the CNR platform, Caorle, Grado, and Monfalcone, the downscaling with observational data as a predictand was performed only at Punta della Salute and Trieste. These locations offer consistent and long-term observed data (Figures 8 and 9), a crucial requirement for training the implemented neural network models.

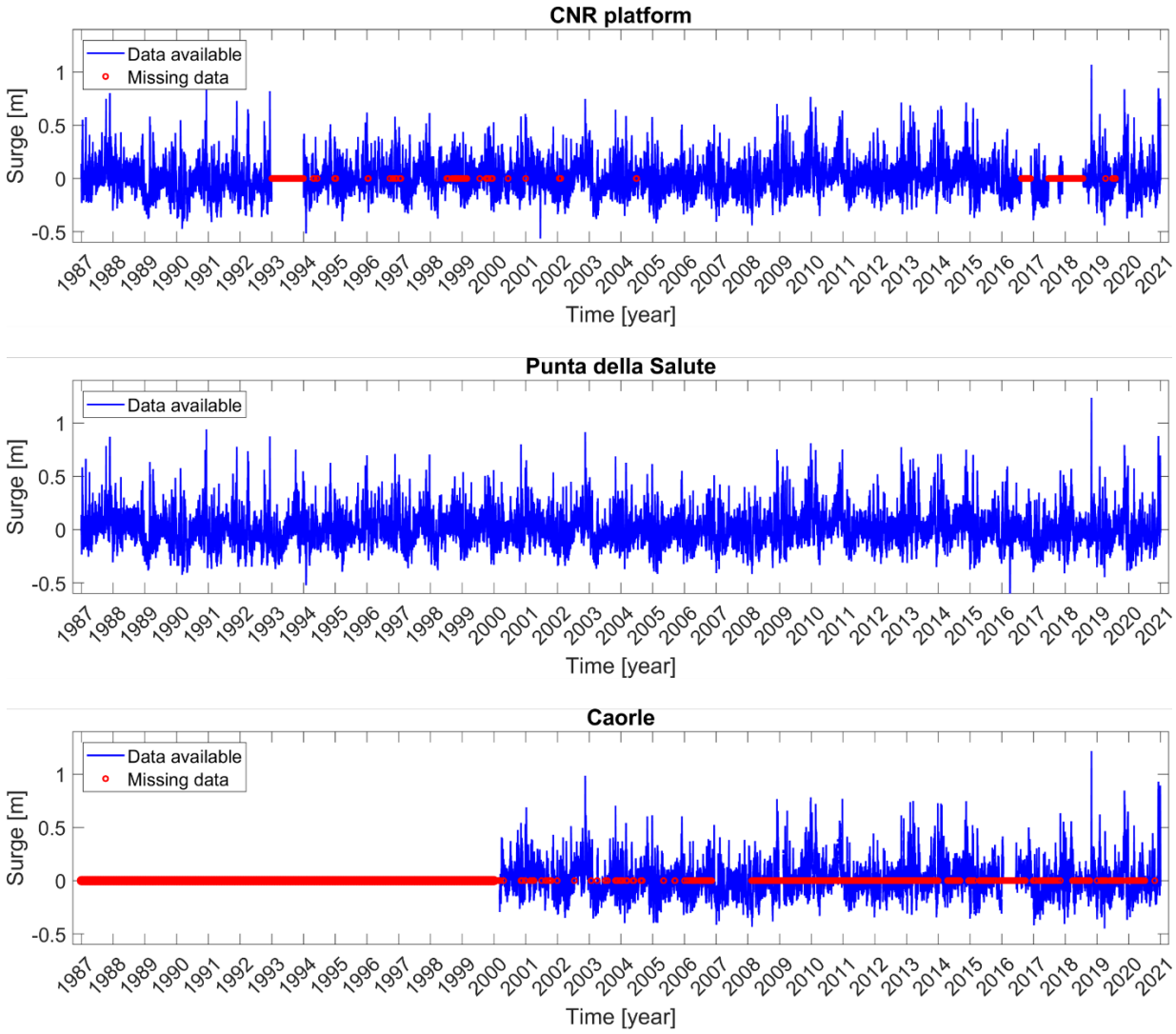

Figure 8: Time-series for observed storm surge available at the CNR platform, Punta della Salute, and Caorle.

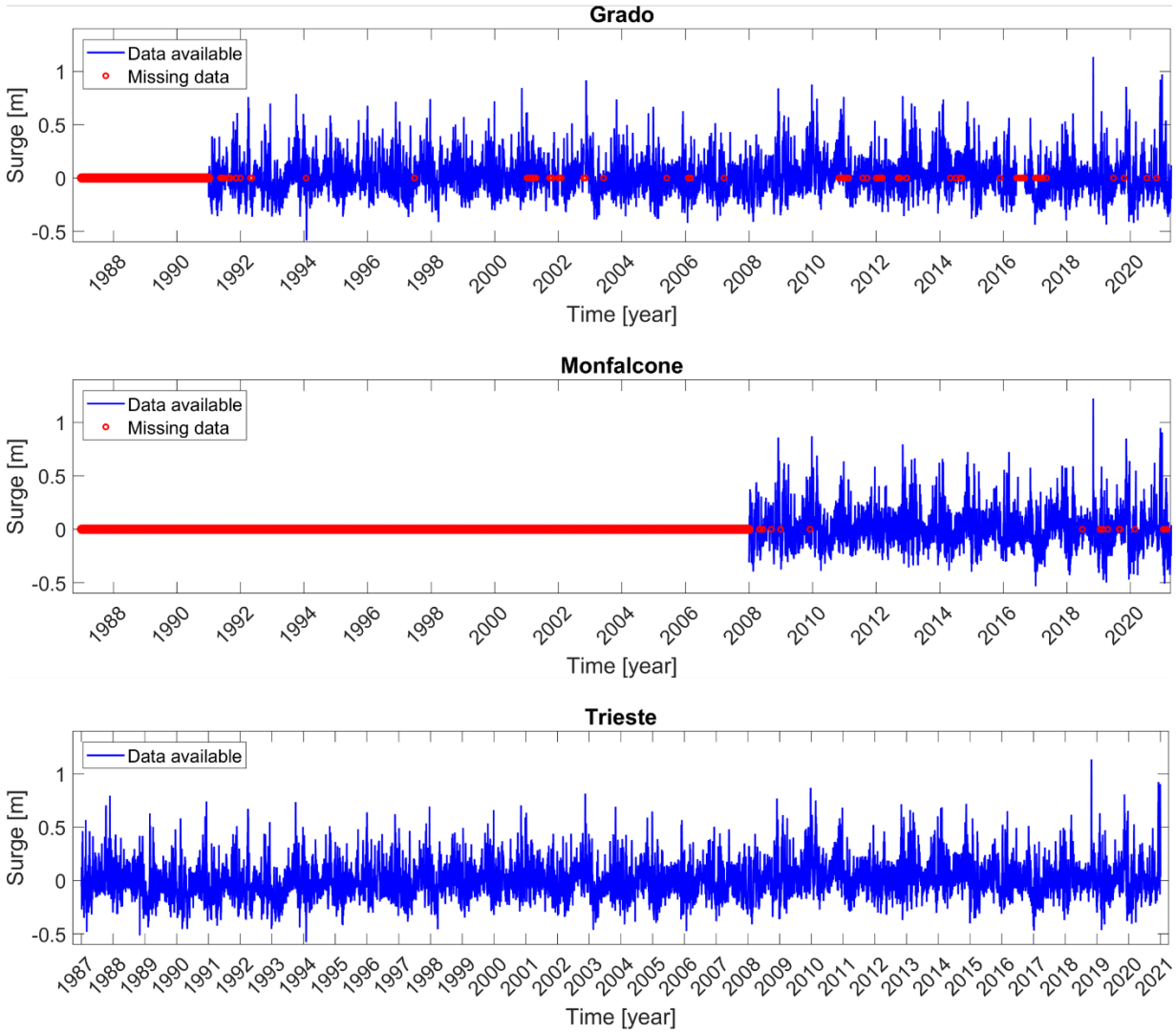

Figure 9: Time-series for observed storm surge available at Grado, Monfalcone, and Trieste.

2.2.2 PREDICTORS

The predictors included for the ML downscaling are meteorological and oceanographic variables known to influence the prediction of storm surge and have been used in recent works for data-driven downscaling (e.g., Kim et al., 2019; Žust et al., 2021; Chen et al., 2022; Rus et al., 2023a; Tausía et al., 2023; Harter et al., 2024; Dang et al., 2024). Specifically, the following predictors were considered, which were also used during the dynamical downscaling process: sea surface height from Med-MFC, mean sea level pressure and wind fields from ERA5, and tides from FES2014. To

ensure consistent data dimensions, the spatial ERA5 data were interpolated to match the Med-MFC spatial resolution, following the method outlined by Wang et al. (2024).

The spatial domain used for the predictors was determined through multiple executions and performance analysis of Multivariate Linear Regression models. Figure 10 illustrates the selected domain by displaying sea surface height at a randomly chosen time. Since most of the predictors considered are parameter fields and the problem to solve is a time-series regression, a dimensionality reduction step is applied to reduce spatial variability using PCA. For this task, the "pca" function available on MATLAB was used. With this function the following parameters were obtained:

- Principal Component coefficients (coeff): also known as "loadings", this matrix contains the eigenvectors of the covariance matrix of the predictor. Each column corresponds to a principal component and shows how much each original variable (grid points of the predictor field in this study) contributes to that component. The components are sorted in descending order of explained variance.

- Scores (score): these are the principal component representations of the predictor in the transformed space. Each row represents an observation (time step in this study), and each column corresponds to a principal component.

- Explained variance: represents the percentage of the total variance explained by each principal component. It helps determine how many components are needed to capture most of the variability in the dataset.

After the application of the PCA, the spatial patterns (EOFs) and time-series for each principal component (PC) were obtained, following the expressions in Equations 45 and 46, respectively.

$$EOF = coeff_i * std(predictor) \quad (45)$$

$$PC = coeff_i * score_i \quad (46)$$

Where $std(predictor)$ corresponds to the standard deviation of each grid node of the predictor field across time, and $coeff_i$ and $score_i$ are the Principal Component coefficient and Scores of the ith Principal Component.

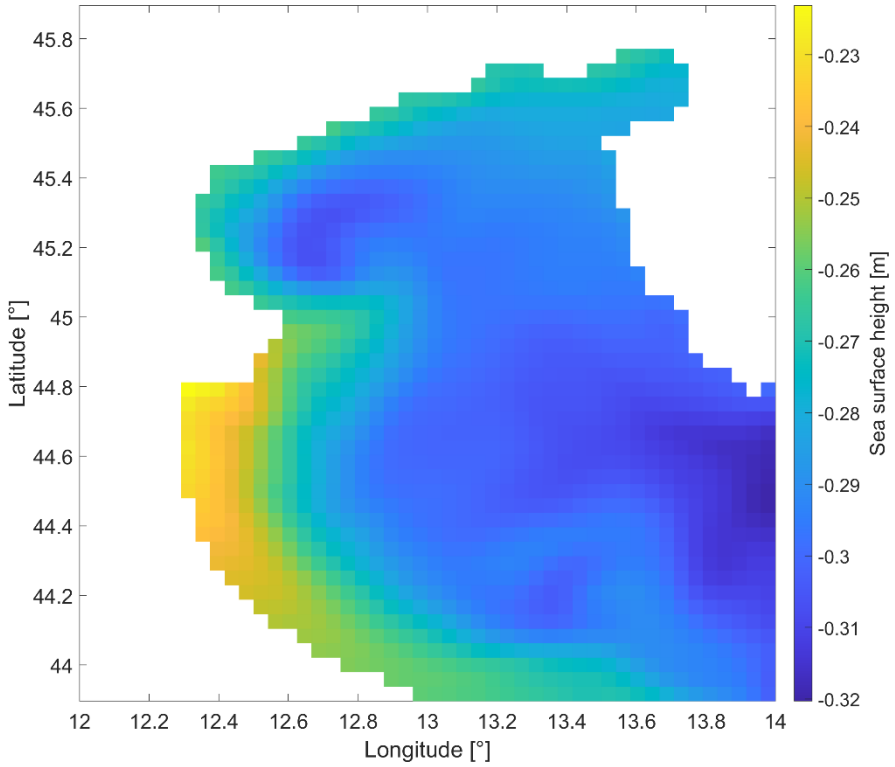

Figure 10: Spatial domain of predictors, exemplified by sea surface height at a random time from the Med-MFC database.

The construction of the predictors used for the implementation of the ML models considers the 7 first PCs of each predictor included separately, i.e., without reconstructing a single time-series of the predictor. The quantity of PCs used were selected based on execution of Multivariate Linear Regression models with different numbers of PCs and evaluating the performance of the models, finding an optimum value of 7. Is important to note that this process was not carried out for tides, which were reconstructed at each location as a single hourly time-series.

2.2.3 CONFIGURATION OF THE MODELS

The ML models were implemented in Python with the PyTorch library, specifically designed for time-series prediction tasks. The models considered in this study are Multivariate Linear Regression, Multilayer-Perceptron, Recurrent Neural Network, and

Long Short-Term Memory network. These models were implemented under different configurations, as follows:

- Multivariate Linear Regression (MLR) model: The MLR model is designed to model the linear relationship between the predictors and the predictand. The model estimates the coefficients of the linear equation through least squares optimization, minimizing the error between the predicted and target values.

- Multilayer Perceptron (MLP) model: The MLP is a fully connected feedforward neural network with two hidden layers, each containing 120 neurons, tailored for time-series regression tasks. The architecture includes an input layer, followed by hidden layers with ReLU activations, which introduce non-linearity to capture complex patterns. The output layer produces the final predictions.

- Simple Recurrent Neural Network (RNNs) model: This model features a single hidden layer with 120 hidden units, suited for sequential data. The recurrent nature of the hidden layer allows information to flow across time steps, helping the model learn temporal dependencies. The output layer maps the recurrent outputs to the final prediction.

- Hybrid Recurrent Neural Network (RNNh) model: The RNNh model combines a traditional RNN layer with a linear layer for enhanced feature learning. The RNN layer has 60 hidden units, while the linear layer independently transforms the input features, which are then concatenated with the recurrent outputs. ReLU activation is applied to the linear layer output before concatenation, enhancing non-linearity and potentially improving performance on complex data.

- Simple Long Short-Term Memory (LSTMs) model: This model includes an LSTM layer with 120 hidden units, initialized to zero for each sequence. The LSTM's internal gating enables it to retain information over longer time frames, making it suitable for time-series data with delayed dependencies.

- Hybrid Long Short-Term Memory (LSTMh) model: The LSTMh model incorporates both an LSTM layer (60 units) and a linear layer, combining their outputs to enhance predictive accuracy. The LSTM processes the sequential data, while the linear layer independently transforms the input features. The linear output is activated by ReLU, then concatenated with the LSTM output, leveraging both long-term dependencies and feature transformation.

The depth and the size of the hidden units for the NN models (MLP, RNN, and LSTM) was determined as the optimal configuration for each algorithm, based on various test cases conducted during their implementation.

As loss functions both MSE and $MADc^2$, were applied across all models. It is important to mention that only quadratic loss functions were considered due to their differentiability and smooth gradient profile, which creates an optimization landscape that is better suited for gradient descent. The smooth and continuous gradients facilitate stable parameter adjustments, promote faster convergence, and support more accurate results by allowing fine-tuning in low-error regions, concepts explained on Section 1.4.2. Considering the different configurations and loss functions, 12 ML models were implemented (Table 5).

Table 5: Machine Learning models implemented.

ML model configuration	Loss function	Acronym
MLR model	MSE	MLR-MSE
MLP model		MLP-MSE
RNNs model		RNNs-MSE
RNNh model		RNNh-MSE
LSTMs model		LSTMs-MSE
LSTMh model		LSTMh-MSE
MLR model	$MADc^2$	MLR- $MADc^2$
MLP model		MLP- $MADc^2$
RNNs model		RNNs- $MADc^2$
RNNh model		RNNh- $MADc^2$
LSTMs model		LSTMs- $MADc^2$
LSTMh model		LSTMh- $MADc^2$

2.2.4 TRAIN AND TEST

To prepare the data for the train and test of the ML models, both the predictand and predictors were normalized by removing the mean and dividing by the standard deviation, resulting in zero mean and unit variance. This normalization helps the models assign appropriate weights to each predictor, avoiding the influence of larger-magnitude variables.

For training, 80% of the data (28 years) was allocated, following the approach of Gholamy et al. (2018), while the remaining 20% (6 years) was divided into 1 year for validation and 5 years for testing. Notably, the training and testing sets were chosen to have similar distributions, ensuring consistency and comparability across the sets,

as recommended by Uçar et al. (2020). Each model was trained 10 times, with each run consisting of 600 epochs, except for the MLR model for which 10000 epochs were considered. To optimize the models' parameters the Adam optimizer was used.

The validation set was used to select the best-performing model run based on the validation-based model selection method. This widely used approach evaluates and identifies the optimal model or hyperparameter configuration during training (Bishop, 2006; Goodfellow et al., 2016). In this study, the criterion for applying validation-based model selection was the MADc value for surge events above the 99th percentile, ensuring that the chosen run for each model delivers the best possible performance on extreme surges.

Once the optimal run was identified, the testing set was employed to evaluate model performance. Metrics for evaluation, detailed in Section 2.1.3, include assessments over the entire dataset and values above the 99th percentile. Additionally, to quantify the impact of using MADc^2 as the loss function, the percentage variation of the error metrics obtained from ML models using MSE and MADc^2 as loss functions was calculated. This analysis aimed to evaluate potential improvements with the proposed metric, following the expression in Equation 47.

$$Percentage\ variation = \frac{AI_{MSE} - AI_{MADc^2}}{AI_{MSE}} * 100 \quad (47)$$

Where AI_{MSE} corresponds to the value of each error metric from the ML model using MSE as loss function, and AI_{MADc^2} represents the value of the same error metric from the ML model with the implementation of MADc^2 as the loss function. Positive values of the percentage variation indicate improvements in performance with MADc^2, while negative values represent decrease in performance.

In this study, the capabilities of the ML models to represent observed surges were analyzed using two distinct approaches. The first approach involved training the ML models with output from the BC-UniGe dynamical downscaling and testing their performance against observed data. Due to observed data availability, in Caorle and Monfalcone only one year of data is used for the performance evaluation for the first approach. The second approach utilized observations for both training and testing the ML models. This second method was applied exclusively at Punta della Salute and Trieste, where long-term observed data were available (Figures 8 and 9).

The purpose of employing these two approaches was to evaluate the performance of the ML models under different data availability scenarios and to assess how

effectively these models compare to a high-resolution, state-of-the-art dynamical model, which serves as the baseline for the comparisons. Table 6 provides a summary of the ML model implementations used for comparison with observational data.

Table 6: Approaches applied to compare ML models with observations.

Approach	Train target	Test target	Models applied	Locations
First	BC-UniGe	Observations	All models from Table 5	All locations from Table 3
Second	Observations			Punta della Salute and Trieste

3 RESULTS

3.1 DYNAMICAL DOWNSCALING

The Probability Distribution Estimates (PDE) and Empirical Cumulative Distribution Functions (ECDF), Figures 11 to 16, show that BC-UniGe better represents the higher values of storm surge when compared with observations, particularly when considering values above the 99th percentile However, some overestimations are noticeable at Caorle (Figure 13) and Monfalcone (Figure 15) with BC-UniGe. In contrast, simulations with ERA5 forcing tend to underestimate these higher values, which is more noticeable for BT-ERA5.

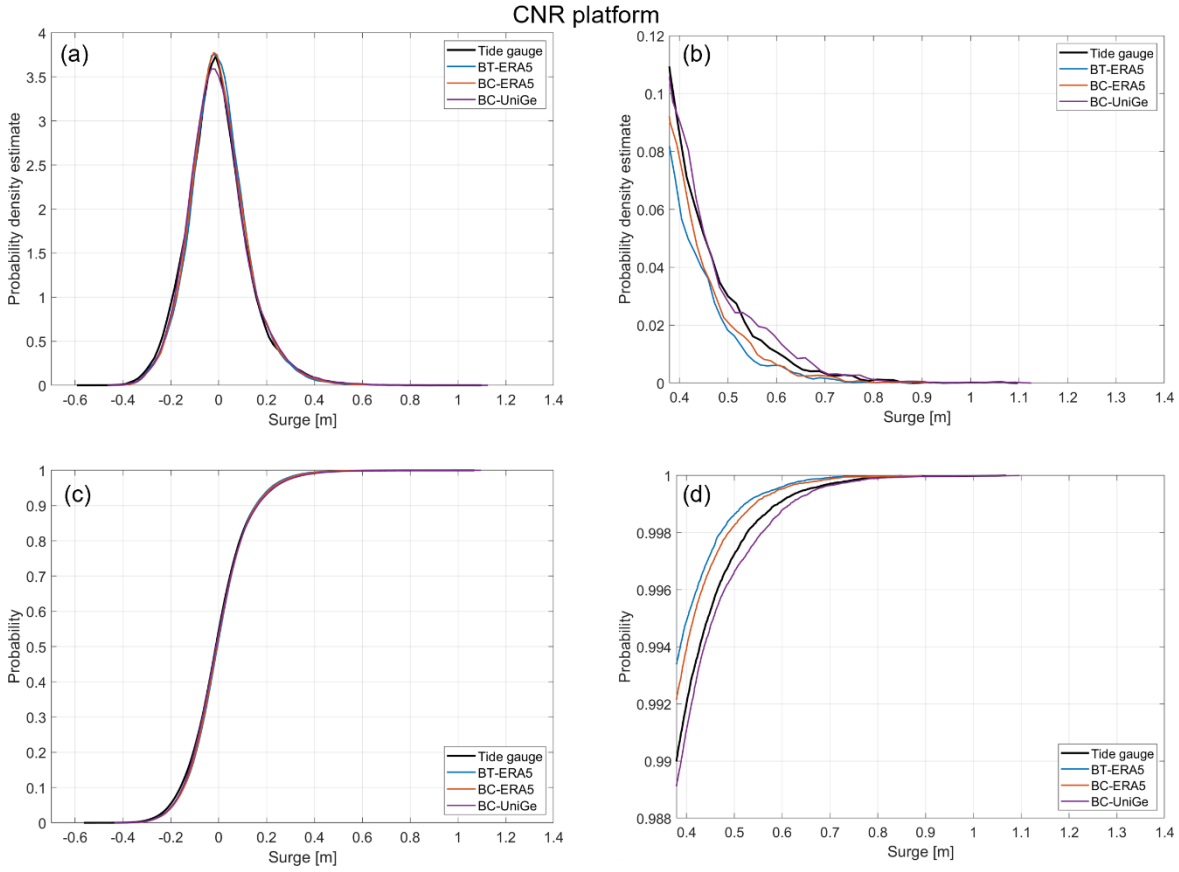

Figure 11: Probability Density Estimates (PDE) and Empiricial Cumulative Distribution Function (ECDF), CNR platform. (a) PDE for the total amount of data; (b) PDE for values above 99th percentile of the observed data; (c) ECDF for the total amount of data; (d) ECDF for the total amount of data; (d) ECDF for values above the 99th percentile of the observed data.

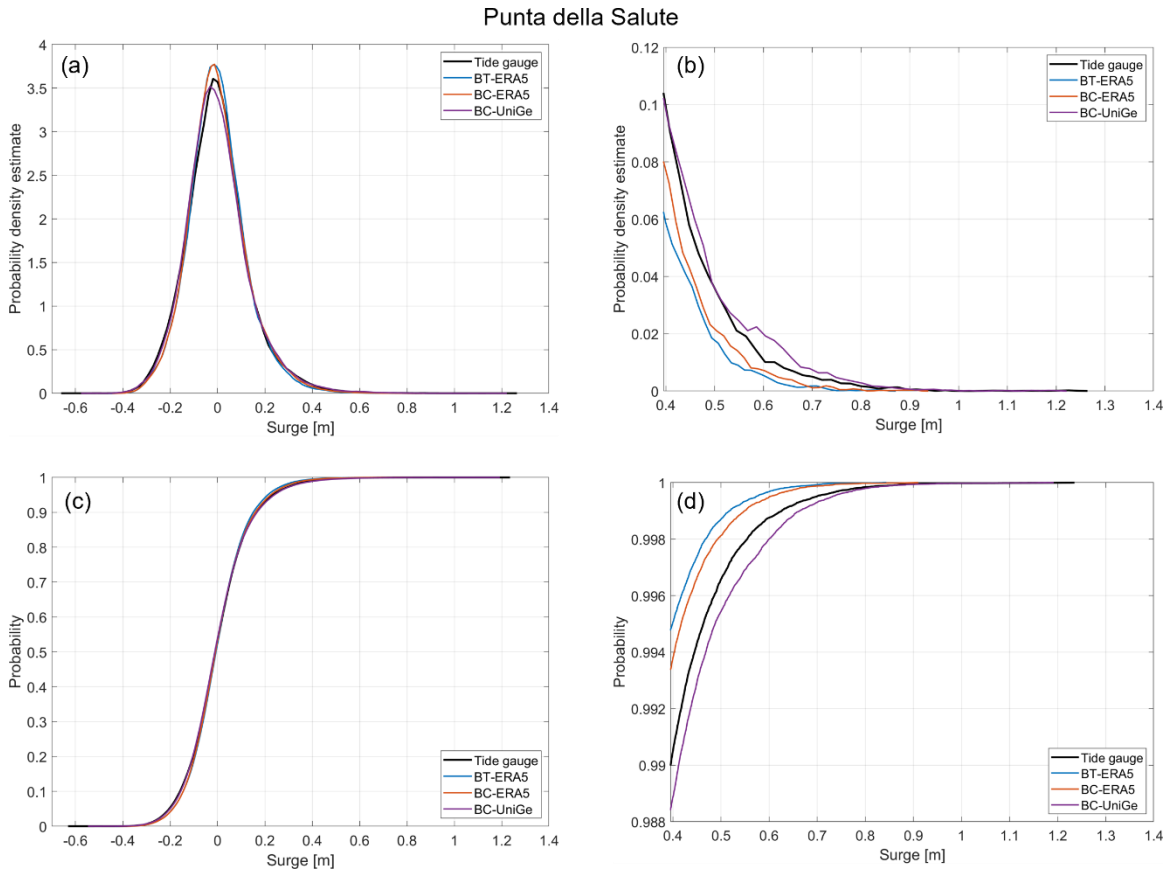

Figure 12: Probability Density Estimates (PDE) and Empiricial Cumulative Distribution Function (ECDF), Punta della Salute. (a) PDE for the total amount of data; (b) PDE for values above 99th percentile of the observed data; (c) ECDF for the total amount of data; (d) ECDF for the total amount of data; (d) ECDF for values above the 99th percentile of the observed data.

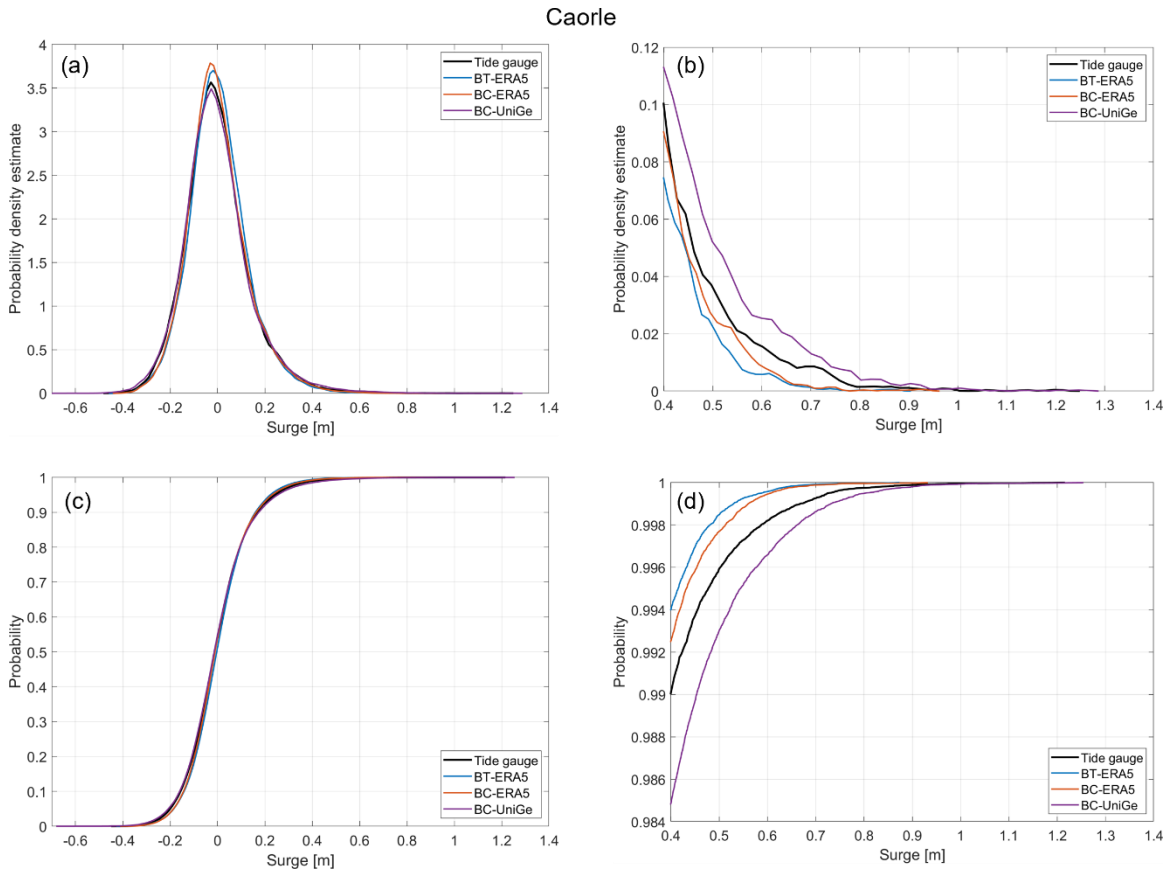

Figure 13: Probability Density Estimates (PDE) and Empiricial Cumulative Distribution Function (ECDF), Caorle. (a) PDE for the total amount of data; (b) PDE for values above 99th percentile of the observed data; (c) ECDF for the total amount of data; (d) ECDF for the total amount of data; (d) ECDF for values above the 99th percentile of the observed data.

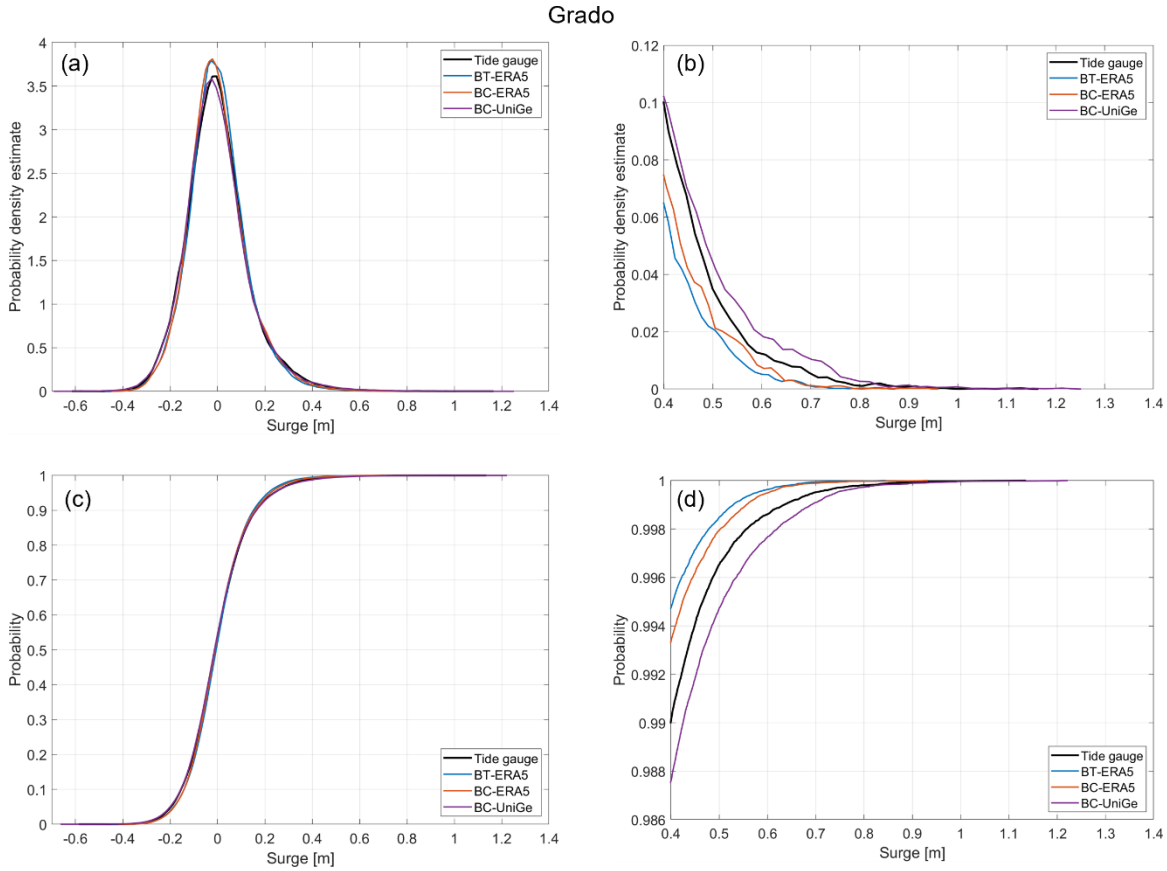

Figure 14: Probability Density Estimates (PDE) and Empiricial Cumulative Distribution Function (ECDF), Grado. (a) PDE for the total amount of data; (b) PDE for values above 99th percentile of the observed data; (c) ECDF for the total amount of data; (d) ECDF for the total amount of data; (d) ECDF for values above the 99th percentile of the observed data.

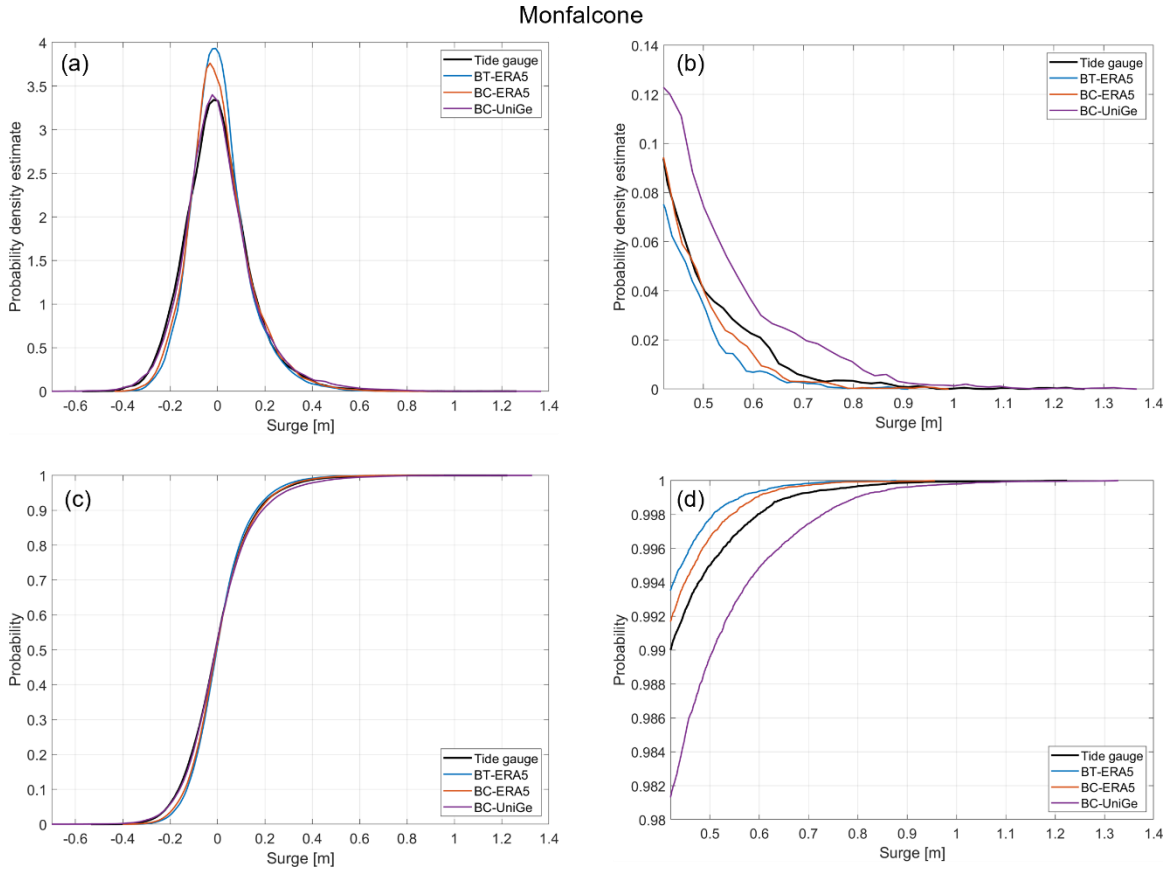

Figure 15: Probability Density Estimates (PDE) and Empiricial Cumulative Distribution Function (ECDF), Monfalcone. (a) PDE for the total amount of data; (b) PDE for values above 99th percentile of the observed data; (c) ECDF for the total amount of data; (d) ECDF for the total amount of data; (d) ECDF for values above the 99th percentile of the observed data.

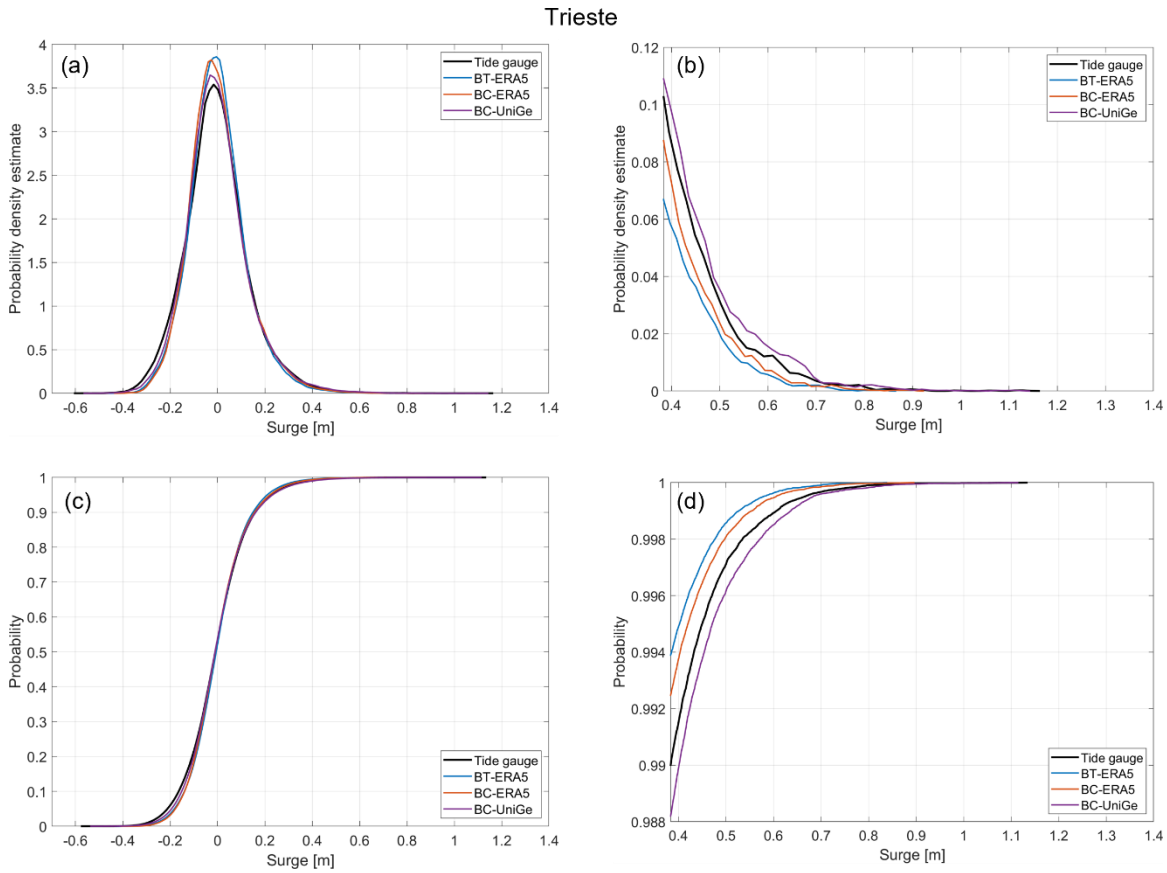

Figure 16: Probability Density Estimates (PDE) and Empiricial Cumulative Distribution Function (ECDF), Trieste. (a) PDE for the total amount of data; (b) PDE for values above 99th percentile of the observed data; (c) ECDF for the total amount of data; (d) ECDF for the total amount of data; (d) ECDF for values above the 99th percentile of the observed data.

The performance evaluation in Figure 17 shows that, if the model performance is assessed in terms of Pearson correlation, RMSE, and MAD, the surges simulated with the ERA5 forcing fit better to the measured data. The Pearson correlation coefficients obtained range between 0.8 and 0.9 in all locations for the three simulations, with maximum of 0.842 with BT-ERA5 in Grado (Figure 17d). Regarding the RMSE, mean values of 0.077 m for BT-ERA5, 0.075 m for BC-ERA5, and 0.079 m for BC-UniGe were obtained, with a minimum of 0.072 m (BT-ERA5 in Grado, Figure 17d) and a maximum of 0.094 m (BC-UniGe in Monfalcone, Figure 17e). Similar results are obtained for MAD, which shows better performance for the simulations with ERA5 forcing at all locations. Only in Trieste does BC-UniGe achieve the same performance as BC-ERA5 for this metric. Despite the aforementioned, the best performance is achieved by BC-UniGe in the linear fit slope, with values above 0.8 in all locations and a maximum of 0.869 in Monfalcone (Figure 17e). For this parameter, the less favorable performance is obtained with BT-ERA5 in all locations.

For MADp, the best performance is achieved by BC-UniGe in all locations, with a mean value of 0.004 m, while less favorable results are obtained with BT-ERA5, with a mean of 0.011 m. Similar results were obtained for MADc, except in Caorle (Figure 17c) and Monfalcone (Figure 17e), where BC-ERA5 showed better performance, likely due to overestimation in the mentioned sites. These results underscore the importance of considering percentiles as part of the performance evaluation. BC-UniGe simulations demonstrate an improvement in representing extreme values, showing a better fit of the highest percentiles, which can be noticed in Figure 18 and Figure 19. Additionally, these figures indicate that BC-UniGe simulations produce greater dispersion of data, likely due to a more frequent occurrence of phase error, which was quantified as 3.1% higher than in BT-ERA5, and 4.5% higher than in BC-ERA5. However, they also exhibit a better fit of the linear regression, and a more accurate representation of extreme values compared to BC-ERA5, which fail to represent the most extreme events in each location.

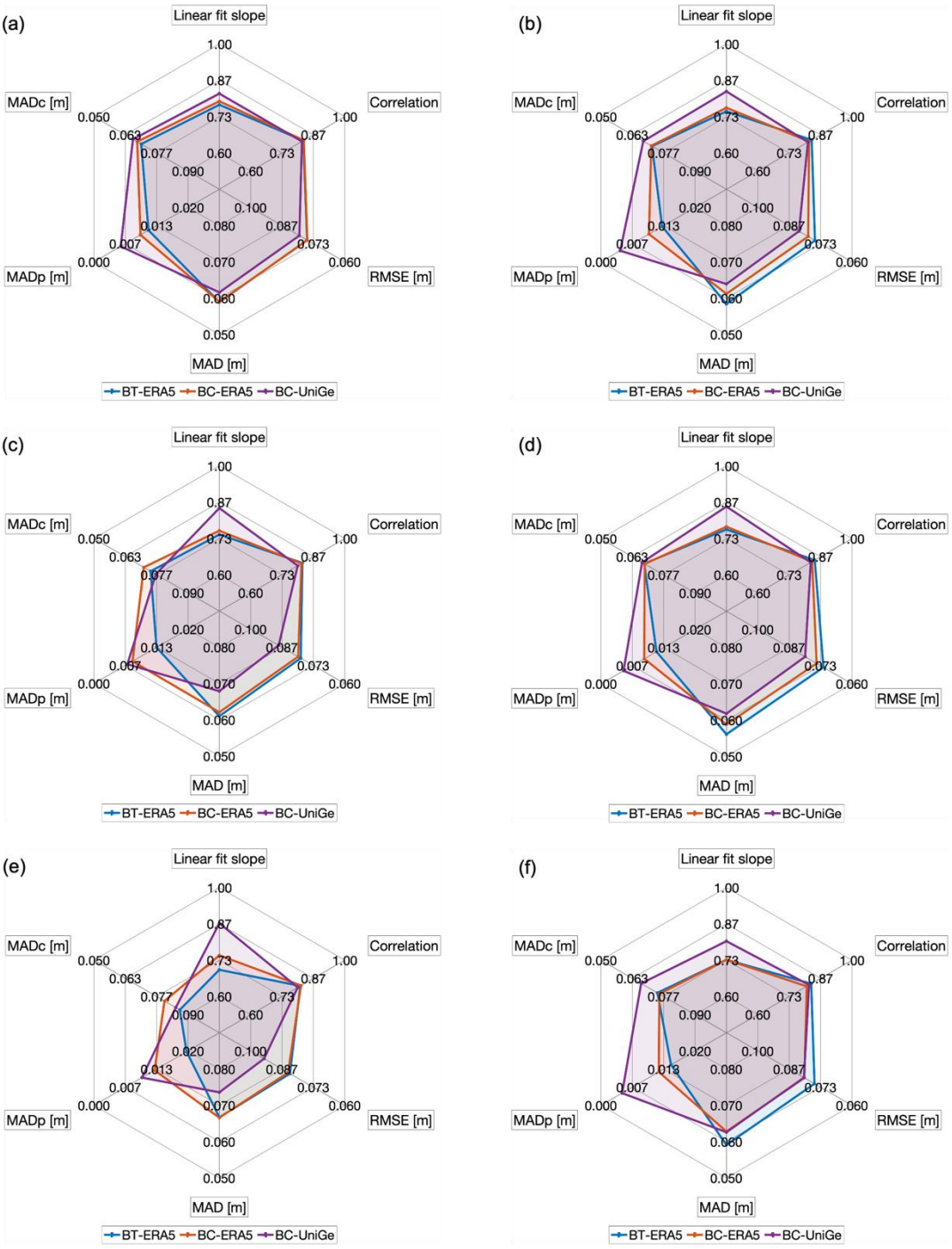

Figure 17: Radar charts of evaluation metrics for the total amount of data in all locations. (a) CNR platform; (b) Punta della Salute; (c) Caorle; (d) Grado; (e) Monfalcone; (f) Trieste. For RMSE, MADp and MADc a reverse axis is used, this ensures that simulations covering a larger area on each metric represent a better performance (i.e. values on the fringe refer to better performance).

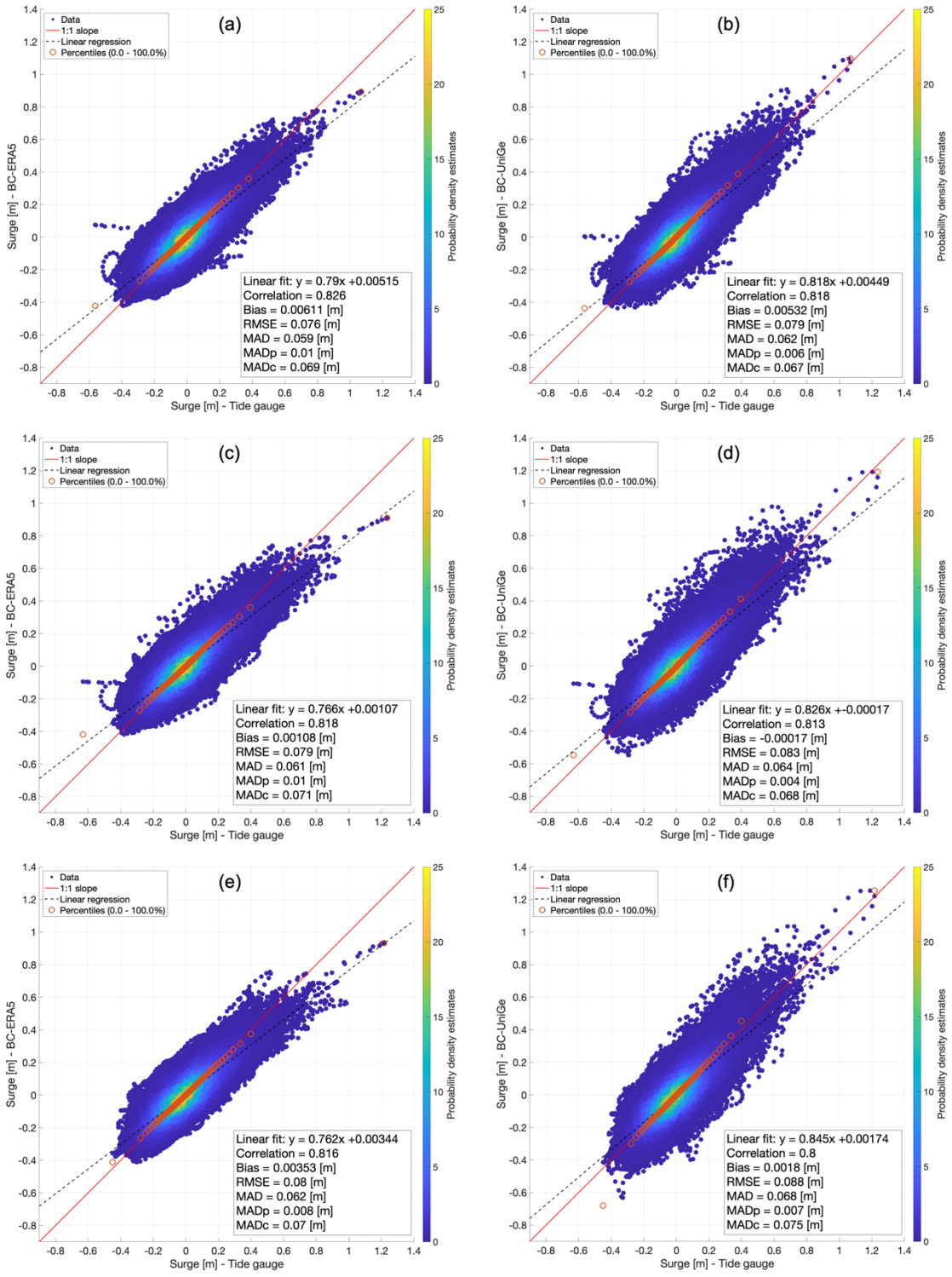

Figure 18: Scatter plots between tide gauges and baroclinic simulations. CNR platform: (a) BC-ERA5, (b) BC-UniGe; Punta della Salute: (c) BC-ERA5, (d) BC-UniGe; Caorle: (e) BC-ERA5, (f) BC-UniGe.

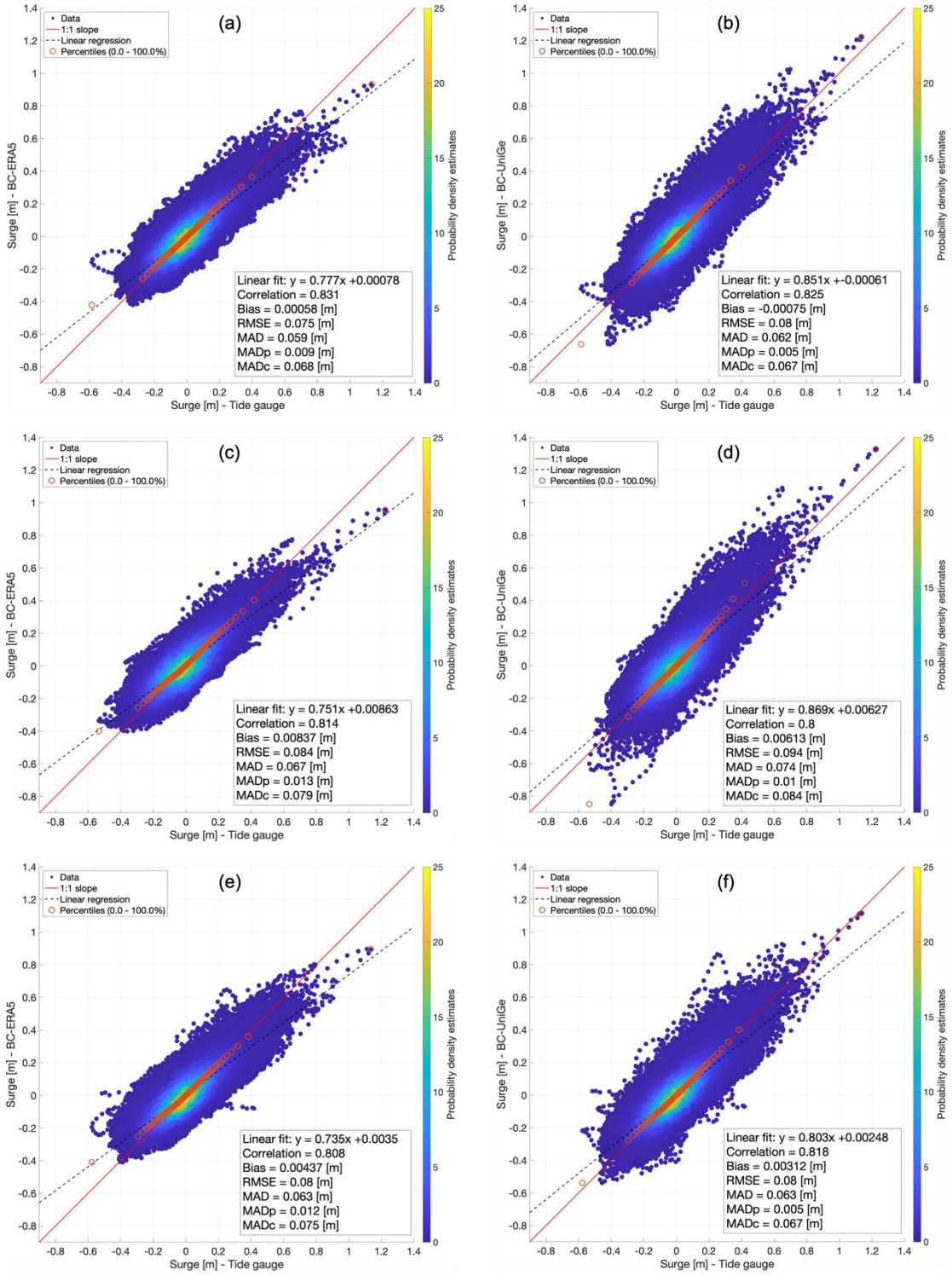

Figure 19: Scatter plots between tide gauges and baroclinic simulations. Grado: (a) BC-ERA5, (b) BC-UniGe; Monfalcone: (c) BC-ERA5, (d) BC-UniGe; Trieste: (e) BC-ERA5, (f) BC-UniGe.

The results of the error metrics for surge values above the 99th percentile, represented using radar charts (Figure 20), confirm that, in general, better performance is observed with BC-UniGe, while less favorable results are obtained for BT-ERA5. Although the transition from barotropic to baroclinic configuration indicates an improvement in the representation of extremes (Weisberg & Zheng, 2008; Staneva et al., 2016; Hetzel et al., 2017; Ye et al., 2020; Muñoz et al., 2022), the utilization of UniGe forcing represents the best improvement across practically all metrics. Only in Caorle (Figure 20c) and Monfalcone (Figure 20e) does BC-ERA5 show better Pearson correlation, RMSE, and MAD; additionally, in the latter, MADc exhibits better performance for that simulation, likely due to overestimation of the peaks by BC-UniGe in Monfalcone. In the other locations, it's evident that BC-UniGe performs better in representing the highest storm surge values.

In order to show the capacity of the different model configurations to represent certain known storm events at each location, Figure 21 shows time series of different storm surge events at each location. These extreme events were chosen according to the contributions of Lionello et al. (2012), Međugorac et al. (2018), Ferrarin et al. (2020), Umgiesser et al. (2021), and Giesen et al. (2021). As mentioned before, the incorporation of the UniGe forcing implies a significant improvement in the representation of extreme events, clearly evident in the peak values of the storm surge. Despite, an overestimation of some surge peaks is also observed in the events chosen at Punta della Salute (Figure 21b), Caorle (Figure 21c), and Monfalcone (Figure 21e) with BC-UniGe. On the other hand, a systematic underestimation of extremes obtained in simulations with ERA5 forcing is notable on every surge peak.

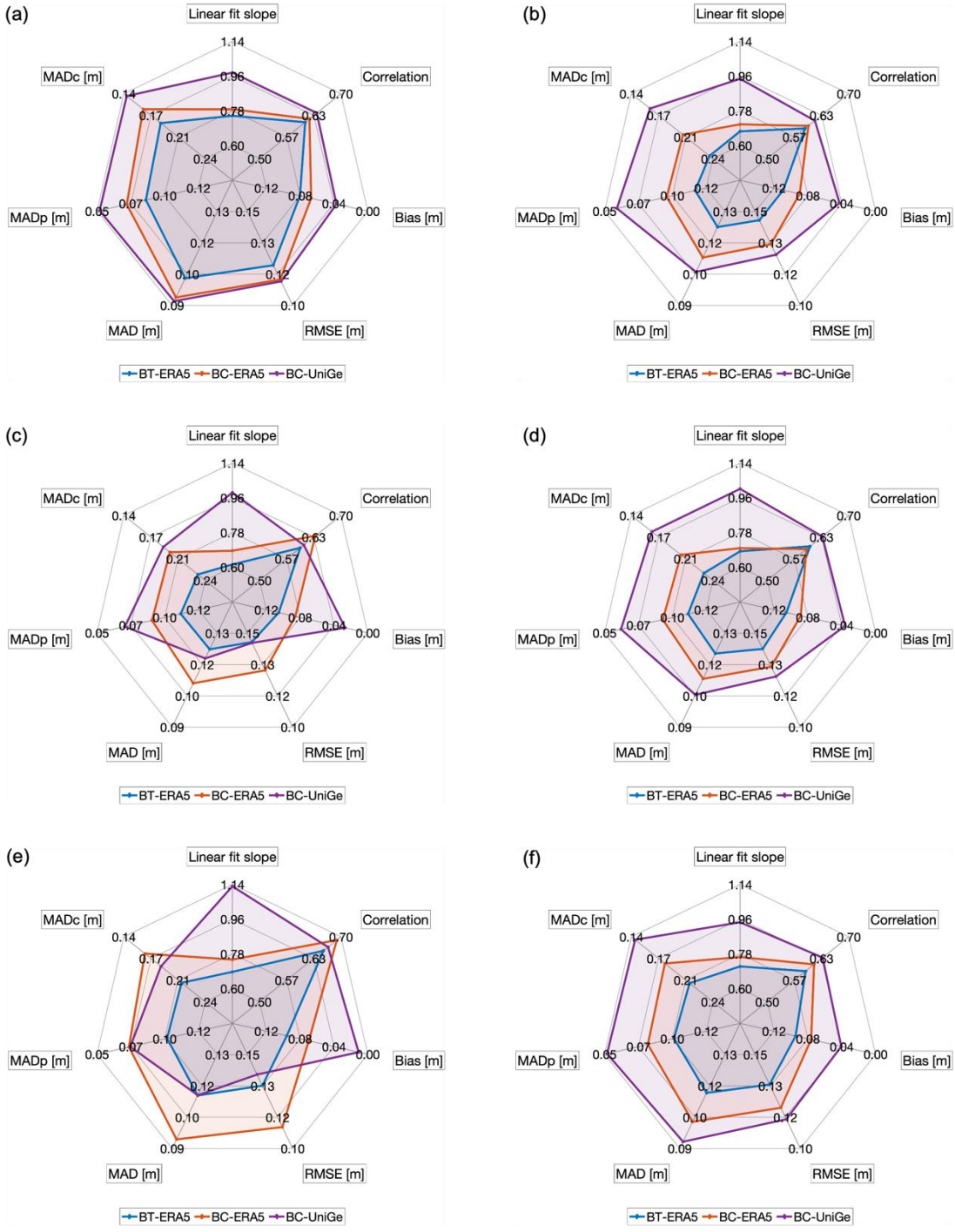

Figure 20: Radar charts of evaluation metrics for surge values above the 99th percentile of the cumulative distribution at each location. (a) CNR platform; (b) Punta della Salute; (c) Caorle; (d) Grado; (e) Monfalcone; (f) Trieste. Bias is represented by absolute value. Also, for RMSE, Bias, and MADp and MADc a reverse axis is used, this ensures that simulations covering a larger area on each metric represent a better performance (i.e. values on the fringe refer to better performance).

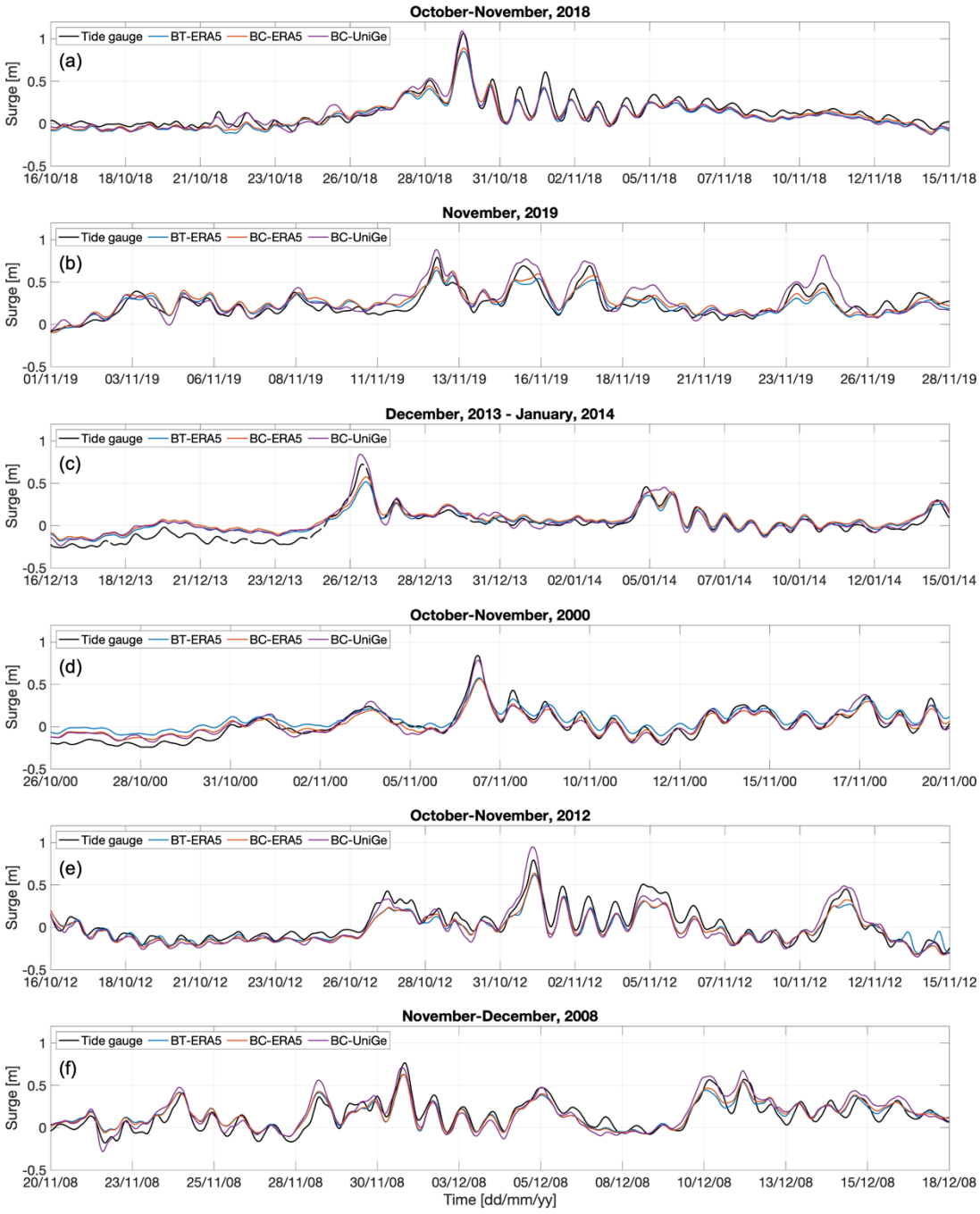

Figure 21: Time series of different storm surge events in all the locations, tidal gauge versus model. (a) CNR platform; (b) Punta della Salute; (c) Caorle; (d) Grado; (e) Monfalcone; (f) Trieste.

3.2 MACHINE LEARNING DOWNSCALING

After evaluating the performance of the dynamical downscaling and recognizing BC-UniGe as the simulation with the best representation of extreme surge values, the ML models were implemented using its output for both training and testing, alongside observations. First, Section 3.2.1 presents the main results obtained from the Principal Component Analysis applied to the predictors. The results of ML downscaling process are presented in Sections 3.2.2, 3.2.3, and 3.2.4, following the methods outlined in Section 2.2. This subdivision is made to assess the impact of different training and testing data configurations on the performance of the ML models, allowing for a comprehensive understanding of how the models behave under various conditions.

Section 3.2.2 focuses on analyzing the performance of the different ML models configurations and the impact of implementing $MADc^2$ as the loss function, with BC-UniGe used for both training and testing.

Section 3.2.3 presents the results of training the ML models with BC-UniGe and testing on observed surges, comparing their performance with the high-resolution dynamical downscaling. Finally, Section 3.2.4 evaluates the performance of the ML models when both training and testing are based on observations, providing insight into how well the models generalize to real-world data and comparing with the state-of-the-art dynamical downscaling.

As a remainder, the following are the acronyms of the ML models implemented in this study: MLR (Multivariate Linear Regression), MLP (Multilayer-Perceptron), RNNs (Simple Recurrent Neural Network), RNNh (Hybrid Recurrent Neural Network), LSTMs (Simple Long Short-Term Memory network), and LSTMh (Hybrid Long Short-Term Memory network).

3.2.1 PRINCIPAL COMPONENT ANALYSIS

The results of the Principal Component Analysis applied to the considered predictors show that a significant portion of the variance is captured by the first principal component (PC1) in all cases. This is illustrated in Figures 22 and 23, which depict the explained variance for each predictor. For mean sea level pressure (MSLP), PC1 alone accounts for 99.57% of the total variance, while the first 7 PCs collectively explain 99.98% (Figure 22), indicating that the dataset's variability is predominantly governed by a single dominant mode. Similarly, sea surface height (SSH) exhibits a

strong first mode, with PC1 explaining 97.87% of the variance and the first 7 PCs capturing 99.85% in total (Figure 22). The wind components, however, exhibit a more distributed variance structure. The zonal wind component has PC1 accounting for 73.50% of the variance, while the first 7 PCs explain 97.83% (Figure 23), suggesting the presence of multiple contributing patterns beyond the dominant mode. Likewise, the meridional wind component shows a slightly stronger first mode, with PC1 explaining 79.49% of the variance and the cumulative variance of the first 7 PCs reaching 98.50% (Figure 23). These results highlight that SSH and MSLP are characterized by a dominant large-scale mode of variability, whereas wind components exhibit a more complex structure, requiring additional PCs to account for their variability.

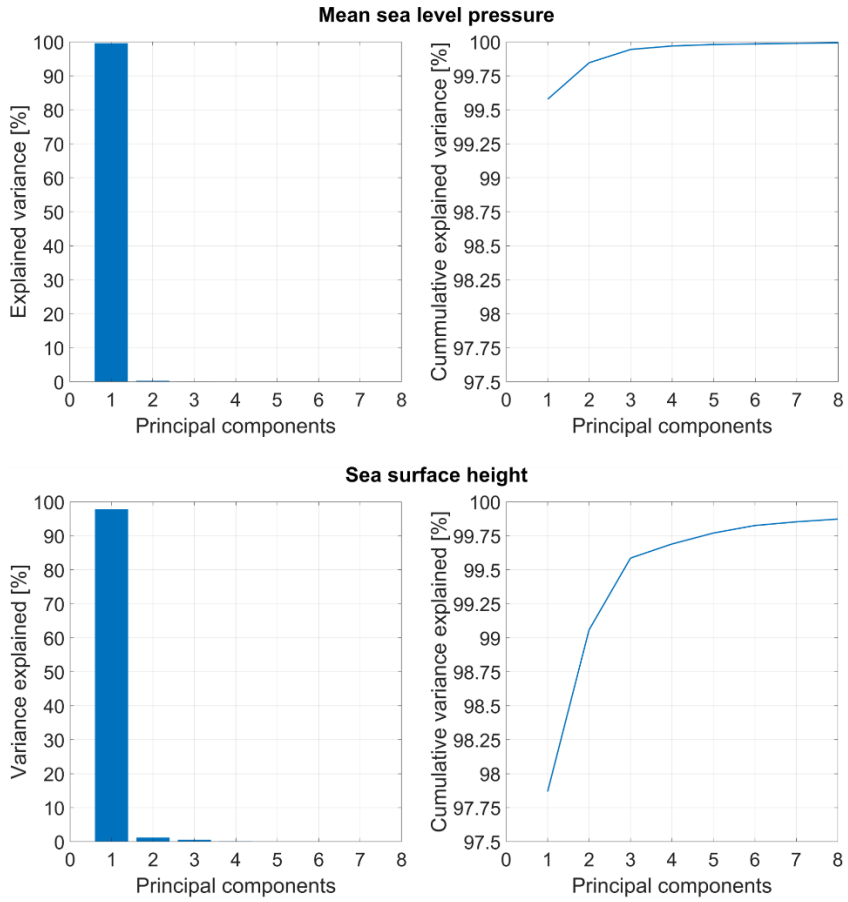

Figure 22: Explained variance for mean sea level pressure (upper panel) and sea surface height (lower panel).

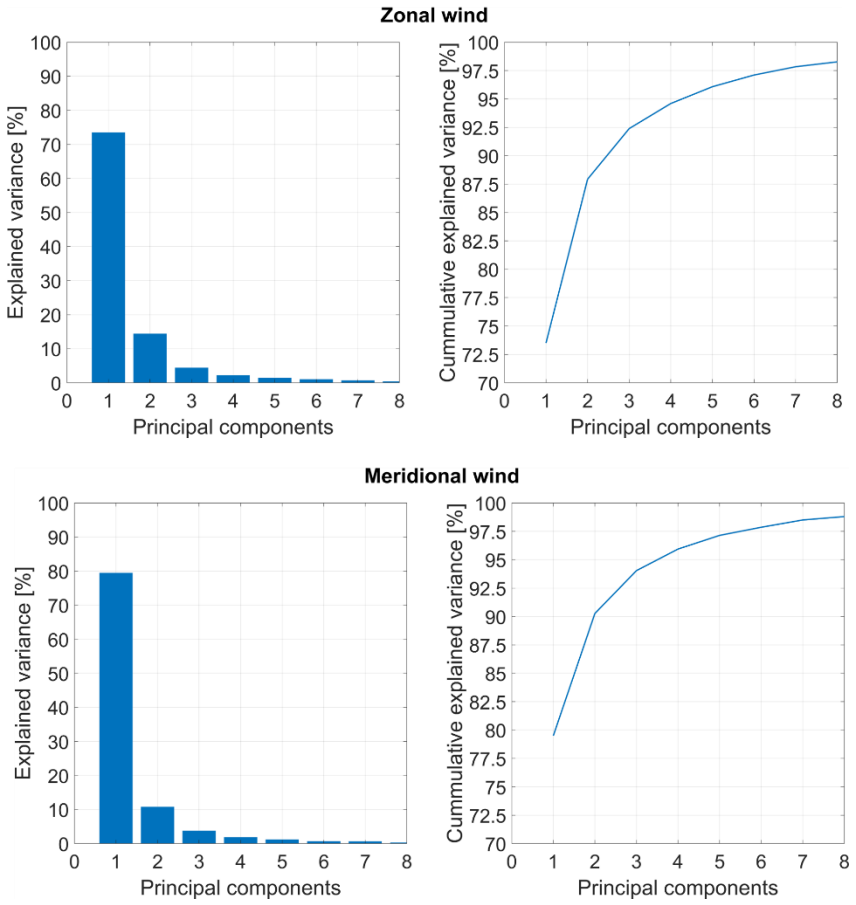

Figure 23: Explained variance for zonal (upper panel) and meridional (lower panel) wind components.

Figures 24 to 27 show the first seven Empirical Orthogonal Functions (EOFs) for each predictor. Before describing the results, it is important to clarify that these plots represent the spatial patterns of variability in each predictor field, based on the principal components, rather than the absolute values of the predictors themselves. EOFs capture how the values of a predictor vary spatially across the domain and indicate the relative magnitude of these variations in each mode. In essence, the EOFs illustrate how the predictor fields change across different regions, highlighting the dominant patterns of spatial variability.

The EOF analysis of SSH (Figure 24) indicates that its variability is largely dominated by a single spatial pattern: the first mode (EOF #1), which, as previously mentioned, explains 97.87% of the variance. This leading EOF exhibits a nearly uniform spatial structure across the domain. EOF #2 contributes an additional 1.19% of the variance

and presents a dipole-like pattern, suggesting the presence of more localized sea level variations potentially related to wind-driven coastal processes. The subsequent modes explain progressively smaller portions of the variance: EOF #3 accounts for 0.53%, EOF #4 for 0.10%, EOF #5 for 0.08%, EOF #6 for 0.05%, and EOF #7 for 0.03%. These higher-order modes display increasingly complex spatial structures, representing finer-scale variations that contribute to a more detailed representation of the variable.

The spatial patterns of the first seven EOFs for MSLP are shown in Figure 25. The first mode (EOF #1) explains 99.578% of the total variance, indicating that the dominant variability in MSLP is characterized by a highly uniform pattern across the domain. This suggests that the overall pressure field in the region is largely controlled by a single large-scale mode. The second mode (EOF #2) accounts for 0.263% of the variance, while the third (EOF #3) explains 0.098%. These lower-order modes exhibit spatial structures with weak gradients and localized variations, implying that they capture secondary and more localized atmospheric processes. The remaining modes (EOF #4 to EOF #7) contribute progressively smaller fractions of the variance, with their combined contribution being negligible.

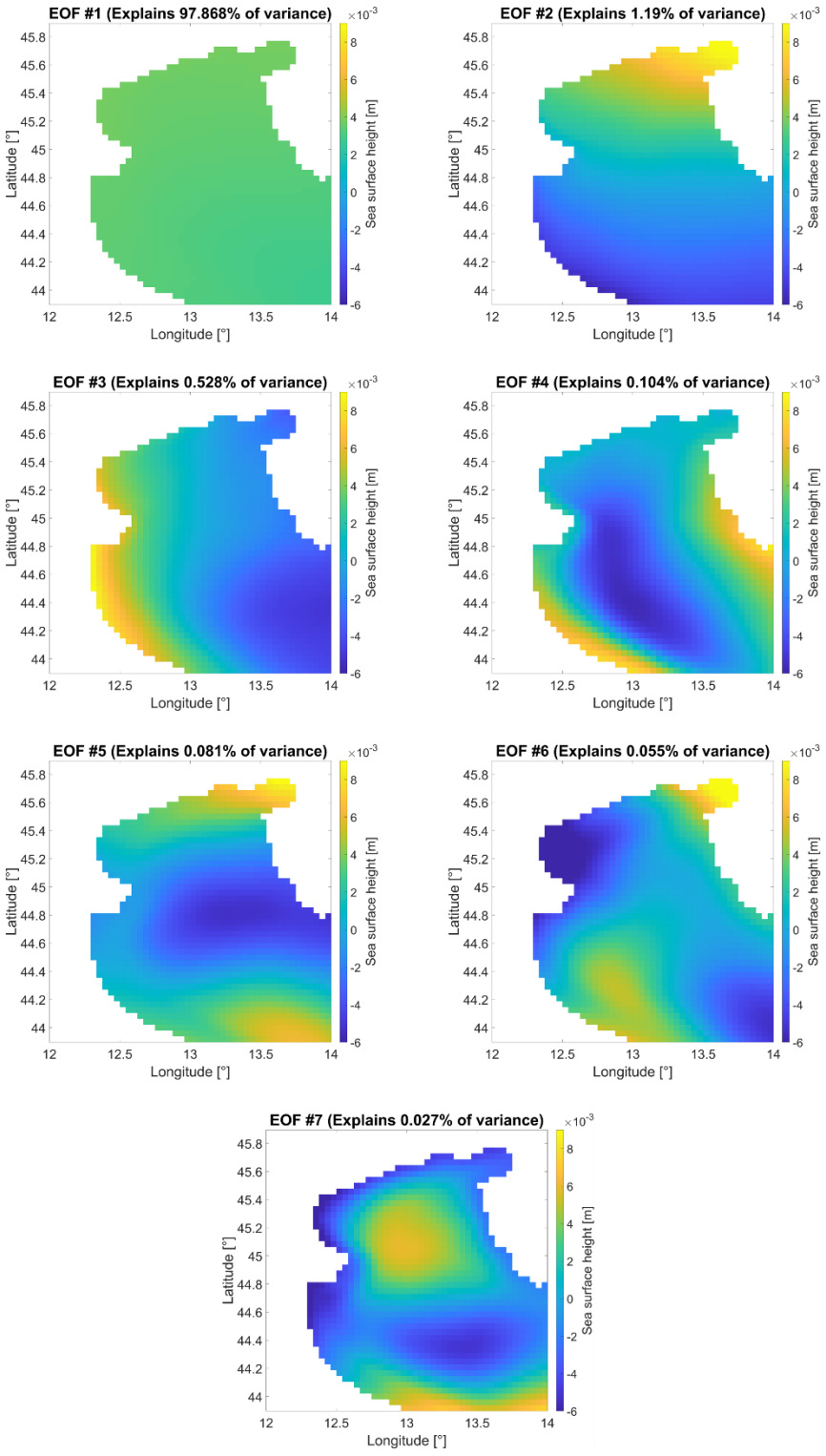

Figure 24: Empirical Orthogonal Functions (spatial patterns) for sea surface height.

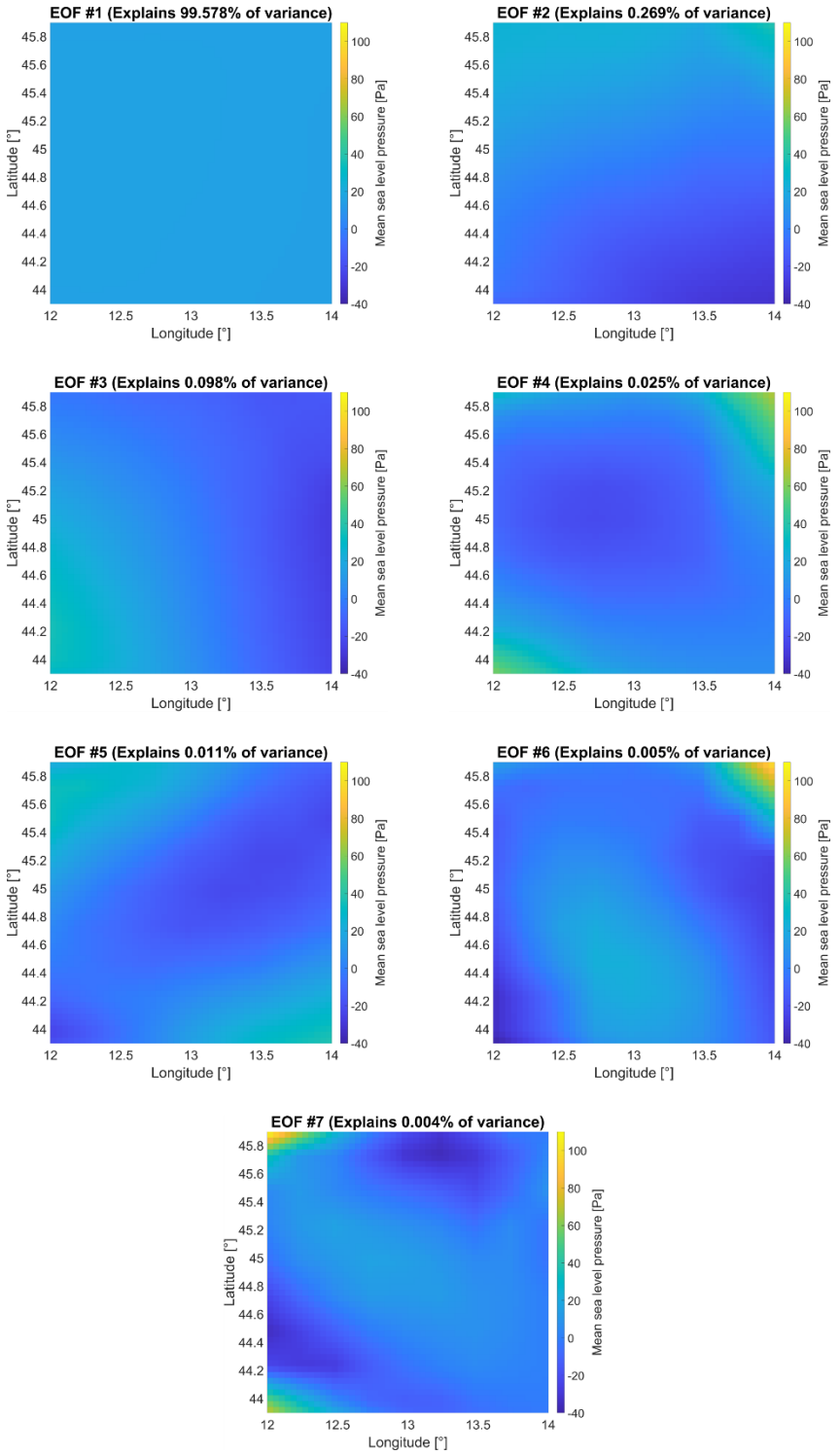

Figure 25: Empirical Orthogonal Functions (spatial patterns) for mean sea level pressure.

The spatial patterns of the first 7 EOFs derived from the zonal wind velocity (Figure 26) reveal the dominant modes of variability over the study area. The first EOF, which explains 73.50% of the total variance, is characterized by a broad north–south dipole structure, with positive anomalies over the central-southern region and negative anomalies in the north. The second mode accounts for 14.43% of the variance and displays a southeast–northwest gradient. EOFs 3 and 4, explaining 4.54% and 2.21% of the variance respectively, exhibit more localized structures, with patterns that may correspond to secondary modes. The remaining EOFs (5 to 7), each accounting for less than 1.5% of the total variance, capture increasingly finer-scale features and spatial complexity, likely reflecting less frequent or more transient wind events, and potentially noise or residual variability not represented by the dominant modes.

The leading EOFs of the meridional wind component (Figure 27) reveal the dominant spatial patterns associated with variability in the north-south wind velocity over the study area. EOF #1 explains 79.43% of the total variance and exhibits a broad-scale dipole pattern, with predominantly negative loadings in the southeast and positive loadings in the northwest, suggesting a coherent meridional wind structure along the basin's diagonal. EOF #2 accounts for 10.77% of the variance and is characterized by a meridional gradient with stronger positive anomalies in the northern sector and negative anomalies to the south, indicating fluctuations likely associated with latitudinal wind shifts. EOF #3 (3.76% variance) presents a clear southeast-northwest dipole, which may be linked to transient or localized wind events affecting the central-eastern part of the domain. EOF #4 (1.89% variance) shows a localized negative core around 45°N, 13.2°E surrounded by positive anomalies, hinting at more spatially confined wind variability. EOFs #5 through #7, each explaining less than 1.3% of the variance individually, display increasingly complex and noisier patterns, including localized cells and smaller-scale structures. These higher-order modes reflect less dominant, possibly more transient or stochastic wind fluctuations, and together account for a minor fraction of the total variability in the meridional wind field.

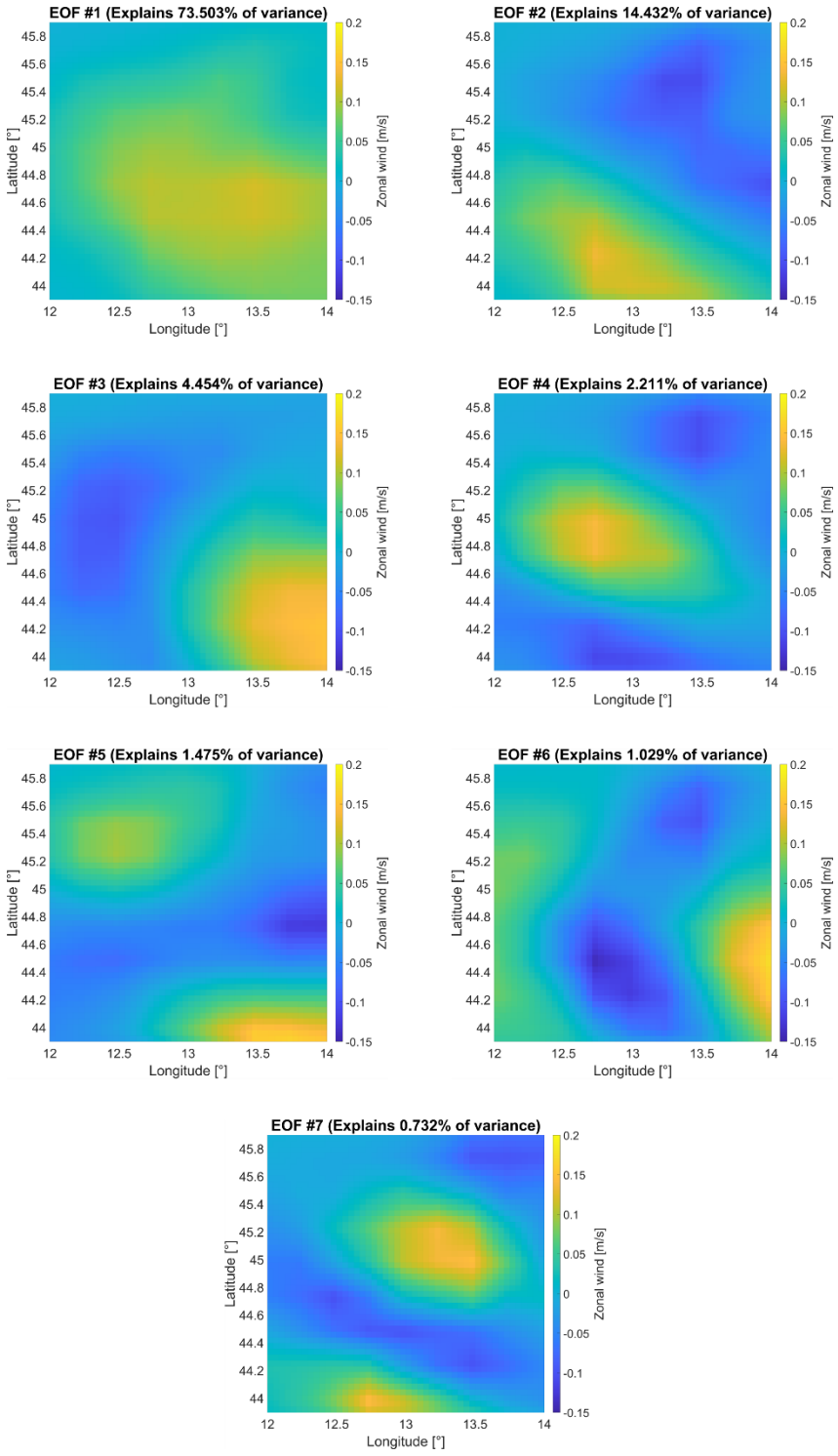

Figure 26: Empirical Orthogonal Functions (spatial patterns) for zonal wind.

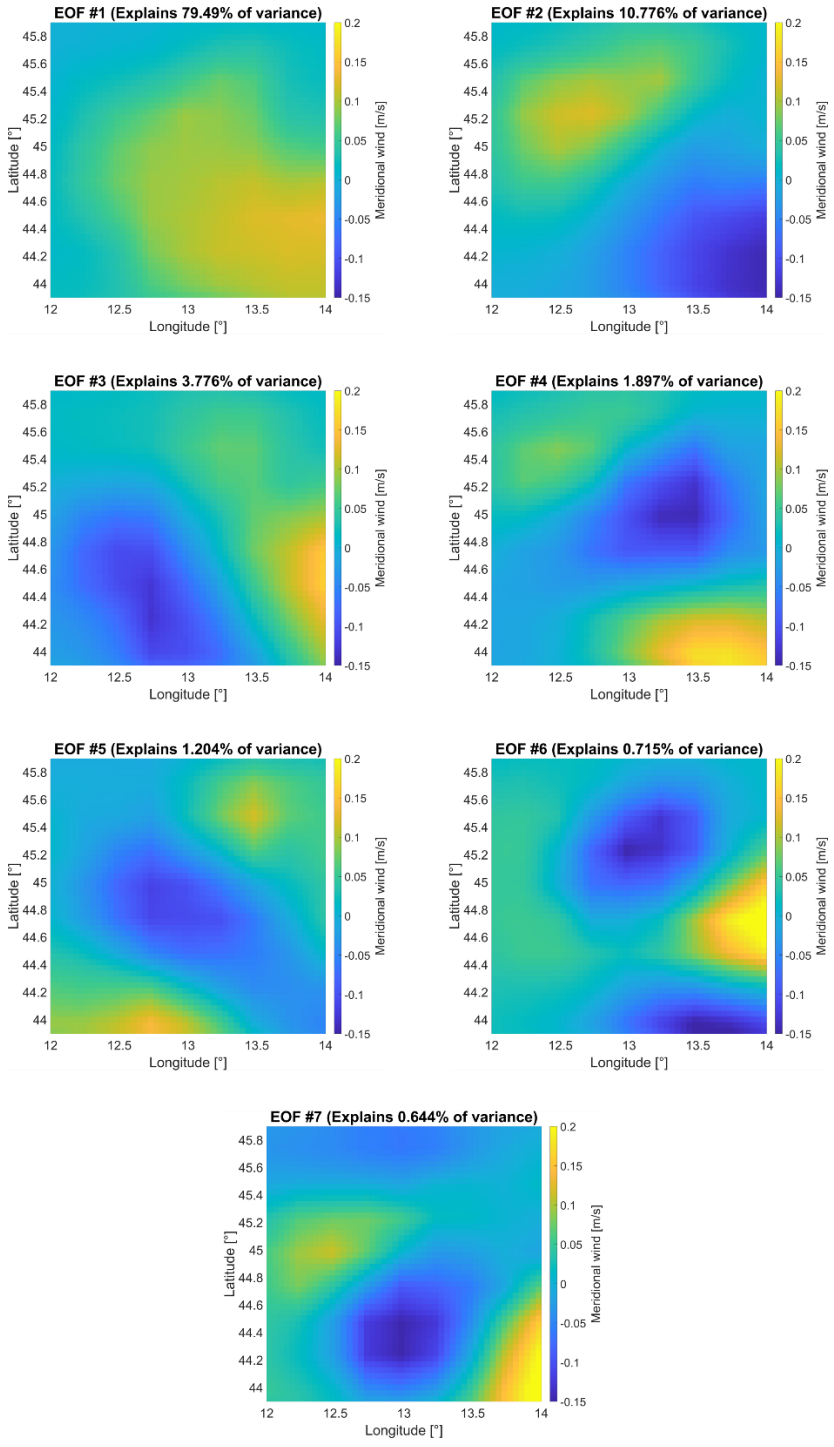

Figure 27: Empirical Orthogonal Functions (spatial patterns) for meridional wind.

3.2.2 PERFORMANCE OF MACHINE LEARNING MODELS USING BC-UNIGE FOR TRAINING AND TESTING

Table 7 presents the mean values of the performance metrics for the ML data-driven models, calculated as the average of each metric across all locations. In general, the MLR and MLP models perform reliably with lower complexity, achieving consistent accuracy in RMSE, MAD, and Pearson correlation. The MLR-MSE model achieves a high Pearson correlation coefficient (0.933) and a relatively low RMSE (0.048 m) and MAD (0.037 m), showing strong predictive accuracy with minimal error. However, the MADc (0.044 m) and MADp (0.007 m) suggest slight discrepancies in percentile predictions. MLR-MADc^2 improves on SLF (0.939) and MADp (0.005 m) compared to MLR-MSE, though it slightly increases the RMSE (0.052 m) and MAD (0.041 m). This behavior is illustrated in Figure 28, which presents the results for MLR models in Caorle and Punta della Salute.

The MLP-MSE model performs similarly to MLR-MSE, with a Pearson correlation of 0.931 and an RMSE of 0.049 m. The MADc remains low (0.045 m), and MADp (0.007 m) is consistent with the MLR models, indicating competitive performance in overall error metrics. MLP-MADc^2 improves SLF (0.934) while slightly reducing percentile deviations (MADp of 0.005 m) and MADc (0.044 m), suggesting that focusing on minimizing corrected deviations benefits percentile accuracy without a significant impact on RMSE.

Related to the more sophisticated models, RNN and LSTM models provide stronger alignment with predictand extreme values, especially in their MADc^2 variants, where percentile deviations are minimized. In terms of the mean values obtained, the LSTMs-MADc^2 model emerges as the most robust, demonstrating the highest SLF, Pearson correlation, and minimized error metrics, making it the preferred model for balancing accuracy with alignment across all evaluation criteria.

RNNs-MSE and RNNh-MSE show comparable performance to MLR and MLP, achieving Pearson correlation coefficients above 0.93 and low RMSE values (0.048 m). However, the MADp metric (0.008 m) is higher than that of the MLR and MLP models, indicating that while RNN models capture the general trend well, they show slightly greater variability in extreme values. RNNs-MADc^2 and RNNh-MADc^2 variants focus on corrected deviation (MADc), which is effectively minimized to 0.045 m. This correction reduces variability in percentile error (MADp of 0.005 m) without sacrificing RMSE accuracy, indicating a robust performance in percentile alignment.

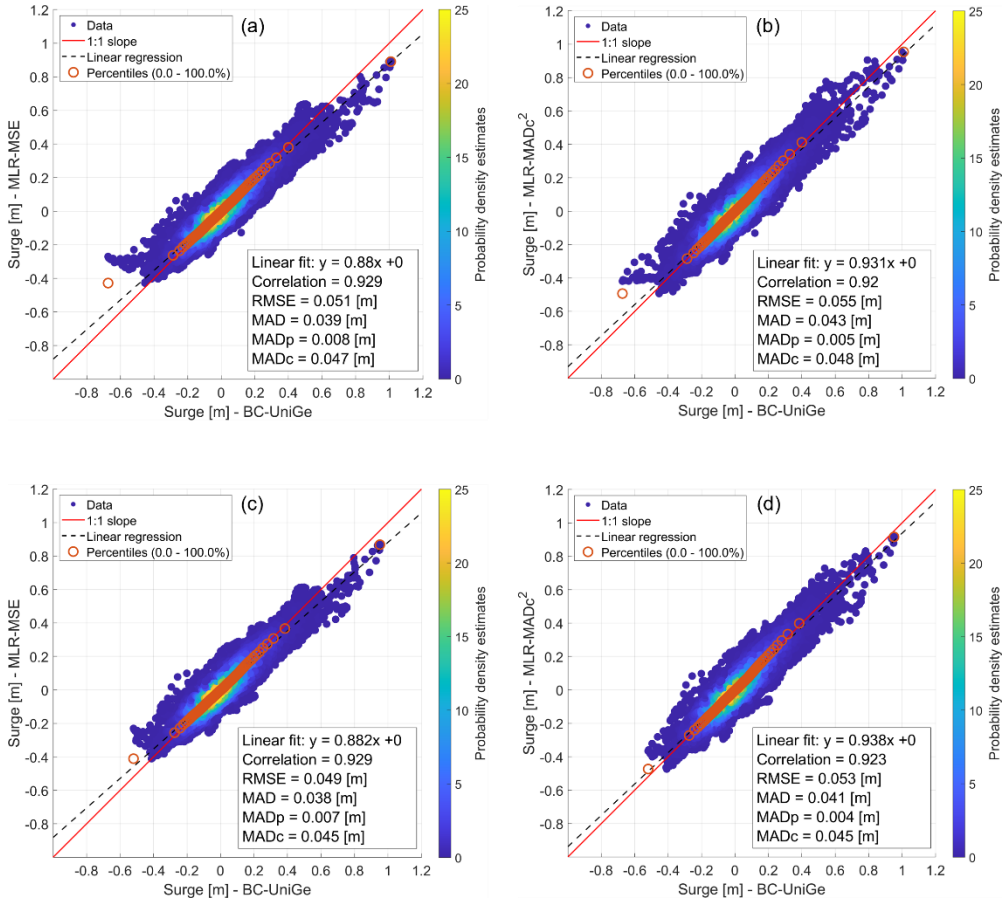

Figure 28: Scatter plots of ML downscaling in Caorle and Punta della Salute, using BC-UniGe for training and testing. (a) and (b) MLR-MSE and MLR- MADc² in Caorle, respectively; (c) and (d) MLR-MSE and MLR-MADc² in Punta della Salute, respectively.

LSTMs-MSE and LSTMh-MSE models have slightly higher RMSE (0.053 m and 0.054 m, respectively) and lower Pearson correlation coefficients (0.922 and 0.917), suggesting slightly reduced predictive capabilities. As mentioned, the LSTMs-MADc² variant demonstrates the best overall SLF (0.940) and a high Pearson correlation (0.934), with a low RMSE (0.049 m) and minimal MADc (0.043 m), highlighting its effective error reduction when using the MADc² metric as loss function. This suggests LSTMs-MADc²'s improved alignment with predictand values across extreme percentiles and corrected deviations. LSTMh-MADc² slightly increases RMSE to 0.050 m but achieves a robust correlation of 0.929. The MADp and MADc values (0.006 m and 0.045 m) indicate balanced accuracy across all error metrics, showing that the hybrid structure enhances model performance in both standard and corrected metrics.

Table 7: Mean values of the performance metrics for the implemented ML models, trained and tested using BC-UniGe data.

		Metric					
		SLF	Corr	RMSE m	MAD m	MADp m	MADc m
ML model	MLR-MSE	0.889	0.933	0.048	0.037	0.007	0.044
	MLR-MADc^2	0.939	0.925	0.052	0.041	0.005	0.045
	MLP-MSE	0.900	0.931	0.049	0.038	0.007	0.045
	MLP-MADc^2	0.934	0.927	0.051	0.039	0.005	0.044
	RNNs-MSE	0.902	0.932	0.049	0.039	0.008	0.046
	RNNs-MADc^2	0.936	0.928	0.051	0.039	0.005	0.045
	RNNh-MSE	0.904	0.934	0.048	0.037	0.008	0.045
	RNNh-MADc^2	0.937	0.928	0.051	0.040	0.005	0.045
	LSTMs-MSE	0.913	0.922	0.053	0.041	0.005	0.046
	LSTMs-MADc^2	0.940	0.934	0.049	0.038	0.005	0.043
	LSTMh-MSE	0.922	0.917	0.054	0.043	0.005	0.048
	LSTMh-MADc^2	0.935	0.929	0.050	0.039	0.006	0.045

Figure 29 shows scatter plots for MLP, RNN, and LSTM models in their MSE and MADc^2 variants for Grado, a location with a MADc score similar to the mean values shown in Table 7. This location is used to exemplify the performance of the ML models at a specific site. The implementation of MADc^2 as the loss function in the MLP model improves performance across all metrics except for the Pearson correlation, leading to a better representation of extreme values. The RNN-MADc^2 variant does not show significant changes compared to its MSE counterpart, but a slight improvement in MADc is observed. For the LSTM-MADc^2 model, all metrics are improved except for MADp. In Grado, this model does not demonstrate a significant improvement in extreme values, but it exhibits less data dispersion, contributing to better results for most error metrics.

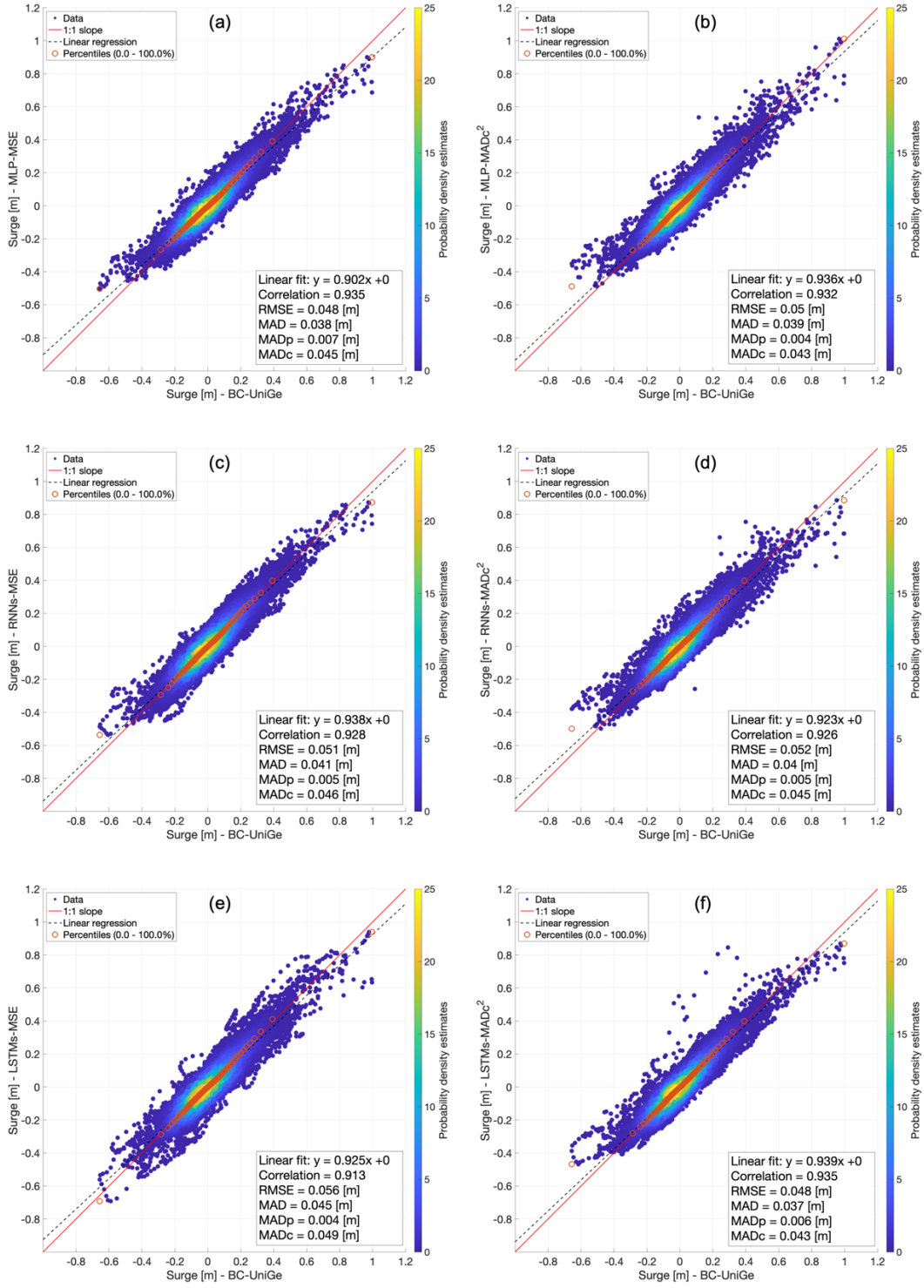

Figure 29: Scatter plots of ML downscaling in Grado, using BC-UniGe for training and testing. (a) MLP-MSE, (b) MLP-MADc^2, (c) RNNs-MSE, (d) RNNs-MADc^2, (e) LSTMs-MSE, and (f) LSTMs-MADc^2.

To evaluate the impact of the different model configurations among the extreme values, Table 8 shows the mean performance metrics for the ML models emphasizing their behavior on data above the 99th percentile. In addition to metrics from Table 7, Table 8 includes bias, where positive values indicate overestimations and negative values indicate underestimations by the models.

The MLR-MSE model shows underestimation bias of -0.045 m, the highest in magnitude among the models. Although the RMSE is relatively high (0.084 m), it maintains a moderate Pearson correlation (0.820). MADp (0.046 m) and MADc (0.111 m) indicate variability in extreme predictions, suggesting the MLR-MSE model may struggle with peak value alignment. MLR-MADc^2 significantly reduces the bias to -0.014 m, indicating near-zero systematic error for the upper extremes. Additionally, it improves the slope fit (SLF of 0.877) and maintains a similar RMSE (0.080 m) while reducing MADp (0.021 m) and MADc (0.085 m).

The MLP-MSE model reduces bias to -0.025 m, with an SLF of 0.871 and a Pearson correlation of 0.836. Its RMSE (0.075 m) is among the lowest, and the MADc (0.086 m) shows relatively high precision for extreme values, making this a strong candidate for percentile-based accuracy. MLP-MADc^2 further minimizes bias to -0.015 m, suggesting a high accuracy in terms of both underestimations and overestimations. With an SLF of 0.872 and an RMSE of 0.075 m, the model balances error metrics well. MADp and MADc metrics are the lowest among MLP variants (0.021 m and 0.078 m), indicating that MLP-MADc^2 is particularly adept at capturing extreme value trends with minimal variability.

RNNs-MSE performs with a balanced SLF of 0.840 and Pearson correlation of 0.844. However, a slightly elevated MADp (0.029 m) and MADc (0.086 m) suggest there may be variability in its high-percentile predictions. RNNs-MADc^2 further reduces bias to -0.017 m, though it shows a slight decrease in Pearson correlation (0.781) and increase in RMSE (0.078 m). The corrected deviation metrics (MADc of 0.083 m and MADp of 0.022 m) are improved, suggesting that while percentile precision is enhanced, general trend alignment may be slightly compromised. RNNh-MSE and RNNh-MADc^2 maintain low biases (-0.026 m and -0.014 m, respectively). The RNNh-MADc^2 model offers an improved MADc (0.079 m), indicating reduced corrected deviation variability, making it a balanced model for extreme event predictions.

LSTMs-MSE achieves a SLF value of 0.893 and a Pearson correlation of 0.799. However, its RMSE (0.087 m) is relatively high, and its bias (-0.026 m) shows slight underestimation tendencies. The MADc metric (0.098 m) is the highest among all models, suggesting more variability in percentile alignment. LSTMs-MADc^2 significantly improves

percentile accuracy, showing minimal bias (-0.016 m), a robust SLF (0.877), and a low MADp (0.020 m). With an RMSE of 0.073 m, this model effectively balances high accuracy for extreme values with minimal deviation, making it the most stable among LSTM models. LSTMh-MSE shows the highest SLF (0.964) among all models, but its RMSE (0.085 m) and MADc (0.092 m) are comparatively high. This indicates strong trend alignment but higher error variability at extreme values. LSTMh-MADc^2 achieves minimal bias (-0.019 m) with improved percentile precision (MADp of 0.025 m) and a reduced MADc (0.083 m).

Table 8: Mean values of the performance metrics for values above the 99th percentile for the implemented ML models, trained and tested using BC-UniGe data.

		Metric						
		SLF	Corr	RMSE m	Bias m	MAD m	MADp m	MADc m
ML model	MLR-MSE	0.752	0.820	0.084	-0.045	0.065	0.046	0.111
	MLR-MADc^2	0.877	0.807	0.080	-0.014	0.064	0.021	0.085
	MLP-MSE	0.871	0.836	0.075	-0.025	0.058	0.027	0.086
	MLP-MADc^2	0.872	0.826	0.075	-0.015	0.057	0.021	0.078
	RNNs-MSE	0.840	0.844	0.073	-0.027	0.057	0.029	0.086
	RNNs-MADc^2	0.782	0.795	0.079	-0.020	0.061	0.024	0.085
	RNNh-MSE	0.825	0.843	0.072	-0.026	0.055	0.028	0.083
	RNNh-MADc^2	0.829	0.810	0.077	-0.014	0.059	0.020	0.079
	LSTMs-MSE	0.893	0.799	0.087	-0.026	0.067	0.031	0.098
	LSTMs-MADc^2	0.877	0.835	0.073	-0.016	0.056	0.020	0.076
	LSTMh-MSE	0.964	0.816	0.085	-0.017	0.065	0.028	0.092
	LSTMh-MADc^2	0.838	0.818	0.077	-0.019	0.059	0.025	0.083

The results presented in Table 8 are graphically illustrated in Figure 30, which displays diagrams of the mean values for bias, MADp, and MADc for surges above the 99th percentile. It is evident that the MLR-MSE model achieves the lowest scores in representing extreme surges. However, with the implementation of MADc^2 as the loss function, the MLR model shows a considerable improvement, achieving performance levels comparable to more sophisticated models. The implementation of MADc^2 results in improvements in the representation of extremes across most models for the metrics included in Figure 30. The only model that exhibits a decrease in performance for surge values above the 99th percentile is LSTMh-MADc^2, which shows a decline in bias.

Figure 30: Diagrams of the mean values obtained for bias, MADp, and MADc for surges above the 99th percentile, for the implemented ML models, trained and tested using BC-UniGe data.

Figure 31 illustrates the percentage variation in performance metrics when MADc^2 is used as the loss function, for surges above the 99th percentile. For MADp and MADc, improvements were observed across all models, with increases of 54% for MLR, 34.9% for LSTMs, and 29.9% for RNNh in MADp, and 23.8% for MLR and 22.1% for LSTMs in MADc. As previously mentioned, LSTMh showed a performance decrease in terms of bias with the implementation of MADc^2, quantified at -12.7%. For the other models, only improvements in bias were recorded, with notable increases of 70% for MLR, 47.1% for RNNh, and 42% for MLP. The remaining metrics do not exhibit a consistent pattern of improvement or decline. Overall, the LSTM models showed improvements in Pearson correlation, RMSE, and MAD, whereas the RNN models experienced performance declines in these metrics. No significant changes were observed for MLR and MLP models in the mentioned metrics.

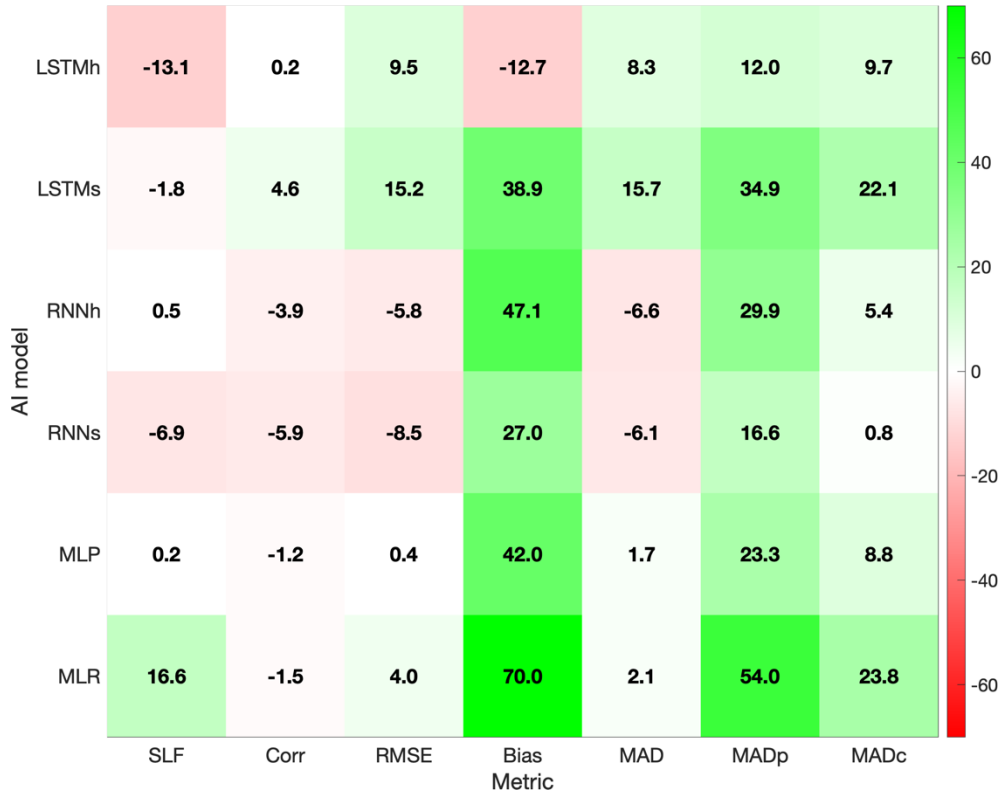

Figure 31: Percentage variation plot of performance metrics for surges above the 99th percentile when MADc² is used as the loss function in the implemented ML models, using BC-UniGe for training and testing. Positive values indicate improvements, while negative values indicate a decrease in performance.

3.2.3 PERFORMANCE OF MACHINE LEARNING MODELS USING BC-UNIGE FOR TRAINING AND OBSERVATIONS FOR TESTING

As a first approach, the models were trained using BC-UniGe output, with the testing period based on observed surge data. Table 9 shows the mean values obtained for each metric and model in this first approach, including the performance of the BC-UniGe dynamical downscaling for comparison, which serves as the baseline reference in this section. Additionally, to visually support the description of the results, Figures 32 and 33 present scatter plots comparing the performance of BC-UniGe and the MLR, MLP, RNNh, and LSTMh models in their MSE and MADc² variants, respectively, against observations.

In terms of mean values, BC-UniGe exhibits an SLF of 0.868, indicating a strong linear relationship with observed values. The Pearson correlation is also high at 0.846, suggesting that BC-UniGe produces reliable estimates. The RMSE of 0.078 m is relatively low, which indicates that the model's downscaling is accurate. Additionally, the MAD is 0.060 m, and the MADc is 0.079 m, further highlighting BC-UniGe's overall strong performance.

The performance of MLR models, particularly MLR-MSE, is comparable to that of BC-UniGe. The SLF of 0.803 for MLR-MSE is slightly lower than that of BC-UniGe, indicating a weaker linear fit. However, the Pearson correlation of 0.835 is close to BC-UniGe's value, suggesting that the predictions are well-aligned with the observed data. The RMSE of 0.076 m is similarly close to the baseline, indicating that the MLR model is quite accurate in terms of prediction. The MAD of 0.059 m is very close to that of BC-UniGe, and the MADc of 0.074 m is also slightly lower than BC-UniGe's value. Additionally, MADp of 0.015 m is slightly lower than the corresponding value for BC-UniGe, suggesting that the MLR model performs well in capturing variations at different percentiles of the data. The MLR-MADc^2 model shows a slightly improved SLF of 0.860, making it closer to BC-UniGe's performance. The Pearson correlation remains at a similar level (0.834), and the RMSE is virtually identical to that of MLR-MSE at 0.079 m. The MAD increases slightly to 0.062 m, which is still comparable to the baseline. In terms of the MADc, MLR-MADc^2 performs similarly to MLR-MSE, with a value of 0.076 m. Specifically at the CNR platform, MLR models (Figure 32b and Figure 32c) show lower performance than BC-UniGe in terms of SLF, Pearson correlation, RMSE, and MAD, while achieving comparable performance in terms of MADc and better performance for MADp.

The MLP models show similar performance to the MLR models but with some differences. The SLF for MLP-MSE is 0.815, which is lower than the SLF of BC-UniGe, indicating a slightly weaker linear fit. However, the Pearson correlation of 0.829 is close to BC-UniGe's value, and the RMSE of 0.078 m is identical. The MAD of 0.061 m is very close to that of BC-UniGe. The MADc of 0.075 m is slightly lower, and MADp of 0.014 m is better than BC-UniGe's value, suggesting that the MLP model is more adept at capturing variations across different percentiles. The MLP-MADc^2 model shows a slightly improved SLF of 0.845, which is still lower than BC-UniGe's performance but represents a slight improvement over MLP-MSE. The Pearson correlation of 0.827 remains quite similar to that of MLP-MSE. The RMSE of 0.080 m and MAD of 0.062 m are slightly worse than MLP-MSE, indicating a small decrease in accuracy. MADp and MADc are similar to MLP-MSE, indicating that the MLP models

perform consistently across different metrics, as can be seen for the CNR platform in Figure 32d and Figure 32e.

The RNN models perform similarly to the MLP models, but with some differences. For RNNs-MSE, the SLF is 0.813, which is lower than both BC-UniGe and the MLP models. However, the Pearson correlation of 0.827 is close to that of BC-UniGe, and the RMSE of 0.078 m is nearly identical. The MAD of 0.061 m is comparable to BC-UniGe, and the MADp of 0.013 m is lower than BC-UniGe's value, suggesting that the RNN model is more accurate in this regard. The MADc of 0.074 m is also similar to BC-UniGe, indicating that the model's performance is relatively strong. The RNNs-MADc^2 model shows an improved SLF of 0.841, which is higher than RNNs-MSE, but still lower than BC-UniGe. The Pearson correlation of 0.826 is slightly lower than RNNs-MSE, and the RMSE of 0.080 m and MAD of 0.063 m are slightly worse. The MADp is 0.015 m, which is slightly worse than RNNs-MSE, and the MADc of 0.078 m is higher than RNNs-MSE, indicating a slight decrease in performance in this version of the RNN model. The results described are similar to those obtained for the RNNh models, with slights differences in performance in terms of Pearson correlation (0.831 for RNNh-MSE and 0.825 for RNNh-MADc2), MADp (0.015 for RNNh-MSE and 0.014 for RNNh-MADc2), and MADc (0.075 for RNNh-MSE and 0.077for RNNh-MADc2). In the CNR platform the RNNh models (b and Figure 26c) obtained comparable performance to BC-UniGe. However, RNNh models underperform BC-UniGe in most of the metrics, except for MADp in which BC-UniGe gets a value of 0.01, while RNNh-MADc2 obtained a value of 0.007, showing the positive impact of the implementation of the MADc2 loss function in the proper representation of the percentile's distributions in certain locations.

Finally, the LSTM models show similar performance to the RNN models but with some variations. The SLF of LSTMs-MSE is 0.824, which is lower than BC-UniGe, and the Pearson correlation of 0.821 is also slightly lower. The RMSE of 0.081 m is worse than BC-UniGe, indicating a decrease in prediction accuracy. The MAD of 0.063 m is slightly worse than BC-UniGe, while the MADp of 0.015 m is similar. The MADc of 0.078 m is also comparable to BC-UniGe's performance. The LSTMs-MADc^2 model shows a higher SLF of 0.852, which brings it closer to BC-UniGe's performance. The Pearson correlation increases to 0.832, and the RMSE improves slightly to 0.079 m. The MAD improves to 0.061 m, indicating better overall performance compared to LSTMs-MSE. The MADp and MADc remain similar to LSTMs-MSE, further confirming that the LSTM models can capture the trends in the data, though they still show some room for improvement in terms of accuracy.

The LSTMh models exhibit a mixed performance compared to both, the other ML models and the BC-UniGe results. When trained using MSE as the loss function (LSTMh-MSE), the model achieves a SLF of 0.837, which is higher than most other ML models except for LSTMs-MADc^2 (0.852). However, its Pearson correlation coefficient (0.818) is among the lowest, indicating weaker agreement with observed data. The RMSE is 0.082 m, slightly higher than BC-UniGe (0.078 m) and most other models, suggesting a marginally higher overall error in predictions. For LSTMh-MADc^2, the performance improves slightly for some metrics but remains consistent overall. This is evident in the results obtained for the CNR platform (Figure 33e), where the implementation of MADc^2 improves all metrics, even compared to the other ML models. The SLF and Pearson correlation coefficient are 0.837 and 0.825, respectively. The RMSE improves slightly to 0.080 m, closer to the BC-UniGe value. The MAD metrics (MAD, MADp, MADc) are comparable to other ML models trained with MADc^2, with values of 0.062 m, 0.015 m, and 0.077 m, respectively.

Table 9: Mean values of the performance metrics for BC-UniGe and the ML models trained on BC-UniGe and testing on observations.

		Metric					
		SLF	Corr	RMSE m	MAD m	MADp m	MADc m
	BC-UniGe	**0.868**	**0.846**	**0.078**	**0.060**	**0.019**	**0.079**
ML model	MLR-MSE	0.803	0.835	0.076	0.059	0.015	0.074
	MLR-MADc^2	0.860	0.834	0.079	0.062	0.015	0.076
	MLP-MSE	0.815	0.829	0.078	0.061	0.014	0.075
	MLP-MADc^2	0.845	0.827	0.080	0.062	0.014	0.076
	RNNs-MSE	0.813	0.827	0.078	0.061	0.013	0.074
	RNNs-MADc^2	0.841	0.826	0.080	0.063	0.015	0.078
	RNNh-MSE	0.814	0.831	0.078	0.061	0.015	0.075
	RNNh-MADc^2	0.841	0.825	0.080	0.063	0.014	0.077
	LSTMs-MSE	0.824	0.821	0.081	0.063	0.015	0.078
	LSTMs-MADc^2	0.852	0.832	0.079	0.061	0.015	0.076
	LSTMh-MSE	0.837	0.818	0.082	0.064	0.013	0.077
	LSTMh-MADc^2	0.837	0.825	0.080	0.062	0.015	0.077

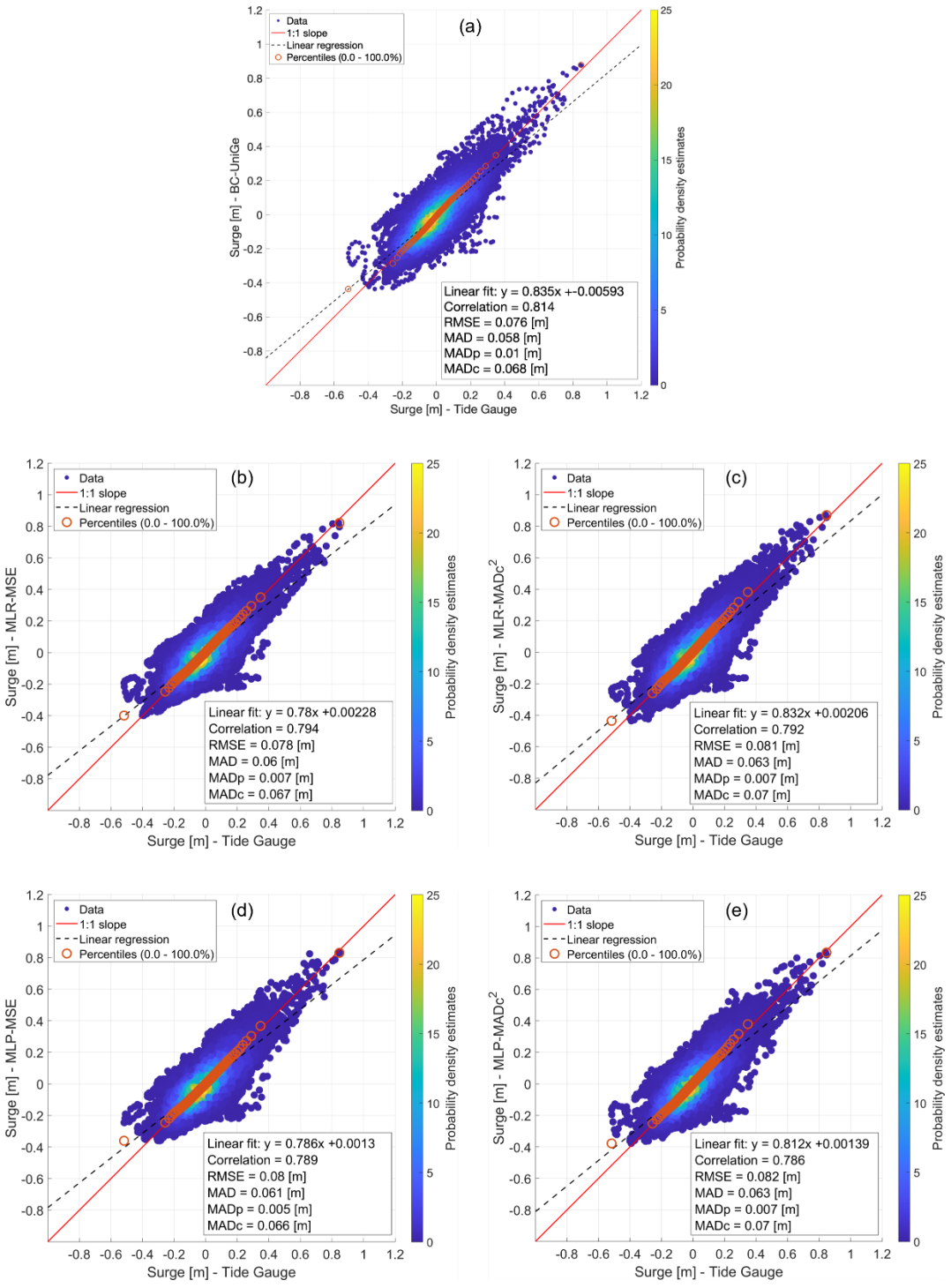

Figure 32: Scatter plots between observations, dynamical downscaling, and MLR and MLP models in the CNR platform, using BC-UniGe for training and observations for testing. (a) BC-UniGe, (b) MLR-MSE, (c) MLR-MADc^2, (d) MLP-MSE, and (e) MLP-MADc^2.

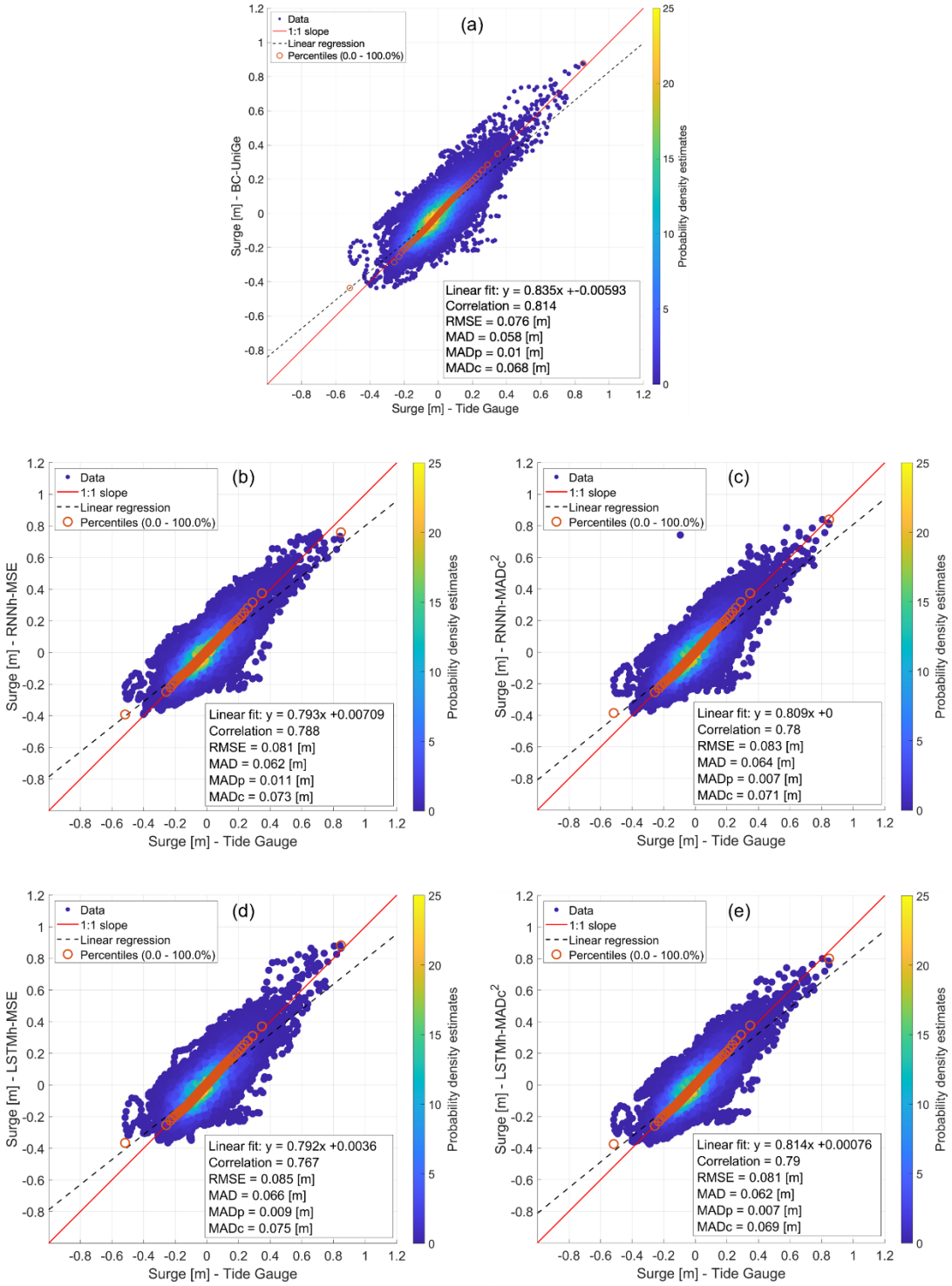

Figure 33: Scatter plots between observations, dynamical downscaling, and RNNh and LSTMh models in the CNR platform, using BC-UniGe for training and observations for testing. (a) BC-UniGe, (b) RNNh-MSE, (c) RNNh-MADc^2, (d) LSTMh-MSE, and (e) LSTMh-MADc^2.

Considering values above the 99th percentile, Table 10 shows that BC-UniGe remains as the benchmark in terms of SLF and Pearson correlation, but some ML models, particularly those trained with MADc^2, demonstrate comparable or even superior performance in terms of RMSE and MAD metrics. The dynamical downscaling approach, BC-UniGe, achieves a SLF of 1.149, the highest among all models, and a Pearson correlation of 0.799, demonstrating strong agreement with observed data. The RMSE for BC-UniGe is 0.109 m, slightly higher than some ML models, but its bias of -0.029 m and MAD metrics (0.090 m for MAD, 0.056 m for MADp, and 0.145 m for MADc) highlight its capability to minimize error variability across the dataset.

Among the ML models, MLR-MSE achieves a comparable RMSE of 0.107 m but underperforms in SLF (0.894) and Pearson correlation (0.767). It exhibits a relatively high bias (-0.061 m) and larger MADp and MADc values (0.063 m and 0.152 m, respectively), indicating less consistency in capturing surge variability. MLR-MADc^2 demonstrates significant improvement in SLF (1.062) and bias reduction (-0.017 m). Its RMSE of 0.101 m is lower than BC-UniGe, and its MAD metrics (0.081 m for MAD, 0.047 m for MADp, and 0.129 m for MADc) suggest better overall accuracy and variability reduction compared to MLR-MSE.

MLP-MSE achieves an SLF of 1.021 and a low RMSE of 0.105 m, but its Pearson correlation (0.768) and bias (-0.037 m) remain slightly below BC-UniGe. Conversely, MLP-MADc^2 reduces the MAD metrics further (0.083 m for MAD, 0.045 m for MADp, and 0.128 m for) while maintaining a similar RMSE and bias performance.

For RNN-based models, both RNNs-MSE and RNNs-MADc^2 perform relatively consistently. RNNs-MSE achieves an SLF of 0.967, a Pearson correlation of 0.760, and an RMSE of 0.105 m, with moderate bias (-0.034 m). RNNs-MADc^2 reduces the RMSE slightly to 0.102 m and shows lower MAD metrics (MADp of 0.046 m and MADc of 0.127 m) indicating that the MADc^2 loss improves variability capture while maintaining comparable accuracy.

The RNNh models follow a similar trend, with RNNh-MSE achieving an SLF of 0.967 and slightly better Pearson correlation (0.774) and RMSE (0.102 m) compared to RNNs-MSE. However, the bias remains slightly higher (-0.041 m). RNNh-MADc^2 balances the metrics with an SLF of 0.987, a Pearson correlation of 0.761, and improved MADc value (0.133 m), showing the benefits of the MADc^2 loss function.

The LSTM models display mixed results. LSTMs-MSE achieves a high SLF (1.024) but exhibits the lowest Pearson correlation (0.748) among the models, with a higher RMSE of 0.112 m. On the other hand, LSTMs-MADc^2 achieves a more balanced

performance, with an SLF of 1.012, an improved Pearson correlation of 0.760, and lower RMSE (0.105 m). Its MAD (0.082 m), MADp (0.046 m), and MADc (0.127 m) reflect reduced error variability compared to its MSE counterpart.

The LSTMh models also show variability depending on the training loss function. LSTMh-MSE achieves a high SLF (1.059) but has the lowest Pearson correlation (0.738) and the highest RMSE (0.119 m) among all models. Its MAD metrics (MAD of 0.095 m, MADp of 0.057 m, and MADc of 0.152 m) indicate greater variability in predictions. Conversely, LSTMh-MADc^2 reduces the RMSE to 0.106 m and achieves better MAD metrics (MAD of 0.085 m, MADp of 0.050 m, and MADc of 0.136 m), although its correlation (0.757) remains below BC-UniGe and other models.

Table 10: Mean values of the performance metrics for values above the 99th percentile for the implemented ML models, using BC-UniGe for training and observations for testing.

		Metric						
		SLF	Corr	RMSE m	Bias m	MAD m	MADp m	MADc m
	BC-UniGe	**1.149**	**0.799**	**0.109**	**-0.029**	**0.090**	**0.056**	**0.145**
ML model	MLR-MSE	0.894	0.767	0.107	-0.061	0.089	0.063	0.152
	MLR-MADc^2	1.062	0.781	0.101	-0.017	0.081	0.047	0.129
	MLP-MSE	1.021	0.768	0.105	-0.037	0.085	0.047	0.132
	MLP-MADc^2	1.014	0.760	0.105	-0.028	0.083	0.045	0.128
	RNNs-MSE	0.967	0.760	0.105	-0.034	0.084	0.051	0.135
	RNNs-MADc^2	0.932	0.751	0.102	-0.036	0.082	0.046	0.127
	RNNh-MSE	0.967	0.774	0.102	-0.041	0.082	0.050	0.132
	RNNh-MADc^2	0.987	0.761	0.104	-0.029	0.084	0.049	0.133
	LSTMs-MSE	1.024	0.748	0.112	-0.037	0.090	0.050	0.139
	LSTMs-MADc^2	1.012	0.760	0.105	-0.020	0.082	0.046	0.127
	LSTMh-MSE	1.059	0.738	0.119	-0.020	0.095	0.057	0.152
	LSTMh-MADc^2	0.966	0.757	0.106	-0.037	0.085	0.050	0.136

Figure 34 presents the mean values for bias, MADp, and MADc obtained across the different model variants. It highlights clear improvements with the implementation of MADc^2 as the loss function, except for RNNs and LSTMh in terms of bias, and RNNh in terms of MADc. The use of this customized loss function enables the models to even outperform BC-UniGe in representing extreme surges. A clear demonstration of this is seen in the results for MADp and MADc, where all the ML models with the MADc^2 variant achieve better scores than BC-UniGe. However, in terms of Bias, only MLR-

MADc^2 and LSTMS-MADc^2 outperform BC-UniGe, while the remaining models obtain comparable performance.

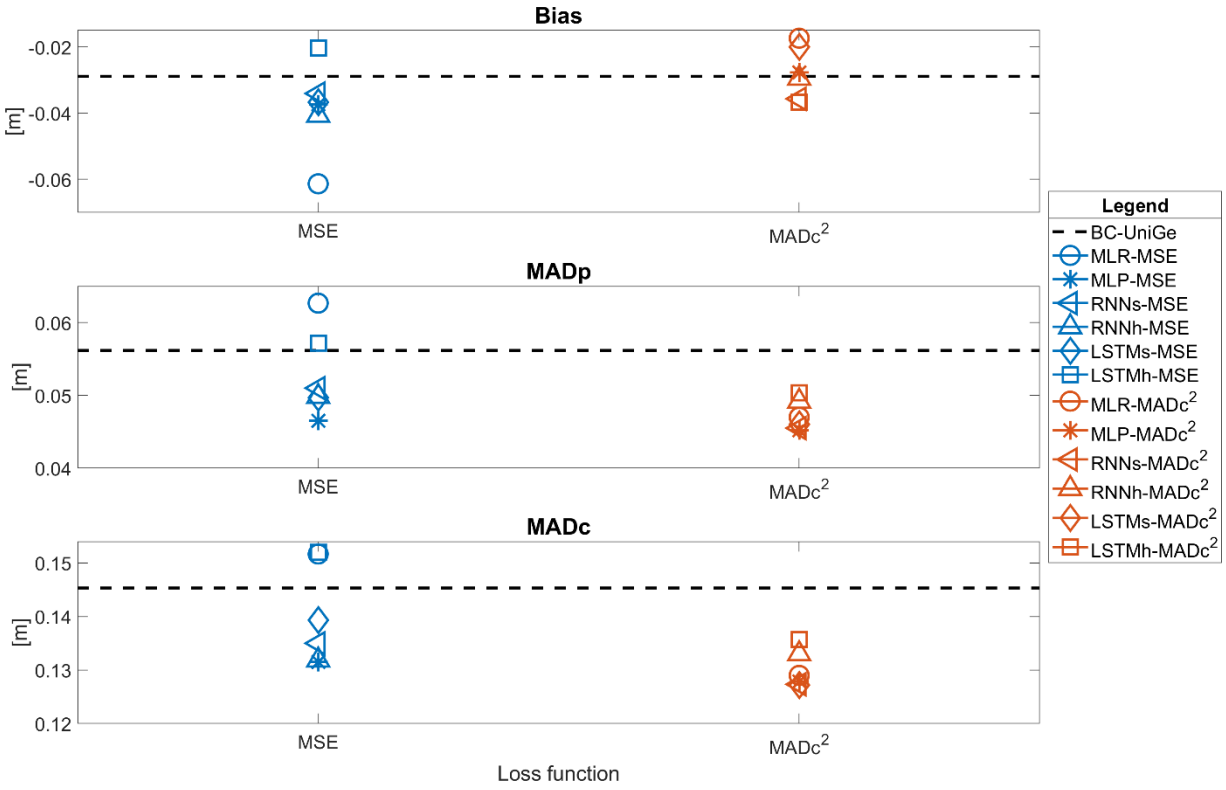

Figure 34: Diagrams of the mean values obtained for bias, MADp, and MADc for surges above the 99th percentile, for the implemented ML models, using BC-UniGe for training and observations for testing.

The percentage quantification of the metric variations for each model, shown in Figure 35, reveals that MLR exhibits the most significant percentage improvements (better performance), particularly in SLF (18.8% improvement), bias (71.1% improvement), and MADp (25% improvement). In contrast, MLP demonstrates relatively stable performance across all metrics, with minimal variations, except for bias, where a 25.8% improvement is observed. The RNN models show mixed behavior: RNNs improve performance in terms of MADp (10.8%) and MADc (5.7%), while RNNh reduces performance in terms of Pearson correlation (-1.7%), RMSE (-2.1%), and MAD (-2.2%), but significantly improves bias (27.5%). The LSTM models exhibit distinct trade-offs, with LSTMs improving most metrics, particularly in Bias (45.6%), MAD (9.3%), and MADc (8.7%). On the other hand, LSTMh experiences significant performance

reductions in bias (-80.7%) and SLF (-8.8%) but shows improvements in all other metrics.

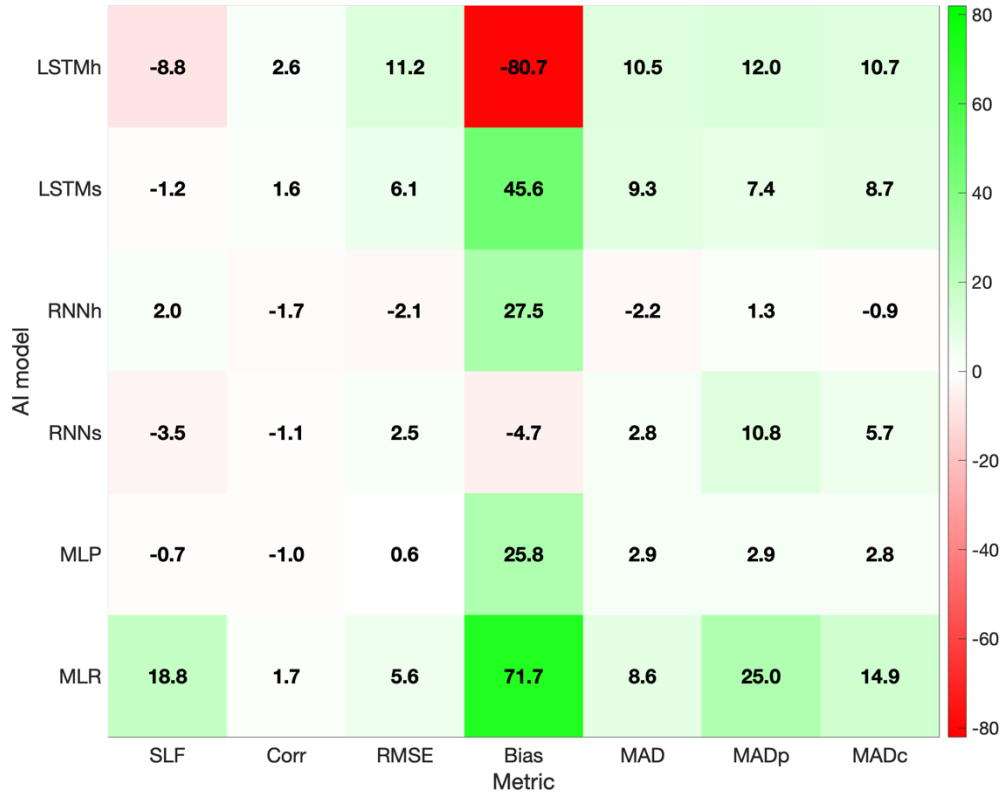

Figure 35: Percentage variation plot of performance metrics for surges above the 99th percentile when MADc^2 is used as the loss function in the implemented ML models, using BC-UniGe for training and observations for testing. Positive values indicate improvements, while negative values indicate a decrease in performance.

3.2.4 PERFORMANCE OF MACHINE LEARNING MODELS USING OBSERVATIONS FOR TRAINING AND TESTING

The second approach for evaluating the performance comparison between the dynamical downscaling and ML models involves training and testing directly on observed data. Due to the availability of homogeneous data over time, this analysis was conducted only for Punta della Salute and Trieste. Table 11 shows the mean values of each metric obtained by the ML models. Overall, the results show that while BC-UniGe sets the benchmark for comparison, several ML models, particularly those

trained with the MADc^2 loss function, demonstrate superior performance in terms of slope accuracy, error reduction, and variability control.

The BC-UniGe dynamical downscaling achieves a strong performance with an SLF of 0.812 and a Pearson correlation of 0.818, demonstrating robust agreement with observed data. It has a moderate RMSE of 0.080 m, a relatively low MAD of 0.061 m, and very small MADp (0.009 m) and MADc (0.071 m) values, indicating consistent predictions with minimal variability.

The MLR-MSE model outperforms BC-UniGe in terms of Pearson correlation (0.825) and achieves a lower RMSE (0.073 m). However, its SLF of 0.718 is the lowest among all models, indicating less accurate slope representation. The MAD metrics (MAD of 0.058 m, MADp of 0.016 m, and MADc of 0.074 m) show slightly higher error variability compared to BC-UniGe. MLR-MADc^2 improves the SLF to 0.777, closer to BC-UniGe. However, its Pearson correlation drops to 0.761, and its RMSE increases to 0.090 m, indicating reduced overall accuracy. The MADp (0.006 m) is lower than BC-UniGe, but the higher MAD and MADc metrics (0.069 m and 0.076 m, respectively) suggest greater error variability.

MLP-MSE shows balanced performance with an SLF of 0.751, a Pearson correlation of 0.812, and an RMSE of 0.077 m, slightly lower than BC-UniGe. Its MAD (0.060 m) and MADp (0.012 m) are comparable to BC-UniGe, but its MADc (0.072 m) is marginally higher. MLP-MADc^2 delivers one of the best performances among all models, with an SLF of 0.836, matching BC-UniGe's correlation (0.818) and achieving a comparable RMSE (0.079 m). Its MAD (0.062 m) is similar to BC-UniGe, while its MADp (0.003 m) and MADc (0.065 m) are the lowest across all models, indicating excellent consistency and reduced variability.

RNNs-MSE achieves a Pearson correlation of 0.839, the highest among all models, coupled with a low RMSE of 0.071 m, outperforming BC-UniGe in overall accuracy. Its SLF of 0.761 and moderate MAD (0.055 m) make it one of the best-performing models, though its MADp (0.014 m) is slightly higher. RNNs-MADc^2 improves the SLF to 0.838, comparable to BC-UniGe, while maintaining a Pearson correlation of 0.817 and an RMSE of 0.079 m. Its MADp (0.004 m) and MADc (0.067 m) metrics are among the best, indicating consistent and accurate predictions.

RNNh-MSE achieves a solid balance with an SLF of 0.740, a Pearson correlation of 0.827, and an RMSE of 0.073 m. Its MAD metrics (MAD of 0.057 m, MADp of 0.015 m, and MADc of 0.073 m) are similar to BC-UniGe, though slightly higher in variability. RNNh-MADc^2 delivers a strong SLF of 0.832, slightly below BC-UniGe, and maintains

good performance with an RMSE of 0.082 m. Its MADp (0.003 m) and MADc (0.067 m) metrics indicate strong variability control, although its Pearson correlation (0.806) is slightly lower.

LSTMs-MSE achieves excellent results, with an SLF of 0.842, a Pearson correlation of 0.823, and a RMSE of 0.078 m, slightly better than BC-UniGe. Its MADp (0.004 m) and MADc (0.065 m) metrics show excellent error variability reduction. LSTMs-MADc^2 achieves the best SLF of 0.860 among all models and a strong Pearson correlation of 0.829, with a low RMSE of 0.077 m. Its MAD metrics (MAD of 0.061 m, MADp of 0.005 m, and MADc of 0.066 m) demonstrate balanced accuracy and low variability.

LSTMh-MSE strikes a good balance, with an SLF of 0.826, the second-highest Pearson correlation (0.836), and an RMSE of 0.073 m, outperforming BC-UniGe. Its MADp (0.006 m) and MADc (0.064 m) are among the lowest, reflecting strong consistency. LSTMh-MADc^2 achieves a strong SLF of 0.834, comparable to BC-UniGe, and balances performance with an RMSE of 0.082 m. Its MADp (0.004 m) and MADc (0.069 m) metrics are slightly higher but still reflect good variability control.

Table 11: Mean values of performance metrics for the ML models implemented using observational data for training and testing.

		Metric					
		SLF	Corr	RMSE m	MAD m	MADp m	MADc m
	BC-UniGe	**0.812**	**0.818**	**0.080**	**0.061**	**0.009**	**0.071**
ML model	MLR-MSE	0.718	0.825	0.073	0.058	0.016	0.074
	MLR-MADc^2	0.777	0.761	0.09	0.069	0.006	0.076
	MLP-MSE	0.751	0.812	0.077	0.06	0.012	0.072
	MLP-MADc^2	0.836	0.818	0.079	0.062	0.003	0.065
	RNNs-MSE	0.761	0.839	0.071	0.055	0.014	0.07
	RNNs-MADc^2	0.838	0.817	0.079	0.062	0.004	0.067
	RNNh-MSE	0.74	0.827	0.073	0.057	0.015	0.073
	RNNh-MADc^2	0.832	0.806	0.082	0.064	0.003	0.067
	LSTMs-MSE	0.842	0.823	0.078	0.062	0.004	0.065
	LSTMs-MADc^2	0.86	0.829	0.077	0.061	0.005	0.066
	LSTMh-MSE	0.826	0.836	0.073	0.058	0.006	0.064
	LSTMh-MADc^2	0.834	0.804	0.082	0.065	0.004	0.069

Figures 36 to 38 present scatter plots for the ML models in Punta della Salute, and Figures 39 to 41 show the scatter plots for the ML models in Trieste, alongside BC-UniGe. In Punta della Salute, the MLR-MSE model (Figure 36b) demonstrates the lowest performance in terms of the proposed MADc metric, with a significant

underestimation of extremes. MLR-MADc^2 (Figure 36c) improves the representation of extremes, as reflected by better SLF and MADp scores, but shows decreased performance across all other metrics. Additionally, in this location, MLP-MADc^2 (Figure 36e), RNNs-MADc^2 (Figure 37c), and RNNh-MADc^2 (Figure 37e) achieve performance comparable to dynamical downscaling, even outperforming it in terms of MADp and MADc. Regarding the LSTM models (Figure 38), most outperform BC-UniGe in terms of MADc (except for LSTMh-MADc^2), with LSTMs-MADc^2 (Figure 38c) surpassing dynamical downscaling in most metrics, demonstrating the capabilities of more sophisticated models in accurately representing observations.

In Trieste, the positive impact of implementing the MADc^2 loss function is evident, with all the ML models in their MADc^2 variant outperforming BC-UniGe across all metrics. These results underscore the importance of high-quality observational data for the application of ML models, demonstrating that even the simplest model with the customized loss function (MLR-MADc^2, Figure 39c) can be a viable and efficient option for storm surge downscaling. However, some underestimations are observed in the highest percentiles by MLP-MADc^2 (Figure 39e), RNNs-MADc^2 (Figure 40c), RNNh-MADc^2 (Figure 40e), and LSTMs-MADc^2 (Figure 41c), highlighting the potential for further refinements in the ML models.

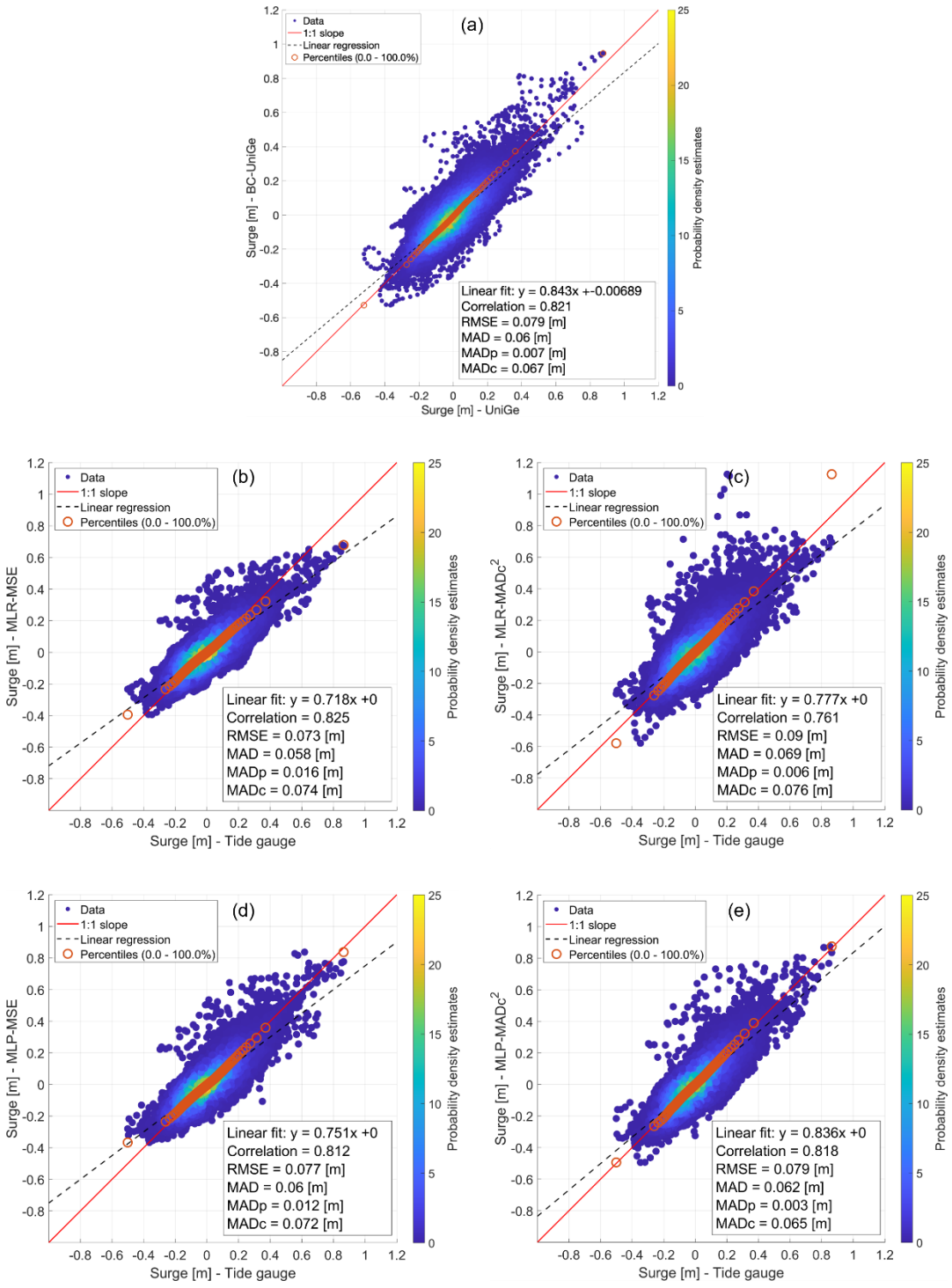

Figure 36: Scatter plots between observations, dynamical downscaling, and MLR and MLP models in Punta della Salute, using observations for training and testing. (a) BC-UniGe, (b) MLR-MSE, (c) MLR-MADc^2, (d) MLP-MSE, and (e) MLP-MADc^2.

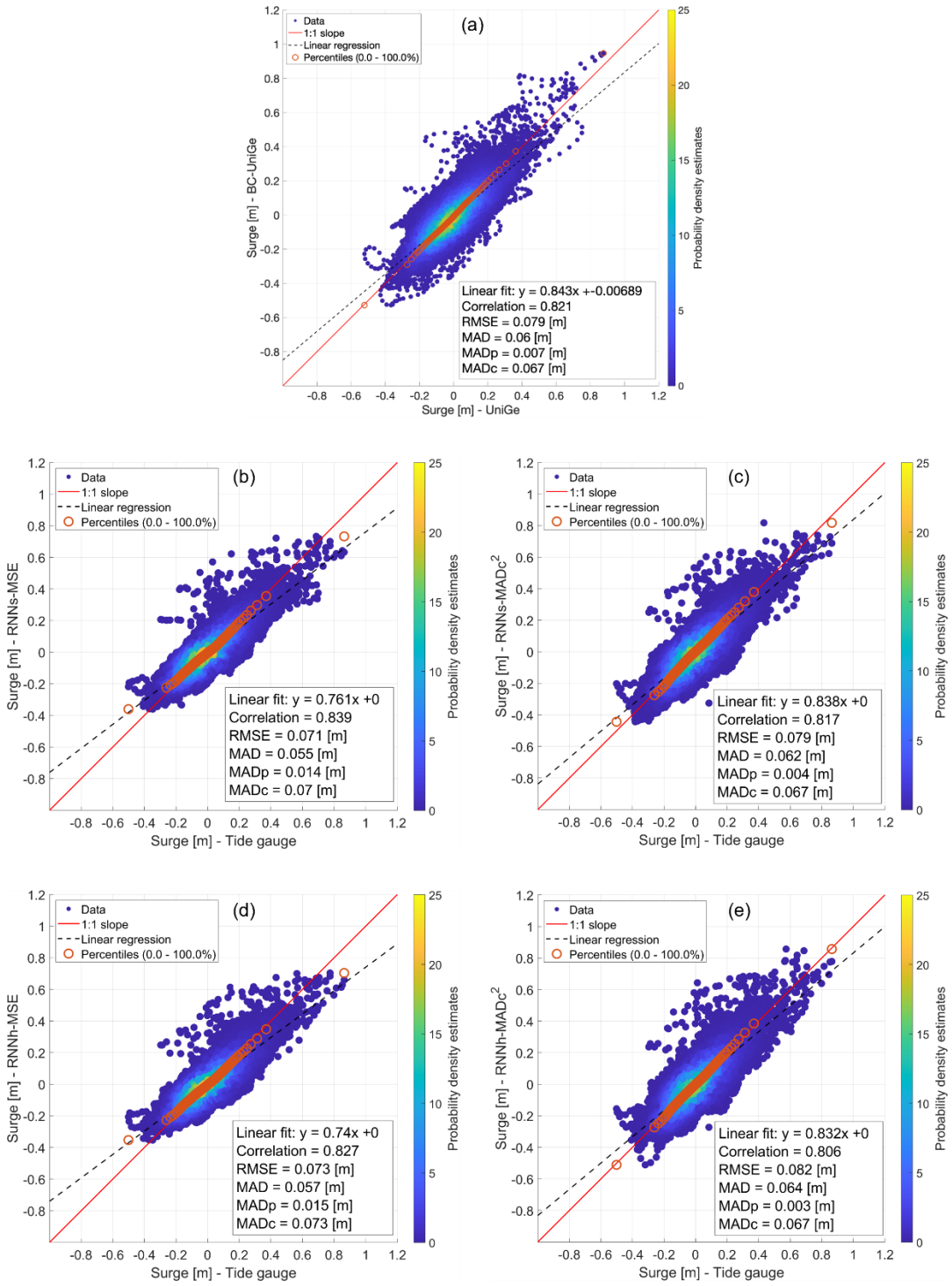

Figure 37: Scatter plots between observations, dynamical downscaling, and RNN models in Punta della Salute, using observations for training and testing. (a) BC-UniGe, (b) RNNs-MSE, (c) RNNs-MADc^2, (d) RNNh-MSE, and (e) RNNh-MADc^2.

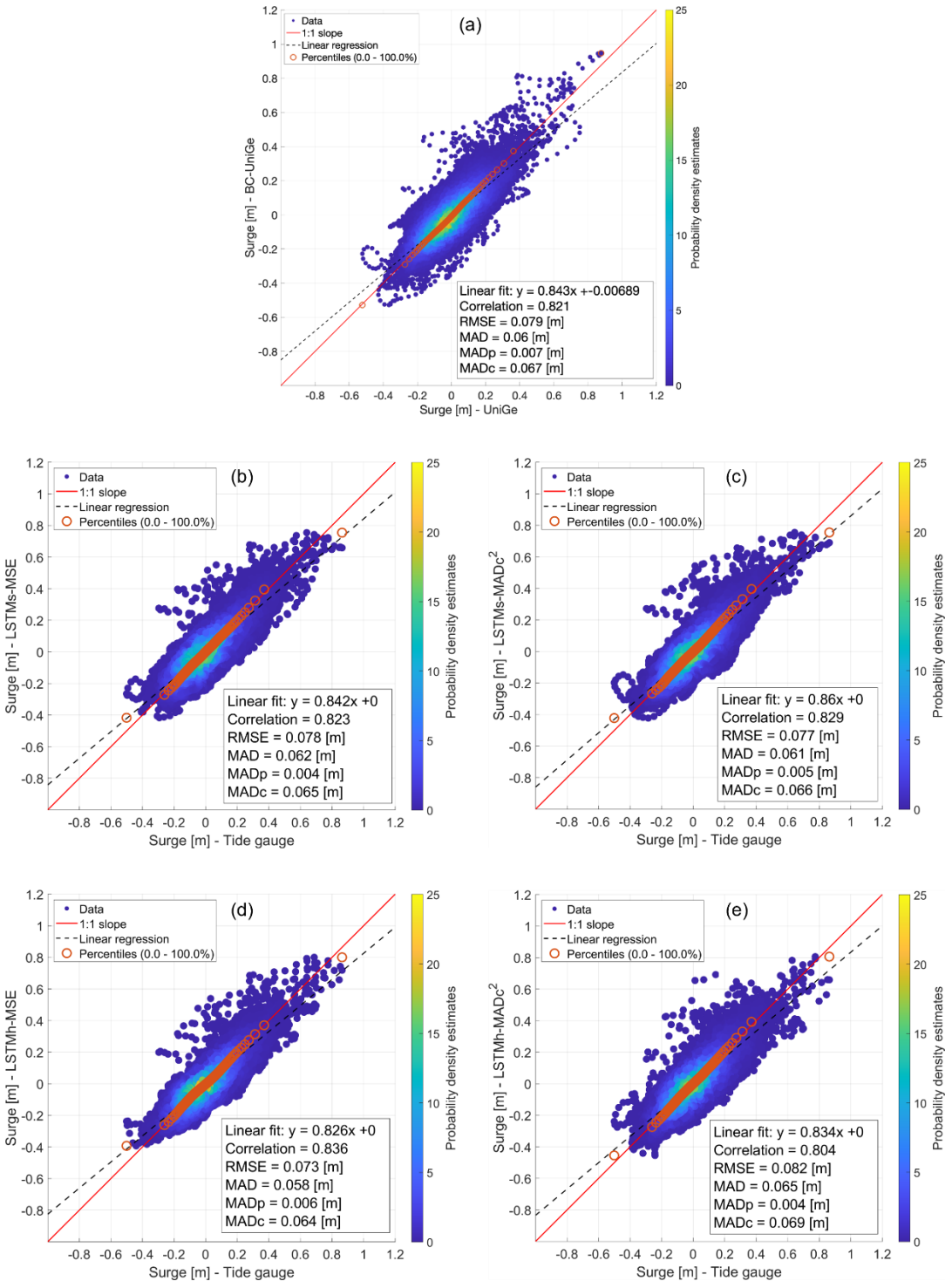

Figure 38: Scatter plots between observations, dynamical downscaling, and LSTM models in Punta della Salute, using observations for training and testing. (a) BC-UniGe, (b) LSTMs-MSE, (c) LSTMs-MADc², (d) LSTMh-MSE, and (e) LSTMh-MADc².

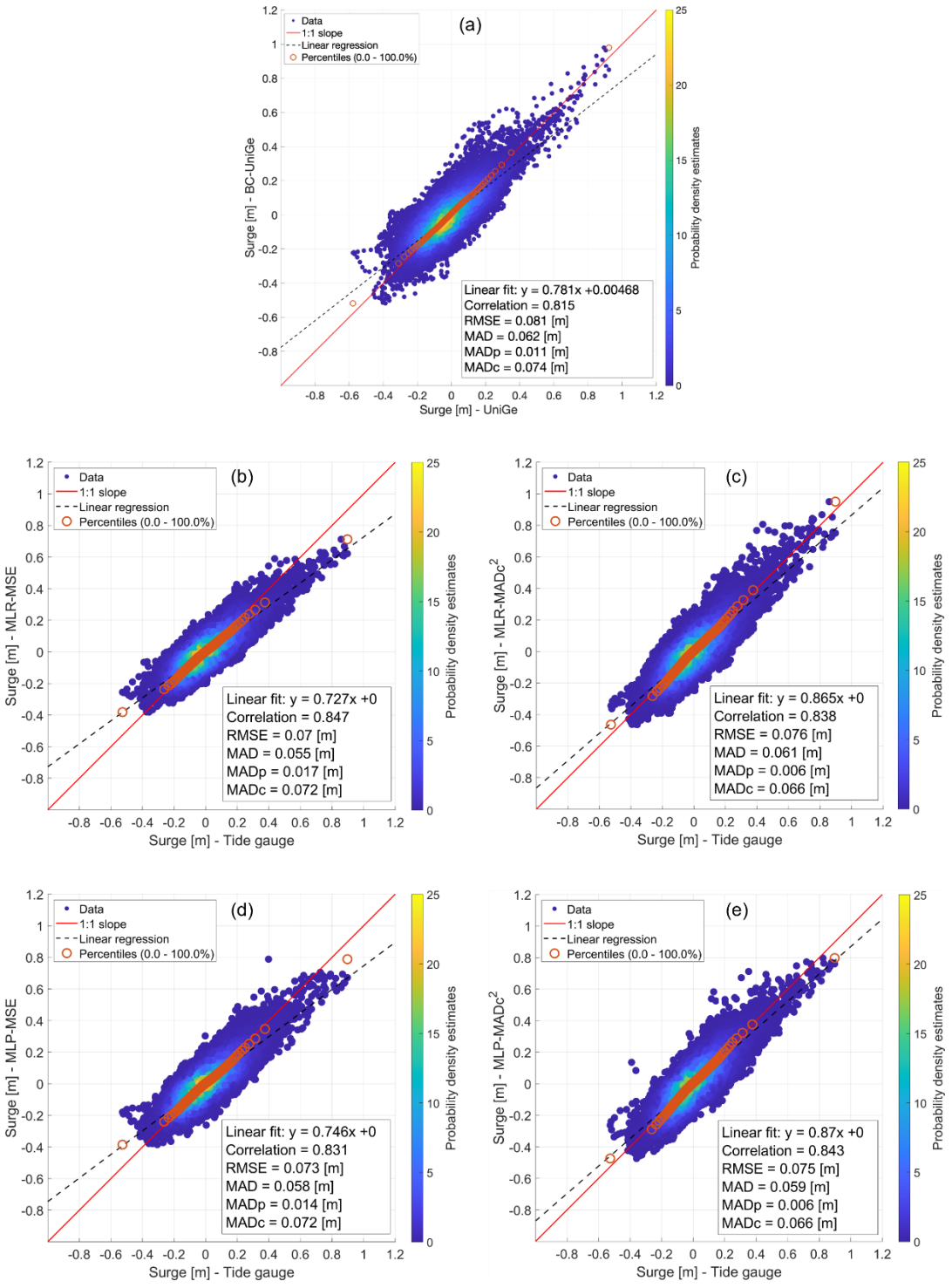

Figure 39: Scatter plots between observations, dynamical downscaling, and MLR and MLP models in Trieste, using observations for training and testing. (a) BC-UniGe, (b) MLR-MSE, (c) MLR-MADc^2, (d) MLP-MSE, and (e) MLP-MADc^2.

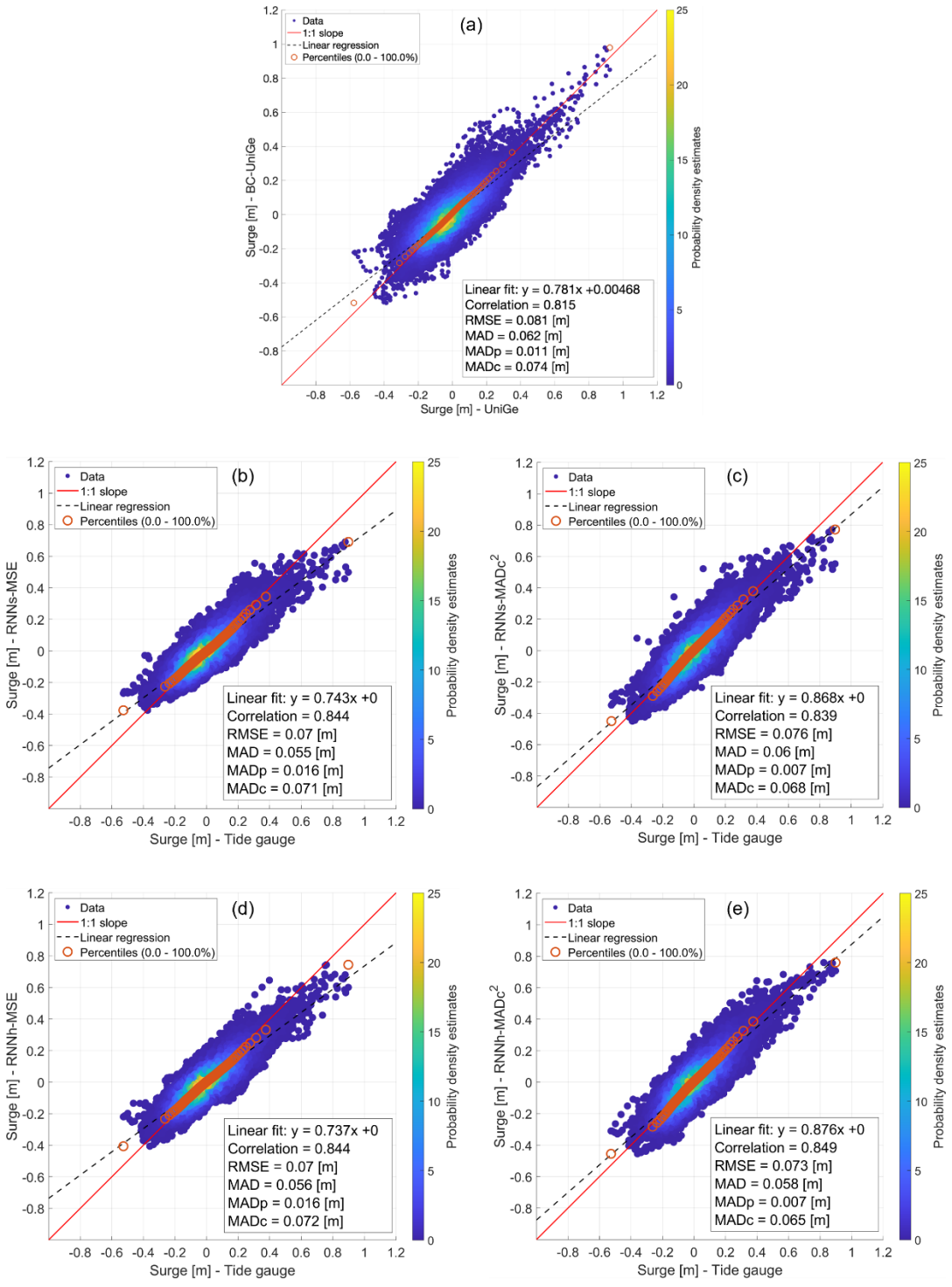

Figure 40: Scatter plots between observations, dynamical downscaling, and RNN models in Trieste, using observations for training and testing. (a) BC-UniGe, (b) RNNs-MSE, (c) RNNs-MADc², (d) RNNh-MSE, and (e) RNNh-MADc².

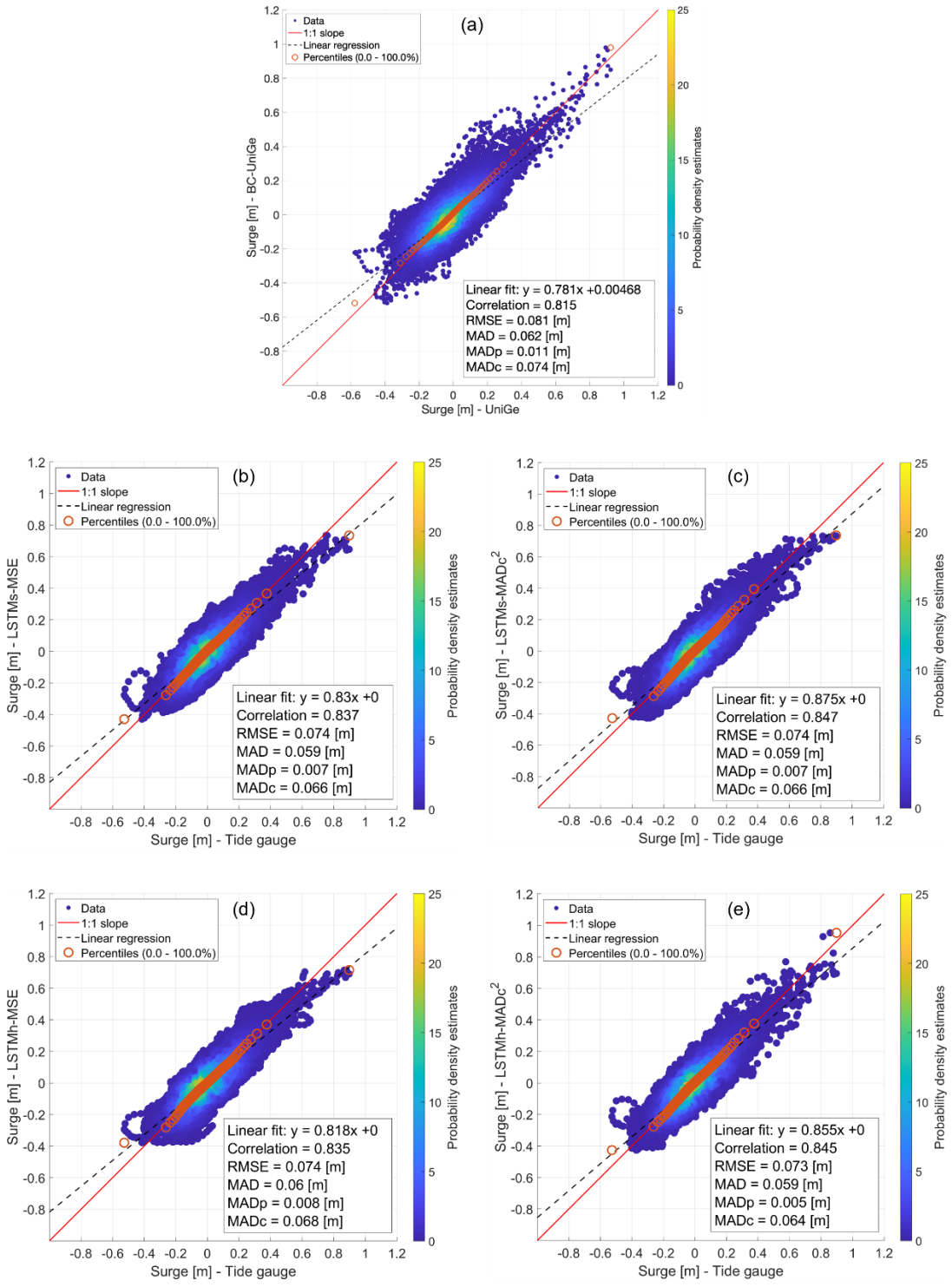

Figure 41: Scatter plots between observations, dynamical downscaling, and LSTM models in Trieste, using observations for training and testing. (a) BC-UniGe, (b) LSTMs-MSE, (c) LSTMs-MADc^2, (d) LSTMh-MSE, and (e) LSTMh-MADc^2.

For values above the 99th percentile, Table 12 shows that this comparison highlights that certain ML models, especially those using the MADc^2 loss function, can rival BC-UniGe in predicting surge values. However, the choice of model and loss function significantly impacts performance.

The dynamical downscaling model, BC-UniGe, achieves a strong Pearson correlation (0.770) and a moderate RMSE of 0.104 m, though it exhibits a slight negative bias of -0.044 m, indicating a consistent underestimation. BC-UniGe demonstrates robust performance in MAD metrics, with values of 0.086 m (MAD), 0.057 m (MADp), and 0.142 m (MADc).

MLR-MSE exhibits lower SLF (0.894) than BC-UniGe, indicating underestimation of surge values. It achieves a comparable RMSE (0.107 m) but shows a larger negative bias (-0.061 m), reflecting a tendency to underpredict. The MAD metrics are slightly higher than BC-UniGe, with 0.089 m (MAD), 0.063 m (MADp), and 0.152 m (MADc). MLR-MADc^2 achieves an SLF (1.062) closer to BC-UniGe, suggesting improved alignment with observed magnitudes. It shows the lowest bias (-0.017 m) and reduced MAD metrics (MAD of 0.081 m, MADp of 0.047 m, and MADc of 0.129 m), outperforming BC-UniGe in these metrics.

MLP-MSE shows an SLF of 1.021, higher than BC-UniGe, with a lower RMSE (0.105 m) and moderate bias (-0.037 m). Its MAD metrics (MAD of 0.085 m, MADp of 0.047 m, and MADc of 0.132 m) indicate similar or better performance compared to BC-UniGe. MLP-MADc^2 achieves an SLF of 1.014 and reduces the MADp (0.045 m) and MADc (0.128 m) errors while maintaining a low bias (-0.028 m). This model performs consistently well across all metrics.

RNNs-MSE achieves an SLF of 0.967 and a low RMSE (0.105 m) but shows moderate bias (-0.034 m). The MAD metrics are slightly better than BC-UniGe, with values of 0.084 m for MAD, 0.051 m for MADp, and 0.135 m for MADc. RNNs-MADc^2 has the lowest SLF (0.932) among all models, indicating underestimation. It achieves reduced MAD values (MAD of 0.082 m, MADp of 0.046 m, and MADc of 0.127 m), showing in general better performance than BC-UniGe.

RNNh-MSE achieves similar results to RNNs-MSE, with improved Pearson correlation (0.774) and slightly lower MAD values (0.082 m for MAD, 0.050 m for MADp, and 0.132 m for MADc). RNNh-MADc^2 has an SLF of 0.987 and achieves comparable results to RNNs-MADc^2, with reduced MADp and MADc values (0.049 m and 0.133 m).

LSTMs-MSE achieves an SLF of 1.024, but its RMSE (0.112 m) and MAD (0.090 m) are slightly higher than BC-UniGe. It shows moderate bias (-0.037 m) and comparable MADp and MADc values (0.050 m and 0.139 m). LSTMs-MADc^2 improves on MAD metrics (MAD of 0.082 m, MADp of 0.046 m, and MADc 0.127 m) while maintaining a low bias (-0.020 m), achieving better overall performance than BC-UniGe.

LSTMh-MSE has an SLF of 1.059, suggesting overestimation. It shows higher RMSE (0.119 m) and MAD metrics (0.095 m for MAD, 0.057 m for MADp, and 0.152 m for MADc), indicating reduced accuracy compared to BC-UniGe. LSTMh-MADc^2 demonstrates a more balanced performance, with reduced SLF (0.966) and improved MAD metrics (MAD of 0.085 m, MADp of 0.050 m, and MADc of 0.136 m), making it competitive with BC-UniGe.

Table 12: Mean values of the performance metrics for values above the 99th percentile for the implemented ML models, using observations for training and testing.

		Metric						
		SLF	Corr	RMSE m	Bias m	MAD m	MADp m	MADc m
	BC-UniGe	**1.099**	**0.770**	**0.104**	**-0.044**	**0.086**	**0.057**	**0.142**
AI model	MLR-MSE	0.894	0.767	0.107	-0.061	0.089	0.063	0.152
	MLR-MADc^2	1.062	0.781	0.101	-0.017	0.081	0.047	0.129
	MLP-MSE	1.021	0.768	0.105	-0.037	0.085	0.047	0.132
	MLP-MADc^2	1.014	0.760	0.105	-0.028	0.083	0.045	0.128
	RNNs-MSE	0.967	0.760	0.105	-0.034	0.084	0.051	0.135
	RNNs-MADc^2	0.932	0.751	0.102	-0.036	0.082	0.046	0.127
	RNNh-MSE	0.967	0.774	0.102	-0.041	0.082	0.050	0.132
	RNNh-MADc^2	0.987	0.761	0.104	-0.029	0.084	0.049	0.133
	LSTMs-MSE	1.024	0.748	0.112	-0.037	0.090	0.050	0.139
	LSTMs-MADc^2	1.012	0.760	0.105	-0.020	0.082	0.046	0.127
	LSTMh-MSE	1.059	0.738	0.119	-0.020	0.095	0.057	0.152
	LSTMh-MADc^2	0.966	0.757	0.106	-0.037	0.085	0.050	0.136

Figure 42 clearly illustrates the improvement in performance in terms of bias, MADp, and MADc with the implementation of MADc^2 as the loss function. The LSTMh-MADc^2 model achieves better performance than BC-UniGe in these metrics and remains competitive even in its MSE variant. The other MADc^2-based ML models outperform the dynamical downscaling in terms of MADp and MADc, demonstrating the positive impact of the MADc^2 loss function on accurately representing the distribution of

extreme surges from observations and reducing errors. Only the MLR-MADc^2 model falls short of surpassing BC-UniGe, although it achieves comparable performance.

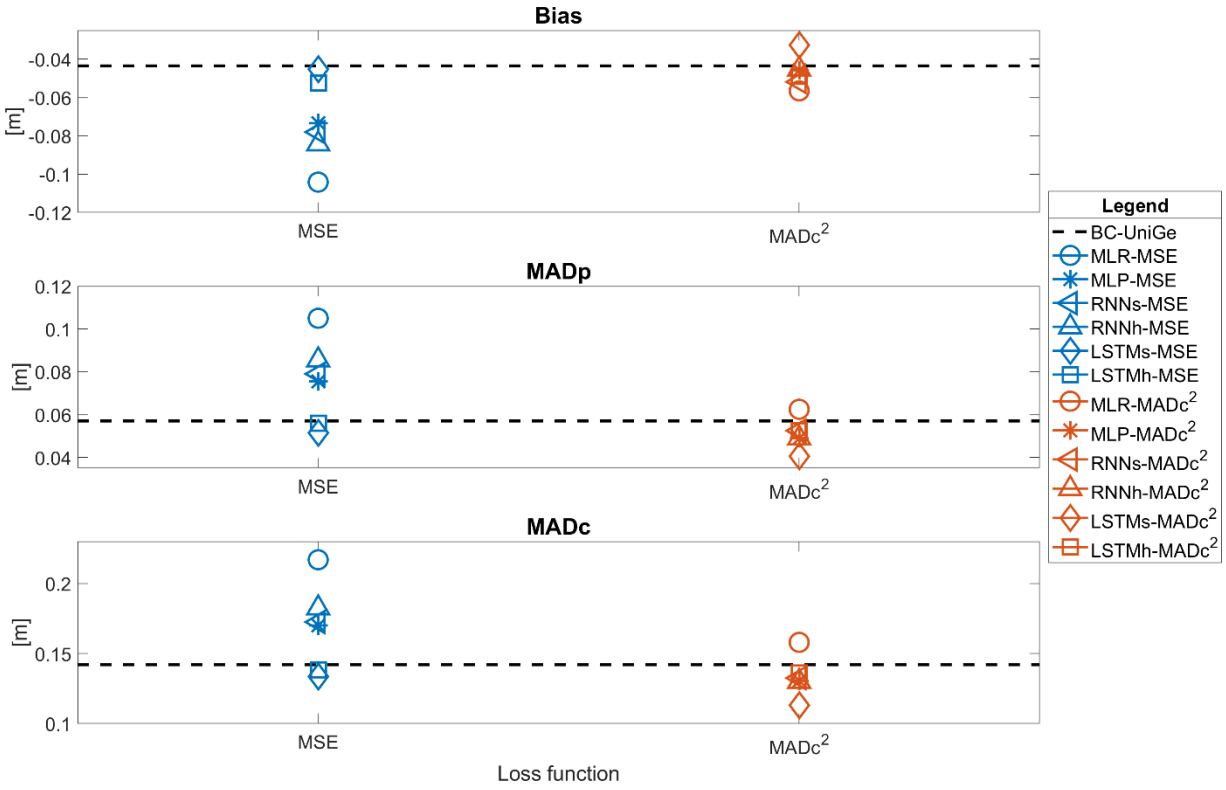

Figure 42: Diagrams of the mean values obtained for bias, MADp, and MADc for surges above the 99th percentile, for the implemented ML models, using observations for training and testing.

The quantitative evaluation of the percentage variation in performance for different metrics of surge values above the 99th percentile (Figure 43) reveals that RMSE, bias, MAD, MADp, and MADc improve across all ML models with the implementation of MADc^2 as the loss function, except for LSTMh, which shows a performance decrease in RMSE (-4.4%) and MAD (-1.2%). The RNNs and LSTMs models demonstrate consistent improvements across all metrics, with the most notable gains in bias (33.6% for RNNs and 27.6% for LSTMs) and MADp (33.5% for RNNs and 21.4% for LSTMs). The MLR and RNNh models also benefit across most metrics, showing significant performance increases in bias (45.7% for MLR and 46.3% for RNNh), MADp (40.5% for MLR and 42.7% for RNNh), and MADc (27.2% for MLR and 28.8% for RNNh), though a decrease is observed in Pearson correlation (-9.8% for MLR and -3.8% for RNNh). The MLP model

also shows improvements across most metrics, with notable gains in bias (37.5%), MADp (35.1%), and MADc (23.2%).

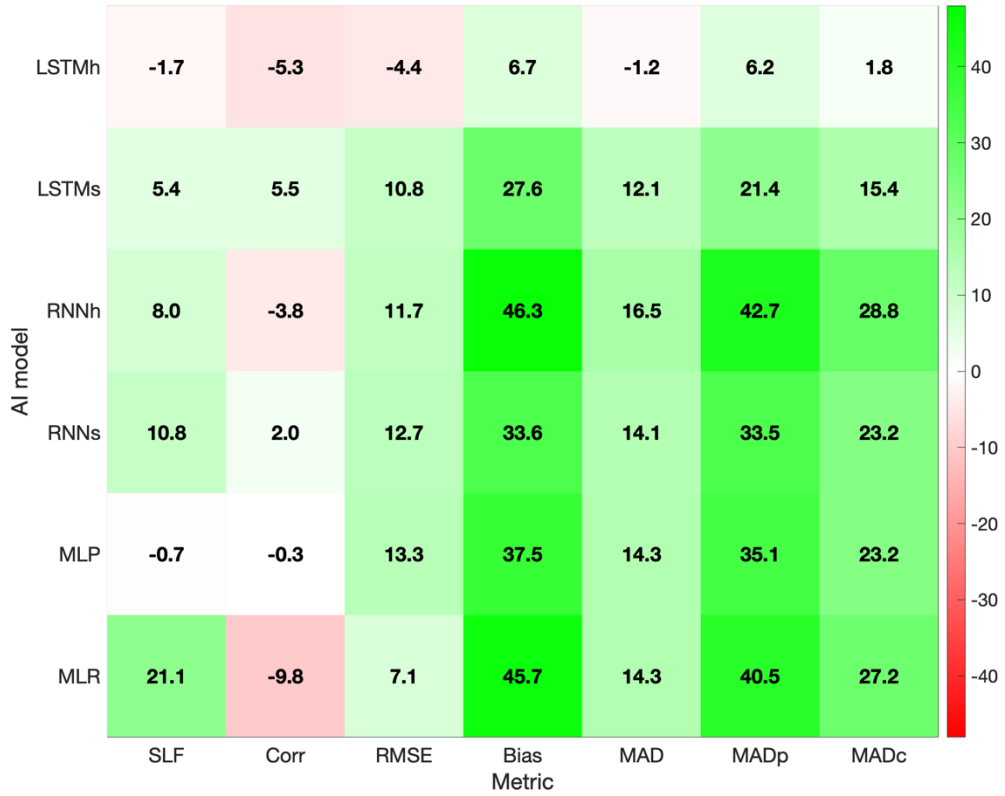

Figure 43: Percentage variation plot of performance metrics for surges above the 99th percentile when MADc^2 is used as the loss function in the implemented ML models, using observations for training and testing. Positive values indicate improvements, while negative values indicate a decrease in performance.

4 DISCUSSION

4.1 DYNAMICAL DOWNSCALING

The utilization of different atmospheric forcing databases has revealed significant implications for the representation of storm surge in numerical simulations. Given the direct influence of wind speed and sea level pressure on this phenomenon, as represented in both forcings databases, the resulting model performances present significant differences. While simulations using ERA5 forcing generally show slightly better performance on traditional metrics such as RMSE, MAD, and Pearson correlation coefficient, a more detailed analysis reveals that using the UniGe forcing results in better performance, especially the extreme values when considering additional metrics.

Simulations using ERA5 forcing tend to underestimate the highest surge values, primarily due to a corresponding underestimation of extreme wind speed by this database, a variable crucially linked to surge amplitude (Campos et al., 2022). Despite this, metrics such as Pearson correlation, RMSE, and MAD generally indicate better performance for ERA5 simulations. Conversely, the utilization of UniGe forcing shows an improvement in representing the peaks of storm surge events (with the noticeable exception of Monfalcone, where the extremes are overestimated, and where MADp present similar values for BC-ERA5 and BC-UniGe). These results demonstrate that the increase in atmospheric forcing resolution does not consistently translate into better values of all the statistical metrics.

It is important to recognize that identifying the optimal model configuration cannot rely solely on a few statistical metrics. As outlined in Section 3.1 no single simulation emerges as superior across all metrics and locations. While ERA5 simulations may demonstrate better performance on RMSE, Pearson correlation, and MAD, BC-UniGe exhibits superior performance in terms of the slope of the linear fit, MADp and MADc.

From an epistemic point of view BC-UniGe is a significantly more sophisticated model compared to BT-ERA5. Not only does it employ a higher resolution forcing. It also takes into account the baroclinicity and the vertical motion within the water column, whereas the barotropic configuration of BT-ERA5 approximates the ocean as a 2D sheet only subject to vertically uniform motions and waves. This suggests that widespread indicators such as RMSE, Pearson correlation, and MAD, which in this case identify BT-ERA5 as the best model, should not be considered as the sole source of information in model skill assessment, since a higher resolution forcing and a

baroclinic setup are known in literature to better capture the variability of the sea levels (Weisberg & Zheng, 2008; Hetzel et al., 2017; Muñoz et al., 2022).

Similar results were found by Zampato et al. (2006) using SHYFEM with three different forcings for wind and atmospheric pressure fields: ECMWF global model, high-resolution LAMI (Limited Area Model Italy) model and satellite QuickSCAT. In this work, the authors found well correlated sea levels with observations near Venice using the ECMWF forcings, but underestimation on highest values. On the other hand, simulations driven by the high-resolution model (LAMI) succeeded in simulating the storm surge, giving a good reproduction of the sea level peaks. Nevertheless, the correlation with observed data was lower than in the case of ECMWF forcing.

The complexity in simulations performance evaluation is echoed in the work of Mentaschi et al. (2013), who caution against over-reliance on metrics like RMSE, NRMSE (Normalized RMSE), and SI (Scatter Index) as indicators of model performance. These metrics may not fully capture the intricacies of natural processes such as atmospheric dynamics, ocean circulation, or wave generation and propagation. These authors mention that the RMSE and its variations tend to assume typical values of the best performance for simulations that underestimate the physical process of interest. The discrepancy between metrics and the representation of extremes highlights the need for a comprehensive understanding of model performance beyond traditional statistical measures.

This results on performance evaluation are usually related to phase error in high-resolution models and RMSE "double penalty". The phase error refers to a discrepancy between the timing or phase of a simulated event and its actual occurrence on measured data. In the context of atmospheric models, phase errors can manifest as delays or advances in the timing of weather events, such as the onset of precipitation, the movement of storm systems, or the arrival of fronts. Double penalty refers to a situation where the errors in the model output are penalized twice, in indicators such as RMSE and MAD, once for missing the observations and again for giving a false alarm (e.g. Gilleland et al., 2009). This is a well-known problem during performance evaluation of numerical models and different contributions have sought to overcome it, with approaches specialized in atmospheric and oceanographic fields (e.g., Ebert & Mcbride, 2000; Zingerle & Nurmi, 2008; Roberts & Lean, 2008; Mittermaier, 2014; Skok & Roberts, 2016; Crocker et al., 2020).

In RMSE, "double penalty" is further amplified compared to MAD, as the penalizations due to the peak mismatch are squared. This means that phase errors have a disproportionately large impact on RMSE. A more sophisticated model may be better

able to capture the magnitude of the peaks, but as it is more prone to phase error compared to low-resolution ones this ability will be doubly penalized. This is the reason why a less sophisticated model employing a low-resolution forcing (BT-ERA5) appears to out-perform the other two in terms of RMSE. Conversely, MAD, although it also experiences a form of "double penalty," reduces the impact of this effect compared to RMSE. As a result, the performance differences between simulations, particularly above the 99th percentile, are generally more pronounced for MAD than for RMSE, better highlighting the superiority of BC-Unige. This enhanced differentiation is likely due to MAD's linear weighting of errors, which reduces the inflated impact of large deviations that characterize RMSE.

In other words, RMSE tends to be better for "blurring" models, whereas high-resolution models, known to be more capable of reproducing small-scale dynamics (e.g. BC-UniGe), perform worse in terms of RMSE due to phase error (Crocker et al., 2020). Although in many aspects, capturing a peak with a phase error is preferable to missing the peak entirely, this does not lead to a reduction in the RMSE.

This limitation of RMSE also impacts the Pearson correlation. Indeed, RMSE can be decomposed into a bias component and a scatter component that depends solely on the Pearson correlation (Mentaschi et al., 2013, Equation 8). All these considerations call for caution when claiming that one model outperforms another one just based on a better value of RMSE or MAD or Pearson correlation.

The MADc indicator was introduced here as a possible way to correct MAD to make it less prone to the double penalty effect. The incorporation in MADc of a term that takes into account the distribution of the data (the MAD of the percentiles MADp) rewards the ability of a high-resolution and more sophisticated model to reproduce the variability in the observations without systematic errors. In other words, MADc remains more resilient to phase errors compared to other metrics, ensuring that discrepancies in the timing of events do not unduly influence the assessment of model performance.

The MADp and MADc metrics were designed to complement, not replace, traditional metrics like RMSE or distribution-based tests (e.g., Kolmogorov-Smirnov, Cramér–von Mises, and Anderson-Darling). While these distributional tests focus on vertical differences and are scale-invariant, MADp captures horizontal differences between percentiles, revealing systematic biases in the distribution, such as underestimation or overestimation of specific quantiles. This can be particularly valuable in applications where preserving the shape and spread of the distribution is critical, such as in storm surge modeling.

Unlike pointwise metrics like RMSE, which can favor models with better timing but worse amplitude, MADp and MADc highlight discrepancies in the model's ability to match the distribution's shape, offering insights into structural errors. While MADp shares similarities with the Cramér–von Mises test, its interpretation as a horizontal distance provides a more direct measure of bias across percentiles, making it a useful tool for model evaluation. Further comparisons with established metrics like Cramér–von Mises or Anderson-Darling would help clarify its advantages.

As shown in Section 3.1, the differences between the simulation metrics are generally in the range of millimeters when considering the overall data, but these differences are significant in relative terms. For the MADc metric, BC-UniGe shows improvements ranging from 1.3% (Grado) to 9.3% (Trieste) compared to BT-ERA5, and from 1.6% (Grado) to 10.3% (Trieste) compared to BC-ERA5. The improvements are even more notable when focusing on values above the 99th percentile, where BC-UniGe outperforms BT-ERA5 by 12% (Monfalcone) to 31.6% (Trieste), and BC-ERA5 by 4.1% (Caorle) to 20.2% (Trieste).

Additionally, some discrepancies were observed in Caorle and Monfalcone, where BC-ERA5 achieved better performance in terms of MADc. A possible explanation for this could be related to the location of the tide gauges at these sites. The tide gauge at Caorle is situated in a protected area inside the Livenza River, a location not fully represented by the simulations due to the resolution of the coastline, even though high-resolution model data were used. A similar issue is found in Monfalcone, where the tide gauge is located in front of a breakwater not fully captured by the coastline representation in the model. These factors could affect the signals obtained from both observations and simulations, primarily due to unresolved local effects at the tide gauge locations. In this line, it is important to consider the potential influence of wave setup in these coastal areas. Wave setup refers to the elevation of the mean water level caused by wave breaking, and it can be particularly relevant in shallow and semi-enclosed environments, such as the tide gauge locations at Caorle and Monfalcone. Since the dynamical downscaling performed in this study does not explicitly account for wave setup contributions, it is possible that part of the observed water level at these tide gauges reflects this unmodeled component, especially during energetic wave conditions. This could further explain the discrepancies observed at Caorle and Monfalcone, where local wave conditions and coastal morphology may enhance wave setup.

In light of the findings of the developed dynamical downscaling, it is worth considering the availability of other storm surge hindcasts for the Adriatic Sea, and

the rationale behind developing a new downscaling in this study. A number of existing storm surge simulations are available for the Adriatic Sea, including operational products from the Euro-Mediterranean Center on Climate Change (CMCC) and high-resolution global datasets such as those by Muis et al. (2016) and Mentaschi et al. (2023). While these datasets are valuable for regional and global-scale assessments, they often fall short in accurately resolving the fine-scale dynamics of storm surges in the shallow and semi-enclosed Northern Adriatic Sea. In particular, they may suffer from underrepresentation of coastal topography, insufficient spatial resolution near the coastline, or lack of local calibration against tide gauge observations. These limitations can lead to biased or smoothed surge estimates, especially during extreme events, which are the primary focus of this study. Therefore, the development of a dedicated dynamic downscaling tailored to the Northern Adriatic Sea provides a significant improvement in spatial resolution, coastal accuracy, and physical realism. Moreover, for training and evaluating ML models, it was critical to work with a physically consistent and locally validated dataset to ensure that model performance could be attributed to the learning capacity rather than limitations in the input data. While existing datasets could, in theory, be used for training, they do not meet the high-fidelity criteria required for the benchmark role that the BC-UniGe downscaling fulfills in this work.

4.2 MACHINE LEARNING DOWNSCALING

The predictors used for the ML downscaling were selected to ensure both performance and broader applicability. As detailed in Section 2.2.2, mean sea level pressure and wind fields from ERA5 were used, alongside sea surface height from Med-MFC. While the dynamical downscaling results highlighted that the UniGe atmospheric forcing provides a more accurate representation of storm surge in the Northern Adriatic, this dataset is not openly available nor continuously updated. In contrast, ERA5 is a globally available, regularly updated reanalysis product, making it more suitable for developing ML models intended for continuous application and transferability to other regions.

The evaluation of different ML model configurations highlights a range of strengths and weaknesses, which vary based on the complexity of the models. Simpler models such as MLR and MLP demonstrate reliable performance across general error metrics while maintaining low computational costs. However, their predictive capabilities falter when addressing extreme events, particularly in percentile alignment and error

correction. For example, MLR-MSE performs significantly worse for values above the 99th percentile, as observed across various evaluation scenarios, including comparisons with both dynamical downscaling data and observations. This limitation underscores the restricted flexibility of simpler models in capturing extreme events. The introduction of $MADc^2$ as a loss function partially addresses these deficiencies by improving corrected deviations and percentile predictions, offering a pathway to enhance model precision without significantly increasing complexity.

In contrast, more sophisticated models like RNN and LSTM leverage their sequential learning capabilities to capture intricate temporal dependencies, resulting in superior alignment with extreme values. However, these models occasionally exhibit higher variability in percentile error metrics, especially in their baseline configurations (e.g., those trained with MSE). The adoption of $MADc^2$ markedly improves percentile alignment and reduces variability for these models, though it sometimes results in slightly diminished performance in general trend metrics for certain RNN configurations. This trade-off highlights the complex interplay between overall accuracy and the ability to predict extreme values effectively. Notably, MLR, particularly when trained with MSE, struggles to represent the tail distribution accurately, leading to significant underperformance in predicting extreme events. Conversely, models incorporating non-linearities, such as the implemented NN models, excel in predicting extreme values, with LSTM models demonstrating a clear advantage in tail predictions due to their ability to capture complex temporal dynamics. While $MADc^2$ mitigates some deficiencies in simpler models, the superior performance of non-linear models like LSTMs underscores the critical role of model complexity in extreme value prediction.

The comparative analysis further reveals the context-dependent nature of these performance differences. While sophisticated models like LSTMs with $MADc^2$ emerge as top performers, simpler models such as MLR and MLP, when optimized with $MADc^2$, achieve comparable performance in certain metrics, including bias reduction and corrected deviations. These findings suggest that simpler models can serve as viable alternatives in specific applications, particularly when computational efficiency and interpretability are prioritized. However, the observed variability across models highlights the importance of tailoring loss functions to prioritize metrics most relevant to the specific use case. This balance between simplicity and sophistication underscores the need for critical consideration in model selection.

The study's results also reveal valuable insights into the interplay between training datasets and model performance. When trained using BC-UniGe data and tested

against observations, the ML models failed to surpass the benchmark across all metrics. This outcome reflects the inherent constraints of the training dataset, as the models were effectively tasked with emulating BC-UniGe rather than learning directly from observed dynamics. The biases and inaccuracies in BC-UniGe limited the models' ability to align predictions closely with reality, raising questions about the adequacy of dynamical downscaling data as a training source for ML-driven modeling efforts.

In contrast, training and testing the models using observational data resulted in clear improvements, with ML models successfully surpassing BC-UniGe performance. This approach highlights the ability of ML models to learn directly from the physical and statistical properties of the observations, thereby capturing subtler dynamics and reducing systematic biases present in the benchmark. However, this method is not without challenges. Observational data often suffer from limitations such as sparse spatial and temporal coverage or measurement uncertainties, which could influence the training process. Moreover, the success of this approach assumes the availability of sufficient high-quality observational data, a condition that may not always be met for many regions or phenomena of interest.

These findings provoke important reflections on the optimal approach for ML model training. Should ML models rely solely on observational data, even when such data are limited? Or should there be a strategic integration of both dynamical models and observational datasets to combine their strengths while mitigating their respective weaknesses? Addressing these questions is essential for developing robust ML-based solutions capable of outperforming traditional dynamical downscaling models across a wide range of applications. Ultimately, the choice of model and training approach must balance the trade-offs between complexity, accuracy, computational efficiency, and the specific requirements of the application at hand.

Regarding computational efficiency, the dynamical downscaling in this study was performed at the CMCC Supercomputing Center using the Zeus computing infrastructure. Zeus is a supercomputer composed of 348 dual-processor Lenovo SD530 nodes (for a total of 12528 cores) interconnected via an InfiniBand EDR network. This high-performance computing system delivers a total theoretical peak performance of 1.202 TFlops. With these computational resources, one year of dynamical downscaling simulation took, on average, 36 hours to complete.

In contrast, the ML model training was conducted on a personal laptop equipped with an Intel® Core™ Ultra 9 185H processor (16 cores, 22 threads, up to 5.1 GHz, 24 MB L3 cache) and 32 GB of DDR5 RAM at 5600 MT/s (2 × 16 GB, dual-channel). The system also featured a 2 TB NVMe SSD for high-speed data access. For GPU-accelerated computations, the laptop was equipped with an NVIDIA RTX™ 3000 Ada Generation Laptop GPU with 8 GB of GDDR6 VRAM, in addition to integrated Intel® Arc™ graphics. This hardware configuration allowed for efficient training of deep learning models and parallel processing tasks. On this laptop, the average time required for a single run (including training and validation) of the ML models using the NVIDIA GPU was as follows:

- MLR models: 20–40 seconds
- MLP models: 15–30 seconds
- RNN models: 85–110 seconds
- RNNh models: 45–90 seconds
- LSTMs models: 190–400 seconds
- LSTMh models: 55–70 seconds

4.2.1 LIMITATIONS

The performance evaluation of the ML models showed on Sections 3.2 and 3.3 could be influenced by the selection of the training and testing sets. As mentioned in Section 2.2.4, the training and testing sets in this study were selected to have similar distributions, a selection that represents a limitation of this research. Figure 44 illustrates a comparison of percentiles between three selected training and testing sets derived from the BC-UniGe output in Punta della Salute, maintaining the train and test data proportion described on Section 2.2.4, with MLR-MSE performance used as a reference. When the training set consists of 28 consecutive years (Figure 44a), the training percentiles are generally higher, while the testing set contains the highest percentile, representing an extreme event. The absence of such extreme events in the training set reduces the model's performance during testing (Figure 44b), as it lacks exposure to these cases. Figure 44c shows an attempt to achieve a more uniform distribution; however, the highest percentile remains in the testing set, and differences persist in percentiles above 0.2 m. Figure 44d shows a more balanced distribution, which is the one considered in this study, with the highest percentile included in the training set. This selection allows the model to learn from a broader range of events, thereby improving predictive accuracy.

An option to alleviate the dependence on training and testing set selection is the use of cross-validation during training. Cross-validation divides the available dataset into multiple subsets, or folds, and systematically trains and tests the model on different combinations of these folds. This approach ensures that each data point is used for both training and testing, reducing biases introduced by any particular data split. Unlike a single training-testing selection, cross-validation provides a more robust evaluation of model performance, as it accounts for variability in the data and helps prevent overfitting to specific subsets. However, its implementation requires more computational resources compared to a single train-test split. This is because the model must be trained and evaluated multiple times, once for each fold or subset. For this reason, and due to the computational resources available during the development of this study, the ML models were implemented without cross-validation but it should be considered on further developments.

A second limitation in this study pertains to the use of PCA to reduce the spatial dimensionality of the predictors. While PCA allows for the retention of predictors that explain most of the variance in the data, it inherently applies a linear transformation to the original dataset. This linearization may overlook important nonlinear patterns or interactions present in the predictors, which could be critical for capturing the complex dynamics of surge behavior.

To address this limitation, an alternative approach involves the use of encoding layers, such as those in neural networks applied by Žust et al. (2021), Rus et al. (2023a), and Rus et al. (2023b). These layers can learn and extract nonlinear patterns and relationships from the predictors, providing a more flexible and data-driven representation of the input features. By leveraging encoding layers, the models can potentially enhance their ability to capture subtle spatial and temporal dependencies, ultimately improving the predictive performance of the ML models. During the development of this study encoding layers were implemented, obtaining heavier models without improvements compared to the ones using PCA. For this reason, the PCA-based dimensionality reduction approach was preferred in this study, as it offers a more computationally efficient solution without compromising model performance. However, the potential of encoding layers to capture complex patterns should not be dismissed, and future research could explore optimized architectures or hybrid approaches that balance model complexity with predictive accuracy.

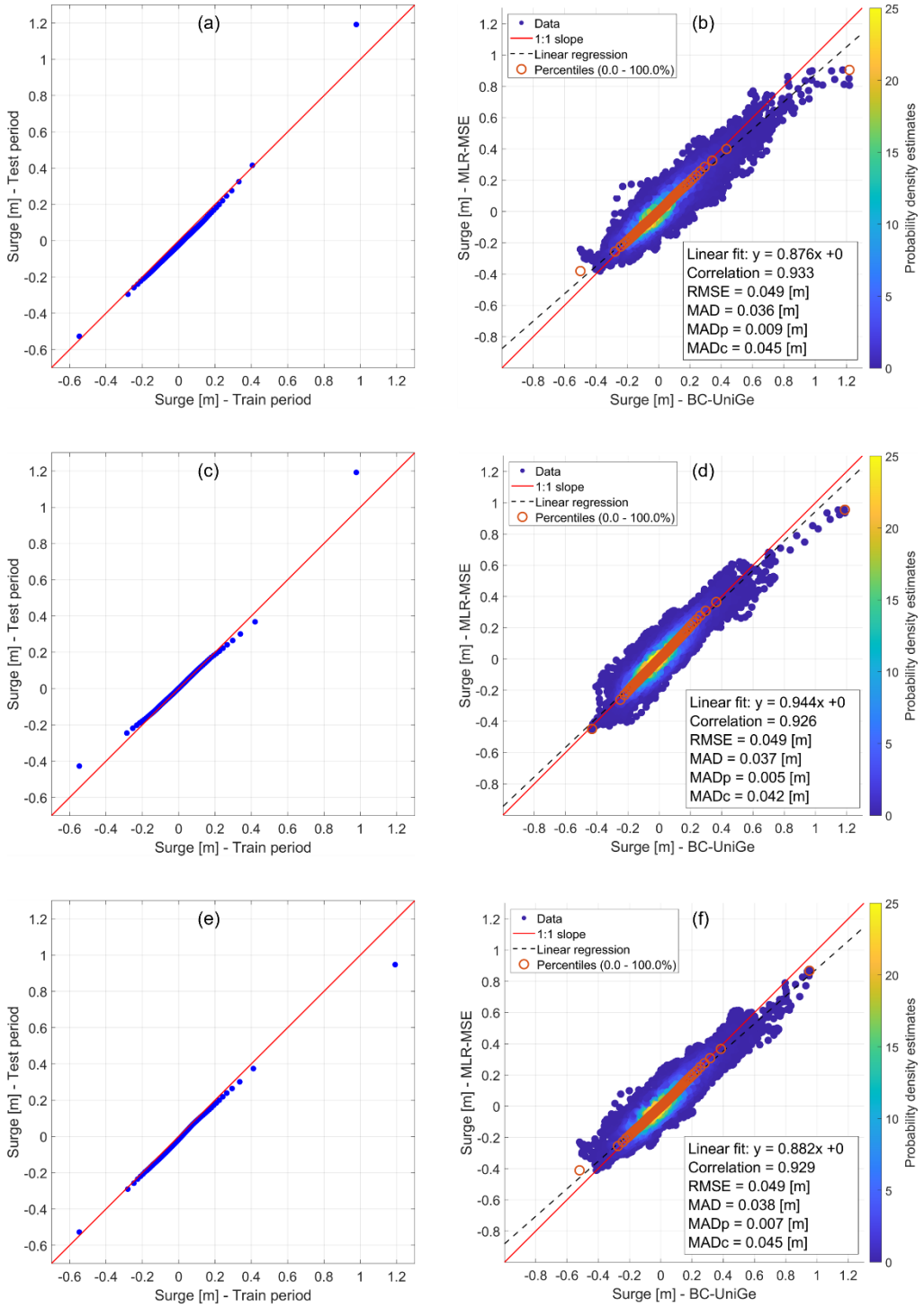

Figure 44: Comparison of percentiles between three selected training and testing sets from the BC-UniGe output in Punta della Salute, alongside MLR-MSE performance as a reference. (a) and (b): train and test sets considered as consecutive years; (c) and (d): attempt of uniform distributions; (e) and (f): improvement of uniform distributions between train and test sets.

5 CONCLUSIONS

This study developed high-resolution storm surge simulations for the Northern Adriatic Sea (1987–2020) using the SHYFEM-MPI model and explored the potential of ML models for storm surge downscaling. The SHYFEM-MPI model was designed to achieve high accuracy by employing various forcing datasets and physical configurations, providing a reliable benchmark for evaluating ML models. Such an approach is unique compared to studies that rely on less optimized dynamical models. Among the SHYFEM-MPI configurations, the baroclinic setup atmospherically forced by high-resolution data (BC-UniGe) demonstrated superior ability to capture storm surge variability and extreme events, despite limitations like phase errors. To address biases in traditional metrics, this study introduced a novel corrected mean absolute deviation (MADc), which proved effective in distinguishing model performance and offers broader applicability in model validation.

The exploration of ML models revealed their potential as efficient alternatives to dynamical downscaling. Simpler models, such as MLR and MLP, performed reliably with minimal computational cost, particularly when using the customized $MADc^2$ loss function. However, MLR has limitations in recognizing non-linear patterns and accurately representing extremes. More complex architectures, including RNN and LSTM networks, excelled in capturing temporal dependencies and non-linearities, aligning with extreme percentiles. Among these, LSTM models trained with $MADc^2$ demonstrated the best overall performance, achieving high accuracy across standard metrics and effectively representing extreme surges. Although the improvement in skill of the ML models over the dynamical model was relatively small, it is important to note that the dynamical model was specifically optimized through extensive development efforts to achieve high accuracy, serving as a close approximation of the best possible performance of numerical models.

When ML models were trained using data from the dynamical model, they naturally absorbed the biases present in the training data. However, the results demonstrate that advanced ML architectures, such as LSTM models, particularly when paired with the $MADc^2$ customized loss function, exhibit competitive performance and even surpass the dynamical model in predicting extreme surge values, especially above the 99^{th} percentile. This suggests that the relationships learned by ML models at lower percentiles remain robust and effective at the extremes, where numerical models tend to face greater challenges. Overall, these results show that the output of the dynamical model can serve as a reliable source of data for training ML models in the absence of high-quality observational data.

Direct training on observed data, specifically in Punta della Salute and Trieste, highlighted the strength of ML models, particularly LSTMs, which surpassed BC-UniGe in key metrics like RMSE, MAD, MADc, and Pearson correlation. This underscores the importance of high-quality observational data for training ML models and demonstrates their capacity to capture temporal patterns and extreme events more effectively than dynamical models in specific cases.

The findings reveal that ML models, particularly advanced architectures like LSTMs, can leverage both dynamical model outputs and observational data to deliver exceptional performance. While dynamical model training introduces inherent biases, the robust design of LSTM models paired with the $MADc^2$ loss function enables them to excel, particularly in predicting extreme surge events. Moreover, when trained on high-quality observational data, these models surpass dynamical models like BC-UniGe across key metrics, showcasing their ability to effectively capture temporal patterns and extremes. These results highlight ML's potential as a competitive and versatile tool for storm surge prediction.

Considering the high computational cost and complexity of setting up and running numerical models, ML models present a transformative alternative. Their ability to deliver competitive results with minimal setup (requiring only a modest workstation with a decent GPU) underscores their potential to outcompete numerical methods in the long run. As data quality improves and ML techniques continue to advance, these models may redefine the landscape of coastal engineering and storm surge prediction, providing faster, cost-effective, and highly accurate solutions.

6 REFERENCES

Adeli, E., Sun, L., Wang, J., & Taflanidis, A. A. (2023). An advanced spatio-temporal convolutional recurrent neural network for storm surge predictions. *Neural Computing and Applications*, *35*(26), 18971–18987. https://doi.org/10.1007/s00521-023-08719-2

Al Kajbaf, A., & Bensi, M. (2020). Application of surrogate models in estimation of storm surge: A comparative assessment. *Applied Soft Computing Journal*, *91*. https://doi.org/10.1016/j.asoc.2020.106184

Al-Attabi, Z., Xu, Y., Tso, G., & Narayan, S. (2023). The impacts of tidal wetland loss and coastal development on storm surge damages to people and property: a Hurricane Ike case-study. *Scientific Reports*, *13*(1). https://doi.org/10.1038/s41598-023-31409-x

Alessandri, J. (2022). *Coastal modelling studies for forecasting and remediation solutions*. Alma Mater Studiorum Università di Bologna.

Alessandri, J., Pinardi, N., Federico, I., & Valentini, A. (2023). Storm Surge Ensemble Prediction System for Lagoons and Transitional Environments. *American Meteorological Society*, *38*. https://doi.org/10.1175/WAF-D-23

Aragão, L., Mentaschi, L., Pinardi, N., Verri, G., Senatore, A., & Di Sabatino, S. (2024). The freshwater discharge into the Adriatic Sea revisited. *Frontiers in Climate*, *6*. https://doi.org/10.3389/fclim.2024.1368456

Asciutto, E., Maioli, F., Manfredi, C., Anibaldi, A., Cimini, J., Isailović, I., Marčeta, B., & Casini, M. (2024). Spatio-temporal patterns of whiting (Merlangius merlangus) in the Adriatic Sea under environmental forcing. *PLoS ONE*, *19*(3 March). https://doi.org/10.1371/journal.pone.0289999

Ayyad, M., Hajj, M. R., & Marsooli, R. (2022). Aritificial intelligence for hurricane storm surge hazard assessment. *Ocean Engineering*, *245*. https://doi.org/10.1016/j.oceaneng.2021.110435

Bajo, M., Medugorac, I., Umgiesser, G., & Orlić, M. (2019). Storm surge and seiche modelling in the Adriatic Sea and the impact of data assimilation. *Quarterly Journal of the Royal Meteorological Society*. https://doi.org/10.1002/qj.3544

Bajo, M., & Umgiesser, G. (2010). Storm surge forecast through a combination of dynamic and neural network models. *Ocean Modelling*, *33*. https://doi.org/10.1016/j.ocemod.2009.12.007

Bellafiore, D., & Umgiesser, G. (2010). Hydrodynamic coastal processes in the north Adriatic investigated with a 3D finite element model. *Ocean Dynam.*, *60*, 225–273. https://doi.org/10.1007/s10236-009-0254-x

Benetazzo, A., Davison, S., Barbariol, F., Mercogliano, P., Favaretto, C., & Sclavo, M. (2022). Correction of ERA5 Wind for Regional Climate Projections of Sea Waves. *Water (Switzerland)*, *14*(10). https://doi.org/10.3390/w14101590

Bertotti, L., Bidlot, J. R., Buizza, R., Cavaleri, L., & Janousek, M. (2011). Deterministic and ensemble-based prediction of Adriatic Sea sirocco storms leading to "acqua alta" in Venice. *Quarterly Journal of the Royal Meteorological Society*, *137*(659), 1446–1466. https://doi.org/10.1002/qj.861

Bezuglov, A., Blanton, B., & Santiago, R. (2016). Multi-Output Artificial Neural Network for Storm Surge Prediction in North Carolina. *ArXiv*, *arXiv:1609.07378*. http://arxiv.org/abs/1609.07378

Bishop, C. (2006). *Pattern recognition and machine learning*. Springer.

Blumberg, A., & Mellor, G. (1987). A description of a three-dimensional coastal ocean circulation model. In *Three Dimensional Coastal Ocean Models* (pp. 1–16). American Geophysical Union.

Bosa, S., Petti, M., & Pascolo, S. (2021). Improvement in the sediment management of a lagoon harbor: The case of Marano Lagunare, Italy. *Water*, *13*. https://doi.org/10.3390/w13213074

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Bruneau, N., Polton, J., Williams, J., & Holt, J. (2020). Estimation of global coastal sea level extremes using neural networks. *Environmental Research Letters*, *15*(7). https://doi.org/10.1088/1748-9326/ab89d6

Bruyère, C. L., Done, J. M., Jaye, A. B., Holland, G. J., Buckley, B., Henderson, D. J., Leplastrier, M., & Chan, P. (2019). Physically-based landfalling tropical cyclone scenarios in support of risk assessment. *Weather and Climate Extremes*, *26*. https://doi.org/10.1016/j.wace.2019.100229

Burchard, H., & Petersen, O. (1999). Models of turbulence in the marine environment-a comparative study of two-equation turbulence models. *Journal of Marine Systems*, *21*, 29–53.

Campos, R. M., Gramcianinov, C. B., de Camargo, R., & da Silva Dias, P. L. (2022). Assessment and Calibration of ERA5 Severe Winds in the Atlantic Ocean Using Satellite Data. *Remote Sensing*, *14*(19). https://doi.org/10.3390/rs14194918

Chaumillon, E., Bertin, X., Fortunato, A. B., Bajo, M., Schneider, J. L., Dezileau, L., Walsh, J. P., Michelot, A., Chauveau, E., Créach, A., Hénaff, A., Sauzeau, T., Waeles, B., Gervais, B., Jan, G., Baumann, J., Breilh, J. F., & Pedreros, R. (2017). Storm-induced marine flooding: Lessons from a multidisciplinary approach. *Earth-Science Reviews*, *165*, 151–184. https://doi.org/10.1016/j.earscirev.2016.12.005

Chen, C., Liu, H., & Beardsley, R. C. (2003). An Unstructured Grid, Finite-Volume, Three-Dimensional, Primitive Equations Ocean Model: Application to Coastal Ocean and Estuaries. *Ocean. Technol., 20*, 159–186.

Chen, K., Kuang, C., Wang, L., Chen, K., Han, X., & Fan, J. (2022). Storm surge prediction based on long short-term memory neural network in the east China sea. *Applied Sciences (Switzerland)*, *12*(1). https://doi.org/10.3390/app12010181

Chepurin, G. A., Carton, J. A., & Leuliette, E. (2014). Sea level in ocean reanalyses and tide gauges. *Journal of Geophysical Research: Oceans*, *119*(1), 147–155. https://doi.org/10.1002/2013JC009365

Cid, A., Camus, P., Castanedo, S., Méndez, F. J., & Medina, R. (2017). Global reconstructed daily surge levels from the 20th Century Reanalysis (1871–2010). *Global and Planetary Change*, *148*, 9–21. https://doi.org/10.1016/j.gloplacha.2016.11.006

Cid, A., Wahl, T., Chamber, D. P., & Muis, S. (2018). Storm surge reconstruction and return water level estimation in Southeast Asia for the 20th century. *Journal of Geophysical Research: Oceans*, *123*, 437–451. https://doi.org/10.1002/2017JC013143

Costa, W., Idier, D., Rohmer, J., Menendez, M., & Camus, P. (2020). Statistical prediction of extreme storm surges based on a fully supervised weather-type downscaling model. *Journal of Marine Science and Engineering*, *8*(12), 1–20. https://doi.org/10.3390/jmse8121028

Crocker, R., Maksymczuk, J., Mittermaier, M., Tonani, M., & Pequignet, C. (2020). An approach to the verification of high-resolution ocean models using spatial methods. *Ocean Science*, *16*, 831–845. https://doi.org/10.5194/os-16-831-2020

Dang, W., Feng, J., Li, D., Fan, M., & Zhao, L. (2024). A dataset of storm surge reconstructions in the Western North Pacific using CNN. *Scientific Data*, *11*(1). https://doi.org/10.1038/s41597-024-03249-5

Danilov, S. (2013). Ocean modeling on unstructured meshes. *Ocean Modelling*, *69*, 195–210. https://doi.org/10.1016/j.ocemod.2013.05.005

De Vries, H., Breton, M., De Mulder, T., Krestenitis, Y., Ozer, J., Proctor, R., Ruddick, K., Salomon, J. C., & Voorrips, A. (1995). A comparison of 2D storm surge models applied to three shallow European seas. *Environmental Software*, *10*, 23–42.

De Zolt, S., Lionello, P., Nuhu, A., & Tomasin, A. (2006). The disastrous storm of 4 November 1966 on Italy. *Natural Hazards and Earth System Sciences*, *6*, 861–879. www.nat-hazards-earth-syst-sci.net/6/861/2006/

Defant, A. (1961). *Physical oceanography* (Vol. 2). Pergamon.

Deltares: Delft. (2024). *Delft3D-FLOW User Manual*.

Desplanque, C., & Mossman, D. (2004). Tides and their seminal impact on the geology, geography, history, and socio-economics of the Bay of Fundy, eastern Canada. *Atlantic Geology*, *40*, 1–130. https://doi.org/10.4138/729

Ebert, E. E., & Mcbride, J. L. (2000). Verification of precipitation in weather systems: determination of systematic errors. *Journal of Hydrology*, *239*, 179–202. www.elsevier.com/locate/jhydrol

Escudier, R., Clementi, E., Cipollone, A., Pistoia, J., Drudi, M., Grandi, A., Lyubartsev, V., Lecci, R., Aydogdu, A., Delrosso, D., Omar, M., Masina, S., Coppini, G., & Pinardi, N. (2021). A high resolution Reanalysis for the Mediterranean Sea. *Frontiers in Earth Science*, 9. https://doi.org/10.3389/feart.2021.702285

Fagherazzi, S., Palermo, C., Rulli, M. C., Carniello, L., & Defina, A. (2007). Wind waves in shallow microtidal basins and the dynamic equilibrium of tidal flats. *Journal of Geophysical Research: Earth Surface*, *112*(2). https://doi.org/10.1029/2006JF000572

Federico, I., Pinardi, N., Coppini, G., Oddo, P., Lecci, R., & Mossa, M. (2017). Coastal ocean forecasting with an unstructured grid model in the southern Adriatic and northern Ionian seas. *Natural Hazards and Earth System Sciences*, *17*(1), 45–59. https://doi.org/10.5194/nhess-17-45-2017

Feng, X.-C., & Xu, H. (2024). Accurate storm surge prediction in hurricane area of the Atlantic Ocean using a new multi-recursive neural network based on gate recursive unit. *Journal of Ocean Engineering and Science*. https://doi.org/10.1016/j.joes.2024.01.001

Fernández-Montblanc, T., Vousdoukas, M. I., Mentaschi, L., & Ciavola, P. (2020). A Pan-European high resolution storm surge hindcast. *Environ. Int., 135*. https://doi.org/10.1016/j.envint.2019.105367

Ferrarese, S., Cassardo, C., Elmi, A., Genovese, R., Longhetto, A., Manfrin, M., & Richiardone, R. (2009). Air-sea interactions in the Adriatic basin: simulations of Bora and Sirocco wind events. *Geofizika*, *26*.

Ferrarin, C., Davolio, S., Bellafiore, D., Ghezzo, M., Maicu, F., Mc Kiver, W., Drofa, O., Umgiesser, G., Bajo, M., De Pascalis, F., Malguzzi, P., Zaggia, L., Lorenzetti, G., & Manfe, G. (2019). Cross-scale operational oceanography in the Adriatic Sea. *Journal of Operational Oceanography*, *12*, 86–103. https://doi.org/10.1080/1755876X.2019.1576275

Ferrarin, C., Lionello, P., Orlić, M., Raicich, F., & Salvadori, G. (2022). Venice as a paradigm of coastal flooding under multiple compound drivers. *Scientific Reports*, *12*(1). https://doi.org/10.1038/s41598-022-09652-5

Ferrarin, C., Valentini, A., Vodopivec, M., Klaric, D., Massaro, G., Bajo, M., De Pascalis, F., Fadini, A., Ghezzo, M., Menegon, S., Bressan, L., Unguendoli, S., Fettich, A., Jerman, J., Licer, M., Fustar, L., Papa, A., & Carraro, E. (2020). Integrated sea storm management strategy: the 29 October 2018 event in the Adriatic Sea. *Nat. Hazards Earth Syst. Sci.*, *20*, 73–93. https://doi.org/10.5194/nhess-20-73-2020

Fetzer, J. H. (1990). What is artificial intelligence? In *Artificial Intelligence: Its Scope and Limits* (Vol. 4). Springer, Dordrecht. https://doi.org/10.1007/978-94-009-1900-6_1

Fofonoff, N. P., & Millard, R. C. (1983). Algorithms for computation of fundamental properties of seawater. *UNESCO Technical Papers in Marine Science*.

Gholamy, A., Kreinovich, V., & Kosheleva, O. (2018). *Why 70/30 or 80/20 relation between training and testing sets: A pedagogical explanation*. https://scholarworks.utep.edu/cs_techrep/1209/

Giaremis, S., Nader, N., Dawson, C., Kaiser, H., Kaiser, C., & Nikidis, E. (2024). Storm surge modeling in the AI era: using LSTM-based machine learning for enhancing forecasting accuracy. *ArXiv*. https://arxiv.org/abs/2403.04818

Giesen, R., Clementi, E., Bajo, M., Federico, I., Stoffelen, A., & Santoleri, R. (2021). Copernicus Marine Service Ocean State Report, Issue 5. Section 4.3: The November 2019 record high water level in Venice, Italy. *Journal of Operational Oceanography*, 1–185. https://doi.org/10.1080/1755876X.2021.1946240

Gilleland, E., Ahijevych, D., Brown, B. G., Casati, B., & Ebert, E. E. (2009). Intercomparison of spatial forecast verification methods. *Weather and Forecasting*, *24*(5), 1416–1430. https://doi.org/10.1175/2009WAF2222269.1

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep Sparse Rectifier Neural Networks. *14th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Gregory, J. M., Griffies, S. M., Hughes, C. W., Lowe, J. A., Church, J. A., Fukimori, I., Gomez, N., Kopp, R. E., Landerer, F., Cozannet, G. Le, Ponte, R. M., Stammer, D., Tamisiea, M. E., & van de Wal, R. S. W. (2019). Concepts and terminology for sea level: mean, variability and change, both local and global. *Surveys in Geophysics*, *40*(6), 1251–1289. https://doi.org/10.1007/s10712-019-09525-z

Gumuscu, I., Islek, F., Yuksel, Y., & Sahin, C. (2023). Spatiotemporal long-term wind and storm characteristics over the eastern Mediterranean Sea. *Regional Studies in Marine Science, 63*. https://doi.org/10.1016/j.rsma.2023.102996

Harter, L., Pineau-Guillou, L., & Chapron, B. (2024). Underestimation of extremes in sea level surge reconstruction. *Scientific Reports*, *14*(1). https://doi.org/10.1038/s41598-024-65718-6

Hastie, T., Tibshirani, R., & Friedman, J. (2007). *The elements of statistical learning* (2nd ed.). Springer.

Herrera Silveira, J. A., Teutli Hernandez, C., Secaira Fajardo, F., Braun, R., Bowman, J., Geselbracht, L., Musgrove, M., Rogers, M., Schmidt, J., Robles Toral, P. J., Andrés,

J., Cabrera, C., & Cano, L. G. (2022). *Hurricane damages to mangrove forests and post-storm restoration techniques and costs*.

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., Simmons, A., Soci, C., Abdalla, S., Abellan, X., Balsamo, G., Bechtold, P., Biavati, G., Bidlot, J., Bonavita, M., … Thépaut, J. N. (2020). The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, *146*(730), 1999–2049. https://doi.org/10.1002/qj.3803

Hervouet, J.-M., & Bates, P. (2000). The TELEMAC modelling system Special issue. *Hydrological Processes*, *14*(13), 2207–2208. https://doi.org/10.1002/1099-1085(200009)14:13<2207::aid-hyp22>3.0.co;2-b

Hetzel, Y., Janekovic, I., & Pattiaratchi, C. (2017). Assessing the ability of storm surge models to simulate coastal trapped waves around Australia. *Coasts & Ports*. https://doi.org/10.3316/informit.929951406285439

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Holthuijsen, L. (2007). *Waves in oceanic and coastal waters*. University Press.

Igarashi, Y., & Tajima, Y. (2021). Application of recurrent neural network for prediction of the time-varying storm surge. *Coastal Engineering Journal*, *63*(1), 68–82. https://doi.org/10.1080/21664250.2020.1868736

Jeromel, M., Malacic, V., & Rakovec, J. E. (2009). Weibull distribution of bora and sirocco winds in the northern Adriatic Sea. *Geofizika*, *26*, 85–100.

Jia, G., Taflanidis, A. A., Nadal-Caraballo, N. C., Melby, J. A., Kennedy, A. B., & Smith, J. M. (2016). Surrogate modeling for peak or time-dependent storm surge prediction over an extended coastal region using an existing database of synthetic storms. *Natural Hazards*, *81*(2), 909–938. https://doi.org/10.1007/s11069-015-2111-1

Jin, X., Shi, X., Gao, J., Xu, T., & Yin, K. (2018). Evaluation of loss due to storm surge disasters in China based on econometric model groups. *International Journal of Environmental Research and Public Health*, *15*(4). https://doi.org/10.3390/ijerph15040604

Jollife, I. T., & Cadima, J. (2016). Principal component analysis: A review and recent developments. In *Philosophical Transactions of the Royal Society A:*

Mathematical, Physical and Engineering Sciences (Vol. 374, Issue 2065). Royal Society of London. https://doi.org/10.1098/rsta.2015.0202

Jolliffe, I. T. (2002). *Principal Component Analysis* (2nd ed.). Springer.

Jurčec, V. (1981). On Mesoscale Characteristics of Bora Conditions in Yugoslavia. In *Contributions to Current Research in Geophysics*. https://doi.org/10.1007/978-3-0348-5148-0_15

Kim, S., Pan, S., & Mase, H. (2019). Artificial neural network-based storm surge forecast model: Practical application to Sakai Minato, Japan. *Applied Ocean Research*, *91*. https://doi.org/10.1016/j.apor.2019.101871

Kingma, D. P., & Ba, J. (2015, December 22). Adam: A method for stochastic optimization. *ICLR*. http://arxiv.org/abs/1412.6980

Kolmogorov, A. (1941). The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers. *Akademiia Nauk SSSR Doklady*, *30*, 301–305.

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lee, T. L. (2006). Neural network prediction of a storm surge. *Ocean Engineering*, *33*, 483–494. https://doi.org/10.1016/j.oceaneng.2005.04.012

Lionello, P., Barriopedro, D., Ferrarin, C., Nicholls, R. J., Orlic, M., Raicich, F., Reale, M., Umgiesser, G., Vousdoukas, M., & Zanchettin, D. (2021). Extreme floods in Venice: characteristics, dynamics, past and future evolution (review article). *Nat. Hazards Earth Syst. Sci.*, *21*, 2705–2731. https://doi.org/10.5194/nhess-21-2705-2021

Lionello, P., Cavaleri, L., Nissen, K. M., Pino, C., Raicich, F., & Ulbrich, U. (2012). Severe marine storms in the Northern Adriatic: Characteristics and trends. *Physics and Chemistry of the Earth*, 93–105. https://doi.org/10.1016/j.pce.2010.10.002

Lionello, P., Galati, M. B., & Elvini, E. (2010). Extreme storm surge and wind wave climate scenario simulations at the Venetian littoral. *Physics and Chemistry of the Earth*, *40–41*, 86–92. https://doi.org/10.1016/j.pce.2010.04.001

Lockwood, J. W., Lin, N., Oppenheimer, M., & Lai, C. Y. (2022). Using Neural Networks to Predict Hurricane Storm Surge and to Assess the Sensitivity of Surge to Storm Characteristics. *Journal of Geophysical Research: Atmospheres*, *127*(24). https://doi.org/10.1029/2022JD037617

Lovato, T., Androsov, A., Romanenkov, D., & Rubino, A. (2010). The tidal and wind induced hydrodynamics of the composite system Adriatic Sea/Lagoon of Venice. *Continental Shelf Research*, *30*(6), 692–706. https://doi.org/10.1016/j.csr.2010.01.005

Luettich, R. A., Westerink, J. J., & Scheffner, N. W. (1992). *ADCIRC: an advanced three-dimensional circulation model for shelves, coasts, and estuaries. Report 1, Theory and methodology of ADCIRC-2DD1 and ADCIRC-3DL.*

Lyard, F., Allain, D. J., Cancet, M., Carrère, L., & Picot, N. (2021). FES2014 global ocean tide atlas: design and performance. *Ocean Sci.*, *17*, 615–649. https://doi.org/10.5194/os-17-615-2021

Malačič, V., Viezzoli, D., & Cushman-Roisin, B. (2000). Tidal dynamics in the northern Adriatic Sea. *Journal of Geophysical Research: Oceans*, *105*(C11), 26265–26280. https://doi.org/10.1029/2000jc900123

Međugorac, I., Orlić, M., Janeković, I., Pasaric, Z., & Pasaric, M. (2018). Adriatic storm surges and related cross-basin sea-level slope. *Journal of Marine Systems*. https://doi.org/10.1016/j.jmarsys.2018.02.005

Mel, R., & Lionello, P. (2014). Verification of an ensemble prediction system for storm surge forecast in the Adriatic Sea. *Ocean Dynamics*, *64:1803-1814*. https://doi.org/10.1007/s10236-014-0782-x

Mel, R. A., Coraci, E., Morucci, S., Crosato, F., Cornello, M., Casaioli, M., Mariani, S., Carniello, L., Papa, A., Bonometto, A., & Ferla, M. (2023). Insights on the Extreme Storm Surge Event of the 22 November 2022 in the Venice Lagoon. *Journal of Marine Science and Engineering*, *11*(9). https://doi.org/10.3390/jmse11091750

Melet, A., van de Wal, R., Amores, A., Arns, A., Chaigneau, A. A., Dinu, I., Haigh, I. D., Hermans, T. H. J., Lionello, P., Marcos, M., Meier, H. E. M., Meyssignac, B., Palmer, M. D., Reese, R., Simpson, M. J. R., & Slangen, A. B. A. (2024). Sea Level Rise in Europe: Observations and projections. In *Sea Level Rise in Europe: 1st Assessment Report of the Knowledge Hub on Sea Level Rise (SLRE1)* (Vol. 4). https://doi.org/10.5194/sp-3-slre1-4-2024

Mentaschi, L., Besio, G., Cassola, F., & Mazzino, A. (2013). Problems in RMSE-based wave model validations. *Ocean Modelling, 72*, 53–58. https://doi.org/10.1016/j.ocemod.2013.08.003

Mentaschi, L., Besio, G., Cassola, F., & Mazzino, A. (2015b). Performance evaluation of Wavewatch III in the Mediterranean Sea. *Ocean Modelling, 90*, 82–94. https://doi.org/10.1016/j.ocemod.2015.04.003

Mentaschi, L., Vousdoukas, M. I., García-Sánchez, G., Fernández-Montblanc, T., Roland, A., Voukouvalas, E., Federico, I., Abdolali, A., Zhang, Y. J., & Feyen, L. (2023). A global unstructured, coupled, high-resolution hindcast of waves and storm surge. *Frontiers in Marine Science*, *10*. https://doi.org/10.3389/fmars.2023.1233679

Mentaschi, L., Vousdoukas, M., Montblanc, T. F., Kakoulaki, G., Voukouvalas, E., Besio, G., & Salamon, P. (2020). Assessment of global wave models on regular and unstructured grids using the Unresolved Obstacles Source Term. *Ocean Dynamics*, *70*(11), 1475–1483. https://doi.org/10.1007/s10236-020-01410-3

Merrifield, M. A., Genz, A. S., Kontoes, C. P., & Marra, J. J. (2013). Annual maximum water levels from tide gauges: Contributing factors and geographic patterns. *Journal of Geophysical Research: Oceans*, *118*(5), 2535–2546. https://doi.org/10.1002/jgrc.20173

Mi Zhang, D. D., Min Yang, X. P., & He, X. (2019). Modeling extreme events in time series prediction. *25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4-8, 2019, Anchorage, AK, USA. ACM, New York, NY, USA*. https://doi.org/10.1145/3292500.3330896

Micaletto, G., Barletta, I., Mocavero, S., Federico, I., Epicoco, I., Verri, G., Coppini, G., Schiano, P., Aloisio, G., & Pinardi, N. (2022). Parallel Implementation of the SHYFEM Model. *Geosci. Model Dev., 15*, 6025–6046. https://doi.org/10.5194/gmd-2021-319

Mittermaier, M. P. (2014). A strategy for verifying near-convection-resolving model forecasts at observing sites. *Weather and Forecasting*, *29*(2), 185–204. https://doi.org/10.1175/WAF-D-12-00075.1

Muis, S., Verlaan, M., Winsemius, H. C., Aerts, J. C. J. H., & Ward, P. J. (2016). A global reanalysis of storm surges and extreme sea levels. *Nature Communications*, *7*. https://doi.org/10.1038/ncomms11969

Muñoz, D. F., Yin, D., Bakhtyar, R., Moftakhari, H., Xue, Z., Mandli, K., & Ferreira, C. (2022). Inter-Model Comparison of Delft3D-FM and 2D HEC-RAS for Total Water Level Prediction in Coastal to Inland Transition Zones. *Journal of the American*

Water Resources Association, *58*(1), 34–49. https://doi.org/10.1111/1752-1688.12952

Murty, T. S., Flather, R. A., & Henry, R. F. (1986). The storm surge problem in the Bay of Bengal. *Prog. Oceanog.*, *16*, 195–233.

Nocedal, J., & Wright, S. (2006). *Numerical optimization* (2nd ed.). Springer.

Oddo, P., Pinardi, N., & Zavatarelli, M. (2005). A numerical study of the interannual variability of the Adriatic Sea (2000-2002). *Science of the Total Environment*, *353*(1–3), 39–56. https://doi.org/10.1016/j.scitotenv.2005.09.061

Orlić, M., Kuzmić, M., & Pasarić, Z. (1994). Response of the Adriatic Sea to the bora and sirocco forcing. *Continental Shelf Research*, *14*, 91–116.

Ostoich, M., Ghezzo, M., Umgiesser, G., Zambon, M., Tomiato, L., Ingegneri, F., & Mezzadri, G. (2018). Modelling as decision support for the localisation of submarine urban wastewater outfall: Venice lagoon (Italy) as a case study. *Environmental Science and Pollution Research*, *25*(34), 34306–34318. https://doi.org/10.1007/s11356-018-3316-0

Park, K., Federico, I., Di Lorenzo, E., Ezer, T., Cobb, K. M., Pinardi, N., & Coppini, G. (2022). The contribution of hurricane remote ocean forcing to storm surge along the Southeastern U.S. coast. *Coastal Engineering*, *173*. https://doi.org/10.1016/j.coastaleng.2022.104098

Paulson, C. A., & Simpson, J. J. (1977). Irradiance measurements in the upper ocean. *J. Physical Oceanography*, *7*(6), 952–956.

Pawlowicz, R., Beardsley, B., & Lentz, S. (2022). Classical harmonic analysis including error estimates in MATLAB using T_TIDE. *Computers & Geosciences*, *28*, 929–937.

Pettenuzzo, D., Large, W. G., & Pinardi, N. (2010). On the corrections of ERA-40 surface flux products consistent with the Mediterranean heat and water budgets and the connection between basin surface total heat flux and NAO. *Journal of Geophysical Research: Oceans*, *115*(6). https://doi.org/10.1029/2009JC005631

Petti, M., Pascolo, S., Bosa, S., Bezzi, A., & Fontolan, G. (2019). Tidal flats morphodynamics: A new conceptual model to predict their evolution over a medium-long period. *Water (Switzerland)*, *11*(6). https://doi.org/10.3390/w11061176

Pineau-Guillou, L., Ardhuin, F., Bouin, M. N., Redelsperger, J. L., Chapron, B., Bidlot, J. R., & Quilfen, Y. (2018). Strong winds in a coupled wave–atmosphere model during a North Atlantic storm event: evaluation against observations. *Quarterly Journal of the Royal Meteorological Society*, *144*(711), 317–332. https://doi.org/10.1002/qj.3205

Polli, S. (1959). La propagazione delle maree nell'adriatico. *Atti Del 9. Convegno Dell'associazione Geofisica Italiana*, 20–21.

Poudyal, A., Lamichhane, S., Wertz, C., Mahmud, S. U., & Dubey, A. (2023). *Hurricane and storm surges-induced power system vulnerabilities and their socioeconomic impact*. http://arxiv.org/abs/2311.15118

Prandtl, L. (1945). Über ein neues formelsystem für die ausgebildete turbulenz. *Nachr. Akad. Wiss. Gottingen*, *11*.

Pringle, W. J., Wirasaet, D., Roberts, K. J., & Westerink, J. J. (2021). Global storm tide modeling with ADCIRC v55: unstructured mesh design and performance. *Geosci. Model. Dev.*, *14*, 1125–1145. https://doi.org/10.5194/gmd-14-1125-2021

Pugh, D. T. (1987). *Tides, surges and mean sea-level*. John Wiley & Sons Ltd.

Pugh, D., & Woodworth, P. (2014). *Sea-Level Science*. Cambridge University Press. https://doi.org/10.1017/CBO9781139235778

Qin, Y., Su, C., Chu, D., Zhang, J., & Song, J. (2023). A review of application of machine learning in storm surge problems. *Journal of Marine Science and Engineering*, *11*. https://doi.org/10.3390/jmse11091729

Raicich, F. (2023). The sea level time series of Trieste, Molo Sartorio, Italy (1869-2021). *Earth System Science Data*, *15*, 1749–1763. https://doi.org/10.5194/essd-15-1749-2023

Ramos-Valle, A. N., Curchitser, E. N., Bruyère, C. L., & McOwen, S. (2021). Implementation of an Artificial Neural Network for Storm Surge Forecasting. *Journal of Geophysical Research: Atmospheres*, *126*(13). https://doi.org/10.1029/2020JD033266

Reimann, L., Vafeidis, A. T., Brown, S., Hinkel, J., & Tol, R. (2018). Mediterranean UNESCO World Heritage at risk from coastal flooding and erosion due to sea-level rise. *Nature Communication*, *9:4161*. https://doi.org/10.1038/s41467-018-06645-9

Roberts, K. J., Pringle, W. J., & Westerink, J. J. (2019). OceanMesh2D 1.0: MATLAB-based software for two-dimensional unstructured mesh generation in coastal ocean modeling. *Geosci. Model. Dev., 12*, 1847–1868. https://doi.org/10.5194/gmd-12-1847-2019

Roberts, N. M., & Lean, H. W. (2008). Scale-selective verification of rainfall accumulations from high-resolution forecasts of convective events. *Monthly Weather Review*, *136*(1), 78–97. https://doi.org/10.1175/2007MWR2123.1

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *ArXiv: 1609.04747v2*.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*.

Rus, M., Fettich, A., Kristan, M., & Ličer, M. (2023a). HIDRA2: deep-learning ensemble sea level and storm tide forecasting in the presence of seiches - the case of the northern Adriatic. *Geosci. Model Dev.*, *16*, 271–288. https://doi.org/10.5194/gmd-16-271-2023

Rus, M., Fettich, A., Kristan, M., & Ličer, M. (2023b). HIDRA-T - A transformer-based sea level forecasting method. *ERK 2023 Computational Intelligence Section*.

Saha, S., Moorthi, S., Pan, H. L., Wu, X., Wang, J., Nadiga, S., Tripp, P., Kistler, R., Woollen, J., Behringer, D., Liu, H., Stokes, D., Grumbine, R., Gayno, G., Wang, J., Hou, Y. T., Chuang, H. Y., Juang, H. M. H., Sela, J., … Goldberg, M. (2010). The NCEP climate forecast system reanalysis. *Bulletin of the American Meteorological Society*, *91*(8), 1015–1057. https://doi.org/10.1175/2010BAMS3001.1

Saillour, T., Cozzuto, G., Ligorio, F., Lupoi, G., & Bourban, S. E. (2021). Modeling the world oceans with TELEMAC. *2020 TELEMAC-MASCARET User Conference*, 86–91.

Šepić, J., Pasarić, M., Međugorac, I., Vilibić, I., Karlović, M., & Mlinar, M. (2022). Climatology and process-oriented analysis of the Adriatic Sea level extremes. *Progress in Oceanography*, *209*. https://doi.org/10.1016/j.pocean.2022.102908

Signell, R. P., Chiggiato, J., Horstmann, J., Doyle, J. D., Pullen, J., & Askari, F. (2010). High-resolution mapping of Bora winds in the northern Adriatic Sea using synthetic aperture radar. *Journal of Geophysical Research: Oceans*, *115*(4). https://doi.org/10.1029/2009JC005524

Skok, G., & Roberts, N. (2016). Analysis of Fractions Skill Score properties for random precipitation fields and ECMWF forecasts. *Quarterly Journal of the Royal Meteorological Society*, *142*(700), 2599–2610. https://doi.org/10.1002/qj.2849

Smagorinsky, J. (1963). General circulation experiment with the primitive equations. *Monthly Weather Review*, *91*, 99–164. https://doi.org/10.1175/1520-0493(1963)091<099:gcewtp>2.3.co;2.9,25

Srivastava, N., Hinton, G., Krizhevsky, A., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Staneva, J., Wahle, K., Koch, W., Behrens, A., Fenoglio-Marc, L., & Stanev, E. (2016). Coastal flooding: impact of waves on storm surge during extremes - a case study for the German Bight. *Nat. Hazards Earth Syst. Sci.*, *16*, 2373–2389. https://doi.org/10.5194/nhess-16-2373-2016

Su, X., Yan, X., & Tsai, C. L. (2012). Linear regression. *Wiley Interdisciplinary Reviews: Computational Statistics*, *4*(3), 275–294. https://doi.org/10.1002/wics.1198

Sun, K., & Pan, J. (2023). Model of storm surge maximum water level increase in a coastal area using ensemble machine learning and explicable algorithm. *Earth and Space Science*, *10*. https://doi.org/10.1029/2023EA003243

Suradhaniwar, S., Kar, S., Durbha, S. S., & Jagarlapudi, A. (2021). Time series forecasting of univariate agrometeorological data: A comparative performance evaluation via one-step and multi-step ahead forecasting strategies. *Sensors*, *21*. https://doi.org/10.3390/s21072430

Tausía, J., Delaux, S., Camus, P., Rueda, A., Méndez, F., Bryan, K. R., Pérez, J., Costa, C. G. R., Zyngfogel, R., & Cofiño, A. (2023). Rapid response data-driven reconstructions for storm surge around New Zealand. *Applied Ocean Research*, *133*. https://doi.org/10.1016/j.apor.2023.103496

Tiggeloven, T., Couasnon, A., van Straaten, C., Muis, S., & Ward, P. J. (2021). Exploring deep learning capabilities for surge predictions in coastal areas. *Scientific Reports*, *11*(1). https://doi.org/10.1038/s41598-021-96674-0

Toomey, T., Amores, A., Marcos, M., & Orfila, A. (2022). Coastal sea levels and wind-waves in the Mediterranean Sea since 1950 from a high-resolution ocean

reanalysis. *Frontiers in Marine Science*, 9. https://doi.org/10.3389/fmars.2022.991504

Trotta, F., Fenu, E., Pinardi, N., Bruciaferri, D., Giacomelli, L., Federico, I., & Coppini, G. (2016). A structured and unstructured grid relocatable ocean platform for forecasting (SURF). *Deep-Sea Research II*, *133*, 54–75. https://doi.org/10.1016/j.dsr2.2016.05.004

Uçar, M. K., Nour, M., Sindi, H., & Polat, K. (2020). The effect of training and testing process on machine learning in biomedical datasets. *Mathematical Problems in Engineering*, *2020*. https://doi.org/10.1155/2020/2836236

Umgiesser, G., Bajo, M., Ferrrarin, C., Cucco, A., Lionello, P., Zanchettin, D., Papa, A., Tosoni, A., Ferla, M., Coraci, E., Morucci, S., Crosato, F., Bonometto, A., Valentini, A., Orlic, M., Haigh, I. D., Nielsen, J. W., Bertin, X., Bustorff Fortunato, A., … Nicholls, R. J. (2021). The prediction of floods in Venice: Methods, models and uncertainty (review article). *Nat. Hazards Earth Syst. Sci.*, *21*, 2679–2704. https://doi.org/10.5194/nhess-21-2679-2021

Umgiesser, G., Canu, D. M., Cucco, A., & Solidoro, C. (2004). A finite element model for the Venice lagoon: Development, set up, calibration and validation. *J. Marine Syst.*, 123–145. https://doi.org/10.1016/j.jmarsys.2004.05.009

Uyanik, G. K., & Guler, N. (2013). A study on multiple linear regression analysis. *Procedia - Social and Behavioral Sciences*, *106*, 234–240. https://doi.org/10.1016/j.sbspro.2013.12.027

Vannucchi, V., Taddei, S., Capecchi, V., Bendoni, M., & Brandini, C. (2021). Dynamical downscaling of era5 data on the north-western mediterranean sea: From atmosphere to high-resolution coastal wave climate. *Journal of Marine Science and Engineering*, *9*(2), 1–29. https://doi.org/10.3390/jmse9020208

Vlachogianni, T., Fortibuoni, T., Ronchi, F., Zeri, C., Mazziotti, C., Tutman, P., Varezić, D. B., Palatinus, A., Trdan, Š., Peterlin, M., Mandić, M., Markovic, O., Prvan, M., Kaberi, H., Prevenios, M., Kolitari, J., Kroqi, G., Fusco, M., Kalampokis, E., & Scoullos, M. (2018). Marine litter on the beaches of the Adriatic and Ionian Seas: An assessment of their abundance, composition and sources. *Marine Pollution Bulletin*, *131*, 745–756. https://doi.org/10.1016/j.marpolbul.2018.05.006

Vousdoukas, M. I., Clarke, J., Ranasinghe, R., Reimann, L., Khalaf, N., Duong, T. M., Ouweneel, B., Sabour, S., Iles, C. E., Trisos, C. H., Feyen, L., Mentaschi, L., &

Simpson, N. P. (2022). African heritage sites threatened as sea-level rise accelerates. *Nature Climate Change*, *12*(3), 256–262. https://doi.org/10.1038/s41558-022-01280-1

Vousdoukas, M. I., Mentaschi, L., Voukouvalas, E., Verlaan, M., Jevrejeva, S., Jackson, L. P., & Feyen, L. (2018). Global probabilistic projections of extreme sea levels show intensification of coastal flood hazard. *Nature Communications*, *9*(1). https://doi.org/10.1038/s41467-018-04692-w

Wang, B., Liu, S., Wang, B., Wu, W., Wang, J., & Shen, D. (2021). Multi-step ahead short-term predictions of storm surge level using CNN and LSTM network. *Acta Oceanologica Sinica*, *40*(11), 104–118. https://doi.org/10.1007/s13131-021-1763-9

Wang, X., Verlaan, M., Veenstra, J., & Lin, H. X. (2022). Data-assimilation-based parameter estimation of bathymetry and bottom friction coefficient to improve coastal accuracy in a global tide model. *Ocean Sci., 18*, 881–904. https://doi.org/10.5194/os-18-881-2022

Wang, X., Wang, R., Hu, N., Wang, P., Huo, P., Wang, G., Wang, H., Wang, S., Zhu, J., Xu, J., Yin, J., Bao, S., Luo, C., Zu, Z., Han, Y., Zhang, W., Ren, K., Deng, K., & Song, J. (2024). XiHE: A data-driven model for global ocean eddy-resolving forecasting. https://doi.org/10.48550/arXiv.2402.02995

Weatherall, P., Marks, K. M., Jakobsson, M., Schmitt, T., Tani, S., Arndt, J. E., Rovere, M., Chayes, D., Ferrini, V., & Wigley, R. (2015). A new digital bathymetric model of the world's oceans. *Earth and Space Science*, *2*(8), 331–345. https://doi.org/10.1002/2015EA000107

Weisberg, R. H., & Zheng, L. (2008). Hurricane storm surge simulations comparing three-dimensional with two-dimensional formulations based on an Ivan-like storm over the Tampa Bay, Florida region. *Journal of Geophysical Research: Oceans*, *113*(12). https://doi.org/10.1029/2008JC005115

World Meteorological Organization. (2011). *Guide to storm surge forecasting*.

Ye, F., Zhang, Y., Yu, H., Sun, W., Moghimi, S., Myers, E., Nunez, K., Zhang, R., Wang, H., Roland, A., Martins, K., Bertin, X., Du, J., & Liu, Z. (2020). Simulating storm surge and compound flooding events with a creek-to-ocean model: Importance of baroclinic effects. *Ocean Modelling*, *145*. https://doi.org/10.1016/j.ocemod.2019.101526

Yu, C. S., Decouttere, C., & Berlamont, J. (1998). Storm surge simulations in the Adriatic Sea. In *G. Gambolati (Ed.), CENAS. Coastline Evolution of the Upper Adriatic Sea due to Sea Level Rise and Natural and Anthropogenic Land Subsidence* (pp. 207–232).

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: Lstm cells and network architectures. In *Neural Computation* (Vol. 31, Issue 7, pp. 1235–1270). MIT Press Journals. https://doi.org/10.1162/neco_a_01199

Zaggia, L., Lorenzetti, G., Manfé, G., Scarpa, G. M., Molinaroli, E., Parnell, K. E., Rapaglia, J. P., Gionta, M., & Soomere, T. (2017). Fast shoreline erosion induced by ship wakes in a coastal lagoon: Field evidence and remote sensing analysis. *PLoS ONE*, *12*(10). https://doi.org/10.1371/journal.pone.0187210

Zampato, L., Umgiesser, G., & Zecchetto, S. (2006). Storm surge in the Adriatic Sea: observational and numerical diagnosis of an extreme event. *Advances in Geosciences*, *7*, 371–378.

Zhang, Y., & Baptista, A. M. (2008). SELFE: A semi-implicit Eulerian–Lagrangian finite-element model for cross-scale ocean circulation. *Ocean Model., 21*, 71–96. https://doi.org/10.1016/j.ocemod.2007.11.005

Zhang, Y. J., Fernandez-Montblanc, T., Pringle, W. J., Yu, H. C., & Cui, L. (2023). Global seamless tidal simulation using a 3D unstructured-grid model (SCHISM v5.10.0). *Geosci. Model. Dev*. https://doi.org/10.5194/gmd0-16-2565-2023

Zhang, Y., Ye, F., Stanev, E. V., & Grashorn, S. (2016). Seamless cross-scale modeling with SCHISM. *Ocean Model., 102*, 64–81. https://doi.org/10.1016/j.ocemod.2016.05.002

Zhao, T., Wang, S., Ouyang, C., Chen, M., Liu, C., Zhang, J., Yu, L., Wang, F., Xie, Y., Li, J., Wang, F., Grunwald, S., Wong, B. M., Zhang, F., Qian, Z., Xu, Y., Yu, C., Han, W., Sun, T., … Wang, L. (2024). Artificial intelligence for geoscience: Progress, challenges, and perspectives. *The Innovation*, *5*(5). https://doi.org/10.1016/j.xinn.2024.100691

Zingerle, C., & Nurmi, P. (2008). Monitoring and verifying cloud forecasts originating from operational numerical models. *Meteorological Applications*, *15*(3), 325–330. https://doi.org/10.1002/met.73

Žust, L., Fettich, A., Kristan, M., & Licer, M. (2021). HIDRA 1.0: deep-learning-based ensemble sea level forecasting in the northern Adriatic. *Geosci. Model Dev.*, *14*, 2057–2074. https://doi.org/10.5194/gmd-14-2057-2021