EPFL

Thèse n° 10 335

Network time series forecasting in photovoltaics power production

Présentée le 25 avril 2024

Faculté des sciences et techniques de l'ingénieur
Laboratoire de traitement des signaux 4
Programme doctoral en génie électrique

pour l'obtention du grade de Docteur ès Sciences

par

Jelena SIMEUNOVIĆ

Acceptée sur proposition du jury

Dr D. Gillet, président du jury
Prof. P. Frossard, Dr R. E. Carrillo Rangel, directeurs de thèse
Prof. E. G. Ogliari Carlo, rapporteur
Prof. G. Kariniotakis, rapporteur
Prof. N. Geroliminis, rapporteur

École polytechnique fédérale de Lausanne

2024

Life is and will ever remain an equation
incapable of solutions, but
it contains certain known factors.
— Nikola Tesla

To my family and friends, for their uncoditional love and support...

Acknowledgements

First, I would like to thank my supervisors, Rafael Carrillo and professor Pascal Frossard, for giving me this unique opportunity and your guidance during the previous years. Thank you for your immense support during this time. I am deeply grateful to Pierre-Jean for the chance to come to Switzerland and to work with truly impressive and inspiring people. I owe a huge thanks to Baptiste Schubnel, from whom I learnt a lot and whose insights always helped me. All of you helped me understand my research area better, motivated me, and gave me invaluable technical and educational lessons.

Then, I want to thank my PhD committee, Gillet Denis, Nikolaos Geroliminis, Emanuele Giovanni Carlo Ogliari and Georges Kariniotakis. Thank you for dedicating your time and efforts to review my work.

To my CSEM colleagues and all the members of LTS4, both past and present, thank you for sharing this journey with me. Thank you for all the laughs, coffee breaks, team building, scientific discussions and support. I special thank Paul, Diya and Pietro, who were always positive, helpful and supportive.

I am grateful to my friends from Serbia who supported me in coming to Switzerland and were great support through the years. A special thanks go to my flatmates Winnie and Leonardo for welcoming me and being the best possible companions one could ask for. I am grateful for all the wonderful people that I have met and got to know during this journey and who became my dear friends: Soumaya, Aude, Johny, Sandra, Tijana, Suzana, Slobodan, Gordana, Darinka, Ema and many more. You all made Switzerland feel like home for me. Also, I am eternally grateful to my closest friends Jovana, Nevena, Luka, Dragana, Marko, Andjela, Tamara, Dobroslav, Sanja and Ivana, for their belief in me and for never letting me doubt myself. You have made these years the most memorable.

Tremendous thanks go to my family, particularly my parents and sisters, for their unwavering support and endless confidence in me. You were my pillar of strength. Particularly, I am incredibly thankful to my grandparents for all the life lessons, you taught me perseverance, optimism and faith.

Finally, the most profound gratitude is for Slobodan, my husband and the greatest support, without whom this achievement would not have been possible.

Bern, December 16, 2023 J.S.

Abstract

Accurate forecasting of photovoltaic (PV) power production is crucial for the integration of more renewable energy sources into the power grid. PV power production is highly intermittent, due to the stochastic cloud behaviour and cloud dynamics. Previous works focused on predicting the dynamics by combining inputs from ground-based cameras, satellite images and numerical weather predictions with physical or statistical models. However, they are costly or have coarse resolution.

The focus of this thesis is to advance the state-of-the-art on short-term solar resources forecasting. We take past PV power from a dense network of PV stations as the main input for forecasting. We leverage a graph signal processing perspective and model multi-site PV production data as signals on a graph to capture their spatio-temporal dependencies and achieve higher spatial and temporal resolution forecasts. In our first contribution two graph neural networks, based on graph convolutional layers to exploit the spatial information, are proposed for deterministic multi-site PV forecasting: the graph-convolutional long short-term memory (GCLSTM) and the graph-convolutional transformer (GCTrafo). These methods rely only on production data and exploit the intuition that PV systems provide a network of virtual weather stations. We show that the proposed models outperform state-of-the-art methods for intra-day forecasting with high spatial and temporal resolution. However, they are difficult to interpret.

Utility operators and grid managers could use insights derived from interpretable models to make more informed decisions. Therefore, we introduce a novel interpretable temporal-spatial multi-windows graph attention network (TSM-GAT) for predicting future PV power. TSM-GAT captures different dynamical spatio-temporal correlations for different parts of the forecasting horizon. Thus, it is possible to interpret which PV stations have the most influence when making a prediction for short-, medium- and long-term intra-day forecasts. We show that the proposed model outperforms multi-site state-of-the-art models for four to six hours ahead predictions and that it yields predicted signals with a closer shape to ground truth.

Although machine learning models for PV production achieve high resolution forecasts without loss in accuracy using only PV power data, they are often black box models, leading to overly smoothed predictions. These models might overlook the impact of variable weather conditions on PV power, indicating the model cannot fully capture cloud dynamics. Since physically informed neural networks have shown great success when modelling physical phenomena, we introduce a physics-informed graph neural network (PING) for forecasting the future concentrations in the advection-diffusion processes on an irregular grid. PING

captures the dynamics by estimating historical velocities. It outperforms baseline models for forecasting cloud concentration index and when combined with GCLSTM outperforms baselines for forecasting PV production.
In this thesis, we introduce state-of-the-art models for high resolution and interpretable PV power production forecasts. Even though the accuracy of the physics-informed model is not better than state of the art, it provides insight into the physical behaviour of the cloud dynamics. This insight into cloud dynamics holds potential for future integration with deep learning models to further enhance forecasting capabilities.
Keywords: Photovoltaic systems, time-series forecasting, machine learning, graph signal processing, graph neural networks, physics informed neural networks, advection-diffusion processes

Résumé

La prévision précise de la production d'énergie photovoltaïque (PV) est cruciale pour promouvoir l'intégration d'un plus grand nombre de sources d'énergie renouvelables dans le réseau électrique. La production d'énergie photovoltaïque est fortement intermittente, en raison du comportement stochastique et de la dynamique des nuages. Les travaux antérieurs se sont concentrés sur la prévision de la dynamique en combinant des données provenant de caméras au sol, d'images satellite et de prévisions numériques du temps avec des modèles physiques ou statistiques. Cependant, ces modèles sont coûteux ou ont une résolution grossière.
L'objectif de cette thèse est de faire progresser l'état de l'art en matière de prévision des ressources solaires à court terme. Nous prenons la puissance photovoltaïque passée d'un réseau dense de stations photovoltaïques comme principale donnée d'entrée pour les prévisions. Nous nous appuyons sur une perspective de traitement des signaux sur graphe et modélisons les données de production photovoltaïque multi-sites comme des signaux sur un graphe afin de capturer leurs dépendances spatio-temporelles et d'obtenir des prévisions à plus haute résolution spatiale et temporelle. Dans notre première contribution, deux réseaux neuronaux graphiques, basés sur des couches convolutives graphiques pour exploiter les informations spatiales, sont proposés pour la prévision déterministe de l'énergie photovoltaïque sur plusieurs sites : la mémoire à long terme grapho-convolutionnelle (GCLSTM) et le transformateur grapho-convolutionnel (GCTrafo). Ces méthodes reposent uniquement sur les données de production et exploitent l'intuition selon laquelle les systèmes photovoltaïques constituent un réseau de stations météorologiques virtuelles. Nous montrons que les modèles proposés sont plus performants que les méthodes de l'état de l'art pour les prévisions intra-journalières à haute résolution spatiale et temporelle. Cependant, ils sont difficiles à interpréter.
Les opérateurs de services publics et les gestionnaires de réseaux pourraient utiliser les informations dérivées de modèles interprétable pour prendre des décisions plus informées. Nous présentons donc un nouveau réseau d'attention graphique multi-fenêtres temporel-spatial interprétable (TSM-GAT) pour prédire la production future d'énergie photovoltaïque. Le TSM-GAT capture différentes corrélations dynamiques spatio-temporelles pour différentes parties de l'horizon de prévision. Ainsi, il est possible d'interpréter quelles stations photovoltaïques ont le plus d'influence sur les prévisions intrajournalières à court, moyen et long terme. Nous montrons que le modèle proposé est plus performant que les modèles multisites de pointe pour les prévisions de quatre ou six heures et qu'il produit des signaux prédits dont la forme est plus proche de la réalité de terrain.
Bien que les modèles d'apprentissage automatique pour la production d'énergie photovol-

taïque permettent d'obtenir des prévisions à haute résolution sans perte de précision en utilisant uniquement des données sur l'énergie photovoltaïque, il s'agit souvent de modèles à boîte noire, ce qui conduit à des prévisions trop lissées. Ces modèles peuvent négliger l'impact des conditions météorologiques variables sur la puissance photovoltaïque, ce qui indique que le modèle ne peut pas saisir pleinement la dynamique des nuages. Étant donné que les réseaux neuronaux à information physique ont fait leurs preuves dans la modélisation des phénomènes physiques, nous introduisons un réseau neuronal graphique à information physique (PING) pour prévoir les concentrations futures dans les processus d'advection-diffusion sur une grille irrégulière. Le modèle PING capture la dynamique en estimant les vitesses historiques. Il surpasse les modèles de référence pour la prévision de l'indice de concentration des nuages et, lorsqu'il est combiné avec GCLSTM, il surpasse les modèles de référence pour la prévision de la production photovoltaïque.
Dans cette thèse, nous introduisons un modèle pour les prévisions de puissance photovoltaïque intra-journalière à haute résolution et interprétables. Même si la précision du modèle informé par la physique n'est pas meilleure que celle de l'état de l'art, il donne un aperçu du comportement physique de la dynamique des nuages. Cet aperçu de la dynamique des nuages offre la possibilité d'une intégration future avec des modèles d'apprentissage profond afin d'améliorer les capacités de prévision.
Mots clés : Systèmes photovoltaïques, prévision de séries temporelles, apprentissage automatique, traitement du signal sur graphe, réseaux neuronaux graphique, réseaux neuronaux informés par la physique, processus d'advection-diffusion

Zusammenfassung

Eine genaue Vorhersage der Photovoltaik (PV) Produktion ist entscheidend für die Integration weiterer erneuerbarer Energiequellen in das Stromnetz. Die PV-Erzeugung ist aufgrund des stochastischen Wolkenverhaltens und der Wolkendynamik stark schwankend. Frühere Arbeiten konzentrierten sich auf die Vorhersage der Dynamik durch die Kombination von Daten aus bodengestützten Kameras, Satellitenbildern und numerischen Wettervorhersagen (NWP) mit physikalischen oder statistischen Modellen. Allerdings sind diese kostspielig oder sie haben eine grobe Auflösung.
Der Fokus dieser Arbeit liegt auf der Weiterentwicklung des Stands der Technik bei der kurzfristigen Vorhersage von Solarressourcen. Wir nehmen die vergangene PV-Leistung aus einem dichten Netz von PV-Stationen als Hauptinput für die Vorhersage. Wir nutzen die Perspektive der Graphen-Signalverarbeitung und modellieren Zeitreihen der Photovoltaik (PV)-Produktion an mehreren Standorten als Signale auf einem Graphen, um ihre räumlich-zeitlichen Abhängigkeiten zu erfassen und Vorhersagen mit höherer räumlicher und zeitlicher Auflösung zu erzielen. In unserem ersten Beitrag werden zwei Graph-Neuronale Netze vorgeschlagen, die auf Graph-Faltungsschichten basieren, um die räumlichen Informationen für deterministische PV-Vorhersagen für mehrere Standorte zu nutzen: die Modelle „Graph-Convolutional Long Short Term Memory“ (GCLSTM) und „Graph-Convolutional Transformer“ (GCTrafo). Diese Methoden basieren ausschließlich auf Produktionsdaten und nutzen die Intuition, dass PV-Systeme ein Netzwerk virtueller Wetterstationen darstellen. Wir zeigen, dass die vorgeschlagenen Modelle die modernsten standortübergreifenden Vorhersagemethoden für Vorhersagehorizonte von sechs Stunden mit hoher räumlicher und zeitlicher Auflösung übertreffen. Sie sind jedoch schwer zu interpretieren
Energieversorgungsunternehmen und Netzbetreiber könnten die aus interpretierbaren Modellen gewonnenen Erkenntnisse nutzen, um fundiertere Entscheidungen zu treffen. Die Beteiligten, wie Versorgungsunternehmen und Netzbetreiber, können jedoch die aus einem interpretierbaren Modell gewonnenen Erkenntnisse nutzen, um fundiertere Entscheidungen zu treffen. Daher stellen wir ein neuartiges interpretierbares zeitlich-räumliches Multi-Windows-Graph-Attention-Network (TSM-GAT) zur Vorhersage der zukünftigen PV-Stromproduktion vor. TSM-GAT kann sich an die Dynamik des Problems anpassen und erfasst unterschiedliche dynamische räumlich-zeitliche Korrelationen für verschiedene Teile des Vorhersagehorizonts. So ist es möglich zu interpretieren, welche PV-Stationen den größten Einfluss auf die Vorhersage für kurz-, mittel- und langfristige tagesinterne Vorhersagen haben. Wir zeigen, dass das vorgeschlagene Modell die modernsten Multi-Site-Modelle für Vorhersagen von vier bis sechs

Stunden im Voraus übertrifft und dass es Vorhersagesignale liefert, deren Form näher an der Wirklichkeit liegt als der Stand der Technik.
Obwohl Modelle des maschinellen Lernens für die PV-Stromerzeugung hochauflösende Vorhersagen ohne Genauigkeitsverluste erzielen, indem sie nur PV-Stromdaten verwenden, handelt es sich häufig um Black-Box-Modelle, die zu übermäßig geglätteten Vorhersagen führen. Diese Modelle übersehen möglicherweise die Auswirkungen variabler Wetterbedingungen auf die PV-Leistung, was darauf hinweist, dass das Modell die Wolkendynamik nicht vollständig erfassen kann. Da physikalisch informierte neuronale Netze (PINNs) großen Erfolg bei der Modellierung physikalischer Phänomene gezeigt haben, führen wir ein physikalisch informiertes Graph-Neuronales Netz (PInG) ein, um die zukünftigen Konzentrationen in den auf Advektion und Diffusion basierenden Prozessen vorherzusagen, die auf einem unregelmäßigen Gitter liegen. Das PING erfasst die Dynamik durch Schätzung der historischen Geschwindigkeiten. Es übertrifft die Basismodelle bei der Vorhersage des Wolkenkonzentrationsindex und in Kombination mit GCLSTM die Basismodelle bei der Vorhersage der PV-Produktion.
In dieser Arbeit stellen wir ein hochmodernes Modell für hochauflösende und interpretierbare Intra-Day-PV-Leistungsprognosen vor. UAuch wenn die Genauigkeit des physikalisch informierten Modells nicht besser ist als der Stand der Technik, bietet es doch einen Einblick in das physikalische Verhalten der Wolkendynamik. Dieser Einblick in die Wolkendynamik birgt Potenzial für die zukünftige Integration mit Deep-Learning-Modellen, um die Vorhersagefähigkeiten weiter zu verbessern.
Stichwörter: Photovoltaikanlagen, Zeitreihenvorhersage, maschinelles Lernen, Graphen - Signalverarbeitung, Graph-Neuronale Netze, physikalisch informierte Neuronale Netze, Advektions - Diffusionsprozesse

Contents

List of Figures

List of Tables

1 Introduction

Motivation

Improving power predictions of intermittent and non-dispatchable energy sources is one of the key elements contributing to increasing the penetration of variable renewable energy in the power grid. Accurate time-series forecasting of renewable power generation is vital for improving electricity management, power system scheduling and trading on the electricity market (Gonçalves et al., 2021). It is challenging to integrate variable solar power sources into the existing power grid due to the dynamics of their production and the dependence of their production on weather conditions, including irradiance and cloud movements. Thus, it is crucial to accurately forecast PV power generation on all horizons, including intra-hour (up to one hour ahead), intra-day (from one hour up to six hours ahead), day-ahead (six hours to twenty-four hours ahead) and long term forecasts (from two days to year(s) ahead). This thesis focuses on the deterministic intra-day PV production forecasts on a dense network of PV stations since they are essential for grid congestion management and energy trading.

The main challenges of predicting PV power generation are related to its volatile characteristics and the temporal and spatial dependencies of the irradiance and cloud patterns. The intermittency of solar production has a deterministic origin, coming from the earth and sun's astronomical parameters, and a stochastic one, whose significant daily contribution varies with the cloud dynamics. In order to address the variability of cloud dynamics, substantial research endeavours are dedicated to extracting and predicting the cloud motion vectors from ground-based cameras, satellite images, and numerical weather prediction (NWP) (Antonanzas et al., 2016; Li et al., 2016; Sirch et al., 2016; Song et al., 2022). Although ground-based cameras provide precise local information, they are expensive to deploy and maintain over many PV stations. On the other hand, satellite-based images or numerical weather predictions provide wide-area observational data but are computationally highly expensive. A question arises whether multi-site data-driven methods that rely solely on past production data can provide better intra-day forecasts than those incorporating additional sensors or numerical weather forecasts.

Different machine learning models investigated this question using only the past PV data from multiple sites (Benavides Cesar et al., 2022). Both linear and non-linear models had considerable success modelling the spatio-temporal correlations between PV power production data (Vyas et al., 2022; Lai et al., 2018). Furthermore, different non-linear models employed recurrent neural networks or attention mechanisms to capture the temporal correlations (Agoua et al., 2018; Dai et al., 2023; Harrou et al., 2020). Attention mechanisms and convolutional neural networks are often utilized to capture spatial correlations (Shih et al., 2019). However, these models only partially exploit the spatio-temporal relations from multiple PV stations. In order to improve the accuracy of the model and capture better spatio-temporal correlation between the PV stations, a Graph signal processing (GSP) perspective has been taken. GSP is an emerging field that allows the processing of signals on irregular domains, leveraging graphs to capture their spatial relationships (Sahili and Awad, 2023). While spatio-temporal graph forecasting is studied in various fields, the application for deterministic multi-site PV power is yet to be explored. Hence, the graph signal processing perspective should be leveraged to fully exploit spatio-temporal correlations among PV stations and infer part of the cloud dynamics.

Graph-based models for time series forecasting tasks successfully improved the accuracy of the forecast in different domains. However, these models predominantly employ predefined graph architectures, confining their correlation detection to a limited set of predefined nodes (Khodayar et al., 2019; Khodayar and Wang, 2019). This limitation emerges when graphs utilize predefined k-neighbours; they inherently restrict correlation discovery to those specific nodes and neglect the impact of the further away nodes. Moreover, we model physical phenomena when modelling PV power production, and an interpretable model is desirable. However, state-of-the-art graph-based models utilize recurrent and graph convolutional neural networks, which are difficult to interpret. This limitation highlights the need for an interpretable graph neural network for time-series forecasting in the PV power generation domain.

Another challenge encountered by most of the data-driven machine learning methods, which rely solely on PV power production data when making a forecast, is that they are used as black-box models. The forecasts from these models are usually too smooth and, as such, do not fully capture cloud dynamics. Thus, physical processes guiding the cloud dynamics, advection-diffusion processes, are entirely neglected in these models. Although PV power forecasting models that use numerical weather predictions use physical models, they require numerical methods to solve the set of physical equations, making them highly computationally expensive. On the other hand, satellite images can extract and propagate the cloud motion, but they utilize optical flow or particle image velocimetry (Quesada-Ruiz et al., 2014; Yang et al., 2020), which are more suitable for rigid bodies. In addition, cloud vector detection and propagation require analyzing and processing images, which can be computationally expensive for a wide area. Oppositely, methods that utilize only ground-based PV power data can accelerate 100 times the forecast computation (Carrillo et al., 2022). These outcomes lead us to conclude that the knowledge of advection-diffusion processes should be utilized in data-driven PV forecasting models with ground-based PV power data.

The accurate forecasting of spatio-temporal PV power production with high spatial and temporal resolution using only ground-based PV power data, remains a predominant challenge in the field today. Similarly, designing an interpretable machine learning model tailored for intra-day PV power prediction is yet another complex endeavour that has to be addressed. An equally significant challenge is the development of a physically informed machine learning model that captures cloud dynamics and can accurately forecast PV power production. This thesis seeks to address previously mentioned challenges.

Thesis Outline

This thesis exploits a Graph Signal Processing perspective and machine learning for PV power generation forecasting on a dense network of PV stations. The work is organized in four main objectives: background analysis, exploiting spatio-temporal relations between PV power data using graph-based neural network, building an interpretable graph-based model for PV power forecasting and studying if introduction of the advection-diffusion physical laws improve capturing the cloud dynamics and PV power forecast. The schematic representation of these objectives is given in Figure 1.1. Each of these objectives are described in the dedicated chapters:

Chapter 2 covers the background analysis of the multi-site PV power generation forecasting. It discusses the main advantages and drawbacks of the models that incorporate additional information besides ground-based PV power data, satellite images, NWP and meteorological data. It introduces the traditional PV forecasting models, persistence and smart persistence, as well as the state-of-the-art models for photovoltaic power production. Then, it presents the foundational concepts in graph signal processing and different neural networks, including graph convolution and graph attention networks, and recurrent neural networks which serve as building blocks in the next chapters.

Chapter 3 answers whether multi-site data-driven methods that rely solely on past production data can provide better intra-day forecasts than those incorporating additional sensors or numerical weather forecasts. First, state-of-the-art machine learning methods for PV power forecasting are discussed in detail. Then, a graph signal processing perspective is proposed to model PV production time series as signals on a graph for high temporal and spatial resolution of the PV forecast. The proposed models utilize the graph convolutional neural networks to model spatial correlations and recurrent and transformer networks to model temporal relations between the data. They outperform state-of-the-art multi-site models and single-site models that utilise numerical weather predictions as additional source of information.

Chapter 4 addresses the challenge of creating an interpretable model for the PV power forecasting task in order to capture cloud dynamics. First, it discusses state-of-the-art graph-based models for time-series forecasting and rises question of their interpretability. It

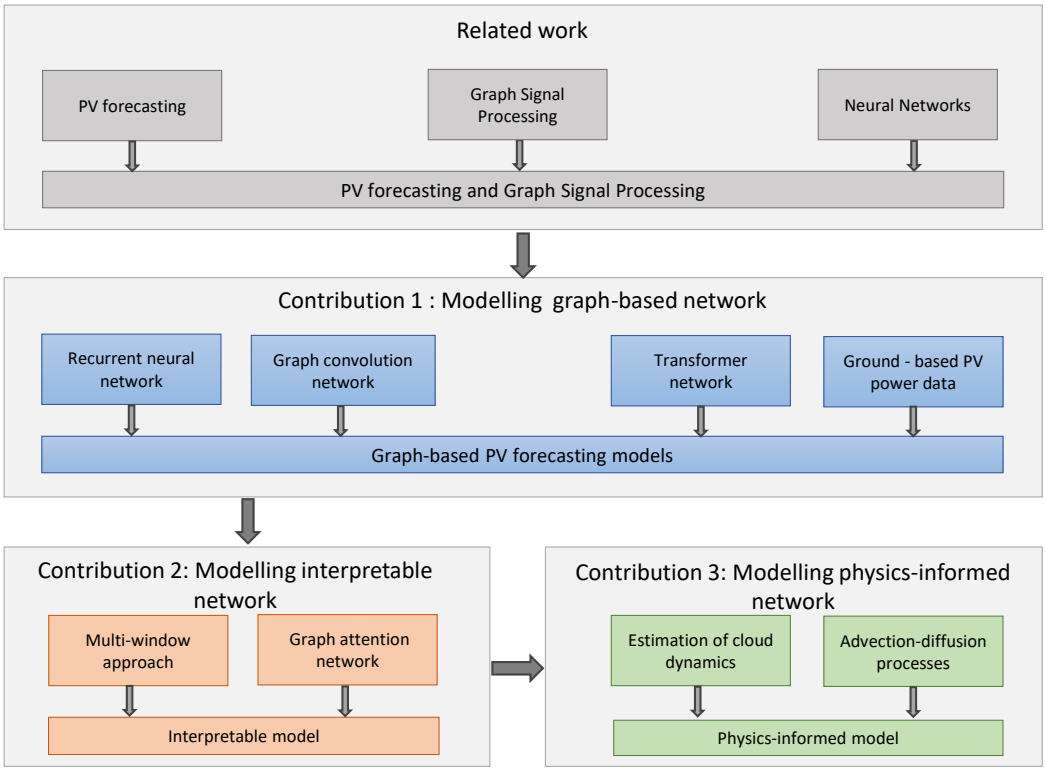


Figure 1.1: Schematic representation of the objectives of the thesis.

proposes to model the time series on a dynamic graph and to utilize the multi-window approach in order to capture cloud dynamics for short-, medium- and long-term part of intra-day forecasting horizon. Interpretable graph-based model, which utilizes graph attention, is proposed for intra-day PV power forecasting. Compared to the state of the art, the proposed model's shape of predicted signal is closer to the ground truth, indicating the model's ability to capture cloud dynamics. The proposed model outperforms state of the art for four to six hours ahead PV power prediction.

Chapter 5 tackles the question of capturing cloud dynamics by introducing physical laws in the model for PV power forecasting. Particularly, a novel physically informed graph neural network for forecasting advection-diffusion processes is proposed to address the scenario when limited historical data is available on both regular and irregular grid. First, the state-of-the-art physically inspired neural networks on regular grids and

meshes are investigated. Then, it proposes to utilize the physical laws in conjecture with graph neural networks to estimate historical velocity direction and forecast future fluid concentration on regular and irregular domains. The proposed model is evaluated on different advection-diffusion processes including cloud index, sea surface temperature, synthetic datasets and PV power generation. When compared to state-of-the-art models, the proposed model outperforms them for cloud index and PV power prediction dataset.

Chapter 6 contains a summary of the main outcomes and future perspectives.

List of Contributions

This thesis is based on the following contributions.

Journal papers:

- Simeunović, Jelena, Baptiste Schubnel, Pierre-Jean Alet, and Rafael E. Carrillo, “Spatio-temporal graph neural networks for multi-site PV power forecasting." IEEE Transactions on Sustainable Energy, Volume 13, no. 2 (2022): 1210-1220.
- Simeunović, Jelena, Baptiste Schubnel, Pierre-Jean Alet, Rafael E. Carrillo, and Pascal Frossard, “Interpretable temporal-spatial graph attention network for multi-site PV power forecasting." Applied Energy, Volume: 327 (2022): 120127.
- Simeunović, Jelena, Baptiste Schubnel, Pierre-Jean Alet, Rafael E. Carrillo, and Pascal Frossard. “PInG: physically informed graph neural networks for forecasting advection-diffusion processes”, – to be submited 2023

Conference papers :

- Carrillo, Rafael, Baptiste Schubnel, Jelena Simeunovic, Renaud Langou, Pierre-Jean Alet, "Spatio-temporal machine learning methods for multi-site PV power forecasting", Proceedings of the 38th European Photovoltaic Solar Energy Conference and Exhibition, EU PVSEC (2021).

Conference talks :

- “Spatio-temporal graph neural networks for multi-site PV power forecasting", IEEE Power & Energy Society General Meeting Conference 2022 in Denver, US.
- “Spatio-temporal graph neural networks for multi-site PV power forecasting”, Applied Machine Learning Days Conference 2022 in Lausanne, Switzerland.
- “Interpretable temporal-spatial graph attention network for multi-site PV power forecasting", Graph Signal Processing Workshop 2023 in Oxford, UK.

2 Background

In this Chapter, we introduce the main topics of the thesis and define the principal concepts. We first introduce the concept of PV power forecasting. Then, the Chapter focuses on Graph Signal Processing as the foundation for processing signals on irregular domains, such as sensory networks and PV stations. We introduce two machine learning architectures inspired by Graph Signal Processing (GSP) concepts: the Graph Convolutional Network (GCN) and the Graph Attention Network (GAT). These architectures effectively model spatial correlations present in time-series data. Then for capturing temporal correlations, recurrent neural networks can be employed. Among them, the Long Short-Term Memory (LSTM) network stands out as one of the most frequently utilized methods.

2.1 PV power forecasting

PV power forecasting is essential to ensure grid stability when integrating highly variable solar energy sources. Accurate forecasts are important for different aspects of grid management, including optimization of energy distribution, managing peak demand to avoid grid congestion and cost reduction. Research in PV power forecasting can be categorized based on the temporal horizon of the forecast on intra-hour forecast or nowcasting, intra-day or short-term, day-ahead or medium-term and long-term forecasts (Antonanzas et al., 2016). Intra-hour forecasting (nowcasting) addresses immediate dynamics, and the prediction span is from a few seconds to up to an hour ahead. They are used in real-time optimization for energy management systems (Moreno et al., 2021). Intra-day forecasting focuses on predictions up to six hours ahead, and they are used for intra-day market participation and day-ahead operation optimization to ensure commitment, scheduling, and dispatch of generated electrical power (Iheanetu, 2022). Day-ahead forecasts cover the horizon from six hours to a day ahead, which is essential in grid management and energy trading strategies. Long-term forecasts are focused on periods longer than two days ahead, providing a broader perspective for strategic planning, transmission and distribution management. According to the origin of the inputs, forecasting models could be divided into models that use outputs from sky imagers, satellite images, NWP, meteorological measurements and information from nearby PV plants. In Table 2.1,

Table 2.1: Classification of data sources for PV power forecasting according to temporal horizon

	Intra-hour (0-1h)	Short-term (1-6h)	Medium-term (6-48h)	Long-term (2+ days)
Data sources	- Sky-imagers - Satellite images - NWP - Meteorological data - Neighbouring PV plants	- Satellite image - NWP - Meteorological data - Neighbouring PV plants	- NWP - Meteorological data - Neighbouring PV plants	- NWP - Neighbouring PV plants
Application	- Grid quality - Grid stability - Scheduling reserves - Demand response	- Load - following - Control of different load zones - Trading	- Planning - Unit commitment	- Transmission management - Trading and Planning - Asset optimization -Planning plant maintenance

forecasting methods are classified according to the time horizon, input data sources, and the role they play in a grid operation.

Precise local meteorological data, including irradiance, may be available from meteorological providers, however, procurement of the high-resolution weather data might be expensive, and it requires constant communication with weather providers, which might fail. Good quality meteorological data might not be available for every location where a PV system is installed. In the state of the art, they are used for intra-hour, short- and medium-term forecasts. Clear-sky models are usually fed with meteorological variables to the model. Clear-sky irradiance is the maximum theoretical irradiance at a certain point in space and time under clear-sky conditions. It is calculated as a deterministic variable (Ineichen, 2006), based on the geographical location and time of the year. Clear-sky irradiance contributes to more accurate PV power forecasting models by offering insights into the daily trends and seasonal effects.

2.1.1 Sky-imagers

Since the variability of PV power generation is correlated with cloud dynamics, different data sources have been used to track cloud movement and improve the accuracy of PV power forecasts. Ground-based cameras offer high spatial resolution of the images, which are analysed in order to identify and classify clouds. Then, the cloud motion is estimated, and the cloud location and velocity data are obtained (Kuhn et al., 2018; Song et al., 2022; Le Guen and Thome, 2020). Real-time irradiance measurements or clear-sky irradiance values are utilised with cloud motion vectors to predict solar irradiance and PV power production. They are often used for intra-hour forecasting due to their high spatial and temporal resolution. However, they are cite-specific, lack spatial coverage for a wide forecast area, and have high installation and maintenance costs (Si et al., 2021; Kumar et al., 2020).

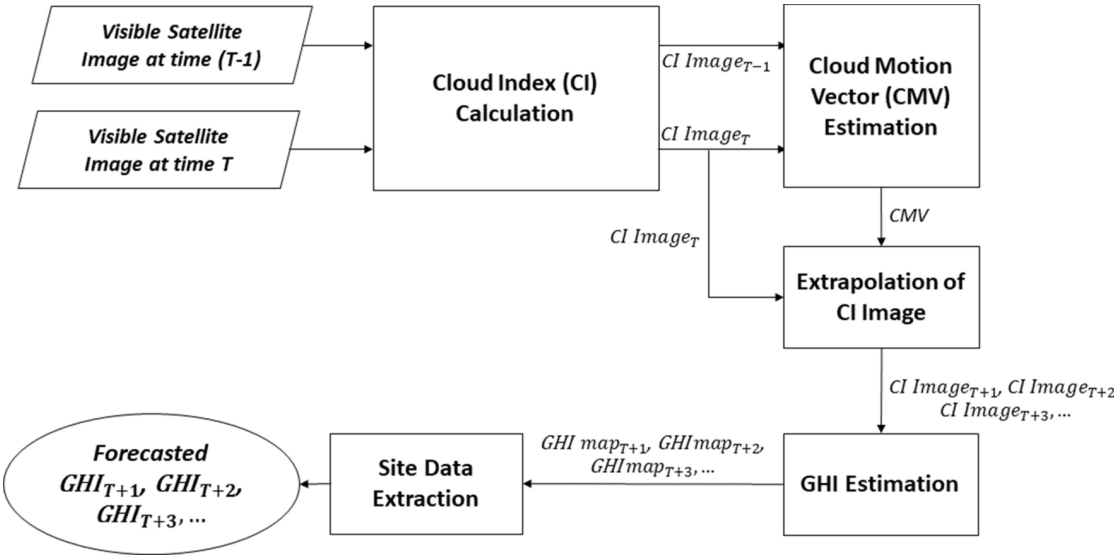


Figure 2.1: Schematic representation of the steps in irradiance forecasting using satellite images. Reproduced from Mitra et al. (2022).

2.1.2 Satellite-based images

Satellite-based images excel at intra-day forecasts, by addressing the problem of tracking cloud motions in a large region. Like sky-imager methods, satellite image-based methods first detect the clouds and calculate the cloud index as in Figure 2.1. Then, the particle image velocimetry, optical flows and other cloud-motion vector models are used in order to estimate the clouds (Quesada-Ruiz et al., 2014; Sirch et al., 2016). Once the cloud velocities are estimated, they are used to obtain the future cloud cover, which subsequently is utilized for forecasting solar irradiance or PV power generation. However, the assumption made is that cloud features do not change between two consecutive images, thus treating the clouds as rigid bodies which move in a straight line with a cloud motion vector (Si et al., 2021). Since the cloud concentration and cloud shape are dynamically changing, a non-linear model is needed to describe the cloud motion. What is more, the analysis and processing of satellite images are computationally expensive processes, leading up to 100 times slower forecast computations compared to non-linear machine learning models that utilize ground-based information (Carrillo et al., 2022). Another challenge with satellite-based models is related to cloud detection and representation of the PV station as a single pixel on the image. This might lead to huge deviation in PV power forecasting due to the prediction error of a pixel (Cheng et al., 2022). Satellite images do not have information about the irradiance and sun position, making it difficult to dynamically detect which cloud region is blocking the sunlight at certain PV station, thus requiring either ground-based measurement of irradiance or NWP data (Si et al., 2021).

2.1.3 Numerical weather predictions

Satellite images are often used in NWP models (Kumar et al., 2020). NWP uses atmospheric physics knowledge to propagate the cloud dynamics and forecast the cloud index and other weather variables. Traditionally, numerical models are employed to solve a set of differential equations, describing the underlying physics, and they obtaining the evolution of the weather conditions, including cloud movement, humidity, irradiance and temperature (Buizza, 2019). Then, predicted weather data is used to forecast PV power production accurately. NWP model are widely used in PV forecasting for medium- and long-term forecasts since they have a coarse spatial and temporal resolution. However, for intra-hour and intra-day forecasts, data-driven models are mostly preferred since they do not require prior assumptions, while offering short inference time (Chu et al., 2021). Furthermore, they can provide a data-driven approach for optimizing privacy-preserving data, which is highly important decentralized renewable energy sources forecasting (Sweeney et al., 2020).

2.1.4 Benchmark of PV forecasting models

Persistence and smart persistence models are often used as benchmarks for PV power forecasting tasks, due to their simplicity (Antonanzas et al., 2016). They represent traditional models for PV power forecasts. The persistence model assumes that PV power production remains the same between a time point t and $t+\Delta t$. It is assumed that the forecasted power for future time horizon is the same as the last measured value, or the same value of the previous day, at the same time of the day:

$$p(t+\Delta) = p(t) \tag{2.1}$$

where $p(t)$ represents the power produced at a specific PV plant at time t and $p(t+\Delta t)$ is the future PV power production. However, this only holds for stationary time series. This is why it is only applied for intra-hour forecasts (Antonanzas et al., 2016). Smart persistence was developed for longer horizons and it is represented as sum of the stationary and stochastic component of PV power production. In the work of Pedro and Coimbra (2012), it is defined as:

$$p(t+\Delta t) = \begin{cases} p_{clsky}(t+\Delta t) & \text{if } p_{clsky} = 0 \\ p_{clsky}(t+\Delta t)\frac{p(t)}{p_{clsky}(t)} & \text{otherwise} \end{cases} \tag{2.2}$$

where $p_{clsky}(t+\Delta t)$ is the future clear-sky irradiance at time $t+\Delta t$. However, it is not able to adjust the angle of the sunlight when cloud conditions are persistent within the forecasting time window (Kumler et al., 2019). In addition, it can not adapt to the advection since it assumes that the sky conditions remain constant (Kumler et al., 2018; Huertas Tato and Centeno Brito, 2018).

2.1.5 Short-term PV power forecasting

Although persistence and smart persistence models are often used as benchmarks, they are successful on intra-hour and short-term forecasts only when the consecutive power generation values are correlated (Persson et al., 2017). These models relay on ground-based PV power production data. Hence, many researchers for short-term PV power forecasting use additional data sources, including: ground-based cameras, satellite images, and numerical weather prediction (NWP) (Antonanzas et al., 2016). On one hand, NWP (Huang and Perry, 2015; Li et al., 2016; Sperati et al., 2016; Pierro et al., 2017) excel in long-term forecasts, but they perform rather poorly at short-term horizons and have coarse spatial and temporal resolution.On the other hand, satellite images are computationally expensive to process (Schmidt et al., 2017), whereas ground-based cameras are expensive to deploy and maintain over many PV stations, as already discussed.

Plethora of machine learning models which relay solely on ground-based PV power data when forecasting future production is developed. First linear models are developed, which include simple linear auto-regressive methods for intra-day and longer-term forecasts (Carriere et al., 2020). Linear models include vector autoregressive methods (Agoua et al., 2018; Vyas et al., 2022), auto-regressive (AR) and auto-regressive moving-average (ARMA) (Singh and Pozo, 2019). However, these models can not capture complex non-linear temporal patterns in the PV data (Zhang et al., 2022b). Moreover, the performance of linear and non-linear machine learning methods are compared in the work of Lauret et al. (2015). They show that non-linear machine learning methods outperform the simple linear models for forecasting horizons larger than one hour ahead.

Different non-linear machine learning models are proposed to improve the accuracy of ground-based PV power forecasting data. Non-linear models include different recurrent neural networks which are used to capture temporal patterns, particularly gated recurrent unit (GRU) and long-short term memory network (LSTM). Since clouds affect neighbouring PV stations sequentially, PV power production data is correlated in time and space. Thus, convolutional networks are proposed, on top of the recurrent neural networks, to capture spatial correlations (Lai et al., 2018; Dai et al., 2023). Furthermore, the attention mechanism, which had demonstrated significant success in natural language processing and other domains such as computer vision and machine translation, is also introduced in PV forecasting tasks. The ability to dynamically weigh the importance of different parts of the input sequence based on their relevance to observed value, enhanced the accuracy of PV forecasting tasks. Thus, it is often coupled with recurrent and convolutional networks and applied in PV forecasting task (Zhou et al., 2019; Shih et al., 2019).

While machine learning models achieved notable results in PV power forecasting, they often do not fully exploit the spatio-temporal relations and improve the forecast accuracy. Graph signal processing (GSP) perspective is taken in order to leverage data which lies on irregular grid and to capture their spatial relationships Sahili and Awad (2023). Spatio-temporal graph

forecasting is studied in various fields for spatio-temporal time series forecasting tasks, including traffic forecasting (Kwak et al., 2021), weather forecasting (Keisler, 2022), irradiance forecasting (Zhang et al., 2022a), wind speed and power forecasting (Li et al., 2023; Park and Park, 2019) among others. These architectures use recurrent neural networks to capture temporal correlation and graph convolution neural network to capture spatial correlations.

2.2 Graph Signal Processing

We review some relevant concepts in graph signal processing in order to formulate the PV forecasting problem as a time-series forecasting problem on graphs. A weighted undirected graph G is represented as a tuple $G = (v, \varepsilon, \mathbf{A})$, where $v = \{v_1, v_2, \ldots, v_N\}$ is its set of vertices (nodes) and ε its set of edges (links). If nodes v_i and v_j are connected, the edge between v_i and v_j is denoted by $e_{ij} \in \varepsilon$. The topology of the graph is determined by its symmetric adjacency matrix $\mathbf{A}$ of size $N \times N$. The matrix element A_{ij} gives the edge weight between vertices v_i and v_j and is zero in the absence of an edge e_{ij}. Another important operator is the Laplacian matrix $\mathbf{L}$, defined by

$$\mathbf{L} = \mathbf{D} - \mathbf{A}, \tag{2.3}$$

where $\mathbf{D}$ is the degree matrix. It represents the diagonal matrix of nodes' degrees:

$$D_{ii} = \sum_j A_{ij}. \tag{2.4}$$

where D_{ii} represents an entry of diagonal matrix. Normalised Graph Laplacian is usually used in the machine learning models:

$$\mathbf{L}_n = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} \tag{2.5}$$

Laplacian matrix is positive semidefinite and its eigendecomposition is defined with:

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T, \tag{2.6}$$

where $\mathbf{U}$ is a unitary matrix of eigenvectors and $\mathbf{\Lambda} \in \mathbb{R}^{N\times N}$ is the diagonal matrix of associated eigenvalues λ_i, $i = 1, \ldots, N$. Finally, we define a graph signal as a mapping $\mathbf{x} : v \to \mathbb{R}$, such that $x_v \in \mathbb{R}$ is the signal value at node v. The graph Fourier transform of a signal $\mathbf{x}$ is defined as

$$\dot{\mathbf{x}} = \mathbf{U}^\mathbf{T}\mathbf{x}. \tag{2.7}$$

The inverse Fourier transform is then obtained as

$$\mathbf{x} = \mathbf{U}\dot{\mathbf{x}}. \tag{2.8}$$

The graph Fourier transform of signal $\mathbf{x}$ enables the principles developed in classical signal processing to be extended on graphs, such as graph signal filtering, sampling graph signals and spectral analysis of graph signal. Therefore, we can define filtering of a graph signal $\mathbf{x}$ by

filter $\mathbf{h}$ as:

$$\mathbf{h}(\boldsymbol{L})\mathbf{x} = \mathbf{h}(\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T)\mathbf{x}, \tag{2.9}$$

where $\mathbf{h}(\boldsymbol{\Lambda})$ is the diagonal matrix with entries $h(\lambda_i) \in \mathbb{R}$, $i = 1, ..., N$. Finally, we can define in the graph Fourier domain graph convolution, of a signal $\mathbf{x}$ with a real function h as:

$$h *_{\mathcal{G}} \mathbf{x} := \mathbf{U}\mathbf{h}(\boldsymbol{\Lambda})\mathbf{U}^{\mathbf{T}}\mathbf{x}, \tag{2.10}$$

where $*_{\mathcal{G}}$ denotes the spectral graph convolution operation. This operation requires calculating the eigendecomposition of the graph Laplacian, which might be computationally very expensive for large graphs. Furthermore, it requires multiplication with the eigenvector matrix $\mathbf{U}$, leading to high complexity $\mathcal{O}(N^2)$. In addition, the size of the filters $\mathbf{h}$ defined in the spectral domain depends on the number of vertices and they are not localized.

To overcome these issues, the authors in Hammond et al. (2011) have proposed to parametrize the filter as:

$$\mathbf{h}(\boldsymbol{\Lambda}) = \sum_{k=0}^{K-1} \theta_k \boldsymbol{\Lambda}^k \tag{2.11}$$

where parameter $\theta \in \mathbb{R}^K$ is a vector of polynomial coefficients. Furthermore, in the works of (Hammond et al., 2011) is shown that the spectral polynomial filter with order of polynomial K are K-localized.

For more details and in-depth review of GSP we refer the reader to the work of Ortega et al. (2018).

2.3 Graph Convolutional Networks

Graph convolutional networks are used in our work to find the spatial correlations between PV stations, hence, we discuss it in more detail. In general, Graph Convolutional Neural Networks can be divided into two categories: spectral convolution (Ortega et al., 2018; Bruna et al., 2014; Defferrard et al., 2016) and spatial graph convolution (Duvenaud et al., 2015; Atwood and Towsley, 2016).

In the work of Defferrard et al. (2016), the authors have shown that using the localised polynomial filter, defined in 2.11 is still computationally expensive due to multiplication with the Fourier basis $\mathbf{U}$. They propose to solve this problem using a polynomial function that could be computed recursively, such as Chebyshev expansion.

Hence, the Chebyshev series expansion is combined together with the scaling of the Laplacian eigenvalues to parametrize and approximate $\mathbf{h}(\boldsymbol{\Lambda})$ as:

$$\mathbf{h}(\boldsymbol{\Lambda}) \approx \sum_{k=0}^{K-1} \theta_k \mathbf{T}_k(\tilde{\boldsymbol{\Lambda}}), \tag{2.12}$$

where $\theta_k \in \mathbb{R}$ are Chebyshev coefficients, $\tilde{\mathbf{\Lambda}} = 2\mathbf{\Lambda}/\lambda_{max} - \boldsymbol{I_N}$ is the scaled eigenvalue matrix, and $\mathbf{T}_k(\tilde{\mathbf{\Lambda}}) \in \mathbb{R}^{N\times N}$ is the diagonal matrix with diagonal entries the Chebyshev polynomial of order k applied to the scaled eigenvalues. Using functional calculus and plugging into (2.10), one finally gets

$$h *_{\mathcal{G}} \mathbf{x} \approx \sum_{k=0}^{K-1} \theta_k \boldsymbol{T_k}(\tilde{\boldsymbol{L}})\mathbf{x}, \tag{2.13}$$

where $\tilde{\boldsymbol{L}} = 2\boldsymbol{L}/\lambda_{max} - \boldsymbol{I_n}$ is the scaled Laplacian. The main practical advantage of the right side in (2.13) is to reduce the computation complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(K|\varepsilon|)$. Moreover, the graph convolutional filter represented with polynomials of order K of the scaled Laplacian is spatially localized and only depends on nodes that are K-hops away from the central node.

The authors have shown the ability of the proposed neural network to extract local features through convolutional layers, when trained for image and text classification tasks on a regular and irregular grids. Furthermore, they show that filter approximation with Chebyshev expansion of Graph Laplacian reduces the computational complexity of spectral graph convolution to a linear complexity. However, in these works the learned filters depend on the graph structure and can not be directly applied to the graph with different structured. Furthermore, the number of neighbours considered is always restricted to a local neighbourhood of K, which might represent a memory issue on densely connected graphs with large number of nodes.

2.4 Graph Attention Networks

In this work, we also use Graph Attention Networks (GAT), from the work of Veličković et al. (2018), to infer the correlation between nodes. Let a signal value at node $i \in N$ be represented with a column vector $\mathbf{x_i} = [x_i^1, \ldots, x_i^f] \in \mathbb{R}^f$, where f is the number of features per node. The attention mechanism is used to weight the importance of node j features to the node i. A shared matrix $\mathbf{W} \in \mathbb{R}^{f'\times f}$ is used to embed input features f into a f'-level feature space. Then the normalized attention coefficients α_{ij} are computed from (4.4):

$$\alpha_{ij} = softmax_j\left(l\left(\mathbf{a}\cdot\left[\mathbf{Wx_i}||\mathbf{Wx_j}\right]\right)\right) \tag{2.14}$$

where $\mathbf{a} \in \mathbb{R}^{2f'\times 1}$ is a row vector parametrizing the attention mechanism and $\cdot$ denotes dot product multiplication between vectors. A concatenation is represented with $||$ and $l(\cdot)$ denotes the activation function LeakyReLU. Finally, the obtained normalized attention coefficients are used to compute the final output $\mathbf{h_i}$ for every node (4.5):

$$\mathbf{h_i} = \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}\mathbf{Wx_j}\right) \tag{2.15}$$

where $\mathcal{N}_i$ represents the neighbourhood of node i and $\sigma(\cdot)$ is the LeakyReLu non-linearity.

In order to stabilise the learning process of self-attention, the authors have proposed multi-head attention. Attention operation is repeated K times with different parameters, before

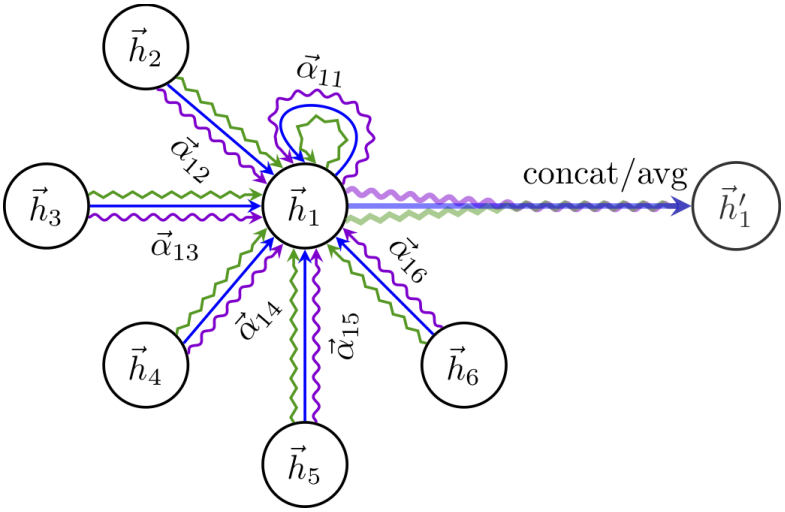


Figure 2.2: The architecture of the GAT layer with multi-head attention. Reproduced from Veličković et al. (2018).

aggregating the output features $\mathbf{h}'_\mathbf{i}$:

$$\mathbf{h}'_\mathbf{i} = \|_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \mathbf{x}_\mathbf{j}\right) \tag{2.16}$$

where α_{ij}^k are attention coefficients in k-th attention head, and wight matrix $\mathbf{W}^k$ are learnt for each head. The output $\mathbf{h}'$ contains the aggregated information from each attention head, see Figure 2.2.

Graph attention networks are computationally and memory efficient, since the number of parameters does not depend on the number of nodes and edges. They implicitly allow model to assign different weights to different nodes in the same neighbourhood. This allows them to capture correlations among the data and to be utilized in different deep learning models in various tasks, including traffic forecasting Guo et al. (2019), recommender systems Wang et al. (2019), drug discovery Jiménez-Luna et al. (2020), and many others.

Although, they are initially developed for a node classification task, they could be extended for graph classification tasks, as well as time-series forecasting task. Another research direction that is not addressed by authors is thorough analysis on the model interpretability, see the work of Veličković et al. (2018) for more detail.

2.5 Long-short term memory network

Graph attention and graph convolutional networks are widely used to model spatial relationships between the data points. On the other hand, the recurrent neural networks are designed to find the patterns and process sequential data. Hence, they are suitable for capturing the tem-

poral dynamics in time series data. Two popular types of recurrent neural networks are Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997). They are used in conjecture with convolutional and attention models in time series forecasting. The gating mechanism is introduced in LSTM in order to address the vanishing gradient problem, from which classical recurrent neural networks suffer. The gates enable LSTM cells blocks to forget irrelevant information, while keeping the important information for a longer period of time. They are designed to handle time-series data with the longer dependencies. These gates can capture both the long-term and short-term relations between the points in time series data.

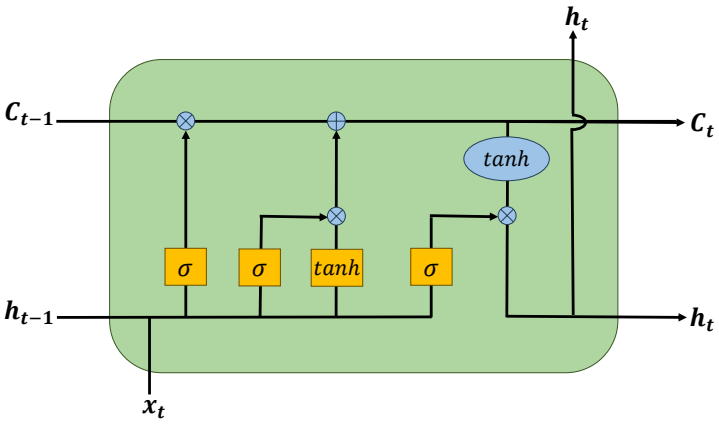


Figure 2.3: The architecture of the LSTM cell.

LSTMs have 4 important components: cell state, forget, input and output gate, shown in Figure 2.3. The forget gate uses a sigmoid activation function to decide whether to keep the current information via:

$$\mathbf{f}(\tau) = \sigma(\mathbf{W}_{f,h} \cdot \mathbf{h}(\tau - 1) + \mathbf{W}_{f,\mathrm{x}} \cdot \mathbf{x}(\tau) + \mathbf{b}_f). \tag{2.17}$$

The input gate considers new information to the LSTM and the hidden state from the previous time step. Then it decides whether to update with the new information the current cell state:

$$\mathbf{i}(\tau) = \sigma(\mathbf{W}_{i,h} \cdot \mathbf{h}(\tau - 1) + \mathbf{W}_{i,\mathrm{x}} \cdot \mathbf{x}(\tau) + \mathbf{b}_i), \tag{2.18}$$

where sigmoid activation function decide what percentage of the information is used for update. The cell state is used to store long-tern memory, and it is updated by both forget gate and the input gate:

$$\mathbf{c}(\tau) = \mathbf{i}(\tau) \otimes \tilde{\mathbf{c}}(\tau) + \mathbf{f}(\tau) \otimes \mathbf{c}(\tau - 1), \tag{2.19}$$

where $\tilde{\mathbf{c}}(\tau)$ represents the new information from the input gate that is taken into account

$$\tilde{\mathbf{c}}(\tau) = \tanh(\mathbf{W}_{c,h} \cdot \mathbf{h}(\tau - 1) + \mathbf{W}_{c,\mathrm{x}} \cdot \mathbf{x}(\tau), \tag{2.20}$$

when updating the memory state. Finally, the hidden state of LSTM cell is defined:

$$\mathbf{h}(\tau) = \mathbf{o}(\tau) \otimes \tanh(\mathbf{c}(\tau)) \tag{2.21}$$

where σ is the sigmoid function and $\otimes$ is the Hadamard product.

LSTM networks overcome vanishing gradient problem, from which simple recurrent neural networks suffer when modelling dependencies in the long-term sequences (Bengio et al., 1994). LSTMs use the forget gate to remove the information that is irrelevant to the model while deciding which information is important and stored in the models cell state. Thus, they control the information and gradient flow.

LSTM networks have been used in various tasks, including: traffic speed and congestion forecasting, renewable energy production forecasting, equipment fault diagnosis, classification of medical diagnosis and many others (Liu et al., 2022; Li et al., 2017; Xiao et al., 2023; Lipton et al., 2016). Most notably, they are used in time-series forecasting task, due to ability to handle longer-term sequences and capture longer-term dependencies.

Even though LSTMs are able to capture correlations across longer sequences, compared to vanilla recurrent neural networks, they still diminish the information which was stored much earlier, when processing long sequences. Additionally, they still might suffer from exploding gradient, when input sequences are extremely long.

3 Spatio-temporal graph neural networks for multi-site PV power forecasting

3.1 Introduction

Intra-day PV power production forecasting methods often combine additional data source, which might have coarse spatial and temporal resolution. Moreover, processing the additional data sources can be computationally expensive. State-of-the-art PV power production forecasting methods use inputs from various sources in order to improve the accuracy of the forecast, in particular: ground-based cameras (Chu et al., 2015), satellite images (Jang et al., 2016; Schmidt et al., 2017), and NWP (Antonanzas et al., 2016). Ground-based cameras are expensive to deploy and maintain; thus, they are better suited for intra-hour forecasts. Oppositely, NWPs excel at longer-term forecasts but usually have coarse spatial and temporal resolution. Satellite images improve the accuracy of short-term forecasts; however, they are computationally expensive. Therefore, in order to avoid the issues brought by the additional exogenous data, the question arises whether it is possible to achieve state-of-the-art results relying only on past PV power generation data.[1]

Different classes of machine learning models have previously been reported to investigate this question. Traditional approaches, based on auto-regressive (AR) linear models, outperform persistence model (Agoua et al., 2018; Carrillo et al., 2020). However, these models are outperformed by non-linear neural network models, which rely on recurrent and convolutional neural networks. The recurrent structures are finding the temporal correlations between PV power data. Since passing clouds influence neighbouring PV sites sequentially, the cloud cover and movements can be captured by considering spatial relations between PV stations. Consequently, convolutional neural networks (CNN) and attention mechanisms have been proposed to capture spatial correlations (Zhou et al., 2019; Jeong and Kim, 2019). Although these models are able to capture complex patterns, they do not fully exploit the spatio-temporal information of multiple sites.

Graph signal processing is a recent framework that allows the processing of signals defined

[1] The content of this Chapter is based on the publication (Simeunović et al., 2022a).

in irregular domains by using graphs to capture their interdependence (Ortega et al., 2018). Recently, graph neural networks (GNN) have attracted a lot of attention due to their expressive power and ability to infer information from complex data such as brain signals, social network interactions and traffic congestion patterns (Zhou et al., 2020; Wu et al., 2020). Recurrent neural networks coupled with graph convolutional structures were recently proposed for wind speed forecasting (Khodayar and Wang, 2019). The main drawback of this approach is that it requires one model for each step ahead in the prediction horizon, which is not sample efficient (needing around four years of data for training) and not scalable for a large number of nodes. Graph models have also been used to produce probabilistic forecasts for solar irradiance in (Khodayar et al., 2019), which proposed a graph convolutional auto-encoder to model the irradiance's probability distribution at node level in a scalable fashion.

We take a GSP perspective and model the PV production time-series as signals on a graph. The intuition behind this choice is that for a sufficiently dense network of PV systems, graph-based models can exploit the spatio-temporal dependencies of PV production data to infer part of the cloud dynamics and forecast production more accurately. Multi-site photovoltaic production time series are modelled as signals on a graph in order to achieve higher spatial and temporal resolution forecasts. We present two novel spatio-temporal GNN models for deterministic multi-site PV power forecasting which rely entirely on production data: the Graph-Convolutional Long Short Term Memory (GCLSTM) and the Graph-Convolutional Transformer (GCTrafo) models. Both models use graph convolutional layers to infer the spatial patterns from the data though they use different structures to model the time dependence: GCLSTM uses recurrent structures, whereas GCTrafo uses attention mechanisms. The proposed models are compared with state-of-the-art methods for deterministic multi-site PV forecasting, for a forecasting horizon of six hours ahead, over an entire year in two datasets distributed over Switzerland: (1) production data from 304 real PV systems, and (2) simulated production of 1000 PV systems. Additionally, the proposed forecasting models are compared with single-site state-of-the-art forecasting methods that use NWP as inputs for two sites also in Switzerland.

The rest of the Chapter is organized as follows. Section 5.2 introduces preliminaries on graph convolution and graph time series forecasting of PV generation. Section 5.3 details the proposed GCLSTM and GCTrafo GNN architectures. Experimental results of our evaluation are presented and discussed in Section 5.4. Finally, we conclude this Chapter in Section 5.5.

3.1.1 Related work

Researchers use various data sources, including ground-based cameras (Chu et al., 2015), satellite images (Jang et al., 2016; Schmidt et al., 2017), and NWP (Antonanzas et al., 2016) to improve the accuracy of the short-term PV power production forecast. Ground-based cameras are expensive to deploy and maintain in a grid with a large number of PV stations. Furthermore, they yield high accuracy only for intra-hour forecasts. On the other hand, satellite images

are more suitable for regional forecasting when PV stations are clustered, since the wide-area images are inadequate for providing site-specific information. Methods that combine satellite images with NWP (Huang and Perry, 2015; Li et al., 2016; Sperati et al., 2016; Pierro et al., 2017) excel in long-term forecasts, but they perform rather poorly at short-term horizons and high spatial resolution. Precise local numerical weather forecasts may be accessible by dedicated meteorological providers, but are often very costly to acquire and require heavy processing.

In order to avoid issues coming from the additional data source, data-driven models that use only PV power production without additional sources are developed. The traditional auto-regressive (AR) linear models model which use only PV power production data are proposed in works of Yang et al. (2015). These were further extended to vector auto-regressive (VAR), Lasso-VAR (Cavalcante and Bessa, 2017; Agoua et al., 2018), graph-based spatio-temporal AR (Carrillo et al., 2020) and auto-regressive moving average (ARM) models (Singh and Pozo, 2019). Simple non-linear neural networks, however, outperform persistence model and simple linear methods for forecasting horizons longer than one hour (Lauret et al., 2015). Nonlinear neural network models include recurrent and convolutional neural networks. The recurrent structures with long short-term memory (LSTM) network (Ghaderi et al., 2017), (Lai et al., 2018; Lee et al., 2018) perform well at capturing temporal patterns. Since passing clouds influence neighbouring PV sites sequentially, the cloud cover and the cloud movements can be captured by considering spatial and temporal relations between PV stations. For that purpose, convolutional neural networks (CNN) have been proposed to extract the spatio-temporal correlations by stacking the PV signals as an image and reordering their position in the image based on their location (Jeong and Kim, 2019; Zhu et al., 2018). In addition, attention mechanisms have been also introduced to capture spatial correlations (Zhou et al., 2019; Shih et al., 2019). One of the main advantages of (Shih et al., 2019) is the high accuracy for different spatio-temporal forecasting tasks including electricity, PV, exchange rate and traffic forecasting, without tailoring the model to a specific task. Spatio-temporal forecasting models have been mainly applied for the traffic speed forecasting problem. LSTMs have been used to capture temporal correlations, while convolutional and attention structures have been proposed to capture spatial relations (Lai et al., 2018; Shih et al., 2019). Although these models are complex, they use only a limited number of data steps from previous days as input to the model, thus, neglecting the temporal shift and periodicity. Therefore, bidirectional LSTMs have been proposed to exploit not only forward dependencies but also backward dependencies (Cui et al., 2018). Bidirectional LSTM structures have also been used in (Toubeau et al., 2021) to improve the probabilistic forecast of distribution locational marginal prices by accessing long-term dependencies. One drawback of the latter work is that the spatial information needs to be carefully encoded and concatenated as features to the input data and the bidirectional LSTM is used to implicitly learn the spatial relations.

Although the aforementioned works exploit spatio-temporal correlations, they do not fully exploit the spatial information of multiple sites. Recently, graph neural networks (GNN) have attracted a lot of attention due to their expressive power and ability to infer information from complex data such as brain signals, social networks interactions and traffic congestion

patterns (Zhou et al., 2020; Wu et al., 2020). Spatio-temporal graph forecasting has been studied in various fields, including traffic forecasting, weather forecasting and price forecasting among others. Spatio-temporal techniques in the traffic speed forecasting use gated graph convolutional structures (Yu et al., 2018) and encoder-decoder recurrent diffusion convolution (Li et al., 2017) to capture spatial and temporal correlations. However, these models only predict one step ahead in one iteration and then use predictions as historical observations, which increases the error for longer-term predictions. This problem was addressed in (Zhang et al., 2019; Zheng et al., 2020) that use attention mechanisms for multi-step prediction. However, traffic speed forecasting has a predefined graph topology, constructed from the road network, which makes the problem easier in comparison to PV or wind speed forecasting, where the correlations and connections between PV (or wind) systems are not known in advance. LSTMs coupled with graph convolutional structures for capturing spatio-temporal patterns were recently proposed for wind speed forecasting (Khodayar and Wang, 2019). This mechanism requires a different LSTM network for each site to learn the temporal relations followed by graph convolutional layers to learn the spatial dependencies.

3.2 Problem formulation

3.2.1 Graph convolution

We have reviewed some relevant concepts of graph signal processing in Chapter 2. In the multi-site PV case, each PV station corresponds to a node in the graph G and edges might represent the spatial proximity between the PV stations v_i and v_j or other relationship between stations. We define a graph signal as a mapping $\mathbf{x}: v \rightarrow \mathbb{R}$, such that $x_v \in \mathbb{R}$ is the signal value at node v. In our case the graph signal $\mathbf{x}$ represents the vector containing the power production of all PV stations at some point time.

The spectral graph convolution is defined in the graph Fourier domain: for a real function h, the graph convolution of a signal $\mathbf{x}$ with the function h is defined by:

$$h *_{\mathcal{G}} \mathbf{x} := \mathbf{U}\mathbf{h}(\mathbf{\Lambda})\mathbf{U}^{\mathbf{T}}\mathbf{x}, \tag{3.1}$$

where $\mathbf{h}(\mathbf{\Lambda})$ is the diagonal matrix with entries $h(\lambda_i) \in \mathbb{R}$, $i = 1, ..., N$. In (Defferrard et al., 2016), the authors use Chebyshev series expansion together with the scaling of the Laplacian eigenvalues to parametrize and approximate $\mathbf{h}(\mathbf{\Lambda})$. Furthermore, using functional calculus, as shown in Chapter 2, and plugging into (3.1), one finally gets

$$h *_{\mathcal{G}} \mathbf{x} \approx \sum_{k=0}^{K-1} \theta_k \boldsymbol{T_k}(\tilde{\boldsymbol{L}})\mathbf{x}, \tag{3.2}$$

where $\tilde{\boldsymbol{L}} = 2\boldsymbol{L}/\lambda_{max} - \boldsymbol{I_n}$ is the scaled Laplacian. The main practical advantage of the right side in (3.2) is to reduce the computation complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(K|\varepsilon|)$. See (Defferrard et al., 2016) for a detailed discussion. In the rest of this Chapter, we consider filters h for which an

equality in (3.2) holds, and make the abuse of notation

$$\theta *_{\mathcal{G}} \mathbf{x} \equiv h *_{\mathcal{G}} \mathbf{x}.$$

Moreover, we consider the extension of (3.2) to multivariate graph signals $\mathbf{x}_v = (x_v^{(1)}, \ldots, x_v^{(\mathrm{f_{in}})}) \in \mathbb{R}^{\mathrm{f_{in}}}$ and replace $\theta \in \mathbb{R}^K$ by $\mathbf{W} \in \mathbb{R}^{K \times \mathrm{f_{out}} \times \mathrm{f_{in}}}$, where f_{in} and f_{out} denote the number of input and output features, respectively, in the layer. Thus, for any output feature j in $\mathrm{f_{out}}$,

$$(\mathbf{W} *_{\mathcal{G}} \mathbf{x})^{(j)} = \sum_{k=0}^{K-1} \sum_{i=0}^{\mathrm{f_{in}}} W_k^{(ji)} \boldsymbol{T_k}(\tilde{\boldsymbol{L}}) \mathbf{x}^{(i)}. \tag{3.3}$$

The neural network architectures presented in this Chapter use in some layers the operation defined in (3.3), the weights $\mathbf{W}$ being learnable parameters as in standard CNN.

3.2.2 Multi-site time-series forecasting on graphs

Atmospheric clouds act as a dynamic mask that affects local PV power production. For a sufficiently dense network of PV plants, parts of this dynamics (diffusion and advection) can be inferred from past production data and used to predict future production on the entire network. Suppose we have N PV stations. Thus, each station corresponds to a node in the network graph and the observed PV data are temporal signals attached to each node. The edge weight between two nodes is a measure of the expected correlation between two sites. Typical choices are bivariate (Pearson) correlation, distance correlation and different kernel-based methods (Kriege et al., 2020).

Let $\mathbf{p}(t) \in \mathbb{R}^N$ denote the vector of PV power production over all PV stations at time step t with the value at node v being denoted by $p_v(t)$. Formally, we want to forecast $\mathbf{p}(t)$ for the next H discrete time steps ahead given M past observations as:

$$\hat{\mathbf{p}}(t), \ldots, \hat{\mathbf{p}}(t+H-1) = f_\beta\left(\mathbf{p}(t-M), \ldots, \mathbf{p}(t-1)\right), \tag{3.4}$$

where f_β is a chosen family of parametric estimators. The learning problem consists in finding a set of parameters β that minimizes the prediction error over the entire horizon by solving

$$\arg\min_{\beta} \sum_{t \in \mathcal{T}} \sum_{\tau=t}^{t+H-1} \|\hat{\mathbf{p}}(\tau) - \mathbf{p}(\tau)\|_2^2, \tag{3.5}$$

where $\mathcal{T}$ is the set of times of past observations taken into consideration to fit the model (training set).

3.3 Graph convolutional forecasting models

In this section we present two sequence-to-sequence forecasting models based on spectral graph convolutions. Both share the same structure: an encoder to process the past M observed data and a decoder to predict the next H future observations.

3.3.1 Graph convolutional long-short term memory neural network

The first architecture is a sequence to sequence model based on graph convolutional long-short term memory (GCLSTM) (see (Hochreiter and Schmidhuber, 1997) for LSTM networks, (Bresson and Laurent, 2017) for graph convolutional recurrent networks). Both the encoder and decoder combine recurrence and spectral graph convolution to model jointly temporal and spatial correlations. The encoder is a GCLSTM network that estimates the state of the system, given a sequence of past observations, its initial state being set to zero. The decoder is another GCLSTM cell that is initialized with the final encoder state and predicts the power for the chosen horizon period of H steps ahead; see Figure 3.1. A multi-layer perceptron (MLP) is used at the output of the decoder to transform the GCLSTM outputs into the desired power production $\hat{\mathbf{p}}(\tau)$, where $\tau \in \{t, \dots, t+H-1\}$. The specific inputs features $\mathbf{x}(\tau)$ and $\mathbf{y}(\tau)$, concatenations of power and clear sky irradiance signals, are presented at the end of the section.

The usage of LSTM cells as recurrent structures of the model is justified by their capacity of learning and retaining both short - and long-term dependencies; see (Hochreiter and Schmidhuber, 1997). We denote by lat the number of dimensions of the LSTM cell latent representation. In the classical LSTM cell, the cell state $\mathbf{c}(\tau) \in \mathbb{R}^{\text{lat}}$ and the output $\mathbf{h}(\tau) \in \mathbb{R}^{\text{lat}}$ are updated recursively from the input sequence $\mathbf{x}(\tau) \in \mathbb{R}^{\text{f}_{\text{in}}}$ using gating operations involving matrix multiplications. In GCLSTM cells, $\mathbf{c}(\tau) \in \mathbb{R}^{N\times\text{lat}}$, $\mathbf{h}(\tau) \in \mathbb{R}^{N\times\text{lat}}$, $\mathbf{x}(\tau) \in \mathbb{R}^{N\times\text{f}_{\text{in}}}$ and the gating operations are modified by replacing the matrix multiplications with spectral graph convolutions as defined in (3.3). Doing so, signals are diffused across neighbouring nodes and local spatial information is better captured (see (Defferrard et al., 2016) ,(Seo et al., 2018)). For a given input sequence $(\mathbf{x}(\tau))_\tau$, the GCLSTM cell equations are given by

$$
\begin{aligned}
\mathbf{f}(\tau) &= \sigma(\mathbf{W}_{f,h} *_{\mathcal{G}} \mathbf{h}(\tau-1) + \mathbf{W}_{f,\text{x}} *_{\mathcal{G}} \mathbf{x}(\tau) + \mathbf{b}_f) \\
\mathbf{i}(\tau) &= \sigma(\mathbf{W}_{i,h} *_{\mathcal{G}} \mathbf{h}(\tau-1) + \mathbf{W}_{i,\text{x}} *_{\mathcal{G}} \mathbf{x}(\tau) + \mathbf{b}_i) \\
\mathbf{o}(\tau) &= \sigma(\mathbf{W}_{o,h} *_{\mathcal{G}} \mathbf{h}(\tau-1) + \mathbf{W}_{o,\text{x}} *_{\mathcal{G}} \mathbf{x}(\tau) + \mathbf{b}_o) \\
\mathbf{c}(\tau) &= \mathbf{i}(\tau) \otimes \tanh(\mathbf{W}_{c,h} *_{\mathcal{G}} \mathbf{h}(\tau-1) + \mathbf{W}_{c,\text{x}} *_{\mathcal{G}} \mathbf{x}(\tau) + \mathbf{b}_c) \\
&\quad + \mathbf{f}(\tau) \otimes \mathbf{c}(\tau-1) \\
\mathbf{h}(\tau) &= \mathbf{o}(\tau) \otimes \tanh(\mathbf{c}(\tau))
\end{aligned}
\tag{3.6}
$$

where σ is the sigmoid function, $\mathbf{W} *_{\mathcal{G}} \cdot$ is defined in (3.3) and $\otimes$ is the Hadamard product. The dimension of the weights $\mathbf{W}_{\cdot,\cdot}$ and biases $\mathbf{b}_\cdot$ are determined by the number of dimensions of

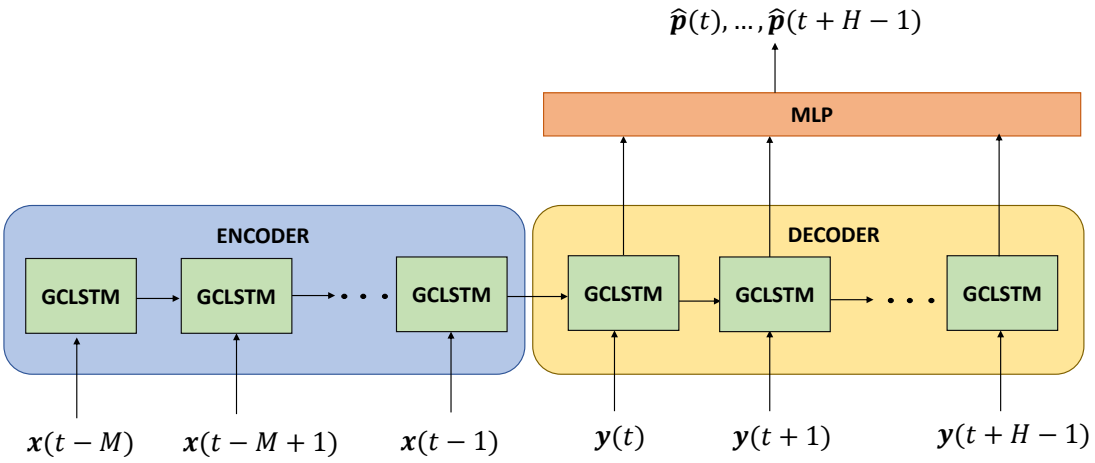


Figure 3.1: Encoder-decoder Graph convolutional LSTM architecture.

the input feature space, $\mathrm{f_{in}}$, the latent space, lat, and the order K of the Chebyshev expansion: $W_{\cdot,h} \in \mathbb{R}^{K\times \mathrm{lat}\times \mathrm{lat}}$ and $W_{\cdot,\mathrm{x}} \in \mathbb{R}^{K\times \mathrm{lat}\times \mathrm{f_{in}}}$, the biases being in $\mathbb{R}^{K\times \mathrm{lat}}$.

The adjacency matrix $\mathbf{A}$ of the graph is initialized using the k-nearest neighbours algorithm: $A_{ij} = 1$ if v_i and v_j are nearest neighbours, and 0 otherwise. The scaled Laplacian $\tilde{\mathbf{L}}$ involved in the graph convolutions in (3.6) is calculated initially from $\mathbf{A}$ and represented as a sparse tensor. In the course of training, not only weights and biases in (3.6) are learnt, but also the non-zero entries of the sparse Laplacian in each cell operation, so as to capture very local specifics related, for instance, to the topology of the terrain or nodes separation distance. Encoder and decoder are trained simultaneously.

The input sequence of the encoder consists of tuples $\mathbf{x}(\tau) = (\mathbf{p}(\tau), \bar{\mathbf{p}}(\tau), \mathbf{g}(\tau))$, $\tau \in \{t-M, \ldots, t-1\}$, where $\mathbf{p}(\tau) \in \mathbb{R}^N$ is the power produced at time τ, $\mathbf{g}(\tau) \in \mathbb{R}^N$ the global clear sky irradiance at time τ and $\bar{\mathbf{p}}(\tau) \in \mathbb{R}^N$ is the rolling mean power produced over the interval $[\tau - 72h, \tau - 24h]$. The clear sky irradiance values are computed at any location and any time on the map using the Ineichen and Perez clear sky model from PVlib (Stein et al., 2016). This computation is deterministic and only relies on the geographical coordinates of the nodes (latitude, longitude and altitude). Similarly, inputs to the decoder are sequences of $\mathbf{y}(\tau) = (\mathbf{g}(\tau), \mathbf{d}(\tau), \bar{\mathbf{p}}(\tau))$, where $\mathbf{d}(\tau)$ is the direct clear sky irradiance at time $\tau \in \{t, \ldots, t+H-1\}$.

3.3.2 Graph convolutional transformer

Even if the gate operations in (3.6) protect the cell state $\mathbf{c}(\tau)$ and allow it to keep information over time, this information has the tendency to fade and be diluted (Schoene et al., 2020). This shortcoming has been addressed in recent natural language processing architectures, in particular in the sequence to sequence model presented in (Vaswani et al., 2017), called

transformer. In transformers, access to past signals at any time is guaranteed thanks to the usage of dot-product attention between any two elements of the time sequence. The second architecture presented in this Chapter, dubbed graph convolutional transformer (GCTrafo), is inspired by the base transformer architecture but incorporates a slight number of modifications so as to make it suitable for the multi-site PV generation forecasting problem. As the GCLSTM presented in Section 3.3.1, the GCTrafo architecture is made of an encoder to process past signal values and a decoder to predict the future outcomes. Moreover, it shares the same encoder input, output and decoder output signals. However, the inner operations are quite different and the GCTrafo does not incorporate any recurrent structure.

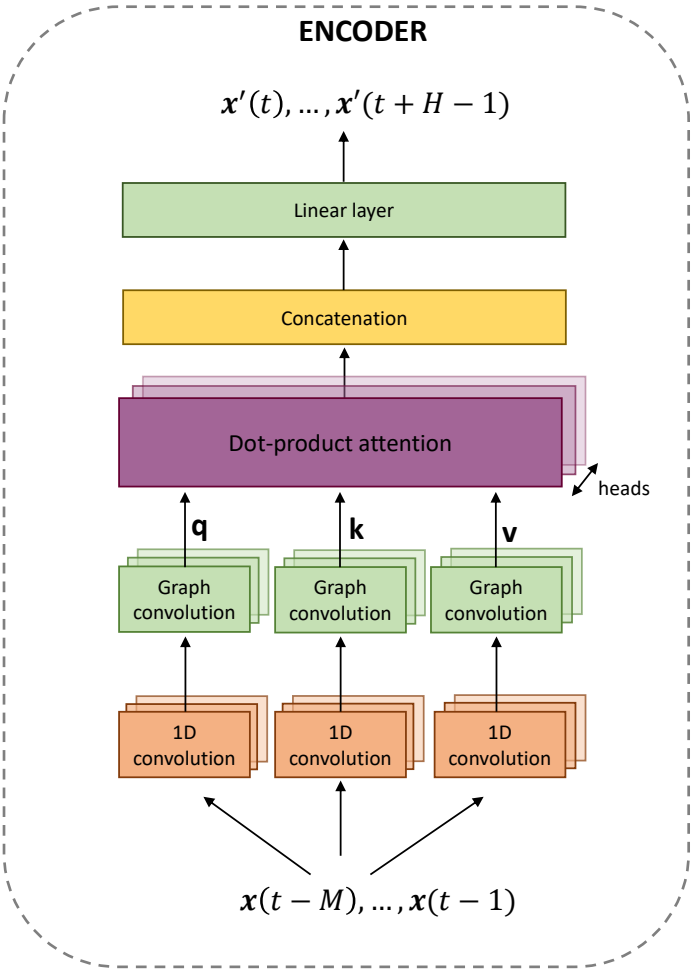


Figure 3.2: Encoder of Graph Convolutional Transformer architecture (GCTrafo).

The encoder consists of three main stages, see Figure 3.2. At the first stage, a 1D-convolutional layer is applied to the input sequence $(\mathbf{x}(\tau))_\tau$, where $\tau \in \{t-M, \ldots, t-1\}$, along the time axis to extract valuable variation features of the raw signals at single node level. This operation is made 3 times in parallel and produces three output sequences $(\tilde{\mathbf{x}}(\tau))_\tau$, $(\check{\mathbf{x}}(\tau))_\tau$ and $(\hat{\mathbf{x}}(\tau))_\tau$. At the second stage, a dot product attention mechanism preceded by graph convolutions is used to embed every node variation feature in its spatio-temporal context. Queries $\mathbf{q}(\tau) \in \mathbb{R}^{N \times \text{lat}}$,

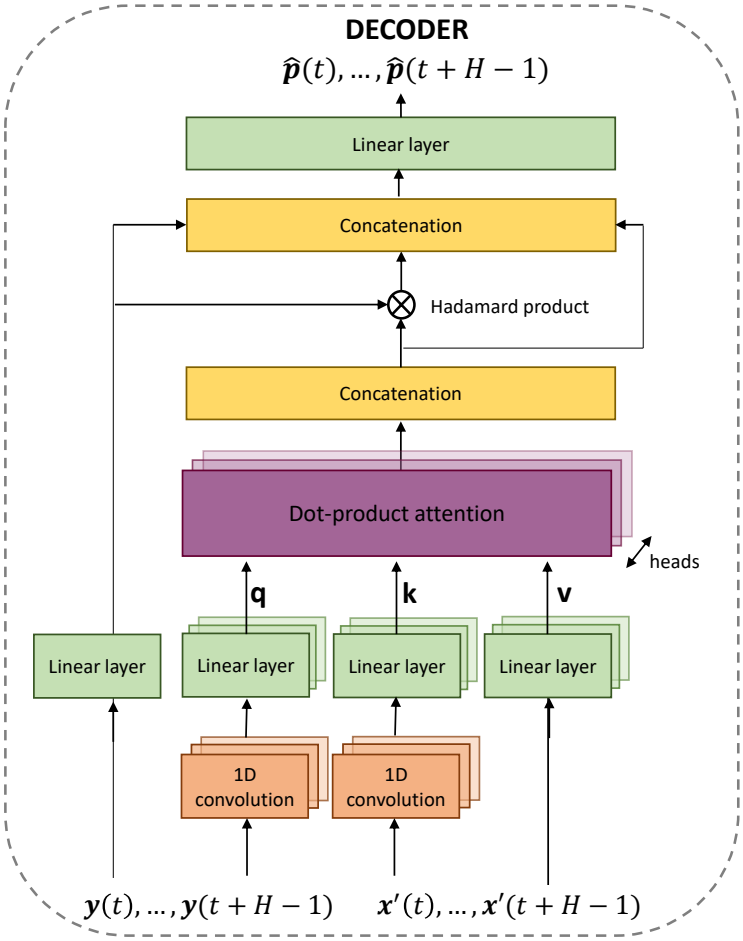


Figure 3.3: Decoder of Graph Convolutional Transformer architecture (GCTrafo).

keys $\mathbf{k}(\tau) \in \mathbb{R}^{N \times \text{lat}}$ and values $\mathbf{v}(\tau) \in \mathbb{R}^{N \times \text{lat}}$ signals are first created by graph convolutional layers based on (3.3), extracting neighbouring node information:

$$\begin{aligned} \mathbf{q}(\tau) &= \mathbf{W_q} *_{\mathcal{G}} \tilde{\mathbf{x}}(\tau), \\ \mathbf{k}(\tau) &= \mathbf{W_k} *_{\mathcal{G}} \check{\mathbf{x}}(\tau), \\ \mathbf{v}(\tau) &= \mathbf{W_v} *_{\mathcal{G}} \check{\mathbf{x}}(\tau). \end{aligned} \tag{3.7}$$

Queries, keys and values are then fed to a softmax dot product attention layer (see (Vaswani et al., 2017)), given for the query $\mathbf{q}(\tau)$ at time τ and node v by

$$\begin{aligned} &\mathbf{att}(\mathbf{q}(\tau), (\mathbf{k}(\tau'))_{\tau'}, (\mathbf{v}(\tau'))_{\tau'})_v = \\ &\qquad \sum_{\tau'} \frac{\exp(\mathbf{q}_v(\tau) \cdot \mathbf{k}_v(\tau'))}{\sum_{\tau''} \exp(\mathbf{q}_v(\tau) \cdot \mathbf{k}_v(\tau''))} \mathbf{v}_v(\tau') \in \mathbb{R}^{\text{lat}}. \end{aligned} \tag{3.8}$$

In (3.8), the dot product inside the exponents contracts the latent dimensions to produce a

scalar for each node v. A multi-head approach similar to (Vaswani et al., 2017) is used: the 1D convolutions, graph convolutions and dot product attention layers are duplicated n times and concatenated before being fed to the last linear layer, producing the encoded sequence $(\mathbf{x}'(\tau))_\tau$. The intuition behind multi-head attention mechanism is that each head will learn to focus at different patterns of the input sequence.

The decoder operations are similar to the encoder with specifics related to the input sequence of the decoder, see Figure 3.3. The main difference is the absence of graph convolution in the decoder before the attention layer. Indeed, the input sequence of the decoder consists of clear sky irradiance values and rolling mean values whose propagation across neighbouring nodes is not expected to add any further useful information. Moreover, the value $\mathbf{v}(\tau)$ is directly set to be equal to the encoded vector $\mathbf{x}'(\tau)$, without any prior layer mapping; see Figure 3.3. Finally, viewing the output of the attention layer of the decoder as a vector encoding shading information coming from the cloud dynamics, this vector is concatenated with an embedding of the input $\mathbf{y}(\tau)$ at time $\tau \in \{t, \dots, t+H-1\}$ and its Hadamard product with this embedding before the last linear layer. The last layer produces the output power production $\hat{\mathbf{p}}(\tau)$, where $\tau \in \{t, \dots, t+H-1\}$.

During training, we adopt a similar strategy as for the GCLSTM encoder-decoder: all weights are learnt by stochastic gradient descent, and the non-zero entries of the scaled Laplacian operator entering in the convolutions in (3.3) are learnt during training, starting with values equal to the ones derived using the k-nearest neighbour algorithm for the adjacency matrix.

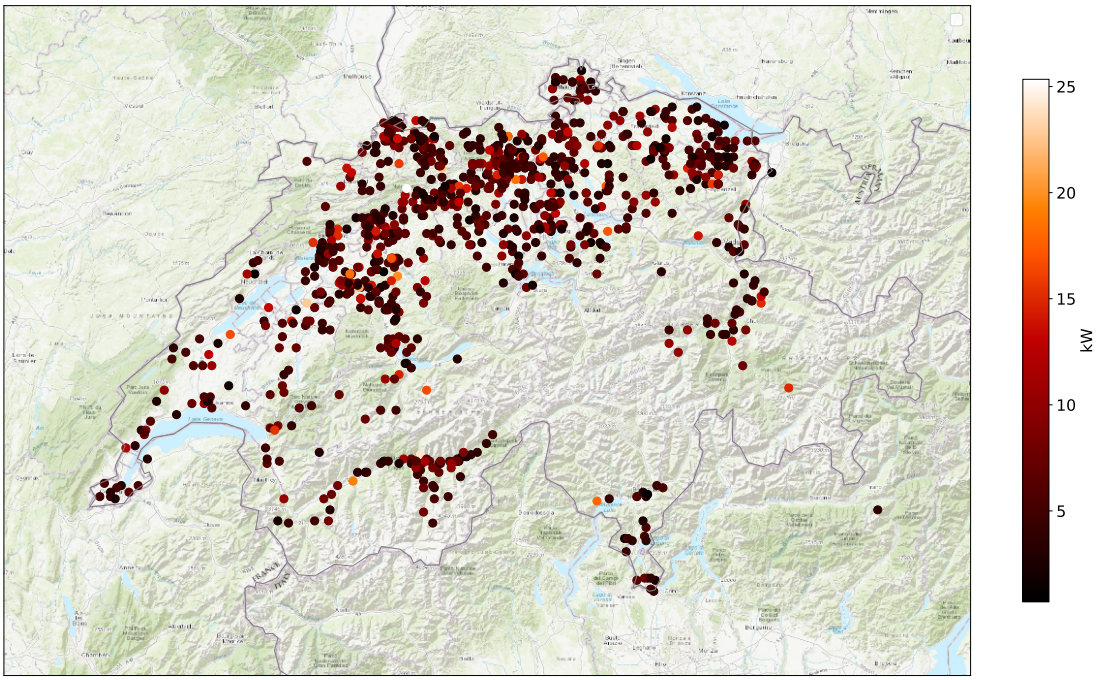
3.4 Experimental Results

In this section, GCLSTM and GCTrafo architectures are evaluated on two datasets, for both multi-site and single-site forecasts. In the following we describe the experimental setting first and then present the results.

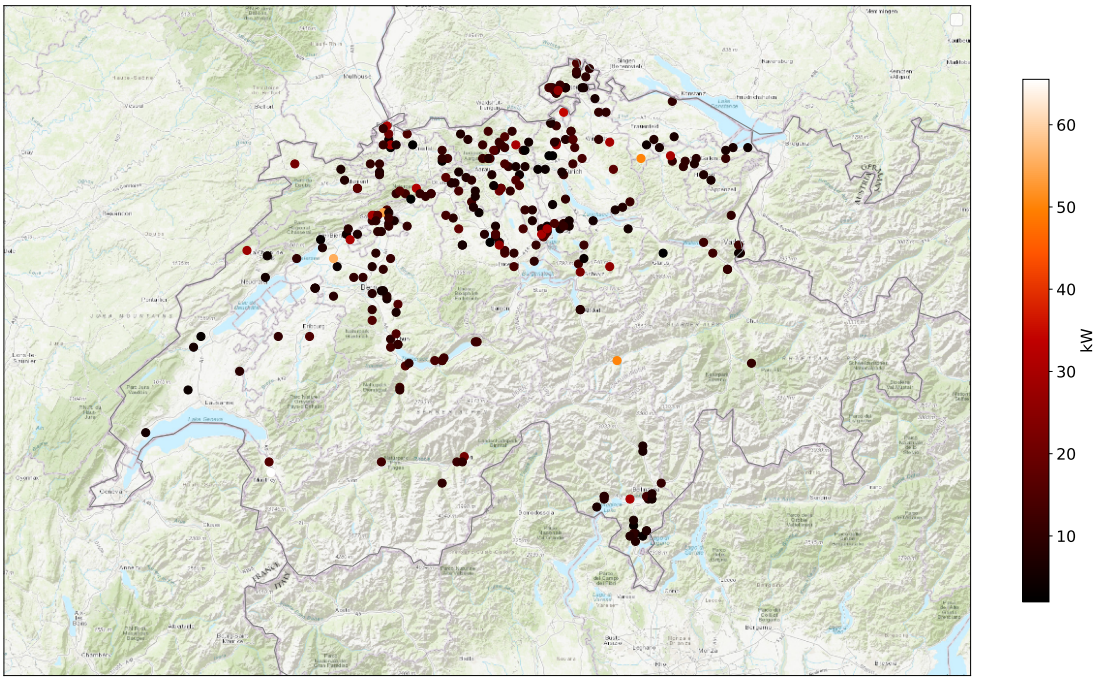
3.4.1 Datasets

Two datasets were used in our study. The first dataset, dubbed the real dataset, consists of records from 304 PV plants across Switzerland. The PV plants are spread inhomogeneously over the entire country, with a density reflecting the population density. The second dataset, dubbed the synthetic dataset, has 1000 PV plants and has been generated with statistical models that match the statistics of the real dataset in terms of location density, size, orientation and pitch angles. Production time series were simulated using the PVlib python library (Stein et al., 2016) and historical weather data from the HelioClim 3 database[2], with high temporal and spatial resolution, as inputs (see (Carrillo et al., 2020) for further details). The spatial distribution of the real and synthetic dataset are shown in Figure 3.4. Both datasets have a 15-minutes resolution for the years 2016-2018.

[2] http://www.soda-pro.com/help/helioclim/helioclim-3-overview



(a)



(b)

Figure 3.4: Spatial distributions of datasets. colours indicate the peak production at each site. a) Synthetic dataset. b) Real dataset.

The weather data used for the single-site benchmarks methods were obtained from two different meteorological providers. The forecast for Bern was computed using historical NWP from the global forecast system (GFS)[3] that has a temporal resolution of 3 hours. On the other hand, the forecast for Bätterkinden was computed using historical NWP from Meteotest[4] with a temporal resolution of 1 hour.

3.4.2 Baselines

Two state-of-the-art methods were used as benchmarks in the multi-site forecasting evaluation. The first one is the recently proposed graph-based Spatio-temporal autoregressive model (STAR) (Carrillo et al., 2020). This method uses an AR model and a group Lasso estimator to select relevant plants (nodes) for the prediction of each individual site (node). The second baseline for multi-site forecasting is the non-graph-based Space-time convolutional neural network (STCNN) (Jeong and Kim, 2019). It uses a greedy-adjoining algorithm that rearranges the plants based on their geographical proximity, one by one, before applying 2D convolution layers as in image processing to produce spatio-temporal features.

Apart from the benchmark methods used for the multi-site evaluation, in the single-site evaluation we used two state-of-the-art methods for single-site PV forecasting that use NWP. The first baseline for single-site comparison is a Support Vector Regression (SVR) model with NWP (global irradiance and temperature) as inputs. It was chosen as benchmark, since it was shown in (Boegli et al., 2018) that SVR outperforms several state-of-the-art methods for intra-day forecasts. The second single-site baseline is a state-of-the-art deep learning model, an Encoder-Decoder long-short term memory neural network (EDLSTM) (Mukhoty et al., 2019). It has a similar architecture to GCLSTM, such that both the encoder and decoder are based on LSTM networks. The decoder uses past observations of weather and PV site data to estimate the state of the system, and the decoder uses the state from the decoder as input as well as NWP (global irradiance and temperature) to forecast the site power. In addition to NWP data, EDLSTM uses the clear sky global irradiance for the site.

3.4.3 Data preprocessing

Power data were normalized for both the real and synthetic dataset in the same manner for GCLSTM, GCTrafo and STCNN: The data for each node are normalized by the maximum power production over the training year. The STAR, albeit requires careful normalization in order to extract daily mode profiles from the past measurements. The tailor made normalization scheme is of utmost importance in the case of linear methods and for STAR is described in (Carrillo et al., 2020).

The considered NWP for the SVR and EDLSTM models contains gaps and have a coarser

[3] https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/global-forcast-system-gfs
[4] https://meteotest.ch/en/

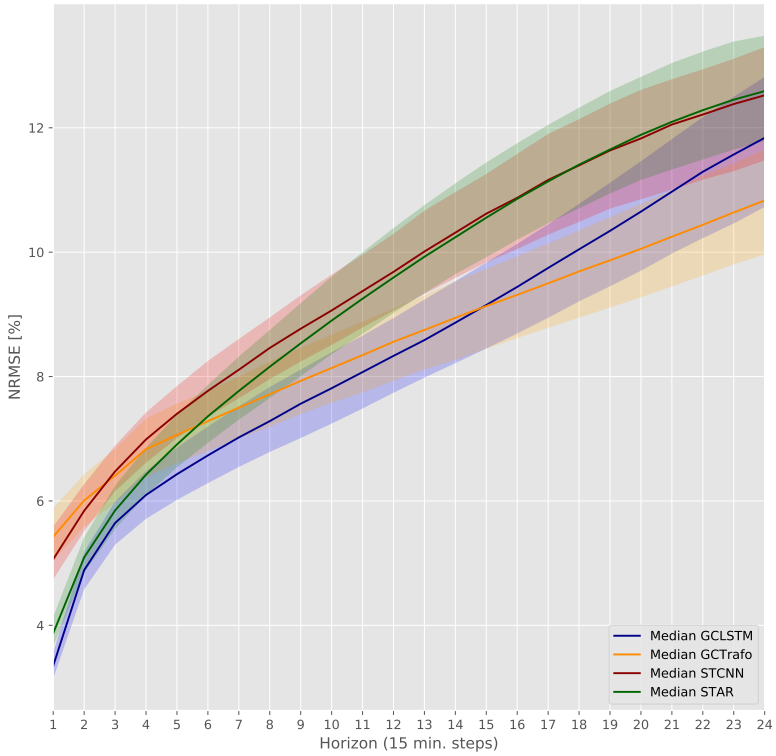


Figure 3.5: Forecast NRMSE comparison for synthetic multi-site PV power prediction. The forecast horizon is six hours in steps of 15 minutes. Solid lines show the median error while the shaded areas show the inter-quantile distance of the errors.

resolution than the power data. In order to obtain 15-minutes resolution data without gaps, polynomial interpolation was used for GFS data (Bern) and a sample-and-hold interpolation was applied to Meteotest data (Bätterkinden). All weather data were normalized before training using min-max scaling.

3.4.4 Training

All methods, except STAR, were trained on the first year of available data (2016) and evaluated on the second year (2017). STAR model coefficients were fitted over two months and then used to predict the power production over the next two weeks. The models were re-fitted every two weeks in a rolling window fashion for the entire 2017 year. The hyperparameters used in the developed and baseline models are described in more detail in the Appendix. All models were trained on a workstation with 16 cores, 128 GB of RAM memory and a Nvidia RTX 2080 Ti GPU.

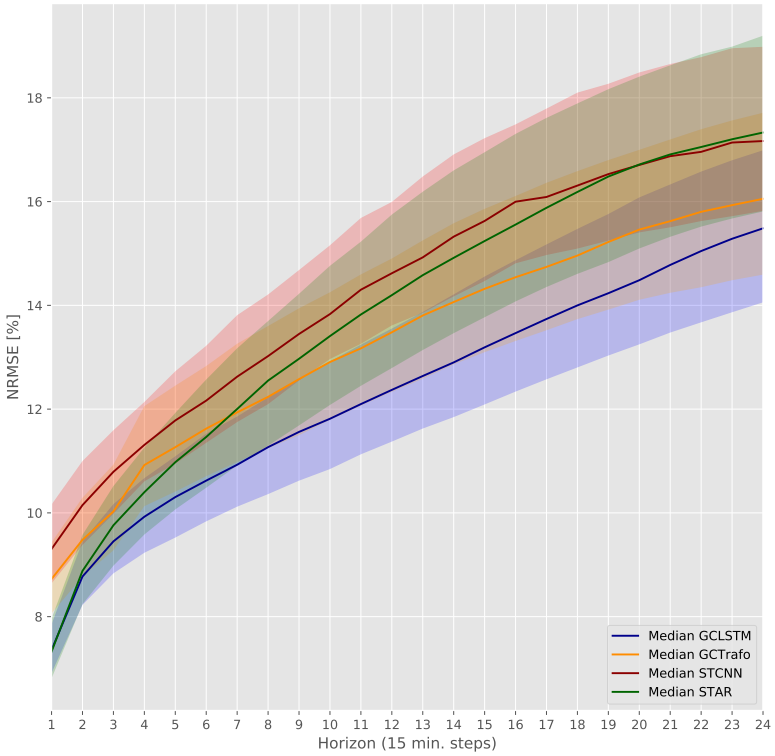


Figure 3.6: Forecast NRMSE comparison for real multi-site PV power prediction. The forecast horizon is six hours in steps of 15 minutes. Solid lines show the median error while the shaded areas show the inter-quantile distance of the errors.

3.4.5 Evaluation and metrics

The proposed models were compared over the year 2017 on the two datasets. Both the peak normalized root mean-squared error (NRMSE) and the averaged normalized mean absolute error (NMAE) are used as metrics. They are defined at site v and forecasting step (horizon) i as:

$$NRMSE(v,i) = \sqrt{\frac{1}{T}\sum_{t\in S}\left(\frac{\hat{p}_v(t+i) - p_v(t+i)}{p_v^{max}}\right)^2},$$

$$NMAE(v,i) = \frac{\sum_{t\in S}|\hat{p}_v(t+i) - p_v(t+i)|}{\sum_{t\in S} p_v(t+i)},$$

where $p_v(t)$ and $\hat{p}_v(t)$ denote the ground truth power and predicted power, respectively, of site v at time t, p_v^{max} is the maximum power of site v over the evaluation period S, i.e., the 2017 year, and T is the number of time steps in the evaluation interval S. Night times are excluded from error computations.

Table 3.1: Forecasting performance of proposed and baseline models on the synthetic dataset

	Synthetic dataset							
	15min		1h		3h		6h	
	NRMSE	NMAE	NRMSE	NMAE	NRMSE	NMAE	NRMSE	NMAE
STAR	3.870	8.68	6.42	14.87	9.59	22.98	12.59	**28.52**
GCLSTM	**3.350**	**7.23**	**6.09**	**13.32**	**8.33**	**20.49**	11.84	29.14
GCTrafo	5.420	18.75	6.83	21.42	8.55	24.65	**10.83**	29.89
STCNN	5.060	13.63	6.99	17.89	9.68	25.06	12.52	31.97

Table 3.2: Forecasting performance of proposed and baseline models on the real dataset

	Real dataset							
	15min		1h		3h		6h	
	NRMSE	NMAE	NRMSE	NMAE	NRMSE	NMAE	NRMSE	NMAE
STAR	**7.32**	15.80	10.40	23.83	14.20	33.15	17.33	40.41
GCLSTM	7.36	**15.71**	**9.93**	**22.48**	**12.40**	**29.36**	**15.53**	**39.44**
GCTrafo	8.76	20.27	10.95	25.93	13.54	33.13	16.07	40.50
STCNN	9.30	21.91	11.31	26.98	14.62	37.22	17.17	43.85

3.4.6 Results

We start by evaluating the performance of GCLSTM and GCTrafo on multi-site PV forecasting. Figures 3.5 and 3.6 show the evolution of the prediction errors of aforementioned methods over a horizon of 6 hours ahead, in steps of 15 minutes, for the synthetic and real dataset, respectively. The shaded regions represent the inter-quartile (25%-75%) error range over all nodes and solid lines represent the median NRMSE over all nodes. Results show that in the real dataset GCLSTM outperforms all other methods for the entire prediction horizon. However, in the synthetic dataset GCLSTM yields the lowest error up to 4 hours ahead, whereas GCTrafo outperforms the other models for predictions from 4 to 6 hours ahead. This shows the effectiveness of GNNs to capture the spatio-temporal correlations of the PV production data. The synthetic dataset has a lower forecasting error due to the spatial and time smoothing in the generation of the HelioClim 3 irradiance database used to synthesize the PV power profiles.

Although the linear STAR method performs better than the GCTrafo within the first hour on both datasets, GCTrafo shows a lower error slope for horizons from one hour ahead on than the other methods, making it a promising model for longer prediction horizons, e.g. from 4 hours to day ahead. The main reason is that GCTrafo has the attention weights that can focus on different spatial or temporal information. Attention weights are more powerful than the recurrent structures, which suffer from fading memory for longer sequences.

The comparative NRMSE and NMAE on both datasets are shown in Table 3.2 and Table 3.1 for 15 minutes, 1 hour, 3 hours and 6 hours ahead predictions. NRMSE is more sensitive to outliers, because it considers squared errors, therefore, gives more weight to large errors,

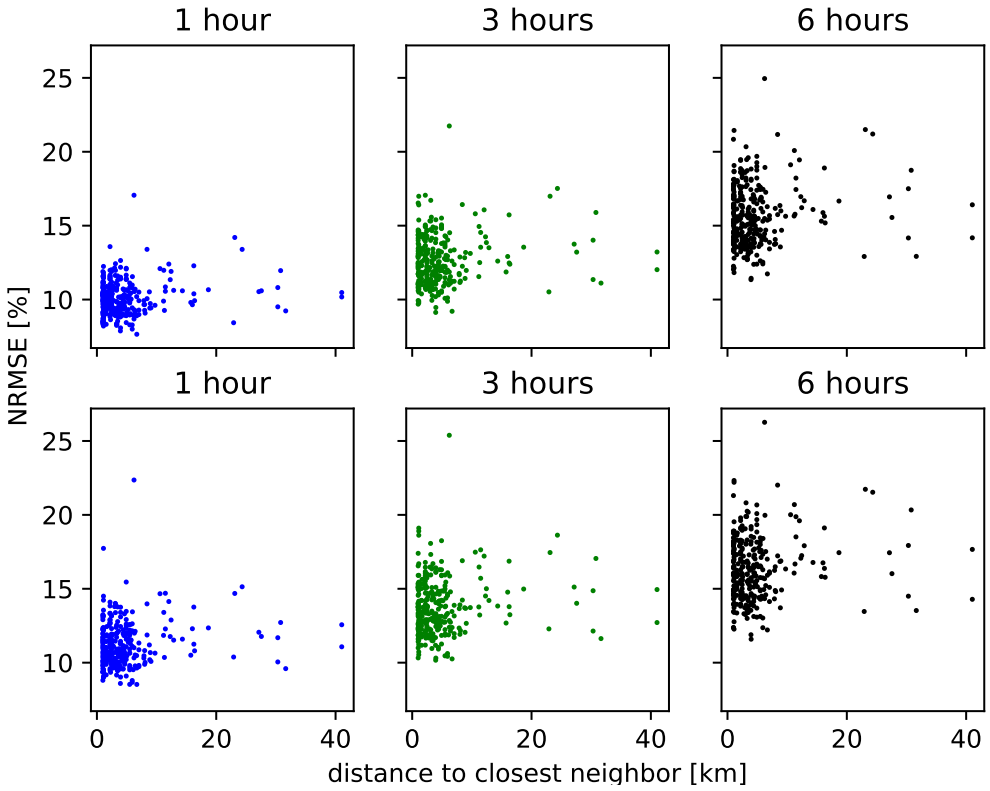


Figure 3.7: NRMSE with respect to the distance to the closest neighbour for 1, 3 and 6 hours predictions for GCLSTM (top) and GCTrafo (bottom).

thus using both metrics is useful when comparing methods. For example, this difference is highlighted in the errors for 6 hours ahead in the synthetic dataset. Although GCLSTM has a lower NMAE, the NRMSE is lower for GCTrafo. Also, the NMAE gives a figure of the forecast error in percentage of the total yearly production.

We analysed the distance between nodes for which the model performances start to degrade on the real data set. To this end, the distance to the closest neighbour for each node is calculated and isolated nodes are found. Figure 3.7 shows the NMRSE for 1, 3 and 6 hours ahead predictions versus the distance to the closest neighbour for all nodes. The analysis does not indicate a higher error for sites that are further away and more isolated. For instance, the NRMSE for nodes with close neighbours (less than 5km) is between 10% and 19% for 3 hours ahead predictions. On the other hand, the NRMSE for the same horizon of isolated sites, i.e. 30 to 43 km away from the closest node, is between 11% and 17%. The same behaviour is shown for all other prediction horizons. Therefore, up to 40km, the models don't show a drop in performance.

Next, we show the forecasting results for two sites in the central part of Switzerland: Bern and Bätterkinden. The two locations are about 25 km apart. Figure 3.8 and Figure 3.9 show the

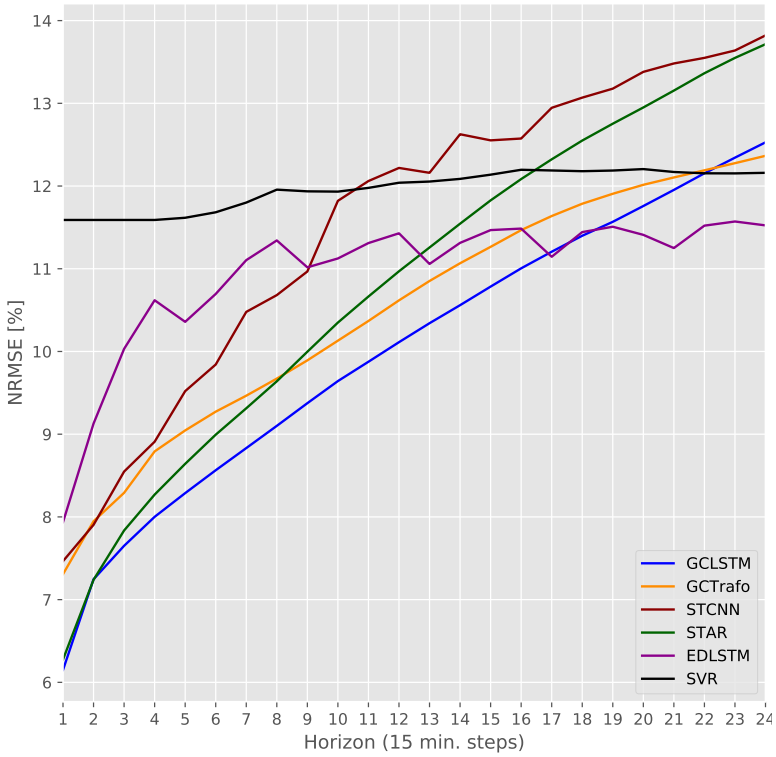


Figure 3.8: Single-site forecast for Bätterkinden. NRMSE comparison between the proposed models (GCLSTM and GCTrafo), alternative multi-site methods with similar inputs (STAR and STCNN), and models that use NWP as inputs (SVR and EDLSTM).

NRMSE evolution for GCLSTM, GCTrafo, STAR and STCNN, and the EDLSTM and SVR methods that use NWP as inputs. The error trend in the multi-site comparison, between GCLSTM, GCTrafo, STAR and STCNN is similar to the one observed in the single-site comparison. For both sites, GCLTSM outperforms other methods between 1 and 4 hours ahead predictions. Interestingly, during the first hour (4 steps ahead) GCLSTM is on a par with the linear STAR method. However, for longer term forecasts, 5 to 6 hours ahead, EDLSTM and SVR methods yield lower errors than the proposed methods. The main reason lies in the fact that they use as the additional input NWP data, which has higher accuracy for six hours to day ahead predictions. However, NWP-based forecasts have higher error rate in comparison to other methods for intra-day forecasts. Additionally, we observe the high impact of the temporal resolution of the weather data on the results, since a higher resolution of the weather data leads to higher accuracy of the forecast, which is the case of Bätterkinden (Figure 3.8).

As an illustration of some of the advantages and limitations of the proposed methods, Figure 3.10 and Figure 3.11 show a visualization of the time series for one hour and six hours ahead forecasts for two days in Bern. The first day has a clear sky with a few clouds passing by during the middle of the day whereas the second day is a cloudy day with low production during whole day. The daily NRMSE for the two days and the two horizons are shown in Table 3.3.

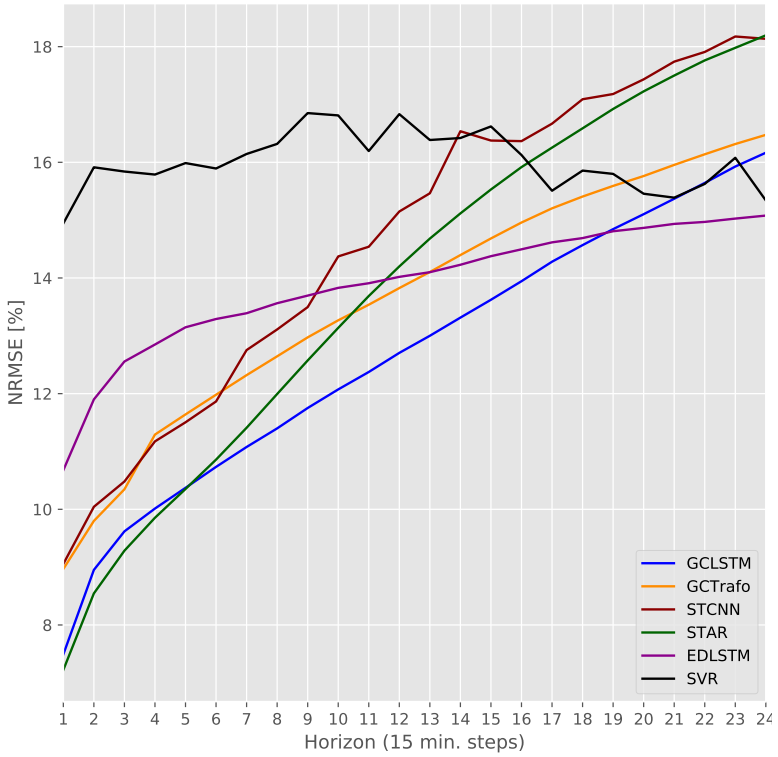


Figure 3.9: Single-site forecast for Bern. NRMSE comparison between the proposed models (GCLSTM and GCTrafo), alternative multi-site methods with similar inputs (STAR and STCNN), and models that use NWP as inputs (SVR and EDLSTM).

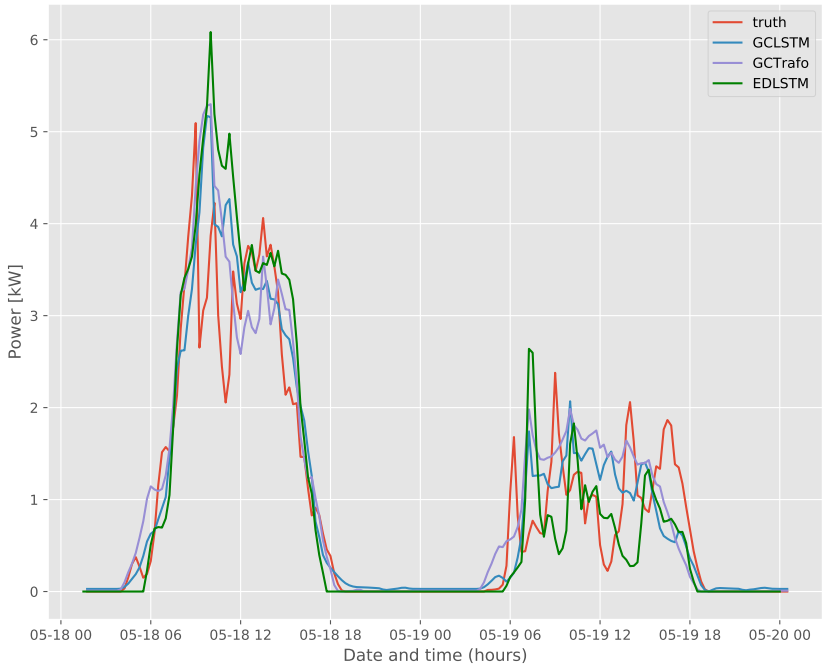


Figure 3.10: Illustration of measured production and 1 hour ahead forecasted power production for two days in Bern. Only forecasts from GCLSTM, GCTrafo and EDLSTM are included.

This visual comparison is made to show an extreme case where the proposed models might fail to provide an accurate forecast for long-term horizons (6h ahead), especially at the beginning of the day, and where models that use NWP as inputs (EDLSTM) have an advantage.

Figure 3.10 shows time series of the true PV production and 1 hour ahead forecasts using GCLSTM, GCTrafo and EDLSTM. From the daily errors and the visual assessment, we can conclude that during cloudy days, for short-term forecasts, graph-based methods outperform EDLSTM because of their ability to capture cloud movement and spatial information. On the other hand EDLSTM relays on NWP that have low spatial and temporal resolution yielding poor forecasts, even though it uses past site data to initialize the encoder.

Table 3.3: Daily NRMSE for Bern illustration

Model	Day1 1h	Day2 1h	Day1 6h	Day2 6h
GCLSTM	8.79	7.96	8.94	20.87
GCTrafo	9.31	8.22	15.46	26.74
EDLSTM	11.85	8.91	13.98	7.71

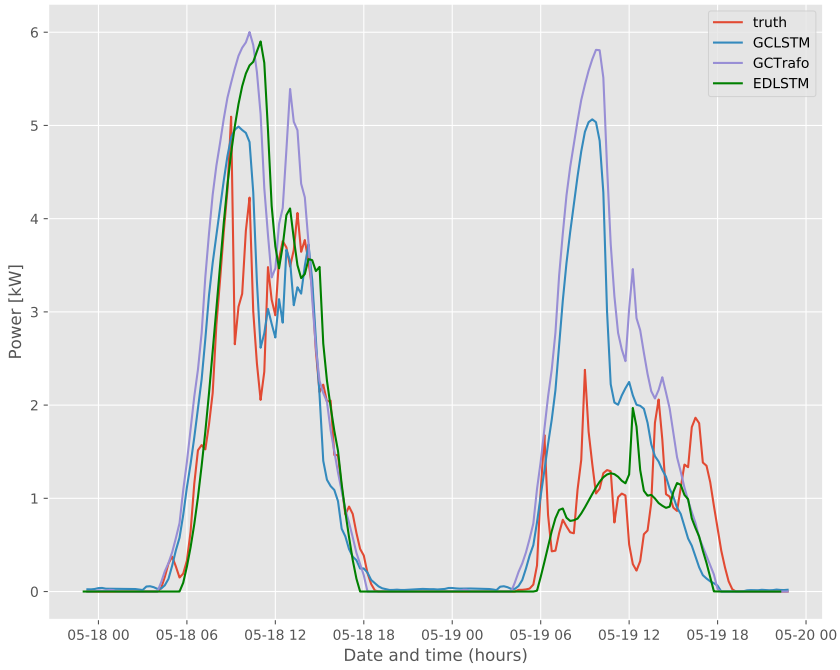


Figure 3.11: Illustration of measured production and 6 hour ahead forecasted power production for two days in Bern. Only forecasts from GCLSTM, GCTrafo and EDLSTM are included.

However, for 6 hours ahead, GCLSTM and GCTrafo forecasts in Figure 3.11 show a bias during the first six hours of the day after sunrise. Since night PV production values are zero, the graph-based architectures only receive clear sky data and average power from the two previous days. Thus, GCTrafo and GCLSTM forecast higher production values during the first 6 hours after the sunrise. Once the graph-based architectures start to receive non-zero PV power information from the day, these methods start to correct the predictions and we observe a sudden drop in the predicted values. During very cloudy days, such as the second day, EDLSTM benefits from NWP for 6 hours ahead forecasts.

3.5 Conclusions

Two novel graph convolutional neural network architectures for multi-site deterministic PV generation forecasting, GCLSTM and GCTrafo, have been introduced and compared with state-of-the-art algorithms, both at single and multi-site levels. The extensive comparison on two PV power generation datasets (the real dataset with 304 plants and the synthetic dataset with 1000 PV plants) has shown that they outperform state-of-the-art methods, with an average NRMSE error over the entire horizon (6 hours ahead) of 8.3% (GCLSTM) and 8.4% (GCTrafo) node setting, and 12.6% (GCLSTM) and 13.6% (GCTrafo) on the real dataset. Both architectures were

trained on a single GPU for the 1000 nodes case and can be scaled to a higher number of nodes using multi-GPU computing, making them appealing for grid management applications with large number of nodes.

In forthcoming works, we will address some inherent limitations in the way spatio-temporal information is diffused across the nodes in these models. The number of nodes taken into account within the graph convolutions were limited to the K closest neighbours because of the increase in computational complexity. However, it is expected that further away nodes might be important predictors if advection is dominant in the regional cloud dynamics at a specific time. Another research direction is to investigate the robustness and adaptability of the models to different weather conditions. Therefore, the performance of the presented models should be investigated during sunny, cloudy and variable days. Finally, another possible avenue of research is to transform the proposed deterministic models into probabilistic models by integrating a quantile regression as in works of Carrillo et al. (2023) or by integrating noise into the deterministic model to build a generator in a similar fashion to (Koochali et al., 2021). This generator should be trained using an appropriate classifier as discriminator in an adversarial setting to make a probabilistic forecast.

4 Interpretable temporal-spatial graph attention network for multi-site PV power forecasting

4.1 Introduction

In the previous Chapter, we have shown that using a graph signal processing perspective, we can exploit spatio-temporal correlations between sites by representing PV systems as nodes of a graph and embedding production data as signals on that graph. PV power forecasting task for intra-day predictions using solely ground-based PV power production data has been addressed from a machine learning perspective. Different linear and non-linear models have been employed. However, they are not capable of capturing spatio-temporal dynamics. Then, the PV power forecasting task has been addressed from a Graph Signal Processing (GSP) perspective successfully (Khodayar et al., 2019; Simeunović et al., 2022a). Although these advanced architectures capture spatio-temporal correlations, they are restricted to predefined k-neighbours graphs, neglecting the impact of the nodes which are further away, which is vital for longer-term (from 2 to 6 hours ahead) forecasts. Furthermore, we model physical phenomena when modelling PV power production; thus, an interpretable model is desirable. Currently, the machine learning models that achieve state-of-the-art accuracy and high temporal and spatial resolution for PV forecasting lack the interpretability.The question poses whether it is possible to model a graph-based interpretable model without the limitation of restricting the neighbourhood for the longer-term part of the intra-day forecast.[1]

Recurrent and graph convolutional neural networks, used to find spatio-temporal correlations between PV power data, are difficult to interpret. On the other hand, state-of-the-art attention-based architectures for time series forecasting use multi-head attention, which is not fully interpretable. The works of Michel et al. (2019) and Cordonnier et al. (2020) have shown that multi-head attention tends to learn redundant relationships. Moreover, the work of Baan et al. (2019) demonstrated that the multi-head approach is only partially interpretable and concluded that in order to be transparent, it should have no overlap in specialization but focus on distinct representational subspaces. Therefore, an attention-based model which focuses on distinct representational subspaces is required in order to be able to understand what

[1] The content of this Chapter is based on the publication Simeunović et al. (2022b).

influences the forecast.

In this work, we propose a temporal-spatial multi-window graph attention network (TSM-GAT) applied to the PV power forecasting problem to solve the aforementioned issues of low interpretability and restricting graph topology to a small number of the closest neighbours, thus excluding the long-term correlations between sites. A dense network of PV stations could be used as a network of virtual weather stations, where spatio-temporal correlations are modelled using dynamic graphs. Thus, PV stations are modelled as nodes in a dynamic graph, and embedded past PV data, coupled with geographical information and clear-sky data, represents the graph signals. TSM-GAT can adapt to the dynamics of the problem by learning different graphs over time. It consists of temporal attention with an overlapping-window mechanism and spatial attention with multi-window mechanism. The output of the temporal attention is temporal correlations and embedded temporal features which are passed to the spatial attention, which finds different spatio-temporal correlations. Furthermore, in the spatial attention, a multi-window mechanism is introduced in order to increase interpretability and accuracy at longer-term horizons. Moreover, this model yields the shape of the predicted signal closer to the shape of the ground truth, which indicates that the model is better at capturing the movement of clouds and cloud coverage. Since PV production is dependent on weather conditions and information from PV stations is limited to certain spatial location, PV stations across different countries have different weather conditions. The interpretability of the model indicates a close link between the model and physical phenomena, making it promising in terms of generalization and performance in different weather conditions. The contributions of the current work include:

- A TSM-GAT model that extends of graph attention networks for time series forecasting tasks. Temporal attention embeds temporal features from times-series data into temporal windows, creating inputs for the dynamic graph. The following spatial attention finds spatio-temporal correlations for each node and each temporal window, yielding a dynamical adjacency.
- A multi-window mechanism is developed in TSM-GAT to allow the model to learn different dynamical adjacency matrices for shorter-, medium- and longer-term predictions. Thus, it is possible to interpret which nodes (PV stations) the model focuses on, when predicting short-, medium- and long-term intra-day forecasts.
- A performance evaluation of TSM-GAT on real and synthetic PV datasets and comparison with state-of-the-art multi-site and single-site models. For multi-site models, an analysis of the impact of weather conditions, and node density on the forecast is presented on the real dataset in Switzerland, in order to understand what drives the performance. A study has further been conducted to compare widely used multi-head and proposed multi-window approaches, which showed higher accuracy and interpretability of the multi-window approach.

The rest of the Chapter is organized as follows. Section 5.2 introduces preliminaries on

graph attention and graph time series forecasting of PV generation. Section 5.3 details the proposed TSM-GAT architecture. The experimental results of our evaluation and the analyses are presented and discussed in Section 5.4 and Section 5.5. Finally, we conclude in Section 4.6.

4.1.1 Related work

Multi-site time series forecasting task has been studied not only in the PV domain but also in wind speed, wind power production, electricity and irradiance forecasting. In the work of Ghaderi et al. (2017) recurrent long-short term memory (LSTM) cells are used to extract temporal correlations. In addition to recurrent neural networks, convolutional neural networks were used in the work of Lai et al. (2018). However, they were outperformed by the attention mechanisms, which capture correlations due to their ability to select relevant time steps and locations; see the work of Shih et al. (2019). Graph convolutional structures coupled with recurrent structures proposed in wind forecasting task in work of Khodayar and Wang (2019), requires a different model for each step ahead prediction, which is not scalable. The lower scalability and higher error are some of the shortcomings which have been addressed by introducing the graph multi-head attention. However, graph multi-head attention is not fully interpretable, as previously discussed. All these spatio-temporal networks restrict the graph topology to a small number of k neighbours.

On the other hand, a lot of work has been focused on single-site forecasting task for PV and wind power production. The models proposed in the works of Azad et al. (2014) and Hossain et al. (2018) focus on the long-term (up to year ahead) forecasts of wind speed and power density. However, these approaches require learning different models for each station which increases the computational costs. Recent single-site PV power production models in the works of Zhou et al. (2019); Kharlova et al. (2020); Ren et al. (2022) and Dairi et al. (2021) use LSTM to find temporal dynamics and attention mechanism to extract relevant features. They show that LSTM coupled with the attention mechanism improves accuracy, especially in the case of long sequence inputs. However, these methods also exploit NWP and weather data, which have low spatial and temporal resolution for multi-site forecasting models with high number of PV plants. Moreover, by taking into account only single-site time series, it is not possible for these models to exploit spatial correlations between the sites. The addition of LSTM cell to the attention block makes it more difficult to interpret on which temporal steps the model is focusing when making the prediction.

4.2 Multi-site PV power time series forecasting on graphs

The PV power forecasting task is a time series prediction task, predicting future PV power production, given past PV data. PV data is highly correlated in time since slowly varying local weather conditions and cloud cover affect PV production across time steps. Further, cloud motion affects PV production of neighbouring production sites sequentially, thus, making the data correlated in both spatial and temporal domains. Thus, part of the cloud dynamics

influence can be inferred by modelling the past PV data as signals on a dynamic graph. This permits to capture of the dynamically changing spatio-temporal correlations in the PV data. Specifically, we denote the vector of PV power production over N PV stations at time step τ as $\mathbf{p}(\tau) \in \mathbb{R}^N$. The goal is to forecast $\mathbf{p}(\tau)$ for the next H discrete time steps ahead given M past observations:

$$\hat{\mathbf{p}}(\tau), \ldots, \hat{\mathbf{p}}(\tau + H - 1) = f_\beta \left(\mathbf{p}(\tau - M), \ldots, \mathbf{p}(\tau - 1) \right), \tag{4.1}$$

where for any τ, f_β is a chosen family of parametric estimators. A set of parameters β is learnt such that it minimizes the prediction error over the entire horizon by solving the following problem:

$$\arg\min_{\beta} \sum_{\tau \in \mathscr{T}} \sum_{\xi=\tau}^{\tau+H-1} \| \hat{\mathbf{p}}(\xi) - \mathbf{p}(\xi) \|_2^2, \tag{4.2}$$

where $\mathscr{T}$ represents the times of historical time steps which are used for fitting the model during training. Although we have chosen the mean square error as the metric, other metrics could be used as well.

In this work, we model the PV production time series as signals on a dynamic graph. The network of PV power stations is represented as a weighted undirected graph G^t with PV power production, where the graph is a tuple $G^t = (v, \varepsilon^t, \boldsymbol{\gamma}^t)$ and $t \in \{1, \ldots, T\}$ represents the number of graphs in a dynamic graph. The graphs G^t and G^{t+1} are m time steps τ apart from each other. A set of vertices (nodes) $v = \{v_1, v_2, \ldots, v_N\}$ in the graph corresponds to a set of PV stations and ε is a set of edges (links), corresponding to the correlations between the stations within the graph G^t. The most commonly used correlations are Pearson or distance correlation. The topology of the graph is described with its adjacency matrix $\boldsymbol{\gamma}^t \in R^{N \times N}$ that dynamically captures the connectivity of the nodes. Each adjacency entry represents an edge between v_i and v_j in the graph G^t, denoted with $e_{ij}^t \in \varepsilon^t$. If there is no edge between v_i and v_j in the certain graph G^t, then the matrix entry γ_{ij}^t is equal to zero. The dynamically changing correlations between PV stations are modelled with a dynamic adjacency matrix. Finally, we define a graph signal as a mapping $\mathbf{x} : v \to \mathbb{R}$, such that $x_v^t \in \mathbb{R}$ is the signal value at node v and graph G^t. For more details and an in-depth review of GSP we refer the reader to Ortega et al. (2018).

In our case, the graph signal at time step τ is denoted by $\mathbf{x}(\tau)$ and it represents the vector of the power production of all PV stations $\mathbf{p}(\tau)$ concatenated with additional spatial information and geographical location. Furthermore, in order to provide the model with additional information regarding the sunrise, sunset and seasonality information, the clear sky irradiance values are additionally included $\mathbf{y} = \left(\mathbf{y}(\tau), \ldots, \mathbf{y}(\tau + H - 1) \right) \in \mathbb{R}^{N \times H}$ for predicting H steps ahead, computed at a particular location on the map at any time, using the Ineichen and Perez clear sky model from PVlib (Stein et al., 2016). This computation is completely deterministic and only relies on the geographical coordinates of the nodes (latitude, longitude and altitude).

Thus we can reformulate our learning problem

$$\hat{\mathbf{p}}(\tau),\ldots,\hat{\mathbf{p}}(\tau+H-1) = f_\beta\left(\mathbf{p}(\tau-M),\ldots,\mathbf{p}(\tau-1),\right.\\ \left.\mathbf{y}(\tau),\ldots,\mathbf{y}(\tau+H-1),\mathbf{long},\mathbf{lat},\mathbf{U}\right), \tag{4.3}$$

where $\mathbf{U}$ is a mask used to initialize the neighbourhood of each node i. The initial neighbourhood is calculated using k-nearest neighbour mechanism. The set of parameters β includes not only parameters for learning correlations in the dynamic graph, but also parameters for embedding the spatio-temporal features and finally parameters for predicting the future PV production.

4.3 Temporal-spatial multi-windows graph attention network

4.3.1 The overall architecture

In this Chapter, we propose a sequence to sequence model TSM-GAT. The model represents a solution for multi-site PV power forecasting from GSP perspective, shown in Figure 4.1. A sequence of the past M PV measurements $\mathbf{p} = \left(\mathbf{p}(\tau-M),\ldots,\mathbf{p}(\tau-1)\right) \in \mathbb{R}^{N\times M}$ over N nodes is taken as input to the model, when predicting H steps ahead $\hat{\mathbf{p}} = \hat{\mathbf{p}}(\tau),\ldots,\hat{\mathbf{p}}(\tau+H-1)$. In order to capture both short- and longer-term dependencies, the attention mechanism from Veličković et al. (2018) is used since it has access to any part of the sequence.

TSM-GAT consists of temporal attention with the overlapping-window mechanism and spatial attention with the multi-window mechanism. The temporal attention operator is capturing temporal non-linear correlations using a modified attention mechanism. However, finding a correlation between all M past time steps leads to suboptimal solutions since it assigns a single attention value for all past time steps. Furthermore, weather conditions fluctuate and the wind might suddenly change directions, creating a challenge to model cloud dynamics. Therefore, the past M PV measurements are divided into T temporal windows. The past PV data in each temporal window is concatenated with vectors of geographical coordinates $\mathbf{long},\mathbf{lat} \in \mathbb{R}^N$. The temporal attention is capturing temporal non-linear correlations for each temporal window. The embedded temporal features per each window represent entries of different graphs in the dynamic graph.

Thus, embedded features are passed to the spatial attention operator, which captures dynamically changing spatial correlations, yielding the dynamical adjacency. Finally, viewing the output of the temporal-spatial attention block as a vector encoding shading information, this vector is concatenated with an embedding of the clear-sky irradiance $\mathbf{y} = \left(\mathbf{y}(\tau),\ldots,\mathbf{y}(\tau+H-1)\right) \in \mathbb{R}^{N\times H}$ before the last multi-perceptron layer (MLP); see Figure 4.1. The exact concatenation of PV power and clear sky irradiance signals is presented at the end of the section.

In order to calculate spatial attention, it is crucial to define the neighbourhood of each node since it is not predefined. To this end, the adjacencies are initialized using the k-nearest neigh-

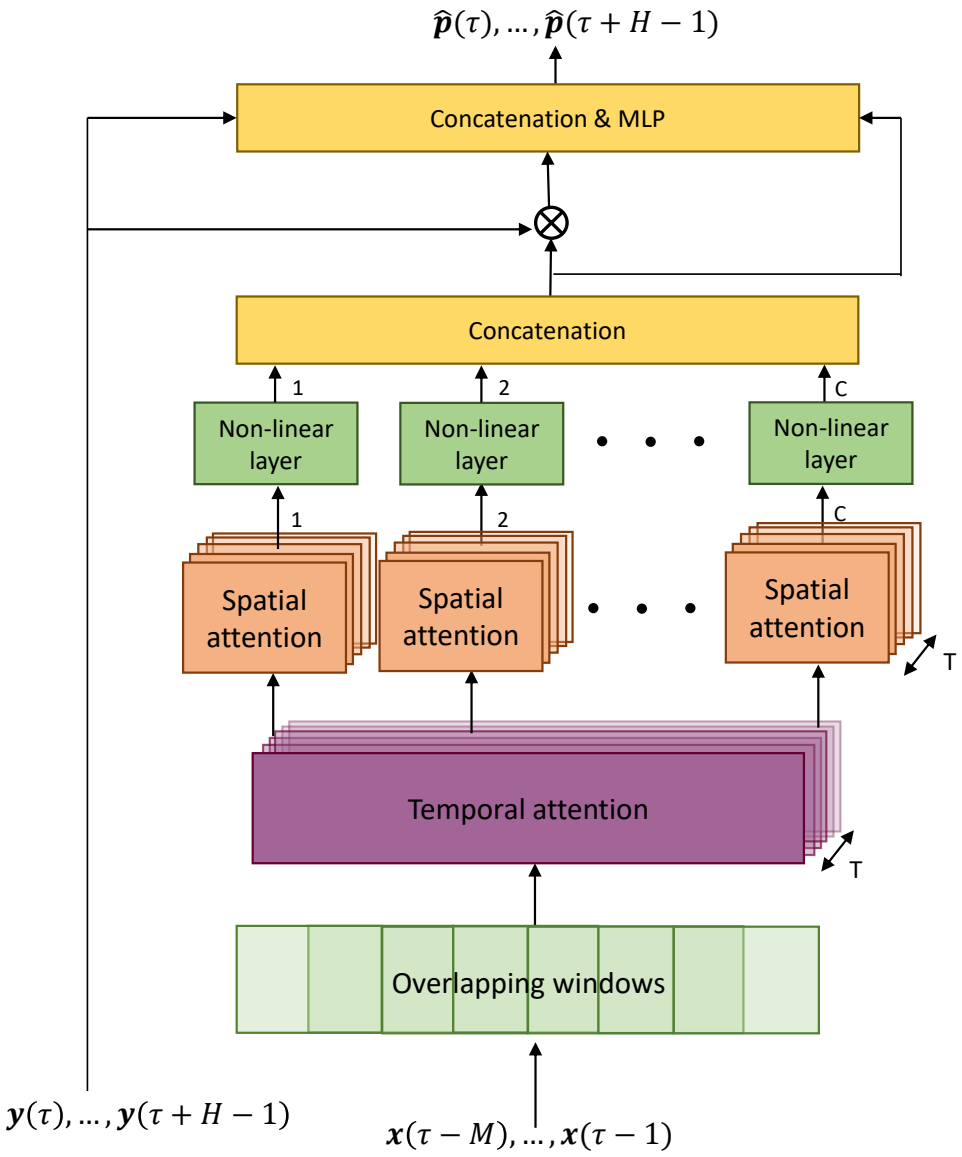


Figure 4.1: TSM-GAT model.

bour algorithm. Intuitively, for the short-term forecasts (up to 2 hours) the neighbourhood taken into account should not be too large since the weather changes and cloud movements are affecting the nodes in the smaller localized area. As the forecasting window increases, nodes further away should be taken into account in order to capture cloud movement, since it takes more time for clouds to move between nodes that are spatially further away. Therefore, the multi-window mechanism is introduced, such that the different neighbourhood is initialized for each window, where C is the total number of windows in the multi-window mechanism.

4.3.2 Graph Attention

In this work, we use Graph Attention Networks (GAT) from Veličković et al. (2018) to infer the correlation between nodes. Let a signal value at node $i \in N$ be represented with a column vector $\mathbf{x_i} = [x_i^1, \ldots, x_i^f] \in \mathbb{R}^f$, where f is the number of features per node. The attention mechanism is used to contextually weight the importance of node j features to the node i. A shared matrix $\mathbf{W} \in \mathbb{R}^{f' \times f}$ is used to embed input features f into a f'-level feature space. Then the normalized attention coefficients α_{ij} are computed from (4.4):

$$\alpha_{ij} = softmax_j\left(l\left(\mathbf{a} \cdot \left[\mathbf{W}\mathbf{x_i} || \mathbf{W}\mathbf{x_j}\right]\right)\right) \tag{4.4}$$

where $\mathbf{a} \in \mathbb{R}^{2f' \times 1}$ is a row vector parameterizing the attention mechanism and $\cdot$ denotes dot product multiplication between vectors. A concatenation is represented with $||$ and $l(\cdot)$ denotes the activation function LeakyReLU. Finally, the obtained normalized attention coefficients are used to compute the final output $\mathbf{h_i}$ for every node (4.5):

$$\mathbf{h_i} = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W}\mathbf{x_j}\right) \tag{4.5}$$

where $\mathcal{N}_i$ represents the neighbourhood of node i and $\sigma(\cdot)$ is the LeakyReLu nonlinearity. For more details see Chapter 2.

4.3.3 Temporal attention

The temporal attention is capturing temporal non-linear correlations for each temporal window, by embedding the temporal features using modified attention. In the temporal attention the past data is divided in the T temporal windows. Then attention mechanism is applied in order to find the correlations between the temporal windows. Following a classical signal processing perspective (Bahoura, 2019), we have divided input feature space into the temporal windows with 50% overlap. The first half of each window is shared with the last half of the previous window, and the last half is shared with the first half of the subsequent one. Therefore, we divide the input M lags among T overlapping temporal windows of size $2m$. The window size is calculated based on the overlap m, where $m = \frac{M}{T+1} \in \mathbb{N}$. Since in each window longitude and latitude are concatenated as additional spatial information, the input signal is $\mathbf{x} = \left(\mathbf{p}(\theta), \mathbf{p}(\theta+1), \ldots, \mathbf{p}(\theta+2m) || \mathbf{long}, \mathbf{lat}\right) \in \mathbb{R}^{N \times T \times \mathrm{f_{in}}}$, where $f_{in} = 2m+2$ and $\theta \in \{\tau - M, \tau - M + m, \tau - M + 2m, \tau - M + 3m, \ldots, \tau - 2m\}$.

In the work of Veličković et al. (2018), in the Equation 4.4 the weight matrix $\mathbf{W}$ is shared across nodes. On the other hand, in the temporal attention of TSM-GAT, we created a matrix $\mathbf{W^t_{temp}} \in \mathbb{R}^{f' \times f_{in}}$ with different weight matrices for each time window $t \in \{1, \ldots, T\}$ to increase expressive power. Let $\mathbf{x_t^i} \in \mathbb{R}^{\mathrm{f_{in}} \times 1}$ denote the input signal for node i and temporal window t. We use the softmax function with $LeakyReLU$ activation function to get normalized temporal

attention coefficients across all temporal windows:

$$\alpha_{kt}^{i} = softmax_t(l(\mathbf{a}_{\mathbf{temp}}^{\mathbf{t}} \cdot [\mathbf{W}_{\mathbf{temp}}^{\mathbf{t}}\mathbf{x}_{\mathbf{k}}^{\mathbf{i}} || \mathbf{W}_{\mathbf{temp}}^{\mathbf{t}}\mathbf{x}_{\mathbf{t}}^{\mathbf{i}}])) \tag{4.6}$$

where the coefficient α_{kt}^{i} represents the importance of the temporal window k when making the prediction. A row vector $\mathbf{a}_{\mathbf{temp}}^{\mathbf{t}} \in \mathbb{R}^{2f' \times 1}$ is a weight vector of the attention mechanism calculated for each temporal window $t \in \{1,\dots,T\}$ and shared across nodes. These normalized attention coefficients are used to compute the output of the temporal block $\mathbf{h} \in \mathbb{R}^{N \times T \times f'}$ where we define the column vector of features $\mathbf{h}_{\mathbf{t}}^{\mathbf{i}} \in \mathbb{R}^{f' \times 1}$, which is calculated for each node i and temporal window t:

$$\mathbf{h}_{\mathbf{t}}^{\mathbf{i}} = \sigma\left(\sum_{k=1}^{T} \alpha_{kt}^{i} \mathbf{Q}_{\mathbf{temp}}^{\mathbf{t}} \mathbf{x}_{\mathbf{k}}^{\mathbf{i}}\right) \tag{4.7}$$

where $\mathbf{Q}_{\mathbf{temp}}^{\mathbf{t}}$ is of the same size as $\mathbf{W}_{\mathbf{temp}}^{\mathbf{t}}$. The temporal attention matrix shows insight about which window influences the model when making the prediction for a certain horizon.

4.3.4 Spatial multi-windows attention

The TSM-GAT model captures temporal patterns across different time steps using the overlapping window mechanism with a temporal operator. However, in order to capture the correlation between different nodes at different time steps, a spatial graph attention operator is used on top of the temporal one. The input data to spatial attention are embedded temporal features from different temporal windows. Those features represent entries of different graphs, creating the dynamic graph. Further, we refer to temporal windows as different parts of the input data, which embedded in the temporal attention are passed to the graphs in the dynamic graph.

Let the spatial attention tensor $\boldsymbol{\gamma}$ represent a sequence of dynamical spatial correlation matrices, where $\boldsymbol{\gamma} \in \mathbb{R}^{T \times N \times N}$. A single spatial attention matrix $\boldsymbol{\gamma}^{t} \in \mathbb{R}^{N \times N}$ is calculated for each temporal window $t \in \{1,\dots,T\}$. The correlation between nodes i and j at time window t is

$$\gamma_{ij}^{t} = softmax(l(\mathbf{a}_{\mathbf{spat}}^{\mathbf{i}} \cdot [\mathbf{W}_{\mathbf{spat}}^{\mathbf{t}}\mathbf{h}_{\mathbf{t}}^{\mathbf{i}} || \mathbf{W}_{\mathbf{spat}}^{\mathbf{t}}\mathbf{h}_{\mathbf{t}}^{\mathbf{j}}])), \tag{4.8}$$

such that row vector $\mathbf{a}_{\mathbf{spat}}^{\mathbf{i}} \in \mathbb{R}^{2f'' \times 1}$ have different values for each node $i \in \{1,\dots,N\}$. The weight matrix $\mathbf{W}_{\mathbf{spat}}^{\mathbf{t}} \in \mathbb{R}^{f'' \times f'}$, in the spatial attention operator, is used to embed the input feature space f' to lower dimensional feature space f''in each time window t where $t \in \{1,\dots,T\}$. A spatial attention coefficient γ_{ij}^{t} indicates the importance of node j's features to the node i in the temporal window t. The spatial attention matrix represents the dynamical adjacency matrix since the attention coefficients are dynamically changing in each time window. In addition to spatial weight matrix $\mathbf{W}_{\mathbf{spat}}^{\mathbf{t}}$, the matrix $\mathbf{Q}_{\mathbf{spat}}^{\mathbf{t}} \in \mathbb{R}^{f'' \times f'}$ where $t \in \{1,\dots,T\}$ is parameterizing the output of the temporal attention $\mathbf{h}_{\mathbf{t}}^{\mathbf{i}}$. The output of temporal-spatial attention block is a tensor $\mathbf{g} \in \mathbb{R}^{N \times T \times f''}$. We define a slice of the tensor $\mathbf{g}$ for node i and

temporal window t as the column vector $\mathbf{g_t^i} \in \mathbb{R}^{\mathbf{f''}\times 1}$ obtained from:

$$\mathbf{g_t^i} = \sum_{j\in\mathcal{N}_i} \gamma_{ij}^t \mathbf{Q_{spat}^t} \mathbf{h_t^i}, \tag{4.9}$$

where matrix $\mathbf{Q_{spat}^t}$, is of the same size as the matrix $\mathbf{W_{spat}^t}$. In $\mathbf{g}$, the important information is embedded in f'' distinctive features for each node and for each temporal window. Therefore, flattening the feature vectors $\mathbf{g_t^i}$ across temporal windows $t \in \{1,...,T\}$ yields an augmented matrix $\mathbf{G} \in \mathbb{R}^{N\times Tf''}$, where each row is a vector $[\mathbf{g_1^i}, \mathbf{g_2^i}, \ldots, \mathbf{g_T^i}] \in \mathbb{R}^{Tf''}$ of embedded spatio-temporal features for the node i. This information is passed through a non-linear layer producing the sequence $\mathbf{s} \in \mathbb{R}^{N\times H}$, where H represents the length of the predicting horizon. Finally, viewing the output of the graph temporal-spatial attention block $\mathbf{s}$ as a matrix encoding the shading information, the clear-sky data is additionally introduced before the last MLP; see Figure 4.1. Hence, $\mathbf{s}$ is concatenated with an embedding of the clear-sky data $\mathbf{y}$ and Hadamard product between $\mathbf{s}$ and $\mathbf{y}$.

The dynamical adjacency matrix $\boldsymbol{\gamma^t}$ of the graph is inferred in the spatial attention block using masked attention, in order to inject the graph structure. The adjacency values γ_{ij}^t in each graph are computed for all $j \in \mathcal{N}_i$ where $\mathcal{N}_i$ is the neighbourhood of the node i on graph G^t. Thus, it is crucial to define the support of the neighbourhood set $\mathcal{N}_i$ with a predefined mask. To this end, the mask is initialized using the Euclidean distance in the k-nearest neighbour (knn) algorithm. Thus, if the distance e_{ij}^t is larger than the one defined in knn algorithm, we will mask the adjacency entry γ_{ij}^t with 0. For different parts of the forecasting horizon H, the different neighbourhoods are taken into account and different spatial attentions are calculated. Multi-window mechanism is introduced in the spatial attention, such that for short-, medium- and long-term dependencies we define $C = 3$ spatial attention windows. Thus, for each of the 3 windows, the k-neighbourhood is considered to be 30%, 50%, 100% of total number of nodes, respectively.

The spatial window c yields different spatio-temporal feature matrices $\mathbf{s_c} \in \mathbb{R}^{N\times \frac{H}{C}}$ for $c \in \{1,\ldots,C\}$. Let the matrix $\mathbf{G_c} \in \mathbb{R}^{N\times Tf''}$ for window c be defined by stacking $t \in \{1,\ldots,T\}$ feature vectors $\mathbf{g_t^{i,c}} \in \mathbb{R}^{f''}$ for node i. The vector of embedded features $\mathbf{g_t^{i,c}} \in \mathbb{R}^{\mathbf{f''}}$ per window t and node i is given by:

$$\mathbf{g_t^{i,c}} = \sum_{j\in\mathcal{N}_i^c} \gamma_{ij}^{t,c} \mathbf{Q_{spat}^t} \mathbf{h_t^i}, \tag{4.10}$$

where dynamical adjacency entry $\gamma_{ij}^{t,c}$ is defined for each spatial window c is defined on different neighbourhood $\mathcal{N}_i^c$. Corresponding different sets of weight matrices $\mathbf{W_{spat}^{i,c}} \in \mathbb{R}^{f''\times f'}$, $\mathbf{a_{spat}^{i,c}} \in \mathbb{R}^{1\times 2f''}$ are learnt for each window c. Hence, C spatial attention windows are calculated for different C parts of predicting horizon H which yields the feature matrix $\mathbf{s_c}$:

$$\mathbf{s_c} = \sigma\left(\mathbf{G_c}\mathbf{B_c} + \mathbf{b_c}\right), \tag{4.11}$$

where $\mathbf{B_c} \in \mathbb{R}^{Tf''\times \frac{H}{C}}$, $\mathbf{b_c} \in \mathbb{R}^{\frac{H}{c}}$ represent weights and biases in the last linear layer of each spatial

attention window $c \in \{1,\dots,C\}$. Finally, the feature matrix $\mathbf{S}$ is concatenation of all C feature matrices across all windows $S = [\mathbf{s_1}||\dots||\mathbf{s_c}]$. The feature matrix is only then concatenated with clear sky data. The predicted power for N nodes and H steps ahead is defined with

$$\hat{\mathbf{p}} = MLP\left([\mathbf{y}||\mathbf{S}||\mathbf{y}\odot\mathbf{S}]\right), \tag{4.12}$$

where the Hadamard product is denoted with $\odot$.

Thus, the TSM-GAT model finds dynamical adjacency which is different for each temporal window and for different predicting horizons. We can easily find the root cause and interpret on which nodes the model is focusing for short-, medium- and long-term forecasts within some given prediction horizon H. This approach makes the temporal-spatial multi-window graph attention model interpretable for the time series forecasting task.

4.3.5 Architecture configuration

We divide the input M lags among T overlapping temporal windows of size $2m$. The window size is calculated based on the overlap m, where $m = \frac{M}{T+1} \in \mathbb{N}$. One entire day of input data is chosen as the length M of the past observations, in order to find periodic dependencies and temporal shifts. The number T of the input overlapping sliding windows is 15 and the window length is 12 time steps which is equivalent to the past 3 hours. Therefore, the size of the shift (and the overlap) between the two consecutive temporal windows is 1.5 hours and these values have been chosen empirically. This choice is made with an assumption that the weather does not change drastically more than once within this horizon for each station. Thus, by embedding features in each window, we are embedding the change in the weather information each 1.5 hours. In the spatial attention the output feature space is then divided into $C = 3$ different windows, thus, in each window the length of input clear sky data is $\frac{H}{3} \in \mathbb{N}$; see Figure 4.1. The length of the window can be adapted in case where $\frac{H}{3} \notin \mathbb{N}$ such that an approximately similar number of time steps is in each window.

The k-neighbourhood in the first window represents the closest 30% of the total number of nodes and it finds spatial correlations between the nodes focusing on the first two hours ahead prediction, i.e. it is assumed that only the closest neighbours contribute to first two hours prediction. The k-neighbourhood in the second window is limited to 50% of the total number of nodes, focusing on the 2-4 hours ahead prediction. The assumption for the second window is that the neighbourhood should analyse nodes that are further away to consider longer term features. The neighbourhoods of 30% and 50% of the total number of nodes are selected by hyper-tuning parameters to improve accuracy. Finally, the fully connected graph is used to find the correlations between nodes in the last time window that focuses on the 4h to 6h ahead prediction. The goal is to take into account both the closest and the furthest away nodes, when making predictions in the last window of the forecasting horizon. An MLP is used at the output of the second attention block to transform the temporal-spatial attention block outputs into the power production power production $\hat{\mathbf{p}} \in \mathbb{R}^{N\times H}$.

4.4 Experiments

We applied the TSM-GAT architecture on PV production data from real and synthetic datasets. Furthermore, we compared the performance of the TSM-GAT against state-of-the-art models for both single- and multi-site forecasts on the real dataset. Single-site forecasts are used to compare results against models that use NWP data and with traditional model that uses only power data. On the other hand, multi-site forecasts are used in order to compare results with models that use multi-site PV power data, for both real and synthetic datasets. Then we analysed what drives the performance of TSM-GAT in multi-site forecasts, on the real dataset, and conducted the study in order to compare the multi-window approach with the multi-head approach.

4.4.1 Datasets

The real dataset used in our study consists of records from 304 PV plants across Switzerland (Carrillo et al., 2020) for two years (2016-2017). The PV plants are spread inhomogeneously over the entire country, with a density reflecting the population density. The spatial distribution of this dataset is shown in Figure 4.10. The dataset has a 15-minute resolution. The second dataset used is a synthetic dataset from NREL, with 405 PV stations distributed across California, USA, with 15-minutes resolution. This synthetic dataset consists of PV production data simulated from weather data for 1 year (2006) and is publicly available[2]. For both datasets, all available PV stations are used as both input and forecasting nodes.

4.4.2 Benchmark models

Five state-of-the-art methods were used as a benchmark in the multi-site forecasting evaluation. The first baseline is the spatio-temporal autoregressive model (STAR) (Carrillo et al., 2020) which is a linear autoregressive model. It uses the group Lasso regularization to promote sparse solutions, thus, creating the effect of selecting relevant nodes for the prediction of each site. The second and the third baselines are the recently proposed graph-based Graph convolutional long-short term memory neural network (GCLSTM) and Graph Convolutional Transformer (GCTrafo) (Simeunović et al., 2022a). Both of these models have an encoder-decoder structure, however, the former is the recurrent-based graph convolutional model whereas the latter uses a transformer architecture coupled with graph convolution. The fourth baseline for multi-site forecasting is the non-graph-based space-time convolutional neural network (STCNN) (Jeong and Kim, 2019). It uses a greedy-adjoining algorithm to rearrange the stations based on their geographical proximity, before applying 2D convolution layers, as in image processing. The fifth baseline is TS-multi-head-GAT which represents modification of TSM-GAT where multi-head mechanism was used instead of multi-window.

For the single-site forecasts we used three state-of-the-art models and a a simple smart persis-

[2] https://www.nrel.gov/grid/solar-power-data.html

tence model as benchmarks. The first two models, which use NWP data, are Support Vector Regression (SVR) and the encoder-decoder long-short term memory network (EDLSTM). Both models use predictions of global irradiance and temperature as input weather data, which were obtained from Meteotest[3] with a temporal resolution of 1 hour. Thus, in order to have 15 minutes resolution predictions, a simple sample-and-hold interpolation was applied to the NWP data. SVR is chosen as benchmark since Boegli et al. (2018) have shown that SVR outperforms several state-of-the-art models for intra-day forecasts. The second model, EDLSTM is a state-of-the-art encoder-decoder model, especially suited for time-series processing (Gao et al., 2019; Hamberg, 2021). The EDLSTM uses past PV production data as well as NWP data to make a forecast. The encoder uses as inputs past observed weather and PV power data in order to estimate the state of the system. Then the decoder uses these estimations and NWP over the prediction horizon as input to the decoder. The third benchmark model is a Seasonal Auto-Regressive Integrated Moving Average with eXogenous factors (SARIMAX). The SARIMAX model uses past PV power and clear-sky irradiance as inputs.

4.4.3 Data preprocessing and Training

The power data is normalized for both real and synthetic datasets by the maximum power production over the training year for TSM-GAT, TS-multi-head-GAT, GCLSTM, GCTrafo, SARIMAX, EDLSTM and STCNN. The STAR model, on the other hand, requires careful normalization in order to extract daily profiles from the historical data. The real dataset in Switzerland has two years of data, thus, for non-linear models, the first year (2016) is taken as training dataset, and the following year (2017) represents the evaluation dataset. On the other hand, the synthetic dataset in California has only one year of the data (2006). Therefore, the training dataset is from January until the end of July, and evaluation dataset is from August until December of 2006.

STAR and SARIMAX are linear autoregressive models, thus, in order to take into account seasonal weather changes, for the real dataset, the evaluation year (2017) is divided in small test batches. The models were trained over a period of data which is taken prior to each test batch. The STAR model coefficients were fitted over a period of two months and then used to predict the power of the next two weeks. SARIMAX coefficients were fitted over a period of three days and then used to predict power of the following day. The parameters for STAR and SARIMAX methods are re-fitted in a rolling window fashion for every two months and for every three days, respectively. The hyperparameters used in the TSM-GAT and multi-site baseline models are presented at the end of this Section. All models were trained on a workstation with 16 physical cores, 128 GB of RAM memory and an Nvidia RTX 2080 Ti GPU.

[3] https://meteotest.ch/en/

4.4.4 Evaluation and metrics

The model performance was assessed using several metrics. We use the peak normalized root-mean-square error (NRMSE) and the average normalized mean absolute error (NMAE), defined in previous Chapter. The second type of metric represents the difference in the shape between the predicted power and the ground truth. Dynamic time wrapping (DTW) is used as a measurement of the distance between shapes in time series. DTW distance $\delta_v(\tau+i,\tau+j)$ for node v and between time series $\hat{p}_v(\tau),\ldots,\hat{p}_v(\tau+i)$ and $p_v(\tau),\ldots,p_v(\tau+j)$ is defined as the minimum of accumulated distances $dist(p_v(\tau+i),\hat{p}_v(\tau+j))$:

$$\begin{aligned}\delta_v(\tau+i,\tau+j) = dist(\hat{p}_v(\tau+i),p_v(\tau+j))+\\ \min\{\delta_v(\tau+i-1,\tau+j),\delta_v(\tau+i,\tau+j-1),\\ \delta_v(\tau+i-1,\tau+j-1)\}\end{aligned} \tag{4.13}$$

where Euclidean distance was used as the distance measurement. DTW is an indication of how good a model is at predicting shapes, especially clouds since they largely affect the shape of production. For calculations we used the approximation of DTW with linear time complexity presented in Salvador and Chan (2007).

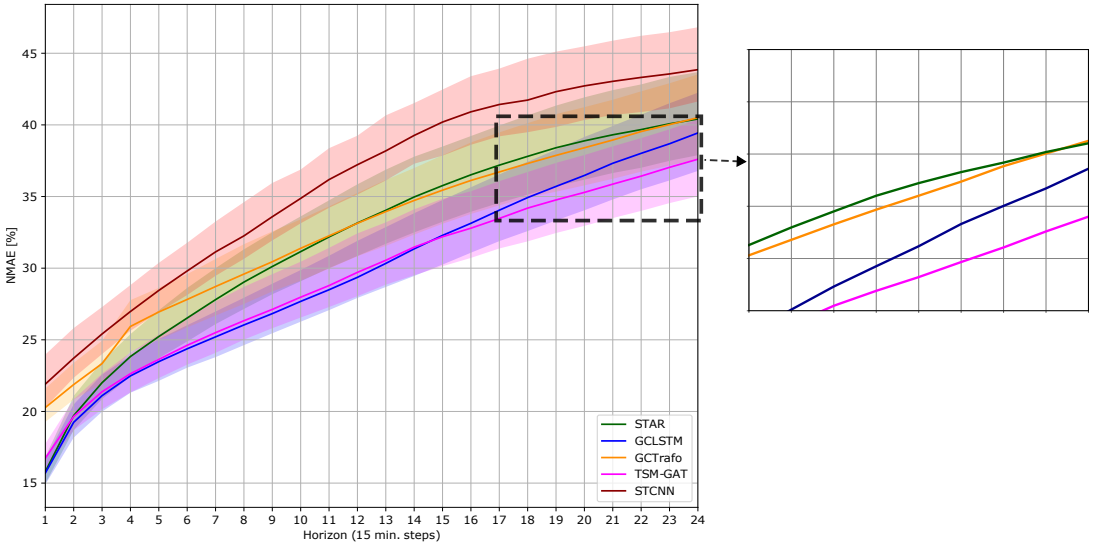
We classified days based on production variability into sunny, cloudy and variable in order to analyse the impact of weather conditions on the error rate and shape difference of time series, following definition from Van Haaren et al. (2014). The main difference is that instead of irradiance in our case the input data is PV power data. Two metrics were used for this classification: daily aggregate ramp rate (DARR) and daily index K_v^d, defined in Nespoli et al. (2019). The DARR is a simple metric given by

$$DARR_d = \sum_{\tau\in I_d}\frac{|p_v(\tau)-p_v(\tau+1)|}{p_v^{max}}, \tag{4.14}$$

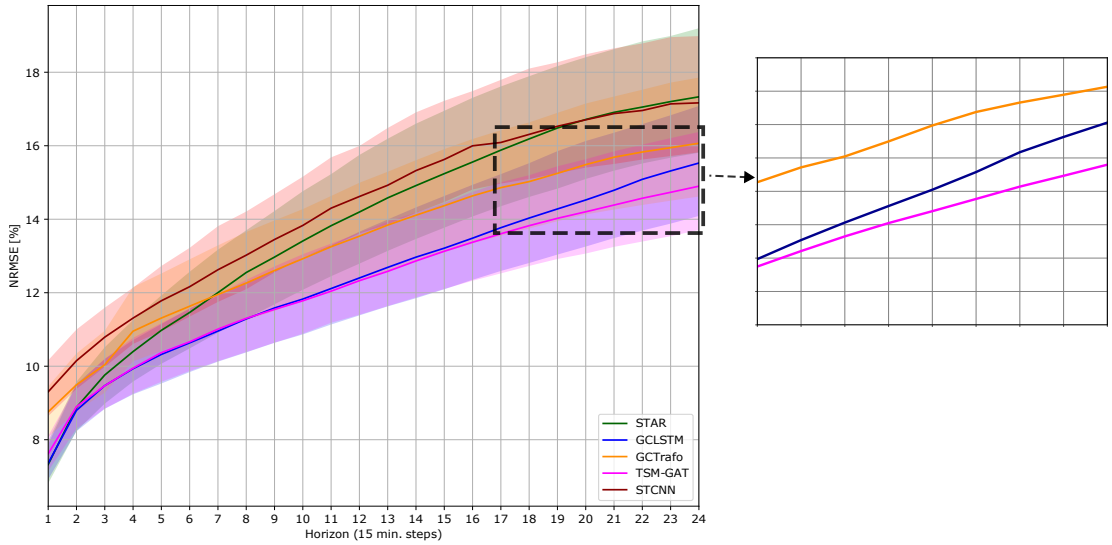
where I_d represents the set of the daily values (night values are not taken into account) during the day d. Days when $DARR_d > 3$ are classified as variable days, caused by having sun at some parts of the day and then clouds for the rest of the day, with sudden peaks and drops, which is essentially causing the high variability. However, the DARR metric has one limitation when it comes to classifying sunny and overcast days since overcast days show small rates, like the sunny days with $DARR_d < 3$. Therefore, to account for overcast days the additional metric with the daily index is introduced. For each time step the daily index K_v^d for each node and each day d is calculated

$$K_v^d = \frac{1}{|I_d|}\sum_{\tau\in I_d}\frac{p_v(\tau)}{y_v(\tau)p_v^{max}} \tag{4.15}$$

where $y_v(\tau)$ is the normalized clear-sky irradiance at site v. As a threshold value for classification we chose $K_v^d = 0.3$, where values above the threshold are indicate a sunny day, whereas lower values imply that the day in question is cloudy.



(a)



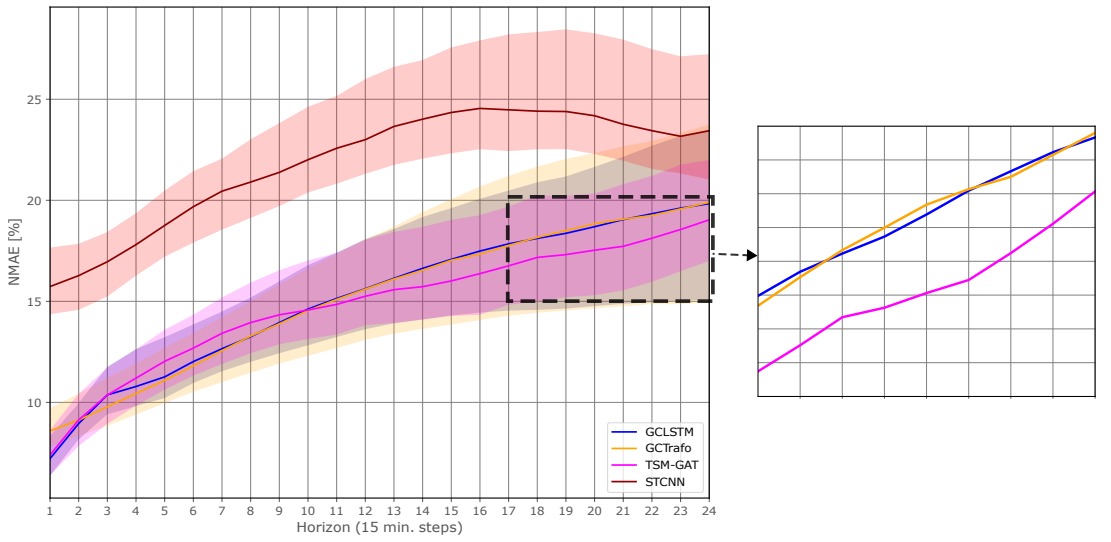
(b)

Figure 4.2: Error comparison of TSM-GAT and state-of-the-art models for six-hour ahead prediction for the real dataset in Switzerland with magnified part between 4 and 6 hours ahead prediction. Solid line shows the median value among all nodes, shaded bands show the interquantile range among all nodes. a) Forecast NMAE for the real dataset. b) Forecast NRMSE for the real dataset.

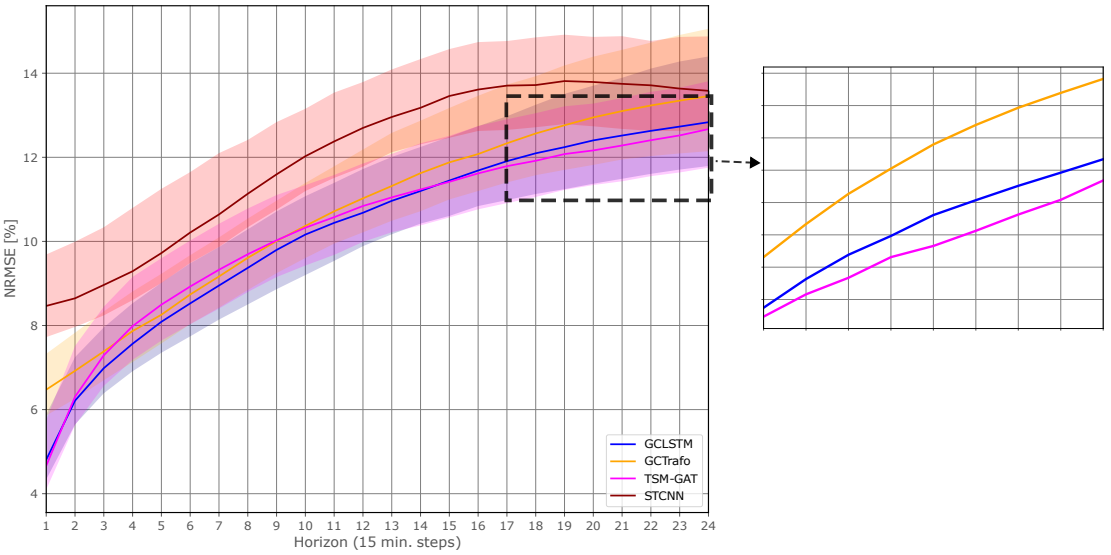
4.5 Results

4.5.1 Prediction accuracy

In Figure 4.2 the NRMSE and NMAE evolution of the TSM-GAT is compared to the STAR, GCLSTM, GCTrafo and STCNN models over a predicting horizon of 6 hours ahead, with a



(a)



(b)

Figure 4.3: Error comparison of TSM-GAT and state-of-the-art models for six-hour ahead prediction for the synthetic dataset in California with magnified part between 4 and 6 hours ahead prediction. Solid line shows the median value among all nodes, shaded bands show the interquantile range among all nodes. a) Forecast NMAE for the synthetic dataset. b) Forecast NRMSE for the synthetic dataset.

15-minute resolution for the real dataset in Switzerland. The shaded regions represent the inter-quartile (25%-75%) error range and solid lines represent the median of error over all

nodes. Although the GCLSTM method is on par with the TSM-GAT for the first hour ahead prediction, the TSM-GAT outperforms the GCLSTM from four to six hours ahead predictions. The region of interest, between four and six hours ahead of prediction, is highlighted and shown magnified on the right. Furthermore, the lower error slope and high accuracy in the 5th and 6th hour ahead prediction of the TSM-GAT could make it a promising multi-site model for longer prediction horizons. NRMSE is more sensitive to outliers since it considers squared errors. Therefore, NMAE is introduced in addition and the error evolution of NMAE is visually shown in Figure 4.2a. The GCLSTM still has the lowest NMAE for 1 to 4 hours ahead predictions. However, the TSM-GAT has lower errors than any other state-of-the-art model for 4 to 6 hours ahead forecast on both metrics. Since the error at the end of the forecasting horizon becomes higher, the advantage of TSM-GAT in capturing long-term correlations becomes noticeable.

In Figure 4.3 the NRMSE and NMAE evolution of the TSM-GAT is compared to the non-linear methods, GCLSTM, GCTrafo and STCNN, over a prediction horizon of 6 hours, with a 15-minute resolution for synthetic dataset in California. The authors of STCNN, Jeong and Kim (2019), report lower error in their paper, since they use only a subset of 238 nodes and hourly temporal resolution. Thus, they average the data which leads to smoothing signals and lower error rate. Similarly to the performance on the real dataset, the TSM-GAT outperforms state-of-the-art non-linear methods for 4 to 6 hours ahead prediction on the synthetic dataset. Thus, the region of interest, between four and six hours ahead of prediction, is highlighted and shown magnified on the right. TSM-GAT has lower error slope, while staying on par with GCLSTM in the first four hours of the predicting horizon. The performance of all models on the synthetic dataset shows lower error compared to the real dataset. This is the result of smoothing in the synthetically created PV power dataset, generated from irradiance data using the Sub-Hour Irradiance Algorithm (Hummon et al., 2012).

We also compare forecasting results for one site in the central part of Switzerland, Bätterkinden which is close to Bern. Figure 4.4 shows the NRMSE evolution for the proposed method and benchmark single-site models. The error of the persistence model is up to 25% in the last hour of prediction. However, the NRMSE of persistence model is clipped at 20% in Figure 4.4 in order to be able to see the error of other models more clearly. The error rate in single-site comparison shows that TSM-GAT outperforms all single-site models. The exception is in the sixth hour ahead, where for the last four steps of the horizon (21st to 24th step ahead), EDLSTM yields lower error than the proposed method. Although EDLSTM has higher error from 0 to 5 hours ahead, compared to the proposed TSM-GAT, it clearly benefits from NWP information at the end of the horizon since NWP data improves predictions from 6 hours ahead to one day ahead.

4.5.2 Comparative analysis

In order to better understand what drives each model, discover pitfalls or particular advantages we use the DTW metric in addition to NRMSE and NMAE. We make comparison only with

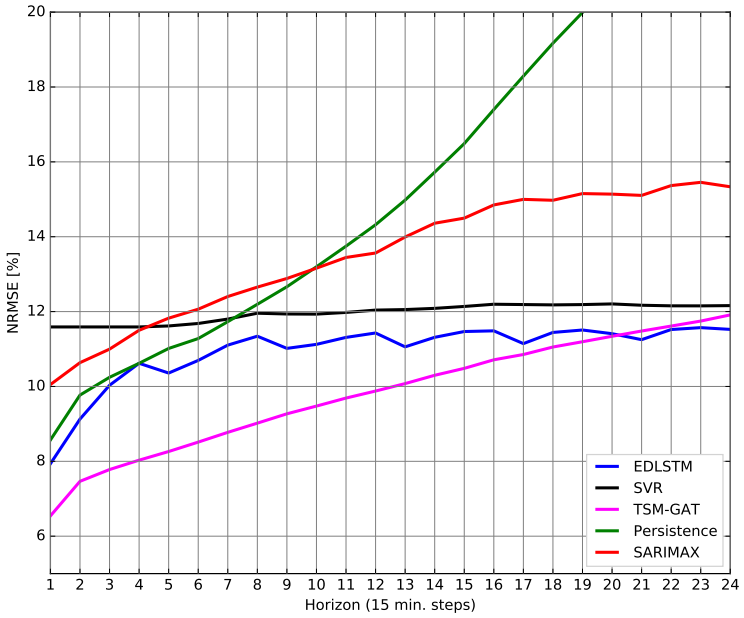


Figure 4.4: Single-site error comparison between the TSM-GAT and state-of-the-art single-site models.

the real dataset, in order to analyse better the impact of weather conditions on PV power forecast, since on the synthetic dataset the signals are smoothened out and the sharp weather transitions are less pronounced. Furthermore, the comparison is focused only on the GCLSTM and TSM-GAT since they have the lowest errors. Table 4.1 illustrates the similarity in shape between time series, obtained based on DTW metric for sunny, cloudy and variable days. Despite the lower NMAE error for the GCLSTM in the first two hours of the prediction, TSM-GAT has lower DTW values during cloudy and variable days in Table 4.1 as well as lower NRMSE on 1 to 6 hours ahead forecasting horizon. This suggests the ability of TSM-GAT to accurately predict abrupt changes in the cloud movement or day to day weather. Hence, it yields a daily time series shape closer to the ground truth, which is important for energy management application.

We analyse the correlation between the NRMSE and the number of variable, sunny and cloudy days. Nodes with the higher number of variable days have higher error for 6 hours ahead prediction, whereas for the sunny and cloudy days, there is no clear dependence. The analysis of the error with respect to the distance to the centroid indicate that methods performs similarly for the central nodes and on the edges on the graph. Furthermore, the analysis of the error with respect to average distance to the closest neighbours shows that being in a cluster or being isolated, does not affect the performance of the model. The results are shown and discussed in more detail later, along with the limitations of the model.

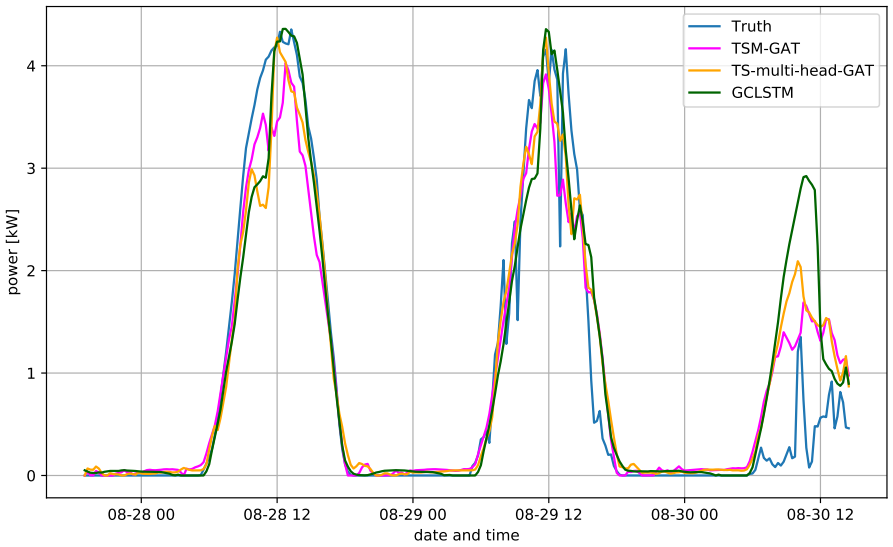


Figure 4.5: Ground truth and prediction for 6 hours ahead.

4.5.3 Effect of multi-window approach

A study has been conducted on the same dataset to verify the hypothesis regarding the higher accuracy and interpretability of the proposed multi-window attention against multi-head attention. We modified TSM-GAT by replacing the multiple windows with multiple heads, creating the temporal-spatial multi-head graph attention mechanism (TS-multi-head-GAT). Here the TSM-GAT and TS-multi-head-GAT are evaluated. Although the GCTrafo model also uses a multi-head approach, its model architecture is completely different compared to the TSM-GAT. Moreover, it is reporting higher error compared to both temporal-spatial GAT architectures (multi-window and multi-head), thus, it has not been used in this study. The main difference between TSM-GAT and multi-head approach is that the TSM-GAT has a multi-window approach where each window focuses explicitly on one part of the predicting sequence, whereas the multi-head approach focuses on the entire predicting sequence at once. The number of attention heads and attention windows in this part of the study is the same: $C = 3$.

The first advantage of the TSM-GAT model is learning lower number of parameters in the spatial attention, in the last linear layer before concatenation with clear-sky data. The size of the weights and biases in each attention window is $\mathbb{R}^{Tf'' \times \frac{H}{C}}$, $\mathbb{R}^{\frac{H}{C}}$, respectively, since it is embedding feature sequence Tf'' into the length of the future window sequence $\frac{H}{C} \in \mathbb{N}$. On the other hand, in the multi-head approach, with C heads, the sizes of weights and biases in this linear layer are $\mathbb{R}^{Tf'' \times H}$,$\mathbb{R}^{H}$. Moreover, multi-window approach requires an additional layer, to reduce the number of concatenated features from CH to H in order to make it possible to integrate the clear-sky data. On the other hand, TSM-GAT has the advantage of

Table 4.1: Shape distance of predicted and observed time series using DTW

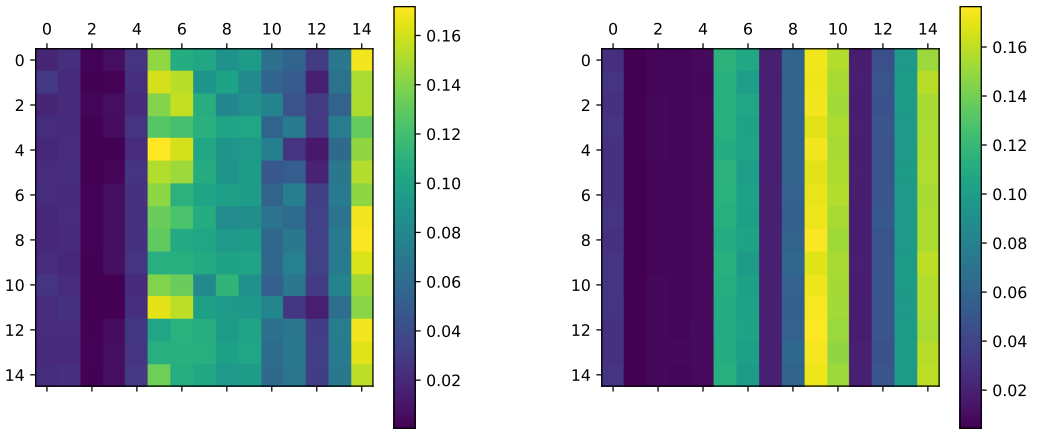
Prediction ahead:	Weather conditions	TSM-GAT	GCLSTM	TS-multi-head-GAT
1h	overall	**1.42**	1.69	1.69
	sunny	**1.15**	1.36	1.40
	cloudy	**0.65**	0.92	0.83
	variable	**2.87**	3.24	3.26
3h	overall	**1.86**	2.08	2.08
	sunny	**1.52**	1.64	1.76
	cloudy	**1.18**	1.46	1.24
	variable	**3.37**	3.61	3.72
6h	overall	**2.47**	2.72	2.75
	sunny	**2.21**	2.24	2.50
	cloudy	**1.93**	2.56	2.21
	variable	**3.69**	3.81	3.98

focusing on the different parts of the input sequence due to the fact that windows do not have overlap in specializations. Trainable parameters and defined variables are dependent on the number of sliding windows T, number of features f_{in}, f', f'', number of multi-head windows C and total number of nodes N. Although we have a restricted number of neighbours on which the model focuses when making the prediction, a full matrix-vector multiplication is calculated to find the correlation between the nodes in the dynamical adjacency matrix $\boldsymbol{\gamma^t}$. Only then the entries which are not in the predefined neighbourhood are masked with zeros. Since sparse matrix structures are not used, the number of the neighbours doesn't affect the computational complexity. Therefore, computational complexity and memory requirements scale with $\mathcal{O}(N^2 T)$.

To better understand the model we focus on the forecasting results from a specific moment in time and place. The analysis was made on 30th of August, which was the cloudy day, at 7 a.m. for the next 6 hours ahead prediction, which is the third day in Figure 4.5. We choose this day since it is cloudy one after the sunny days and this time since it is difficult to make forecast in the morning when the most recent past data is limited to only one hour of data and then followed by night values. We analysed temporal adjacency in detail for this example in Figure 4.6, and the spatial adjacencies in Figure 4.7 and Figure 4.8.

The temporal adjacency is shown in Figure 4.6. In this example the temporal attention of multi-window model focuses on the last temporal window before the prediction and on 2 windows just before the sunset of the previous day, which is intuitive. On the other hand, the multi-head model focuses on the last temporal window, but it also focuses on the temporal windows during night, which should not be relevant part of the input sequence.

The analysis of the full dynamical adjacency for only one snapshot in time with 15×304^2 entries would be difficult. Therefore, we will focus on the 15th window of adjacency, since for both models temporal attention is high in this window. Nodes above an arbitrary threshold



(a) (b)

Figure 4.6: Temporal attention between 15 overlapping windows. Darker colours signal lower attention. a) TSM-GAT. b) TS-multi-head-GAT.

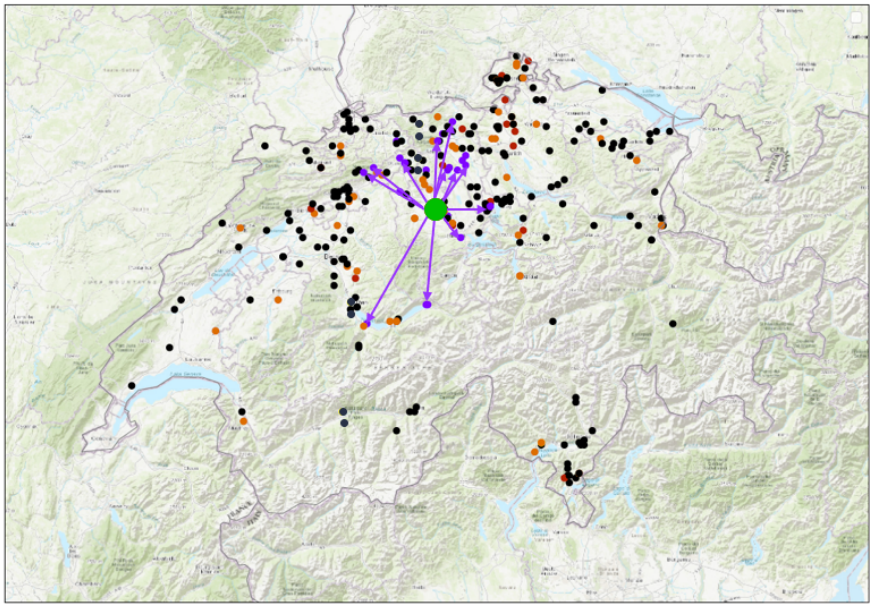
Table 4.2: Analysis of the nodes with the highest spatial attention coefficients

	Head/Window 1		Head/Window 2		Head/Window 3	
Important nodes	nodes	shared[%]	nodes	shared[%]	nodes	shared[%]
TS-multi-head-GAT	37	39%	54	54%	23	30%
TSM-GAT(multi-window)	30	16%	19	10%	52	9%

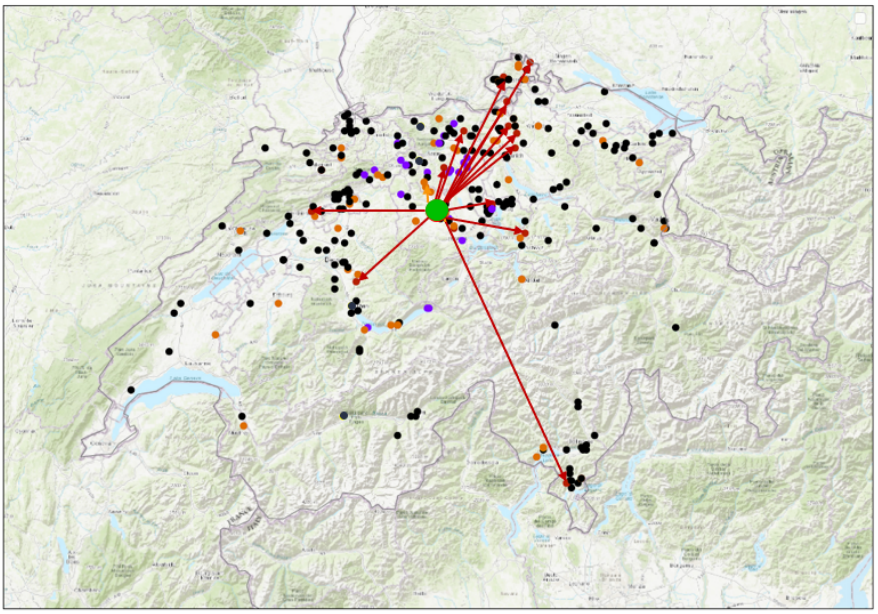
are considered important. In Figure 4.7 it is shown on which stations the model focuses in the first and the second forecasting window. In the first window, model focuses on the stations in the vicinity from the observed node, when making the forecast for up to two hours ahead. When forecasting from two to four hours ahead, the model focuses on the nodes that are a bit further away. Finally, when making forecast from four to six hours ahead, the model is focusing on the nodes that are close to the observed nodes, but also on the nodes that are very far away, see Figure 4.8. This indicates that model is able to capture both local and global correlations, thus, local and global dynamics.

In Table 4.2 the number of important nodes is shown, as well as the percentage of these important nodes from each window/head, which also represent an important node in the other head(s)/window(s). The percentage of the important nodes per head, which is shared with other head(s), is much higher than the percentage of nodes per window, shared with the other window(s). Thus, it is more difficult to interpret the results in the multi-head attention due to the high overlap. On the opposite, in the multi-window approach, each window has a lower percentage of the overlap. TSM-GAT benefits from the clear specialization since with the increase of forecasting horizon the average distance between the observed node and those to which the mechanism is focusing on increases as well, see Figure 4.9.

Finally, we compare errors of TSM-GAT, TS-multi-window-GAT and GCLSTM in Table 4.3 and Table 4.4. According to Murdoch et al. (2019) interpretability is defined as predictive accuracy,



(a)



(b)

Figure 4.7: Spatial attention between the observed node and its neighbours at different forecasting windows. The arrows are connecting the observed node (in green) and its neighbours with the highest attention coefficients. a) Spatial attention for the forecast up to two hours ahead. b) Spatial attention for the forecast from two to four hours ahead.

descriptive accuracy, and relevancy, where relevancy is judged by a human audience. We measured the predictive accuracy by using NMAE in Table 4.3 and NRMSE Table 4.4. Overall

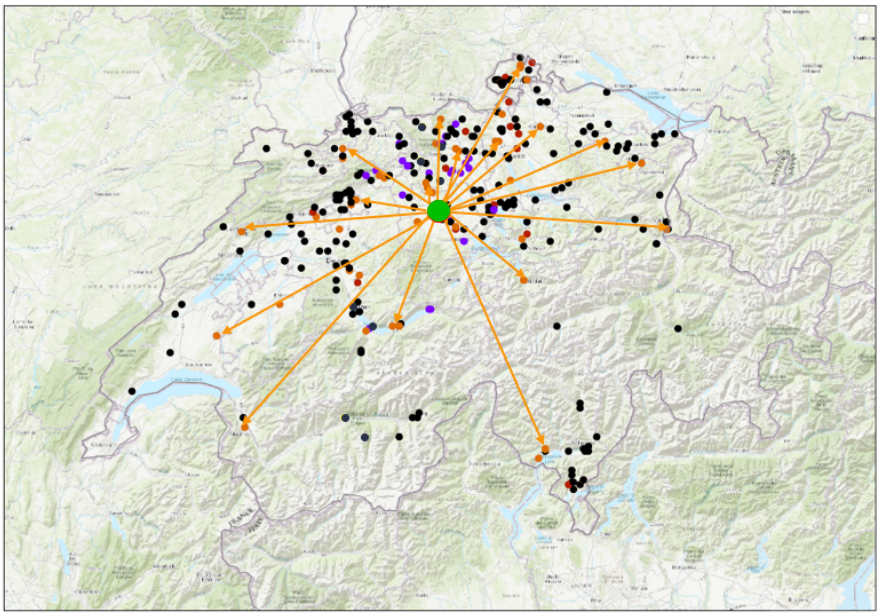


Figure 4.8: Spatial attention between the observed node (in green) and its neighbours for the third forecasting window. The arrows are connecting the observed node and its neighbours with the highest attention coefficients for forecast from four to six hours ahead.

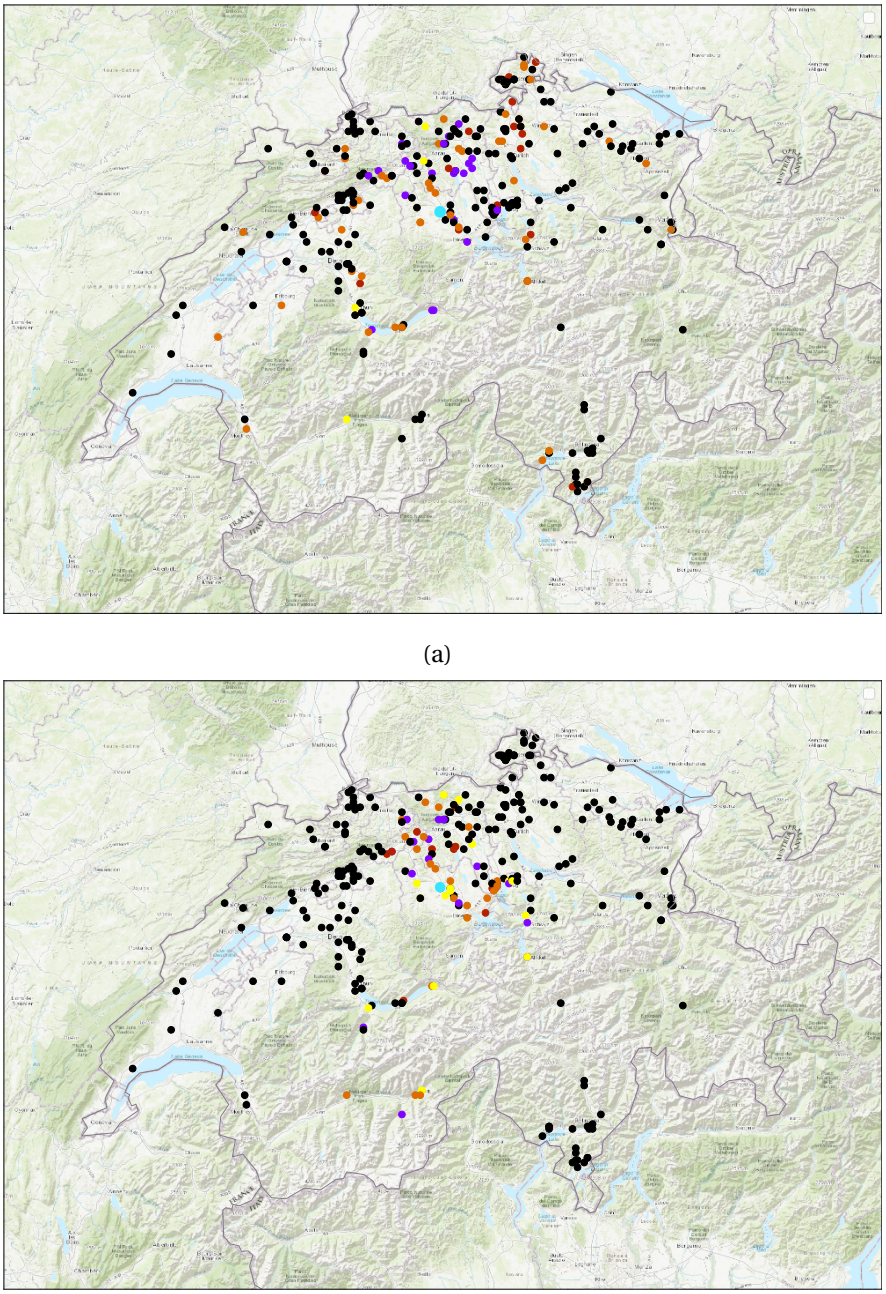
Table 4.3: Forecasting performance of TSM-GAT and baseline models using NMAE

	15min	1h	2h	3h	4h	5h	6h
GCLSTM	**15.70**	22.48	**26.04**	**29.36**	33.13	36.47	39.44
TSM-GAT	16.74	22.63	26.33	29.70	**32.79**	**35.29**	**37.60**
TS-multi-head-GAT	16.80	**22.44**	26.20	29.92	33.63	36.32	38.95

Table 4.4: Forecasting performance of TSM-GAT and baseline models using NRMSE

	15min	1h	2h	3h	4h	5h	6h
GCLSTM	**7.36**	**9.93**	**11.29**	12.40	13.48	14.52	15.53
TSM-GAT	7.61	**9.94**	**11.30**	**12.32**	**13.37**	**14.20**	**14.89**
TS-multi-head-GAT	7.57	9.99	11.40	12.47	13.72	14.80	15.72

the highest accuracy is for the TSM-GAT model. What is more, the descriptive accuracy is higher with the TSM-GAT since we can understand on which nodes the model is focusing for different parts of the forecasting horizon. However, this is not the case with the multi-head approach where the model is focusing on the same nodes when making 1 hour ahead and 6 hours ahead prediction. On the other hand, with recurrent structures in the GCLSTM it is extremely difficult to find where is mechanism focusing, since it required tracking the activations and checking their updates. Finally, relevancy is the last interpretation measure where DTW metric in Table 4.1 indicates lower shape difference to the ground truth of the TSM-



(a)

(b)

Figure 4.9: Map of the spatial attention in the last overlapping window. The prediction is made for the node in turquoise colour. Attention coefficients below the threshold are black. The purple, red and orange nodes have coefficient values above the threshold in the first, second and third attention head/window, respectively. Yellow nodes have values above threshold on which at least two out of three heads/window focus (shared between heads). a) TSM-GAT spatial attention. b) TS-multi-head-GAT spatial attention.

GAT, which is making this model more relevant for energy management application. Thus, this suggests that the TSM-GAT model with the multi-window approach is more interpretable than the multi-head mechanism or state-of-the-art model GCLSTM.

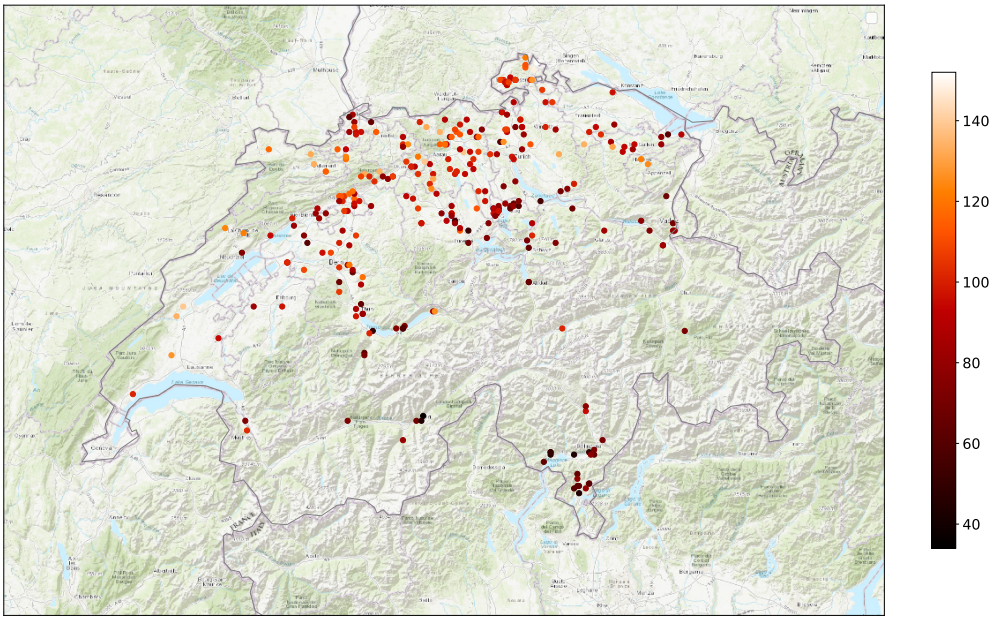
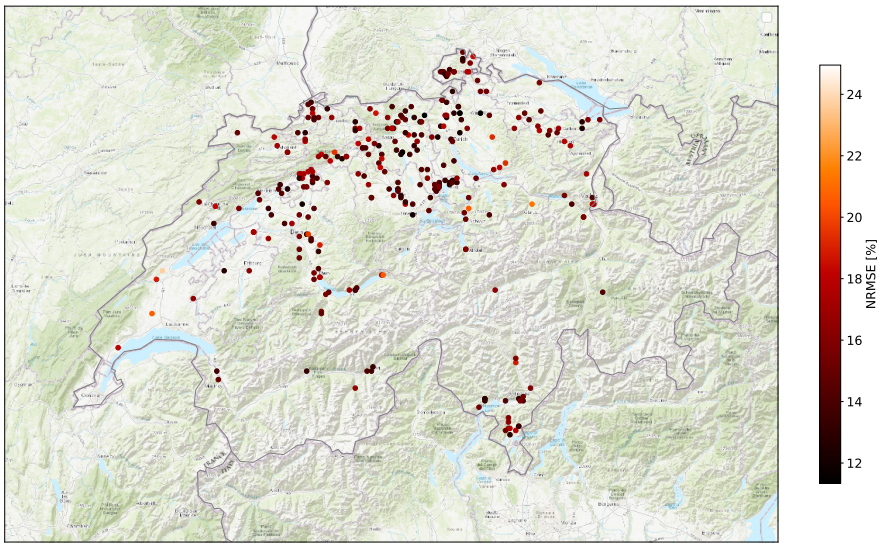


Figure 4.10: Number of variable days per node. The darker colours are signalizing the lower number of variable days in year 2017 per node.

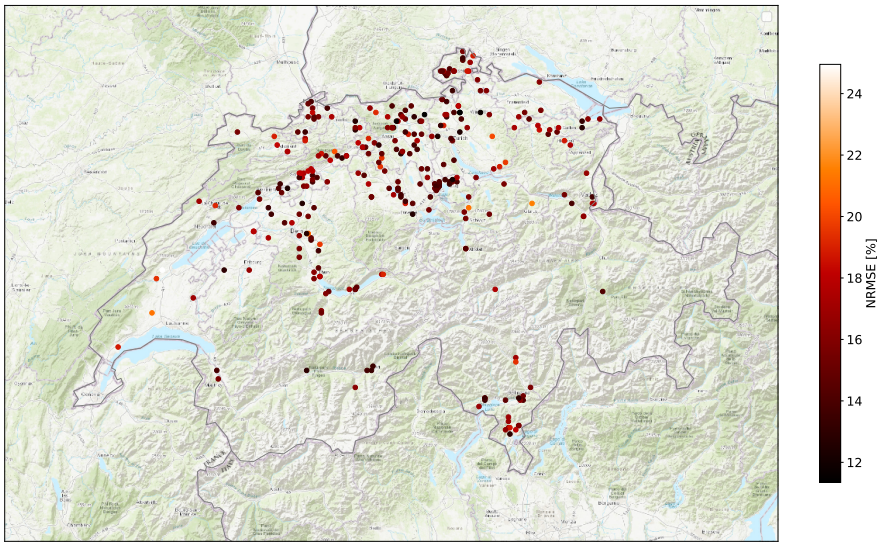
4.5.4 Analysis and limitations of the model

Both GCLSTM and TSM-GAT display a similar behavior in terms of correlation between error and the number of variable days per node. Figure 4.10 shows the number of variable days in a year per node, where darker colours indicate the lower number of variable days per node. Several nodes are in a dense cluster, and yet have more variable days than the rest of the nodes in their clusters. This indicates that they could have different micro-climate, shadowing effects or more foggy days than the rest of the nodes in their clusters. At these nodes both GCLSTM and TSM-GAT showed higher error for six hours ahead prediction; see Figure 4.11. Thus, the first limitation of the model is higher error in case of the nodes with more variable days than the rest of the nodes in the cluster. Relying on the neighbourhood information increases the error when the predicting nodes have much higher number of variable days than the close neighbours in their clusters. This represents an indication that the models are not able to capture the specific microclimate at these distinctive nodes. As expected, from Figure 4.12 we can see that the higher the number of variable days per node, the higher the error per node. However, there is no clear dependence between the error and the number of sunny or cloudy days.

Another limitation of the model is high computational complexity compared to GCLSTM,



(a)

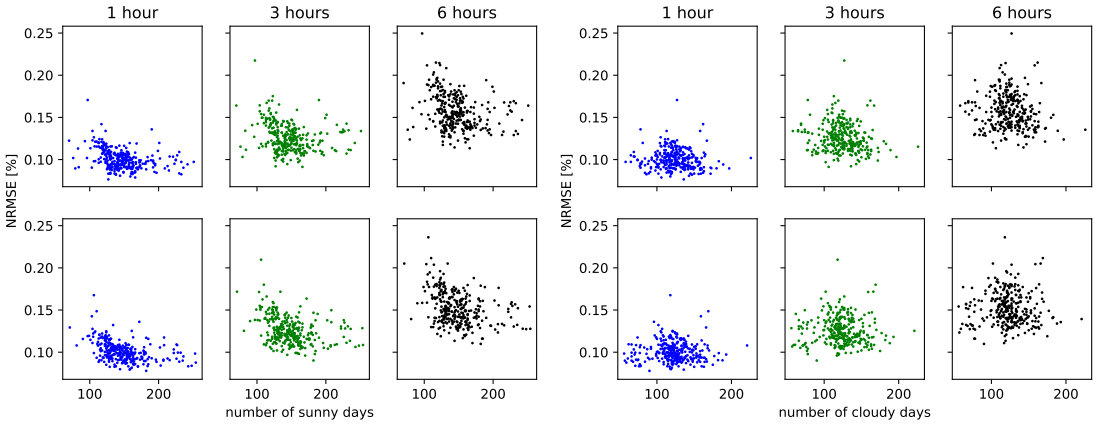


(b)

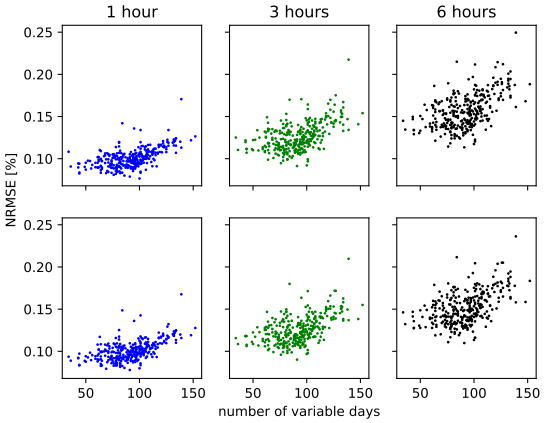
Figure 4.11: NRMSE for 6 hours ahead prediction per node (in [%]) in the year 2017. Darker colours are indicating the lower NRMSE per node. a) TSM-GAT b) GCLSTM.

since the computational complexity of the TSM-GAT model scales with $\mathcal{O}(N^2T)$ where N is the total number of nodes, T is the number of temporal windows. Oppositely, GCLSTM computational complexity scales with $\mathcal{O}(Nkl)$, where k is the number of closest neighbours in GCLSTM, and l is the number of time steps taken into account for recurrence, such that $NT \gg kl$. Furthermore, in TSM-GAT model the number of learned weights is T times higher, increasing the memory requirements.

The analysis of the error with respect to distance from the centroid, where the centroid is



(a) (b)



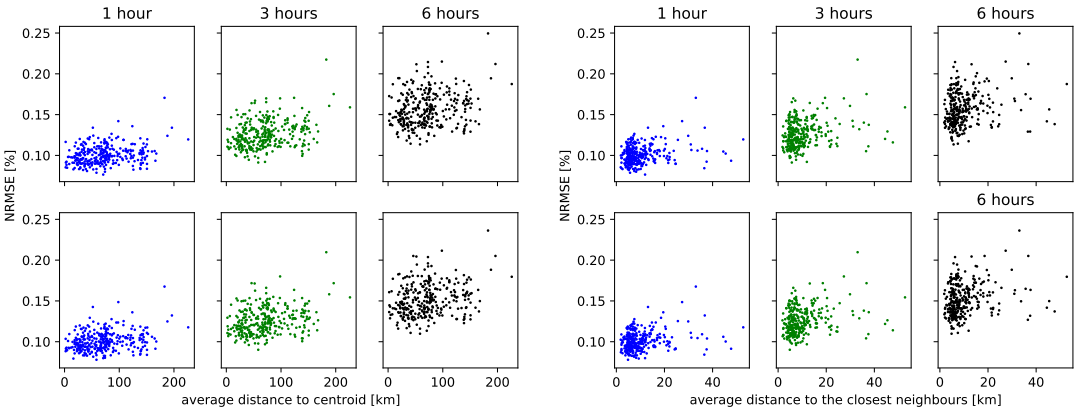
(c)

Figure 4.12: NRMSE of TSM-GAT model (bottom) and GCLSTM model (top) for 1,3,6 hours ahead prediction per node for 2017 year divided into 3 different type of days. a) Sunny days. b) Cloudy days. c) Variable days.

calculated as the average of nodes' coordinates, does not indicate a higher error for nodes that are further away from the central node, see Figure 4.13a, which indicates that the method performs similarly for the central nodes and the nodes which are on the edges of the graph. Figure 4.13b displays the error with respect to the average distance to five closest neighbours. It shows the advantage of the model that being in a cluster or oppositely, being isolated, does not affect the performance of the model.

4.5.5 Comparison with cloud-tracking model

In the previous subsection, we have already discussed that the high variability of the PV power production within one day represents one of the model's limitations. Since the high variability



(a) (b)

Figure 4.13: NRMSE per node for 1,3 and 6 hours ahead prediction with respect to different distances for TSM-GAT (bottom) and GCLSTM (top). a) NRMSE with respect to distance to the centroid. b) NRMSE with respect to distance to 5 closest neighbours.

of PV production mainly comes from cloud movement, we have compared our proposed model to CloudMove, a state-of-the-art commercial solution (Müller and Remund, 2014). CloudMove is the short-term forecasting service for irradiance and PV power production from Meteotest. CloudMove uses weather models and cloud positions from satellite images to propagate the cloud's movement in the future. Ground-based solar irradiance or PV production is also used to correct the forecasts. The cloud propagation is then used to forecast the solar irradiance for up to six hours ahead.

CloudMove yields state-of-the-art accuracy for predictions in the six-hour ahead horizon for irradiance and PV power forecasts. However, one of the pitfalls of this method is high inference time, such that spatio-temporal graph-based methods report acceleration of the forecast computation by a factor 100 (Carrillo et al., 2022).

The analysis includes comparisons in different seasons and weather conditions. A representative set of 18 locations and 21 days were selected to cover the whole range of possible conditions in Switzerland in terms of weather, terrain, and distance to other instrumented sites. Forecasts at these stations are used to evaluate the accuracy of proposed models against baselines. Figure 4.14 shows the NRMSE evolution over the forecasting horizon of 6 hours ahead with 15-minute steps. We compare the proposed TSM-GAT model with GCLSTM, smart persistence model and CloudMove model. Although the error spread is larger with graph-based methods (the GCLSTM and TSM-GAT) than with CloudMove for the first three hours of the forecast, for the rest of the forecasting horizon, the proposed model has lower the upper quartile of error values than the lower quartile of the cloud-motion tracking method. Furthermore, the TSM-GAT has the lowest NRMSE compared to the GCLSTM and cloud-tracking model CloudMove for the forecasting horizon above one hour.

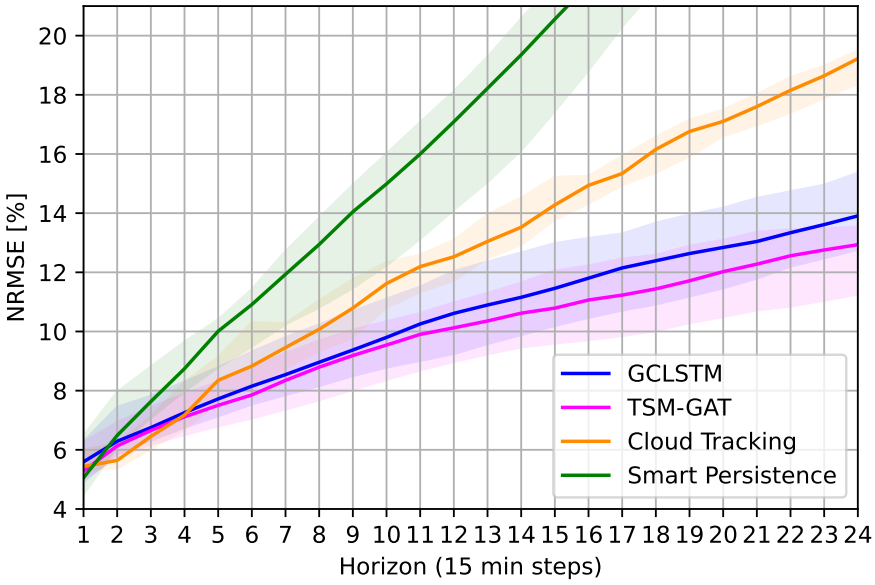


Figure 4.14: NRMSE evolution of GCLSTM, CloudMove, TSM-GAT and persistence model over the forecasting horizon of 6 hours ahead in steps of 15 minutes.

4.5.6 The effectiveness of the multi-window approach

In the study regarding the effectiveness of multi-window approach over multi-head we analyse the dynamical adjacency and temporal attention coefficients obtained from the two models in question during a cloudy day in Eich, Switzerland. This node is selected due to its central position, marked with turquoise colour in Figure 4.9. For illustration purposes, the predictions for 6 hours ahead made by GCLSTM are also shown in Figure 4.5 and compared to temporal-spatial graph attention model. The analysis was made on 30th of August, which was the cloudy day, at 7 a.m. for the next 6 hours ahead prediction. We choose this time since it is difficult to make forecast in the morning when the most recent past data is limited to only one hour of data and then followed by night values.

The temporal adjacencies of TSM-GAT and TS-multi-head-GAT for this prediction are shown in Figure 4.6. TSM-GAT model has the highest attention in the 15th, then 6th and 7th overlapping window, whereas multi-head model has the highest attention on the 10th, and then 11th and 15th overlapping window. This means that TSM-GAT pays the most attention to the observations in the morning of the current day and then on the last 4 hours of non-zero PV production in the previous day. This could be interpreted as looking at the past values during the cloudy morning and the last values before sunset of the previous day. Since the previous day was much sunnier it would not make sense for the algorithm to focus on other parts of the sequence. On the other hand, multi-head approach pays the most attention to the 10th and 11th window, which is during the night and then on the 15th window. It has the

highest attention values for the night windows when production is zero, which should not be a relevant part of the input data.

The next part of the study considers the dynamical spatial adjacency. We chose 15th window of the spatial attention for analysis, since both the multi-head and the multi-window architectures have high attention coefficients in the last temporal window. For both models we have chosen the arbitrary value 0.0035 as a threshold, such that nodes with coefficients above the threshold are considered important when making the prediction. Figure 4.9 shows the map of important nodes represented with different colours based on the window/head they belong to, when making the prediction for the chosen day. The first difference between the models is that in the multi-window approach the model is focusing on the closer nodes when making prediction for up to 2 hours ahead and it is focusing on further away nodes when making a prediction for 4-6 hours ahead. Intuitively, this represents a clear advantage compared to multi-head attention, since in the multi-head attention all heads give the highest focus on the similar small neighbourhoods. Furthermore, in this example the multi-window approach in the first window, coefficients are above the threshold for the self-attention, which is reasonable since for the short-term forecasts the persistence model gives the best results. Therefore, it could be interpreted as paying the most attention to its own past data when making forecasts up to two hours ahead. However, in the multi-head approach, none of these three attention heads have high self-attention coefficient values in the adjacency. Furthermore, in the multi-head approach the first and the last heads share one out of two nodes with highest attention. In the multi-window approach all three windows have different nodes with highest coefficient, meaning that in each window different nodes are selected as the most important.

4.6 Conclusions

A novel method, TSM-GAT, for capturing dynamically changing spatio-temporal correlations in deterministic PV power forecasting has been introduced. It was evaluated on real and synthetic PV production datasets and the performance was compared against state of the art, for both multi-site and single-site models. The TSM-GAT outperformed state-of-the-art methods from 4 to 6 hours ahead prediction on the real and synthetic datasets. Results suggest that the TSM-GAT model is better than the state of the art at capturing shadowing, including cloud motion, and weather changes since it yields signal shapes closer to the ground truth on the entire horizon. Thus, it is addressing the limitation of cloud prediction in the spatio-temporal task, by directly capturing dynamically changing adjacency matrices for different parts of predicting horizon. A study was conducted to analyse the difference between the widely used multi-head attention and the proposed multi-window attention. It indicates that the proposed TSM-GAT model is more interpretable, and therefore more suitable for the prediction of multi-site time series driven by physical phenomena, such as PV and wind forecasting. This model could be used for other time series forecasting tasks where physical phenomena are modelled or when it is important to understand what influences the forecast. An in-depth analysis was performed in order to better understand what drives the performance

on PV dataset. It is shown that the distance between PV power plants does not affect accuracy of the forecast, which is important for countries where PV stations are not homogeneously and densely distributed. Among the investigated state-of-the-art methods, the TSM-GAT model has the lowest distance between predicted and ground-truth signal shapes, as well as the lowest error for 4 to 6 hours ahead forecasts, which is important for energy trading and energy management. On top of that, the lower dependence of its error on the time horizon makes TSM-GAT very promising for longer-term (day-ahead) predictions.

The architecture is scalable only for hundreds of nodes and the scalability limitation should be addressed in future work. Although we have shown here that the method does not have limitation in terms of distance between power plants, this finding should be confirmed in further research on a denser dataset containing a higher number of homogeneously spread nodes. Another research direction is improving the model's ability to capture cloud dynamics during the variable days, or in early mornings of the cloudy day. Thus, a framework that is capable of modelling cloud formation and cloud movement should improve PV power prediction accuracy during variable and cloudy days.

5 PING: Physics informed graph neural networks for forecasting solar resources

5.1 Introduction

Cloud formation and movement directly influence irradiance forecasting, a key aspect of solar photovoltaic (PV) power generation. Since cloud formation and cloud movement are guided by the advection-diffusion processes, we can conclude that PV power generation and advection-diffusion processes are closely linked. State-of-the-art PV power production forecasting models, discussed in the previous Chapters, fail to fully capture cloud movement during the variable days or mornings of cloudy days. The over-smooth forecasting signal suggests an incomplete representation of cloud dynamics within these models. Consequently, understanding the physical processes, specifically the advection-diffusion dynamics governing cloud movements, needs to be more utilised. Thus, a clear need for physics informed machine learning models, capable of modelling these advection-diffusion process on irregular grids, arises.

Advection-diffusion processes play a crucial role in understanding and predicting not only cloud formation and cloud movement but also other natural atmospheric phenomena, including ocean temperature distribution, air pollution spread, groundwater movement, the spread of forest fires, atmospheric temperature and many others. Advection-diffusion differential equations could describe these dynamical spatio-temporal processes. Hence, many scientists have used numerical methods to seek solutions for these challenges across different fields. NWPs, often used in PV forecasting, require solving physical equations using numerical methods. However, the numerical solvers are computationally expensive and require expert knowledge to fully describe the process (Sanchez-Gonzalez et al., 2020). Since the dynamics and factors which affect atmospheric phenomena are not always fully known and the PV power production is not solely dependent on advection-diffusion, machine learning models are often used to estimate the unknown part of the dynamics.

Machine learning models successfully predicted physical phenomena, such as weather (Lam

et al., 2023) and renewable power production, as shown in the previous chapters. However, these models represent black box models, which do not necessarily follow the advection-diffusion laws. Although they offer lower computation costs than the numerical methods, without the explicit physical constraints, they might yield implausible forecasts (Wang and Yu, 2021). There has been a growing need for incorporating known physical knowledge into machine learning models, not only in forecasting production from renewable sources and cloud movement but also in modelling other physical phenomena guided by advection-diffusion. Recently, much attention has been focused on physically informed neural networks (PINNs). PINNs integrate data-driven neural networks with prior scientific knowledge, ensuring consistency with the physical laws. They have succeeded in accelerating data simulations while offering solutions that obey physical laws. However, most PINN models focus on solving the tasks on regular grids, while PV forecasting represents the problem that inherently lies on an irregular grid.

Graph neural networks address the problem of simulating physical phenomena on irregular domains. Predominantly, they employ message-passing algorithms to learn the dynamics of different physical phenomena, (Pfaff et al., 2020). However, they suffer from quadratic complexity and over-smoothing when spatial resolution is high since many update steps are needed in order to pass the information. Although these issues were addressed with a multi-scale approach (Fortunato et al., 2022), including different scales of passing the information, assumed that a large amount of the information is available. What is more, the multi-level approach increased the needed computational memory. However, PV power production data usually has limited data available and already has issues with the over-smoothed predictions. Therefore, existing methods could not be utilized.

We introduce a physics-informed graph neural network (PING) model designed to capture cloud dynamics while offering accurate forecasts of PV power production. Here, we leverage the graph neural networks to simulate the underlying dynamics of an advection-diffusion process in order to forecast future production. The model estimates velocities of the historical input data in an unsupervised fashion. In order to make sure that the model can capture cloud dynamics, we have evaluated the proposed model on the cloud concentration index dataset on both regular and irregular domains. Furthermore, since PINNs have higher generalization capabilities, we have evaluated our model on both regular and irregular grids across different advection-diffusion datasets to demonstrate the generalization in modelling different physical phenomena. We address the modelling of the dynamics on a purely advective synthetic dataset, synthetic advection-diffusion-based datasets, sea surface temperature datasets as a highly diffusive process, cloud index as a highly advective process and PV power, which encompasses more complex phenomena than the previously mentioned datasets.

The main advantage of the proposed model is lower computational costs than the numerical methods or physically informed models that rely heavily on message passing. The proposed model is also characterized by its capacity to maintain same accuracy on both regular and irregular domains, even without extensive historical data, underscoring its efficiency in data

usage. Since the governing differential equation of the advection-diffusion processes is known in advance, we propose a novel discretization of this particular PDE on the irregular grid. Although we show that the same solution applies to problems that reside on the regular grid, our primary goal is to forecast data intrinsically defined on an irregular domain. Our objective is to forecast future concentration while estimating the historical velocities and uncovering the underlying dynamics in an unsupervised manner. The contributions of this chapter are the following:

- We introduce a physics-informed graph neural network (PING) model for forecasting the future particle concentration values in the advection-diffusion-based processes that reside on both regular and irregular grids. The information of advection-diffusion processes is added by including the PDE equation that satisfies the governing physical laws as a soft constraint in the loss function.
- The proposed model captures the dynamics of the processes and estimates velocities of the input data by introducing an Euler-based discretization scheme for irregular domains. Estimated velocities are used to improve the forecasting accuracy of future particle concentrations.
- A performance of the PING model is evaluated on multiple domains, including cloud concentration, sea surface temperature, irradiance and two different synthetic fluid-based datasets. That shows the generalization capabilities of the model in terms of modelling different physical phenomena.
- The model is also evaluated on PV power production, which is is more complex phenomena, since PV power production is not only influenced by advection-diffusion from cloud dynamics, but also from shading effects from local factors. Moreover, it is affected by temperature and atmospheric aerosols, as well as the orientation and tilt angle of PV module.
- Ablation study is conducted, and it indicates the robustness of the framework regardless of the stochastic subsampling over the irregular grid.

The rest of the chapter is organized as follows. Section 5.2 introduces preliminaries on advection-diffusion in fluid dynamics and graph time series forecasting. Section details the proposed PING architecture. The experimental results of our evaluation and the analyses are presented and discussed in Section . Finally, we conclude in Section .

5.1.1 Related work

Modelling dynamically changing spatial and temporal physical processes has been challenging in climate modelling, simulating fluid dynamics, and molecule interactions. Traditionally, scientists have tackled these problems by using prior knowledge of physical phenomena

and describing the known physical laws with differential equations. Different numerical solvers have been used to solve these equations. However, these solvers are computationally expensive and require expert knowledge in order to describe the process (Fortunato et al., 2022; Sanchez-Gonzalez et al., 2020). Thus, a substantial amount of effort is needed to choose physically meaningful parameters when modelling a simulator, and this process needs to be repeated for every task. Furthermore, the dynamics of many complex spatio-temporal processes are only partially known. Therefore, different machine learning techniques have been used to estimate the unknown part of the dynamics since they can offer generalization across various tasks.

Machine learning models have had considerable successes in forecasting weather (Bi et al., 2023; Lam et al., 2023), renewable power production (Simeunović et al., 2022b), epidemic forecasting (Wang et al., 2023), traffic forecasting (Khaled et al., 2022) and other complex dynamic spatio-temporal tasks. Recent works have shown that data-driven models are capable of learning the underlying dynamics without using expensive numerical solvers. Researchers have used neural networks as a black-box model for fluid dynamics forecasting by directly mapping the input sequence to the future predictions (Bi et al., 2023; Guo et al., 2022; Pathak et al., 2022; Xiao et al., 2019; Krivec et al., 2021). Machine learning models do not use traditional numerical solvers. Thus, they offer lower computation costs and have potential for generalization (Meng et al., 2022). However, purely data-driven models do not have explicit physical constraints. Thus, they might yield physically implausible forecasts that violate physical laws (Wang and Yu, 2021).

Therefore, the growing need for incorporating prior physics knowledge into machine learning models led to the development of physics-informed machine learning models for complex physical phenomena. PINNs yield scientifically valid models in terms of obeying physical laws and improve the generalizability of spatio-temporal machine learning model (Wang and Yu, 2021). Numerous PINN studies are focused on solving PDEs by approximating the solution of the differential equation with a neural network and bypassing the numerical solvers (Raissi et al., 2019b,a). However, when the governing differential equation is known apriori, the solution of the PDE is used as a soft constraint in the loss function of the neural network. Added constraint controls the trainable weights' effect and preserves the physics-based variables' semantics. In a scenario when only part of the dynamics is known, the neural networks are typically used to learn an error made by a physics-based model (Takeishi and Kalousis, 2021; Yin et al., 2021; Belbute-Peres et al., 2020). Thus, they can correct the bias from the physics-based part of the models. However, hybrid models still need to solve the known part of the physics-based equation numerically. When the governing equation is a priori completely unknown, the researchers either try to learn the dynamics and then predict the model's output (De Bézenac et al., 2019; Kashinath et al., 2021). In these works, the numerical solutions are bypassed, and future values are predicted directly using neural networks. However, most PINNs focus on solving the tasks that reside on a regular grid. On top of this, they typically employ automatic differentiation, which can fail with insufficient collocation points, making them both computationally demanding and unsuitable for tasks when a lower number of

points is available (Chiu et al., 2022; He et al., 2023).

In many tasks, graph neural networks (GNNs) had many successes when modelling processes on irregular grid. Recently, GNNs have also shown promising approaches when simulating different physical phenomena, including flow dynamics on meshes. Different types of particle interaction, including one in the fluids, have been modelled in the work of Sanchez-Gonzalez et al. (2020); Pfaff et al. (2020). They have used message-passing graph neural networks to learn a simulation model of the physical phenomenon. However, these networks suffer from quadratic complexity due to dependence on the number of nodes. On top of this, meshes with fine resolution might suffer from over-smoothing since a high number of update steps is needed to pass the same amount of information. These issues were alleviated in the work of Fortunato et al. (2022) by the introduction of a multi-scale approach on two different resolutions. A multi-scale approach was also used in the works of Lam et al. (2023), where six different projections of meshes contributed to accurate medium-range weather forecast. Weather forecast for high resolution dense and sparse grids is proposed in works of Andrychowicz et al. (2023). However, both weather-forecasting architectures are highly expensive computationally and in terms of memory. Furthermore, several studies, including the works of Lam et al. (2023); Fortunato et al. (2022); Pfaff et al. (2020); Sanchez-Gonzalez et al. (2020) which focus on either fluid dynamics simulation or weather forecasting problems, assume that a large amount of data is available, making it difficult to use in the real-world scenarios, where historical information is available for rather a short period.

In the real world, most of the phenomena of interest, such as renewable power generation or weather data, are measured through a network of sensors irregularly distributed in space., yielding problems that inherently lie in irregular domains. In the works of Gao et al. (2022) the Garlekin method is introduced as a discretization method for problems that reside on irregular domains. However, the main aim of the proposed framework is solving forward and inverse PDEs in a unified manner, not a future forecast. Furthermore, it is focused on steady-state PDEs and, thus, requires additional research in order to handle the spatio-temporal PDEs.

5.2 Problem formulation

5.2.1 Time series forecasting on Graphs

Forecasting future quantities of the observed particles in the advection-diffusion processes can be posed as a time series prediction task, where the goal is to forecast the concentration of the quantities $\mathbf{C}(t)$ at time t for the next H discrete time steps ahead, where $\mathbf{C}(t)$ is the concentration at time t measured on a set of N locations in space. For example, we can predict the future sea surface temperature, the cloud concentration index or PV power production given the past data of the same type. The problem of forecasting the next H discrete time steps given M past observations can be formulated:

$$\hat{\mathbf{C}}(t),\ldots,\hat{\mathbf{C}}(t+H-1) = f_\beta\left(\mathbf{C}(t-M),\ldots,\mathbf{C}(t-1)\right), \tag{5.1}$$

where for any t, f_β is a chosen family of parametric estimators. A set of parameters β is learnt such that it minimizes the prediction error over the entire horizon by solving the following problem:

$$\arg\min_{\beta} \sum_{\tau\in\mathscr{T}} \sum_{\xi=\tau}^{\tau+H-1} \|\hat{\mathbf{C}}(\xi) - \mathbf{C}(\xi)\|_2^2, \tag{5.2}$$

where $\mathscr{T}$ represents the times of historical time steps which are used for fitting the model during training. We use mean square error (MSE), denoted with $\|\cdot\|_2^2$ as a loss function. MSE function, often used in time series forecasting, is chosen to ensure that predictions of the proposed model do not deviate significantly from ground truth.

In the previous chapters we have shown that the initial intuition of inferring part of cloud dynamics, using only the past production data is correct. The past data was used to find the spatial and temporal correlations which demonstrated to be meaningful for accurate prediction of future production. Here we follow the similar intuition and leverage different types of the input data in order to estimate the spatio-temporal correlations and the dynamics, caused by advection-diffusion. Power production at each PV station or concentration value of weather data, including cloud index density and sea surface temperature, are modelled as signals on a spatio-temporal graph G . We can model PV stations and measurement locations of weather data as nodes, represented as a set $v = \{v_1, v_2, \ldots, v_N\}$. In addition, we model velocity features of the underlying flows, which are guiding the advection-diffusion processes, as a graph signal. The edges εreflects the correlation between the concentration and velocity features in the advection-diffusion processes. We define a graph signal as a mapping $\mathbf{C} : v \to \mathbb{R}$, such that $\mathbf{C}_v^{\mathbf{t}} \in \mathbb{R}$ is the concentration value at node v at time t. Multiplying the graph signal $\mathbf{C}$ with the Laplacian $\mathscr{L}$ yields:

$$\mathscr{L}\mathbf{C} = \sum_{j\in\mathscr{N}_i} \mathscr{L}_{ij} C_j = \sum_{j\in\mathscr{N}_i} w_{ij}(\mathbf{C}_i - \mathbf{C}_j) \tag{5.3}$$

where w_{ij} represents the weight on the edge between nodes v_i and v_j. An important property of Graph Laplacian matrix is that it is positive, semi-definite matrix and its quadratic form is explicitly given by:

$$\mathbf{C}^T \mathscr{L}\mathbf{C} = \sum_i \sum_{j\in\mathscr{N}_i} w_{ij}(\mathbf{C}_i - \mathbf{C}_j)^2. \tag{5.4}$$

For more details and an in-depth review of GSP we refer the reader to the work of Ortega et al. (2018). In our case, the graph signal at time step t is denoted by $\mathbf{C}(t)$ and it represents the vector of measured quantities such as temperature, cloud coverage index and concentration of simulated quantity or PV power production.

5.2.2 Advection-diffusion processes

Advection-diffusion processes describe how quantities travel and spread in various mediums, especially fluid dynamics. Advection and diffusion are two key processes that govern the transport of energy and matter in fluids, described by the following differential equation:

$$\frac{\partial C}{\partial t} + \nabla \cdot (\mathbf{u}C) = \nabla \cdot (D\nabla C) + P, \tag{5.5}$$

where C represents the concentration of particles and P represents sinks and sources of particles. Velocity of the particles is defined as the vector field $\mathbf{u}$, D is a scalar diffusion coefficient and $\nabla\cdot$ is the divergence operator (Stocker, 2011), defined in 2D Cartesian coordinates:

$$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right) \tag{5.6}$$

where x and y are the coordinates.

In the fluid dynamics the advection is driven by the bulk motion of fluid, transporting particles with concentration $C(x, y, t)$, which move with the velocity of fluid $\mathbf{u}(x, y, t)$, over a specific period of time. The advection represents the left part of Equation 5.5, and it is denoted with the sum of proxies α, which represents the change of the concentration in the space and β, the change of the concentration in time, such that:

$$\alpha + \beta = \frac{\partial C}{\partial t} + \nabla \cdot (\mathbf{u}C). \tag{5.7}$$

The diffusion, on the other hand, is caused by the movement of particles from area of higher concentration to the area of lower concentration (Stocker, 2011). The diffusion describes how the concentration of the substance changes over time due to a concentration gradient. Therefore, the diffusion represents the first term in the right hand side of Equation 5.5 and it is denoted with γ:

$$\gamma = \nabla \cdot (D\nabla C), \tag{5.8}$$

where D represents the diffusion constant that depends on physical properties of diffusing particles and the compound containing these particles.

Thus, the combined effects of advection and diffusion play a key role in forecasting advection-diffusion processes. For example, they guide the motion and distribution of heat within the ocean's surface layer in predicting sea surface temperature. For forecasting the cloud movement and cloud concentration index, advection and diffusion play pivotal roles in the motion and dispersion of clouds. Eventually, the motion of the air in the atmosphere, which causes the clouds' movement, inadvertently affects irradiance and PV power production forecasting.

5.2.3 Discretization of the advection-diffusion equation on an irregular grid

In order to address the issue of forecasting on the irregular grid, we have to discretize the solution of the advection-diffusion equation on the irregular grid. This entails discretizing the change of concentration in space and time, and the diffusion. The concentration change in the time is the same as defined easily for every node v_i as $C_i^t - C_i^{t-1}$. The diffusion term is defined as a second-order derivative and it corresponds to the product between the Graph Laplacian and the graph signal. The only term left to be defined is the discretization of the spatial concentration change due to the advection.

The velocities $\mathbf{u}_i^t$ at node v_i at time t are defined in two-dimensional feature space. In order to calculate the vector flow ϑ_{ij} at the edge ε_{ij}, we define the projection of the velocity vector $\mathbf{u}_i$ onto the edge direction between the observed node v_i and its neighbour v_j as:

$$\vartheta_{ij} = \mathbf{u}_i \cdot \mathbf{e}_{ij}. \tag{5.9}$$

where $\mathbf{e_{ij}} = \frac{\Delta \mathbf{z}_{ij}}{||\Delta \mathbf{z}_{ij}||_2^2}$ is the unitary vector of the direction. Therefore the spatial concentration change at node v_i is defined as:

$$\nabla \cdot (\mathbf{uC})_i^t = \sum_{j \in \mathcal{N}_i} \vartheta_{ij} \sqrt{w_{ij}} \nabla C_{ij} \tag{5.10}$$

where $\nabla C_{ij}^t = C_j^t - C_i^t$ represents the difference in the concentration change and w_{ij} represents the edge weight. The RBF kernel chosen to calculate the edge weights provides the similarity measure based on the physical distances between the nodes, which are important when modelling local phenomena. Moreover, the RBF kernel has been successfully used for prediction future sea surface temperature, in the work of De Bézenac et al. (2019). Finally, we define the general advection diffusion equation for an irregular grid:

$$C_i^{t+1} - C_i^t = \sum_{j \in \mathcal{N}_i} \mathbf{u}_i^t \cdot \mathbf{e}_{ij} \sqrt{w_{ij}} (C_i^t - C_j^t) + \mathscr{L}_i \mathbf{C^t}, \tag{5.11}$$

which could be used also on a regular grid.

On top of this, we need to define the divergence of the velocity field on the graph. Similarly, as in previous the previous case, the velocity vectors need to be projected from two-dimensional Cartesian space onto the direction of edge between observed node its neighbour, in order to obtain the flow edge value. This leads us to define the divergence of velocity field on irregular grid between the node v_i and its neighbour v_j:

$$\nabla \cdot \hat{\mathbf{u}}_i^t = \sum_{j \in \mathcal{N}_i} \sqrt{w_{ij}} \left((\mathbf{u}_i^t - \mathbf{u}_j^t) \cdot \mathbf{e_{ij}} \right), \tag{5.12}$$

where the divergence is calculated for every neighbour.

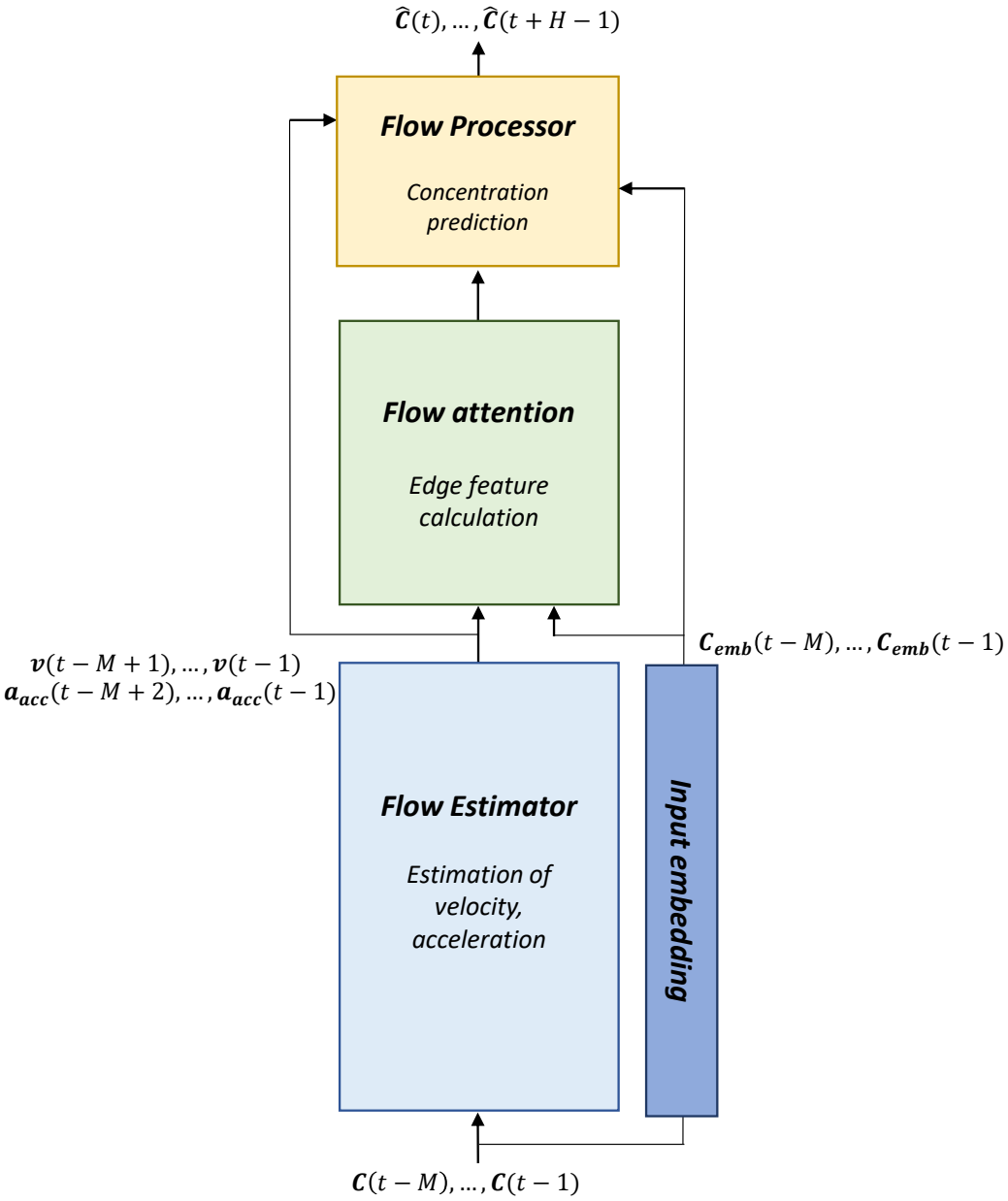


Figure 5.1: PING model.

5.3 Physically-informed graph neural network

In this chapter we propose a sequence-to-sequence model built on a physically-informed graph neural network (PING) for advection-diffusion processes, more precisely for cloud concentration index, sea surface temperature and PV power forecasting tasks. The model represents physics-guided solution for the prediction of particle concentration of the advection-diffusion processes, in 2-dimensional space. The overview of the model is shown in Figure 5.1. Our model leverages knowledge of physical processes, added as a soft constraint to the model,

to guide graph neural networks in order to estimate the flows and predict the concentration of the particles. Subsequently, these flow estimations are utilized within the flow estimation block to generate the velocity and acceleration features, which are useful for the forecast of cloud concentration index, sea surface temperature or PV power production. A sequence of the past M measurements of particle concentrations $\mathbf{C} = [\mathbf{C}(t-M), \ldots, \mathbf{C}(t-1)] \in \mathbb{R}^{N \times M}$ over N sensor network nodes is taken as an input to the model, when predicting concentration H steps ahead $\hat{\mathbf{C}} = \left[\hat{\mathbf{C}}(t), \ldots, \hat{\mathbf{C}}(t+H-1)\right] \in \mathbb{R}^{N \times H}$ over same set of nodes. PING consists of the three blocks: flow estimator, flow attention and flow processor block.

5.3.1 Flow Estimator

We have developed two estimation bocks. In the first estimation block we estimate the velocity features of the the input concentration. More precisely, the input to this estimation block are geographical coordinates and the sequences of past observations. In the second estimation block the obtained velocity features are used as an input to estimate the acceleration features. Both velocity and acceleration are later used as inputs to the flow attention block, as shown in Figure 5.1.

The velocity could be obtained as the solution from a partial differential equation (PDE), given in Equation 5.5. In order to avoid the high computational complexity coming from the classical numerical methods for solving PDEs, graph-based architectures are successfully for prediction of fluid dynamics and other systems modelled by partial differential equations (Boussif et al., 2022). Given a spatial query, spatial embedding and the input features, it is possible to accurately forecast future values at any point. Following a similar path as (Boussif et al., 2022), we propose a model that implicitly solves the subset of PDEs that characterize advection-diffusion processes within a graph-based setting. We assume incompressibility of the flow, thereby eliminating the need to consider sources or sinks, P from Equation 5.5. We can rewrite this equation:

$$\mathbf{u} \cdot \nabla C + \frac{\partial C}{\partial t} - D\nabla^2 C = 0. \tag{5.13}$$

such that the first and the second term in Equation 5.13 denotes the contribution of the advection and the third term represents the diffusion. The advection entails the concentration change in the space $\mathbf{u} \cdot \nabla C$ and the concentration change in time $\frac{\partial C}{\partial t}$.

The spatial change of concentration could be modelled numerically by discretizing Equation 5.13 in space and time. We adopt the following notation: the particle concentration at node v_i at time step t is given by C_i^t. The change of concentration C_i in the space at node v_i, denoted with $C_{\delta,i}^t$, in the 2-dimensional case, shown on Figure 5.2 could be approximated as:

$$C_{\delta,i}^t = \frac{C_{i+1,k}^t - C_{i-1,k}^t}{\Delta x} + \frac{C_{i,k+1}^t - C_{i,k-1}^t}{\Delta y}, \tag{5.14}$$

where Δx and Δy are the distances between every two neighbouring nodes on x and y axis.

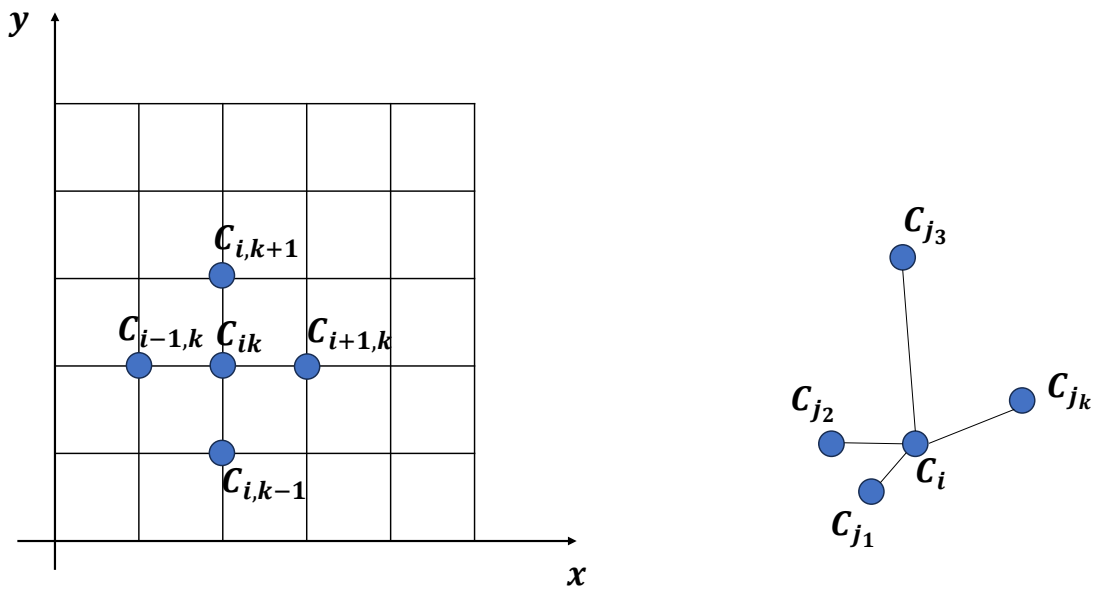


Figure 5.2: Illustration of a 2-dimensional case on the left, where concentration C_i at the node v_i is equidistant from the adjacent nodes $v_{i-1,j}$ and $v_{i+1,j}$ on the x-axis and nodes $v_{i,j-1}$ and $v_{i,j+1}$ on y-axis. On the right concentration at the node C_i has different distances to its neighbour C_j.

Using the Euler discretization, we take into account the nodes which are adjacent to the observed node and on regular grid they are equidistant from it. Nodes v_{i+1} and v_{i-1} are adjacent, equidistant nodes to the observed one, v_i. Although this is easily defined in 2-dimensional regular grid, the definition of the temporal and spatial change in the concentration is more difficult for the irregular grid, see Figure 5.2, where our task inherently lies. The concentration at node v_i is C_i and this node is not equidistant from its neighbours C_j for every node $j \in \mathcal{N}$ in neighbourhood of v_i.

The challenge of modelling the change of concentration on the irregular domain is not the only challenge that we address. We also need to define the projection of vector $\mathbf{u}$ on the gradient of quantity ∇C on irregular domain, in which case $\alpha = \mathbf{u} \cdot \nabla C$ has infinitely many solutions. Thus, to model the change in the space on the irregular graph, we must first define adjacent nodes on the graph. We follow the intuition that the same flow affects the spatially close particles. The adjacent nodes are defined using the closest neighbours on the graph, and then the spatial concentration change across the neighbours is aggregated. The governing assumption is that features (concentrations) from node v_j are relevant for the flow estimation at node v_i, if the node v_j lies in the neighbourhood of the node v_i. The spatial concentration change between

the neighbours is denoted by $\nabla C_i^t = C_i^t - C_j^t$. However, we need to provide the model with information regarding the distance and orientation between the nodes since it is needed with the spatial change between the neighbours for the spatial discretization. The addition of the distance information is possible by calculating the difference in the position between nodes $\Delta \mathbf{z_{ij}} = \left[long_i - long_j, lat_i - lat_j \right]$, where $\mathbf{z}_i = \left[long_i, lat_i \right]$ represent longitude and latitude of the node v_i. We only discuss the latitude and longitude, since all problems that we consider are on the 2-dimensional space. Thus, the proxy for spatial concentration change between the nodes v_i and v_j is defined via concatenation, denoted with $[\cdot||\cdot]$:

$$\boldsymbol{\alpha}_{ij}^t = \left[\left(C_i^t - C_j^t \right) || \Delta \mathbf{z_{ij}} \right]. \tag{5.15}$$

On the other hand, the temporal change in concentration for the node v_i is defined more easily:

$$\beta_i^t = C_i^t - C_i^{t-1}. \tag{5.16}$$

The diffusion is described by the rate of change of concentration, which is the second-order derivative. Thus, we model the discretization of the second order derivative wit the Graph Laplacian $\gamma_i = \mathscr{L}_i \mathbf{C}^t$, (Hein et al., 2007). Although it is possible to directly calculate the temporal change and the diffusion for the observed node v_i (from the second and the third term in Equation 5.13), the spatial change of concentration is difficult to define on the irregular grid, since it requires discretization of the PDE on the irregular grid. Therefore, we consider that the difference in the concentrations between the node v_i and its closest neighbours v_j is a partial contribution to the change of concentration at node v_i. The difference in concentration between every pair of nodes v_i and its neighbour v_j at time t is defined with $\boldsymbol{h}_{ij}^t \in \mathbb{R}^{F_{in}}$:

$$\mathbf{h}_{ij}^t = \left[\Delta \mathbf{z_{ij}} || (C_i^t - C_i^{t-1}) || (C_i^t - C_j^t) || \mathscr{L}_i \mathbf{C}^t \right], \tag{5.17}$$

where we take into consideration the temporal change β_i^t, spatial change across the neighbours and their mutual distances α_{ij}^t and a diffusion term $\mathscr{L}_i \mathbf{C}^t$.

Once the partial contributions($\boldsymbol{h}_{ij}^t$) from the closest neighbours are computed, the aggregation of the neighbourhood information is needed. This is calculated with a weighting coefficient matrix $\mathbf{W}_\tau \in \mathbb{R}^{N \times S}$, where the total number of nodes is N and $S = |\mathscr{N}_i|, \forall v_i$ is the predefined number of the closest neighbours for all nodes. Thus, we want to learn the contribution that each neighbouring node has towards the velocity at the observed node by learning the adjacency matrix $\mathbf{W}^\tau$ where τ is obtained using the floor function, $\lfloor \cdot \rfloor$ which rounds down the division to the nearest integer. The value of $\tau = \lfloor \frac{t-1}{2} \rfloor$ is chosen such that the weight matrix $\mathbf{W}^\tau$ is shared for every two consecutive steps, in order to be able to capture fast changes between time steps. The matrix $\mathbf{W}^\tau$ models the individual contribution from each neighbouring node towards the velocity at the observed node:

$$\hat{\mathbf{h}}_i^t = \sigma \left(\sum_{j \in \mathscr{N}_i} w_{ij}^\tau f(\boldsymbol{h}_{ij}^t) \right) \tag{5.18}$$

where w_{ij}^{τ} represents an entry of the matrix $\mathbf{W}^{\tau}$ and function $f(\cdot)$ represents an MLP. Instead of directly solving PDE, the authors of Raissi et al. (2019a) used MLP to approximate the solution to PDE equation. Following the same idea, we use an MLP to approximate the flow values. Thus, we obtain the aggregated vector of neighbourhood information $\hat{\mathbf{h}}_i^t$ is of size $\mathbb{R}^{F_{in}}$. Then, we can finally define the velocity value $\mathbf{u}_i^t \in \mathbb{R}^{F'}$ in the physics-guided mechanism for advection-diffusion processes as:

$$\mathbf{u}_i^t = \mathbf{A}^T \hat{\mathbf{h}}_i^t, \tag{5.19}$$

where T represents the past time steps for which we are estimating the velocity, F' is the size of projected vector space of the velocity, that corresponds to the Cartesian coordinates in our setting ($F' = 2$) and $\mathbf{A} \in \mathbb{R}^{F_{in} \times F'}$ is learnable parameter that projects features from the space size F_{in} to the on of size F'.

The obtained velocity values are in the 2-dimensional space, thus, in order to increase the expressive power of the model, we project the velocity estimations to higher feature space, using an MLP:

$$\hat{\mathbf{u}}_i^t = \mathbf{W}_3^e \sigma(\mathbf{W}_2^e \sigma(\mathbf{W}_1^e \mathbf{v}_i^t + \mathbf{b}_1^e) + \mathbf{b}_2^e) + \mathbf{b}_3^e \tag{5.20}$$

where $\mathbf{W}_i^e$ and $\mathbf{b}_i^e$, $e \in [1,2,3]$ are learnable weights and biases, and feature vector $\hat{\mathbf{u}}_i^t \in \mathbb{R}^{F''}$. At the end of each layer, the non-linearity σ is applied, which represents the LeakyRelu function.

In order to diffuse information across the sequences, we have performed a 1D temporal convolution on the sequences. The convolution operation is followed by the rolling mean and features aggregation via sliding window, obtaining the final velocity-like features tensor $\mathbf{v}_i^t \in \mathbb{R}^{F_{out}}$. Although we choose mean-aggregation function, a max-aggregation or GCN-like operation could be envisaged instead. We have chosen empirically a mean-aggregation on the time-axis to aggregate feature information from different directions of the irregular grid.

In order to increase the expressiveness of the model, we will take the velocity features from the output of the estimation block $\mathbf{v}_i^t$ as the input to the next estimation block to obtain acceleration features $\mathbf{a}_{acc}^t \in \mathbb{R}^{N \times F_{out}^{acc}}$, as shown in Figure 5.3. We named the computed features $\mathbf{a}_{acc}^t$ which represent the output of the attention flow block, acceleration features, since they use the difference operator on the velocity features. In addition, in Figure 5.3 is shown that the horizon length of estimated velocity features is $M-1$. The size differs from the length of input data M, due to difference operation in 5.18. The same reasoning is explanation of the acceleration horizon length being $M-2$.

The second difference between velocity and acceleration flow estimation blocks, is regarding the choice of weight matrix W^{τ}. In the velocity estimation block τ is shared for every two consecutive time steps of the input horizon. However, in the acceleration flow estimation block $\tau = 1$ and we learn a single matrix W^{τ} across all sequences. Since the number of features F_{in}^{acc} and neighbours S^{acc} used for information aggregation in Equation 5.17 in the acceleration estimation block is significantly higher compared to number of features F_{in} and neighbours S in the velocity estimation block, single weight matrix is learnt to avoid increasing complexity

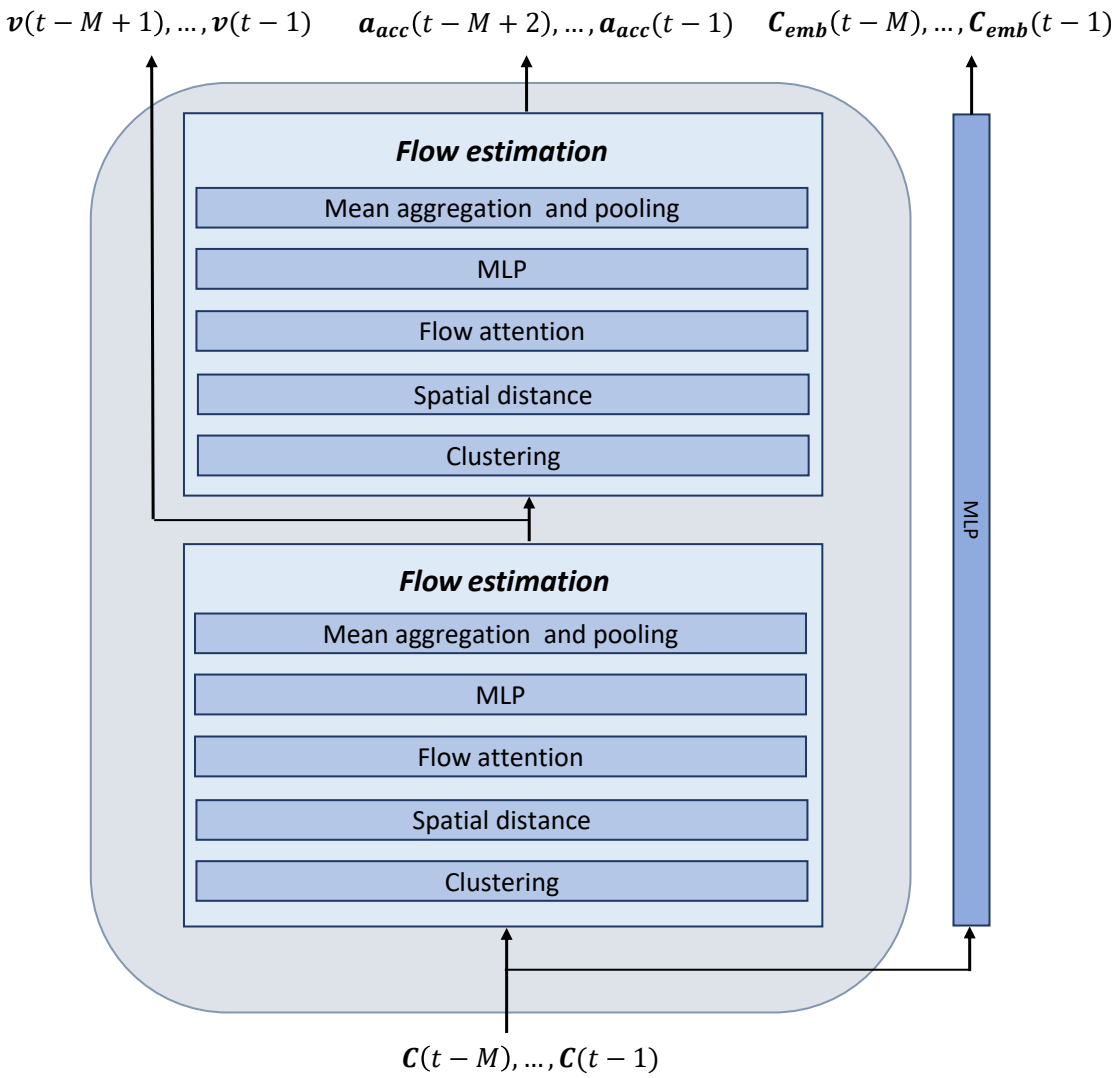


Figure 5.3: The flow estimator block of the PING model.

of the model.

Since both velocities and acceleration features carry information on the neighbour's contribution for the final flow prediction at node v_i, we also embed the concentration input signal at every node through a single MLP in order to carry the original signal value when calculating the future forecast at node v_i. This is calculated on the entire input horizon with the length M.

5.3.2 Flow Attention

The flow attention takes as inputs: estimated velocity is a set of node's velocity feature estimations $\mathbf{v}^t = [\mathbf{v}_1^t, ..., \mathbf{v}_N^t]$ where N is the number of nodes; and acceleration features $\mathbf{a}_{acc}^t$, calculated in the flow estimation block. Additionally, it uses as an input the embedded con-

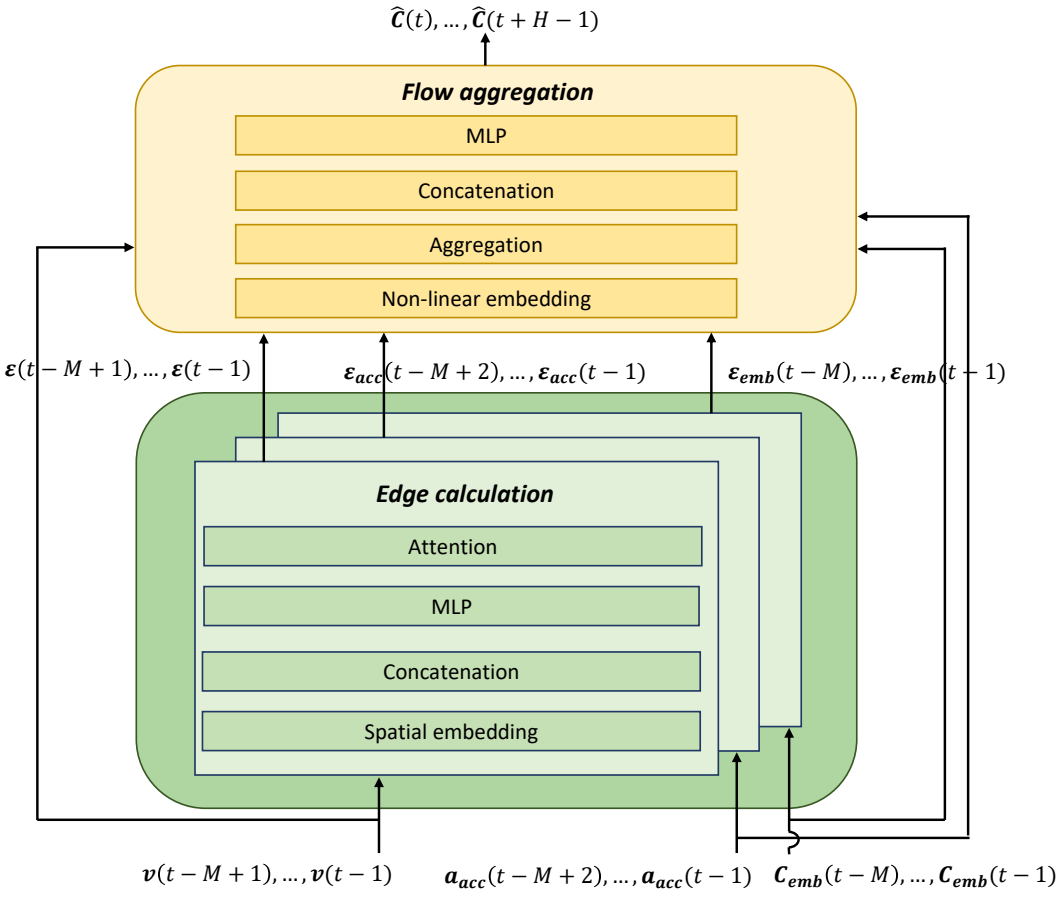


Figure 5.4: The flow attention and flow processor blocks of the PING model.

centration values $\mathbf{C}^t_{emb}$. The number of attention blocks is three since each data type, the embedded concentrations, velocity, and acceleration features, are passed to a different attention flow block, see Figure 5.4. These blocks aim to calculate the contribution of each neighbour's concentration, velocity and acceleration towards the final prediction, which represents the edge values.

The mentioned concentration embeddings are calculated via an MLP in order to transform the input values into higher dimension features, allowing the model to potentially leverage the higher-level patterns that are not evident in the raw data. The methodology for deriving the velocity and acceleration features used as inputs has been described in the previous subsection. The predictive power of this block is derived from its ability to quantify the influence of neighbouring values on the prediction of the final node for each of these inputs. However, we will focus on the single predictor block to explain the predictor, which takes the velocity features as inputs.

First, in the processor block, we construct topological embedding $q(z_i)$ at every node v_i where $q(\cdot)$ represents the multi-layer perceptron function. The MLP is used to transform

the coordinates into the different space, since directly feeding the coordinates as features might not be enough to capture complex spatial relationship within the data. This function maps the coordinates at node v_i, $\mathbf{z}_i$, to potentially different dimensional space and obtains the new values $\tilde{\mathbf{z}}_i = \sigma(\mathbf{z}_i \mathbf{W}^{emb})$. The learnable matrix of positional encoding is denoted by $\mathbf{W}^{emb} \in \mathbb{R}^{2 \times s^{emb}}$, where s^{emb} represents the size of the positional encoding. The flow vectors (velocity and acceleration features) constructed in the previous section, as well as embedded input concentrations are concatenated with the spatial embeddings in order to find the contribution of each vector towards final concentration prediction.

We will focus on the velocity features further in this subsection, although the embedded concentration values and acceleration features are utilized in the same manner. Hence, we concatenate the neighbouring velocity features at node v_i and the difference between spatial embeddings $\Delta\tilde{\mathbf{z}} = \tilde{\mathbf{z}}_i - \tilde{\mathbf{z}}_j$. Additionally, we take into account neighbouring velocities $\forall\, j \in \mathcal{N}_i$, which are denoted with $\hat{\mathbf{v}}^t_{ij}$. The neighbouring velocity features at v_j are concatenated for each node v_i as $\hat{\mathbf{v}}^t_{ij} = \|_{\forall\, j \in \mathcal{N}_i} \left(\mathbf{v}^t_j\right)$, where symbol $\|_{\forall\, j \in \mathcal{N}_i}(\cdot)$ represent the concatenation for all the nodes v_j in the neighbourhood of v_i. The main idea is to take into account velocities with respect to the difference of their topological embeddings:

$$\boldsymbol{g}^t_{ij} = \left[\hat{\mathbf{v}}^t_{ij} || \Delta\tilde{\mathbf{z}}\right], \tag{5.21}$$

where $\mathbf{g}^t_{ij} \in \mathbb{R}^{F_{out} + s^{emb}}$. This particular signal is then subjected to a transformation via an MLP in order to model non-linear relationships between the velocity features and the topology of the graph:

$$\hat{\mathbf{g}}^{\mathbf{t}}_{\mathbf{ij}} = \mathbf{W}^d_3 \sigma(\mathbf{W}^d_2 \sigma(\mathbf{W}^d_1 \mathbf{g}^t_{ij} + \mathbf{b}^d_1) + \mathbf{b}^d_2) + \mathbf{b}^d_3 \tag{5.22}$$

such that $\mathbf{W}^d_i$ and $\mathbf{b}^d_i$, $i \in [1,2,3]$ are MLP learnable weights and biases. This transforms the velocities to a higher-dimension space, allowing the model to represent the location-specific behaviour more effectively. The output value from MLP is of the size $\hat{\mathbf{g}}^{\mathbf{t}}_{\mathbf{ij}} \in \mathbb{R}^{F^d}$, where F^d is the number of output features from the MLP.

Finally, we perform the feature projection operation to reduce the feature size before passing them to a final prediction block. The projection is defined as: $p^t_{ij} = \mathbf{q}^T_d \hat{\mathbf{g}}^{\mathbf{t}}_{\mathbf{ij}} + b^d$, where $\mathbf{q}_d \in \mathbb{R}^{F^d}$. Then the contribution from each neighbour towards the final flow features of each node is calculated via a softmax function as:

$$\varepsilon^t_{ij} = \frac{\exp\left(b^{inv} p^t_{ij}\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(b^{inv} p_{ik}\right)} \tag{5.23}$$

where scalar value b^{inv} could be viewed as a learnable inverse temperature, the coefficients ε^t_{ij} between the nodes v_i and v_j could be viewed as entries in an attention matrix $\boldsymbol{\varepsilon}^{\boldsymbol{t}} \in \mathbb{R}^{N \times S}$, and they represent the interpretable edge values. The attention coefficient value is derived as a quantification of the attention paid to each neighbour when making the prediction.

As shown on Figure 5.4 in the flow attention block we obtain the attention embedded-

concentration matrix $\boldsymbol{\varepsilon}^t_{emb} \in \mathbb{R}^{N \times S^{emb}}$ and the attention acceleration matrix $\boldsymbol{\varepsilon}^t_{acc} \in \mathbb{R}^{N \times S^{acc}}$, where $S^{emb} = S = |\mathcal{N}_i|, \forall v_i$ is the predefined number of the closest neighbours for calculating velocity and concentration embeddings and $S^{acc} = |\mathcal{N}^{acc}_i|, \forall v_i$ is predefined number of the closest neighbours for acceleration estimation. The neighbourhood size for each of these blocks is chosen empirically. We obtain attention matrices of embedded concentrations and accelerations using the same operations, defined for obtaining the attention velocity matrix $\boldsymbol{\varepsilon}^t$.

5.3.3 Flow Processor

The flow processor block processes the estimated feature vectors and edge values to predict future concentration values. The input to the flow processor block are embedded concentration values $\mathbf{C}^t_{emb}$, velocity features $\mathbf{v}^t$ and acceleration features $\mathbf{a}^t_{acc}$, as well as the attention values of the embedded concentrations, velocity and acceleration features. The output from the flow processor block is the future concentration prediction.

In the processing block, we first transform the feature space of its inputs, embedded concentrations, velocity and acceleration features. We will focus on the velocity inputs in the description, although the same operations are repeated for all the input types. We use the weight matrix $\mathbf{W}_{agg}$ to project the final features in each prediction block to the to the same latent feature space of dimension $l = Hq$, where H is the number of the forecasting steps of the model. The weight $\mathbf{W}_{agg} \in \mathbb{R}^{F'' \times l}$ and biases $\mathbf{b}_{agg} \in \mathbb{R}^l$ are learnable parameters. Finally, the flow features are transformed in:

$$\mathbf{o}^t_{ij} = \mathbf{W}^T_{agg} \hat{\mathbf{v}}^t_{ij}, \tag{5.24}$$

where $\mathbf{o}^t_{ij} \in \mathbb{R}^{Hq}$.

Once we obtain the projection of flow features and spatial embedding, we aggregate the information from the neighbours via the weighting function, such that the edge values are used as weights or attention coefficients:

$$\hat{\mathbf{y}}_i = \sum_t \sum_j \sigma(\mathbf{o}^t_{ij}) \varepsilon^t_{ij} \tag{5.25}$$

where $\hat{\mathbf{y}}_i \in \mathbb{R}^{Hq}$ and σ is a non-linearity and ε^t_{ij} are edge weights computed in the attention block in Equation 5.23. We reshape the vector $\hat{\mathbf{y}}_i \in \mathbb{R}^{Hq}$ to matrix of size $\hat{\mathbf{y}}_i \in \mathbb{R}^{H \times q}$

As already mentioned, we consider the flow feature values from the embedded input and acceleration contain meaningful patterns for a concentration prediction. Hence, these features are treated with the same operations and then aggregated in the flow processor block with the velocity-like features in Equation 5.25. The final predictions of the concentration are obtained with MLP f^{fin}, whose inputs are the embedded concentrations, velocity and acceleration

features, denoted with $\hat{\mathbf{y}}_i^{in}, \hat{\mathbf{y}}_i^{vel}, \hat{\mathbf{y}}_i^{acc}$, respectively:

$$\hat{\mathbf{C}} = f^{fin}\left[\hat{\mathbf{y}}_i^{in} || \hat{\mathbf{y}}_i^{vel} || \hat{\mathbf{y}}_i^{acc}\right]. \tag{5.26}$$

In our case, the MLP f^{fin} transforms the concatenated features from all three blocks from the size of $3q$ to a single feature. We have proposed a single MLP to project node's features to the size of future horizon, at the end of the flow processor block. This block does not have capability of propagating future dynamics for long sequences. A recurrent structure, e.g. a recurrent neural network, can be used instead of the MLP to model the future dynamics in case of long forecasting sequences.

5.3.4 Physics-guided optimisation function

The model uses MSE as a loss function to minimise the prediction error over the entire horizon, as shown in the Problem formulation subsection 5.2.2. As the proposed model is a physically-informed data-driven model, it is possible to benefit from the physical knowledge about the process by including PDE that describes advection-diffusion in the loss function. Therefore, we have changed the objective function from Equation 5.2 to guide the model to follow the advection-diffusion by including Equation 5.13 in the loss function. Our goal is to incentivise the model to respect the physical laws without explicitly using past velocity, since they are usually not available in real-world datasets. We will use estimated velocity values obtained in Equation 5.22. Our objective function L becomes:

$$\begin{aligned} L(\theta) = &\sum_{\tau \in \mathscr{T}} \sum_{\xi=\tau}^{\tau+H-1} \|\hat{\mathbf{C}}(\xi) - \mathbf{C}(\xi)\|_2^2 \\ &+ \lambda \sum_{\tau \in \mathscr{T}} \sum_{\xi=\tau-M+1}^{\tau-1} \| - \frac{\partial \mathbf{C}(\xi)}{\partial t} - \nabla \cdot (\mathbf{u}(\xi)\mathbf{C}(\xi)) + \nabla \cdot (D \nabla \mathbf{C}(\xi))\|_2^2, \end{aligned} \tag{5.27}$$

where $\mathscr{T}$ represents the times of historical time steps which are used for fitting the model during training. In the objective function $L(\theta)$, the concentration values $\mathbf{C}(\theta)$ and velocities $\mathbf{u}(\theta)$ are functions of learnable parameter θ, however, for the simplicity and ease of notation, we denote them as $\mathbf{u}, \mathbf{C}$ in Equation 5.27. The first term represents the mean square error between the ground truth and estimated concentration, and term represents the advection-diffusion equation.

However, in the Equation 5.27 there are no constraints on the divergence of the vector field, more precisely the spatial smoothness across the estimated velocities is not imposed explicitly. In the physical phenomena, the sudden changes of the direction are rare, due to the diffusion and the conservation of the mass in the incompressible fluids. The advection represents the transport of the concentration, which is inherently spatially smooth due to continuos nature of the fluid without sudden external force. The diffusion term smoothens out the abrupt

concentration change, whereas the initial assumption that the observed advection-diffusion processes are incompressible fluids, prevents sudden spikes in the velocity field between neighbouring spatial points. Therefore, we have added another term in our regularization function, such that the estimated physical velocity field is spatially smooth:

$$
\begin{aligned}
L(\theta) = & \sum_{\tau \in \mathcal{T}} \sum_{\xi=\tau}^{\tau+H-1} \|\hat{\mathbf{C}}(\xi) - \mathbf{C}(\xi)\|_2^2 \\
& + \lambda_1 \sum_{\tau \in \mathcal{T}} \sum_{\xi=\tau-M+1}^{\tau-1} \| - \frac{\partial \mathbf{C}(\xi)}{\partial t} - \nabla \cdot (\mathbf{u}(\xi)\mathbf{C}(\xi)) + \nabla \cdot (D \nabla \mathbf{C}(\xi))\|_2^2 \\
& + \lambda_2 \sum_{\tau \in \mathcal{T}} \sum_{\xi=\tau-M+2}^{\tau-1} \|\nabla \cdot \mathbf{u}(\xi)\|_2^2,
\end{aligned}
\tag{5.28}
$$

where the objective function is written in the continuous domain. In order to define the forecasting problem on the irregular grid, we use the discretization of the PDE equation and divergence of the velocity field defined in the Subsection 5.2.3.

5.4 Performance Evaluation

5.4.1 Experimental settings

We have trained and evaluated the PING model and compared its performance against state-of-the-art models for three different types of datasets. The first datasets are synthetic and follow Navier-Stokes equations for incompressible flows. They include concentration and velocity values of particles needed to analyse the model thoroughly. Then we consider two real-world datasets, which entail sea surface temperature and cloud concentration indexes. The final dataset is a real PV power production data, which is not an advection-diffusion process but is influenced by the cloud coverage, which itself can be considered as an advection-diffusion process. We compared the performance of PING with a state-of-the-art graph-based model across all datasets.

Datasets

Two simulated synthetic datasets are generated using PhiFlow package of Holl et al. (2020) [1]. The governing equations in the simulation follow advection-diffusion principles of a smoke behaviour, and it is influenced mainly by advection, along with ensuring the incompressibility of the fluid. Although both datasets have been generated using similar conditions, one dataset is purely advective, while in the second one the diffusion is added. The diffusion coefficient in this dataset is D=0.6. For both datasets, we have simulated 24000 time steps for training and 6000 time steps for evaluation. The time step $\Delta t = 1$ is defined as a smoke density moved in one spatial unit per one time unit. Every 50 steps, we inject smoke randomly. Both datasets

[1] https://github.com/tum-pbs/PhiFlow

are simulated on the discretised domain of the size 64×64, and then they are downsampled to the size of 32×32, to reduce the spatial smoothness of the data. Since the goal is capturing of the advection-diffusion dynamics in the case of limited data, we have used 484 nodes of the synthetic datasets when the model is evaluated on the regular grid and 150 nodes on the irregular grid.

The following two datasets in our study consist of records from sea surface temperature and cloud coverage. Both datasets are ERA5 datasets, which are publicly available [2]. ERA5 is a global atmospheric reanalysis dataset produced by the European Centre for Medium Range Weather Forecasts (ECMWF). The cloud coverage index has an hourly resolution, while the sea surface temperature dataset has a daily resolution. Both of these two datasets have spatial resolution of 0.25° latitude-longitude resolution and they are of the size 20×20 for the experiments on the regular domain. The training dataset for cloud data is from 2013 to 2016, while sea surface temperature dataset has range from 2006 to 2017. The evaluation is done for the year 2017 for the cloud dataset and from 2018 to 2021 for sea surface temperature. For these datasets, 400 nodes are used when the model is evaluated on the regular grid and 150 nodes on the irregular grid.

The PV power dataset consists of the real PV data from PV plants across Switzerland (Carrillo et al., 2020) for two years (2016-2017) with 15-minutes resolution. The PV plants are spread inhomogeneously over the entire country, with a density reflecting the population density. In the PV power prediction task, satellite images are also used as an input to the model. They are obtained with hourly resolution from [3], by subsampling the original image with longitude and latitude of PV stations. Once the future cloud concentration predictions are made, a linear interpolation is used on cloud concentration index dataset to obtain 15-minutes resolution of cloud data.

For all four datasets, the same nodes are used both as the input and forecasting nodes. In all four datasets, the graphs of 150 nodes are obtained by randomly sampling the given datasets. In order to demonstrate the effectiveness of the model, it has been evaluated on four different graphs per dataset.

Benchmark models

Two methods have been used as benchmarks in forecasting evaluation. These models have been chosen since they can handle both regular and irregular grid predictions. The first benchmark is a graph convolutional long-short memory network (GCLSTM), defined in Chapter 3. The GCLSTM model has the encoder which estimates the past information and passes it to the decoder which makes the final predictions. To show the effectiveness of the additional dynamics in the decoder, the GCLSTMmlp is used as the second benchmark. As in the previous benchmark, GCLSTMmlp captures spatio-temporal dynamics in the encoder

[2] https://cds.climate.copernicus.eu/
[3] https://cds.climate.copernicus.eu/

via graph convolutional neural network and long-short term memory network. However, the difference with GCLSTM is that GCLSTMmlp does not have the propagation of the dynamics in the decoder. In the decoder of GCLSTMmlp, the embedded inputs and estimations from the encoder are used as an input to MLP where the final forecast is made. This type of simple decoder is similar to the last layer in our PING architecture. The physics-informed model for sea surface temperature is presented in the works of De Bézenac et al. (2019), FlowCNN, for regular grids on sea surface temperature and cloud concentration index dataset. Therefore, we use this method as benchmark on the datasets that reside on a regular grid. Another proposed methods for learning physical processes on irregular grid, such as works of Sanchez-Gonzalez et al. (2020) assume that the large datasets are available, from a few thousand up to 20 000 data points trained on 1000 trajectories with 300-2000 time steps, each. Since both of these were not meant to be trained on a small dataset with limited number of historical training samples, we are not going to benchmark against these models.

Evaluation and metrics

The model performance is assessed usingthe peak normalized root-mean-square error (NRMSE), defined in Chapter 3.

For comparing the flow direction we use the directions of the velocity estimations, such that the angle ϕ_i^t between the observed $\mathbf{v}_i^t$ and estimated vector values $\hat{\mathbf{v}}_\mathbf{i}^\mathbf{t}$ at each time step t in the given estimated sequence $T = 6$ are calculated:

$$\phi_{ij}^t = arccos(|\mathbf{v}_\mathbf{i}^\mathbf{t}| \cdot |\hat{\mathbf{v}}_\mathbf{i}^\mathbf{t}|), \tag{5.29}$$

where $|\mathbf{v}_i^t|$ represents the magnitude of the vector $\mathbf{v}_i^t$. This angle is calculated for every time step in the dataset and every node i.

5.4.2 Experimental results

In this section, we first evaluate the performance and efficiency of the model in a well-calibrated, synthetic environment. This controlled setting allows us to assess the model's capacity to accurately capture flow dynamics by aligning estimated velocity directions with ground truth. Following our synthetic dataset evaluations, we extend our assessment to real-world datasets where the underlying velocity fields are inherently unknown: cloud concentration index, sea surface temperature and PV power generation. Given the absence of ground truth for velocity in these datasets, our focus shifts exclusively to the model's predictive accuracy on these datasets.

We evaluate given datasets, except the PV dataset, first on regular domain and then on the randomly sampled graphs (irregular domain) from a given regular grid. The PV dataset inherently lies on an irregular domain, and it is evaluated on irregular grid, without additional subsampling.

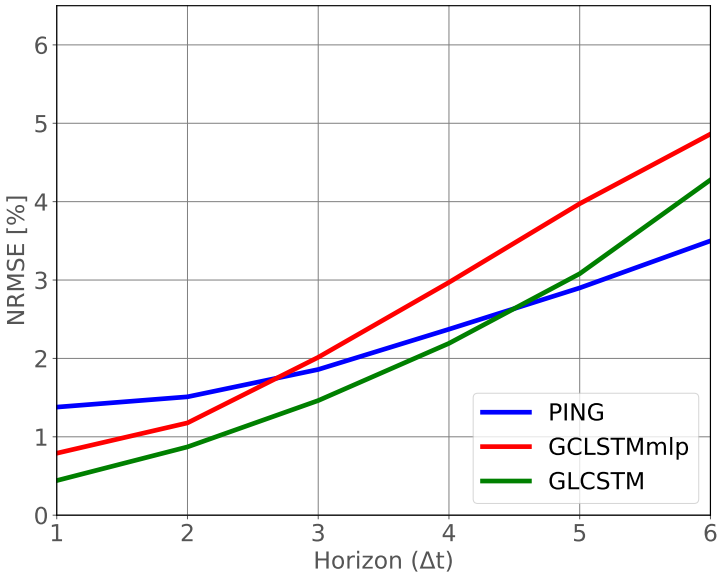
Prediction accuracy on Synthetic datasets

First, we evaluate the proposed model on the synthetic datasets. We show the forecasting NRMSE in Figure 5.5 for both synthetic datasets on a regular grid. Figure 5.5a shows the NRMSE for the advective dataset when nodes reside on the regular grid. The proposed model PING outperforms the GCLSTMmlp on the regular grid for three to six steps ahead of prediction. Both of these two models only have an MLP in the decoder. However, compared to the GCLSTM, which has a dynamics evolution of the signal in the decoder, our proposed model outperforms the baseline in the last two values of the forecast horizon. GCLSTM benefits from the recurrent network in the decoder which is propagating the future dynamics. Both GCLSTMmlp and proposed model would benefit from the propagation of the future dynamics via recurrent neural network. The NRMSE in Figure 5.5b shows the evolution of the error on the second synthetic dataset that has been created with both advective and diffusive properties. PING model outperforms GCLSTM and GCLSTMmlp on this dataset for the second half of the prediction horizon values, for three to six step ahead predictions.

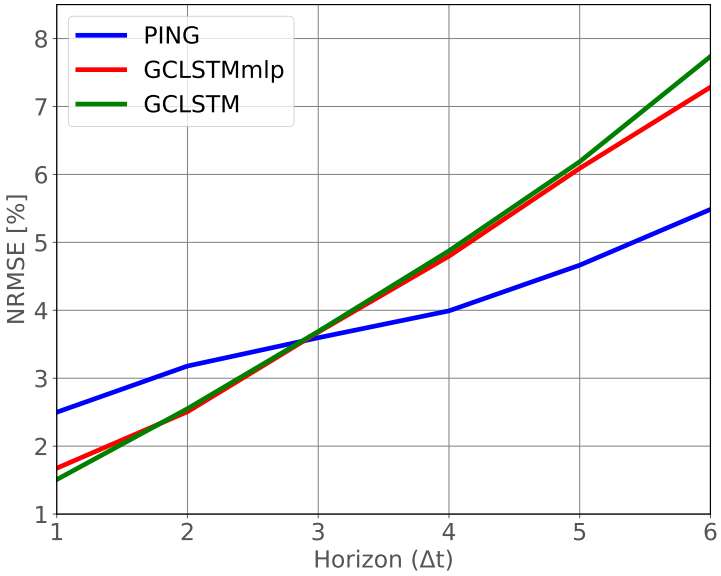
Moreover, we have also evaluated both synthetic datasets on irregularly subsampled domains. We employed a stochastic subsampling strategy on the original regular datasets. Four non-uniform inhomogeneous graphs were generated. These instances were then used as the foundation for constructing graph neural networks.

The NRMSE is shown in Figure 5.6 for both advective and advective-diffusive datasets when signals reside on the irregular domain. The PING model has a similar performance on irregular domains to the one on the regular grid. It has higher accuracy than GCRNN in the last two values of the forecast horizon in the advective dataset and from third to sixth horizon values on the advection-diffusion dataset. PING also outperforms the second baseline, GCLSTMmlp, from third to sixth horizon values on both datasets.

In order to analyse whether the model captures the direction of the flow, we have compared the absolute value of the angle between estimated velocities and the actual velocity for each node, for every time step in the dataset and every node v_i. The angle between the estimated and the ground truth velocity vectors, defined in Equation 5.29, is classified as acceptable or non-acceptable, with two different criteria. These two criteria are established based on the empirical observations and considering the identification of the predominant direction of the overall flow as main goal. The first criterion is strict, defining the “acceptable” angle where the absolute angle value is lower than 30°. Whereas the second, relaxed criteria, defines the acceptable angle as any absolute value of an angle below 45°. Accordingly, the “non-acceptable” angle estimation is considered when the absolute value of the angle is between 60° and 90° using the strict criteria, or the angle above 45° in case of the relaxed criteria. The choice of the thresholds $(30°, 45°, 60°)$ balances the trade-off between flexibility and rigidity.The strict criteria was used to estimate the accuracy of the velocity estimation, whereas the relaxed criteria is used as an indicator of the model’s ability to estimate the overall flow of the model. Table 5.41 below compares the acceptable and acceptable estimates of the velocity, according



(a)

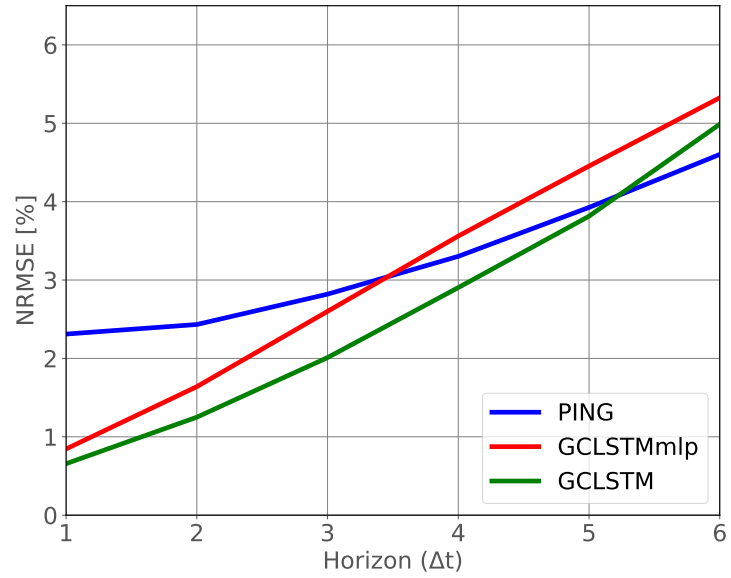


(b)

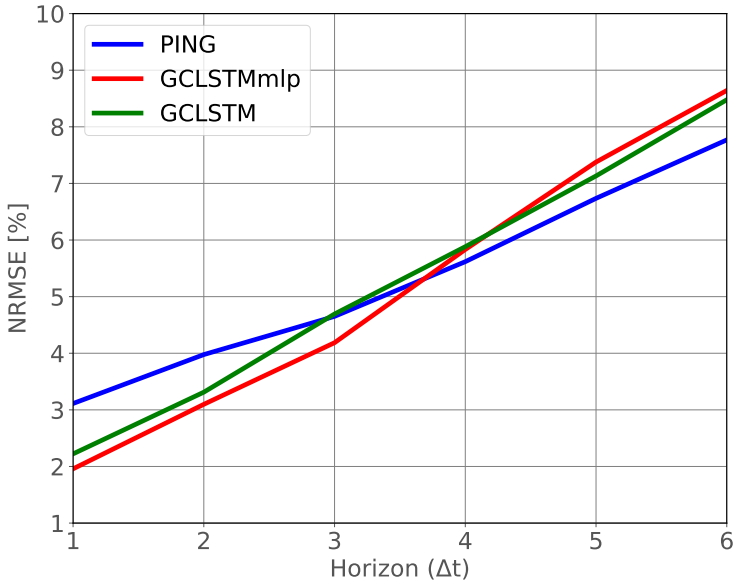
Figure 5.5: Evolution of the NRMSE between PING, GCLSTMmlp and GCLSTM for six-steps ahead prediction for the synthetic datasets on a regular grid. Solid line shows the median value among all nodes. a) NRMSE for the synthetic advective dataset. b) NRMSE for the synthetic advection-diffusion dataset.

to both criteria, for both advective and advective-diffusive datasets.

The results show that the proposed model is capable of estimating 61% of velocity directions on the advective irregular dataset and 77% of velocity directions on the regular advective



(a)



(b)

Figure 5.6: Evolution of the NRMSE between PING, GCLSTMmlp and GCLSTM for six horizon values on the synthetic datasets on an irregular grid. Solid line shows the median value among all nodes. a) NRMSE for the synthetic advective dataset. b) NRMSE for the synthetic advection-diffusion dataset.

dataset, which resides on the regular grid, considering that the angle below 45° is considered small enough for the acceptable flow estimations. We compare the estimated velocity from our proposed model with the Gunnar-Farneback optical flow model (Farnebäck, 2003). The

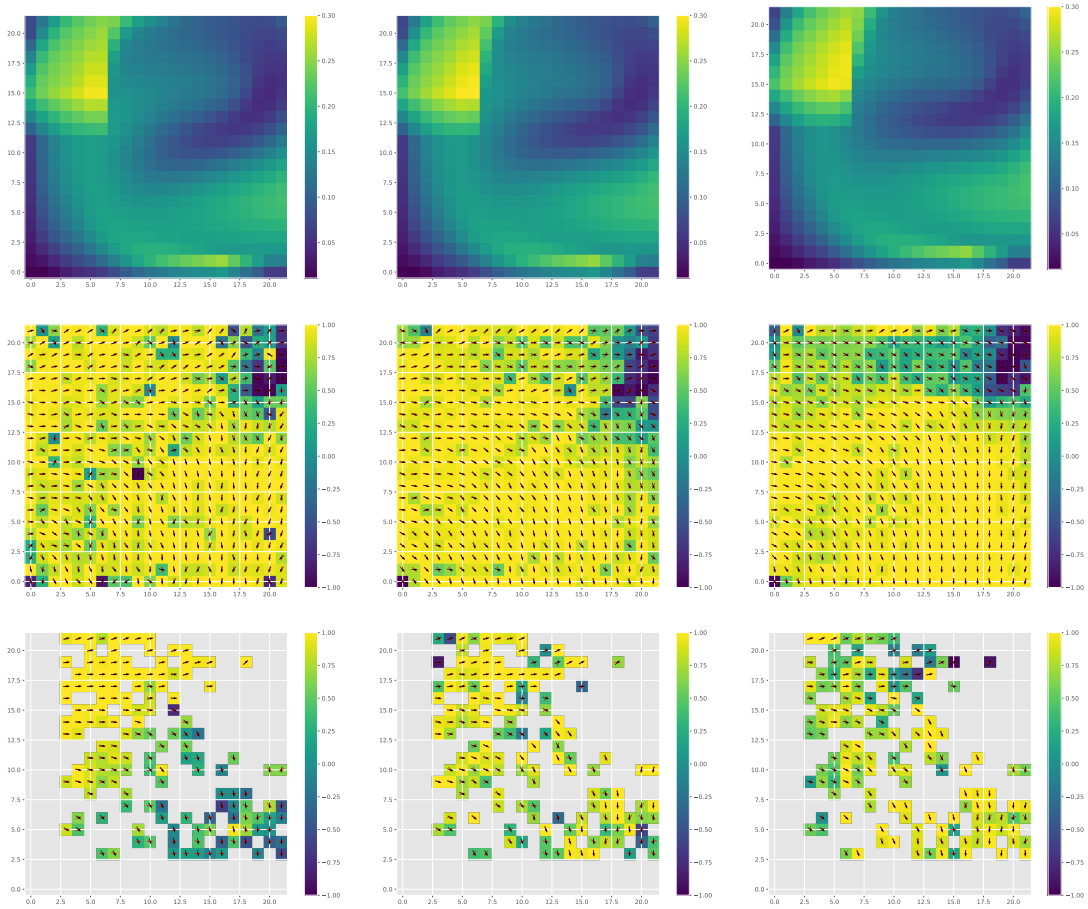


Figure 5.7: Example of particle concentrations (top row), and their corresponding velocity estimations on the regular (middle row) and irregular grid (bottom row) on the advection dataset, in the first, third and fifth values of the input horizon. Top row presents the evolution of the concentration change across the nodes and their ground truth flow values. The lighter colours (yellow) depict higher concentration of the particles at observed nodes, while darker colours (dark blue) depict the values where concentration is close to zero. The middle and bottom row depict the similarity between the ground truth velocity direction and estimated one. The lighter colours (yellow), in these plots, represent the perfect alignment between the estimated and ground truth vectors. The darker colours correspond with the higher angle between the ground truth and estimated velocity direction, such that dark blue represent the opposite direction.

Gunnar-Farneback Optical Flow is traditional optical flow method, chosen since it computes the dense optical flows, for all the points, between the two consecutive time steps on a regular grid. The results are shown in Table 5.42 and show that our proposed model outperforms the optical flow model on a regular grid.

In order to better visualize the similarity between ground truth and estimated velocity vectors, we introduce a colour-coding scheme applied to each node (node) within a grid. The shade

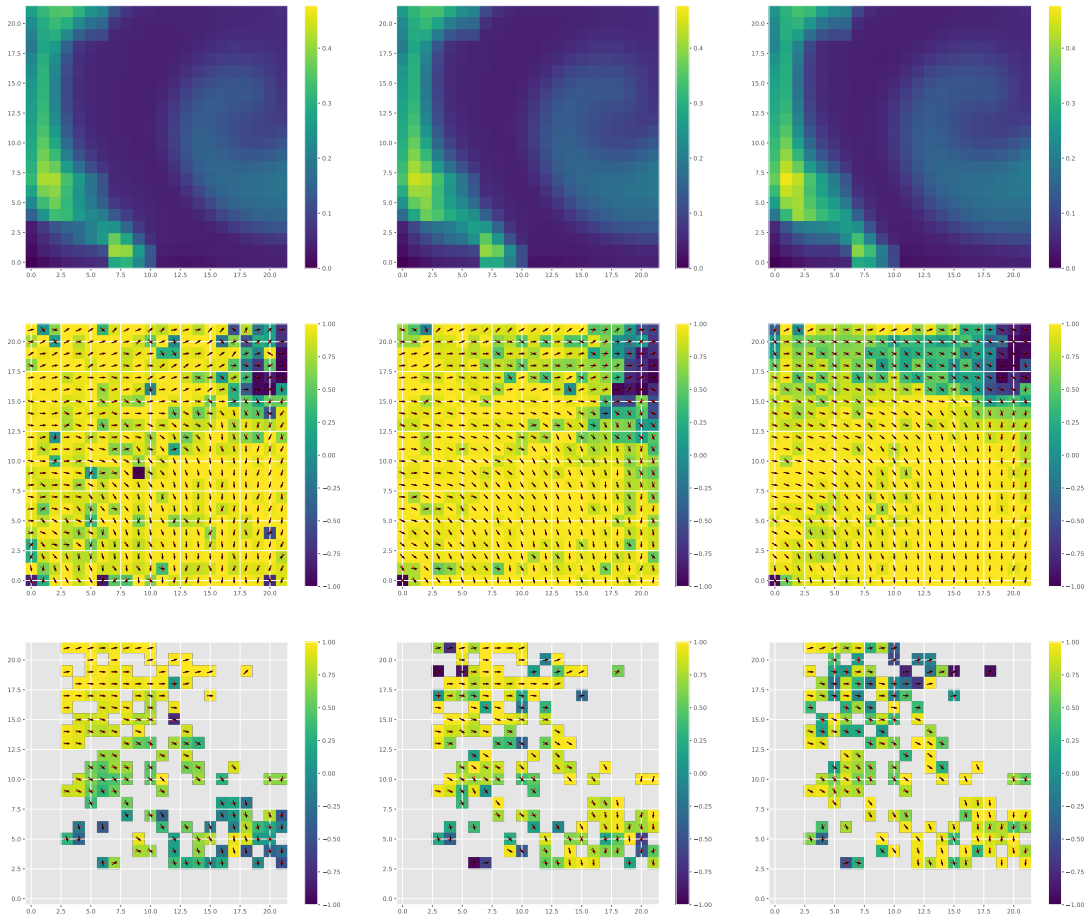


Figure 5.8: Example of particle concentrations (top row), velocity estimations on the regular (middle row) and irregular grid (bottom row) on advection dataset, in the first, third and fifth values of the input horizon. Top row presents the evolution of the concentration change across the nodes and their ground truth flow values. The lighter colours (yellow) depict higher concentration of the particles at observed nodes, while darker colours (dark blue) depict the values where concentration is close to zero. The middle and bottom row depict the similarity between the ground truth velocity direction and estimated one. The lighter colours (yellow), in these plots, represent the perfect alignment between the estimated and ground truth vectors. The darker colours correspond with the higher angle between the ground truth and estimated velocity direction, such that dark blue represent the opposite direction.

of each pixel at node v_i is modulated based on the angle between the two vectors, $\mathbf{v_i^t}$ and $\hat{\mathbf{v}}_\mathbf{i}^\mathbf{t}$, which are located on top of that pixel. We use the dot product between the observed and estimated velocity vectors $\mathbf{v_i^t}$ and $\hat{\mathbf{v}}_\mathbf{i}^\mathbf{t}$ to find the angle value between those two vectors, as defined earlier in Equation 5.29. The colour encoding utilizes a spectrum between 0 and 180 degrees (1 and -1, in radians), such that yellow shades of the node represent the scenario when two vectors are coincident, green nodes represent the scenario when vectors are orthogonal and finally, dark blue nodes represent the parallel vector with opposite direction.

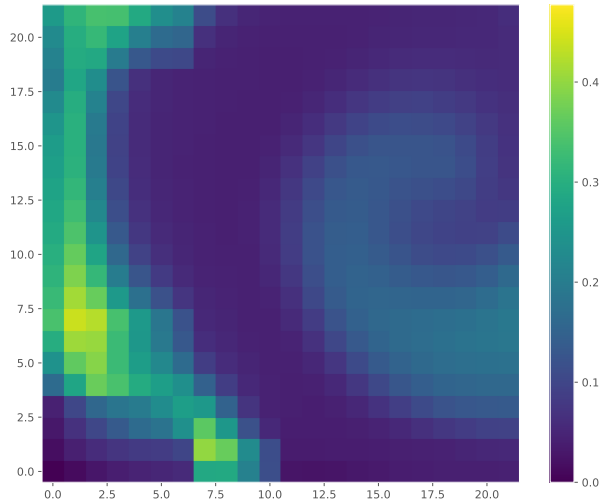


Figure 5.9: Example of particle concentrations at certain time step. Yellow colour is for high concentration values, while dark blue is used when concentration values are zero.

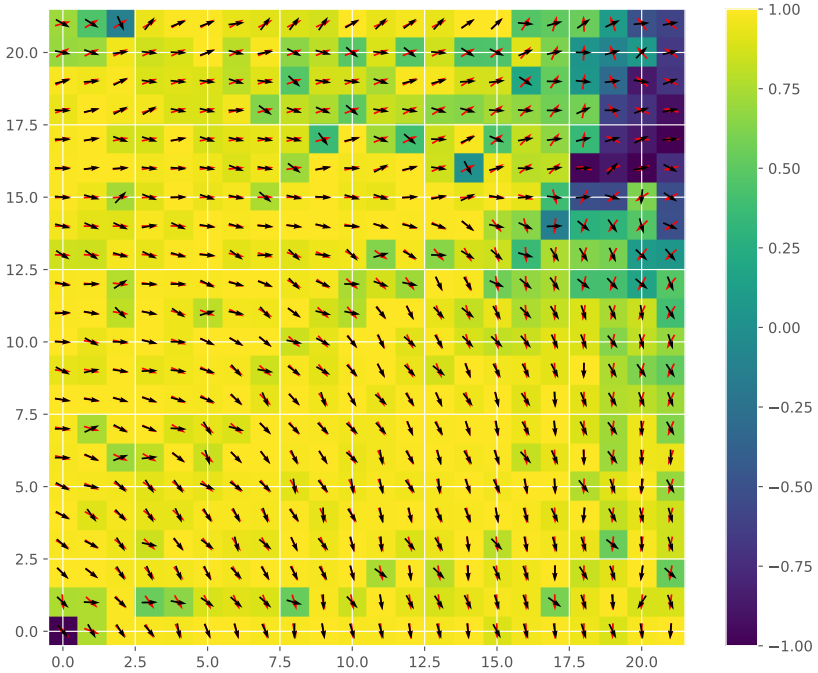


Figure 5.10: Example of velocity estimations on the regular grid on advection dataset, on the regular grid. The similarity between the ground truth velocity direction and estimated one is colour-coded. Yellow represent the perfect alignment between the estimated and ground truth vectors. The darker colours correspond with the larger angle between the ground truth and estimated velocity direction, such that dark blue represent the opposite direction. The arrows represent the direction of the velocity vector, black colour is the ground truth and red is the estimation.

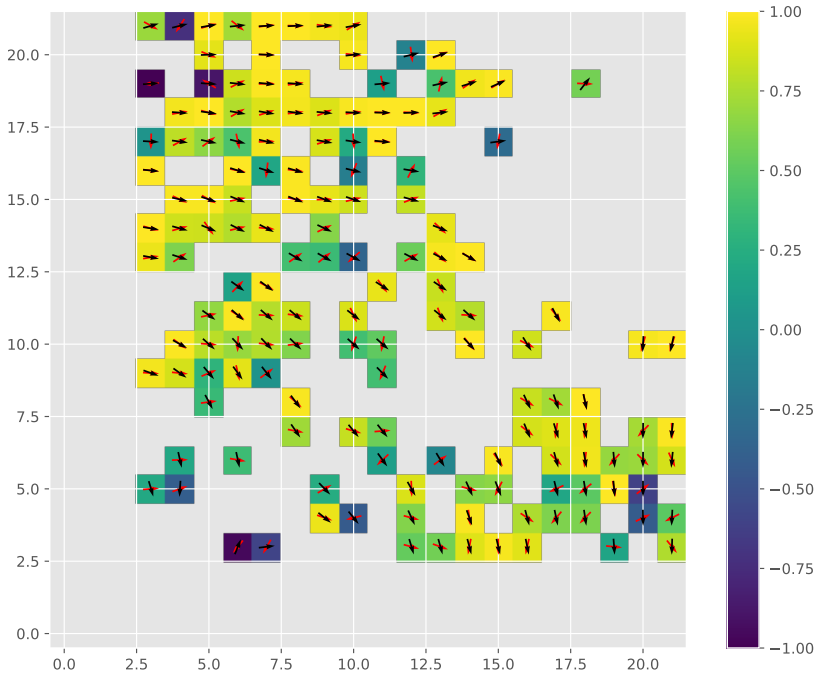


Figure 5.11: Example of velocity estimations on the regular grid on advection dataset, on the irregular grid. The similarity between the ground truth velocity direction and estimated one is colour-coded. Yellow represent the perfect alignment between the estimated and ground truth vectors. The darker colours correspond with the larger angle between the ground truth and estimated velocity direction, such that dark blue represent the opposite direction. The arrows represent the direction of the velocity vector, black colour is the ground truth and red is the estimation.

We visually show particle concentration and the estimated ground truth flow directions for regular and irregular grids on the advective dataset in Figures 5.8 and 5.7. These two examples depict the models' ability to capture the flow direction. The black arrows correspond to the ground truth velocity direction, and the red arrows correspond to the estimated flow direction. On top of this, we use our colour-coded representation of the vector field differences on the same examples for advective synthetic datasets on regular and irregular grids; see Figures 5.7 and 5.8. Both of these samples show that when model is able to estimate the velocity very well in the situation when the concentration changes are smooth on the regular grid. Moreover, on the irregular grid, it is more difficult task to capture the overall flow, since the model does not have complete information. Despite having the incomplete information on the concentration change, the model is able to capture overall flaw direction successfully.

We highlight the results from the first example, depicted in Figure 5.7. The model captures the velocity values perfectly on the regular grid, see Figure 5.10. On the irregular grid on Figure 5.11, it is clear that model is not able to estimate the velocity directions with high percentage of the acceptable directions. However, it is still able to capture the overall flow direction.

Table 5.41: Accuracy of velocity estimation on regular and irregular grid for advection and advection-diffusion datasets

		Strict		Relaxed	
		acceptable	non-acceptable	acceptable	non-acceptable
		$\phi < 30°$	$60° < \phi < 90°$	$\phi < 45°$	$\phi > 45°$
Advection dataset	Regular domain	61%	13%	77%	23%
	Irregular domain	42%	25%	60%	40%
Advection-diffusion dataset dataset	Regular domain	40%	30%	57%	43%
	Irregular domain	44%	22%	63%	37%

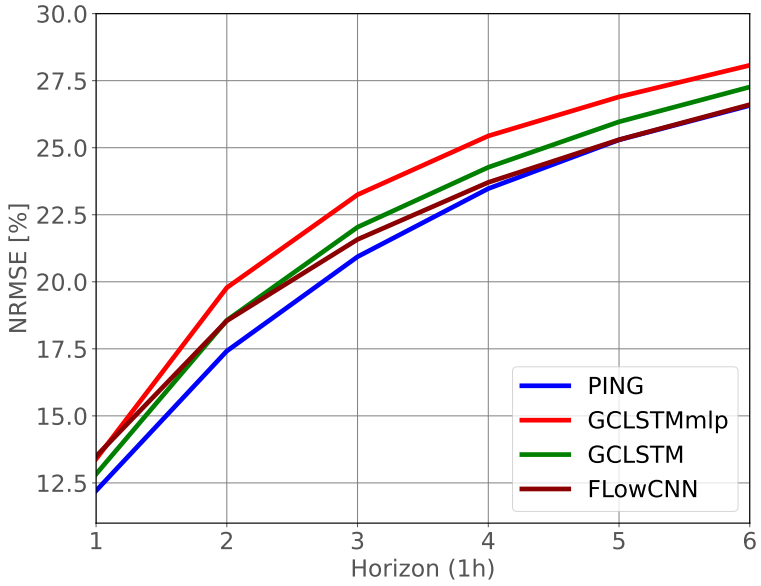
Table 5.42: Accuracy of velocity estimation on regular grid for advection and advection-diffusion dataset, for PING and Optical flow model

		Strict		Relaxed	
		acceptable	non-acceptable	acceptable	non-acceptable
		$\phi < 30°$	$60° < \phi < 90°$	$\phi < 45°$	$\phi > 45°$
Advection dataset	PING model	61%	13%	77%	23%
	Optical flow	28%	41%	43%	57%
Advection-diffusion dataset	PING model	40%	30%	57%	43%
	Optical flow	32%	36%	48%	52%

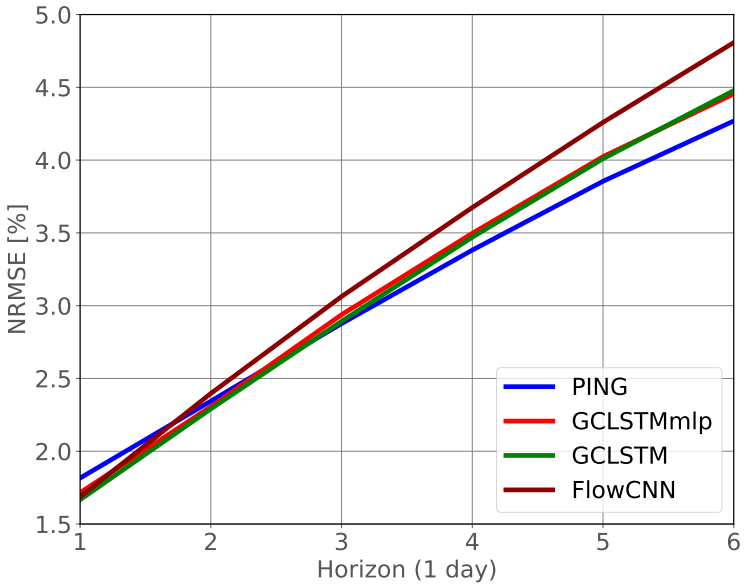
Prediction accuracy on Real datasets

We evaluate the model's performance on the cloud and sea surface temperature datasets over a grid size of 20×20 pixels (nodes). In Figure 5.12, the NRMSE evolution of the PING model is compared to the encoder-decoder GCLSTM, FlowCNN and GCLSTMmlp over the six horizon values on the regular grid. For the cloud dataset, the predicted horizon is 6 hours ahead with hourly resolution, whereas for the sea surface temperature, it is six days, with a daily resolution. The solid lines represent the median of error over all nodes. The proposed model outperforms GCLSTM on the cloud dataset, probably related to the cloud movement being more advective than the diffusive process. Furthermore, it outperforms FlowCNN for the first five hours of the prediction horizon on cloud concentration index dataset. However, when it comes to a spatially and temporarily smoother, more diffusive process, such as the sea surface temperature, they are on par in the first steps of the prediction since GCLSTM benefits from diffusing information across the local neighbourhood via a convolutional graph neural network. PING outperforms FlowCNN on the sea surface temperature from the second to

the sixth day ahead of the forecasting horizon. Nevertheless, the PING model outperforms the GCLSTM from the third to sixth horizon values on the SST dataset, when not only local dynamics are affecting the forecast, but more global dynamics also affect the concentration of the particles.

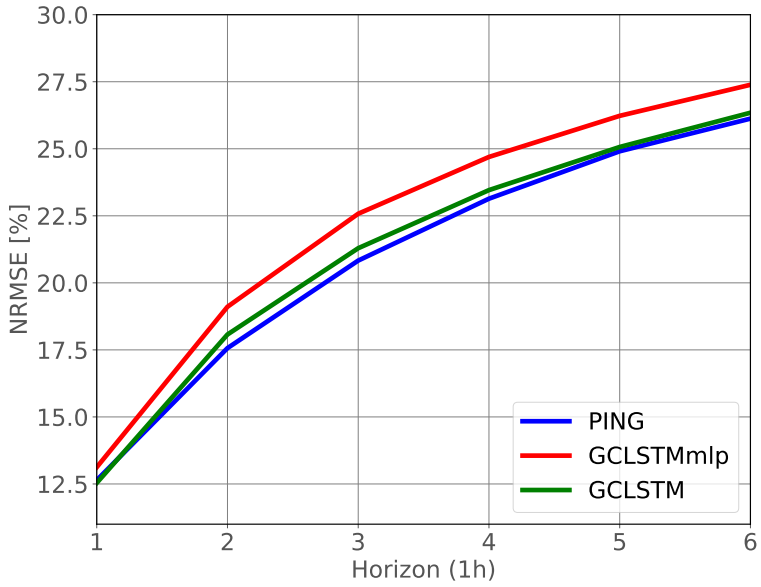


(a)

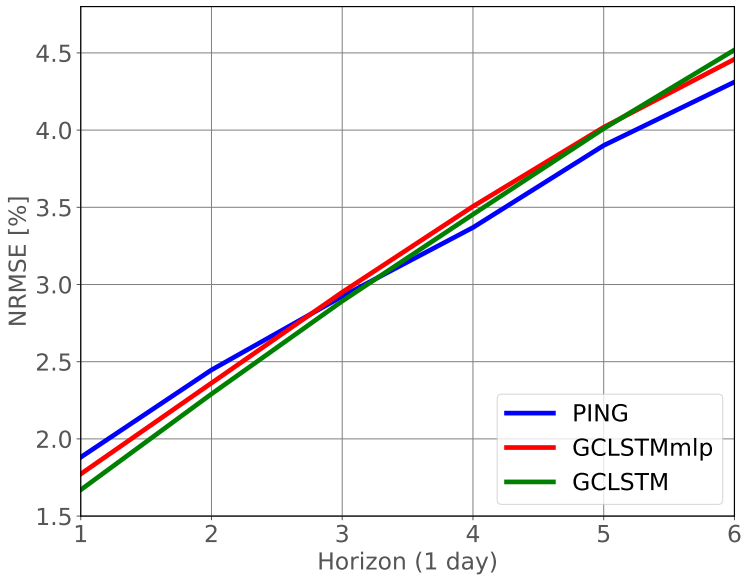


(b)

Figure 5.12: Evolution of the NRMSE between PING, FlowCNN GCLSTMmlp and GCLSTM models for six-step ahead prediction for the weather datasets. Solid line shows the median value among all nodes. a) Forecast NRMSE for the cloud dataset on a regular grid. b) Forecast NRMSE for the SST dataset on a regular grid.



(a)



(b)

Figure 5.13: Evolution of the NRMSE between PING, GCLSTMmlp and GCLSTM models for six-step ahead prediction for the weather datasets on an irregular grid. Solid line shows the median value among all nodes. a) NRMSE for the cloud dataset. b) NRMSE for the SST dataset.

Additionally, we compare the performance of the proposed model and both benchmarks on the irregular grid for 150 nodes. First, we show the evaluation NRMSE for cloud and SST datasets. The results are shown in Figure 5.13. PING and GCLSTM models perform similarly on the regular and the irregular grid. The proposed model still outperforms GCLSTM on a

cloud dataset on an irregular grid. On the SST irregular dataset, it outperforms GCLSTM from the second to sixth horizon values of predictions. We have also evaluated the GCLSTMmlp model, which has an overall higher error than GCLSTM. Although GCLSTMmlp has slightly higher NRMSE error for three-steps ahead forecasts (for up to 2% and 0.25% on the cloud and SST datasets, respectively) than GCLSTM and PING, it has very high error for more than three steps ahead. On the regular and irregular sea surface temperature datasets, the PING model is either on par or outperforms GCLSTM from the third to sixth horizon values of the forecast.

We analyse the forecasts made at different days and different times between 25th and 29th February on the irregular grid for cloud concentration index dataset. We present those forecasts over a 6-hour prediction horizon on Figure 5.14 and Figure 5.15. These specific instances are selected since they exemplify moments of significant dynamical changes in the cloud coverage. Such moments include instances where the observed value rapidly declines from 1 to 0.55 or exhibits abrupt surges. These situations are particularly challenging for forecasting models, and as demonstrated in the Figure, our proposed model PING shows better performance compared to GCLSTM, when predicting these sudden changes. We do not show comparison with the GCLSTMmlp model, since it has the highest NRMSE among compared models.

Prediction accuracy on PV power production dataset

The cloud movement and cloud creation are advection-diffusion processes that affect the amount of the sunlight reaching the photovoltaic panels on the ground. As clouds move, they block the sunlight, causing the fluctuations in the irradiance, and consequently, they cause the variations in the PV power production. Predicting the cloud movements and modelling their dynamics is essential for accurate PV forecasting. We have shown that the proposed model outperformed benchmarks on the cloud concentration index dataset, on both regular and irregular grid. Therefore, PING could be used for sensory data that depends on advection-diffusion, such as PV power production. However, PV power production data is much more complex to predict than cloud concentration, since it depends on other factors, including local atmospheric conditions, solar panels efficiency and the angle between the sunlight and the panels surface. Thus, we use PING to make future prediction of the cloud index concentration and then use those predictions as an inductive bias in a previously developed PV power forecasting model, such as GCLSTM. We investigate the performance of combining the PING and encoder-decoder GCLSTM model that use cloud concentration index and the PV power production data.

The proposed combination of PING and GCLSTM is shown in Figure 5.16. First, the PING model is used to estimate the cloud movement from past cloud concentration index values $\mathbf{C}$ and predicts the future cloud concentration index values $\hat{\mathbf{C}}$. The past cloud concentration values $\mathbf{C}$ are concatenated with the past PV power production $\mathbf{P}$ and clear-sky irradiance values $\mathbf{Y_{sky}}$ and they are used as an input to the encoder of GCLSTM. The encoder estimates the state of the system, and those estimations with predicted future cloud concentration values $\hat{\mathbf{C}}$,

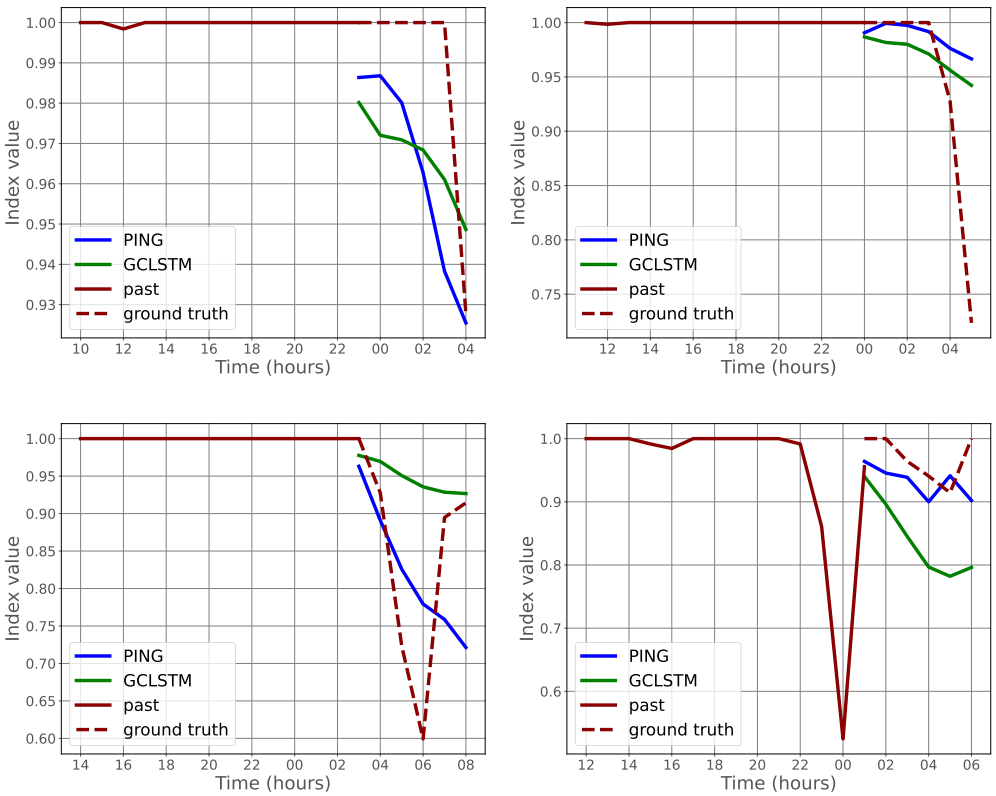


Figure 5.14: Examples of historical cloud concentration indices, ground truth and forecasted signals across six forecasting horizon values for a specific node on irregular grid. The results are shown for PING and GCLSTM models..

from PING model, and future clear-sky irradaince values $\hat{\mathbf{Y}}_{\mathbf{sky}}$, represent inputs to the decoder. Clear-sky irradiance is a deterministic variable, which is calculated at any time of the year and at any geographical location.

The combination of PING and GCLSTM model is evaluated on PV power generation datasets and compared to the baseline model GCLSTM, but also to GCTrafo, developed in the Chapter 3, and an interpretable TSM-GAT model, developed in Chapter 4. The NRMSE evolution is shown in Figure 5.17. The combination of PING and GCLSTM models outperforms all benchmarks on the entire horizon of six hours, with fifteen minutes resolution.

We review the forecasting results for a specific moment and time. The analysis is focused on six hours ahead forecast for three consecutive days beginning on the 29th of August for GCLSTM model and combination of PING and GCLSTM (see Figure 5.18). We chose this time because of the mixture of sunny, cloudy and variable days. Although the PING model performs similarly to the state-of-the-art GCLSTM model during sunny day in Figure 5.18, it outperforms the GCLSTM during the variable and cloudy days (the second and the third day in Figure 5.18). The combination of PING and GCLSTM models has an advantage that is shown on the prediction

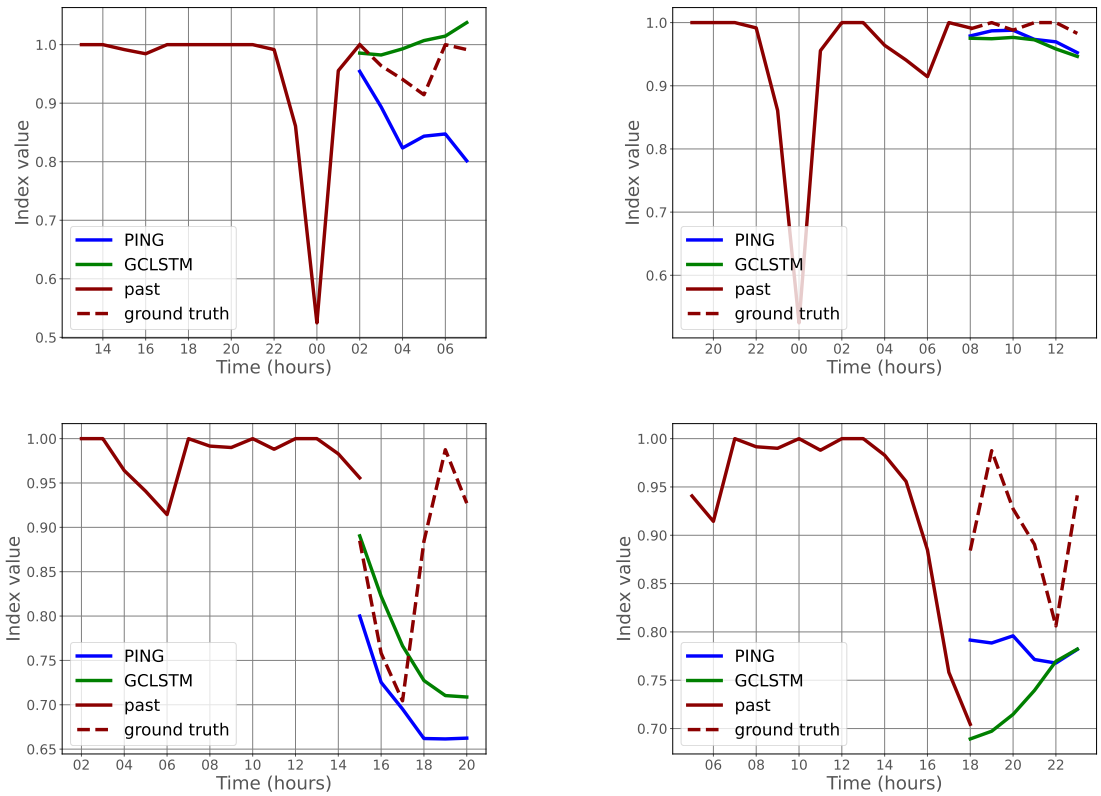


Figure 5.15: Examples of historical cloud concentration indices, ground truth and forecasted signals across six forecasting horizon values for a specific node on irregular grid. The results are shown for PING and GCLSTM models.

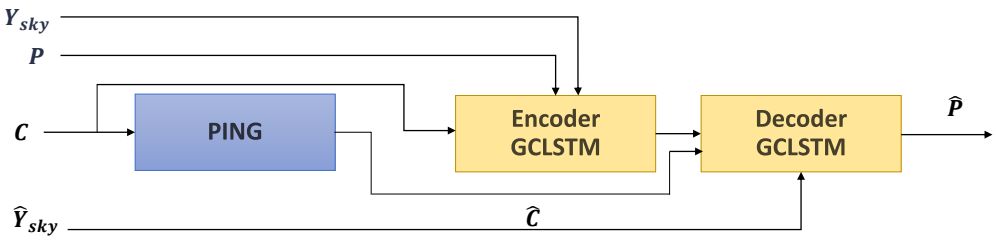


Figure 5.16: PING and GCLSTM setting for PV power prediction.

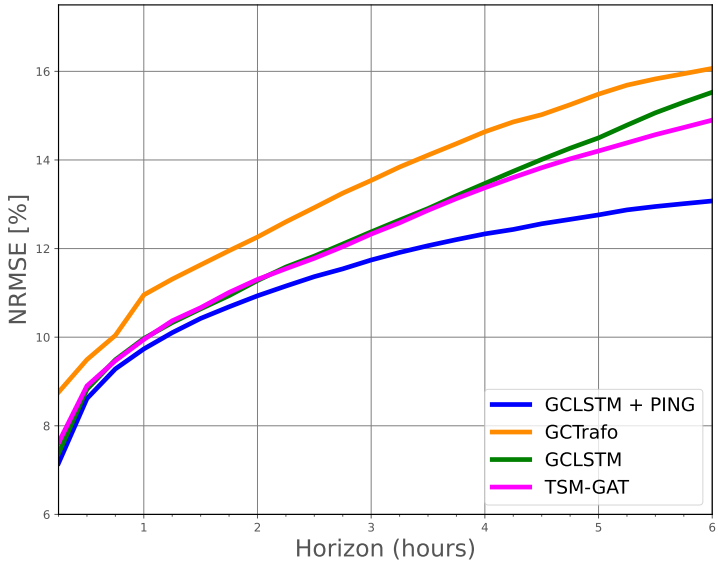


Figure 5.17: Evolution of the NRMSE between PING, GCLSTMmlp and GCLSTM models for six hours ahead for PV power generation, with hourly resolution. The solid line shows the median value among all nodes.

of the second and third day, where the shape of the prediction is closer to the ground truth compared to GCLSTM. Capturing better the shape of the cloud dynamics is the indication of the improvement since the previous analysis has shown that state-of-the-art models such as GCLSTM fail to capture the cloud movement during the variable days (Simeunović et al., 2022b).

In addition to the previous analysis, we compare the forecast made at different times of the day, for twenty-four horizon values, during a variable day in Figure 5.19. We have compared the combination of PING and GCLSTM, and GCLSTM. The proposed combination of PING and GCLSTM outperforms GCLSTM and can better capture and predict the cloud dynamics during the first half of the day. GCLSTM has a high error in the first hours of the morning prediction since it has no information on cloud dynamics during the night. Thus, GCLSTM relies on a clear-sky profile in this situation and local neighbourhood information, predicting a sunny day and making a high error when the first part of the day is cloudy. On the other hand, the combination of PING and GCLSTM extracts the information on the cloud dynamics during the early morning, allowing it to forecast future values with higher accuracy.

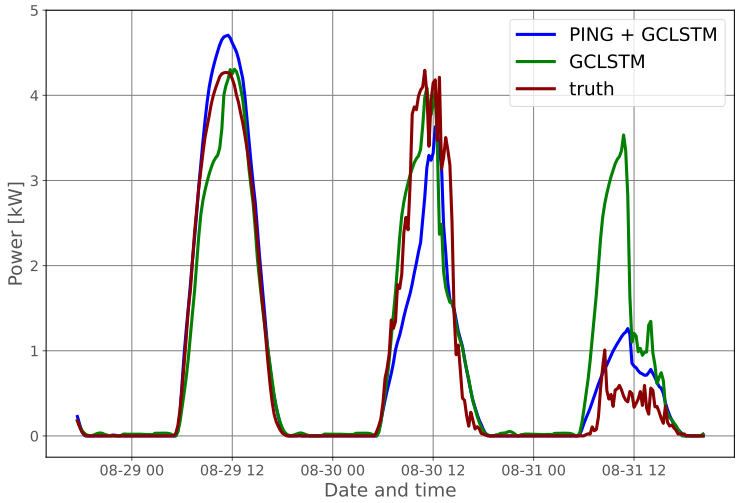


Figure 5.18: Ground truth and prediction for 24 steps ahead during variable days.

5.4.3 Performance analysis

Effect of subsampling

An additional study is performed to estimate the effect of the subsampling on the accuracy of the prediction. All four datasets that reside on the regular grid (cloud, SST, advection and advection-diffusion datasets) are subsampled randomly three times and once regularly, creating four sets of subsampled locations with size of 150. Each subsampled set of pixels can be represented as a set of nodes in a graph, thus forming graphs with different topologies. The accuracy of the proposed model is compared to GCLSTM performance for each set of nodes. GCLSTM is chosen as a benchmark since it showed higher accuracy than GCLSTMmlp when evaluated on the previous datasets. The results from the first subsampling set $\mathscr{S}_1$ are already shown in the previous section on Figure 5.6. The second subsampling set $\mathscr{S}_2$ on which we evaluate the proposed model is a regularly sampled pattern. We have focused on the direction of the flow, since the features representing velocity estimation undergo different non-linear transformations and are learnt in an unsupervised manner, which might results in a completely different scale of the magnitudes. Due to this transformation, direct comparison of the estimated velocity with the ground truth velocity becomes non-trivial. However, the direction of the estimated velocity vector retains its relative significance. It is, thus, more straightforward to compare the direction of the estimated velocity with the ground truth velocity direction. Comparing the magnitudes of the velocity estimation would require careful normalisation of the datasets, since the model might output velocity values in a transformed or scaled range that is mathematically convenient or optimal for its internal representations and not comparable to ground truth.

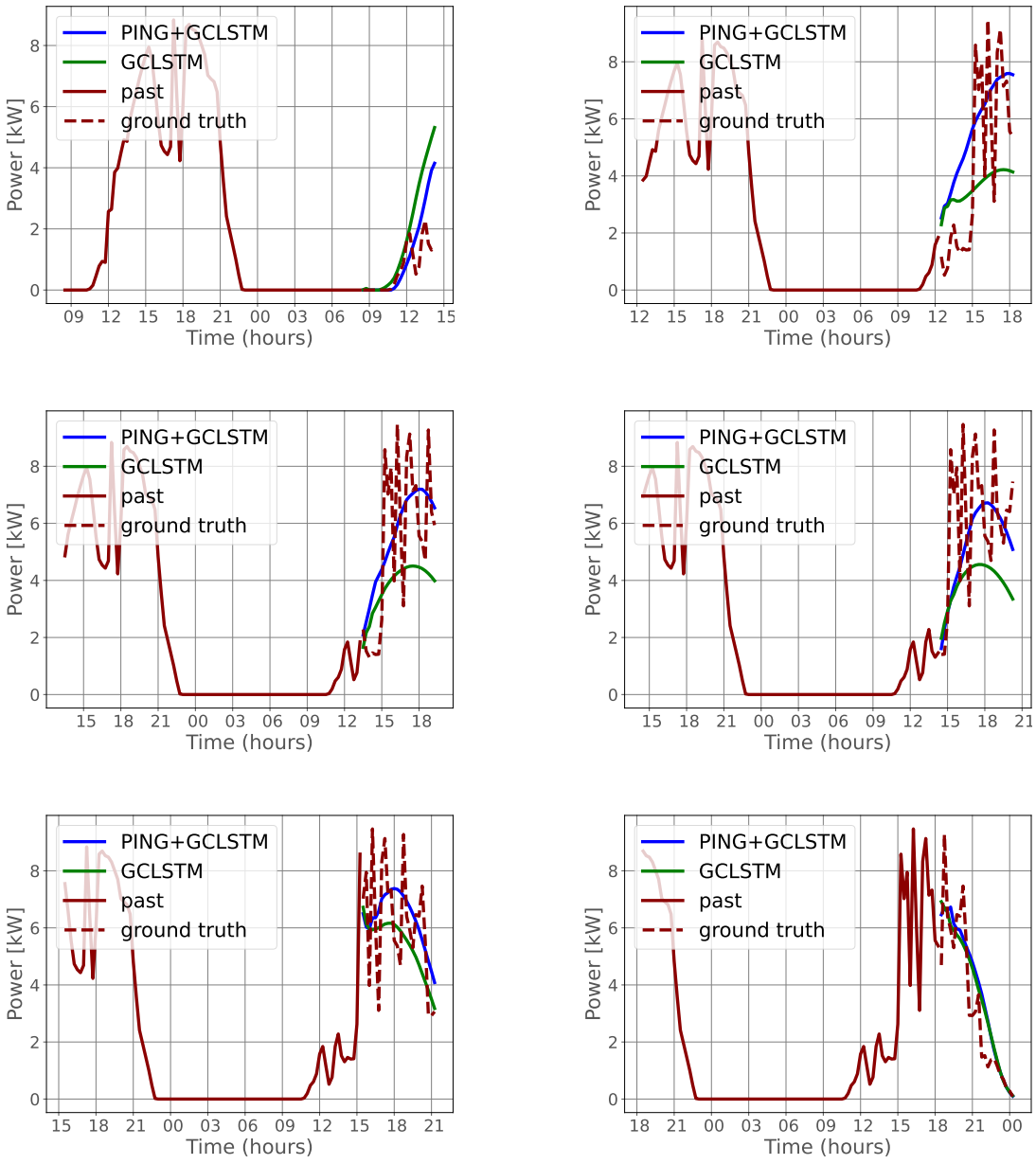


Figure 5.19: Examples of twenty-four prediction horizon values made at different times of day on the PV power prediction dataset. The forecasting signal, and the past signal values for GCLSTM and PING models are shown.

First, we examine the evolution of the error on the synthetic advection-diffusion dataset on the irregular grid for graph instances $\mathcal{G}_2, \mathcal{G}_3$, and $\mathcal{G}_4$. On these subsampling schemes, GCLSTM and GCLSTMmlp outperform the proposed model in the first three steps of the prediction, see Figure 5.21 and Figure 5.22. The underlying dynamics of this dataset are guided by a strong diffusion. The type of spatial distribution we use has a small or no effect on the mean accuracy

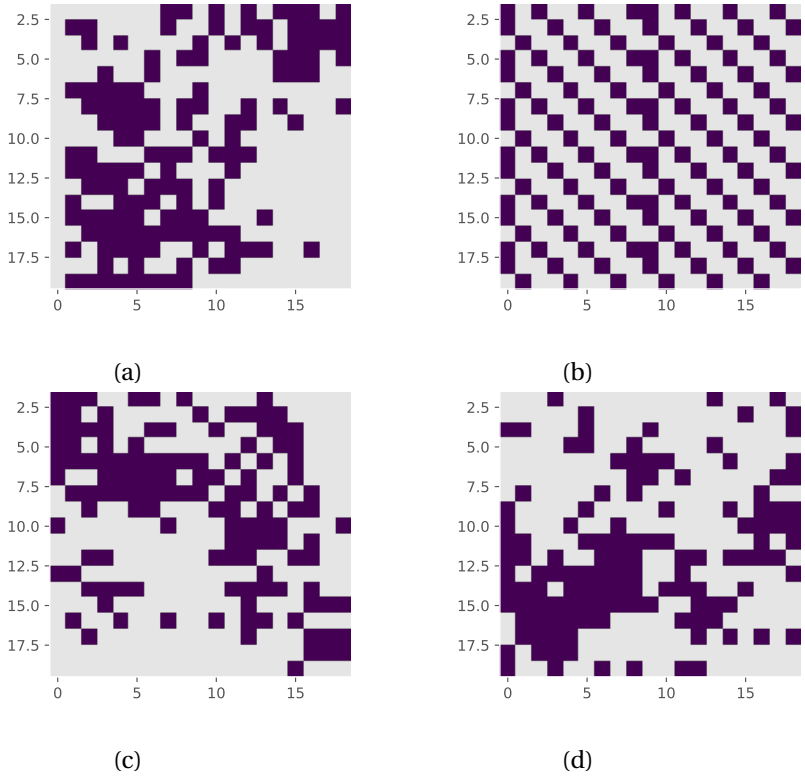


Figure 5.20: The spatial distribution of the subsampling patterns. a) Subsampling set $\mathscr{S}_1$. b) Subsampling set $\mathscr{S}_2$. c) Subsampling set $\mathscr{S}_3$. d) Subsampling set $\mathscr{S}_4$.

of the prediction of the PING model, whose median stays between 3% and 6.5%. Furthermore, the evolution of NRMSE across all four graphs of the PING model is in the same range as the error evolution of the PING model on the regular grid, whereas the spatial distribution has larger impact to both GCLSTM and GCLSTMmlp, where median error varies.The results indicate that the sampling pattern has a limited effect on the accuracy of the proposed model, which indicates that the model captures the underlying dynamics and it is robust to irregular sampling.

Subsequently, we examine the accuracy of the model on the advection dataset on irregular grid for graph instances $\mathscr{G}_2, \mathscr{G}_3$ and $\mathscr{G}_4$ defined for subsampling patterns $\mathscr{S}_2, \mathscr{S}_3$ and $\mathscr{S}_4$. On the purely advective dataset, the proposed model outperforms GCLSTM only in the last horizon value. However, it outperforms the GCLSTMmlp benchmark in the second half of the horizon, from fourth to sixth horizon values. The median NRMSE of PING model is constant for all subsampling patterns, and it is between 2% and 4%. The low error values combined with the fact that the error range is the same across different subsampling patterns suggests that the spatial position of the data is not the main driver of system's behaviour. Although PING model outperforms both benchmark models on the advection-diffusion dataset for second half of the forecasting horizon, on the purely advective dataset it is outperformed by the GCLSTM for the first five values of the forecasting horizon. The main reason lies in the fact that the model has

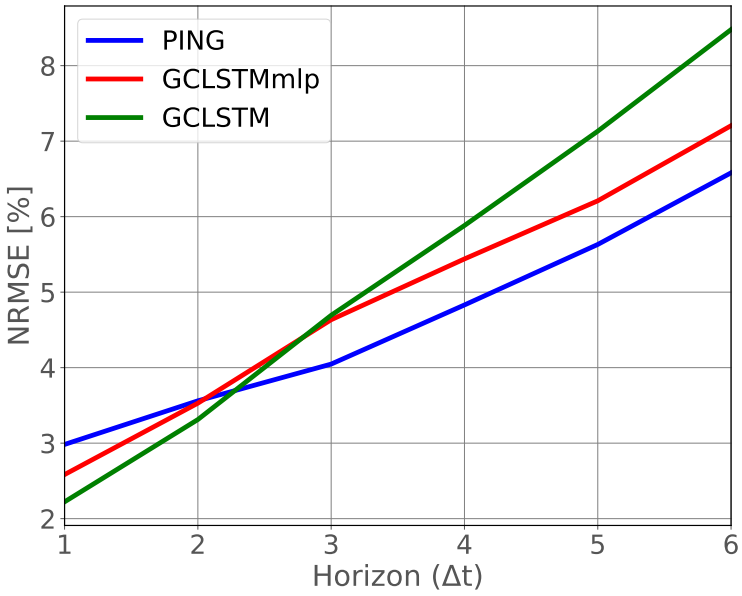


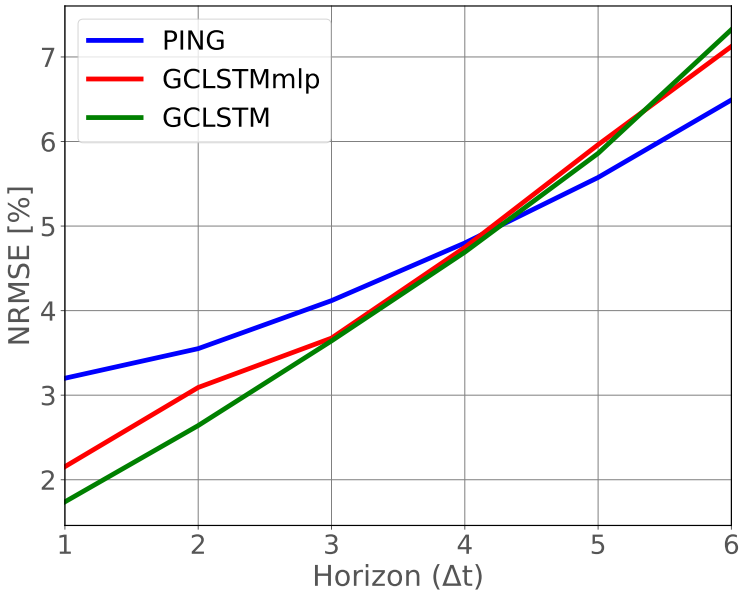
Figure 5.21: The evolution of NRMSE of PING, GCLSTMmlp and GCLSTM models on the irregular subsampling patterns $\mathscr{S}_2$ for synthetic advection-diffusion dataset.

advection-diffusion equation added as soft constraint in the optimisation function. Moreover, the second constraint added to the optimisation function of the model is an explicit constraint on the divergence of the velocity field. Since the spatial smoothness is mainly guided by the diffusion, the model could benefit from removing the diffusion from the soft constraint when trained on purely advective dataset.

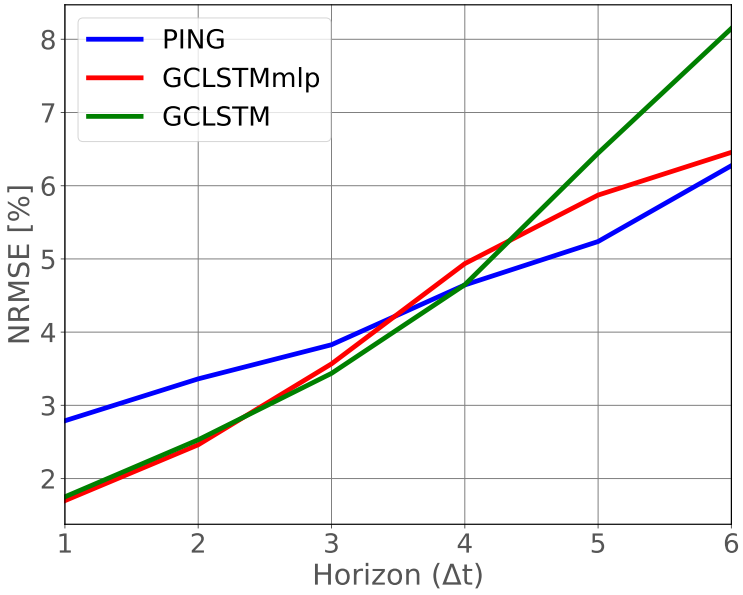
In addition, we examine the evaluation of the error on the cloud dataset on the irregular grid for sets of subsampling locations $\mathscr{S}_2, \mathscr{S}_3 and \mathscr{S}_4$, shown on Figure 5.25 and Figure 5.26. The proposed PING model is outperforming GCLSTMmlp models on the entire forecasting horizon. PING is also outperforming the GCLSTM model, although with a small margin. Thus, the NRMSE evolution of the proposed model is between 13% and 26%, no matter what type

Table 5.43: Accuracy of velocity direction estimation on irregular grid for advection and advection-diffusion datasets for PING model

		Strict		Relaxed	
		acceptable	non-acceptable	acceptable	non-acceptable
		$\phi < 30°$	$60° < \phi < 90°$	$\phi < 45°$	$\phi > 45°$
Advection dataset	$\mathscr{S}_2$	42%	23%	61%	39%
	$\mathscr{S}_3$	42%	23%	63%	37%
	$\mathscr{S}_4$	46%	20%	65%	35%
Advection-diffusion dataset	$\mathscr{S}_2$	44%	23%	61%	39%
	$\mathscr{S}_3$	44%	23%	63%	37%
	$\mathscr{S}_4$	40%	26%	60%	40%



(a)



(b)

Figure 5.22: The evolution of NRMSE of PING, GCLSTMmlp and GCLSTM models on the different irregular subsampling patterns for synthetic advection-diffusion dataset. a) NRMSE for subsampling set $\mathscr{S}_3$. b) NRMSE for subsampling set $\mathscr{S}_4$.

of spatial distribution we use. The NRMSE values for the subsampled dataset on an irregular grid are in the same range as NRMSE on the regular grid for the PING model. Since the same error range is reported on the regular and irregular datasets for both cloud concentration index, only advection and advection-diffusion datasets, which indicates the robustness in

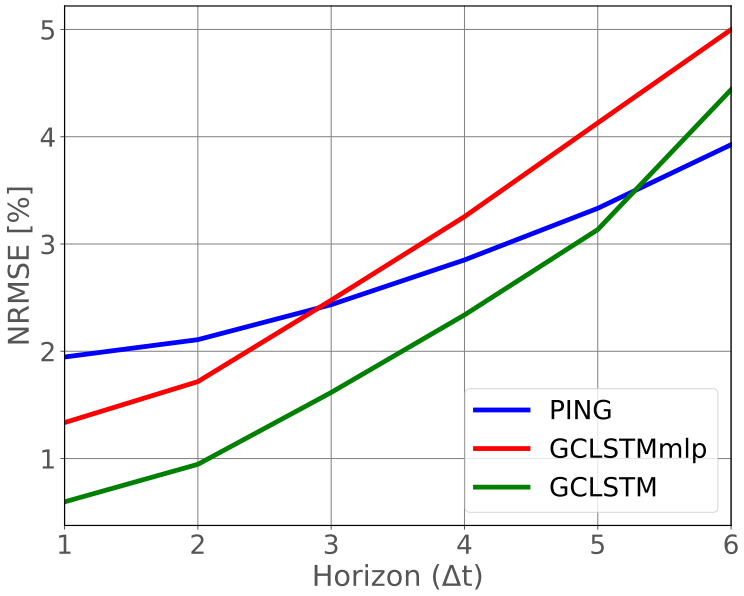


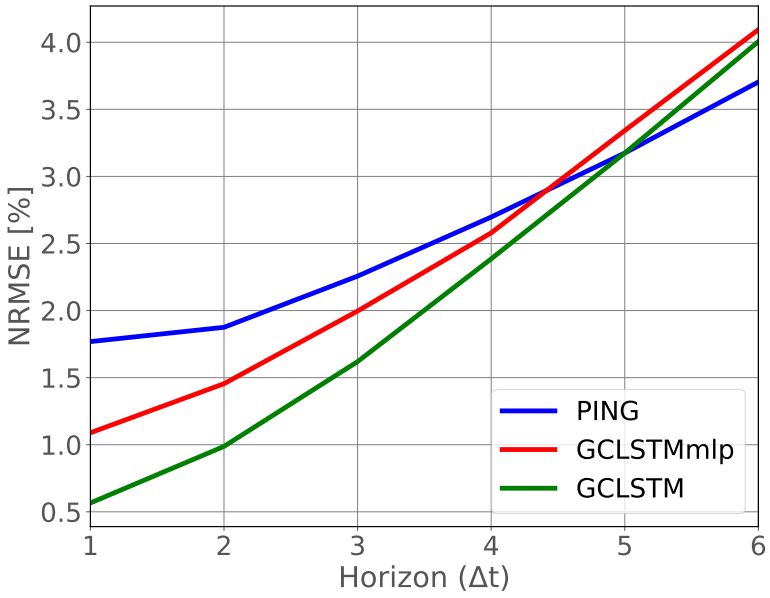
Figure 5.23

terms of the different grid structures, as well as the generalization capabilities. Opposite from the synthetic datasets, where the NRMSE values in the forecasting horizon are between 2% and 6.5%, the PING has higher error on the cloud concentration index dataset, showing the limitation of the model in capturing underlying dynamics, regardless of the grid type.

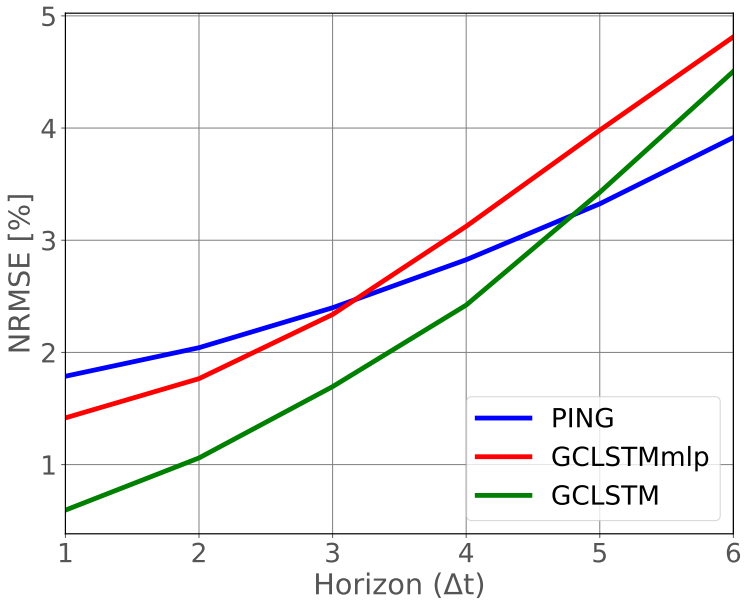
Finally, we examine the accuracy of the sea surface temperature dataset on the irregular grid for graph instances $\mathscr{G}_2, \mathscr{G}_3, \mathscr{G}_4$ and $\mathscr{G}_5$ on Figure 5.27 and Figure 5.28. The proposed model outperforms the GCLSTM model on the SST dataset from three to six horizon values of prediction, while it outperforms GCLSTMmlp from the second day ahead forecast. This dataset is very smooth and diffusive, which benefits both models (PING and GCLSTM). They exhibit similar behaviour in terms of the low predicting error. The type of spatial distribution we use has a small or no effect on the prediction accuracy of the PING model. These NRMSE values are in the same range as NRMSE on the regular grid, highlighting the model's generalization capabilities and efficiency in terms of capturing the underlying dynamics on different types of the grid and its spatial setting.

Analysis of the properties of velocity field and its impact on the model

We analyse the properties of the velocity fields in both synthetic datasets on the regular grid since the PING model has shown similar accuracy in terms of the model's prediction values across the whole forecasting horizon and velocity direction estimation, for both regular grid and four different inhomogeneous subsampling patterns. The main goal is to understand the limitations of the model better. To this end, we investigate the effect of spatial and temporal smoothness of the velocity field on the accuracy of velocity direction estimations.



(a)



(b)

Figure 5.24: The evolution of NRMSE of PING, GCLSTMmlp and GCLSTM models on the different irregular subsampling patterns for synthetic advection dataset. a) NRMSE for subsampling set $\mathscr{S}_3$. b) NRMSE for subsampling set $\mathscr{S}_4$.

We focus on the vector field's spatial smoothness and consider the vector field's divergence, the vector magnitude's gradient, and the velocity field's curl. Therefore, we first compute the

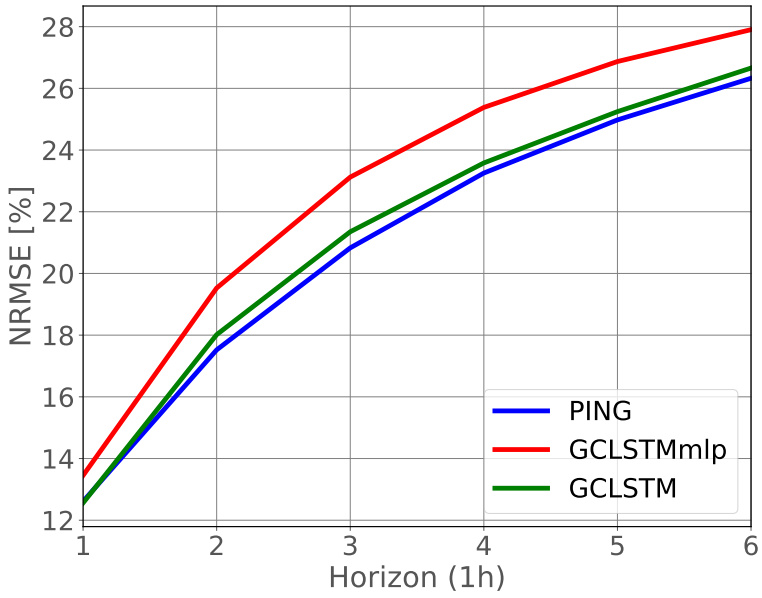


Figure 5.25: The evolution of NRMSE of PING, GCLSTMmlp and GCLSTM models on the graphs for cloud dataset for subsampling set $\mathscr{S}_2$.

divergence of the vector field, which is in the Cartesian coordinates given by:

$$\nabla \cdot \mathbf{u} = \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y}, \tag{5.30}$$

and then calculate the mean value across all nodes. As a second metric, we consider the magnitude of the gradient of the magnitude of each vector. This metric yields the change of the field in each point (node):

$$\nabla |\mathbf{u}| = \sqrt{\left(\frac{\partial \mathbf{u}}{\partial x}\right)^2 + \left(\frac{\partial \mathbf{u}}{\partial y}\right)^2} \tag{5.31}$$

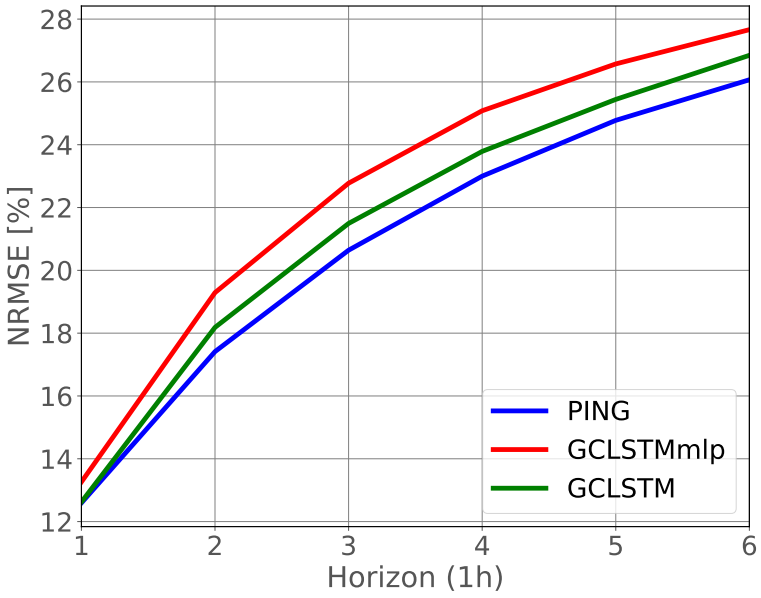
The mean value has been taken as the final result of this metric.

We considered a scalar curl as a third metric for the spatial smoothness of the vector field. It represents a measure of the rotational aspect of a vector field at each time point:

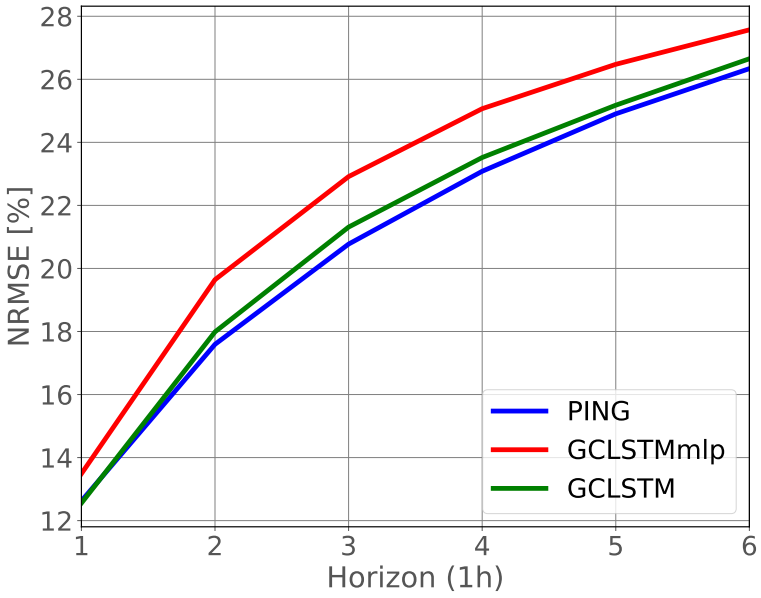
$$curl(\mathbf{u}) = \frac{\partial u_y}{\partial x} - \frac{\partial u_x}{\partial y}. \tag{5.32}$$

To define the temporal smoothness of the vector field, we use a single metric, which represents the difference between consecutive vector fields:

$$\Delta \mathbf{u}^{t,t-1} = \mathbf{u}^t - \mathbf{u}^{t-1}. \tag{5.33}$$



(a)



(b)

Figure 5.26: The evolution of NRMSE of PING, GCLSTMmlp and GCLSTM models on the graphs for cloud dataset. a) NRMSE for subsampling set $\mathscr{S}_3$. b) NRMSE for subsampling set $\mathscr{S}_4$.

One of the key components in the velocity estimation, Equation 5.17, represents the spatial and temporal difference between the particle concentration. Thus, we will calculate the spatial

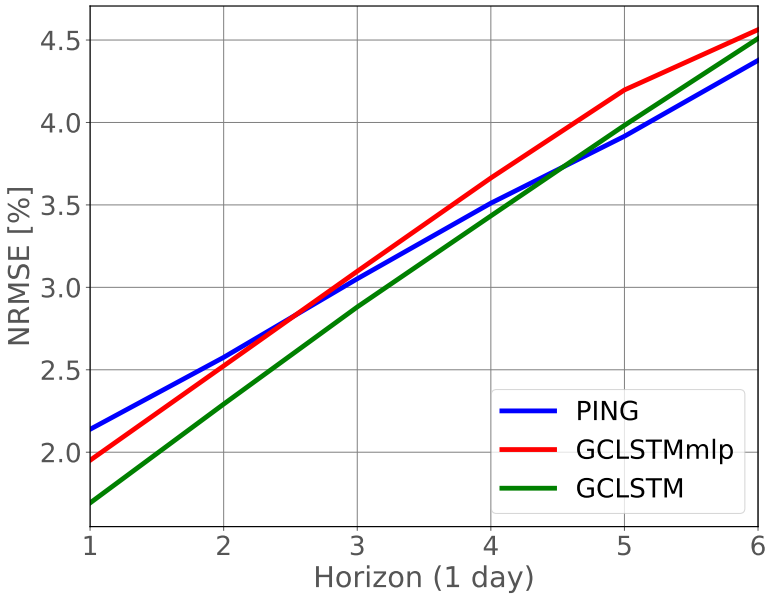


Figure 5.27: The evolution of NRMSE of PING, GCLSTMmlp and GCLSTM models on the graphs for sea surface temperature dataset for subsampling set $\mathscr{S}_2$.

smoothness between the node v_i and its eight closest neighbours v_j:

$$\Delta \mathbf{C}_{ij} = \mathbf{C}_i - \mathbf{C}_j \tag{5.34}$$

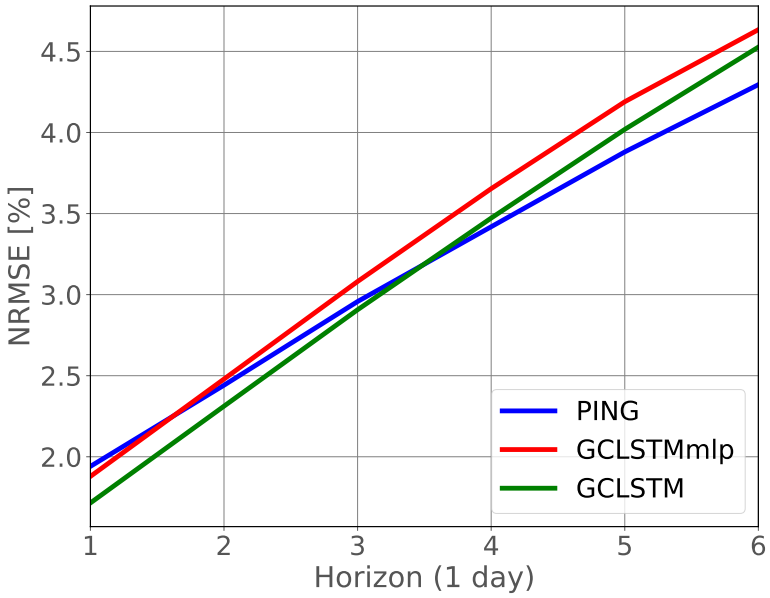
On a regular grid, the eight closest neighbours represent the adjacent nodes. It shows the rate of intensity change across the grid. Furthermore, we also define temporal smoothness as the difference between the consecutive particle values:

$$\Delta \mathbf{C}^{t,t-1} = \mathbf{C}^t - \mathbf{C}^{t-1} \tag{5.35}$$

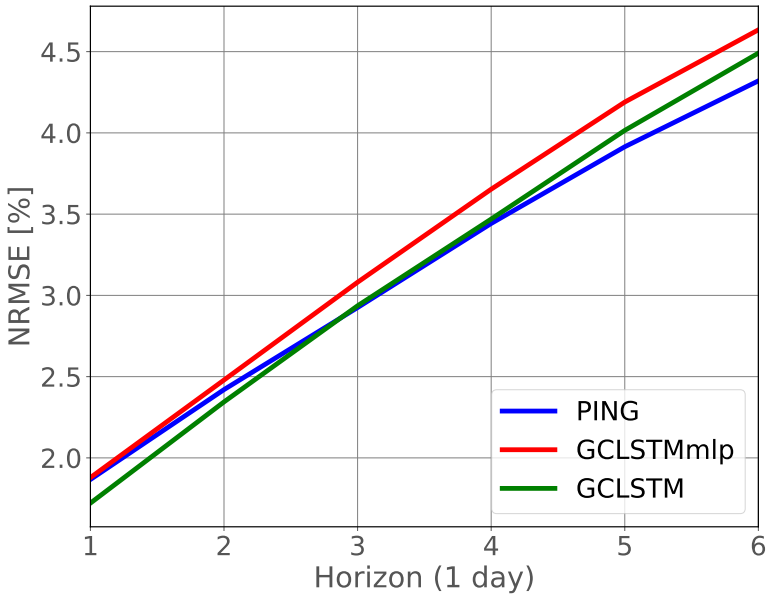
It shows how much the dataset changes between two time steps. All the above defined terms could be also computed on the irregular grid, as shown in Pérez et al. (2005).

The results for both synthetic datasets, evaluated using the above metrics, are shown in Table 5.44. Interestingly, the divergence of the velocity field $\nabla \cdot \mathbf{u}$ has the same value in both datasets and the absolute values of the magnitude of curl $curl(\mathbf{u})$ are also very close to each other. However, the curl values have opposite signs. The positive sign on a purely advective dataset suggests that the vector field is circulating counter-clockwise. The advection-diffusion dataset has an opposite direction, clockwise. These two metrics are near zero, indicating that the velocity field has low rotation.

On the other hand, the value of the gradient of velocity magnitude $\nabla|\mathbf{u}|$ is two times larger on an advective-diffusive dataset compared to purely advective. Therefore, it implies that the advection-diffusion dataset has more rapid changes in vector magnitudes than the advective



(a)



(b)

Figure 5.28: The evolution of NRMSE of PING, GCLSTMmlp and GCLSTM models on the graphs for sea surface temperature dataset. a) NRMSE for subsampling set $\mathscr{S}_3$. b) NRMSE for subsampling set $\mathscr{S}_4$.

dataset. Nevertheless, both values are small, indicating that the velocity field is spatially smooth. Finally, we will compare the temporal smoothness of velocity fields $\Delta\mathbf{u}^{t,t-1}$ between the given datasets. The advection-diffusion dataset has a velocity field that is two times more

smooth than the advective dataset. All calculated values and metrics show the average metric value of pixel per time unit (per time step).

As already defined in Equation 5.17, two key aspects in estimating the velocity in an unsupervised manner are spatial difference $\Delta \mathbf{C}_{ij}$ and temporal difference $\Delta \mathbf{C}^{t,t-1}$ in the concentration of the particles. From Table 5.44, we can see that spatially, these two datasets have spatially smooth concentration values across each time step. However, the temporal change of concentration values is four times higher in purely advective dataset. From previous analyses, the two used datasets differ the most in temporal change of concentration values. This is also related to the forecasting error, shown in Figure 5.5. Purely advective dataset has lower forecasting error range, where NRMSE for six forecasting values are between 1.4% and 3.5%, compared to the forecasting error of the advection-diffusion dataset, where the forecasting NRMSE is between 2.7% and 5.4% for six prediction values. The model benefits from the temporarily less smooth signals.

The advection and advection-diffusion datasets differ also in a temporal change of velocity field, which is higher in the case of the advective dataset. While both datasets are smooth in space, in terms of concentration and velocity field, the advective dataset is less smooth in time. This indicates that the lower accuracy of the velocity estimation on the advective-diffusive dataset on the regular domain comes from the high temporal and spatial smoothness.

Thus, in a scenario where both the velocity field and particle concentration values are smooth spatially, the PING model relies more on the temporal change to estimate velocity values, see Equation 5.17. If the temporal difference is also small due to high temporal smoothness, it becomes difficult for the model to estimate the velocities accurately. This is likely why PING model on the same advection-diffusion dataset, in the setting on the irregular domain, has higher velocity estimation accuracy (see Table 5.41). Even though the temporal change is still smooth, the spatial differences are less smooth on an irregular grid than on a regular grid. Therefore, in the scenario when dataset is smooth temporarily, it is easier for the PING model to capture the flow dynamics when the dataset is not smooth spatially. Consequently, improvement in capturing the flow dynamics is also reflected in the similar prediction error on the regular grid advective dataset (Figure 5.5 b)) and on the graph setting for the same dataset (Figure 5.6 b)), even though the irregularly sampled grid has less information.

In order to further test the failure points of our model, we have created an additional purely advective dataset, named Advection dataset 2 in Table 5.44. The previously evaluated advective dataset is renamed Advection dataset 1 in further discussion. We have shown that the model is able to estimate the velocities in the case of spatially and temporarily smooth datasets. In the newly added dataset, advective dataset 2, we introduce more rotations in the vector field and more rapid changes in the velocity field between consecutive time steps. The PING and GCLSTM models are trained and evaluated on the advection dataset 2 for the forecasting horizon of six values. The PING model is on par with the baseline GCLSTM regarding prediction accuracy, see Figure 5.29. Then, we check if the rapid changes in the

Table 5.44: Accuracy of PING model in terms of velocity direction estimation and spatial and temporal smoothness of synthetic datasets

	Regular grid strict criteria	Irregular grid strict criteria	$\nabla \cdot \mathbf{u}$	$curl(\mathbf{u})$	$\nabla \lvert \mathbf{u} \rvert$	$\Delta \mathbf{u}^{t,t-1}$	$\Delta \mathbf{C}_{ij}$	$\Delta \mathbf{C}^{t,t-1}$
Advection dataset 1	61%	43%	0.084	0.020	0.048	0.79	0.18	1.48
Advection-diffusion dataset	40%	44%	0.084	0.019	0.084	0.36	0.12	0.48
Advection dataset 2	40%	40%	0.0032	0.057	0.1	8.3	0.11	3.15

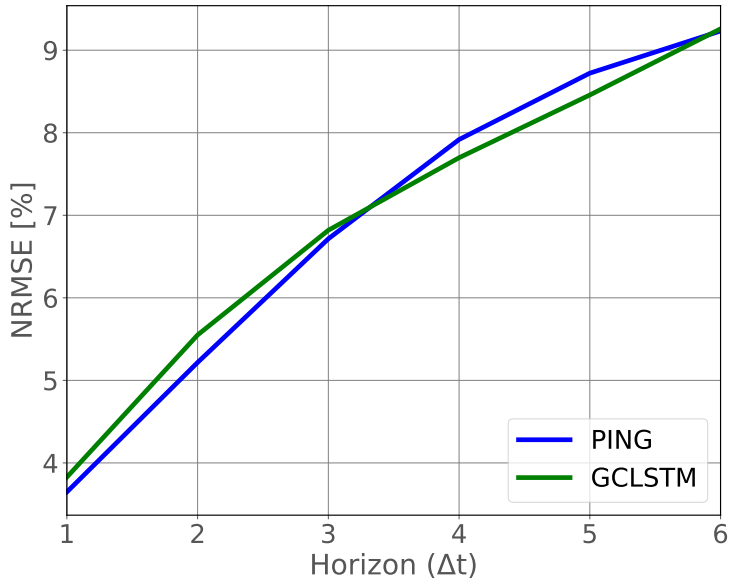


Figure 5.29: Evolution of the NRMSE between PING and GCLSTM models for six hours ahead for advection dataset 2. Solid line shows the median value among all nodes.

velocity field impact the model accuracy. We observe the error of estimating the velocity field on both regular and irregular grids and show the results in Table 5.44. PING leads to a lower accuracy of the estimated velocity direction compared to the other synthetic datasets on both regular and irregular grids. The irregular domain of advection dataset 2 is subsampled in the same manner as the other two synthetic datasets, using the spatial subsampling patterns.

Finally, we evaluate the velocity estimations concerning the spatial and temporal smoothness of the particles' concentration and velocity field. The results are shown in Table 5.44. Although the divergence of the field is lower compared to the two previous datasets, the magnitude of the

curl is three times larger than in the advective dataset 1 and the advective-diffusive dataset. In addition, the magnitudes of the velocity gradient are changing faster than the initial synthetic datasets, indicating a spatially less smooth velocity field of advection dataset 2. Oppositely, the advection dataset 2 presents with a higher spatial smoothness of the concentration, compared to other datasets. The spatial smoothness of the concentrations on advection dataset 2 is similar to the spatial concentration smoothness on the advection-diffusion dataset. Even though the model should benefit from higher temporal differences in the concentration of the particles, in the case of spatially smooth concentration change, it presents with the highest forecasting error on the regular grid so far, comparing the NRMSE on Figure 5.29 and Figure 5.5.

In order to investigate the origin of the high error, we focus on the velocity field's temporal smoothness and particle concentration's temporal smoothness. Both of these metrics are substantially higher in the advective dataset 2 compared to the advective dataset 1 and the advective-diffusive dataset. The temporal difference of particles' concentration in the advective dataset 2 is two times higher than in the advective dataset 1 and 6.5 times higher than in the advective-diffusive dataset. On top of this, the temporal difference between the velocity fields is 10.5 higher in the advective dataset 2 than in the advective dataset 1. The difference is even higher between advection dataset 2 and the advection-diffusion dataset, which is 23 times. The advection dataset 2 has the highest prediction error values among the three synthetic datasets. Even though the model should benefit from higher temporal concentration change, due to very fast velocity changes the PING model is not able to estimate the flow dynamics with the high accuracy (40% of acceptable estimations using the strict metric) when the velocity field is changing very fast temporarily. This is reflected in the accuracy of the prediction, which has the highest NRMSE error across all compared datasets, with the range of error between 3.7% and 9.3% for the prediction horizon of six values.

The experiments revealed the possible link between PING model's ability to estimate the velocity flows and the prediction accuracy. On the regular grid, datasets with higher accuracy of the velocity direction, Table 5.44, also exhibit higher accuracy of the concentration prediction, which could be seen from comparing the error ranges on Figure 5.29 and Figure 5.5. Therefore, understanding what affects the velocity estimation accuracy is very important. To that end we inspect the accuracy of the velocity estimations in Table 5.44. We observe that the dataset, characterized by more significant rotations and velocity field variations, advection dataset 2, presented more challenges in velocity direction estimation. In the setting with fast velocity field changes between two consecutive temporal steps, on the advection dataset, the PING has lower accuracy of the velocity estimations on both regular and irregular grids compared to the datasets with more temporally smooth changes between the velocity fields (advection dataset 1 and advection-diffusion dataset). When two datasets with similar rotation and velocity field variations on the regular grid are compared, advection dataset 1 and advection-diffusion dataset, the dataset with higher temporal change of the concentration of particles between consecutive time steps showed higher accuracy of the velocity direction estimation, which is advection dataset 1. The main reason is that when estimating the velocity values, the

model relies on the temporal change between the consecutive time steps; see Equation 5.17. Thus, the model will have lower velocity accuracy in the case of temporarily slow-changing concentration of the particles and in the case of a highly diffusive process.

5.5 Conclusion

A novel method, PING, is introduced for forecasting the future particle concentration values in the advection-diffusion-based processes. It was evaluated on sea surface temperature, cloud concentration index, and two synthetic datasets designed to simulate advection and advection-diffusion processes. The performance of the proposed model is compared against baselines on the datasets that reside on both regular and irregular grids. Furthermore, PING was combined with encoder-decoder model and evaluated on real PV power generation data. PING outperformed the benchmark models for the last three forecasting horizon values for the sea surface temperature dataset and synthetic datasets. However, the model outperforms the benchmark model on the cloud datasets for the entire forecasting horizon. On the PV power generation dataset, PING combined with GCLSTM outperformed all benchmarks across the whole forecasting horizon.

Currently the forecasting part of the PING model is a single MLP, and as such it is not meant to be used for prediction sequences with many time steps in the forecasting horizon. However, the proposed model could be viewed as an analysis, or an encoder block in the larger architecture for forecasting the dynamics of advection-diffusion processes. Since PING model analysis the past data while capturing the dynamics and extracting meaningful features, it would be an interesting research direction to include the estimated velocities in the future through propagation of dynamics via recurrent models. While we have addressed the constraints of the proposed model concerning the velocity field's smoothness and particle concentration under various subsampling strategies, future studies should investigate the model's bounds regarding the minimal number of nodes or the spatial distance between the subsampled locations needed to estimate the velocity directions accurately.

6 Conclusion and future directions

6.1 Conclusion

The thesis developed and evaluated new deterministic forecasting methods that will improve grid trading and grid congestion management on intra-day horizon to account for dynamical changes of photovoltaic power production due to weather variability. Hence, this will lead to increasing penetration of PV power resources in the power grid. We have used a graph signal processing perspective throughout the thesis. This perspective allowed us to address the challenges of capturing both spatial and temporal correlations and cloud dynamics within the PV power data with high resolution, using only ground-based PV power data.

The challenge of forecasting spatio-temporal photovoltaic power production with high spatial and temporal resolution, using only ground-based PV power data, is one of the first challenges in forecasting intra-day PV power production. We have shown that the graph-based encoder decoder models are able to forecast future photovoltaic power production with higher accuracy compared to the state-of-the-art models for multi-site photovoltaic prediction. On the top of that, we have shown that the proposed methods outperform single-site state-of-the-art forecasting methods that use numerical weather predictions and photovoltaic power production data as inputs for horizon of five hours ahead. However, the number of PV stations taken into in the proposed models were limited to a few closest neighbours because of the increase in computational complexity. Since it is expected that further away nodes might be important predictors if advection is dominant in the regional cloud dynamics at a specific time, a model that is not restricted to small amount of neighbours is needed.

Although the graph-based machine learning models have shown the ability to improve the accuracy of the photovoltaic intra-day forecast, compared to the state of the art, it can not capture cloud dynamics during the morning and it is difficult to interpret. Therefore, the following challenge in the intra-day PV power forecasting is modelling an interpretable machine learning architecture that will allow us to understand on which PV stations model focuses when making the forecast for short-, medium- and long-term part of intra-day forecast. Furthermore, the idea was to create a model which is able to better capture cloud dynamics and to

consider stations that are further away, not only predefine set of stations. We have developed interpretable model that was able to forecast the signal with the shape closer to the ground truth than state-of-the-art models, indicating its ability to better capture cloud dynamics. This method was better at capturing cloud dynamics during the morning of cloudy days, which was not possible with the state of the art. The proposed model is more interpretable, and as such more suitable for the prediction of multi-site time series driven by physical phenomena, such as photovoltaic and wind power forecasting.

Even though an interpretable model for photovoltaic power forecasting is developed, this model still might violate the physical laws of advection-diffusion, which is guiding the cloud dynamics. The known physical knowledge on the underlying physical process should be included when creating the model for PV power production, due to high correlation with cloud dynamics and the PV power production. The last challenge that was addressed in the thesis is creating a physically informed machine learning model that is able to capture the cloud dynamics. We have developed a physically informed model which is able to capture the past dynamics by estimating the velocities in an unsupervised manner. The model was evaluated on the cloud concentration index to show ability to capture cloud dynamics. The proposed model outperformed other graph-based model for time-series forecasting for six horizon values of the forecast on the cloud concentration index dataset. We have investigated the generalization capabilities of the model in terms of modelling different physical phenomena by evaluating it on different advection-diffusion datasets. The proposed model is either on par or outperforms benchmark models for various forecasting tasks where dynamics is guided by advection-diffusion processes. Furthermore, when combined with encoder decoder GCLSTM model, it outperforms all benchmarks for predicting the PV power production.

6.2 Future work

The proposed machine learning models, as well as state of the art, are usually proposed for the complete and clean datasets, without any missing or noisy data. Their performance can be significantly reduced with larger gaps of the missing observations due to communication issues from sensory network or maintenance of the solar panels or inverters. Furthermore, in the case of the newly installed PV stations, there is a lack of historical data, and it takes a lot of time to collect sufficient data in order to make an accurate forecast for newly installed station. It might take large amount of time before this station can be integrated into the power grid. Therefore, investigating the robustness of the proposed models is the first research direction, that we proposed. Since the many real-world datasets include missing data, as well as addition of the new stations, it is crucial to investigate the adaptability of the proposed models to incomplete datasets. The first steps towards addressing the robustness and extension to probabilistic forecasts are being taken in the works of Carrillo et al. (2023).

The proposed models are developed for handling a dataset of hundreds of the photovoltaic stations. As we have shown, there are benefits of accessing the further away nodes, although

this increases the complexity of the model, and it is a quadratic complexity in the case of the proposed interpretable model. For the physically-informed model, although the memory complexity is linear, it can quickly escalates with small increase in the number of the neighbouring PV stations that are considered when making the forecast. Additionally, the memory also increases with the growth of the input size, as well as the number of the forecasting steps. Hence, another research direction that needs to be addressed, if these models are to be integrated in the real operating conditions in the future smart grid, is the scalability of the models.

In our work we have investigated the impact of the number of sunny, cloudy and variable days on the forecasting accuracy. We have investigated the impact that different distances between the photovoltaic power stations have on the accuracy of the forecast for the GCSLTM, GCTrafo and interpretable TSM-GAT models. However, an in-depth research is needed to confirm the initial findings on a denser dataset, containing a higher number of homogeneously spread nodes. Moreover, a model's bounds regarding the conditions when models start to fail is needed.

Physically-informed model offers insights into the dynamics of the historical, input data. One future step is using this model for prediction of rare weather events, which requires not only estimation, but also propagation of the future dynamics. We have partially addressed the dynamics propagation with the combination of PING and GCLSTM, where the future cloud concentration values obtained in PING are used as input to the GCLSTM decoder. In order to predict rear weather events, PING could be used as an encoder building block in a deep learning model, where past velocity features are estimated. Then estimated velocities could be used to initialise the decoder block in order to make predictions.

A Appendix

A.1 Hyperparameters

We present here the hyperparameters used to train GCLSTM and GCTrafo networks, proposed in Chapter 3. We also describe hyperparameters of the EDLSTM and STCNN models, which were introduced as baseline models in the same chapter. Whereas the hyperparameters used to train baseline models, STAR and additional details on STCNN parameters, could be found in works of Jeong and Kim (2019) and Carrillo et al. (2020), respectively. Then we describe in detail the hyperparameters of the TSM-GAT, and the baseline models SVR and SARIMAX, presented in Chapter 4. Finally, we describe hyperparameters of the PING models and baseline model GCLSTMmlp, which we compare in the Chapter 5.

Hyperparameters for the GCLSTM and GCTrafo models

In Chapter 3 we train the GCLSTM, GCTrafo, STCNN and EDLSTM with hyperparameters presented in Table A.11. The number of hidden dimensions lat in the encoder and decoder cells of the GCLSTM network (see Section 3.3.1) were equal to 32. The size of the MLP at the end of the GCLSTM decoder was equal to [8, 48, 48]. For the GCTrafo, the following hyperparameters were chosen: the 1D-convolutional kernel was of size 4, the encoder and decoder convolutional latent spaces were of size 8, and 8 attention heads were used. The STCNN architecture had three 2D convolutional layers, with channel sizes of [128, 64, 32] and a kernel size of 11. Batch normalization and max pooling were applied after each convolutional layer. The single-site Encoder Decoder LSTM (EDLSTM) had a latent representation size of 64 and the decoder was followed by a MLP of size [64, 32]. STCNN, GCLSTM and GCTrafo models were trained with stochastic gradient descent and Adam optimizer, without regularization. EDLSTM was trained with dropout as regularization. Finally, the STAR model used 3 hours of past time steps to forecast the PV production over the 6 hours horizon.

Table A.11: Table of hyperparameters in GCLSTM, GCTrafo, EDLSTM and STCNN trained for PV production datasets

Models	Iterations (real/synthetic)	Batch size (real/synthetic)	Past time steps - M	k-nearest neighbours (graph construction)	Order of Chebyshev polynomial	Learning rate / Dropout rate
GCTrafo	70 000	64	16	24	2	1e-4 / -
GCLSTM	50 000	64	16	15	4	1e-4 / -
STCNN	6000 / 10 000	128 /64	72	-	-	1e-4 / -
EDLSTM	30 000	128	16	-	-	1e-4 / 0.05

Hyperparameters for the TSM-GAT model

In Chapter 4, baseline models STCNN and STAR use only PV power production data as input to the model, whereas GCLSTM, GCTrafo, TSM-GAT and SARIMAX use longitude and latitude as input, to create graph and compute clear-sky irradiance, in addition to PV power production. SVR uses NWP and clear-sky data as input, whereas EDLSTM uses PV power production in addition to NWP data. SARIMAX uses a seasonality order which is equal to the number of days to which data is fitted. The number of past lags that is fed to the model is 12, which corresponds 3 hours of data. The difference operator order is 1 and moving average operator is of order 3. EDLSTM was trained with a dropout rate of 0.05 as regularization for 30 000 iterations. The number of past lags is 16, which is 4 hours of data, with batch size 128 and learning rate $1e-4$. As far as proposed TSM-GAT is concerned, number of hidden features in temporal attention is reduced from $f_{in} = 14$ to $f' = 8$, and then in spatial attention to $f'' = 4$ features per temporal window and per node. This hyperparameter was decided arbitrary, as well as the number of the closest neighbours, which is changing between overlapping windows as already described. The model was trained with batch size 32 and for 600 000 iterations for the real dataset and 680 000 iterations for the synthetic dataset. Layer normalization is performed after spatial attention. The model was trained with stochastic gradient descent and Adam optimizer and learning rate 1e-4. The size of hidden layer in MLP used for final prediction is 64. Model TS-multi-head-GAT has the same sizes of the temporal weights as TSM-GAT. The number of closest neighbours for TS-multi-head-GAT is fixed to 90 for all attention heads. The number of iterations is the same as in TSM-GAT. Three heads were used and number of the weights in the linear layer is 3 times higher than in TSM-GAT, as already discussed.

Hyperparameters for the PING model

In Chapter 5, the proposed PING model uses the same hyperparameters across all datasets. The number of the training iterations and regularisation coefficients are the only hyperparameters that are not same across all datasets. They are shown in Table A.12. The length of the input sequences for all datasets is 7. The model was trained with stochastic gradient descent and Adam optimizer and learning rate $1e-4$. The model is trained with the batch size of 16. We first describe different hyperparameters in the flow estimation blocks. The feature sizes in the block for velocity flow estimations are $F_{in} = 5, F' = 2, F'' = 16, F_{out} = 8$, whereas the feature

sizes in the acceleration flow estimation block are $F_{in}^{acc} = 26, F^{\prime acc} = 2, F^{\prime\prime acc} = 16, F_{out}^{acc} = 8$. The sizes of MLP layer in velocity flow estimation block are [8, 16, 8], and in acceleration flow estimation block they are [8, 16, 16]. Convolutional kernels size in flow estimation blocks is 3. The rolling mean in the flow estimation block is calculated for every 4 feature values and the shift of size 2 is used. In all experiments the value of regularization coefficient λ_1 is set to 0.1 and λ_2 is set to 0.05.

Then we describe the hyperparameters in the flow attention block, needed for edge calculation. The number of neighbours taken into account in the flow attention block for velocity attention calculation and embedded concentration attention is $S = S_{emb} = 8$, while in the attention flow block for acceleration attention calculation it is $S_{acc} = 24$. The size of topological embedding in this block is $s^{emb} = 4$. The MLP sizes in all three attention flow blocks are [16, 16, 16]. In the flow processor block the size of the latent space is $q = 8$ and the number of forecasting steps in all datasets is $H = 6$. The MLP size in the processor block is [32, 1] in all datasets, except in the PV power production. In the PV power production dataset, the size of the last MLP layer is [32,16,1]. Baseline GCLSTMmlp model has the exactly same values of hyperparameters as the encoder of GCLSTM, described in the Table A.11, with the MLP size [8, 48, 48] in the decoder. The PING, GCLSTM and GCLSTMmlp models in Chapter 5 are trained for different number of iterations, shown in Table A.12.

Table A.12: Table of hyperparameters used for training PING, GCLSTM and GCLSTMmlp

Dataset type	Grid type	Number of iterations PING	Number of Iterations GCLSTM	Number of iterations GCLSTMmlp
Cloud dataset	Regular grid	400000	80000	18000
	Irregular grids s1, s1, s3, s4	400000/ 400000/ 400000/400000	80000/80000/ 80000/80000	18000/ 20000/ 20000/18000
Sea surface temperature dataset	Regular grid	120000	80000	20000
	Irregular grids s1, s2, s3, s4	80000/ 80000/ 60000/40000	80000/80000/ 80000/60000	20000/ 20000/ 16000/16000
Synthetic advection dataset 1	Regular grid	500000	80000	20000
	Irregular grids	700000/ 600000/ 700000/500000	60000/80000/ 60000/60000	20000/ 18000/ 18000/16000
Synthetic advection – diffusion dataset	Regular grid	700000	80000	18000
	Irregular grids	650000/ 700000/ 700000/40000	80000/80000/ 60000/60000	18000/ 16000/ 16000/18000
PV power dataset	Irregular grid	140000	75000	20000
Synthetic advection dataset 2	Regular grid	700000	80000	-

Bibliography

Agoua, X. G., Girard, R., and Kariniotakis, G. (2018). Short-term spatio-temporal forecasting of photovoltaic power production. *IEEE Transactions on Sustainable Energy*, 9(2):538–546.

Andrychowicz, M., Espeholt, L., Li, D., Merchant, S., Merose, A., Zyda, F., Agrawal, S., and Kalchbrenner, N. (2023). Deep learning for day forecasts from sparse observations. *arXiv preprint arXiv:2306.06079*.

Antonanzas, J., Osorio, N., Escobar, R., Urraca, R., Martinez-de Pison, F. J., and Antonanzas-Torres, F. (2016). Review of photovoltaic power forecasting. *Solar Energy*, 136:78–111.

Atwood, J. and Towsley, D. (2016). Diffusion-convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29.

Azad, H. B., Mekhilef, S., and Ganapathy, V. G. (2014). Long-term wind speed forecasting and general pattern recognition using neural networks. *IEEE Transactions on Sustainable Energy*, 5(2):546–553.

Baan, J., ter Hoeve, M., Wees, M. V. D., Schuth, A., and Rijke, M. (2019). Understanding multi-head attention in abstractive summarization. *ArXiv*, abs/1911.03898.

Bahoura, M. (2019). Efficient fpga-based architecture of the overlap-add method for short-time fourier analysis/synthesis. *Electronics*, 8(12).

Belbute-Peres, F. D. A., Economon, T., and Kolter, Z. (2020). Combining differentiable pde solvers and graph neural networks for fluid flow prediction. In *International Conference on Machine Learning (ICML)*, pages 2402–2411. PMLR.

Benavides Cesar, L., Amaro e Silva, R., Manso Callejo, M. A., and Cira, C.-I. (2022). Review on spatio-temporal solar forecasting methods driven by in situ measurements or their combination with satellite and numerical weather prediction (nwp) estimates. *Energies*, 15(12):4341.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166.

Bi, K., Xie, L., Zhang, H., Chen, X., Gu, X., and Tian, Q. (2023). Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619:1–6.

Boegli, M., Pierro, M., Moser, D., and Alet, P.-J. (2018). Machine learning techniques for forecasting single-site PV production. In *34thEuropean Photovoltaic Solar Energy Conference and Exhibition (EUPVSEC)*.

Boussif, O., Bengio, Y., Benabbou, L., and Assouline, D. (2022). Magnet: Mesh agnostic n pde solver. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:31972–31985.

Bresson, X. and Laurent, T. (2017). Residual gated graph convnets. *ArXiv*, abs/1711.07553.

Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. (2014). Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR)*.

Buizza, R. (2019). Introduction to the special issue on "25 years of ensemble forecasting". *Quarterly Journal of the Royal Meteorological Society*, 145:1–11.

Carriere, T., Vernay, C., Pitaval, S., and Kariniotakis, G. (2020). A novel approach for seamless probabilistic photovoltaic power forecasting covering multiple time frames. *IEEE Transactions on Smart Grid*, 11(3):2281–2292.

Carrillo, R., Alet, P.-J., Müller, S., and Remund, J. (2022). A computationally light data-driven alternative to cloud-motion prediction for pv forecasting. In *8th World Conference on Photovoltaic Solar Energy Conversion, Milan, Italy*.

Carrillo, R., Schubnel, B., Langou, R., and Alet, P.-J. (2023). Dynamic graph machine learning for multi-site solar forecasting. In *40th European Photovoltaic Solar Energy Conference and Exhibition (EUPVSEC), Lisbon, Portugal*.

Carrillo, R. E., Leblanc, M., Schubnel, B., Langou, R., Topfel, C., and Alet, P.-J. (2020). High-resolution pv forecasting from imperfect data: A graph-based solution. *Energies*, 13(21):5763.

Cavalcante, L. and Bessa, R. J. (2017). Solar power forecasting with sparse vector autoregression structures. In *2017 IEEE Manchester PowerTech*, pages 1–6.

Cheng, L., Zang, H., Wei, Z., Ding, T., and Sun, G. (2022). Solar power prediction based on satellite measurements – a graphical learning method for tracking cloud motion. *IEEE Transactions on Power Systems*, 37(3):2335–2345.

Chiu, P.-H., Wong, J. C., Ooi, C., Dao, M. H., and Ong, Y.-S. (2022). Can-pinn: A fast physics-informed neural network based on coupled-automatic–numerical differentiation method. *Computer Methods in Applied Mechanics and Engineering*, 395:114909.

Chu, Y., Li, M., Coimbra, C., Feng, D., and Wang, H. (2021). Intra-hour irradiance forecasting techniques for solar power integration: A review. *Iscience*, 24(10).

Chu, Y., Urquhart, B., Gohari, S. M. I., Pedro, H., Kleissl, J., and Coimbra, C. (2015). Short-term reforecasting of power output from a 48 mwe solar pv plant. *Solar Energy*, 112:68–77.

Cordonnier, J.-B., Loukas, A., and Jaggi, M. (2020). Multi-head attention: Collaborate instead of concatenate. *ArXiv*, abs/2006.16362.

Cui, Z., Ke, R., Pu, Z., and Wang, Y. (2018). Deep bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction. *arXiv:1801.02143*.

Dai, Q., Huo, X., Hao, Y., and Yu, R. (2023). Spatio-temporal prediction for distributed pv generation system based on deep learning neural network model. *Frontiers in Energy Research*, 11:1204032.

Dairi, A., Harrou, F., and Sun, Y. (2021). A deep attention-driven model to forecast solar irradiance. *IEEE 19th International Conference on Industrial Informatics*.

De Bézenac, E., Pajot, A., and Gallinari, P. (2019). Deep learning for physical processes: Incorporating prior scientific knowledge. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124009.

Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Duvenaud, D., Maclaurin, D., Aguilera-Iparraguirre, J., Gómez-Bombarelli, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. (2015). Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28.

Farnebäck, G. (2003). Two-frame motion estimation based on polynomial expansion. In *Image Analysis: 13th Scandinavian Conference, SCIA 2003 Halmstad, Sweden, June 29–July 2, 2003 Proceedings 13*, pages 363–370. Springer.

Fortunato, M., Pfaff, T., Wirnsberger, P., Pritzel, A., and Battaglia, P. (2022). Multiscale mesh-graphnets. *2nd AI4Science Workshop at the 39 th International Conference on Machine Learning (ICML)*.

Gao, H., Zahr, M. J., and Wang, J.-X. (2022). Physics-informed graph neural galerkin networks: A unified framework for solving pde-governed forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 390:114502.

Gao, M., Li, J., Hong, F., and Long, D. (2019). Short-term forecasting of power production in a large-scale photovoltaic plant based on lstm. *Applied Sciences*, 9(15).

Ghaderi, A., Sanandaji, B. M., and Ghaderi, F. (2017). Deep forecast: Deep learning-based spatio-temporal forecasting. *ArXiv*, abs/1707.08110.

Gonçalves, C., Pinson, P., and Bessa, R. J. (2021). Towards data markets in renewable energy forecasting. *IEEE Transactions on Sustainable Energy*, 12(1):533–542.

Guo, S., Lin, Y., Feng, N., Song, C., and Wan, H. (2019). Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):922–929.

Guo, X., He, J., Wang, B., and Wu, J. (2022). Prediction of sea surface temperature by combining interdimensional and self-attention with neural networks. *Remote Sensing*, 14(19).

Hamberg, L. (2021). Photovoltaic system performance forecasting using lstm neural networks. Master's thesis, Uppsala University, Department of Information Technology.

Hammond, D. K., Vandergheynst, P., and Gribonval, R. (2011). Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150.

Harrou, F., Kadri, F., and Sun, Y. (2020). Forecasting of photovoltaic solar power production using lstm approach. In Harrou, F. and Sun, Y., editors, *Advanced Statistical Modeling, Forecasting, and Fault Detection in Renewable Energy Systems*, chapter 1. IntechOpen, Rijeka.

He, J., Abueidda, D., Koric, S., and Jasiuk, I. (2023). On the use of graph neural networks and shape-function-based gradient computation in the deep energy method. *International Journal for Numerical Methods in Engineering*, 124(4):864–879.

Hein, M., Audibert, J.-Y., and Luxburg, U. v. (2007). Graph laplacians and their convergence on random neighborhood graphs. *Journal of Machine Learning Research*, 8(6).

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Holl, P., Koltun, V., and Thuerey, N. (2020). Learning to control pdes with differentiable physics. *International Conference on Learning Representations (ICLR)*.

Hossain, M., Mekhilef, S., Afifi, F., Halabi, L., Olatomiwa, L., Seyedmahmoudian, M., Horan, B., and Stojcevski, A. (2018). Application of the hybrid anfis models for long term wind power density prediction with extrapolation capability. *PLOS ONE*, 13:e0193772.

Huang, J. and Perry, M. (2015). A semi-empirical approach using gradient boosting and -nearest neighbors regression for gefcom2014 probabilistic solar power forecasting. *International Journal of Forecasting*, 32.

Huertas Tato, J. and Centeno Brito, M. (2018). Using smart persistence and random forests to predict photovoltaic energy production. *Energies*, 12(1):100.

Hummon, M., Ibanez, E., Brinkman, G., and Lew, D. (2012). Sub-hour solar data for power system modeling from static spatial variability analysis: Preprint. *Second International Workshop on Integration of Solar Power in Power Systems*.

Iheanetu, K. J. (2022). Solar photovoltaic power forecasting: A review. *Sustainability*, 14(24):17005.

Ineichen, P. (2006). Comparison of eight clear sky broadband models against 16 independent data banks. *Solar Energy*, 80:468–478.

Jang, H. S., Bae, K. Y., Park, H.-S., and Sung, D. K. (2016). Solar power prediction based on satellite images and support vector machine. *IEEE Transactions on Sustainable Energy*, 7(3):1255–1263.

Jeong, J. and Kim, H. (2019). Multi-site photovoltaic forecasting exploiting space-time convolutional neural network. *Energies*, 12(23):4490.

Jiménez-Luna, J., Grisoni, F., and Schneider, G. (2020). Drug discovery with explainable artificial intelligence. *Nature Machine Intelligence*, 2(10):573–584.

Kashinath, K., Mustafa, M., Albert, A., Wu, J., Jiang, C., Esmaeilzadeh, S., Azizzadenesheli, K., Wang, R., Chattopadhyay, A., Singh, A., et al. (2021). Physics-informed machine learning: case studies for weather and climate modelling. *Philosophical Transactions of the Royal Society A*, 379(2194):20200093.

Keisler, R. (2022). Forecasting global weather with graph neural networks. *arXiv preprint arXiv:2202.07575*.

Khaled, A., Elsir, A. M. T., and Shen, Y. (2022). Tfgan: Traffic forecasting using generative adversarial network with multi-graph convolutional network. *Knowledge-Based Systems*, 249:108990.

Kharlova, E., May, D., and Musílek, P. (2020). Forecasting photovoltaic power production using a deep learning sequence to sequence model with attention. *International Joint Conference on Neural Networks (IJCNN)*, pages 1–7.

Khodayar, M., Mohammadi, S., Khodayar, M. E., Wang, J., and Liu, G. (2019). Convolutional graph autoencoder: A generative deep neural network for probabilistic spatio-temporal solar irradiance forecasting. *IEEE Transactions on Sustainable Energy*, 11(2):571–583.

Khodayar, M. and Wang, J. (2019). Spatio-temporal graph deep neural network for short-term wind speed forecasting. *IEEE Transactions on Sustainable Energy*, 10(2):670–681.

Koochali, A., Dengel, A., and Ahmed, S. (2021). If you like it, gan it. probabilistic multivariate times series forecast with gan. *Engineering Proceedings*, 5(1):40.

Kriege, N. M., Johansson, F. D., and Morris, C. (2020). A survey on graph kernels. *Applied Network Science*, 5(1):1–42.

Krivec, T., Kocijan, J., Perne, M., Grašic, B., Božnar, M. Z., and Mlakar, P. (2021). Data-driven method for the improving forecasts of local weather dynamics. *Engineering Applications of Artificial Intelligence*, 105:104423.

Kuhn, P., Nouri, B., Wilbert, S., Prahl, C., Kozonek, N., Schmidt, T., Yasser, Z., Ramirez, L., Zarzalejo, L., Meyer, A., Vuilleumier, L., Heinemann, D., Blanc, P., and Pitz-Paal, R. (2018). Validation of an all-sky imager–based nowcasting system for industrial pv plants. *Progress in Photovoltaics: Research and Applications*, 26(8):608–621.

Kumar, D. S., Yagli, G. M., Kashyap, M., and Srinivasan, D. (2020). Solar irradiance resource and forecasting: a comprehensive review. *IET Renewable Power Generation*, 14(10):1641–1656.

Kumler, A., Xie, Y., and Zhang, Y. (2018). A new approach for short-term solar radiation forecasting using the estimation of cloud fraction and cloud albedo. *Technical Report*.

Kumler, A., Xie, Y., and Zhang, Y. (2019). A physics-based smart persistence model for intra-hour forecasting of solar radiation (pspi) using ghi measurements and a cloud retrieval technique. *Solar Energy*, 177:494–500.

Kwak, S., Geroliminis, N., and Frossard, P. (2021). Traffic signal prediction on transportation networks using spatio-temporal correlations on graphs. *IEEE Transactions on Signal and Information Processing over Networks*, 7:648–659.

Lai, G., Chang, W.-C., Yang, Y., and Liu, H. (2018). Modeling long- and short-term temporal patterns with deep neural networks. *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Pritzel, A., Ravuri, S., Ewalds, T., Alet, F., Eaton-Rosen, Z., et al. (2023). Graphcast: Learning skillful medium-range global weather forecasting. *Science*.

Lauret, P., Voyant, C., Soubdhan, T., David, M., and Poggi, P. (2015). A benchmarking of machine learning techniques for solar radiation forecasting in an insular context. *Solar Energy*, 112:446–457.

Le Guen, V. and Thome, N. (2020). A deep physical model for solar irradiance forecasting with fisheye images. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2685–2688.

Lee, W., Kim, K., Park, J., Kim, J., and Kim, Y. (2018). Forecasting solar power using long-short term memory and convolutional neural networks. *IEEE Access*, 6:73068–73080.

Li, D., Yang, F., Miao, S., Gan, Y., Yang, B., and Zhang, Y. (2023). An adaptive spatiotemporal fusion graph neural network for short-term power forecasting of multiple wind farms. *Journal of Renewable and Sustainable Energy*, 15(1).

Li, Y., Yu, R., Shahabi, C., and Liu, Y. (2017). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *International Conference on Learning Representations (ICLR)*.

Li, Z., Rahman, S., Vega, R., and Dong, B. (2016). A hierarchical approach using machine learning methods in solar photovoltaic energy production forecasting. *Energies*, 9(1):55.

Lipton, Z. C., Kale, D. C., Elkan, C., and Wetzel, R. (2016). Learning to diagnose with lstm recurrent neural networks. *International Conference on Learning Representations (ICLR)*.

Liu, H., Chen, W., Chen, W., and Gu, Y. (2022). A cnn-lstm-based domain adaptation model for remaining useful life prediction. *Measurement Science and Technology*, 33(11):115118.

Meng, C., Seo, S., Cao, D., Griesemer, S., and Liu, Y. (2022). When physics meets machine learning: A survey of physics-informed machine learning. *arXiv preprint arXiv:2203.16797*.

Michel, P., Levy, O., and Neubig, G. (2019). Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32.

Mitra, I., Heinemann, D., Ramanan, A., Kaur, M., Sharma, S. K., Tripathy, S. K., and Roy, A. (2022). Short-term pv power forecasting in india: recent developments and policy analysis. *International Journal of Energy and Environmental Engineering*, 13(2):515–540.

Moreno, G., Santos, C., Martín, P., Rodríguez, F. J., Peña, R., and Vuksanovic, B. (2021). Intra-day solar power forecasting strategy for managing virtual power plants. *Sensors*, 21(16):5648.

Mukhoty, B. P., Maurya, V., and Shukla, S. K. (2019). Sequence to sequence deep learning models for solar irradiation forecasting. In *2019 IEEE Milan PowerTech*, pages 1–6.

Müller, S. C. and Remund, J. (2014). Satellite based shortest term solar energy forecast system for entire europe for the next hours. In *30th European Photovoltaic Solar Energy Conference and Exhibition (EUPVSEC), Paris, France*.

Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., and Yu, B. (2019). Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080.

Nespoli, A., Ogliari, E., Leva, S., Massi Pavan, A., Mellit, A., Lughi, V., and Dolara, A. (2019). Day-ahead photovoltaic forecasting: A comparison of the most effective techniques. *Energies*, 12(9).

Ortega, A., Frossard, P., Kovačević, J., Moura, J. M., and Vandergheynst, P. (2018). Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828.

Park, J. and Park, J. (2019). Physics-induced graph neural network: An application to wind-farm power estimation. *Energy*, 187:115883.

Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., Kurth, T., Hall, D., Li, Z., Azizzadenesheli, K., et al. (2022). Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *International Conference on Learning Representations (ICLR)*.

Pedro, H. and Coimbra, C. (2012). Assessment of forecasting techniques for solar power production with no exogenous inputs. *Solar Energy*, 86:2017–2028.

Pérez, E. B., Mejías, Á. C., and Bachiller, A. M. E. (2005). Vector calculus on weighted networks. https://api.semanticscholar.org/CorpusID:118032662.

Persson, C., Bacher, P., Shiga, T., and Madsen, H. (2017). Multi-site solar power forecasting using gradient boosted regression trees. *Solar Energy*, 150:423–436.

Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., and Battaglia, P. W. (2020). Learning mesh-based simulation with graph networks. *International Conference on Learning Representations (ICLR)*.

Pierro, M., De Felice, M., Maggioni, E., Moser, D., Perotto, A., Spada, F., and Cornaro, C. (2017). A new approach for regional photovoltaic power estimation and forecast. In *33th European Photovoltaic Solar Energy Conference and Exhibition (EUPVSEC), Amsterdam, The Netherlands*.

Quesada-Ruiz, S., Chu, Y., Tovar-Pescador, J., Pedro, H., and Coimbra, C. (2014). Cloud-tracking methodology for intra-hour dni forecasting. *Solar Energy*, 102:267–275.

Raissi, M., Perdikaris, P., and Karniadakis, G. (2019a). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019b). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. In *Elsevier*, volume 378, pages 686–707.

Ren, X., Gao, Y., Zhang, F., and Gao, L. (2022). A deep learning-based method for ultra-short-term pv power prediction. *Journal of Physics: Conference Series*, 2260(1):012056.

Sahili, Z. A. and Awad, M. (2023). Spatio-temporal graph neural networks: A survey. *arXiv preprint arXiv:2301.10569*.

Salvador, S. and Chan, P. (2007). Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580.

Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., and Battaglia, P. (2020). Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning (ICML)*, pages 8459–8468. PMLR.

Schmidt, T., Calais, M., Roy, E., Burton, A., Heinemann, D., Kilper, T., and Carter, C. (2017). Short-term solar forecasting based on sky images to enable higher pv generation in remote electricity networks. *Renewable Energy and Environmental Sustainability*, 2:23.

Schoene, A. M., Turner, A. P., and Dethlefs, N. (2020). Bidirectional dilated lstm with attention for fine-grained emotion classification in tweets. In *Proceedings of the AAAI-20 Workshop on Affective Content Analysis, New York, USA*.

Seo, Y., Defferrard, M., Vandergheynst, P., and Bresson, X. (2018). Structured sequence modeling with graph convolutional recurrent networks. In *Advances in Neural Neural Information Processing: 25th International Conference, ICONIP 2018*, pages 362–373.

Shih, S.-Y., Sun, F.-K., and Lee, H.-y. (2019). Temporal pattern attention for multivariate time series forecasting. *Machine Learning*, 108:1421–1441.

Si, Z., Yang, M., Yu, Y., and Ding, T. (2021). Photovoltaic power forecast based on satellite images considering effects of solar position. *Applied Energy*, 302:117514.

Simeunović, J., Schubnel, B., Alet, P.-J., and Carrillo, R. E. (2022a). Spatio-temporal graph neural networks for multi-site pv power forecasting. *IEEE Transactions on Sustainable Energy*, 13(2):1210–1220.

Simeunović, J., Schubnel, B., Alet, P.-J., Carrillo, R. E., and Frossard, P. (2022b). Interpretable temporal-spatial graph attention network for multi-site pv power forecasting. *Applied Energy*, 327:120127.

Singh, B. and Pozo, D. (2019). A guide to solar power forecasting using arma models. In *2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, pages 1–4.

Sirch, T., Bugliaro, L., Zinner, T., Möhrlein, M., and Vazquez-Navarro, M. (2016). Cloud and dni nowcasting with msg/seviri for the optimised operation of concentrating solar power plants. *Atmospheric Measurement Techniques Discussions*, pages 1–45.

Song, S., Yang, Z., Goh, H., Huang, Q., and Li, G. (2022). A novel sky image-based solar irradiance nowcasting model with convolutional block attention mechanism. *Energy Report*, 8(1):125–132.

Sperati, S., Alessandrini, S., and Delle Monache, L. (2016). An application of the ecmwf ensemble prediction system for short-term solar power forecasting. *Solar Energy*, 133:437–450.

Stein, J. S., Holmgren, W. F., Forbess, J., and Hansen, C. W. (2016). Pvlib: Open source photovoltaic performance modeling functions for matlab and python. In *Proc. 2016 IEEE 43rd Photovoltaic Specialists Conference (PVSC)*, pages 3425–3430.

Stocker, T. (2011). *Introduction to Climate Modelling*. Advances in Geophysical and Environmental Mechanics and Mathematics. Springer Berlin Heidelberg.

Sweeney, C., Bessa, R. J., Browell, J., and Pinson, P. (2020). The future of forecasting for renewable energy. *WIREs Energy and Environment*, 9(2):e365.

Takeishi, N. and Kalousis, A. (2021). Physics-integrated variational autoencoders for robust and interpretable generative modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 14809–14821.

Toubeau, J.-F., Morstyn, T., Bottieau, J., Zheng, K., Apostolopoulou, D., De Grève, Z., Wang, Y., and Vallée, F. (2021). Capturing spatio-temporal dependencies in the probabilistic forecasting of distribution locational marginal prices. *IEEE Transactions on Smart Grid*, 12(3):2663–2674.

Van Haaren, R., Morjaria, M., and Fthenakis, V. (2014). Empirical assessment of short-term variability from utility-scale solar pv plants. *Progress in Photovoltaics: Research and Applications*, 22:548–559.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2018). Graph attention networks. *International Conference on Learning Representations (ICLR)*.

Vyas, S., Goyal, Y., Bhatt, N., Bhuwania, S., Patel, H., Mishra, S., and Tripathi, B. (2022). Forecasting solar power generation on the basis of predictive and corrective maintenance activities. *arXiv preprint arXiv:2205.08109*.

Wang, P., Liu, H., Zheng, X., and Ma, R. (2023). A new method for spatio-temporal transmission prediction of covid-19. *Chaos, Solitons & Fractals*, 167:112996.

Wang, R. and Yu, R. (2021). Physics-guided deep learning for dynamical systems: A survey. *arXiv preprint arXiv:2107.01272*.

Wang, X., He, X., Cao, Y., Liu, M., and Chua, T.-S. (2019). Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 950–958. Association for Computing Machinery.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. (2020). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21.

Xiao, C., Chen, N., Hu, C., Wang, K., Xu, Z., Cai, Y., Xu, L., Chen, Z., and Gong, J. (2019). A spatiotemporal deep learning model for sea surface temperature field prediction using time-series satellite data. *Environmental Modelling & Software*, 120:104502.

Xiao, Z., Tang, F., and Wang, M. (2023). Wind power short-term forecasting method based on lstm and multiple error correction. *Sustainability*, 15(4):3798.

Yang, C., Thatte, A. A., and Xie, L. (2015). Multitime-scale data-driven spatio-temporal forecast of photovoltaic generation. *IEEE Transactions on Sustainable Energy*, 6:104–112.

Yang, L., Gao, X., Hua, J., Wu, P., Li, Z., and Jia, D. (2020). Very short-term surface solar irradiance forecasting based on fengyun-4 geostationary satellite. *Sensors*, 20(9):2606.

Yin, Y., Le Guen, V., Dona, J., de Bézenac, E., Ayed, I., Thome, N., and Gallinari, P. (2021). Augmenting physical models with deep networks for complex dynamics forecasting. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124012.

Yu, B., Yin, H., and Zhu, Z. (2018). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *International Joint Conference on Neural Networks (IJCNN)*.

Zhang, C., Yu, J. J. Q., and Liu, Y. (2019). Spatial-temporal graph attention networks: A deep learning approach for traffic forecasting. *IEEE Access*, 7:166246–166256.

Zhang, M., Sun, Y., Feng, C., Zhen, Z., Wang, F., Li, G., Liu, D., and Wang, H. (2022a). Graph neural network based short-term solar irradiance forcasting model considering surrounding meteorological factors. In *2022 IEEE/IAS 58th Industrial and Commercial Power Systems Technical Conference (I&CPS)*, pages 1–9.

Zhang, R., Li, G., Bu, S., Kuang, G., He, W., Zhu, Y., and Aziz, S. (2022b). A hybrid deep learning model with error correction for photovoltaic power forecasting. *Frontiers in Energy Research*, 10:948308.

Zheng, C., Fan, X., Wang, C., and Qi, J. (2020). Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1234–1241.

Zhou, H., Zhang, Y., Yang, L., Liu, Q., Yan, K., and Du, Y. (2019). Short-term photovoltaic power forecasting based on long short term memory neural network and attention mechanism. *IEEE Access*, 7:78063–78074.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81.

Zhu, Q., Chen, J., Zhu, L., Duan, X., and Liu, Y. (2018). Wind speed prediction with spatio–temporal correlation: A deep learning approach. *Energies*, 11(4):705.

Curriculum vitae

Personal information:

Name: **JELENA SIMEUNOVIĆ**
E-mail: jelena.simeunovic@epfl.ch
jelena.simeunovic@csem.ch
Google Scholar: **link**

Education:

2019 – 2023 Phd Student at École Polytechnique Fédérale de Lausanne
Department: Graph Signal Processing, LTS4 laboratory (*EEDE School)*

2016 – 2017 Master degree at Faculty of Electrical Engineering, Belgrade
Department: Signals and Systems (*GPA 10,0 out of 10,0)*

2012 – 2016 Graduated at Faculty of Electrical Engineering, Belgrade
Department: Signals and Systems (*GPA 8.57 out of 10,0*)

2008 – 2012 Grammar school "Vuk Karadžić" in Loznica, department: Natural sciences and mathematics
Awarded with: *Vukova diploma (highest accomplishment for perfect GPA 5,0 out of 5,0)*

Experience:

2019 – - PhD Student at Digital Energy Solutions group, PV Center, CSEM, Neuchatel

2019 - McKinsey Virtual Academy PhD Student **(**McKinsey and Harvard ManageMentor courses)

2017 – 2018 - Administrator and consultant for system and application software at Global Engineering Technologies

2018 - Finalist in program for gifted and talented students, organized by US Chamber of Commerce, AmChamp - Young leaders in change, solved IBM's Case Study: Development of Competencies in IBM in the light of Digital Transformation

2017 - Internship at IBM
- Solved Amchamps' Case Study (given by Coca Cola HBC): Competitive dynamics in food retail market, awarded as one of top 3 ideas

2013 - 2017 - Fundraising team member and organizer at Elektrijada, in organization of Student's Union at Belgrade University - SUETF (Montenegro, Budva – 2017 th, Italy, Rimini – 2016 th, Montenegro, Bečići -2015 th, Hungary, Balaton – 2014th)

2013 -2014 - Student - Mentor at School of Electrical Engineering

Papers, conferences, projects:

2023
- Poster presentation " Interpretable temporal-spatial graph attention network for multi-site PV power forecasting " on Graph Signal Processing Workshop, Oxford 2023
- Attended Climate Change AI Summer School 2023

2022
- Paper: J. Simeunović et al., "Interpretable temporal-spatial graph attention network for multi-site PV power forecasting", Applied Energy, Volume 327, 2022, 120127, ISSN 0306-2619, https://doi.org/10.1016/j.apenergy.2022.120127
- Finalist at FameLab National Competition in scientific presentations, Audience award
- Gold medal at CSEM's scientific presentation competition
- Successfully completed summer school: Field-Based Insights into the Implementation of Renewable Energies

2021
- Paper: J. Simeunović et al., "Spatio-Temporal Graph Neural Networks for Multi-Site PV Power Forecasting," in *IEEE Transactions on Sustainable Energy*, vol. 13, no. 2, pp. 1210-1220, April 2022, doi:10.1109/TSTE.2021.3125200.
- Paper: R. Carril et al., "Spatio-temporal machine learning methods for multi-site PV power forecasting", EU PVSEC conference, 2021, doi:10.4229/EUPVSEC20212021-5BO.7.2
- Presentation on Conference Applied Machine Learning Days (AMLD) (video available: https://www.youtube.com/watch?v=gCh-iRma6pw)
- Participant in a Panel discussion on AMLD conference
- 2nd award at CSEM's scientific presentation competition

2017
- Several projects regarding Neuro-marketing, Team-synergy, Stakeholder management, People and Technology in IBM
- Project: Cancer prediction using Neural Networks
- Master thesis: programming movement of the industrial ABB robot

2016
- Final thesis: the application of haptic devices in rehabilitation robotics

Skills:

- TOOLS: Pytorch, Tensorflow, Keras, LaTeX, Microsoft Office, Matlab, Jira, HP Quality Center, Microsoft Visual Studio, Oracle SQL Developer, QlickView
- PROGRAMMING LANGUAGES: Phyton, C, C++, SQL

Personality:

- Good Communication and Teamwork skills
- Excellent conceptual and analytical skills
- Perseverance
- Creative
- Reliability, proactivity and punctuality
- Ambitious
- Leadership

Languages

- Serbian language / *native*
- English language / *level C1*
  CAE Certificate, grade B
- German language / *level B1*
- French language / *level A2*

Other activities:

- Hiking, Swimming, Karate