Democratic and Popular Republic of Algeria

Ministry of Higher Education and Scientific Research

University of Souk Ahras

Faculty of Science and Technology

Laboratory of Electrical Engineering,

Electronic and Renewable Energy

# Thesis

Submitted in fulfilment of the requirements for the degree of Doctor of Philosophy (Renewable Energy) at the University of Souk Ahras

## Title

# Wind Energy Forecasting Using Artificial Intelligence (AI) techniques

**Presented by**

**Lokmene Melalkia**

In front of the following committee

| | | |
|---|---|---|
| Pr. Kamel Messaoudi | University of Souk Ahras | Chair |
| Pr. Farid Berrezzek | University of Souk Ahras | Supervisor |
| Pr. Khaled Khelil | University of Souk Ahras | Co-Supervisor |
| Pr. Naimi Djemai | University of Biskra | Examiner |
| Dr. Lotfi Farah | Badji Mokhtar Annaba University | Examiner |
| Pr. Zoubir Chelli | University of Souk Ahras | Examiner |
| Dr. Abdelhakim Saim | University of Nantes | Invited Member |

# Acknowledgements

*It has been an extraordinary and unforgettable journey to pursue my Ph.D. at the University of Souk Ahras. I owe a deep debt of gratitude to the many people whose support, guidance, and encouragement have made this achievement possible.*

*First and foremost, I would like to express my heartfelt thanks to my parents and family. Their unwavering love, patience, and sacrifices have been the foundation of my academic journey. Without their constant encouragement and belief in me, this work would not have been possible.*

*I am profoundly grateful to my supervisor, Professor Farid Berrezzek, for his invaluable guidance, insightful advice, and continuous support throughout my doctoral studies. His vision and dedication have inspired me to persevere and pursue my research with rigor and passion. I would also like to sincerely thank my co-supervisor, Professor Khaled Khelil, for his constructive feedback, encouragement, and for always challenging me to refine and strengthen my work.*

*I extend my sincere appreciation to Dr. Abdelhakim Saim at IREENA, Nantes university. His expertise, collaboration, and mentorship especially during my research stay in France were instrumental in shaping this thesis and broadening my academic perspective.*

*Finally, I wish to acknowledge the Faculty of Science and Technology and the Laboratory of Electrical Engineering and Renewable Energy (LEER Lab) at the University of Souk Ahras for providing the academic environment and resources that supported this work.*

# Abstract

The reliable integration of renewable energy into modern power systems critically depends on accurate forecasting of wind energy and wind speed. However, the stochastic, nonlinear, and non-stationary nature of wind presents persistent challenges for traditional statistical and standard machine learning methods. This thesis addresses these challenges by investigating advanced hybrid deep learning approaches that leverage signal decomposition, spatiotemporal architectures, and transfer learning to improve forecasting accuracy and robustness. For offshore wind power forecasting, a hybrid model integrating Ensemble Empirical Mode Decomposition (EEMD) with Convolutional Long Short-Term Memory (ConvLSTM) in an error-correction framework is proposed. This approach decomposes error signals into multi-scale components and employs ConvLSTM to capture spatiotemporal dependencies, thereby enhancing forecast reliability across multiple horizons. For wind speed forecasting, a novel hybrid framework is introduced that combines Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) for signal decomposition, a Bidirectional GRU-based Encoder-Decoder (Bi-GRU-ED) for temporal sequence modelling, and a homogeneous transfer learning strategy that reuses and fine-tunes forecasting modules across decomposed components. This design reduces training complexity while improving forecasting accuracy and robustness under diverse operating conditions. Overall, the findings contribute to more reliable wind forecasts and provide a strong foundation for future research in multi-source renewable energy forecasting.

**Keywords:** Wind energy, wind energy forecasting, artificial intelligence techniques, neural networks, Recurrent neural networks, pre-processing, hybrid models.

# Résumé

L'intégration fiable des énergies renouvelables dans les systèmes électriques modernes dépend de manière cruciale de la précision des prévisions de la production éolienne et de la vitesse du vent. Cependant, le caractère stochastique, non linéaire et non stationnaire du vent représente un défi persistant pour les méthodes statistiques traditionnelles et les techniques classiques d'apprentissage automatique. Cette thèse s'attaque à ces défis en explorant des approches avancées d'apprentissage profond hybride, combinant décomposition de signaux, architectures spatio-temporelles et transfert de connaissances, afin d'améliorer la précision et la robustesse des prévisions. Pour la prévision de la production éolienne offshore, un modèle hybride intégrant la Décomposition Empirique en Mode (EEMD) avec un réseau Convolutional Long Short-Term Memory (ConvLSTM) dans une architecture de correction d'erreur est proposé. Cette approche décompose les signaux d'erreur en composantes multi-échelles et utilise le ConvLSTM pour capturer les dépendances spatio-temporelles, améliorant ainsi la fiabilité des prévisions sur plusieurs horizons. Pour la prévision de la vitesse du vent, un nouveau cadre hybride est présenté, combinant la Décomposition Empirique Complète en Mode avec Bruit Adaptatif (CEEMDAN) pour la décomposition des signaux, un encodeur-décodeur Bidirectionnel GRU (Bi-GRU-ED) pour la modélisation des séquences temporelles, et une stratégie de transfert de connaissances homogène qui réutilise et affine les modules de prévision à travers les composantes décomposées. Cette conception réduit la complexité de l'entraînement tout en améliorant la précision et la robustesse des prévisions dans des conditions variées. Dans l'ensemble, ces travaux contribuent à des prévisions éoliennes plus fiables et fournissent une base solide pour les recherches futures sur la prévision des énergies renouvelables multi-sources.

**Mots-clés :** Énergie éolienne, prévision de l'énergie éolienne, techniques d'intelligence artificielle, réseaux de neurones, réseaux de neurones récurrents, prétraitement, modèles hybrides.

# الملخص

تعتمد عملية دمج الطاقة المتجددة بشكل موثوق في أنظمة الطاقة الحديثة بشكل كبير على دقة التنبؤ بإنتاج طاقة الرياح وسرعة الرياح. ومع ذلك، فإن الطابع العشوائي وغير الخطي وغير المستقر للرياح يمثل تحديًا مستمرًا أمام الطرق الإحصائية التقليدية وتقنيات التعلم الآلي القياسية. تهدف هذه الأطروحة إلى مواجهة هذه التحديات من خلال استكشاف نهج متقدم للتعلم العميق الهجين، يجمع بين تحليل الإشارات، والهياكل المكانية-الزمانية، وتقنيات التعلم بالنقل، لتعزيز دقة وموثوقية التنبؤات في مجال التنبؤ بطاقة الرياح البحرية، تم اقتراح نموذج هجين يجمع بين التحلل التجريبي للأنماط (EEMD) وشبكة الذاكرة الطويلة قصيرة المدى التلافيفية (ConvLSTM) ضمن هيكلية تصحيح الخطأ. تقوم هذه الطريقة بتحليل إشارات الخطأ إلى مكونات متعددة المقاييس، وتستفيد من ConvLSTM لالتقاط الاعتمادات المكانية والزمانية، مما يعزز موثوقية التنبؤات على عدة أفق زمني. أما في مجال التنبؤ بسرعة الرياح، فقد تم تقديم إطار هجين جديد يجمع بين التحلل التجريبي الكامل للأنماط مع الضوضاء التكيفية (CEEMDAN) لتحليل الإشارات، ومشفر-فك ترميز GRU ثنائي الاتجاه (Bi-GRU-ED) لنمذجة التسلسلات الزمنية، واستراتيجية تعلم بالنقل متجانسة تعيد استخدام وتخصيص وحدات التنبؤ عبر المكونات المحللة. يقلل هذا التصميم من تعقيد التدريب مع تحسين دقة وموثوقية التنبؤات في ظروف تشغيل متنوعة .بشكل عام، تسهم هذه النتائج في تحسين موثوقية التنبؤات وتوفير أساس قوي للأبحاث المستقبلية في مجال التنبؤ بطاقة المصادر المتجددة المتعددة.

**الكلمات المفتاحية:** طاقة الرياح، التنبؤ بطاقة الرياح، تقنيات الذكاء الاصطناعي، الشبكات العصبية، الشبكات العصبية المتكررة، المعالجة الأولية، النماذج الهجينة.

.

## Table of Contents

## List of Figures

## List of Tables

# General Introduction

The growing need for clean, sustainable energy solutions has placed renewable energy at the centre of the global energy transition. Renewable energy sources such as solar energy, hydropower, geothermal energy, and biomass offer diverse and environmentally friendly ways to meet rising electricity demands while reducing greenhouse gas emissions. Each of these sources plays a unique and complementary role in building a resilient and sustainable energy system.

Solar power, through photovoltaic and concentrated solar technologies, provides a versatile and widely deployable source of clean electricity. Hydropower delivers reliable baseload and peak-load power, with pumped-storage systems crucial for the grid. Geothermal energy offers consistent, weather-independent generation, while bioenergy converts organic matter into electricity, heat, and advanced biofuels. Marine energy, often referred to as ocean energy, represents a growing frontier in renewable energy technologies. This category encompasses tidal energy, wave energy, ocean thermal energy conversion (OTEC), and the use of salinity gradients. Tidal and wave energy technologies harness the kinetic and potential energy of ocean movements, while OTEC systems exploit temperature differences between surface and deep ocean waters to generate electricity. Marine energy offers immense potential, particularly for coastal and island nations, and its consistent nature provides a valuable complement to more variable renewable sources like wind and solar energy.

Amid this diverse energy landscape, wind power has emerged as a leading force in the renewable energy revolution. Both onshore and offshore wind energy play critical roles in reducing carbon emissions and meeting global electricity demand. Onshore wind farms, valued for their accessibility and cost-effectiveness, have become widespread, enabling communities worldwide to benefit from clean, affordable energy. Meanwhile, offshore wind farms, capitalizing on stronger and more consistent wind speeds, offer large-scale generation for regions with high coastal energy need.

The integration of wind power into electrical grids is driving advancements in energy storage, grid management, and forecasting techniques, which are essential for reliable and efficient power delivery. The rapid scalability and environmental benefits of wind power make

it indispensable for achieving carbon neutrality and supporting the transition to a sustainable energy future. By combining wind power with other renewable sources, nations can build a resilient energy system that addresses environmental challenges, enhances energy security, and supports global development goals. Thus, while offshore wind benefits from superior wind resources, onshore wind remains crucial due to its lower costs and easier maintenance. Together, they form a complementary framework that supports diverse energy needs.

**Problem Statement:**

The inherently dynamic and unpredictable nature of wind stems from its dependency on various atmospheric and environmental factors, making it a highly complex resource for power generation. Unlike traditional energy sources, wind is neither steady nor entirely controllable, which presents serious challenges in preserving stability in the grid and providing efficient electricity generation. One major factor contributing to this complexity is its nonlinear behaviour. Wind speed is highly sensitive to meteorological conditions, like pressure, humidity, and temperature, which interact in intricate ways to create irregular variations. This nonlinearity is further compounded by temporal variability. Wind speeds fluctuate over multiple time scales, including short-term changes caused by turbulence or gusts, diurnal patterns driven by thermal dynamics, and seasonal cycles influenced by weather systems. These fluctuations result in uneven energy production, which is challenging to predict and manage.

Spatial heterogeneity adds another layer of complexity, as wind is not uniform across geographical locations. Topographical features like mountains, valleys, and coastlines influence wind flows, while offshore winds, though often steadier, are still affected by marine weather systems. Turbulence, characterized by irregular and chaotic air movement, disrupts smooth wind flows and introduces unpredictability. Sudden gusts exacerbate this issue by causing sharp spikes in wind power generation, which can destabilize the grid.

Wind behaviour is also heavily influenced by large-scale weather systems such as cyclones, anticyclones, and jet streams. Accurately predicting these systems requires advanced meteorological models, but even small errors in these models can propagate and affect wind power forecasts. Additionally, wake effects in wind farms, where turbines impact each other's performance by reducing wind speed and increasing turbulence for downstream turbines, further complicate the predictability of power output.

Thus, the variability and intermittency of wind make accurate forecasting critical for scheduling power generation and managing reserves. Specifically, sudden drops in wind speed can create power deficits, while unanticipated surges in generation can lead to grid overload if not managed properly. To maintain stability, wind power must be balanced with more predictable energy sources, often requiring backup systems like gas turbines or battery storage. Consequently, this unpredictability also increases operational costs for utilities, necessitating investments in flexible generation, grid upgrades, and energy storage solutions.

To address these challenges**,** Precise wind power forecasting is now a key instrument for modern energy networks. It is essential for predicting energy production, facilitating the seamless integration of wind power into the grid, and enhancing overall system reliability.

**Research motivations:**

This research is motivated by recent advancements in artificial intelligence (AI) and its proven applications in processing big data and building predictive models. AI offers unprecedented capabilities for addressing the complexities of wind power forecasting, as it can capture intricate spatiotemporal dependencies and uncover patterns that traditional method's often miss. The resulting accurate forecasts are crucial for enabling grid operators and energy planners to effectively manage variability, optimize energy storage systems, and ensure a reliable balance between energy supply and demand amidst fluctuating wind conditions.

The present study aims to leverage these AI advancements to directly tackle the challenges of wind variability. By developing innovative AI-based frameworks for wind power forecasting.

**Research objectives:**

The main aim of this research is to improve the precision and dependability of wind power forecasts by using sophisticated AI techniques. This overarching goal is pursued through the following specific objectives:

- To analyze large-scale wind datasets to identify key factors influencing wind power variability and to pre-process the data for enhanced model performance.
- To design and implement a hybrid forecasting framework that integrates advanced artificial intelligence techniques to improve forecasting accuracy.

- To evaluate the proposed models using robust error metrics and conduct a comparative analysis with established methods to validate their superiority.

The remainder of this dissertation is organized into four chapters, followed by a general conclusion. Chapter I establishes the fundamental background on wind energy and forecasting approaches. Chapter II reviews artificial intelligence models and frameworks applied to wind power forecasting, with a particular emphasis on machine learning and deep learning architectures. Chapter III presents the proposed forecasting frameworks, detailing the integration of signal decomposition methods with advanced neural networks. Finally, Chapter IV reports the simulation results, provides an in-depth discussion of the experimental findings, and highlights the contributions and implications of this research. Finally, the General Conclusion summarizes the entire work, highlighting the contributions, implications, and potential future research directions.

# Chapter I: Fundamental Concepts of Wind Energy and Forecasting Approaches

# Fundamental Concepts of Wind Energy and Forecasting Approaches

## I.1 Introduction

Wind energy is growing as an essential component of the global transition to renewable energies, providing a clean and limitless source of electricity. However, its inherent intermittency poses a major challenge for large-scale integration into modern power networks. Unlike conventional generation, wind power is particularly tied to weather and geographical factors., making it difficult to predict with complete accuracy [1]. This unpredictability affects grid stability, operational planning, and energy market participation [2]. Consequently, the development of reliable forecasting techniques has become a central research objective, as accurate predictions mitigate the risks of imbalance, reduce reserve costs, and enhance the overall efficiency of power system operations.

Wind energy forecasting includes a variety of methodologies, from physical and statistical models to artificial intelligence (AI) and hybrid approaches, each with distinct strengths and limitations. Forecasts are often classified by time horizon, spatial scale, or modelling technique, and rely on diverse types of input data, including meteorological measurements, numerical weather predictions, and historical turbine records [3]. Forecasting models' efficiency is highly dependent on the quality of input data and the resilience of pre-processing steps. Moreover, challenges such as handling missing values, detecting outliers, and managing uncertainty add further complexity, highlighting the necessity of advanced methodologies capable of capturing nonlinear and dynamic characteristics of wind energy systems [2].

In recent years, AI-based forecasting methods have demonstrated remarkable improvements over traditional approaches by leveraging large datasets and learning complex temporal dependencies. Models based on deep learning specifically showed abilities in enhancing both short-term and long-term forecasting accuracy [4]. Nonetheless, the training phase requires careful consideration of optimization algorithms, hyperparameters, and strategies to avoid overfitting. Against this background, this chapter provides a comprehensive overview of wind energy and its forecasting process, detailing the conversion mechanisms, system components, and integration challenges, before examining the classification of forecasting methods, the role

of data pre-processing, and the fundamentals of training. This foundation establishes the necessary context for the subsequent chapters.

## I.2 The Origins of Wind Energy

Wind energy originates from the movement of air driven by pressure differences in the atmosphere, where air flows from high to low-pressure zones. This process results from a combination of factors, primarily the unequal heating of the Earth surface by solar radiation, which creates large-scale circulation systems as warm air flows near the equator, and cooler air descends near the poles. Local contrasts between land and water further generate breezes that illustrate how thermal differences translate into wind formation [5]. The Earth rotation adds another layer through the Coriolis effect, which deflects air masses and organizes them into global patterns such as trade winds and westerlies. Geographical features also shape wind dynamics: mountains, valleys, coastlines, and bodies of water can channel or intensify airflow, while surface roughness and urban structures influence microclimates relevant for turbine placement. Seasonal changes further modulate wind regimes, as shifts in heating between land and ocean drive circulation patterns like monsoons, which strongly affect wind intensity and persistence. Together, these mechanisms explain the variability, spatial distribution, and temporal availability of wind energy resources across the globe [5]. Figure I.1 presents the geographical locations used in this study Tataouine in southern Tunisia, Adrar in Algeria, and the German North Sea sector the three regions selected for their markedly different climatic characteristics and high wind-energy potential.

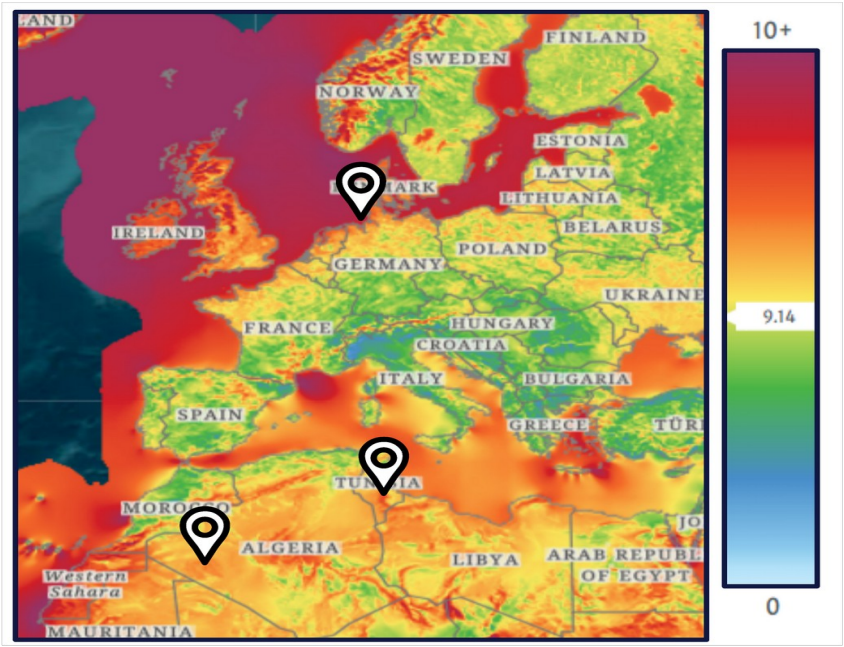



**Figure I.1**: Wind energy potential of the study case's locations

## I.3 Wind Energy Conversion Process

The transformation of wind energy involves a series of steps that transform kinetic energy of airflow to usable electrical energy using wind turbines and generators, which work in tandem to harness and convert the energy of wind effectively, as shown in Figure I.2. The process of converting can be parted into two primary phases, Kinetic energy gets turned into mechanical energy, and mechanical energy gets converted to electrical energy [6].

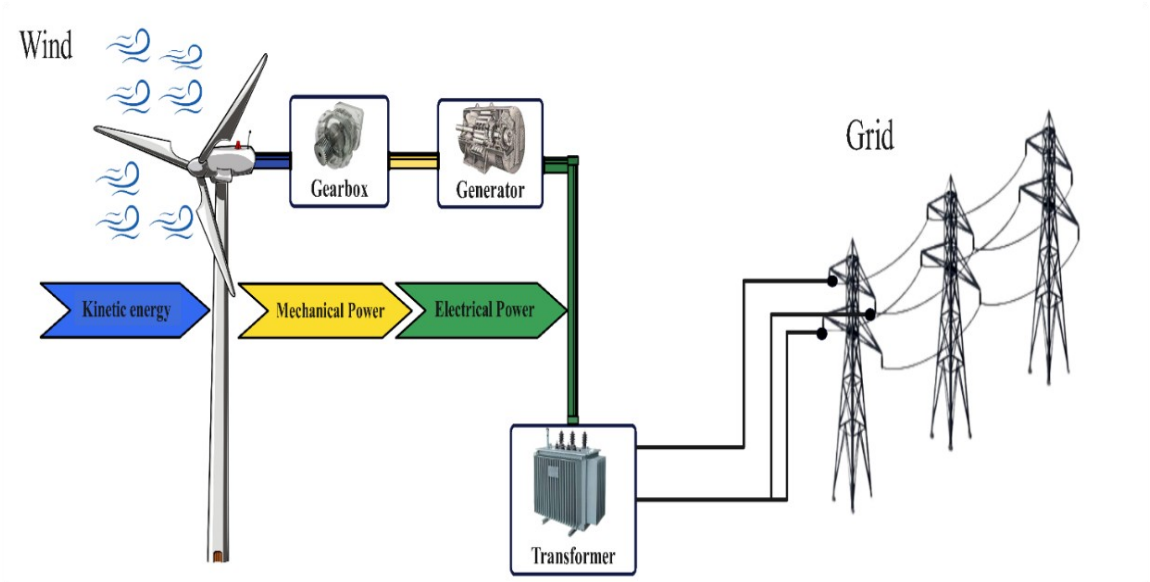



**Figure I.2**: Wind energy conversion process

Wind energy is intrinsically kinetic energy, generated by moving air masses. The kinetic energy $E_k$ of wind is given by Equation I.1 [6].

$$E_k = \frac{1}{2} m v^2 \tag{I.1}$$

The m is the weight of the moving air ( kg ) determined using Equation I.3and $v$ is the wind speed ( m/s ).

$$m = \rho A v \tag{I.2}$$

Where ρ is the air density (kg/m³) and A is the swept area of the turbine blades (A=πr², r is the blade radius in meters). By combining these equations, the full power that comes from wind $P_{wind}$ can be expressed in Equation I.3 [6].

$$P_{wind} = \frac{1}{2} \rho A v^3 \tag{I.3}$$

However, wind turbines cannot absorb all kinetic energy. Betz's Law theoretically states that up to just 59.3% of the wind's kinetic energy may be harnessed. The turbine's actual captured power is $P_{captured}$ [6].

$$P_{captured} = C_p P_{wind} \quad \text{(I.4)}$$

Where $C_p$ is the coefficient of power (typically between 0.4 and 0.5 for modern turbines), which reflects the efficiency of the turbine in converting wind energy into mechanical power.

The turbine's blades convert the wind's kinetic energy into mechanical energy, which in turn drives the rotor and causes it to rotate. Equation I.5 expresses the mechanical power $P_{mech}$ generated by the turbine.

$$P_{mech} = \tau \omega \quad \text{(I.5)}$$

Here τ is the torque ( N m ), in addition ω is the velocity of rotation (rad/s), respectively.

This $P_{mech}$ is then transferred to an electrical generator to generate the electrical power $P_{elec}$ through Equation I.6 [6].

$$P_{elec} = \eta_{gen} P_{mech} \quad \text{(I.6)}$$

Where the $\eta_{gen}$ represents the efficiency of the generator (typically 90–95%).

The overall effectiveness of the wind power transformation process accounts for the efficiencies of both the turbine and the generator. The combined efficiency $\eta_{total}$is given in Equation I.7.

$$\eta_{total} = \eta_{gen} C_p \quad \text{(I.7)}$$

Equation I.8 expresses the wind turbine system's final electricity generation.

$$P_{output} = \eta_{total} P_{wind} = \eta_{gen} C_p \frac{1}{2} \rho A v^3 \quad \text{(I.8)}$$

This equation demonstrates how multiple factors, such as turbine design, generator efficiency, air density, and wind speed interact to determine the overall power output [6].

## I.4 Key Components of the Wind Power System

A wind turbine structure is composed of several interconnected components combined to effectively capture, convert, and Supply wind-generated energy as usable electricity. These components are presented in Figure I.3 [7]. In the core lies the wind turbine structure, which contains the blades, rotor, nacelle, tower, and foundation. The blades, through their aerodynamic design and swept area, absorb and transfer kinetic energy to the rotor, which converts it into rotational mechanical energy [6]. This motion is transmitted to the nacelle, where components such as the gearbox and generator convert the mechanical energy

into electricity. The tower and foundation ensure structural integrity, stability, and access to stronger wind currents. Supporting these primary elements are the control systems, which enhance efficiency and safety by orienting the turbine through yaw motors, adjusting blade pitch, and activating braking or overspeed protection mechanisms during extreme conditions. This obtained electricity is subsequently filtered and transferred over electrical equipment, transformers, and high-capacity cables, ensuring compatibility with grid standards and reliable delivery, whether in onshore or offshore installations [6].

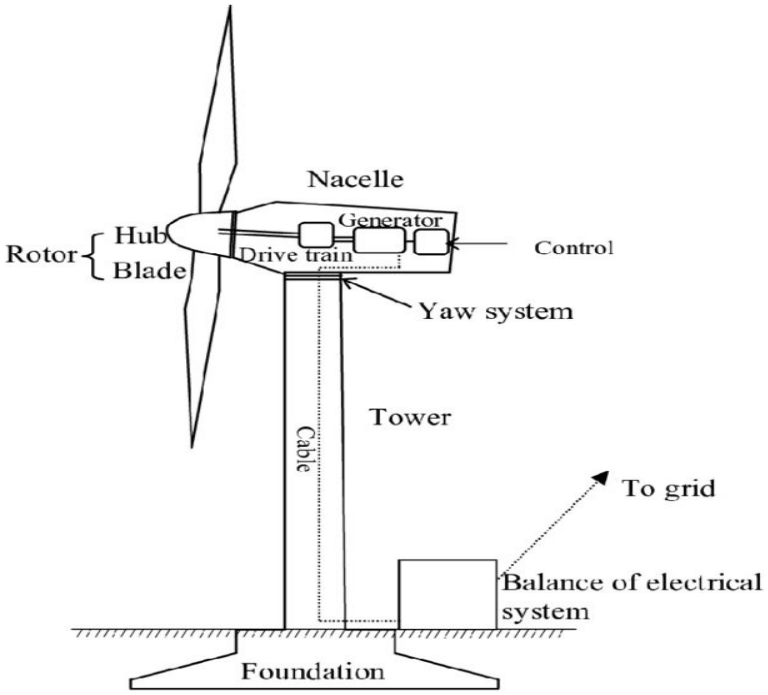



**Figure I.3**: Wind turbines components

## I.5 Types of Wind Power Systems

Wind energy systems are commonly classed either onshore, offshore, or hybrid configurations, each offering unique advantages and constraints depending on geographical, environmental, and technical conditions. Onshore systems, as shown in Figure I.4, represent the most widespread application of wind technology such as the Alta Wind Energy Center (AWEC) in California, one of the largest wind farms in the world, with a capacity of approximately 1,548 MW. They are deployed on land in areas with favourable wind resources, benefiting from lower construction and maintenance costs and easier access to existing infrastructure [8]. Their economic competitiveness has driven global adoption, though challenges including land-use issues, noise, and visual consequences remain important considerations for large-scale deployment [7].

**Figure I.4**: AWEC Onshore Wind Farm [9].

Offshore wind systems, illustrated in Figure I.5, are in marine environments in which wind is usually more forceful and stable compared on land. These projects allow the development of large-scale farms with higher capacity factors while minimising land-use conflicts and reducing aesthetic concerns such as in the Borssele offshore wind farm in the Netherlands, which has a total installed capacity of approximately 752 MW. However, their implementation requires advanced engineering solutions to withstand harsh marine conditions, specialised vessels for installation and maintenance, and costly subsea transmission systems [10]. Some of the largest wind farms worldwide are presented in Table I.1.

**Figure I.5**: Borssele Offshore Wind Farm [9].

**Table I.1**: Some of the largest offshore and onshore wind farms

| Wind Farm Name | Type | Country | Capacity |
|---|---|---|---|
| Gansu (Jiuquan) Wind Power Base | Onshore | China | 10,000 MW |
| Alta Wind Energy Center (AWEC) | Onshore | USA | 3,000 MW |
| Jaisalmer Wind Park | Onshore | India | 1,600 MW |
| Hornsea Wind Farm | Offshore | U K | 5,000 MW |
| Dogger Bank Wind Farm | Offshore | U K | 3,600 MW |
| Borssele Offshore Wind Farm | Offshore | Netherlands | 1,500 MW |

Hybrid wind systems, depicted in Figure I.6 [7], integrate wind energy with complementary resources such as solar, hydropower, or energy storage. By compensating for the variability of wind through diversification, they provide more stable and reliable energy output. These systems are particularly valuable for microgrids, remote or island communities, and regions with limited grid connectivity, where reliability and flexibility are critical. While hybrid solutions involve higher upfront costs and more complex operational management, advances in storage technologies, smart grid control, and modular system design are addressing these challenges, positioning hybrid systems as a promising pathway for resilient and sustainable energy transitions [1].

**Figure I.6**: Hybrid wind Power system

## I.6 Challenges of Variability and Uncertainty in Wind Power Generation

Wind power, though a clean and sustainable resource, is characterised by strong variability and uncertainty stemming from its dependence on atmospheric conditions. The random behaviour of wind speed and direction causes important fluctuations in power output across different timescales, creating operational challenges for modern power systems. From a grid management perspective, these fluctuations complicate the balance between supply and demand, often requiring rapid reserve activation or curtailment. For systems that have substantial wind energy penetration, this variability increases ramping requirements, accelerates mechanical wear on conventional units, and ultimately raises operational costs [11].

Forecasting uncertainty compounds these challenges. Despite advances in numerical weather prediction, short-term forecasts remain prone to errors due to turbulence, local meteorological effects, and terrain influences. Such inaccuracies can cause overestimations, risking supply shortages, or underestimations, leading to excessive reserve commitments and inefficiencies in electricity markets [11]. These combined effects impact frequency regulation, voltage stability, and reserve allocation, and in severe cases may trigger protective actions such as load shedding or renewable curtailment. Consequently, developing more accurate, adaptive, and robust forecasting approaches, particularly those harnessing artificial intelligence and hybrid signal processing, is crucial to mitigate uncertainty, enhance system reliability, and increase the successful integration of wind energy within grids.

## I.7 Wind Power Forecasting Classifications

Wind power forecasting approaches could be classified systematically according to several key dimensions, including time horizon, spatial dimension, model type, output time step, and uncertainty handling, as in Figure I.7 where each classification criterion addresses specific aspects of the forecasting process, enabling a tailored approach to meet diverse operational and planning needs [12].

The time horizon categorizes forecasts based on the lead time between prediction and the expected wind power output. The spatial dimension refers to the geographical scale of the forecast. The model type classifies forecasting approaches according to the methodological framework used to generate predictions. The output time step indicates the temporal resolution of the forecasted values. Forecasts can be generated at intervals ranging from minutes to several hours depending on operational needs. The uncertainty handling dimension describes how

forecasting models represent prediction uncertainty. Forecasts may be deterministic or probabilistic, providing either single estimates or ranges of possible outcomes [13].

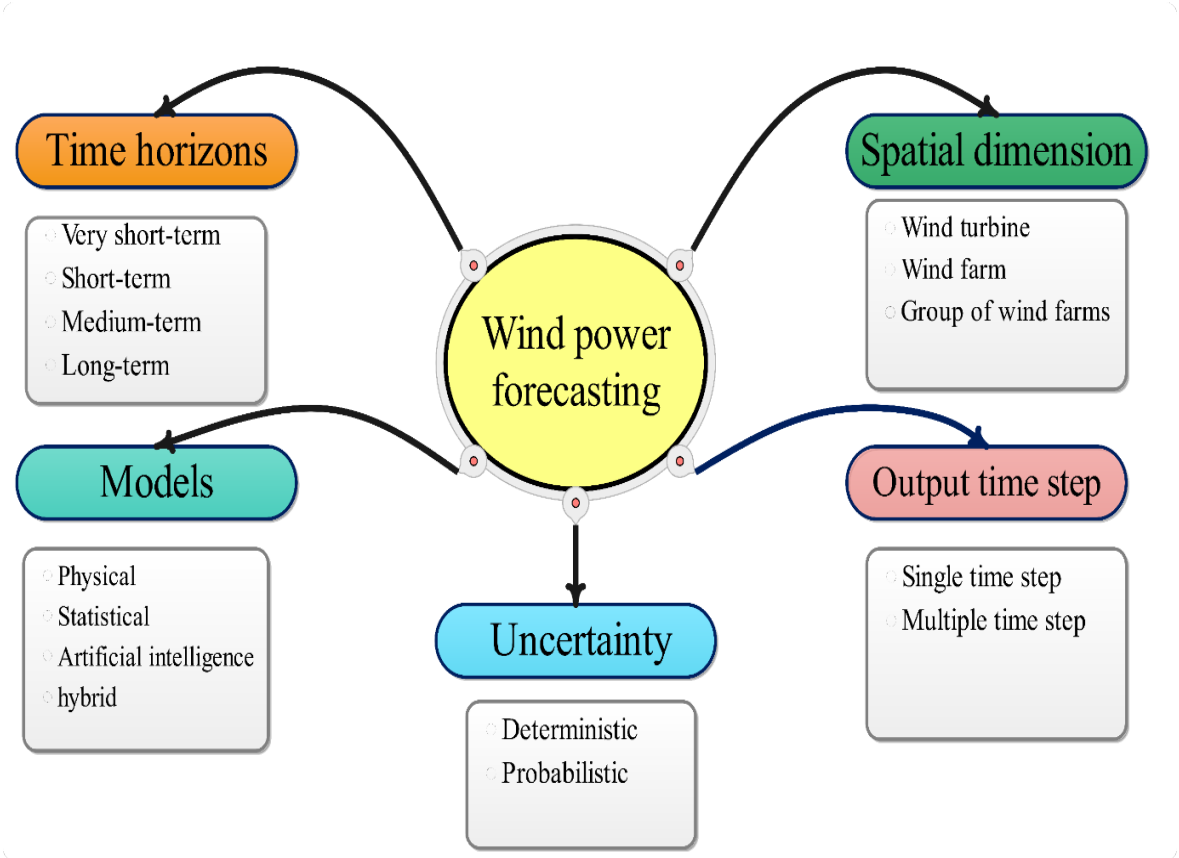



**Figure I.7**: Wind power forecasting classification

### I.7.1 Classification by Time Horizon

One of the most important classification criteria for forecasting wind energy is its time horizon, as it determines the forecasting approach and its applications [12]. The primary time horizons are described in Table I.2, which includes.

- Very Short-Term Forecasting (Seconds to Hours)
- Short-Term Forecasting (Hours to Days)
- Medium-Term Forecasting (Days to Weeks)
- Long-Term Forecasting (Months to Years)

**Table I.2**: Wind Power Forecasting Time Horizon and its Primary Applications

| Forecasting Horizon | Time Range | Typical Focus Area | Primary Applications |
|---|---|---|---|
| **Very Short-Term** | Seconds to Hours | Individual turbines or local systems | - Real-time grid adjustments<br>- Frequency regulation<br>- Load balancing<br>- Immediate system response |
| **Short-Term** | Hours to Days | Wind farms or regional grids | - Day-ahead market operations<br>- Unit commitment<br>- Energy dispatch<br>- Resource scheduling |
| **Medium-Term** | Days to Weeks | Multiple wind farms / Regional level | - Maintenance planning<br>- Medium-term energy trading<br>- Regional generation forecasting |
| **Long-Term** | Months to Years | Regional or national scale | - Strategic planning<br>- Investment analysis<br>- Energy policy-making<br>- Infrastructure development |

### I.7.2 Classification by Spatial Dimension

The spatial dimension of forecasting refers to the geographical scope of the predictions, which can range from individual turbines to large-scale regional assessments. The primary classifications are listed below [14].

- Wind Turbine Level

Forecasting at the wind turbine level focuses on predicting power output for a single turbine. This method is helpful during forecasting in the near future and turbine-specific optimisation. It allows operators to monitor and control individual turbines to maximise efficiency and minimise wear and tear.

- Wind Farm Level

At the wind farm level, forecasting predicts the aggregate power output of an entire wind farm. This is commonly used for short-term forecasting and operational planning. It helps wind farm operators manage the overall performance of the farm and ensure that power generation meets grid requirements.

- Group of Wind Farms

Forecasting for a group of wind farms assesses power generation across multiple wind farms in a region. This approach is suitable for medium-term forecasting and regional grid management.

### I.7.3 Classification by Output Time Step

The output time step refers to the temporal resolution of the forecasting results, which can vary depending on the application. The primary classifications include single and multiple time steps [14].

- Single Time Step

Single-time step forecasting provides predictions for a specific point in time. This approach is commonly used in very short-term forecasting for real-time applications. It is particularly useful for applications that require immediate decision-making, such as grid stability and frequency regulation.

- Multiple Time Steps

Multiple time step forecasting delivers predictions over a sequence of time intervals. This approach is appropriate in short to medium-term forecasts as it offers more thorough view of future power generation. It is commonly used for applications such as energy market operations and maintenance planning.

### I.7.4 Classification by Uncertainty Handling

Uncertainty is an inherent challenge in wind power forecasting given the volatile character of wind energy. Forecasting approaches can be classified depending on the way they manage uncertainty [15].

- Deterministic Forecasting

Deterministic forecasting provides a single predicted value for each time step. This approach is simpler to implement but does not account for uncertainty. It is commonly used in applications where a single, best-estimate forecast is sufficient.

- Probabilistic Forecasting

Probabilistic forecasting delivers a range of possible outcomes with associated probabilities. This approach provides a more comprehensive understanding of potential variations and is Especially effective in decision-making when there is uncertainty. Examples of probabilistic forecasting include quantile regression and ensemble forecasting [15].

### I.7.5 Classification by Model Type

Wind power forecasting techniques are sometimes classed according to the underlying techniques and methodologies used. The primary model types are fellow.

- Physical Models:

Wind power forecasting is grounded in atmospheric physics, numerical weather prediction (NWP), and fluid dynamics, enabling the simulation of wind behaviour based on large-scale meteorological and environmental data. NWP models, such as WRF, ECMWF, and GFS, solve complex equations to predict atmospheric features such as temperature, air pressure, direction and wind speed, making them particularly effective in medium and long-term forecasts spanning several hours to days [14]. Computational Fluid Dynamics (CFD) models extend this capability to microscale analysis by simulating fluid flow over complex terrains, optimising turbine placement, and minimising wake effects in wind farms [16]. Radiative Transfer Models, though primarily designed for solar irradiance studies, also contribute by modelling the impact of solar heating on atmospheric circulation, which influences wind patterns over longer timescales. Similarly, Energy Balance Models assess the flowing of energy between the atmosphere and the surface of the earth to evaluate how variations in surface temperature, humidity, or land use affect wind circulation [15]. While physical models offer predictive capabilities for medium- to long-term horizons and strategic planning, their computational intensity and data requirements can limit their application for highly localised or ultra-short-term forecasts.

- Statistical Models

For wind power forecasting, exploit historical data to identify and model statistical dependencies between past and future wind behaviour, making them particularly effective for short-term horizons ranging from minutes to hours. Classical linear approaches, , include Autoregressive (AR), Moving Average (MA), and combination of them in ARMA, capture temporal dependencies and error structures in stationary time series, while ARIMA and its seasonal extension (SARIMA) address nonstationary and periodic patterns through differencing and seasonal components [14]. Multivariate extensions, such as Vector Autoregression (VAR), and nonlinear formulations, such as the Nonlinear Autoregressive Model with Exogenous Inputs (NARX), incorporate multiple meteorological variables and nonlinear interactions to improve forecast accuracy [17]. Variance-focused methods like Generalised Autoregressive Conditional Heteroskedasticity (GARCH) capture dynamic uncertainty in wind fluctuations, supporting risk assessment [18]. Regression-type approaches

identify the correlations across meteorological inputs and wind power, offering simplicity and interpretability but limited capacity to model complex nonlinearities. Bayesian models provide a probabilistic framework that integrates prior knowledge and updates forecasts as new data become available, enhancing uncertainty quantification. Finally, Exponential Smoothing methods emphasise recent observations, enabling fast adaptation to changing wind conditions and supporting real-time operational decision-making. Statistical models are computationally efficient, straightforward to implement, and well-suited to scenarios with abundant historical data, although their reliance on past patterns can limit performance under rapidly changing atmospheric conditions.

- Artificial Intelligence (AI) models

Established themselves as strong mechanisms for wind energy forecasting, providing the capability of tracking difficult, nonlinear, and dynamic interactions between meteorological data and power. In contrast to traditional statistical techniques, artificial intelligence models such as Artificial Neural Networks (ANNs), support vector machines (SVMs), and deep learning structures such as LSTM networks, or Long Short-Term Memory networks, may identify complex patterns from large amounts of data without the need for predefined physical models [1]. These models excel in integrating multi-source data, handling high-dimensional inputs, and adapting to evolving operational conditions, which is essential for improving forecast accuracy in variable and uncertain wind environments [12].

- Hybrid models

Combines several forecasting approaches to capitalize on their respective advantages while compensating for the limitations of any single method. In wind power prediction, such models can integrate physical principles, statistical techniques, and AI-based methods to represent both the fundamental physical behaviour of the system and the complex nonlinear patterns learned directly from the data [19]. Hybridisation can also involve enhancing a single predictive model with complementary components such as advanced pre-processing methods or optimisation algorithms for parameter tuning. Common strategies include sequential hybridisation, where one method produces preliminary outputs that are refined by another, and parallel hybridisation, which merges the outputs of several models [15].

## I.8 Types of Data for Wind Power Forecasting

The forecasting relies on diverse datasets that capture both the atmospheric conditions driving wind generation as well as the operational and technical features of the wind energy system. Among these, meteorological data are the most critical, as they directly determine the potential

energy available to turbines. Meteorological data can be sourced from on-site measurement stations, remote sensing instruments such as LiDAR and SoDAR, reanalysis datasets like ERA5 [20], or by employing numerical weather prediction (NWP) frameworks, such as the Weather Research and Forecasting (WRF) model [3].

Temporal data provide essential information about recurring patterns and cyclical behaviours in wind speed and power output [3]. Factors like the hour of the day, the specific day of the week, and the season as time series variables allow models to represent both daily and seasonal variations.

Historical power output data form another core input to forecasting models. These records, typically expressed in kilowatts (kW) or megawatts (MW), include turbine status logs and rated capacity information. In supervised learning frameworks, historical power data often serve as the target variable, however, they may also be used as predictors [21]. Incorporating these records enables models to directly learn the mapping between atmospheric conditions and actual turbine output.

Geographical and topographical information further refines forecasting accuracy. Static features such as elevation, geographic coordinates, surface roughness, land cover, and terrain type provide insights into how local landscapes influence wind flow. In large-scale wind farms, turbine layout and spatial configuration must also be considered, as wake effects can create localised reductions in wind speed and subsequent power losses [22].

The technical characteristics of wind turbines encompass parameters such as hub height, rotor diameter, cut-in and cut-out wind speeds, rated power, and the associated power curves define the operational envelope of the wind energy system. These parameters enhance the physical realism of forecasting models and improve generalisation when applying models to new turbines or locations [11].

Finally, numerical weather prediction models, such as WRF, supply forecasted meteorological variables at varying lead times, spatial resolutions, and temporal scales. These data sources are particularly valuable for short-term and day-ahead forecasting, where integrating physics-based outputs with statistical or deep learning approaches has proven effective in boosting predictive skill [1].

### I.9 Overview of AI Models Training Pipeline

The development of artificial intelligence (AI) models follows a systematic training pipeline designed to transform raw data into reliable predictive models. This pipeline provides a structured framework that ensures methodological rigour across different applications. Within

wind power forecasting, the model training process is of paramount importance as it integrates heterogeneous meteorological variables, historical power measurements, and temporal dependencies into an optimised model capable of delivering accurate and robust predictions [23]. The AI training pipeline can be broadly divided into several sequential and interdependent stages.

- Data Input

The initial phase consists of gathering and consolidating input data from pertinent sources. In the context of wind power forecasting, this primarily entails meteorological information features, historical turbine or farm-level power output, and contextual information such as seasonal or geographical features. Ensuring the availability of high-quality data at this stage is essential, as deficiencies in input reliability inevitably propagate through the pipeline [23].

- Data Pre-processing and Feature Engineering

Raw data often contains noise, inconsistencies, or missing values. Therefore, pre-processing techniques such as interpolation, normalisation, standardisation, and outlier removal are utilized to enhance the quality of data [24]. Feature selection and engineering are also critical at this stage, as they enable the conversion of raw input variables into meaningful features that improve the model's learning capability [25]. In forecasting applications, this may include lagged variables, temporal windowing, or derived statistical indicators. The data pre-processing stage enhances training efficiency while simultaneously exerting a direct influence on the model's predictive accuracy [26].

- Model Initialization

Once the dataset is prepared, the model architecture must be defined. This involves selecting the learning paradigm and specifying the structural design. Parameters are initialised either randomly or based on heuristic rules. Proper initialisation is crucial since it affects both the rate of convergence and the stability of the optimization procedure [27].

- Training Phase (Learning Process)

The training phase constitutes the core of the pipeline, during which the model iteratively learns from the training dataset. An optimisation algorithm, most based on gradient descent and its variants, is employed to reduce a predefined loss function that measures the difference between the predicted and actual values. For time-series forecasting, the loss function often incorporates statistical metrics [28].

- Validation and Hyperparameter Tuning

To prevent the occurrence of overfitting, the model's performance it is tested on a validation set separate from the training data. Validation outcomes inform the tuning of hyperparameters. Methods including grid search, random search, or more advanced strategies such as Bayesian optimization are frequently applied. Additionally, early stopping mechanisms may be employed to terminate training when validation error stagnates or increases [29].

- Testing and Performance Evaluation

After training and tuning, the model is subjected to testing on an independent dataset that has not been used in earlier stages. This provides an unbiased estimate of generalisation capability. The evaluation phase typically involves multiple error metrics in wind power forecasting. Selecting appropriate evaluation metrics is crucial, as each metric highlights different, complementary aspects of forecast quality [26].

In summary, the AI training pipeline provides a comprehensive framework that transforms raw meteorological and power data into accurate predictive models. Each stage, ranging from pre-processing to evaluation, it is pivotal for ensuring the reliability of the forecasting system. The following chapters build on this by presenting AI methods including machine learning, deep learning, and hybrid approaches together with their training, evaluation, and limitations in wind power forecasting.

## I.10 Data Pre-processing

Data pre-processing is a fundamental step in the development of reliable and accurate deep learning models for wind power forecasting. Raw data, particularly meteorological and power output data, often contain noise, inconsistencies, and missing values that can negatively affect model performance. Pre-processing seeks to refine and structure raw datasets in a way that maximizes the effectiveness of predictive algorithms [27]. Key approaches frequently applied in this stage encompass exploratory data analysis, error correction, normalization and scaling, extraction of relevant features, decomposition techniques, and division of data into training and testing subsets.

### I.10.1 Data Analysis

Data analysis involves systematically examining and processing raw datasets to derive actionable insights that support model development and evaluation, especially in wind power forecasting. This process typically employs statistical and graphical methods to summarize the dataset, revealing important patterns, distributions, seasonal trends, and anomalies. Frequently

used statistical tools include descriptive measures such as the mean, median, standard deviation, and variance, correlation analysis (Pearson, Spearman), and stationarity tests (Augmented Dickey-Fuller test) [18]. Graphical techniques may include time series plots, histograms, box-and-whisker plots, autocorrelation and partial autocorrelation plots (ACF/PACF), and heatmaps for seasonal or hourly variations [30]. These analyses reveal the underlying structure of the data, identify trends, assess periodicity, detect outliers, and evaluate potential relationships between variables. A thorough data analysis ensures that the dataset is well understood, guiding feature selection, pre-processing strategies, and the development of forecasting models.

### I.10.2 Data Cleaning

A vital pre-processing step to guarantee the precision and dependability of wind power forecasting models. It encompasses identifying and addressing missing values, rectifying inconsistencies, and eliminating outliers that may distort results. This ensures the dataset is accurate, consistent, and prepared for analysis and model training [31].

#### I.10.2.1 Missing Data Imputation

Handling incomplete observations is a recurring challenge in time series forecasting which can arise due to sensor malfunctions, communication errors, data corruption, or logging failures. These missing entries can significantly impact model performance by introducing bias, reducing data quality, and disrupting the temporal dependencies that forecasting models rely on. Thus, the pre-processing of the data, including the careful treatment of missing values, is crucial for accurate modelling [32]. Several imputation techniques exist, ranging from simple statistical methods to more advanced model-based approaches. The selection of an appropriate handling technique is influenced by the amount of missing data, its pattern, and the inherent properties of the time series. Several imputation strategies are commonly employed [27].

- Mean/Median Imputation

A simple statistical approach involves substituting missing values with the mean or median of the available data, as shown in Equation I.9. Median imputation is generally favoured for skewed distributions because it is more robust to outliers [27].

$$x_t = \frac{1}{N}\sum_{i=1}^{N} x_i \; for \; x_t \; missing \tag{I.9}$$

- Forward/Backward Fill

A temporal imputation method where missing entries are substituted with the most recent available value (forward fill) or the next observed value (backward fill). This is particularly effective for short gaps in continuous time series data.

$$x_t = x_{t-1} \vee x_t = x_{t+1} \; for \; x_t \; missing \tag{I.10}$$

- Linear Interpolation

Approximates missing values by interpolating a linear trend between the closest preceding and succeeding known data points. Given time indices $t_1$ and $t_2$ surrounding the missing point t, with corresponding values $x_{t_1}$ and $x_{t_2}$, the imputed value is computed proportionally along the connecting line [27].

$$x_t = x_{t_1} + \frac{(x_{t_2} - x_{t_1})}{(t_2 - t_1)} . (t - t_1) \tag{I.11}$$

- Spline Interpolation

Uses piecewise polynomial functions to achieve smoother transitions between known values, often yielding more realistic estimates than linear interpolation, especially for datasets exhibiting curvature or cyclical variation.

- Time Series Model-Based Imputation:

Fits a statistical forecasting model to the observed data, leveraging trends, seasonality, and autocorrelation structures to estimate missing values in a manner consistent with the temporal dynamics of the series [27].

- Machine Learning-Based Imputation

Employs predictive models to infer missing values based on patterns across multiple correlated variables and time points, making it effective for complex, multivariate datasets [33].

### I.10.2.2 Outlier Detection and Treatment

Outlier detection and treatment are essential steps in preparing wind power datasets, as extreme or inconsistent values can distort model training and degrade forecasting accuracy. Outliers may arise from sensor malfunctions, data transmission errors, or unusual weather events. Methods for identifying outliers are typically grouped into statistical, model-based, and graphical approaches. Statistical techniques, such as the interquartile range (IQR) method [34] or Z-score analysis, identify points that deviate significantly from typical value ranges. Model-

based techniques leverage regression residuals or machine learning algorithms to flag unusual data behaviour. Visual inspection, using tools like boxplots, scatter plots, and time-series plots, often provides intuitive insights into anomalous patterns [35].

Once detected, treatment options depend on the cause and severity of the outlier. Erroneous values resulting from measurement faults may be adjusted through interpolative methods or replaced with approximations based on nearby observations. Legitimate but extreme events, such as unusually strong wind gusts, may be retained to preserve data realism and reflect operational variability [27].

### I.10.3 Normalization and Scaling

For models handling time series data, including wind power forecasting, normalizing, and scaling inputs are critical steps to ensure effective learning. These transformations adjust the range and distribution of features, allowing models to converge faster, learn more effectively, and avoid dominance by variables with large magnitudes [27]. If features are not properly scaled, those with larger values can dominate the learning process, resulting in biased parameter estimates and reduced forecasting accuracy. The following techniques are among the most widely adopted in wind energy data pre-processing were also summarised in Table I.3.

- Min-Max Normalization

Min-max normalization linearly transforms feature values to fit within a specified range, commonly [0, 1] or [-1, 1]. The transformation is defined in Equation I.12.

$$\acute{x} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{I.12}$$

where x is the original value, $x_{min}$ and $x_{max}$ are the minimum and maximum observed values for the feature, and $\acute{x}$ is the scaled result Here, x denotes the original value, $x_{min}$ and $x_{max}$ represent the observed minimum and maximum of the feature, and $\acute{x}$ is the resulting normalized value. This method preserves relationships between data points and can improve optimization convergence but is highly sensitive to outliers, which can distort scaling [36].

- Z-Score Standardization

Also known as standard scaling, adjusts features so that they possess a mean of zero and a variance of one.

$$\acute{x} = \frac{x - \mu}{\sigma} \tag{I.13}$$

Here, $\mu$ represents the feature mean and $\sigma$ its standard deviation. This approach is especially suitable for approximately Gaussian distributions and facilitates comparable variances across features. While it is less affected by differences in magnitude compared to min-max scaling, it remains sensitive to extreme outliers [37].

- Robust Scaling

Robust scaling reduces the influence of outliers by centring features around the median and scaling according to the interquartile range (IQR).

$$\acute{x} = \frac{x - Q_2}{IQR} \tag{I.14}$$

where $Q_2$ is the median, and IQR=$Q_3$–$Q_1$ quantifies the dispersion of the central 50% of the data, this method is ideal for skewed or heavy-tailed distributions, which are common in wind power data due to occasional extreme events [24].

- Logarithmic Transformation

The logarithmic transformation is suitable for variables with right-skewed distributions or spanning multiple orders of magnitude. It compresses large values and expands small ones, stabilizing variance.

$$\acute{x} = \log(x + c) \tag{I.15}$$

where c is a constant added to handle zero values, frequently present in wind power datasets during calm wind conditions [24].

- Max-Absolute Scaling

Max-absolute scaling rescales features by dividing each observation with respect to the feature's largest absolute value using Equation I.16.

$$\acute{x} = \frac{x}{|x_{max}|} \tag{I.16}$$

where $|x_{max}|$ is the largest absolute observed value in the feature; this technique preserves the sign of values, making it particularly effective for directional data such as wind vector components. It is computationally efficient and works well with sparse datasets where most values are near zero [24].

In practice, choosing an appropriate scaling or transformation approach depends on the data statistical properties, the presence of extreme values, and the model specific requirements.

**Table I.3**: Summary of Data Normalization and Transformation Techniques

| Technique | Advantages | Limitations |
|---|---|---|
| **Min-Max Normalization** | -Simple to implement<br>-preserves relationships between values<br>- suitable when feature distribution is unknown. | -Sensitive to outliers<br>-Can distort scaling if extreme values are present. |
| **Z-Score Standardization** | -Handles features with different units<br>-less affected by outliers than min-max<br>-Maintains distribution shape. | -Assumes Gaussian distribution<br>-Extreme outliers can still influence results. |
| **Robust Scaling** | -Robust to extreme values<br>-Preserves distribution shape. | - May not be ideal for normally distributed data without outliers. |
| **Logarithmic Transformation** | -Handles skewed distributions<br>-Reduces influence of extreme values. | - Cannot be applied to zero or negative values<br>-Transformation may distort small values. |
| **Max-Absolute Scaling** | -Preserves sparsity in data<br>-Simple and fast to compute. | -Sensitive to outliers -Does not centre the data. |

### I.10.4 Feature Extraction

Feature extraction denotes the transformation of raw inputs into a smaller, more informative subset of features while retaining the essential structure of the dataset. This step important where meteorological and operational datasets tend to be high-dimensional and exhibit strong correlations [25]. By extracting a smaller number of representative features, the learning process can be made more efficient and less affected by redundancy.

- Principal Component Analysis (PCA)

It is a commonly applied linear method for extracting relevant features. PCA transforms the original dataset into a new coordinate space spanned by orthogonal principal components, obtained through eigenvalue decomposition of the covariance matrix. The first component represents the direction of maximum variance, followed by subsequent components that capture decreasing variance while remaining orthogonal to the previous ones [25]. Formally, for a data matrix, the covariance matrix is given in Equation I.17.

$$C=\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)(x_i-\mu)^T \tag{I.17}$$

where $x_i$ denotes the $i^{th}$ observation and μ the mean vector. Eigenvectors of **C** form the principal directions, with the eigenvalues reflecting the proportion of variance captured by each component.

Within the domain of wind power forecasting, PCA is typically used to correlated meteorological inputs. A limited number of principal components can then be retained to represent the dominant variability in the dataset, reducing dimensionality while preserving key patterns for model training [25].

### I.10.5 Data Decomposition

Data decomposition is a powerful pre-processing approach that breaks complex time series into simpler, more interpretable subcomponents. In raw wind speed or power data often exhibit non-stationary patterns, stochastic fluctuations, and measurement noise, which can obscure underlying trends and hinder model training by breaking down the original signal into multiple component sub-series, each representing distinct temporal or frequency characteristics such as trend, seasonality, or high-frequency oscillations, forecasting models can be tailored to process each component separately [12].

- Empirical Mode Decomposition (EMD)

EMD adaptively separates nonlinear and non-stationary signals into multiple oscillatory components, referred to as Intrinsic Mode Functions (IMFs), together with a remaining trend. The IMFs are extracted iteratively, the initial IMF captures the fastest oscillatory patterns, with subsequent IMFs reflecting progressively slower variations. The final residual captures the long-term trend [38]. EMD strength lies in its ability to represent multi-scale behaviour without assuming linearity or stationarity. However, it is sensitive to noise It may also experience mode mixing, in which multiple frequency components are combined within a single IMF [10].

- Ensemble Empirical Mode Decomposition (EEMD)

EEMD improves upon EMD by addressing mode mixing through noise-assisted analysis. It repeatedly introduces independent white noise to the signal, applies EMD to each perturbed version, and averages the derived IMFs, thereby reducing mode mixing. While more robust than standard EMD, EEMD introduces additional computational cost and may still retain residual noise [39].

- Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN)

CEEMDAN improves decomposition stability by sequentially extracting each IMF from the residual of the preceding step while incorporating adaptive noise at every iteration. This

approach reduces noise interference in the reconstructed signal and provides a more faithful representation of the original data. Recognized as a state-of-the-art EMD variant for complex, non-stationary time series, CEEMDAN offers enhanced reconstruction accuracy and effectively mitigates mode mixing compared with earlier techniques [40].

- Wavelet Transform (WT)

The WT decomposes signals into components localized in both time and frequency, making it particularly suitable for non-stationary, transient-rich data such as wind speed or power. Unlike the Fourier Transform, which only conveys global frequency information, the WT utilizes scaled and shifted mother wavelets to capture both rapid, high-frequency fluctuations and slower, low-frequency trends. While the Continuous Wavelet Transform (CWT) provides detailed multi-resolution insights, the Discrete Wavelet Transform (DWT) is generally preferred in forecasting due to its superior computational efficiency [41].

- Variational Mode Decomposition (VMD)

VMD constitutes a more recent development in decomposition techniques, formulated as an optimisation problem. It separates the signal into a fixed number of modes, each characterized by a distinct centre frequency and constrained bandwidth. Unlike EMD-based methods, VMD operates in a non-recursive manner and is inherently robust to noise. By enforcing compactness of each mode in the frequency domain, VMD offers enhanced separation of overlapping spectral components and improved resilience to measurement errors [4].

### I.10.6 Data Splitting

Data splitting is an essential pre-processing step, ensuring robust model evaluation and avoiding overly optimistic performance estimates. This process entails dividing the dataset into separate subsets for training, validation, and testing [42]. Unlike typical machine learning datasets, where random shuffling is often permissible, time series data is inherently ordered in time. As a result, splitting strategies must preserve temporal structure to prevent the inadvertent influence of future data on historical predictions, known as data leakage [42]. Table I.4 shows details on common Cross-Validation methods.

Formally, let the time series be represented in Equation I.18.

$$\{X = x_1, x_2 \ldots . x_N\} \tag{I.18}$$

In this context, N represents the total number of observations, and data splitting seeks to separate X into non-overlapping partitions such that training sets precede their corresponding testing sets in time.

➢ Traditional Train-Test Split

In the simplest approach, the dataset is split into a training set and a testing set, where the training set is employed to build the model, and the testing set assesses its predictive performance on unseen data. Temporal ordering is strictly preserved.

$$X_{train}=\left\{x_1, x_2 \ldots, x_{N_{train}}\right\}, X_{test}=\left\{x_{N_{train+1}}, \ldots, x_{N_{test}}\right\} \quad \text{(I.19)}$$

where $N_{train}$ and $N_{test}$ denote the final indices of the training and testing subsets, offering a method that is computationally efficient but may yield unstable estimates, especially for non-stationary time series [42].

➢ Time Series Cross-Validation (Rolling Forecast Origin)

To address the variability inherent in a single division between training and testing sets, Time Series Cross-Validation (TSCV) creates multiple splits, each respecting temporal ordering. One of the most widely used strategies is Rolling Forecast Origin. Here, the training set progressively expands, while the testing set consists of a fixed forecast horizon h [42].

For the k-th split.

$$X_{train}^{(k)}=\left\{x_1, x_2 \ldots, x_{N_k}\right\}, X_{test}^{(k)}=\left\{x_{N_k+1}, \ldots, x_{N_k+h}\right\} \quad \text{(I.20)}$$

Here $N_k$ represent an endpoint of the training set in split k, and h is constant across all splits. This method evaluates the model over multiple forecasting periods, reducing bias in performance estimates [28].

➢ Expanding Window Cross-Validation

Within Expanding Window Cross-Validation, the training subset starts from an initial portion and expands iteratively, without omitting any earlier data points. Where testing set consists of the next h future observations.

Formally, for fold k.

$$X_{train}^{(k)}=\left\{x_1, x_2 \ldots, x_{N_k}\right\}, X_{test}^{(k)}=\left\{x_{N_k+1}, \ldots, x_{N_k+h}\right\} \quad \text{(I.21)}$$

where $N_k > N_{k+1}$. This strategy provides notable advantages when more historical data is expected to improve forecast accuracy, as it mirrors the gradual accumulation of observations in real-world forecasting scenarios [28].

➢ Walk-Forward Validation

Walk-Forward Validation is a dynamic evaluation strategy where both the training and testing windows move forward in time. At each iteration, the model is trained on data up to time $N_k$ and

tested on the subsequent h points. After evaluation, the window is shifted forward, and the process repeats.

For fold k.

$$X_{train}^{(k)} = \left\{ x_{s_k}, \ldots, x_{N_k} \right\}, X_{test}^{(k)} = \left\{ x_{N_k+1}, \ldots, x_{N_k+h} \right\} \quad \text{(I.22)}$$

Here $s_k$ denote the starting index of the training window, which may remain fixed (anchored walk-forward) or move forward (sliding walk-forward). This approach is particularly suited to operational forecasting systems where models are continually updated as new observations become available [28].

**Table I.4**: Overview of Common Cross-Validation Methods for Time Series Forecasting

| Method | Description | Advantages | Limitations |
|---|---|---|---|
| **Traditional Train-Test Split** | Divides the dataset into training and testing subsets, with no overlap. | Simple and fast to implement. | May not represent temporal dependencies; results depend on split choice. |
| **K-Fold Cross-Validation (Blocked)** | Divides the dataset to $K$ contiguous folds, each used once as a test set. | Minimizes variability in performance evaluation. | May still leak temporal information if not blocked; computationally heavier. |
| **Time Series Cross-Validation (Rolling Forecast Origin)** | Uses an expanding training window and a fixed-size test set that moves forward in time. | Mimics real forecasting scenario; preserves temporal order. | Computationally intensive; increasing training size over time. |
| **Expanding Window Cross-Validation** | Like rolling forecast origin but always starts from the first observation, expanding the training set at each step. | Uses all available past data; realistic for incremental learning. | Large training sets may become computationally expensive. |

## I.11 Training Fundamentals: Optimisation and Parameters

The successful training of artificial intelligence (AI) models for wind energy forecasting relies on both the quality of the input data and pre-processing approaches, as well as the choice of training procedures and hyperparameter configurations [29]. These elements determine the model ability to converge efficiently and provide accurate predictions under varying conditions. This section provides an overview of the essential training fundamentals, including optimisation methods, activation functions, learning rate scheduling, batch configuration, and iterative training strategies [29].

### I.11.1 Optimisation Algorithms

Optimisation algorithms constitute the core of training artificial intelligence models, as they govern how the model parameters are updated to minimise the defined loss function.

The effectiveness of an AI model is strongly dependent on the choice of optimisation strategy, since it has a direct impact on convergence speed, algorithmic stability, and the accuracy of the outcome. In wind energy forecasting, where datasets are often nonlinear, noisy, and temporally correlated, a robust optimisation algorithm ensures so that the model identifies the inherent patterns while avoiding overfitting or diverging [29]. The goal of an optimisation algorithm is to iteratively update the weights and biases of a neural network to align the predicted outputs with the true values.

**Table I.5**: The Main Categories of Optimization Algorithms

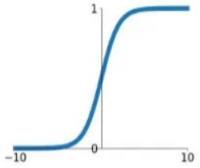
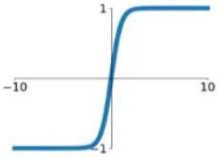
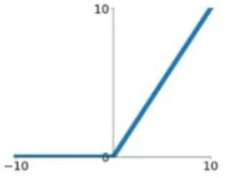
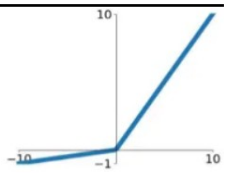
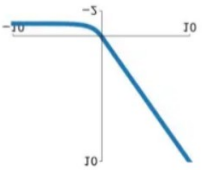
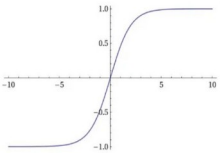
| Category | Description | Representative Algorithms | Key Characteristics |
|---|---|---|---|
| **First-Order Gradient Methods** [43] | Utilize the first derivative (gradient) of the loss function with respect to model parameters. Efficient and scalable for large datasets. | - Gradient Descent (GD): Uses the full dataset for updates, precise but computationally expensive. - Stochastic Gradient Descent (SGD): Updates with one sample per iteration, efficient but noisy. | Scalable, computationally efficient, susceptible to local minima and oscillations. |
| **Momentum-Based Methods** [44] | Extend gradient descent Using information gathered from past gradients to accelerate convergence and decrease oscillations. | - Momentum: Adds inertia term to smooth parameter updates. - Nesterov Accelerated Gradient (NAG) [45]: Anticipates future position, often yielding faster convergence. | Faster convergence, reduces oscillations, widely used in deep learning. |
| **Adaptive Learning Rate Methods** | Dynamically adjust learning rates for each parameter, enabling better handling of sparse or heterogeneous data. | - Adagrad [46]: Adjusts learning rate based on accumulated gradients, effective for sparse data. - Adam (Adaptive Moment Estimation) [47]: Combines momentum and RMSProp, offering robust performance in deep learning. | Adaptive, efficient for complex problems. Adam is among the most widely used in practice. |
| **Second-Order Methods** [48] | Use second derivatives (Hessian) or approximations | - Newton's Method: Uses Hessian for exact updates. - Quasi-Newton (e.g., BFGS, L-BFGS): Approximate curvature for efficiency. | High precision, Memory-intensive, less common in large-scale deep learning. |

Formally, this process strives to reduce a loss function $L(\theta)$, with $\theta$ denotes the model set of parameters. The Main Categories of optimisation algorithms and descriptions, and the principal characteristics are presented in Table I.5.

### I.11.2 Activation Functions

Activation functions play a pivotal role in AI models, especially when it comes to neural networks, by introducing non-linearity into the learning process.

**Table I.6**: The most used Activation Functions for time series forecasting

| Category | Function | Mathematical Expression | Key Characteristics | |
|---|---|---|---|---|
| **Sigmoid** | Logistic Function | $f(x)=\frac{1}{1+e^{-x}}$ | Maps values to (0,1). Suitable for probabilities but suffers from vanishing gradients. |  |
| **Hyperbolic Tangent (tanh)** | Scaled Sigmoid | $f(x)=\tanh(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$ | Outputs in (-1,1). Zero-centered but still prone to vanishing gradients. |  |
| **ReLU(Rectified Linear Unit)** | Piecewise Linear | $f(x)=max(0,x)$ | Simple, efficient, mitigates vanishing gradients, but can cause "dying ReLU" (neurons stuck at 0). |  |
| **Leaky ReLU** | ReLU Variant | $f(x)=\begin{cases} x, x>0 \\ \alpha x, x\leq 0 \end{cases}$ | Allows small negative slope (α\alpha) to avoid dying ReLU. |  |
| **ELU (Exponential Linear Unit)** | Smooth ReLU Variant | $f(x)=\begin{cases} x, x>0 \\ \alpha(e^{x}-1), x\leq 0 \end{cases}$ | Provides smooth negative outputs, improving convergence. |  |
| **Softmax** | Normalized Exponential | $f(x)=\frac{e^{x_i}}{\sum_j e^{x_j}}$ | Converts outputs into probability distributions across multiple classes. |  |

Without the use of activation functions, neural networks would fundamentally operate as linear models, irrespective of their depth and thus fail to learn and predict accurately [49]. By converting the weighted average of inputs to a nonlinear output, activation functions enable

deep learning models to approximate highly intricate mappings between inputs [49]. Table I.6 describe the main used activation functions

Mathematically, an activation function $f(\cdot)$ maps the weighted input $x$ (where $x = W \cdot h + b$ as weights, h as the previous layer's outputs, and b as bias) to the output of a neuron $a = f(x)$ where $x$ is the pre-activation input (weighted sum of inputs + bias), $f(\cdot)$ is the non-linear transformation and $a$ is the neuron's activated output.

### I.11.3 Epochs, Iterations, Batch Size, and Learning Rate

The training of artificial intelligence (AI) models involves iterative optimisation processes where data are repeatedly presented to the model to minimise prediction errors. Four fundamental concepts that govern this process are epochs, iterations, batch size, and the learning rate. A precise understanding and careful configuration of these parameters are crucial for ensuring efficient convergence, avoiding underfitting or overfitting, and achieving high predictive accuracy in wind power forecasting applications.

- **Epochs:** are the number of complete sweeps over the training dataset. Each epoch reflects a complete cycle during which the model processes all available training samples and updates its parameters accordingly. Generally, raising the range of epochs permits the algorithm to acquire more intricate representations. However, excessive epochs may lead to overfitting, where the model memorises noise instead of generalising to unseen data [50].
- **Iterations:** denote the number of parameter updates that occur within a single epoch. The number of iterations per epoch depends directly on the chosen batch size, defined as the subset of training samples processed before a parameter update is performed [50]. Specifically, if the training dataset contains N samples and the batch size is B, the count of iterations each epoch is defined via {I} rsub {e} = B/N.

- **Batch Size:** plays a key role in balancing computational efficiency and gradient stability. A small batch size increases the stochasticity of gradient estimates, which can help escape local minima but introduces higher variance in updates. Conversely, a large batch size provides smoother gradients and computational efficiency due to vectorisation but may reduce generalisation capacity. In practice, mini-batch training (with batch sizes ranging from 16 to 512) is widely adopted since it achieves a combination of efficiency and stability [50].

- **Learning Rate:** adjusts the step size while parameters updates in the optimisation process. A growth in learning rate might create divergence, preventing the algorithm from converging, In contrast, an insufficient learning rate may result in excessively slow training and suboptimal convergence [50].

In summary, the interplay among these hyperparameters defines the training dynamics. For wind power forecasting, these hyperparameters tuning becomes even more critical for AI models.

### I.11.4 Overfitting Issue

Overfitting occurs whenever the model a learning model absorbs not just the underlying patterns but also the noise and random oscillations, causing low generalisation on unseen data. This issue is particularly critical in deep learning, where highly flexible models with a considerable number of parameters are prone to memorising training examples rather than learning the true functional relationships. Detecting overfitting typically involves monitoring the gap between training and validation errors, a steadily decreasing training error coupled with an increasing validation error is a common indicator [51].

To mitigate overfitting and enhance model generalisation, several regularisation techniques are applied. Dropout deactivates a portion of neurons at random during training, lowering feature co-adaptation. and promoting more robust representations. Early stopping monitors validation performance and halts training once no further improvement is observed, preventing the model from over-specialising. Batch normalisation normalises intermediate layer outputs, which stabilises training while introducing a mild regularisation effect [51]. Data augmentation artificially increases the diversity of training samples through transformations, resampling, or noise injection, thereby reducing the model's tendency to memorise specific examples [51].

## I.12 Conclusion

This chapter has outlined the fundamental aspects of wind energy and highlighted the central role of forecasting in addressing its variability and intermittency. By examining the conversion process, system components, and integration challenges, it became evident that the reliability of wind energy as a sustainable resource depends heavily on the accuracy of forecasting methods. Accurate predictions not only improve grid stability but also optimise operational planning and market participation, thereby reinforcing the importance of continuous progress in this area. The discussion further emphasised the diversity of forecasting approaches, from statistical and

physical models to more sophisticated AI-based approaches. It was shown that the outcome of such approaches is highly dependent on the quality of input data and pre-processing stages, which serve as the foundation for extracting meaningful patterns and ensuring model robustness. As forecasting horizons extend and data complexity increases, the limitations of traditional methods highlight the need for innovative approaches capable of handling nonlinear dynamics and uncertainty.

In conclusion, the insights provided in this chapter establish a solid framework for understanding the challenges, processes, and methodological landscape of forecasting. This foundation prepares the ground for the subsequent chapters, where AI and hybrid models, training strategies, evaluation techniques, and research gaps will be explored in greater depth.

# Chapter II: AI Models and Frameworks for Wind Energy Forecasting

# AI Models and Frameworks for Wind Power Forecasting

## II.1 Introduction

Artificial Intelligence (AI) has emerged as a cornerstone technology in advancing renewable energy forecasting, particularly in wind power prediction. Traditional statistical and physical models often struggle to capture the highly nonlinear and stochastic nature of wind behaviour. AI techniques, by contrast, excel in identifying hidden patterns, learning complex relationships, and generalising from large datasets, making them particularly suitable for both short-term and long-term forecasting tasks [23].

This chapter provides a comprehensive overview of AI models applied in wind power forecasting, covering approaches from conventional machine learning to cutting-edge deep learning models and hybrid systems. Machine learning models, such as regression-based methods, decision trees, and ensemble techniques, have demonstrated effectiveness in modelling historical dependencies and improving forecast accuracy. Unsupervised learning methods, including clustering and dimensionality reduction, play a crucial role in data pre-processing and feature extraction [52]. Building on these foundations, deep learning models such as recurrent neural networks (RNN), long short-term memory networks (LSTM), gated recurrent units (GRU), and convolutional neural networks (CNN) have further advanced forecasting accuracy by leveraging temporal and spatial correlations in wind data [19].

In addition to standalone models, hybrid approaches combine several models to exploit complementary advantages while overcoming the limitations inherent to single methods, enhancing robustness and adaptability [23]. This chapter also examines the training and learning process, including optimisation strategies and hyperparameter tuning, followed by evaluation techniques and benchmarking practices that ensure reliable and reproducible results. Finally, the chapter identifies research gaps in the current body of work, highlighting existing methods limitations and outlining how this thesis contributes to addressing these challenges.

## II.2 Machine learning Models for wind power forecasting

Machine learning (ML) methods have been widely applied in wind power forecasting due to their ability to model nonlinear relationships between meteorological variables and power output. Unlike purely statistical approaches, ML techniques rely on data-driven learning to automatically extract patterns and adapt to varying conditions. They are particularly more effective in handling large, heterogeneous datasets and capturing complex dynamics in wind behaviour. Depending on the learning paradigm, as Figure II.1 illustrates, ML methods are typically grouped into supervised and unsupervised categories [53] [52].

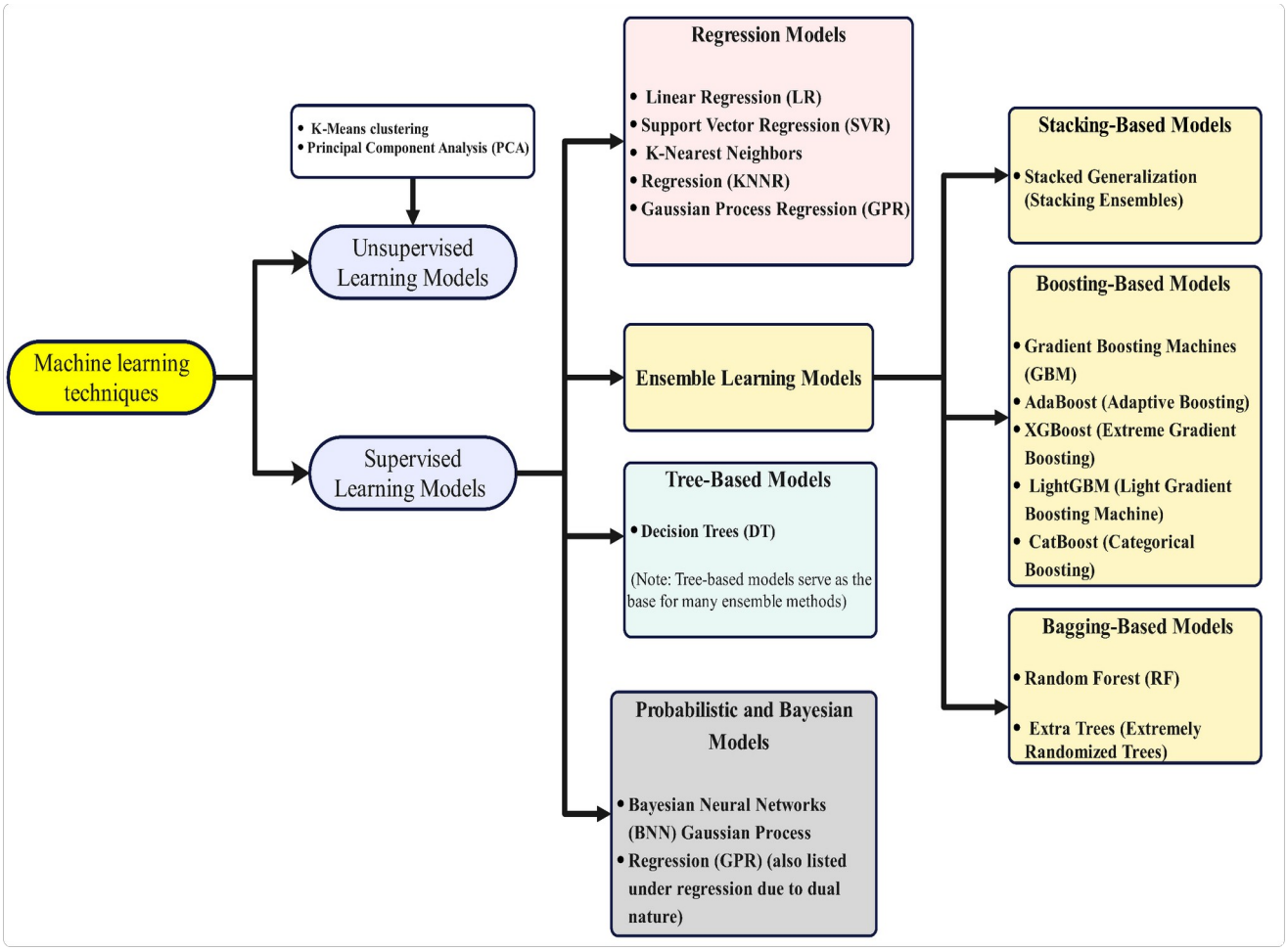



**Figure II.1**: Machine learning-based forecasting techniques

### II.2.1 Supervised Learning Models

Supervised learning models form the backbone of ML-based forecasting techniques. These models operate by mapping input variables. The training process involves minimising a loss function that quantifies prediction errors. Supervised learning in machine learning models can be broadly categorised into regression-based models, tree-based models, Bayesian and probabilistic models, ensemble models, and neural networks [53]. In the context of wind energy

forecasting, supervised learning models' performance depends on data quality and hyperparameter tuning. Below, we provide a detailed mathematical description and explanation of key supervised learning models.

### II.2.1.1 Regression-Based Models

Regression-based models are widely used in wind speed and power forecasting to establish mathematical relationships between input meteorological variables, such as wind speed, air pressure, and temperature, and the target output, which is typically wind power. These models range from simple linear regressions to more advanced machine learning techniques [54].

- Support Vector Regression (SVR)

SVR is a supervised learning technique derived from the principles of Support Vector Machines (SVM), originally developed for classification tasks. SVR is particularly effective for modelling nonlinear relationships between input features and target variables. Instead of minimising the mean squared error, SVR constructs a hyperplane that best fits the data while maintaining a margin ε, within which predictions are considered acceptable [55].

The core idea of SVR is to find a function f(x) that approximates the relationship between inputs $x$ and outputs $y$ with a maximum deviation of ε from the true values, while simultaneously keeping the model complexity minimal. Mathematically, SVR model function is given by Equation II.1.

$$\hat{y}(x)=\sum_{i=1}^{N}(\alpha_i-\alpha_i^{¿})k(x_i,x)+b \tag{II.1}$$

Where $\hat{y}(x)$ denotes the predicted wind power output given the input features. The coefficients $\alpha_i$ and $\alpha_i^{¿}$ are Lagrange multipliers associated with the support vectors, which define the regression function. The kernel function $k(x_i,x)$ maps the input data into a higher-dimensional feature space, enabling SVR to model both linear and nonlinear relationships. The choice of kernel commonly the Radial Basis Function (RBF) is particularly important for capturing the nonlinear variability of wind speed. The bias term b adjusts the position of the regression hyperplane within this transformed space [56].

- Random Forest Regression (RFR)

RFR represents an ensemble approach that leverages a collection of decision trees to improve forecast precision while mitigating overfitting. Instead of relying on a single tree, RFR

generates multiple trees using different subsets of the dataset and averages their predictions. The prediction formula for RFR is in Equation II.2.

$$\hat{y}(x)=\frac{1}{M}\sum_{i=1}^{M} T_m(x) \tag{II.2}$$

where $\hat{y}(x)$ is the final predicted wind power value, M denotes the number of trees in the forest, and $T_m(x)$ represents the prediction made by the $i^{th}$ decision tree. The number of trees M is a key hyperparameter that influences both accuracy and stability. The strength of RFR lies in its ability to reduce variance by aggregating the predictions of diverse models.

- K-Nearest Neighbors Regression (KNNR)

KNNR is a non-parametric model that predicts wind speed or power by exploiting the similarity between the current input and historical observations [57]. Unlike parametric models that learn an explicit function, KNNR operates by searching for the k closest data points $y_i$ in the training set and estimating the output as the average of their corresponding target values. Mathematically, the prediction is expressed in Equation II.3.

$$\hat{y}(x)=\frac{1}{k}\sum_{i=1}^{k} y_i \tag{II.3}$$

The similarity between data points is usually determined using the Euclidean distance in Equation II.4.

$$d(x,x_i)=\sqrt{\sum_{j=1}^{p}(x_j,x_{i,j})^2} \tag{II.4}$$

where $d(x,x_i)$ represents the distance between the input point x and a training point $x_i$. The summation runs over all p input features, capturing the multidimensional nature of the wind forecasting problem. KNNR is advantageous because it does not require assumptions about data distribution and can model nonlinear dependencies. However, its computational cost increases with large datasets [57].

- Gradient Boosting Machines (GBM)

GBM present an ensemble learning method in which decision trees are generated in sequence, each one learning to correct the errors remaining from the previous tree [31]. Unlike Random Forest, where trees are constructed independently, GBM improves its performance iteratively through gradient-based optimisation of a loss function. The objective function minimised by GBM is expressed in Equation II.5 [31].

$$L=\sum_{i=1}^{N} l(y_i,\hat{y}_i)+\sum_{k=1}^{k} \Omega(f_k) \quad \text{(II.5)}$$

Here, L denotes the overall loss function, quantifying the discrepancy between observed wind power values and their predictions. The first term, $\sum_{i=1}^{N} l(y_i,\hat{y}_i)$ accounts for the individual errors of each training example, where $y_i$ is the observed wind power and $\hat{y}_i$ is the predicted output. The second term, $\sum_{k=1}^{k} \Omega(f_k)$ serves as a regularization function that prevents overfitting by controlling the model complexity. The number of boosting iterations is denoted by K, which is a crucial parameter that determines how many weak learners (decision trees) contribute to the final prediction which can be expressed in Equation II.6.

$$\hat{y}_i(x)=\sum_{k=1}^{k} f_k(x_i), f_k \in F \quad \text{(II.6)}$$

Where, $\hat{y}_i(x)$ denotes the predicted value for input $x_i$, and each $f_k$ represents a weak learner (typically a decision tree) selected from the function space $F$. The number of boosting iterations, K determines how many trees are combined in the ensemble [31].

### II.2.1.2 Tree-Based Models

These models partition the input space through successive splits on feature values, producing a hierarchical tree structure that isolates subsets of data exhibiting comparable target behaviours [31]. They are particularly effective in dealing with heterogeneous meteorological data, as they can capture interactions between multiple features without requiring extensive pre-processing. The primary advantage of tree-based models lies in their interpretability and ability to model nonlinearity without requiring feature transformations [58]. However, they may suffer from overfitting if not properly regularised. The three most used tree-based models in wind power forecasting are Decision Trees (DT), Random Forest (RF), and Extreme Gradient Boosting (XGBoost).

- Decision Trees (DT)

DTs are non-parametric supervised learning models that partition the input space into hierarchical regions to predict continuous outputs. In regression tasks such as wind speed or wind power forecasting, DTs operate by recursively splitting the dataset into subsets based on input features until homogeneity in the target variable is maximised within each region [31]. At each internal node, the algorithm selects the feature and split point that minimises an impurity function, commonly the Mean Squared Error (MSE), defined in Equation II.7 [58].

$$MSE=\frac{1}{N}\sum_{i=1}^{N}(y_i,-\overline{y})^2 \quad \text{(II.7)}$$

where $y_i$ is the actual target value, $\overline{y}$ is the average target value in the node, and N is the number of samples in that node. Once the tree is constructed, the prediction for a new input is obtained by following the splits until reaching a terminal leaf node. The model output is computed as the mean of the training observations assigned to that leaf, consistent with Equation II.8.

$$\hat{y}=\frac{1}{N_L}\sum_{i=1}^{N_L}y_i \quad \text{(II.8)}$$

$N_L$ is the samples number in the corresponding leaf node.

While DT can capture nonlinear relationships between meteorological inputs and wind power output, they are prone to overfitting when grown too deeply [58].

- Random Forest Regression (RFR)

As an ensemble learning approach RFR constructs a set of decision trees and aggregates their predictions to enhance accuracy. Each tree is trained using a bootstrap sample of the training dataset, while a random subset of features is considered at each split, which introduces diversity within the ensemble. For regression tasks, The final estimate is derived by computing the mean of the predictions from all individual trees Equation II.9 [59].

$$\hat{y}(x)=\frac{1}{M}\sum_{m=1}^{M}T_m(x) \quad \text{(II.9)}$$

where $\hat{y}(x)$ denotes the predicted value, $T_m(x)$ represents the prediction from the $m^{th}$ tree, and M is the number of trees in the forest. This averaging process reduces variance compared to a single decision tree, making the model more robust to noise and fluctuations in meteorological data. The incorporation of randomness in both data sampling and feature selection prevents overfitting, a common limitation of individual decision trees. Random Forest can be computationally demanding when a large number of trees are required, especially for high-dimensional datasets [59].

- Extreme Gradient Boosting (XGBoost)

XGBoost is an efficient boosting-based ensemble technique in which decision trees are built in sequence, with each tree designed to reduce the residual errors left by its predecessors. In

contrast to Random Forest, whose trees are trained independently, XGBoost jointly optimises prediction accuracy and model complexity by minimising a regularised objective function [58]. The objective function is defined in Equation II.10.

$$L(\varnothing)=\sum_{i=1}^{N} l(y_i,\hat{y}_i)+\sum_{k=1}^{k} \Omega(f_k) \quad \text{(II.10)}$$

where $l(y_i,\hat{y}_i)$ measures the loss between the true wind power value $y_i$ and the prediction $\hat{y}_i$ (commonly squared error for regression), and $\Omega(f_k)$ is the regularization term applied to each tree $f_k$ penalizing excessive depth and complexity to mitigate overfitting.

The model prediction is updated iteratively using Equation II.11 [60].

$$\hat{y}_i^t(x)=\hat{y}_i^{t-1}+\eta f_t(x_i) \quad \text{(II.11)}$$

where $\eta$ is the learning rate that controls the contribution of the newly added tree, and $f_t(x_i)$ represents the output of the $t^{th}$ tree for input $x_i$.

### II.2.1.3 Bayesian and Probabilistic Models

Bayesian and probabilistic models offer a distinct advantage in wind power forecasting because they provide not only predictions but also a quantification of the associated uncertainty [61]. Unlike deterministic approaches that produce a single estimate, Bayesian methods deliver a probabilistic distribution over potential results [62].

The general predictive distribution in Bayesian regression is expressed by Equation II.12.

$$p(\hat{y}|x,D)=\int p(\hat{y}|x,w)\,p(w|D)\,dw \quad \text{(II.12)}$$

where $\hat{y}$ denotes the predicted output, x is the input feature vector, w represents the model parameters, and D is the training dataset. The term $p(w|D)$ is the posterior distribution of the parameters given the data, and $p(\hat{y}|x,w)$ corresponds to the likelihood of the prediction under a specific parameter configuration [61]. This formulation shows that the prediction accounts for all possible parameter values, weighted by their posterior probabilities.

Among probabilistic approaches, Gaussian Process Regression (GPR) [1]. GPR models define a distribution over functions, allowing flexible non-linear regression while maintaining a probabilistic framework. A Gaussian process is completely defined by its mean function $m(x)$ representing the anticipated value of the function at $x$ and the corresponding covariance (kernel) function $k(x,x')$ which defines how similar or correlated the function values are at two input points $x$ and $x'$. A Gaussian Process can be defined in Equation II.13 [1].

$$f(x) \ \varepsilon \mathcal{P}(m(x), k(x, x')) \tag{II.13}$$

### II.2.1.4 Ensemble Models

As an approach from machine learning family, Ensemble models combine multiple individual predictors, referred to as base learners, to produce a more accurate and robust prediction than any single model alone [63]. The motivation behind ensemble methods is the reduction of variance, bias, or noise by leveraging the diversity of different learners. The three major ensemble strategies are stacking, bagging, and boosting [63].

- Stacking

Stacking is an ensemble method that integrates multiple base learners by training a meta-model to optimally combine their predictions [62]. The general prediction function for stacking can be expressed in Equation II.14.

$$\hat{y}(x) = f_{meta}(f_1(x), f_2(x), \ldots, f_M(x)) \tag{II.14}$$

Where $f_1(x), f_2(x), \ldots, f_M(x)$ are the predictions from the M base models, each trained on the same input feature vector x and $f_{meta}$ represents the meta-learner, which learns how to optimally weight and combine the outputs of the base models while $\hat{y}(x)$ is the final forecasted value [62].

- Bagging

Bagging (Bootstrap Aggregating) improves prediction stability by training multiple base models on different bootstrap samples of the training dataset [58]. Each model generates predictions independently, and the final output is obtained by averaging. The general prediction function can be written mathematically in Equation II.15.

$$\hat{y}(x) = \frac{1}{M} \sum_{m=1}^{M} f_m(x) \tag{II.15}$$

Where $f_m(x)$ is the prediction from the $m^{th}$ base learner trained on a bootstrap sample of the dataset and M is the total number of base learners and $\hat{y}(x)$ is the aggregated prediction, typically the mean for regression tasks. Bagging reduces variance by averaging predictions [58].

- Boosting

Boosting sequentially trains base learners, where each subsequent model focuses on correcting the errors of its predecessor. This creates a strong ensemble from multiple weak learners [58]. The prediction function in boosting is expressed in Equation II.16.

$$\hat{y}(x)=\sum_{m=1}^{M}\alpha_m f_m(x) \tag{II.16}$$

where $f_m(x)$ is the prediction from the $m^{th}$ weak learner and $\alpha_m$ is the weight assigned to the $m^{th}$ learner, reflecting its relative importance in reducing errors, while M is the total number of learners and $\hat{y}(x)$ is the weighted sum of all learners predictions [58].

### II.2.2 Unsupervised Learning Models

Unsupervised learning models differ from supervised methods in that they do not rely on labelled data consisting of input-output pairs. Instead, they aim to uncover hidden structures, patterns, or groupings in the data by analysing its intrinsic properties. In the context of wind power forecasting, unsupervised models are valuable for clustering meteorological data, detecting anomalies, and identifying underlying regimes that influence wind behaviour, which can later enhance the performance of supervised predictors [52].

- K-Means clustering

K-Means clustering represents one of the most extensively utilized techniques in unsupervised machine learning, primarily designed to partition a dataset into K distinct clusters by minimising the intra-cluster variance. In the context of wind power forecasting, it is particularly useful for tasks such as classifying wind patterns, grouping similar meteorological conditions, or clustering different geographical locations based on wind speed and wind power characteristics [32]. This allows the identification of dominant wind regimes and provides a structured understanding of heterogeneous wind behaviours. The optimisation problem of K-Means is expressed through its objective function in Equation II.17.

$$J=\sum_{i=1}^{k}\sum_{x_j\in C_i}\left\|x_j-\mu_i\right\|^2 \tag{II.17}$$

Where J is the total within-cluster variance (the objective to be minimized), K is the number of clusters predefined by the user, $x_j \in R^d$ represents the $j^{th}$ data point in a d-dimensional feature space, $C_i$ denotes the set of data points assigned to cluster i, $\mu_i \in R^d$ is the centroid of cluster i,

defined as the mean of all data points in $C_i$ and $\|x_j - \mu_i\|^2$ is the squared Euclidean distance between data point $x_j$ and its corresponding cluster centroid [32].

The algorithm proceeds iteratively in two main steps.

1. Assignment step: each data point is assigned to the cluster whose centroid is nearest in terms of Euclidean distance.
2. Update step: Cluster centroids are updated by computing the mean of all data points allocated to each cluster.

This iterative process continues until convergence is achieved, typically indicated by stability in the cluster assignments or centroids fall below a predefined threshold. In wind power forecasting applications, K-Means clustering enhances prediction by enabling data-driven pre-processing [32]. for instance, wind speed time series can be clustered into different regimes (e.g., low, medium, and high wind speed patterns). Forecasting models trained separately on each regime often outperform a single global model, as they capture the heterogeneity of wind behaviours.

In addition to clustering techniques such as K-means, dimensionality reduction methods like Principal Component Analysis (PCA) are also categorised under unsupervised learning [25]. PCA, already detailed in I.10.4 Feature Extraction section, is a statistical approach that identifies the dominant directions of variability in the data without the need for labelled outputs. Its ability to reduce high-dimensional meteorological inputs into a smaller set of informative components.

### II.3 Stand-Alone Deep Learning Models for wind power forecasting

Stand-alone deep learning models represent the foundational category of neural architectures applied directly to wind power forecasting, without combining them with other models or pre-processing frameworks. These models are capable of automatically extracting complex patterns from raw meteorological data, making them highly effective for capturing the nonlinear and dynamic characteristics of wind speed and power generation [64]. Among them, Artificial Neural Networks (ANNs) are used as a basic structure for mapping input-output relationships, while Recurrent Neural Networks (RNNs) and their advanced variants, such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs), are well-suited for handling temporal dependencies in sequential wind data [19]. Convolutional Neural Networks (CNNs), on the other hand, are particularly effective in identifying local correlations and multi-dimensional features. Despite their architectural differences, all these models share the

common goal of reducing forecasting errors through the optimisation of network parameters. Figure II.2 provides a conceptual view of stand-alone models, while Table II.1 shows a comparative representation of the advantages and disadvantages of these models.

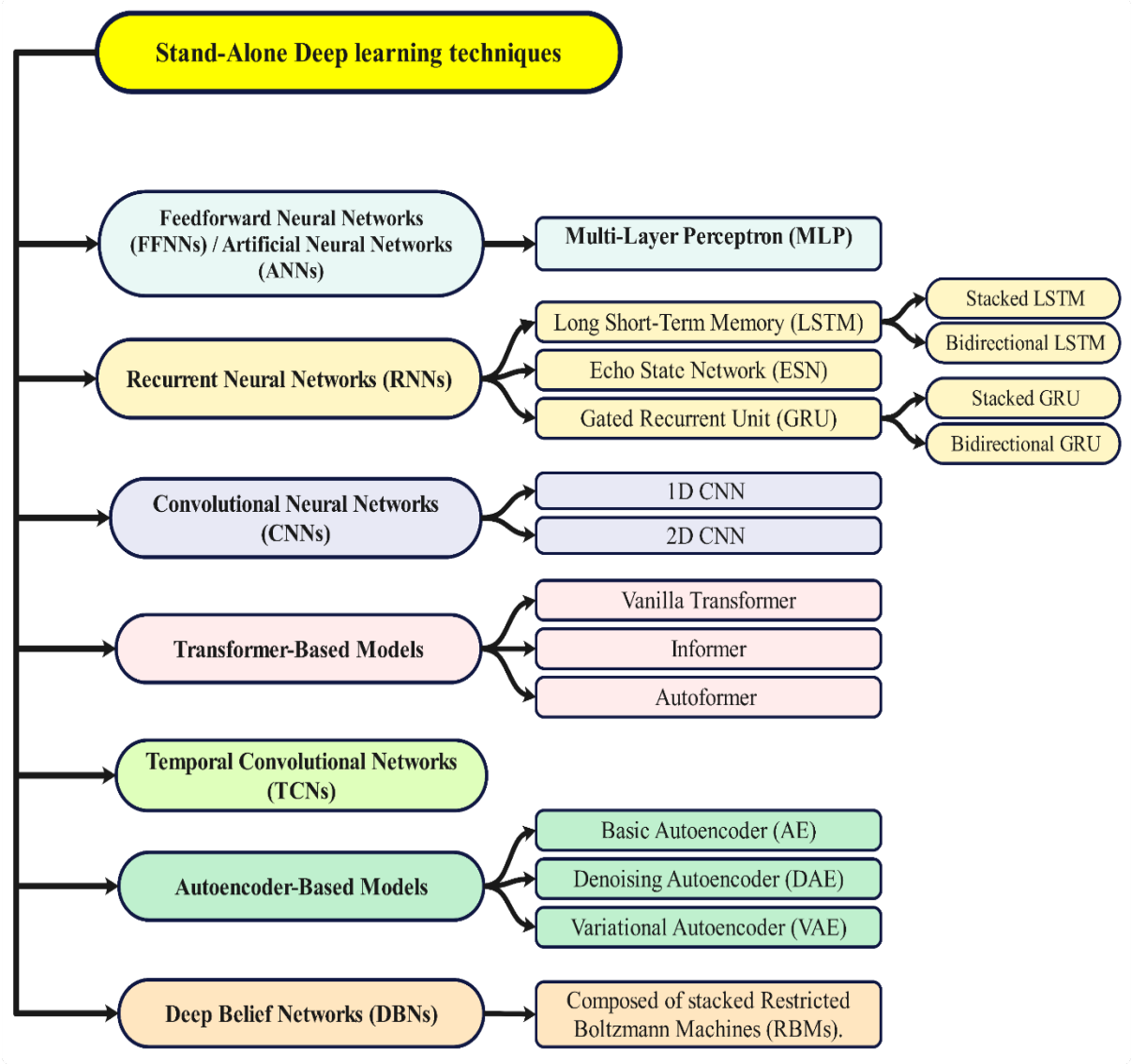



**Figure II.2**: Stand-Alone Deep Learning Models-based forecasting techniques

### ➢ Artificial Neural Networks (ANN)

Artificial Neural Networks (ANNs) represent one of the most widely applied deep learning paradigms for wind power forecasting. Inspired by the biological neural system, ANNs are capable of approximating complex, nonlinear relationships between input meteorological features and the target output, such as wind speed or generated power. Their architecture is composed of interconnected layers of artificial neurons, typically including an input layer, one or more hidden layers, and an output layer [65]. Each neuron computes a weighted sum of its

inputs, incorporates a bias, applies a nonlinear activation function, and transmits the resulting signal to subsequent layers as presented in Figure II.3. Through this layered structure, ANNs can capture intricate dependencies that are often difficult to represent using statistical or physical models. Equation II.18 provides a mathematical expression for the output of a single artificial neuron.

$$\hat{y}(x)=f\left(b+\sum_{i=1}^{n} w_i x_i\right) \tag{II.18}$$

Where $\hat{y}(x)$ is the predicted output of the neuron given the input vector $x=[x_1, x_2,\ldots, x_n]$, while $w_i$ represents the synaptic weight associated with the input $x_i$, b is the bias term, which provides flexibility by shifting the activation threshold, $f(\cdot)$ is the activation function that introduces nonlinearity into the model [65].

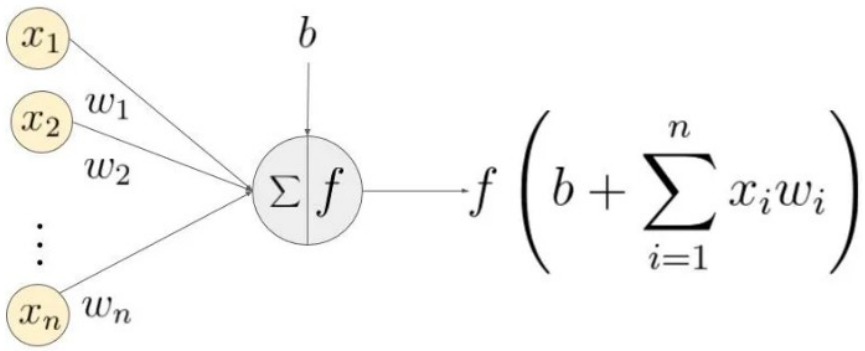



**Figure II.3**: Artificial Neural Networks mathematical

The summation term $\sum_{i=1}^{n} w_i x_i$ computes a weighted linear combination of the input features. The bias b ensures that the activation function is not restricted to passing through the origin. The activation function $f(\cdot)$ can take various forms depending on the task [49].

In practice, a complete ANN consists of multiple interconnected neurons arranged in hidden layers, where the output of one layer serves as the input to the next. This hierarchical processing enables the network to progressively capture increasingly abstract features from raw input data. During training, the weights $w_i$ and biases b through optimisation algorithms such as stochastic gradient descent (SGD) or Adam, with backpropagation applied using the backpropagation algorithm with the objective of minimizing a loss function that evaluates the difference between model predictions and actual outcomes [43].

➢ **Recurrent Neural Networks (RNN)**

RNNs constitute a neural network architecture designed to handle sequential data, rendering them well-suited for time series tasks. By incorporating recurrent links, they maintain information across previous time steps, in contrast to feedforward networks that assume input independence [16], as appears in Figure II.4. The general formulation of an RNN in Equation II.19.

$$h_t = f\left(w_{xh} x_t + w_{hh} h_{t-1} + b_h\right) \quad \text{(II.19)}$$

$$\hat{y}_t = g\left(w_{hy} h_t + b_y\right) \quad \text{(II.20)}$$

where $h_t$ denotes the hidden state at time step t, which encapsulates both the current input $x_t$ and the memory of the past state $h_{t-1}$. The weight matrices $w_{xh}$, $w_{hh}$, and $w_{hy}$Why respectively represent the connections from input to hidden state, hidden to hidden state, and hidden to output [16]. The terms $b_h$ and $b_y$ are bias vectors, while $f$ (·) and $g$ (·) are nonlinear activation functions, commonly chosen as the hyperbolic tangent (tanh) or sigmoid functions for $f$, and linear or softmax functions for $g$, depending on the prediction task. Finally, $\hat{y}_t$ represents the predicted output at time t, reflecting the network's ability to integrate sequential information for forecasting purposes [65].

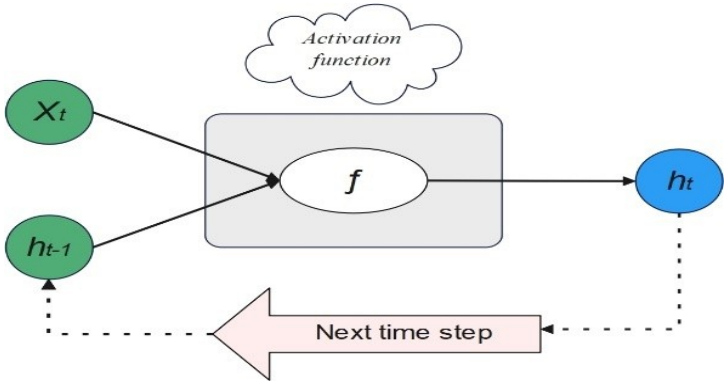



**Figure II.4**: Recurrent Neural Networks Architecture

In Equation II.20, the hidden state $h_t$ acts as a dynamic memory that evolves over time, preserving information from previous observations and updating itself with each new input. This recursive mechanism allows RNNs to model complex temporal patterns, making them especially effective in capturing wind speed fluctuations and power output variability. However, traditional RNNs often suffer from issues such as vanishing or exploding gradients during training, which hinder their ability to capture long-term dependencies [16].

### ➢ Long Short-Term Memory (LSTM)

LSTMs represent an advanced class of RNN developed to overcome the vanishing and exploding gradient problems that hinder the learning of long-term dependencies. Unlike conventional RNNs, which struggle with retaining information across extended sequences, LSTMs introduce a memory cell and a gating mechanism that regulate the flow of information as illustrated in Figure II.5 [66]. This design enables LSTMs to selectively retain, update, or discard information over time, making them particularly effective for sequential forecasting tasks such as wind power forecasting, where long-term temporal dependencies and dynamic patterns play a critical role [67].

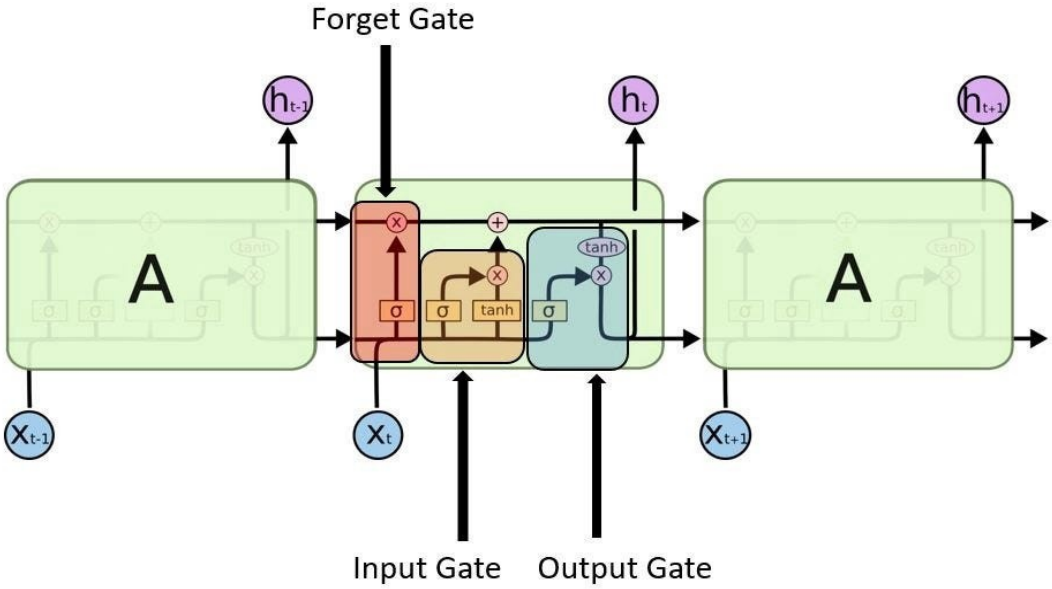



**Figure II.5:** LSTM architecture

This gating architecture enables LSTMs to effectively retain relevant information across long sequences while discarding irrelevant ones, thereby capturing both short and long-term dependencies in sequential data such as wind power time series.

### ➢ Gated Recurrent Unit (GRU)

GRUs is a simplified form of the LSTM designed to decrease model complexity while retaining the capacity to learn long-term dependencies in sequential data. In contrast to LSTMs, which maintain separate hidden and cell states, the GRU employs a single hidden state vector, making its architecture computationally more efficient while still alleviating the vanishing gradient problem. The GRU introduces two primary gating mechanisms: the reset gate and the update gate Figure II.6, which together control the flow of information through the network [68].

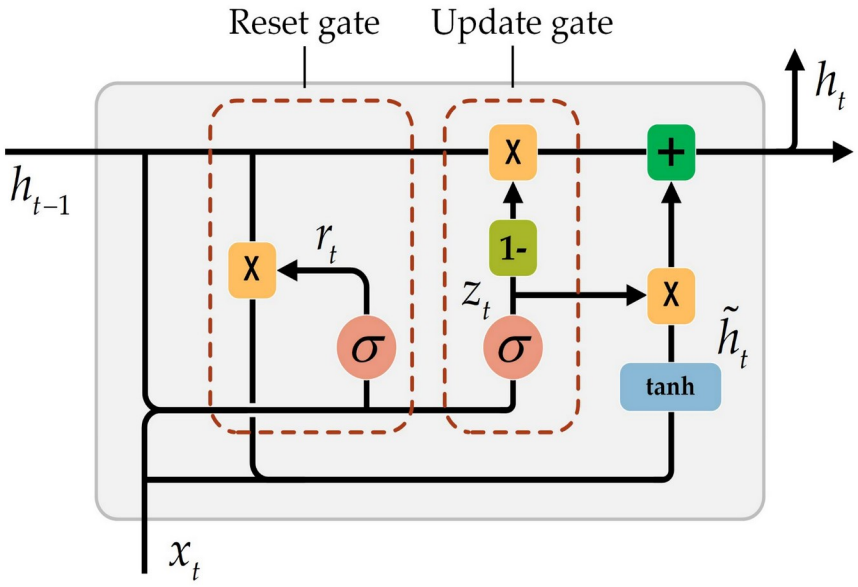



**Figure II.6:** GRU structure

In Figure II.6, sigmoid activation function is σ, tanh is the hyperbolic tangent activation, $x_t$ is the input vector at time t, and $h_{t-1}$ is the previous hidden state.

In the Gated Recurrent Unit (GRU), the reset gate $r_t$ determines how much of the previous hidden state should be ignored, allowing the model to selectively forget past information that may no longer be relevant. The update gate $z_t$ controls the balance between retaining information from the past and incorporating new input, effectively acting as a memory filter. Next, the candidate hidden state $\tilde{h}_t$ arises from applying the tanh activation to a linear transformation of the current input and the reset-filtered previous hidden state, which enables the network to capture new features while considering selected past dependencies. Finally, as a convex combination of the previous hidden state and the candidate hidden state the actual hidden state $h_t$ in is updated, weighted by the update gate, for a smooth integration of previous and new information to maintain long-term dependencies in sequential data [68].

- **Bidirectional Mechanism**

In bidirectional recurrent architectures such as Bi-RNN, Bi-LSTM, and Bi-GRU, the model handles the input sequence in two directions forward and backwards, enabling the network to capture dependencies from past as well as future contexts. In the forward pass, the hidden state is obtained through a nonlinear transformation that integrates the current input and the prior hidden state within the chosen recurrent unit, as expressed. Simultaneously, the backwards pass computes the hidden state by processing the sequence in reverse, taking into account the current input and the next hidden state [69]. Finally, At time step t, the hidden state is fully represented

by merging the forward and backward hidden states, capturing contextual information from both past and future time steps [70].

➢ **Stacked Mechanism**

Stacked recurrent models extend the capability of single-layer recurrent networks by stacking several LSTM or GRU layers, enabling the network to capture hierarchical temporal features. In this structure, the hidden states produced by one recurrent layer serve as the input to the next, facilitating the capture of immediate temporal relationships in lower layers and higher-level, long-range patterns in deeper layers. This deep stacking improves the expressiveness of the model, making it suitable for handling the complex, multi-scale temporal dynamics often present in wind power data [55]. Compared to single-layer configurations, stacked recurrent architectures generally achieve better performance in forecasting tasks.

➢ **Convolutional Neural Networks (CNN)**

CNNs have been widely applied in time-series forecasting due to their capacity to autonomously identify local temporal patterns. Unlike recurrent networks, CNNs process data in parallel, making them computationally efficient. The fundamental building blocks of CNNs include the convolution operation, activation function, and pooling layer, which together reduce dimensionality while preserving essential features [71].

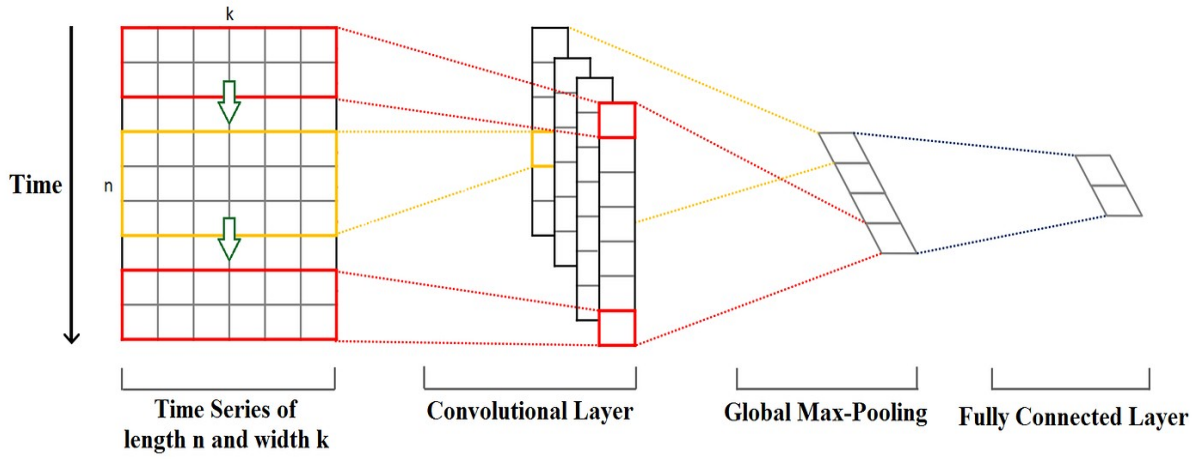



**Figure II.7:** CNN key components

The convolution operation applies a kernel (or filter) W to the input sequence x, followed by a bias term b generating a feature map.

$$y_t = (w * x)_t + b \tag{II.21}$$

where * denotes the convolution operator. The extracted features are then transformed via a nonlinear activation function typically ReLU.

$$h_t = \sigma(y_t) = max\,(0, y_t) \tag{II.22}$$

To reduce dimensionality while retaining dominant information, a pooling layer is applied.

$$p_t = pool(h_t) \tag{II.23}$$

The pooled features are then flattened and passed to fully connected layers, which combine the learned representations and generate the final prediction.

$$\hat{y}_t = f(p_t) \tag{II.24}$$

In a CNN, the input sequence x is first processed through the convolution operation in Equation II.21, where the kernel w slides over the input to extract local features and generate feature maps $y_t$. These maps are then passed through the nonlinear activation function in Equation II.22, where σ (·), often ReLU, introduces nonlinearity and ensures sparse representations [72]. To reduce dimensionality and enhance robustness against fluctuations, the pooling operation in Equation II.23 is applied, producing the pooled features $p_t$. Finally, these pooled features are flattened and passed to fully connected layers Equation II.24, which integrate the extracted hierarchical patterns to generate the final prediction $\hat{y}_t$. In summary, CNNs capture local spatial dependencies through convolution and pooling operations, transform them into compact feature representations, and subsequently map these features to predictions via fully connected layers [73].

CNN can be applied in both one-dimensional (1D) and two-dimensional (2D) forms, depending on the structure of the input data. CNN-1D is typically employed for sequential data, where the convolutional filters slide along a single temporal dimension to capture local temporal dependencies [15]. In contrast, CNN-2D is widely used for spatially structured data, such as images or two-dimensional meteorological maps, where filters move across both horizontal and vertical dimensions to detect spatial correlations. While CNN-1D is computationally more efficient and suitable for forecasting tasks involving univariate or multivariate time series [73], CNN-2D provides richer feature extraction when spatial or spatio-temporal patterns are critical [74]. The choice between CNN-1D and CNN-2D thus relay on the input nature and the forecasting objective [75].

### ➢ Autoencoder-Based Models

Autoencoders constitute a class of unsupervised neural network architectures aimed at learning efficient, low-dimensional representations of input data. They consist of two main components, an encoder, which maps the input data x into a compressed latent feature representation z, and a decoder, which reconstructs the input from this latent representation, as shown in Figure III.8. By minimising the difference between the original input and its reconstruction, autoencoders can extract meaningful patterns and non-linear correlations [76].

#### ❖ Basic Autoencoder

Encoder: The encoder transforms $x$ input data into a latent feature representation z using Equation II.25.

$$z = f_e\left(w_{\varnothing} x + b_{\varnothing}\right) \tag{II.25}$$

where $w_{\varnothing}$ and $b_{\varnothing}$ are the encoder's weights and biases, and $f_e(\cdot)$ is the encoder activation function [14].

Decoder: By applying Equation II.26, this component generates a reconstruction of the input data based on the latent vector z.

$$\hat{x} = f_d\left(w_\theta z + b_\theta\right) \tag{II.26}$$

where $w_\theta$ and $b_\theta$ are the decoder's weights and biases, and $f_d(\cdot)$ is the decoder activation function.

Reconstruction Error: The quality of reconstruction is measured by the error between $x$ and $\hat{x}$. Using Euclidean distance.

$$\Delta\left(x, \hat{x}\right) = \left\| x - \hat{x} \right\|^2 \tag{II.27}$$

The objective is to minimize Δ, thereby ensuring the latent space captures the most relevant features [14].

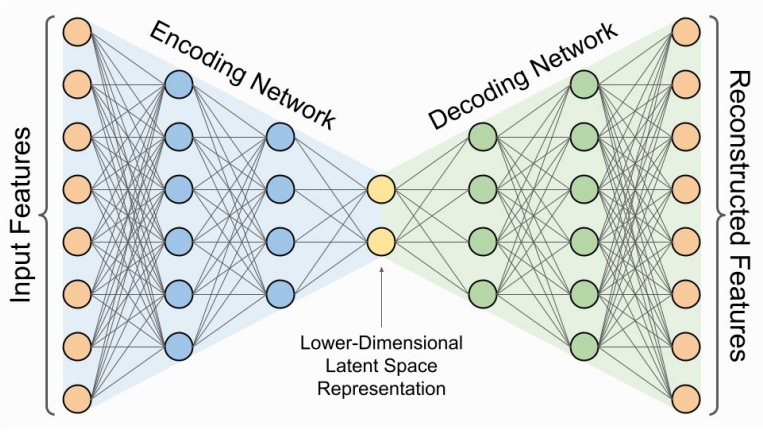



**Figure II.8:** Autoencoder structure

### ❖ Denoising Autoencoder (DAE)

DAE enhances robustness by deliberately corrupting the input data and training the network to reconstruct the original clean signal. Specifically, the clean input $x$ is transformed into a corrupted version $\tilde{x}$ through a stochastic corruption process, applied randomly to each training sample or mini-batch. Among the common approaches, additive isotropic Gaussian noise drawn from the distribution $N(0,\sigma^2 I)$ is frequently employed. The encoder then learns to map $\tilde{x}$ into a latent representation that enables the decoder to recover the original clean input $x$, thereby forcing the model to capture meaningful and noise-invariant features [77].

$$\tilde{x}=x+\epsilon, where \epsilon \quad N(0,\sigma^2 I) \tag{II.28}$$

Encoder:

$$z=f_e(w_\varnothing \tilde{x}+b_\varnothing) \tag{II.29}$$

Decoder:

$$\hat{x}=f_d(w_\theta z+b_\theta) \tag{II.30}$$

Reconstruction Error:

$$\Delta(x,\hat{x})=\|x-\hat{x}\|^2 \tag{II.31}$$

By forcing the model to recover original inputs from noisy versions, DAEs capture more robust and generalizable feature representations, useful in wind power forecasting where data often contains sensor noise or missing values [77].

### ❖ Variational Autoencoder (VAE)

VAE extends the standard autoencoder framework by incorporating concepts from probabilistic graphical models. Unlike deterministic encoders that map inputs directly to fixed latent vectors, The VAE models the latent representation as a probability distribution, commonly assumed to be a multivariate Gaussian. This enables the model to capture uncertainty in the data and generate realistic synthetic samples, which can be particularly useful in wind power forecasting under uncertain or incomplete conditions [19].

Formally, the encoder maps the input $x$ to the parameters of a probability distribution, namely the mean vector $\mu(x)$ and the variance (or standard deviation) vector $\sigma(x)$. The latent variable z is then sampled from this learned distribution using the reparameterization trick, which ensures differentiability during training.

$$z=\mu(x)+\sigma(x)\Theta\epsilon, where \epsilon \quad N(0,I) \tag{II.32}$$

In Equation II.32, $x$ represents the input data, $\mu(x)$ is the encoder's predicted mean vector for the latent distribution, $\sigma(x)$ is the encoder's predicted standard deviation vector, $\epsilon$ is random noise sampled from a standard normal distribution $N(0,I)$ and $\odot$ denotes element-wise multiplication, which injects randomness into z which is the resulting latent representation sampled from the learned distribution.

The decoder then reconstructs the input from the latent variable z using Equation II.33.

$$x^{'}=f_d(w_\theta z+b_\theta) \tag{II.33}$$

$x^{'}$ is the reconstructed input, $f_d(\cdot)$ is the decoder mapping, $w_\theta$ and $b_\theta$ are the decoder's trainable weight matrix and bias vector.

Training a VAE requires optimizing a special loss function known as the Evidence Lower Bound (ELBO), which balances reconstruction accuracy and distribution regularization:

$$L_{VAE}=E_{q_\varnothing(z|x)}\left[\log p_\theta(x|z)\right]-D_{KL}(q_\varnothing(x|z)\|p(z)) \tag{II.34}$$

In Equation.II.34, $q_\varnothing(z|x)$ is the encoder's learned distribution of latent variables given the input. $p_\theta(x|z)$ is the likelihood of reconstructing $x$ from z, $E_{q_\varnothing(z|x)}\left[\log p_\theta(x|z)\right]$ represents the expected reconstruction log-likelihood, encouraging accurate reconstructions, $D_{KL}(q_\varnothing(x|z)\|p(z))$ is the Kullback-Leibler (KL) divergence, which regularizes the learned latent distribution so that it remains close to the prior $p(z)$, usually chosen as a standard Gaussian $N(0,I)$. Figure II.9 shows the structure of the VAE [19].

In summary, the VAE enhances robustness and generalization by combining reconstruction learning with probabilistic regularization [19].

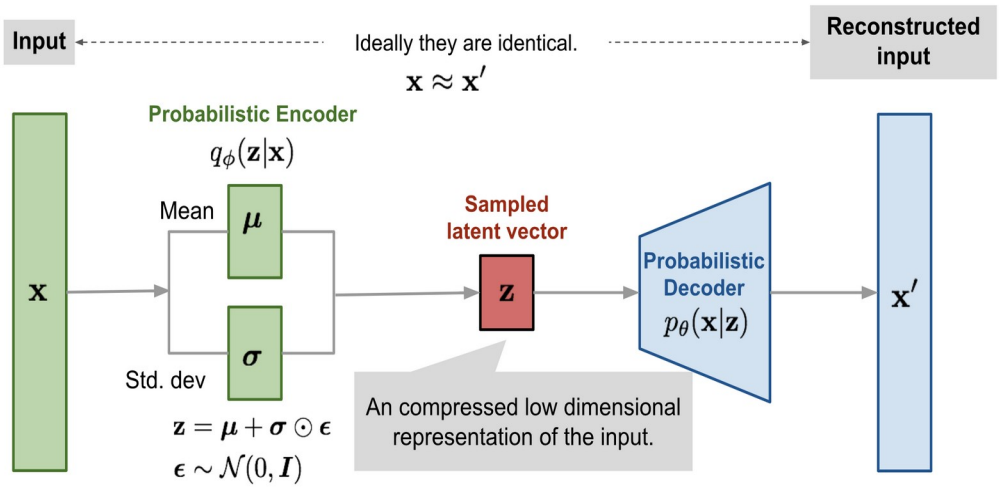



**Figure II.9:** Variational Autoencoder structure

**Other Common Deep Learning Models**

In addition to recurrent, convolutional, and autoencoder-based approaches, other sophisticated deep learning models have also attracted interest in wind energy forecasting. These models offer alternative architectures and mechanisms for handling temporal dependencies and uncertainty in complex datasets.

➢ Transformer-Based Models

Transformer architectures, first developed for natural language processing, have been effectively applied to time series forecasting because they can model long-range dependencies without using recurrent structures.

- *Vanilla Transformer*: It employs self-attention mechanisms to assign varying importance to different time steps within the input sequence [78].
- *Informer*: A streamlined version specifically developed for forecasting extended sequences. It reduces computational complexity through a ProbSparse self-attention mechanism and introduces a generative-style decoder for faster predictions.
- *Autoformer*: Improves interpretability and long-horizon forecasting through the decomposition of time series into trend and seasonal components within the self-attention mechanism [78].

➢ Temporal Convolutional Networks (TCNs)

TCNs leverage causal and dilated convolutional layers to effectively extract temporal patterns from sequential data across multiple scales. Unlike RNNs, they process sequences in parallel, allowing for faster training and stable gradient propagation. Their architecture ensures that predictions at any time step depend only on past information [79].

➢ Deep Belief Networks (DBNs)

DBNs are composed of stacked RBM layers that function as generative probabilistic models. Hierarchical features are first learned through unsupervised layer-wise training, and the network is subsequently refined using supervised learning. In wind energy forecasting, DBNs can capture nonlinear and latent structures in the data [80].

**Table II.1**: Comparative Representation of Advantages and Disadvantages of Stand-Alone Deep Learning Models

| Model | Advantages | Disadvantages |
|---|---|---|
| **RNN** | Captures sequential dependencies, suitable for simple time-series forecasting. | Suffers from vanishing gradients, weak at modelling long-term dependencies. |
| **LSTM** | Handles long-term dependencies effectively, widely applied in wind forecasting. | Computationally expensive, requires large datasets. |
| **BiLSTM** | Processes information in both directions, improves temporal feature extraction. | High computational cost, risk of overfitting with small data. |
| **Stacked LSTM** | Learns hierarchical temporal features, enhances prediction accuracy. | Increased complexity, careful tuning needed to avoid overfitting. |
| **GRU** | More efficient than LSTM, effective for medium-term dependencies. | Less expressive than LSTM for very long sequences, lower flexibility. |
| **CNN** | Strong at extracting local and spatial features, efficient parallel processing. | Less suited for sequential-only data, needs pre-processing. |
| **TCN** | Captures long sequences via dilated convolutions, stable training. | Hyperparameter-sensitive; struggles with extremely long dependencies. |
| **Transformer** | Long-range dependencies with self-attention; scalable to large data. | High computational demand; data-hungry. |
| **Informer** | Efficient for long sequences; optimized Transformer for forecasting. | Complex architecture; resource-intensive tuning. |
| **Autoformer** | Decomposition-based, handles seasonality well; effective for wind data. | Limited to seasonal data; requires large datasets. |
| **DBN** | Strong feature learning via RBMs; useful in early wind forecasting. | Difficult to train; prone to overfitting. |
| **AE** | Learns compact representations; aids anomaly detection and dimensionality reduction. | Sensitive to hyperparameters; weak standalone predictor. |
| **DAE** | Robust to noisy data; improves data quality. | Risk of overfitting; limited direct forecasting ability. |
| **VAE** | Captures uncertainty in forecasts; generates | Computationally intensive; may miss |

| Model | Advantages | Disadvantages |
|---|---|---|
| | probabilistic outputs. | nonlinear dynamics. |
| **MLP** | Simple, fast baseline model; easy to implement. | Poor at modelling temporal dependencies; low accuracy in complex data. |
| **ESN** | Fast training; efficient for quick predictions. | Needs large data; weak at capturing long-term dependencies. |

## II.4 Hybrid Models for wind power forecasting

Hybrid models in wind energy research integrate two or more methodological paradigms to address the inherent complexity of atmospheric dynamics and power generation processes. Wind speed and wind power time series are characterized by nonlinearity, non-stationarity, intermittency, seasonality, and stochastic variability. Since no single modelling approach can fully represent these features, hybridization constitutes a methodologically justified strategy for improving predictive performance. The principal configurations of such hybrid approaches are summarized in Table II.2.

Hybridization may occur between statistical, machine learning, deep learning, pre-processing, and optimization techniques as shown in Figure III.10. The interaction among these categories is not fixed but depends on the structure of the problem, the forecasting horizon, and the nature of the available data. In many configurations, statistical models are employed to represent linear trends, periodicity, and autocorrelation structures, while machine learning algorithms are used to approximate nonlinear relationships between meteorological variables and wind power output. This separation of linear and nonlinear components often improves predictive stability and interpretability.

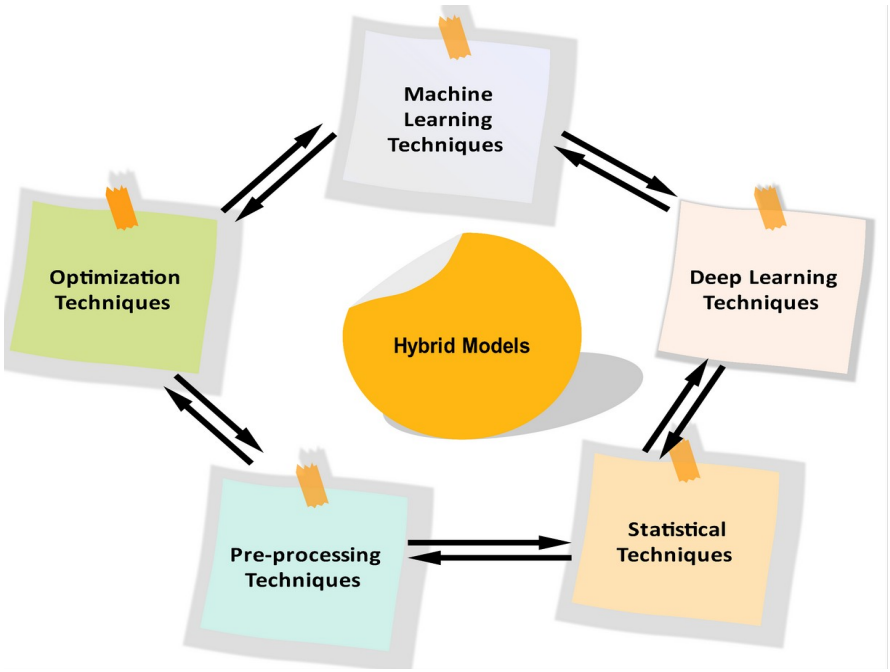



**Figure II.10:** The principal configurations of hybrid models

Deep learning techniques extend this capability by modelling complex temporal dependencies and hierarchical feature representations. When combined with statistical approaches, deep networks can focus on nonlinear residual structures after deterministic components have been extracted [66]. Alternatively, deep learning models may operate alongside conventional machine learning algorithms, either through feature extraction followed by regression or through ensemble strategies that aggregate multiple predictors. Such combinations can reduce variance, mitigate overfitting, and improve generalization performance across varying wind regimes [81] [82].

Pre-processing techniques frequently constitute an integral element of hybrid frameworks. Wind data are often noisy and highly volatile, therefore, decomposition methods, filtering procedures, normalization, and feature selection mechanisms are applied to transform raw signals into more structured subcomponents [39]. By reducing noise and isolating intrinsic modes of variability, pre-processing simplifies the learning task and enhances model convergence. In decomposition-based hybrids, each sub-series may be modelled separately and later recombined [27].

Optimization techniques further reinforce hybrid systems by refining parameter selection and structural configuration [83]. Evolutionary or swarm-based algorithms are commonly

employed to tune hyperparameters, determine optimal lag structures, select relevant input variables, or compute combination weights in ensemble models. In this sense, optimization does not serve as a predictive mechanism on its own but operates as an adaptive layer that enhances performance and robustness [84].

Thus, a hybrid model in wind energy forecasting can be understood as an integrated framework in which at least two of these methodological categories interact to exploit complementary strengths. Statistical approaches contribute structure and interpretability [85], machine learning provides nonlinear mapping capability [17], deep learning captures complex temporal and hierarchical patterns [86], pre-processing improves data quality and signal separability; and optimization ensures efficient parameterization. The resulting architecture is not defined by complexity alone, but by the coherent integration of methods selected to address the multifaceted nature of wind energy systems.

**Table II.2**: Hybridisation Strategies for Wind Power Forecasting

| Hybrid Configuration | Conceptual Interaction Mechanism | Primary Contribution |
|---|---|---|
| Statistical-Machine Learning | Decomposition of linear structure with nonlinear mapping of residual dynamics | Captures complementary linear and nonlinear behaviours |
| Statistical-Deep Learning | Deterministic trend extraction combined with hierarchical temporal feature learning | Enhances stability and long-term dependency modelling |
| Machine Learning-Deep Learning | Hierarchical feature representation with adaptive predictive integration | Improves generalization and model robustness |
| Pre-processing-Predictive Integration | Signal transformation and complexity reduction prior to model inference | Mitigates noise and non-stationarity effects |
| Optimization-Embedded Hybrid | Algorithm-driven parameter and structural refinement | Enhances convergence efficiency and predictive accuracy |

## II.5 Hyperparameter Tuning

Hyperparameter tuning denotes the procedure of identifying the most suitable configuration values that control how a model learns from data. Unlike trainable parameters, which are

optimised during training (e.g., weights and biases), the key hyperparameters in Table II.3 are predefined settings that govern the learning process. Their selection plays a decisive role in balancing underfitting and overfitting, shaping model complexity [29].

Selecting appropriate values for these hyperparameters is non-trivial, as their effects are often interdependent and problem-specific. For example, a small learning rate generally improves training stability by ensuring smoother convergence, while the batch size determines the trade-off between training speed and the precision of gradient estimation [34]. Similarly, the number of hidden units and layers should be carefully calibrated to capture nonlinear temporal dependencies without introducing unnecessary computational complexity or risk of overfitting. Selecting an appropriate hyperparameter tuning approach depends on the dataset's complexity and the available computational resources.

Given the high-dimensional nature of hyperparameter search spaces, systematic tuning strategies are employed to identify optimal configurations including:

- **Grid Search**: An exhaustive approach that evaluates model performance across all possible combinations within predefined ranges. While computationally expensive, it guarantees coverage of the search space [29].
- **Random Search**: Randomly samples hyperparameter combinations, often achieving competitive results at lower computational cost compared to grid search.
- **Bayesian Optimization**: Employs probabilistic models to iteratively refine the search toward promising regions of the hyperparameter space, balancing exploration and exploitation [34].
- **Adaptive/Sequential Methods**: Advanced strategies such as Hyperband or Population-Based Training dynamically allocate resources to more promising configurations, reducing time and computational overhead [34].

**Table II.3**: Key Hyperparameters in Deep Learning Models for Forecasting Applications

| Hyperparameter | Description | Impact on Model |
|---|---|---|
| **Learning Rate** | Step size used by the optimizer to update weights. | Controls convergence speed and stability; too high may cause divergence, too low may slow learning. |
| **Batch Size** | Number of training samples processed before updating weights. | Influences memory usage, training stability, and ability to generalize. |
| **Epochs** | Full passes through the training dataset. | Affects convergence and risk of overfitting; too many may cause overfitting, too few underfitting. |
| **Dropout Rate** | Fraction of neurons randomly deactivated during training. | Prevents overfitting by reducing co-adaptation of neurons. |
| **Number of Hidden Units** | Neurons number assigned to each hidden layer | Determines model capacity; too few limits expressiveness, too many increased risks of overfitting. |
| **Number of Hidden Layers** | Depth of the network. | Enhances ability to capture complex representations but increases training complexity. |
| **Activation Function** | Non-linear transformation applied to neurons. | Shapes feature representations and affects training dynamics. |
| **Optimizer** | Algorithm used to minimize the loss function. | Determines efficiency and stability of training. |

## II.6 Model Evaluation and Validation

Model evaluation and validation are fundamental processes in artificial intelligence-based time series forecasting, as they determine not only how well a model predicts but also how reliably it can generalise to unseen conditions. Where data are nonlinear, nonstationary, and highly influenced by features variability, a rigorous evaluation framework is essential to ensure both accuracy and operational trustworthiness [19]. To achieve this, the evaluation process in this work is structured into three key stages: (i) the use of performance metrics to quantitatively assess predictive accuracy, (ii) benchmarking against baseline models to provide meaningful

comparative insights, and (iii) interpretation and visualisation to translate results into practical understanding for real-world deployment.

### II.6.1 Performance Metrics

The evaluation of forecasting models relies heavily on quantitative performance metrics that measure the accuracy, consistency, and reliability of predictions. In wind speed and wind power forecasting, where fluctuations are nonlinear and strongly influenced by environmental conditions, selecting appropriate metrics is critical. No single metric is sufficient to capture all aspects of predictive performance; hence, a combination of complementary metrics is often employed.

| | | |
|---|---|---|
| Mean Squared Error [87] | $MSE = \frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2$ | (II.35) |
| Root Mean Square Error [87] | $RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2}$ | (II.36) |
| Mean Absolute Error [87] | $MAE = \frac{1}{N}\sum_{t=1}^{N}\lvert y_t - \hat{y}_t \rvert$ | (II.37) |
| Mean Absolute Percentage Error [34] | $MAPE = \frac{100}{N}\sum_{t=1}^{N}\left\lvert \frac{y_t - \hat{y}_t}{y_t} \right\rvert$ | (II.38) |
| Coefficient of Determination [88] | $R^2 = 1 - \frac{\sum_{t=1}^{N}(y_t - \hat{y}_t)^2}{\sum_{t=1}^{N}(y_t - \overline{y_t})^2}$ | (II.39) |
| Normalized RMSE [87] | $nRMSE = \frac{RMSE}{\frac{1}{N}\sum_{t=1}^{N} y_t} \times 100$ | (II.40) |
| Normalized MAE [89] | $nMAE = \frac{MAE}{\frac{1}{N}\sum_{t=1}^{N} y_t} \times 100$ | (II.41) |
| Skill Score (SS) [90] | $SS = 1 - \frac{MAE_{model}}{MAE_{baseline}}$ | (II.42) |

Where, $y_t$ and $\hat{y}_t$ correspond to the actual and forecasted values, respectively, with the total number of observations.

**Table II.4**: Comparative Assessment of Forecasting Evaluation Metrics: Strengths and Weaknesses in Wind Forecasting Tasks

| Metric | Strengths | Weaknesses |
|---|---|---|
| **MSE** | Strongly penalizes large errors, making it useful for highlighting severe forecasting deviations. | Sensitive to outliers, which may distort overall performance assessment. |
| **RMSE** | The metric reports errors in the units of the forecasted variable, making interpretation more intuitive. | Overemphasizes large errors and less representative of average performance. |
| **MAE** | Compared to squared-error metrics, it offers greater robustness to outliers while directly representing the mean magnitude of the errors. | Does not discriminate between small and large errors, limiting sensitivity to extreme events. |
| **MAPE** | Expresses error in percentage, enabling scale-independent comparison across datasets. | Undefined when actual values approach zero; disproportionately penalizes small denominators. |
| **$R^2$** | Captures proportion of variance explained, providing a global measure of model fit. | May be misleading for non-linear or non-stationary data; does not reflect error magnitude. |
| **nRMSE** | Enables fair comparison across different sites or datasets by normalizing RMSE. | Normalization choice (mean, range, etc.) influences interpretation, reducing consistency. |
| **nMAE** | Scale-free measure that improves comparability between datasets with different magnitudes. | Same limitations as MAE, with additional sensitivity to normalization method. |
| **SS** | Evaluates relative improvement over a reference model, highlighting practical forecasting gains. | Dependent on baseline choice; poor baseline selection can inflate perceived performance. |

**Statistical Testing for Reliability:**

➢ Confidence Intervals (CI)

Confidence intervals quantify the uncertainty associated with performance metrics. For MAE and RMSE, 95% intervals are determined by computing the standard error associated with the mean, as expressed in Equation II.43. They define the bounds within which the actual performance of the model is expected to occur, highlighting whether the results reflect consistent predictive behaviour rather than random fluctuations. For $R^2$ and MAPE, only point estimates are presented, as these metrics are typically assessed using their absolute values rather than their variation [91].

$$CI = \bar{x} \pm t_{\alpha/2,N-1} \cdot \left( \frac{S}{\sqrt{N}} \right) \quad \text{(II.43)}$$

➢ p-values and Hypothesis Testing

The evaluation of performance differences relies on paired t-tests, as expressed in Equations II.44 and II.45. This test is well-suited to the forecasting since the errors for the different models are calculated using the same dataset, this introduces a dependency that needs to be considered [77]. Considering this dependency ensures that the comparison is fair and reliable, allowing improvements to be interpreted as statistically meaningful rather than arising from random variation [77].

$$t = \frac{\bar{d}}{\frac{S_d}{\sqrt{N}}} \quad \text{(II.44)}$$

$$p = 2.\left(1 - T_{cdf}\left(|t|, N-1\right)\right) \quad \text{(II.45)}$$

Where $\bar{d}$ is the mean of the differences between the paired errors of the two models and $S_d$ is the standard deviation of the differences, while $\bar{x}$ is the sample mean of the metric $t_{\alpha/2,N-1}$ is the critical value from the t-distribution for a 95% confidence level, $N-1$ is the degrees of freedom and $S$ is the sample standard deviation of the metric and $T_{cdf}$ is the cumulative distribution function of the t-distribution [12] [27].

A small p-value, typically less than 0.05, indicates that the performance differences observed are unlikely to have arisen by chance and can be considered statistically significant.

### II.6.2 Benchmarking Against Baseline Models

A critical element of the evaluation process is the systematic comparison of the proposed forecasting framework with a set of baseline models. Benchmarking serves as a reference point, allowing the predictive capability of the developed approach to be positioned within the

broader landscape of established methods. By situating results in relation to existing techniques, the analysis highlights not only the level of improvement achieved but also the extent to which the framework addresses limitations of traditional approaches.

Beyond providing context, benchmarking ensures that reported gains are both meaningful and quantifiable. Improvements are not evaluated in isolation but are measured against consistent standards, which enhances the credibility of the findings. This approach makes it possible to distinguish genuine advancements from outcomes that may simply reflect dataset characteristics or random variation [52].

### II.6.3 Interpretation and Visualization

While performance metrics provide quantitative evidence of accuracy, they do not always reveal underlying trends, systematic biases, or contextual relevance [27]. Visualisation tools, such as error distribution plots, scatter diagrams, and time-series overlays, complement statistical measures by offering intuitive perspectives on how forecasts align with actual observations. These graphical representations make it possible to detect patterns of over- or under-estimation, identify temporal variations in model reliability, and highlight conditions under which the forecasting framework performs the most [27].

Beyond descriptive purposes, visualisation also supports deeper analytical interpretation. Evaluating the proposed methodology in comparison with established benchmark models allows visual examination to reinforce or challenge the results derived from statistical testing. This combined approach, incorporating both quantitative metrics and visual evidence, ensures a comprehensive understanding of the model's performance characteristics. Ultimately, the integration of interpretation and visualisation strengthens the validity of the assessment [52].

## II.7 Limitations and Research Gaps

- **Limitations of Existing Methods**

Baseline forecasting methods, while widely used, frequently face difficulties in modelling the intricate and nonlinear behaviour of wind energy. Many of these models rely on simplified assumptions, which can lead to reduced adaptability across diverse operating conditions. Their predictive accuracy is frequently limited, especially when dealing with highly variable meteorological inputs or multi-horizon forecasting tasks.

❖ **Identified Research Gaps**

A key gap in the literature lies in the limited ability of existing models to provide consistent and reliable forecasts across different datasets and forecasting horizons. In many cases, improvements are demonstrated only within narrow experimental contexts, making it difficult to generalise findings. Additionally, comparative studies often fail to establish a clear framework for benchmarking performance in a way that highlights the true advancements of newer approaches.

❖ **How This Thesis Addresses the Gaps**

This thesis overcomes these limitations by conducting a systematic benchmarking of the proposed forecasting framework against a set of well-established baseline models. The comparison is designed to evaluate performance under identical experimental conditions, ensuring that improvements are clearly attributable to the strengths of the proposed method. By providing a structured and transparent assessment, the thesis contextualises the predictive capacity of the developed approach and demonstrates its capacity to surpass conventional baseline models regarding accuracy and robustness.

## II.8 Conclusion

This chapter provided a comprehensive review of artificial intelligence techniques applied to wind power forecasting, covering both conventional machine learning techniques and modern deep learning architectures. Machine learning methods, including regression-based, tree-based, Bayesian, and ensemble models, were highlighted for their ability to capture nonlinear dependencies and improve forecasting reliability. Deep learning models, with their capacity to model temporal dependencies and complex nonlinearities, were presented as a natural progression toward more robust forecasting frameworks. Furthermore, hybrid models were emphasised for their ability to integrate the strengths of individual methods and mitigate their limitations, thereby enhancing predictive accuracy.

The chapter also addressed essential aspects of hyperparameter tuning and model evaluation, including performance metrics, benchmarking, and interpretability. By systematically reviewing these elements, the discussion identified the key limitations of existing methods and the gaps that remain unaddressed. Finally, the chapter outlined how this thesis builds upon these foundations by proposing advanced hybrid frameworks that integrate deep learning with complementary techniques to deliver improved accuracy.

# Chapter III: The Proposed Forecasting Frameworks

# The Proposed Forecasting Frameworks

## III.1 Introduction

The present chapter outlines the methodological framework developed to implement and evaluate advanced approaches for wind power and wind speed forecasting. Building on the theoretical background and model descriptions presented in earlier chapters, this section shifts the focus toward the practical realisation of forecasting frameworks, highlighting both standalone deep learning architectures and hybrid models enhanced by signal decomposition techniques. The proposed approach relies on meteorological datasets gathered from purposefully selected locations.

To ensure methodological rigour, the chapter is structured around the two forecasting tasks. For wind power forecasting, the experimental design explores the application of LSTM networks, ConvLSTM architectures, and a hybrid strategy that integrates EEMD decomposition with deep learning models. In contrast, the wind speed forecasting task employs a more sophisticated framework, bringing together CEEMDAN for signal decomposition, a Bi-GRU encoder-decoder architecture, and a transfer learning scheme to improve the model's adaptability and robustness across varying forecasting conditions.

Both forecasting frameworks are presented as end-to-end pipelines, encompassing data pre-processing, model training and forecasting stages. The pre-processing stage is highlighted for its role in extracting intricate temporal and non-stationary structures from wind data, whereas the deep learning models are employed to capture nonlinear behaviours and long-term temporal dependencies.

In summary, this chapter establishes the methodological groundwork that supports the experimental investigations presented later, offering a clear and systematic presentation of the models, procedures, and frameworks employed. This methodological design enables an in-depth and comparative evaluation of forecasting performance, ensuring both accuracy and robustness under varied temporal horizons and operational contexts.

### III.2 Datasets and Study Areas

To ensure the reliability of the forecasting framework, this study employs carefully selected datasets from different geographical locations. The datasets provide comprehensive measurements, and the following subsections describe in detail the wind power datasets used in this work, along with the characteristics of the study areas from which they were collected.

#### III.2.1 Wind Power Dataset

For this study, data were sourced from a Siemens SWT-3.6-120 turbine installed at the Amrumbank West offshore wind farm in the German sector of the North Sea, positioned approximately 35 km northwest of Heligoland Island and 18 km southwest of the Amrumbank sandbank as a region characterized by strong and relatively stable wind resources. The precise geographical location of the turbine is illustrated in Figure III.1.

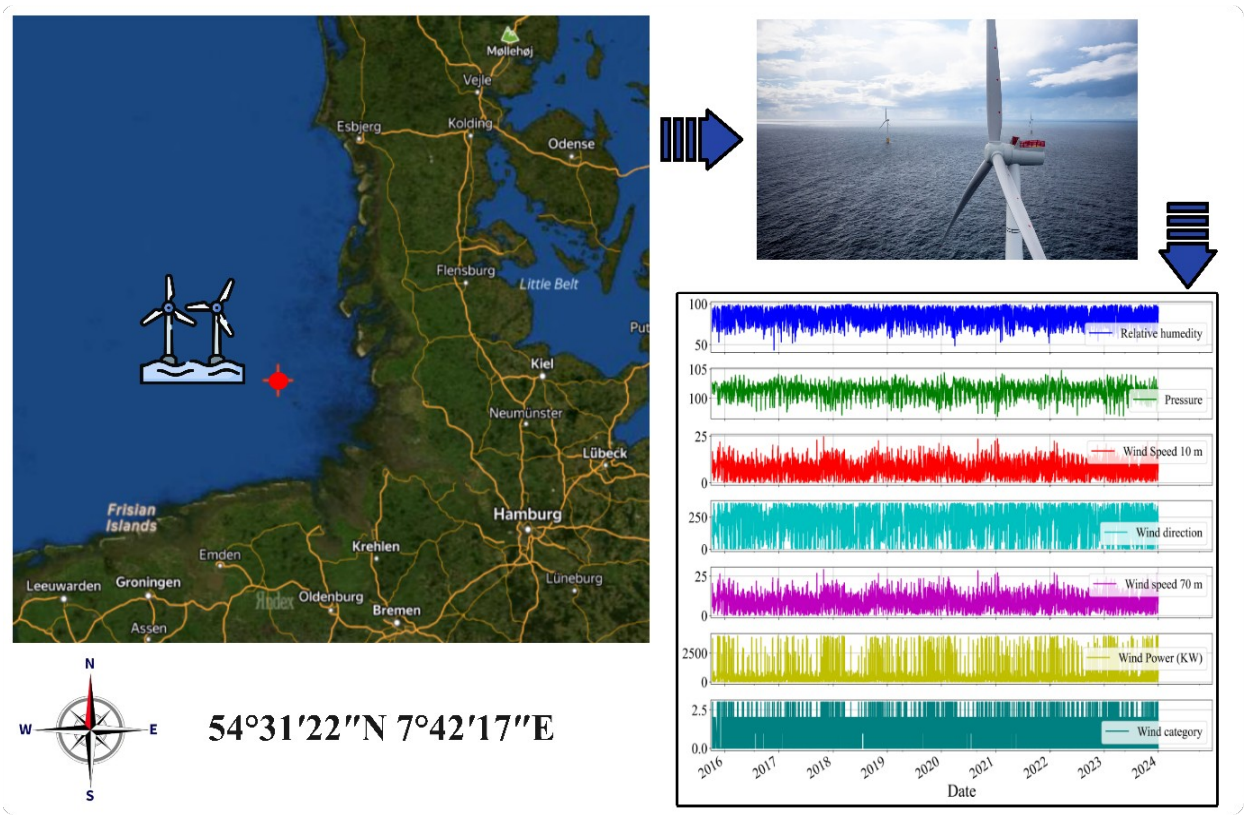



**Figure III.1**: Wind power dataset description

Focusing on data from a single turbine unit offers distinct methodological advantages. Unlike aggregated datasets, which may obscure fine-grained operational patterns due to averaging effects, single-turbine datasets preserve the site-specific environmental influences and machine-related operational characteristics. These include localized variations in wind speed, turbulence intensity, and wake effects, as well as turbine-specific responses to environmental

dynamics. Such granularity enhances the ability of forecasting models to learn detailed temporal dependencies and nonlinear interactions, ultimately improving their predictive performance. Moreover, the offshore setting of the Amrumbank West wind farm provides an ideal testbed for wind forecasting research, as offshore wind conditions are typically less influenced by terrain effects yet remain highly variable due to meteorological dynamics designed to represent both immediate variations and longer-term temporal patterns. This makes the dataset particularly suitable for evaluating forecasting approaches.

The dataset comprises hourly measurements over a nine-year period from January 2015 to December 2023. It includes a range of meteorological variables and wind power measurements relevant to offshore forecasting.

- Air pressure (Pa): affects air density and, consequently, the efficiency of energy capture.
- Wind speed (m/s) at 70 meters: Recorded at the turbine's hub height, it serves as the main factor influencing wind energy generation due to its direct influence on the available kinetic energy.
- Relative humidity (%): modifies atmospheric characteristics and indirectly influences wind dynamics.
- Wind direction (°): determines the yaw alignment of the turbine and its ability to optimally harness wind energy.
- Wind speed (m/s) at 10 meters: provides insight into near-surface wind patterns and vertical wind shear.
- Wind category (low, medium, high): a categorical feature classifying operational states and typical wind regimes.

The selection of the six input features was guided by the corelation to the target label. While including a larger number of variables could potentially improve the ability to capture nonlinear interactions, this may also elevate the risk of overfitting while increasing the required computational resources. Conversely, using fewer features may reduce complexity but would likely compromise prediction quality.

The statistical properties of the selected variables are summarized in Table III.1, which presents their mean values, standard deviations, and ranges. Of particular importance, A Spearman's ρ of 0.97 was observed between wind speed and wind power, reflecting a highly consistent monotonic relationship. This finding reinforces wind speed as the most influential predictor of power generation. The relationship is rooted in the fundamental physics of wind energy, which can be expressed in Equation III.1 [23].

$$P_m = \frac{1}{2}\rho v_w^3 \pi r^2 C_p \quad \text{(III.1)}$$

The formula defines $P_m$ as the power obtained from wind, with ρ indicating air density (kg/m³), r the rotor radius (m), $v_w$ the wind speed (m/s), and $C_p$ representing the turbine's effectiveness in converting wind energy into mechanical power [6].

**Table III.1:** Wind Power Dataset statistical details

| features | Observations | Average | Std Dev | Minimum | Maximum |
|---|---|---|---|---|---|
| Relative humidity | 72334 | 84.54 | 8.08 | 43.5 | 100 |
| Air pressure | 72334 | 101.38 | 1.09 | 96.9 | 104.83 |
| Wind direction 70m | // | 208.11 | 92.69 | 0 | 359.94 |
| Wind speed 10m | // | 7.67 | 3.54 | 0.3 | 24.8 |
| Wind category | // | nan | nan | 0 | 3 |
| Wind speed 70 m | // | 8.98 | 4.15 | 1.3 | 30.38 |
| Wind power (kW) | // | 532.64 | 807.7 | 0 | 3993.6 |

### III.2.2 Wind Speed Dataset

Two meteorological datasets collected from distinct geographic regions were employed to forecast wind speed. The first dataset, hereafter referred to as Dataset 1, was obtained from Adrar, Sebaa (Algeria) and spans the period 2015–2023. The second dataset, Dataset 2, originates from Chennai in Tataouine (Tunisia) and covers the years 2016–2021. Both datasets consist of hourly measurements of the following variables.

- Wind speed at 50 m and 10 m height (m/s).
- Atmospheric pressure (Pa).
- Ambient temperature (°C).

The first dataset includes 70,128 samples and the second 61,343, providing valuable insights for forecasting by reflecting distinct wind patterns arising from different climatic and topographical conditions. The Algerian site, located in a semi-arid desert environment characterized by high wind potential and strong seasonal variability, whereas the Tunisian site reflects a Mediterranean climate with more moderate but fluctuating wind patterns. Figure III.2

presents the geographical locations of the sites along with the correlation matrices of the dataset variables.

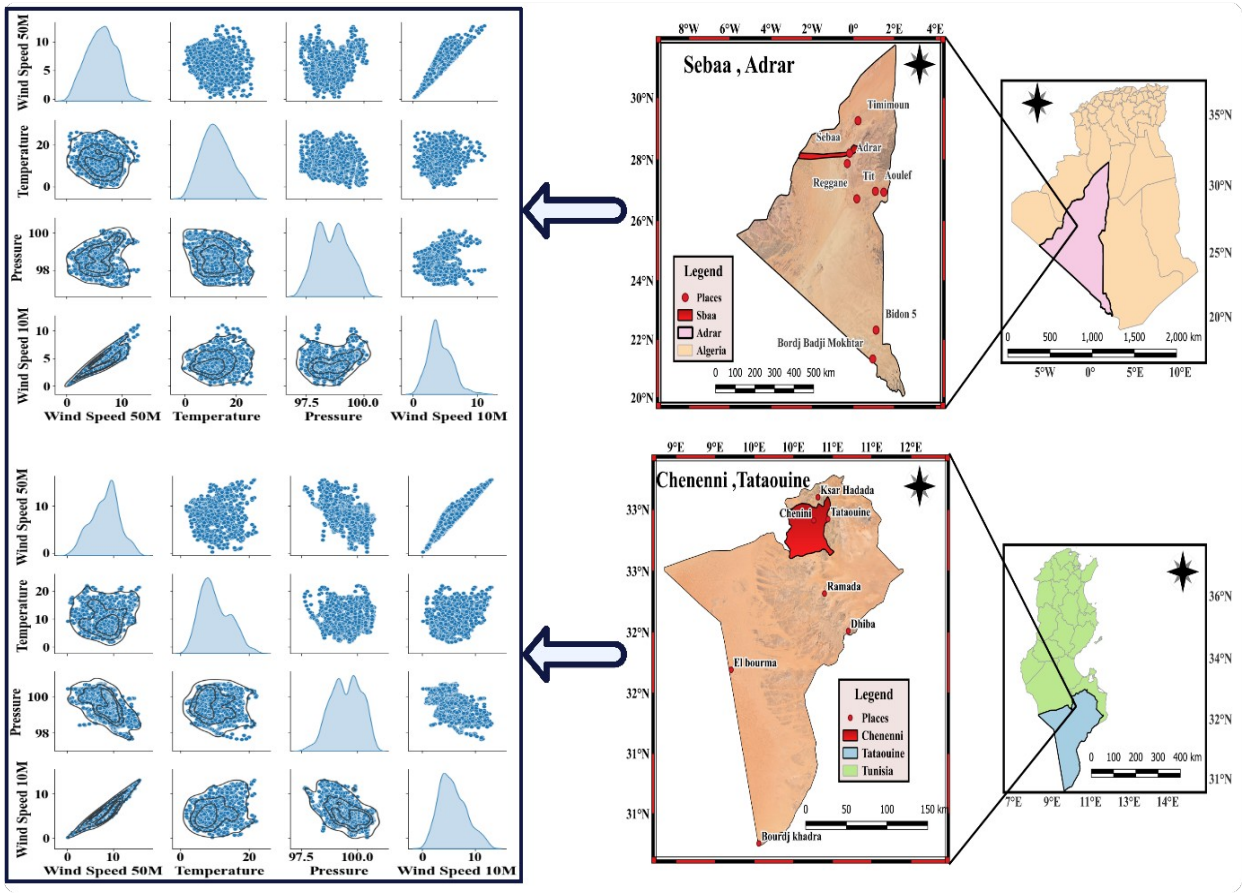



**Figure III.2**: Wind Speed Datasets description

### III.3 Wind power forecasting methodology

The methodology adopted for wind power forecasting in this study is designed to combine data-driven deep learning approaches with signal decomposition techniques to streamline the input data for prediction models. Building upon the theoretical models discussed earlier, the framework integrates deep models such as Long Short-Term Memory (LSTM) and Convolutional LSTM (ConvLSTM), enhanced through Ensemble Empirical Mode Decomposition (EEMD) for noise reduction and feature extraction. The key aim is to model short-term temporal patterns alongside the complex nonlinear dynamics present in wind energy output. This section outlines the step-by-step methodology, including data pre-processing, model design, training strategies, and evaluation metrics, forming the foundation of the forecasting experiments.

### III.3.1 Long Short-Term Memory (LSTM)

The LSTM specifically designed to overcome the limitations of RNNs, particularly the issue of vanishing or exploding gradients when dealing with long sequences. Its core component is the memory cell, which functions as an information reservoir and is regulated by three gates. The input gate determines which new information is added to the memory, the forget gate decides which past information should be removed, and the output gate controls what part of the stored information is transmitted to the next layer. Through this structure, LSTMs can balance the retention of long-term dependencies with the flexibility to discard irrelevant data, making them well suited for time series forecasting tasks such as wind power prediction. The internal structure of the LSTM is schematically illustrated in Figure III.3 [19].

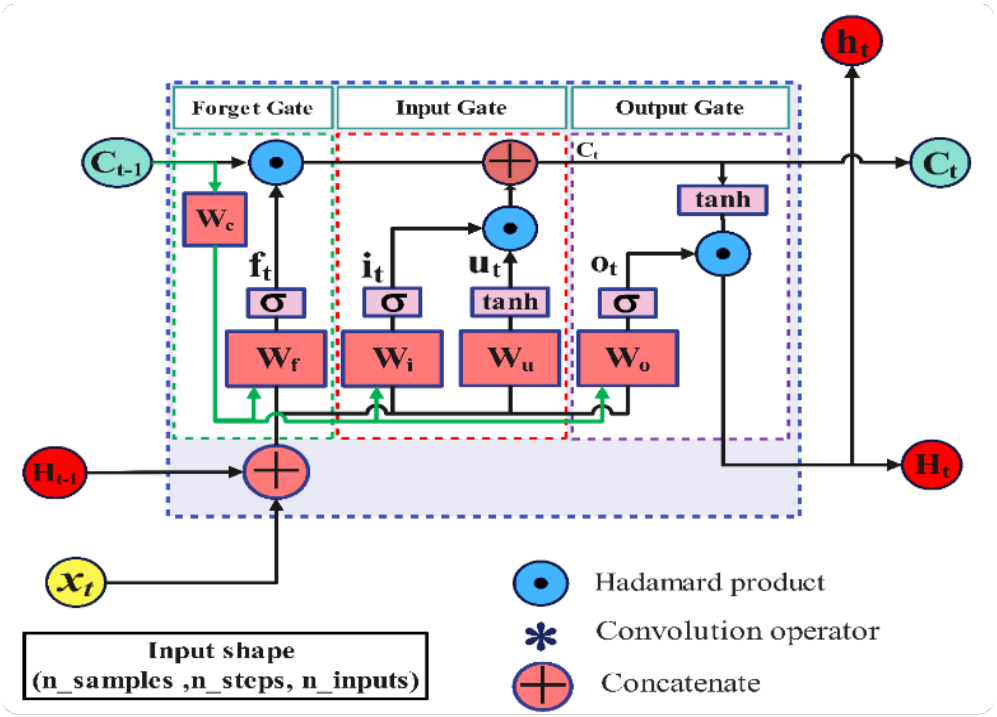



**Figure III.3**: LSTM structure

The input gate $i_t$ in Equation III.2 regulates how much of the new information should be stored in the memory cell at time step t. It considers the current input vector $x_t$ and the hidden state from the previous step $h_{t-1}$. An activation function sigmoid presented in Equation III.3 is applied to create a value between 0 and 1, which acts as a weight indicating the importance of the incoming information. Mathematically, it is expressed as [39].

$$i_t = \sigma\left(w_{xi} x_t + w_{hi} h_{t-1} + w_{ci} \odot c_{t-1} + b_i\right) \tag{III.2}$$

$$\sigma(X)=\frac{1}{1-e^{-x}} \tag{III.3}$$

Here, $w_{xi}$ and $w_{hi}$ denote the input and hidden weights, respectively, while $b_i$ is the associated bias.

As described in Equation III.4, the gate output is multiplied by the candidate vector $u_t$ which encodes potential new information.

$$u_t=\tanh\left(w_{xu}x_t+w_{hu}h_{t-1}+b_u\right) \tag{III.4}$$

with the hyperbolic tangent function defined in Equation III.5.

$$\tanh(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{III.5}$$

The $f_t$ controls the proportion of previous memory that is preserved as a forget gate using Equation III.6 in order of eliminating irrelevant information and avoiding memory saturation [39].

$$f_t=\sigma\left(w_{xf}x_t+w_{hf}h_{t-1}+w_{cf}\Theta c_{t-1}+b_f\right) \tag{III.6}$$

The cell state $c_t$ is updated by combining the retained past memory $c_{t-1}$ and the new information $u_t$, modulated by the respective gates [39].

$$c_t=f_t\Theta c_{t-1}+i_t\Theta u_t \tag{III.7}$$

This step is central to the LSTM's for preserving long-term dependencies across time steps. The output gate $o_t$ determines how much of the updated cell state is transferred to the final output of the unit. It is given by [39].

$$o_t=\sigma\left(w_{xo}x_t+w_{ho}h_{t-1}+w_{co}\Theta c_{t-1}+b_0\right) \tag{III.8}$$

The final hidden state $h_t$ is computed by filtering the updated cell state through a tanh activation.

$$h_t=o_t\Theta\tanh(c_t) \tag{III.9}$$

### III.3.2 Ensemble Empirical Mode Decomposition (EEMD)

EEMD is an enhanced processing-signal technique developed to overcome the limitations of the traditional Empirical Mode Decomposition (EMD). The original EMD often suffers from mode mixing, a phenomenon where oscillatory components with different time scales are

incorrectly merged into a single Intrinsic Mode Function (IMF), or conversely, components with similar time scales are distributed across multiple IMFs [10]. This drawback reduces the interpretability and reliability of the decomposition, particularly in complex, non-stationary signals such as wind-related time series [66]. EEMD addresses this issue by introducing controlled white noise into the original signal and performing multiple decompositions. Through ensemble averaging of the resulting IMFs, the added noise cancels out, leaving a more robust and physically meaningful representation of the underlying signal. Unlike fixed-basis methods such as Fourier or wavelet transforms, EEMD is fully data-driven and adaptively decomposes the signal into intrinsic oscillatory elements known as IMFs, known as IMFs, that capture local variations in both frequency and amplitude. This adaptability makes EEMD particularly well-suited for wind forecasting applications, where signals often exhibit strong nonlinearity and variability [92].

The essential steps of the EEMD algorithm, are summarized in Figure III.4.

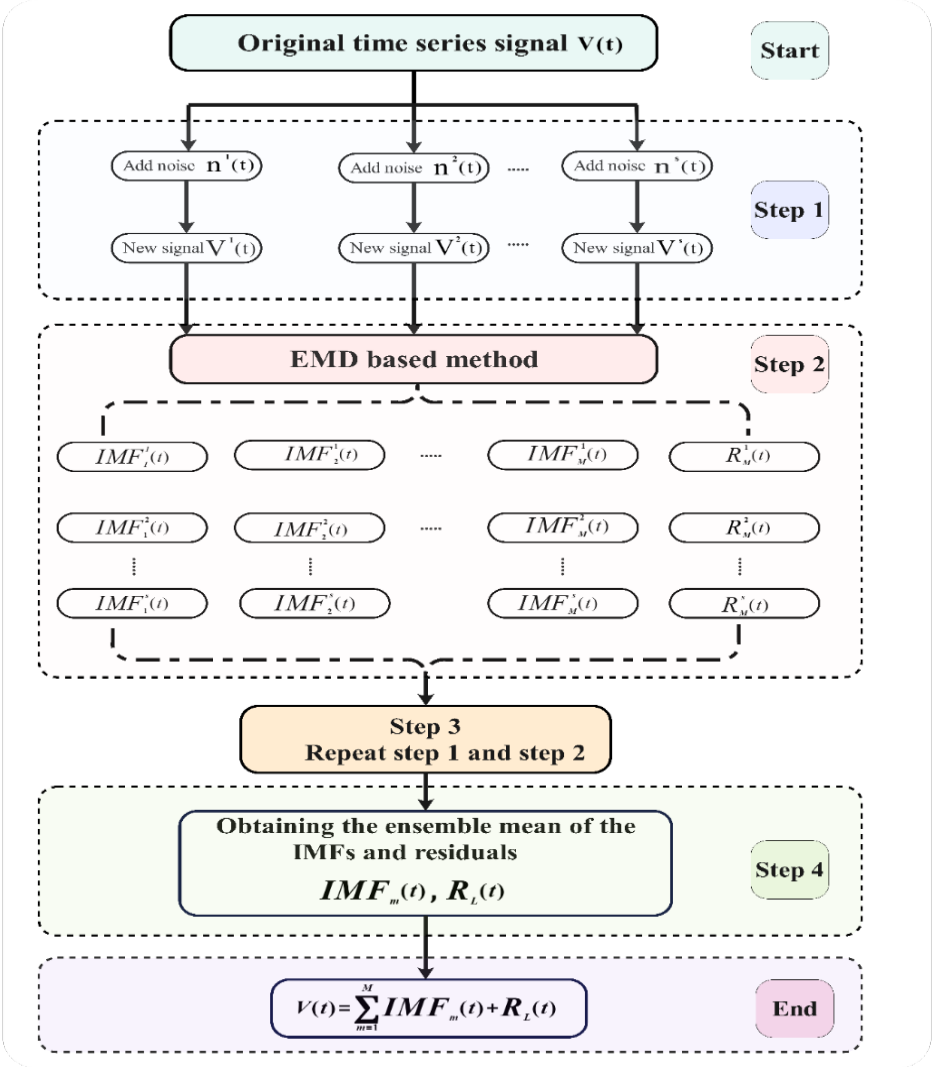



**Figure III.4**: EEMD Decomposition Steps

As the first step, following Equation III.10, independent Gaussian white noise sequences, denoted as $n^s(t)$ (s=1,2,…,s+1) are added to the wind power forecasting error V(t), generating a perturbed signal $V^s(t)$[92].

$$V^s(t)=V(t)+n^s(t) \tag{III.10}$$

In the second step, EMD is applied to the noisy signal to extract multiple IMFs and a residual component. The formulation of $V^s(t)$ is subsequently provided in Equation III.11 [92].

$$V^s(t)=\sum_{m=1}^{M} IMF_m^{EMD,s}(t)+R_M^{EMD,s}(t) \qquad m=1,2,3.....M \tag{III.11}$$

$IMF_m^{EMD,s}(t)$ represents the IMF components, while $R_M^{EMD,s}(t)$ corresponds to the residual sequence, both derived after applying EMD to the noisy signal.

In the third step, Steps 1 and 2 are performed repeatedly S times, incorporating a distinct white noise sequence in every repetition [93].

$$\left(IMF_1^{EMD,s}(t), IMF_2^{EMD,s}(t), IMF_3^{EMD,s}(t).....IMF_M^{EMD,s}(t), R_M^{EMD,s}(t)\right) \quad s=1,2,3.....N$$

The fourth step involves deriving the ensemble mean of the IMFs and residuals, which cancels out the added noise, as described in Equations III.12, III.13, and III.14 [92].

$$IMF_m(t)=\frac{1}{N}\sum_{s=1}^{N} IMF_m^{EMD,S}(t) \quad m=1,2,3.....M \tag{III.12}$$

$$R_L(t)=\frac{1}{N}\sum_{s=1}^{N} R_M^{EMD,s} \tag{III.13}$$

$$V(t)=\sum_{m=1}^{M} IMF_m(t)+R_L(t) \tag{III.14}$$

### III.3.3 Convolutional Long Short-Term Memory (ConvLSTM)

The ConvLSTM is an advanced extension of conventional LSTM, specifically designed to handle spatiotemporal sequence modelling. While standard LSTMs are effective in capturing temporal dependencies, they rely on fully connected operations that disregard the spatial correlations present in structured data such as images, videos, or meteorological fields. ConvLSTM addresses this limitation by embedding convolutional operations within both the input-to-state and state-to-state transitions of the LSTM architecture [74].

In this formulation, the inputs, hidden states, and cell states are represented as three dimensional tensors. The use of convolutions instead of matrix multiplications ensures that local spatial structures are preserved and effectively propagated across time steps [10].

Figure III.5 depicts the overall architecture of the ConvLSTM model while mathematically described in Equations III.15-III.20, which detail the gating mechanisms and convolutional operations governing information flow [74].

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \odot C_{t-1} + b_i) \quad \text{(III.15)}$$

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \odot C_{t-1} + b_f) \quad \text{(III.16)}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot u_t \quad \text{(III.17)}$$

$$u_t = \tanh(W_{xu} * X_t + W_{hu} * H_{t-1} + b_u) \quad \text{(III.18)}$$

$$O_t = \sigma(w_{xo} ¿ X_t + W_{ho} * H_{t-1} + W_{co} \odot C_{t-1} + b_0) \quad \text{(III.19)}$$

$$h_t = O_t \odot \tanh(C_t) \quad \text{(III.20)}$$

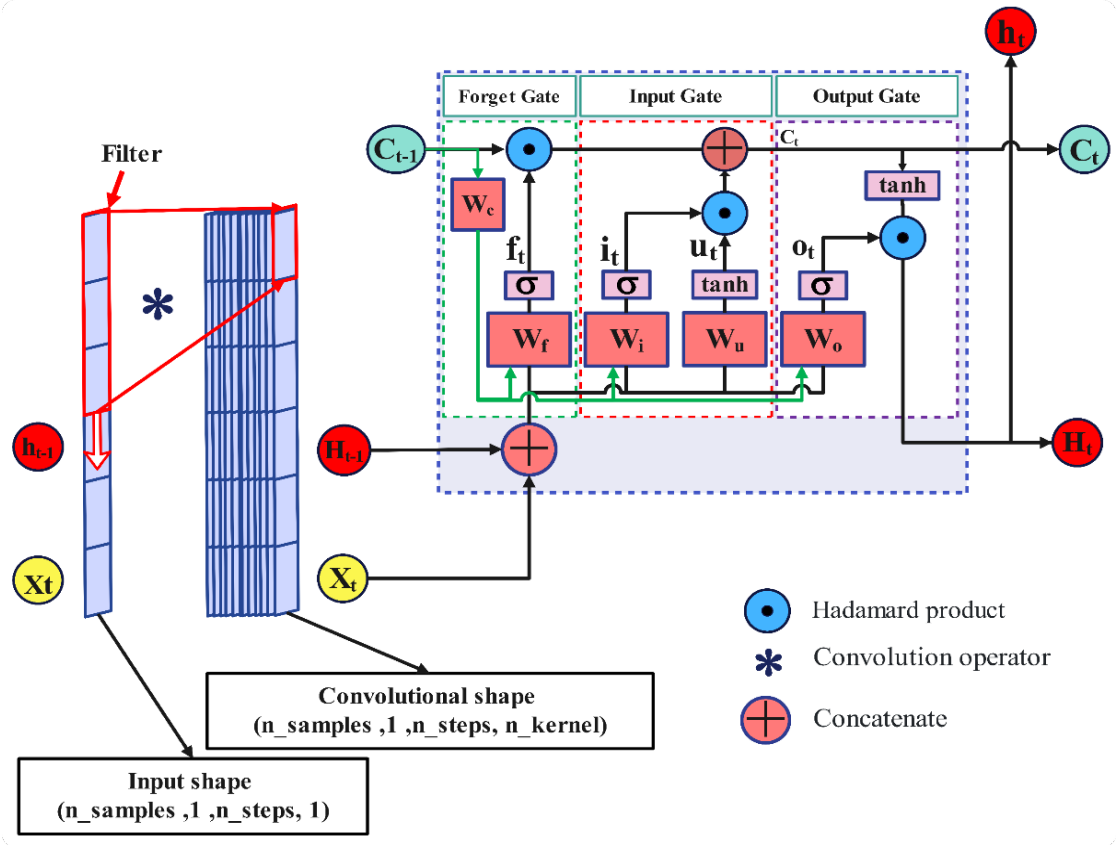



**Figure III.5**: ConvLSTM structure

### III.4 Wind Power Forecasting Framework

The proposed forecasting framework, referred to as LSTM-EEMD-ConvLSTM, which combines deep learning with error decomposition. As depicted in Figure III.6, the methodology is carried out through the following primary steps:

1. Initial Forecasting with LSTM

The LSTM model receives the selected meteorological features and observed wind power as inputs and outputs the initial forecasted wind power, represented as $\hat{P}_n$.

2. Error Extraction

The forecasting error series $Er_n$It is calculated as the difference between the observed wind power values $\hat{P}_n$ and the forecasted values $P_n$.

3. EEMD Decomposition

The error sequence $Er_n$ is then decomposed using into a finite set of IMFs $(I\,MF_1, \ldots\ldots, IMF_{15})$and a residual component *Res* to isolate error fluctuations across multiple frequency bands, facilitating more precise modelling.

4. Component Forecasting with ConvLSTM

The ConvLSTM network takes each IMF and the residual as inputs, producing predictions for each component $(\widehat{Imf}_1, \widehat{Imf}_2 \ldots\ldots \widehat{Imf}_{15} \wedge \widehat{Res})$ , which are subsequently combined to form the reconstructed error sequence $\widehat{Er}_n$.

5. Final Forecast Reconstruction

The ultimate wind power forecast $\hat{P}_f[n]$, is derived by adding the reconstructed error $\widehat{Er}_n$ to the initial LSTM forecasting $\widehat{P}_I[n]$.

To ensure robustness and consistency of the forecasting framework, a rigorous data pre-processing pipeline was implemented. Min-max normalization was applied to all numerical features to map them to a uniform scale. The target variable (wind power) and the categorical feature (wind category) were excluded from normalization to preserve their interpretability and operational meaning. Although the dataset exhibited only a minimal proportion of missing values, these were carefully reconstructed using spline interpolation, which provides smooth and continuous estimates suitable for time-series data.

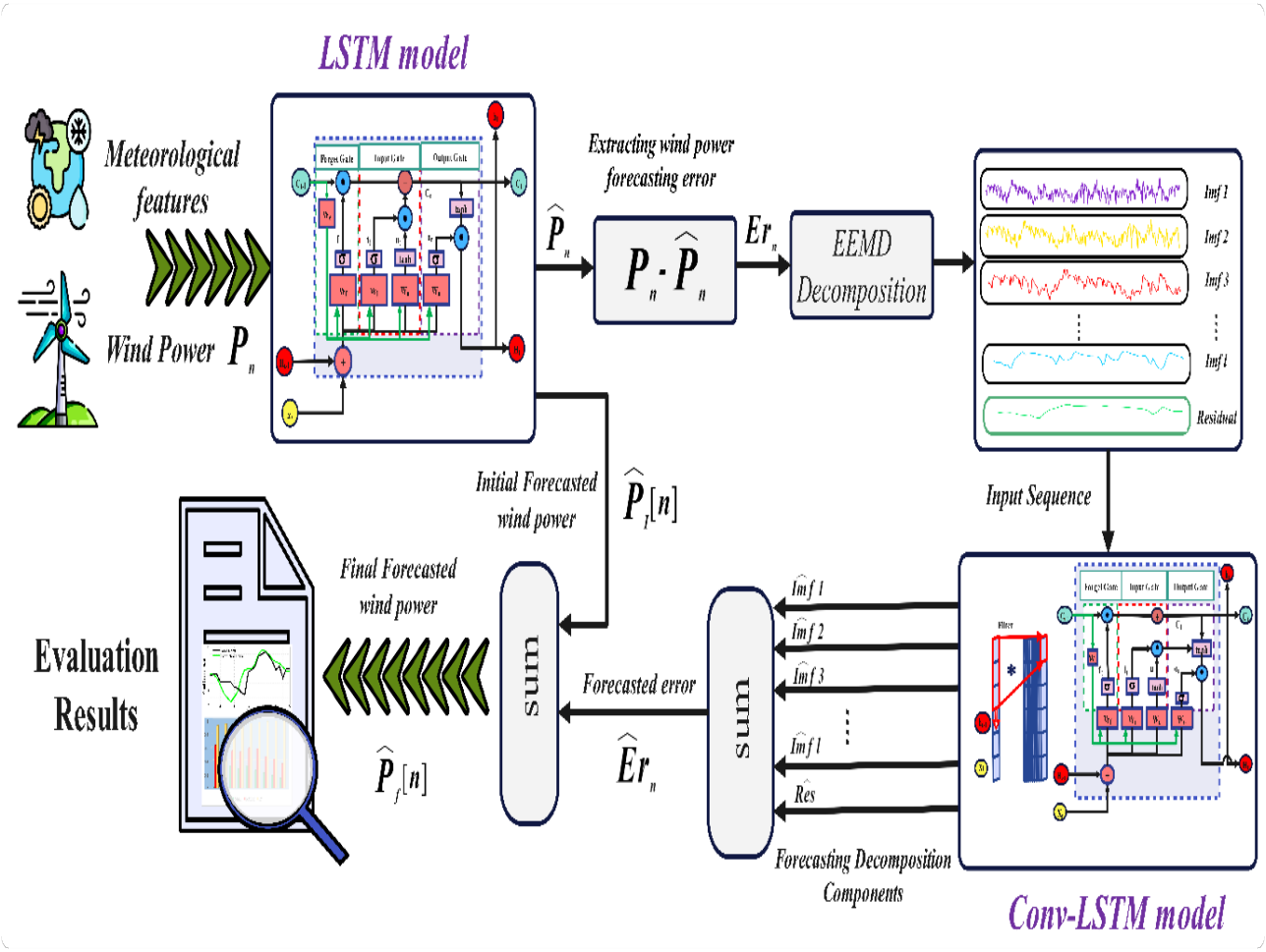



**Figure III.6**: The proposed forecasting structure

For model development and evaluation, the dataset was divided into two distinct subsets, 85% was allocated for training and validation, while the remaining 15% was held out as an unseen test set. Furthermore, a time-series cross-validation strategy with 12 folds was applied to the training subset. This approach generated multiple train-validation splits, allowing the LSTM model to learn from diverse temporal segments and better capture dynamic dependencies in wind power data. The outputs from the validation process were subsequently employed to quantify forecasting errors, which were then used as inputs for the error correction component (EEMD-ConvLSTM). This two-stage process ensured that the final model could maintain reliable performance under varying conditions.

To strike an effective balance between resolution and historical context, extensive experiments were conducted with varying timestep lengths. A window size of nine timesteps was identified as optimal, offering a robust compromise between short-term variability and longer-term temporal structure. Based on this configuration, input matrices were constructed from six

meteorological variables alongside wind power values, forming 7×9 matrices denoted as $x_i$ \(i=1,..,9\). Each matrix captures a sequence of nine consecutive time points ( $t-8, t-7, \ldots, t$), which collectively serve as the historical input context for prediction. These matrices were then supplied to the proposed forecasting framework to predict wind power one step ahead (t+1), as depicted in Figure III.7.

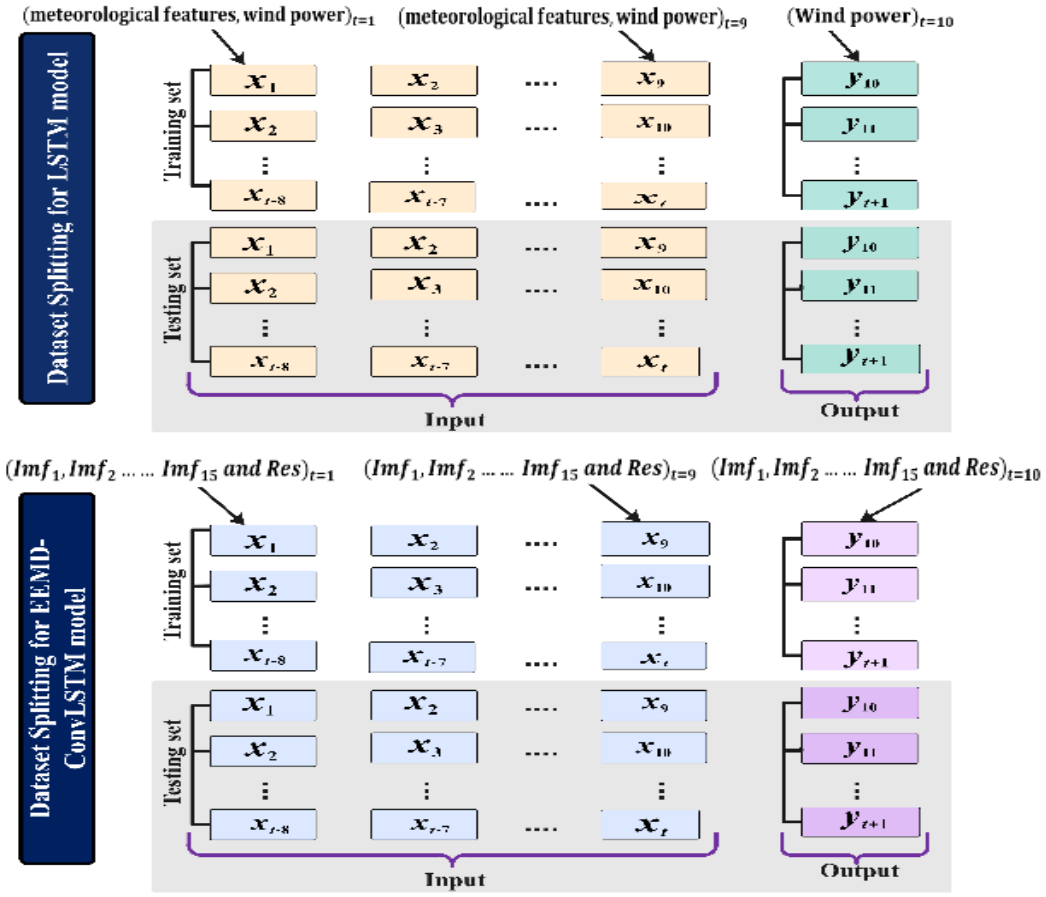



**Figure III.7**: Inputs sequence splitting

The selection of optimal hyperparameters is a crucial and complex step, requiring both structured exploration and in-depth knowledge of the forecasting problem. To tune the hyperparameters, this work employed the GridSearchCV technique in a structured manner, ensuring fair and reproducible selection of model configurations [94]. The finalized hyperparameter settings for both the baseline LSTM and the error correction models are presented in Table III.2. Following training, the LSTM model was validated on unseen data, and the wind power forecasting error $Er_n$, is determined by the difference between the actual wind power $P_n$ and the forecasted values $\hat{P}_n$. These error sequences subsequently serve as inputs to the error correction framework.

**Table III.2:** hyper-parameters of The Proposed Model

| Model | Parameters | Values | Model | Parameters | Values |
|---|---|---|---|---|---|
| **LSTM** | Window size | 9 | **Conv-LSTM** | Input variables | 16 |
| | input variables | 7 | | Window size | 9 |
| | Optimizer | Adam | | Epochs | 300 |
| | Dropout | 0.2 | | Optimizer | Adam |
| | Batch size | 64 | | Batch size | 16 |
| | Epochs | 250 | | Learning rate | 0.001 |
| | Learning rate | 0.001 | | ConvLSTM layers | 3 |
| | LSTM layers | 2 | | FFN layers | 2 |
| | FFN layers | 3 | | ConvLSTM Filters | (128,64,32) |
| | Activation function | ReLU | | ConvLSTM Kernel size | (3,3) |
| | | | | ConvLSTM Activation | ReLU |

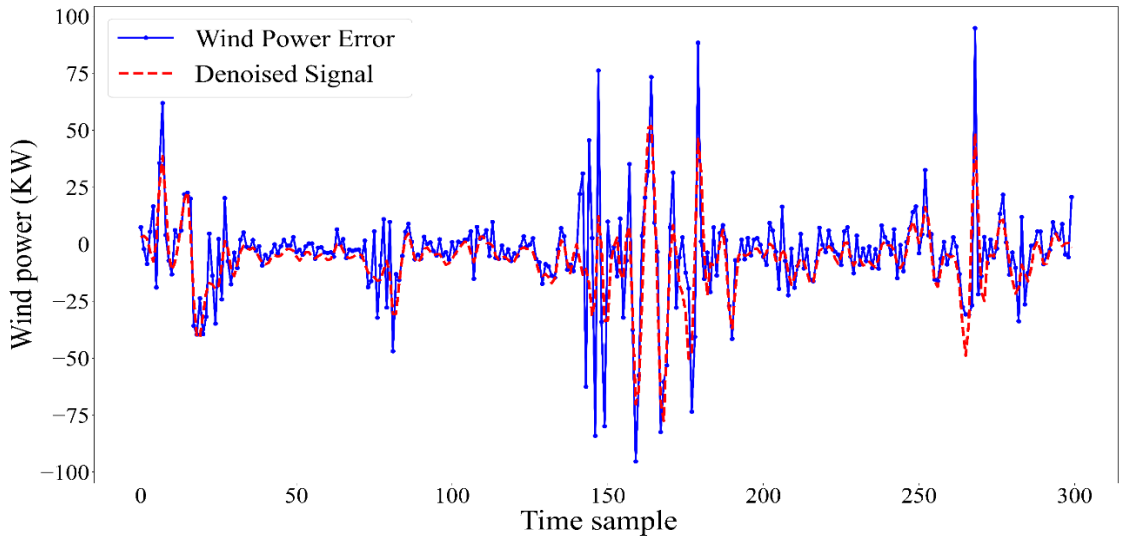



**Figure III.8**: Denoised Error through EEMD

In Figure III.8, the resulting error sequence displays complex nonlinear behaviour, significant variability, and irregular oscillations, indicating that more sophisticated signal processing

methods are required to effectively capture its underlying structure. To address this, EEMD was employed to decompose the error sequence into IMFs, where each IMF captures a specific frequency band of the signal, accompanied by a residual component (Res) capturing the long-term trend. As can be seen in Figure III.9, the EEMD process produced 15 IMFs, which substantially reduced high-frequency noise [93]. The reconstructed error signal is visibly smoother and less volatile compared to the original. Quantitative evaluation further confirms this improvement, using Equation III.21, the Signal-to-Noise Ratio (SNR) was found to improve substantially from 8.14 dB to 13.36 dB which corresponds to a 64.13% increase. Moreover, the reconstructed signal shows a Pearson correlation of 0.985 with the original error series, demonstrating that the EEMD procedure effectively.

$$SNR = 10 \log_{10} \frac{P_1}{P_2} \tag{III.21}$$

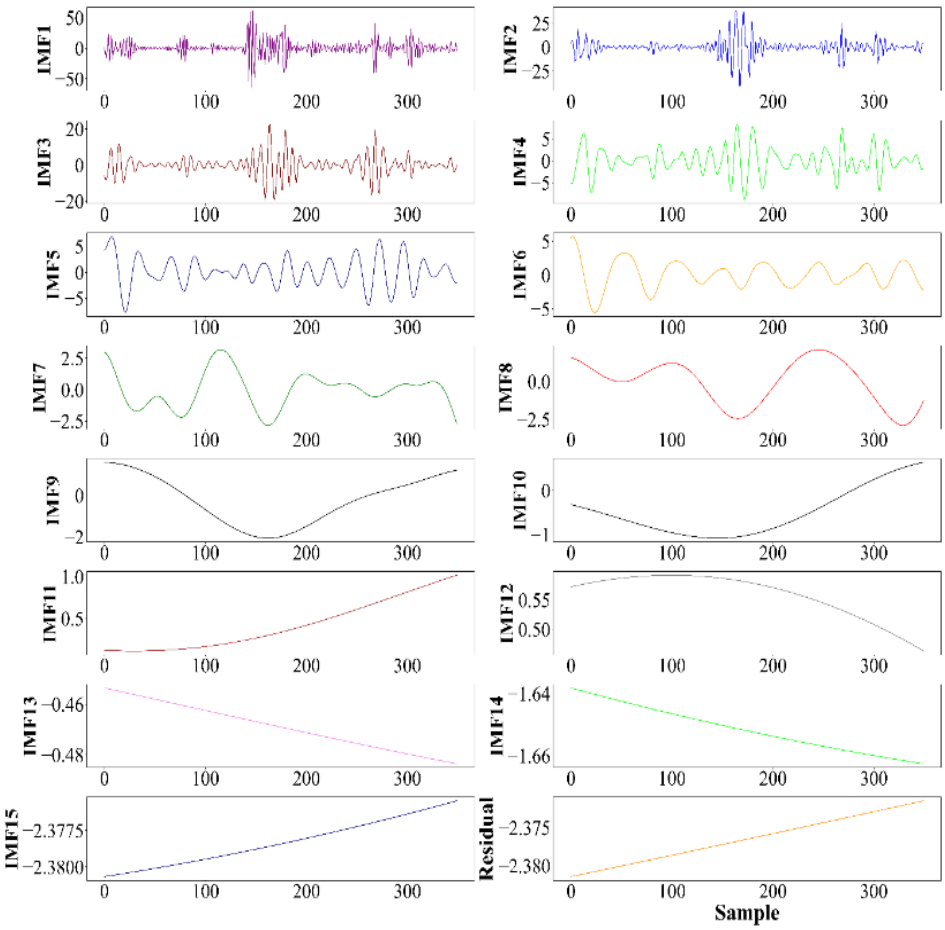



**Figure III.9:** Wind power error decomposed signal

EEMD was selected over traditional denoising techniques, such as wavelet transforms or fixed-basis filters, due to its adaptive and data-driven nature. Unlike methods that rely on predefined basis functions, EEMD directly extracts oscillatory modes from the signal itself.

A 9-step sliding window was used to organize the decomposed error components into 16×9 matrices, with each row representing the sequence from t−8 to t, as depicted in Figure III.7. These organized inputs were subsequently fed into the ConvLSTM model, subsequently predict the future values of each component, yielding $(\widehat{Imf}_1, \widehat{Imf}_2 \ldots\ldots \widehat{Imf}_{15} \wedge \widehat{Res})$.

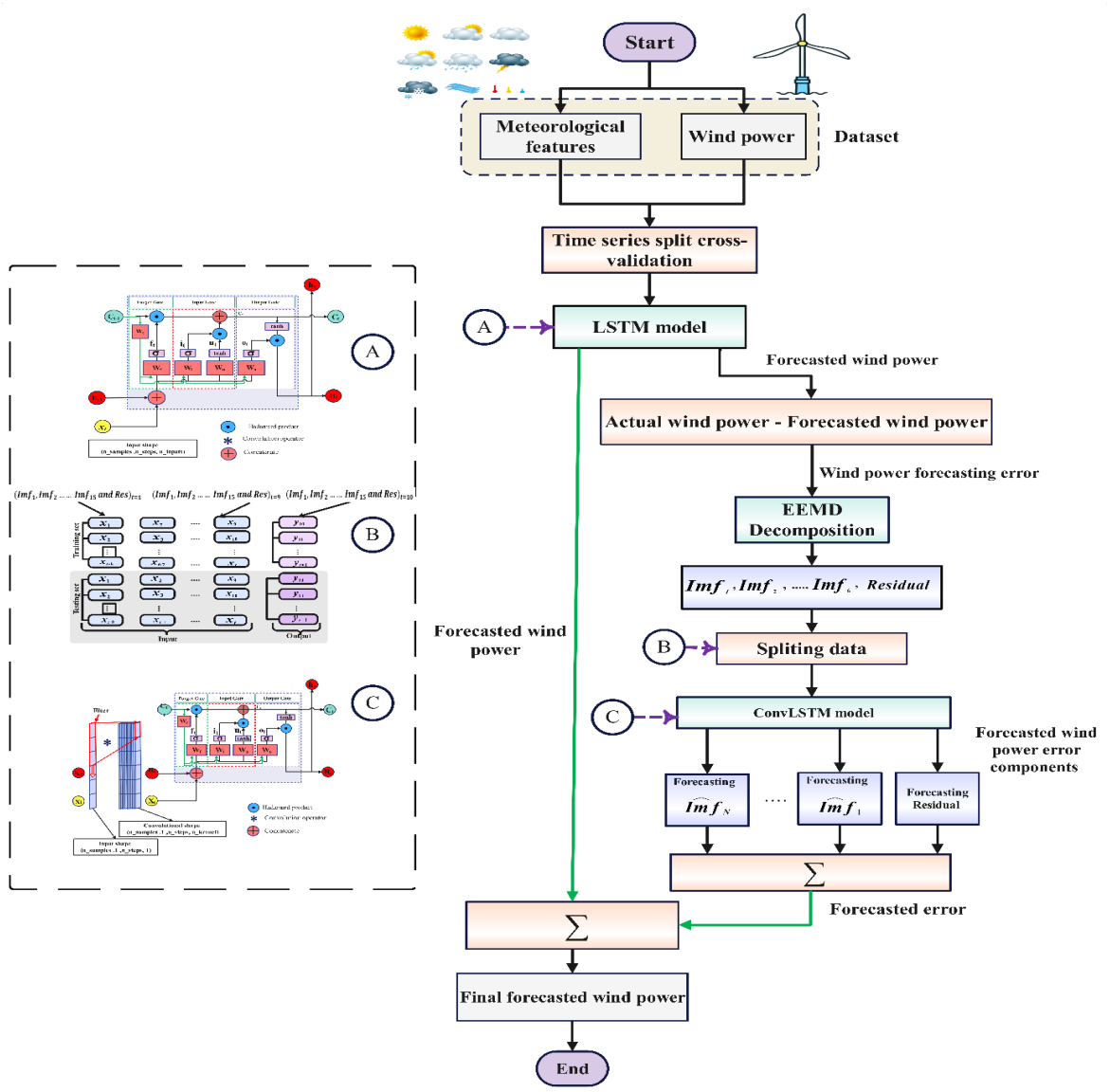



**Figure III.10:** The Proposed Framework

These predicted components were recombined to reconstruct the one-hour-ahead error signal at time t+1. Finally, the estimated error $\widehat{Er}_n$ i was incorporated into the initial LSTM-based wind

power forecast $\widehat{P}_I[n]$, resulting in the corrected forecasting $\hat{P}_f[n]$. The complete process is illustrated in Figure III.10, demonstrating how the error correction mechanism enhances the overall accuracy of wind power forecasting.

## III.5 Wind speed forecasting methodology

### III.5.1 Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN)

Another enhanced version of EMD, called Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN), this method overcomes major drawbacks of conventional EMD, notably mode mixing and sensitivity to noise. It achieves this by generating multiple realizations of the input signal with added adaptive noise and then computing the average of the resulting Intrinsic Mode Functions (IMFs) [38]. This ensemble approach helps smooth out irregularities caused by specific noise patterns or outliers, resulting in more stable and reliable IMFs. The CEEMDAN decomposition process, as illustrated in Figure III.11 follows several sequential steps [95]:

**1-**For each iteration i, a standard Gaussian white noise sequence $\omega^i(t)$, is added to the original time series $y(t)$, forming a perturbed signal, Equation III.21.

$$x^i(t)=y(t)+\varepsilon_0\omega^i(t), i=1,2,3\ldots\ldots,N \tag{III.22}$$

Where $\varepsilon_0$ denotes the noise amplitude coefficient, and N is the total number of noise iterations.

**2-** By applying Equation III.23 and Equation III.24 each noisy signal $x^i(t)$ is decomposed using the EMD operator $E$. The $IMF_1(t)$ is obtained by averaging the decomposition results over all iterations, While the corresponding residual $r_1(t)$ is then computed as the difference between the original signal and the first IMF [95].

$$IMF_1(t)=\frac{1}{N}\sum_{i=1}^{N}E_1\left[x^i(t)\right] \tag{III.23}$$

$$r_1(t)=y(t)-IMF_1(t) \tag{III.24}$$

**3-** Using Equation III.25 and III.26 an adaptive white noise sequence is added to the residual $r_1(t)$ to produce a new signal $r_1(t)+\varepsilon_1 E_1(\omega_i(t))$. This signal is subsequently decomposed with EMD, and the second IMF is derived by averaging across all iterations $IMF_2(t)$. The new residual is obtained by subtracting the $IMF_2(t)$ from $r_1(t)$.

$$IMF_2(t)=\frac{1}{N}\sum_{i=1}^{N} E_1\left(r_1(t)+\varepsilon_1 E_1\left(\omega^i(t)\right)\right) \quad \text{(III.25)}$$

$$r_2(t)=r_1(t)-IMF_2(t) \quad \text{(III.26)}$$

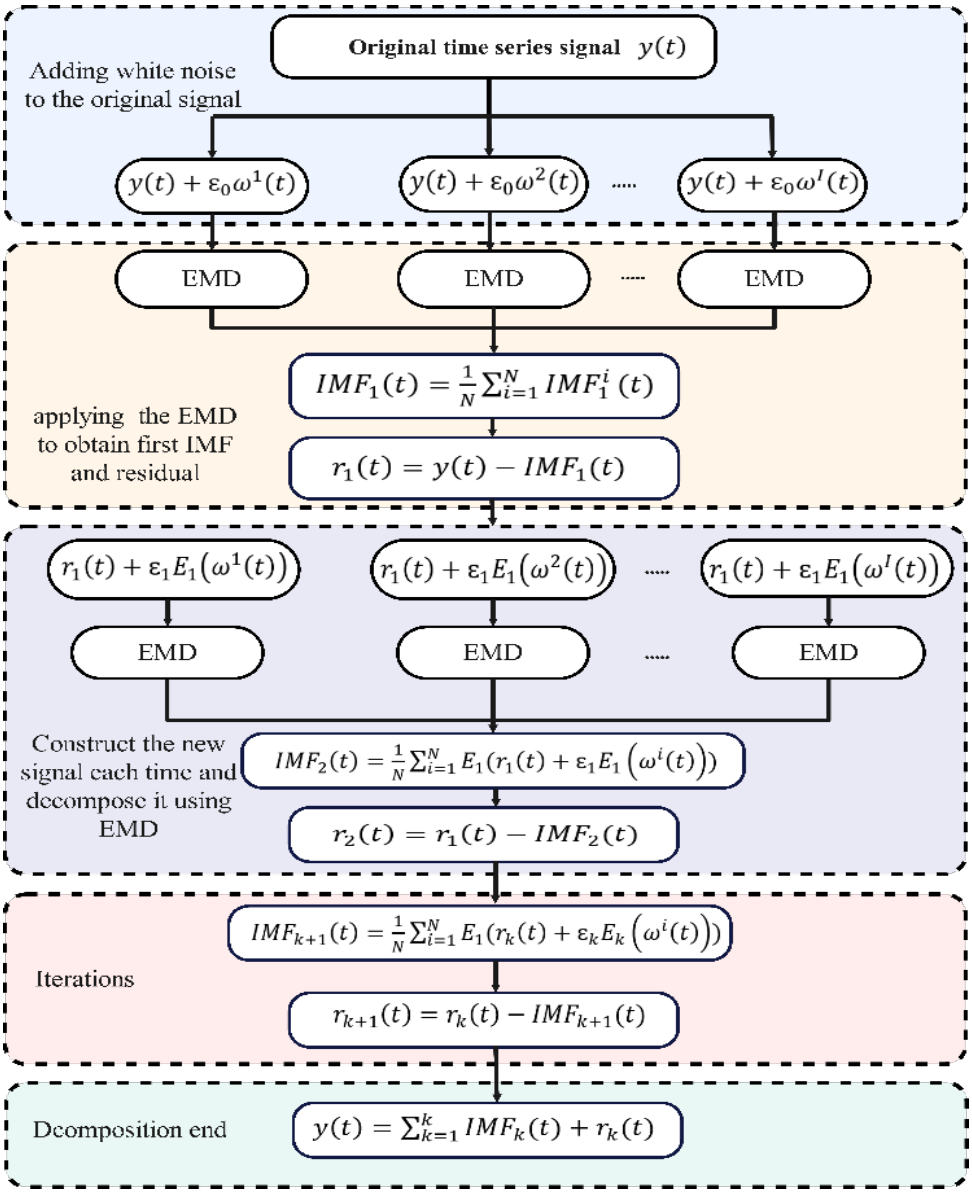



**Figure III.11:** CEEMDAN decomposition flowchart

**4-** Using Equation III.27 and Equation III.28 for $k=2,\ldots,K$ , the kth $IMF_{k+1}(t)$ and $r_{k+1}(t)$ of CEEMDAN are sequentially extracted from the preceding decomposition.

$$IMF_{k+1}(t)=\frac{1}{N}\sum_{i=1}^{N} E_1\left(r_k(t)+\varepsilon_k E_k\left(\omega^i(t)\right)\right) \quad \text{(III.27)}$$

$$r_{k+1}(t)=r_k(t)-IMF_{k+1}(t) \quad \text{(III.28)}$$

**5**-The decomposition process continues iteratively until the residual $r_k(t)$ can no longer be separated, meaning it contains fewer than two extrema [95]. The total number of extracted IMFs is denoted by k, at which point the original time series $y(t)$ is completely represented as the sum of its IMFs and the final residual, as expressed in Equation III.29.

$$y(t)=\sum_{k=1}^{k} IMF_k(t)+r_k(t) \tag{III.29}$$

### III.5.2 Bidirectional GRU Encoder-Decoder structure (Bi-GRU-ED)

The Gated Recurrent Unit (GRU), proposed by Cho et al. in 2014, is a simplified and improved alternative to traditional Recurrent Neural Networks (RNNs), specifically designed to address the vanishing gradient problem in modelling long-term dependencies. The GRU architecture relies on two main gating units: the update gate $z_t$ and the reset gate $r_t$. The update gate determines how much of the earlier information should be carried forward, whereas the reset gate dictates how strongly the previous hidden state contributes to processing the current input. This structure allows the GRU to efficiently preserve relevant information and discard outdated inputs [96]. The GRU's operations are defined mathematically in Equations III.30-III.33.

$$r_t=\sigma(w_r\bullet[h_{t-1},x_t]+b_r) \tag{III.30}$$

$$z_t=\sigma(w_z\bullet[h_{t-1},x_t]+b_z) \tag{III.31}$$

In these expressions, σ refers to the sigmoid function applied within each gate. The input at time t is denoted by $x_t$, and the previous hidden state by $h_{t-1}$. The parameters $w_r$ and $w_z$ constitute the weight matrices for the reset and update gates, whereas $b_r$ and $b_z$ are their associated bias terms [60].

- ❖ Candidate Hidden State

The reset gate $r_t$ controls the degree to which information from the earlier hidden state is allowed to flow into the candidate activation $\tilde{h}_t$, thereby shaping the final hidden state $h_t$ [97].

$$\tilde{h}_t=\tanh(w_h\bullet(r_t\odot h_{t-1},x_t)+b_h) \tag{III.32}$$

The symbol ⊙ denotes the elementwise (Hadamard) multiplication.

- ❖ Hidden state

The hidden state at the current timestep $h_t$, is derived from the candidate activation $\tilde{h}_t$ [60].

$$h_t=\left(z_t \odot h_{t-1}+\left(1-z_t\right) \odot \tilde{h}_t\right) \quad \text{(III.33)}$$

A typical GRU processes the input sequence sequentially in one forward direction, without considering future timesteps. The Bidirectional GRU (Bi-GRU) enhances this by using two GRU layers one processes the sequence forward Equation III.34, and the other in reverse Equation III.35. Their outputs are then combined Equation III.36, enabling the model to capture both past and future context simultaneously [98].

$$\vec{h}_t=GRU_f\left(x_t, \overrightarrow{h_{t-1}}\right) \quad \text{(III.34)}$$

$$\acute{h}_t=GRU_b\left(x_t, \acute{h}_{t+1}\right) \quad \text{(III.35)}$$

$$h_t=[\vec{h}_t \oplus \acute{h}_t] \quad \text{(III.36)}$$

The encoder employs a Bidirectional GRU (Bi-GRU), processing the input sequence $x_t$ in both forward and backward directions to produce a context vector that captures temporal dependencies across the sequence as described in Figure III.12 [77]. Processing the sequence in both directions allows the Bi-GRU encoder to produce a context vector that encapsulates the temporal dependencies present throughout the input sequence. During encoding, the Bi-GRU receives the input sequence $x_t$ in a step-by-step manner. At each time step, the hidden state $h_t$ is updated based on the current input and the previous hidden state. By applying this mechanism across the entire sequence, the encoder captures comprehensive temporal patterns from both directions [34]. The final hidden states from the forward and backward passes are typically combined to form a context vector, which represents a summarized understanding of the entire sequence and serves as the initial hidden state for the decoder [97].

The decoder, also implemented as a Bi-GRU, uses this context vector along with the previously generated output to produce the subsequent output sequence [99]. The output is produced sequentially, generating one timestep at a time, incorporating both the contextual representation learned during encoding and the outputs produced in previous steps. This bidirectional decoding allows the model to reconstruct the target sequence with enhanced temporal awareness, leveraging information from both past and future contexts to improve forecasting accuracy.

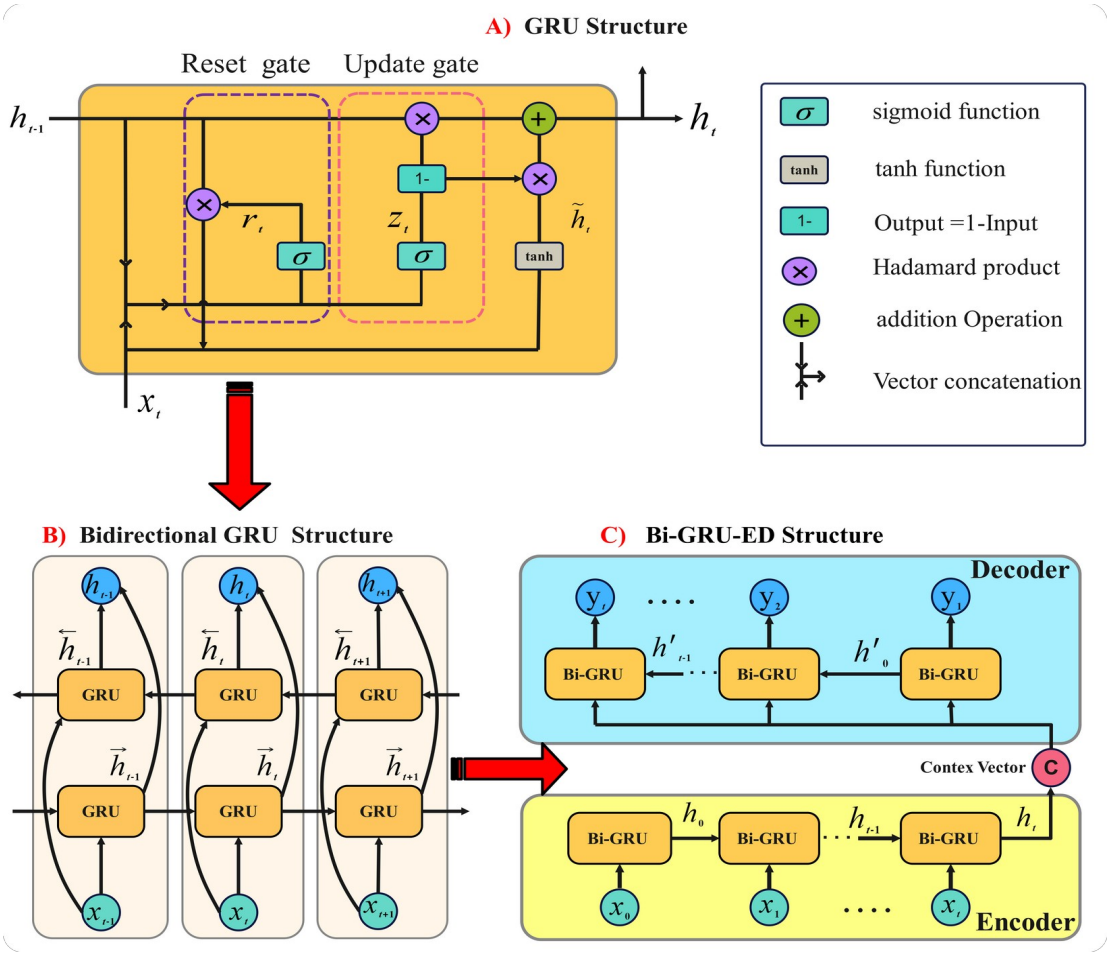



**Figure III.12:** Architectures of GRU, Bi-GRU, and Bi-GRU Encoder-Decoder Models

### III.5.3 Transfer learning

Transfer learning is a robust technique that boosts forecasting accuracy by transferring insights gained from previously trained models and adapting it to related tasks. This strategy is especially beneficial when the target domain has limited data or when building models from scratch would be computationally intensive. Transfer learning can be broadly categorized into two types depending on the relationship between the source and target domains [100].

- Homogeneous transfer learning: This occurs when both the feature space X and the label space y of the source domain ($X^S$, $y^S$) and the target domain ($X^T$, $y^T$) are identical; that is, $X^S = X^T$ and $y^S = y^T$.

- Heterogeneous transfer learning: In contrast, this occurs when either the feature space or the label space between the source and target domains differs ($X^S \neq X^T$, $y^S \neq y^T$). This technique is often utilized when data in the target domain is scarce or the task is highly complex, facilitating the effective transfer of knowledge across domains to enhance model learning and performance [101].

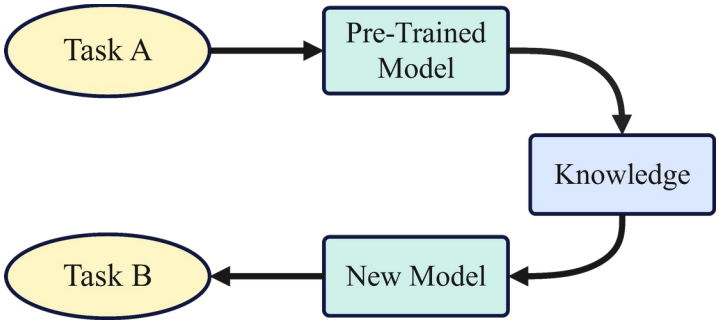



**Figure III.13:** Transfer Learning Concept

In this work, a homogeneous transfer learning approach is employed, given the strong similarity between the source and target tasks, both focused on short-term wind speed forecasting. The source task (Task 1) entails training a Bi-GRU-ED model to predict the residual component of wind speed, extracted through CEEMDAN decomposition, using a multivariate input dataset. The learned parameters from the pre-trained model are subsequently applied to the target task, which focuses on predicting the IMFs obtained via CEEMDAN decomposition. Due to the limited size of each IMF dataset, the model undergoes fine-tuning, with all layers of the pre-trained Bi-GRU-ED retrained for each specific IMF. Thus, the contributions of this Framework are as follow.

- ✓ It eliminates the necessity of developing and training individual deep learning models for each IMF.
- ✓ It lowers both computational demands and training duration.
- ✓ It provides more stable initialization and better convergence by leveraging temporal dynamics already learned in the source task.

### III.6 Wind speed forecasting Framework

The CEEMDAN-Bi-GRU-ED-TL forecasting framework is implemented according to the sequential steps depicted in Figure III.14.

1. A six-level CEEMDAN decomposition is performed to separate the original wind speed series $w[n]$ into six Intrinsic Mode Functions $(IMF_1,\ldots,IMF_6)$ and a residual component ( *Res*). Concurrently, the remaining meteorological variables ($T$ , $P \wedge W_{10}$) are scaled to the [0,1] interval using min-max normalization
2. The *Res*, representing the low-frequency trend, is predicted using a Bi-GRU Encoder-Decoder model. This model receives as inputs the decomposed IMFs, the residual, and the normalized meteorological variables ($T_n, P_n, w_{10n}$), organized over a 10-time-step window. Once trained, the model is stored as $Bi\text{-}GRU\text{-}ED_{Res}$.

3. The $Bi\text{-}GRU\text{-}ED_{Res}$ model is then adapted for each of the six IMFs, yielding six dedicated models, $Bi\text{-}GRU\text{-}ED_{l}\ (l=1,\ldots,6)$. Each model forecasts its corresponding $\widehat{IMF}_{l}$ using the full set of inputs the six IMFs, the residual component, and the normalized meteorological features effectively transferring and leveraging the knowledge gained from residual model training.
4. Ultimately, the reconstructed wind speed forecast $\hat{w}[n]$ is obtained by summing the forecasted residual $\widehat{Res}$ and the six forecasted IMFs ($\widehat{IMF}_{l}(l=1,\ldots,6)$), following the additive reconstruction approach inherent to CEEMDAN.

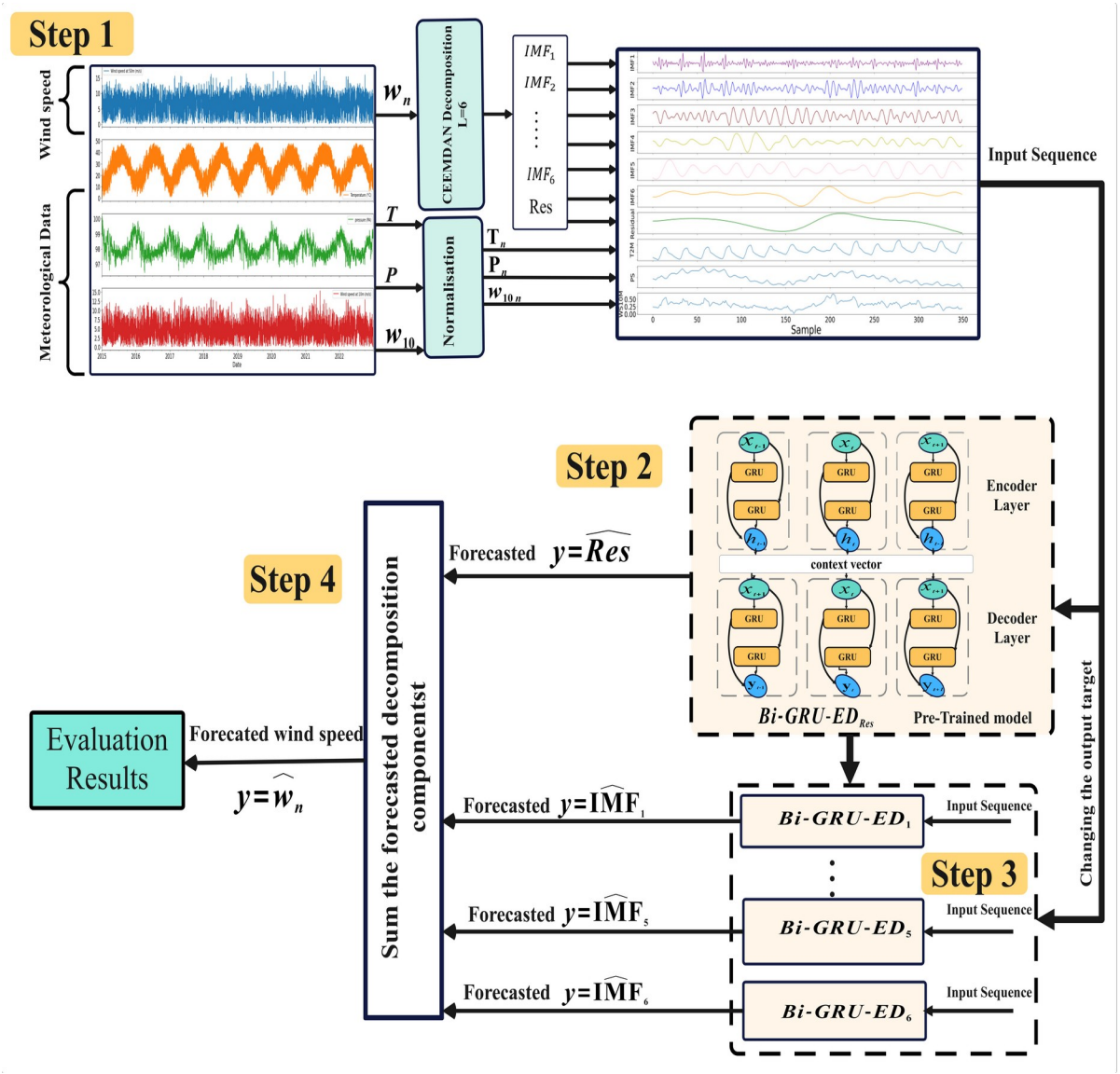



**Figure III.14:** Schematic representation of the proposed CEEMDAN-Bi-GRU-ED framework with transfer learning (TL)

In this study, the dataset is divided into two subsets, with 80% allocated for training and the remaining 20% reserved for testing, as summarized in Table III.3. To further ensure robustness

and prevent overfitting, an 8-fold cross-validation strategy is employed within the training portion. This approach enables the model to be iteratively trained and validated on different subsets of data, thereby providing a more reliable assessment of its generalization capability and stability across unseen scenarios [102].

**Table III.3:** Statistical Details of Wind Speed Datasets

| Location | | Samples | Metrics | | | |
|---|---|---|---|---|---|---|
| | | | Max (m/s) | Min (m/s) | Mean (m/s) | Std (m/s) |
| Location 1 (Adrar) | Full Samples | 70128 | 18.37 | 0.6 | 6.64 | 2.65 |
| | Training-Samples | 58432 | 18.37 | 1.5 | 6.65 | 2.82 |
| | Testing-Samples | 11686 | 16.05 | 0.6 | 5.84 | 2.26 |
| Location 2 (Tataouine) | Full samples | 61343 | 19.83 | 0.4 | 6.02 | 3.73 |
| | Training-Samples | 51111 | 18.50 | 1 | 7.18 | 4.22 |
| | Testing-Samples | 10222 | 19.18 | 0.4 | 6.23 | 3.79 |

To effectively capture the non-linear and multi-scale characteristics of wind speed signals, the CEEMDAN technique was employed to decompose the original series w[n] into IMFs and a Res Figure III.15. The decomposition quality and subsequent forecasting performance are highly sensitive to the choice of CEEMDAN parameters. In this work, the IMFs number empirically limited to six (i.e., N=6), a value that was found to preserve the dominant frequency content of the signal while reducing the risk of over-decomposition or noise amplification. The ensemble size was fixed at 100, a commonly adopted value in the literature that ensures robustness without excessive computational burden. To mitigate mode mixing while maintaining signal integrity, the amplitude of the added white noise was set to 20% of the original signal's standard deviation. Cubic spline interpolation was employed to construct the envelopes during the sifting process, as it ensures smooth and precise fitting. Additionally, the number of sifting iterations per mode was limited to 500 to guarantee convergence and prevent excessive computation. These parameter choices were determined through a combination of previous empirical findings and initial testing performed on the target dataset, ensuring an optimal trade-off between accuracy and computational efficiency.

In parallel with the CEEMDAN decomposition, the remaining meteorological input features are normalized using min-max scaling to the range [0,1], as formulated in Equation III.37. This normalization procedure, denoted as ($T_n, P_n, w_{10n}$) is essential for eliminating scale-related disparities among features, which could otherwise lead the model to favour variables with inherently larger magnitudes. By ensuring that all inputs are on a comparable scale.

$$X_i^{'} = \frac{X_i - X_{min}}{X_{max} - X_{min}} \tag{III.37}$$

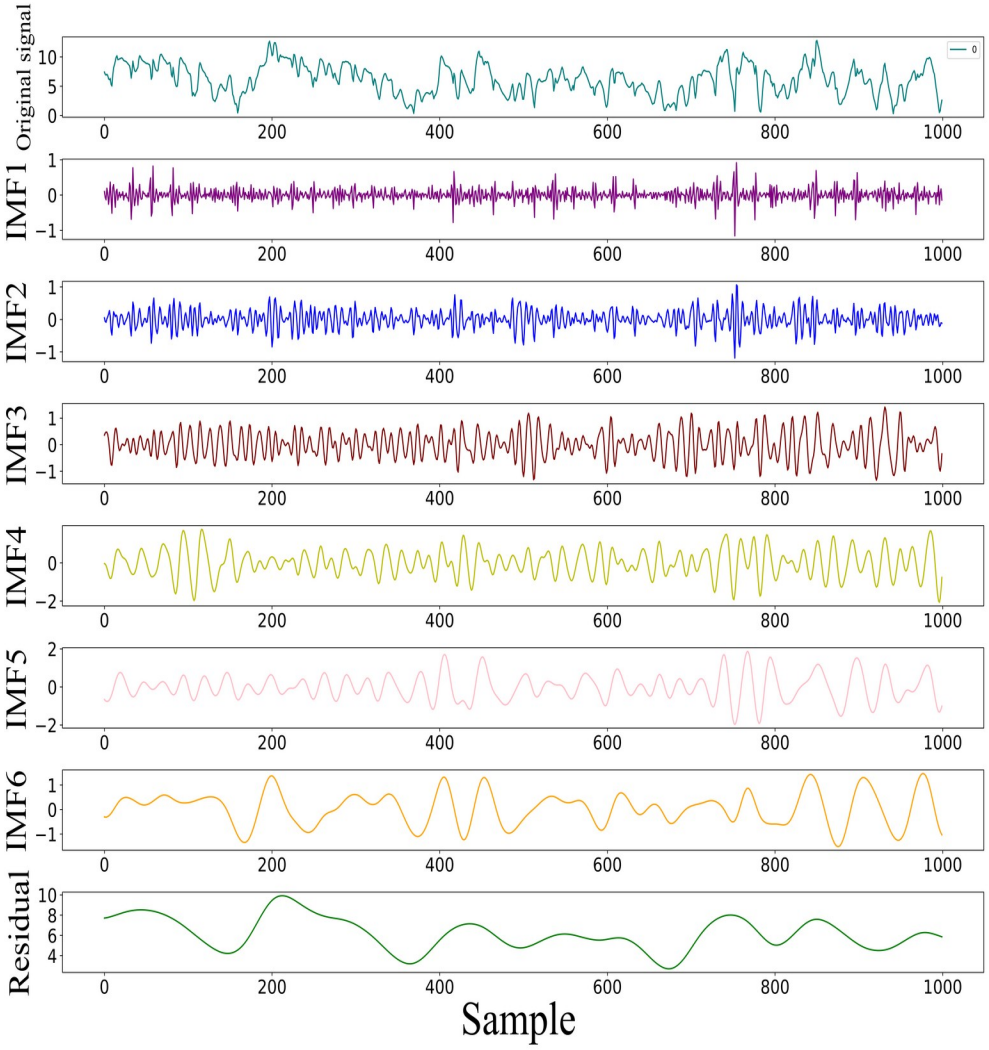



**Figure III.15:** Decomposition of Adrar Wind Speed Data Using CEEMDAN.

Following CEEMDAN decomposition and feature normalization, the residual component, capturing the underlying low-frequency trend, forecasted using *Bi-GRU-ED*$_{Res}$ which trained using the hyperparameters in Table III.4. The input to this model is generated using a sliding window strategy, which captures temporal dependencies across successive time steps. Through empirical tuning, a historical window size of 10 was determined to yield optimal predictive

performance. Thus, each input sample comprises the normalized features ($T_n, P_n, w_{10n}$) along with the decomposed components over 10 successive time steps, organized into a structured 10×10 matrix $x^{(i)}$ where $i=1,..,10$ corresponds to the time steps (t-9, t-8, …,t).These inputs configuration is designed to capture short-term dynamics while mitigating risks of overfitting or insufficient context. The model outputs the forecasted residual value at time t+1, as depicted in Figure III.16. The forecasting framework is then extended via transfer learning to predict each individual IMF, requiring adjustment solely of the target output while retaining the learned model parameters.

**Table III.4:** *Bi-GRU-ED*$_{Res}$ Model Hyperparameters

| Parameters | value |
|---|---|
| Input variables | 10 |
| Window size | 10 |
| Epochs | 250 |
| Optimizer | Adam |
| Batch size | 260 |
| Epochs | 250 |
| Learning rate | 0.001 |
| Layers | 3 |
| activation function | (ReLU, ReLU, Linear) |

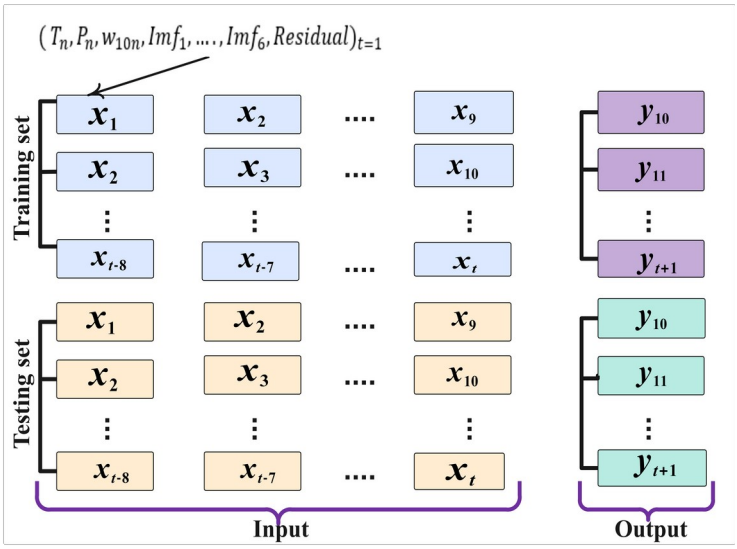



**Figure III.16:** Input-Output data structure

The structured input dataset is first employed to train the *Bi-GRU-ED*$_{Res}$ model, with the objective of forecasting the residual (low-frequency). Prioritizing the residual is essential, as it

captures the dominant low-frequency trend, enabling the encoder-decoder architecture to learn a robust temporal representation. Throughout training, the $Bi\text{-}GRU\text{-}ED_{Res}$ model identifies and internalizes the temporal dynamics of the residual component. Upon convergence, the fully trained model, complete with its optimized weights and biases, is preserved. This pre-trained model serves as the foundation for transfer learning, rather than training separate models for each IMF from the ground up, the pre-trained $Bi\text{-}GRU\text{-}ED_{Res}$ is fine-tuned to forecast each individual IMF, resulting in models $Bi\text{-}GRU\text{-}ED_l, l=1,\ldots,6$. After all the extraction.

This reuse of temporal knowledge enhances learning efficiency, reduces training time, and lowers computational overhead. Finally, the forecasted residual $\widehat{Res}$ and $\widehat{IMF}_l\,(l=1,\ldots,6)$ are aggregated following the CEEMDAN reconstruction principle to yield the final wind speed forecast $\hat{w}[n]$. The overall proposed model architecture is illustrated in Figure III.17.

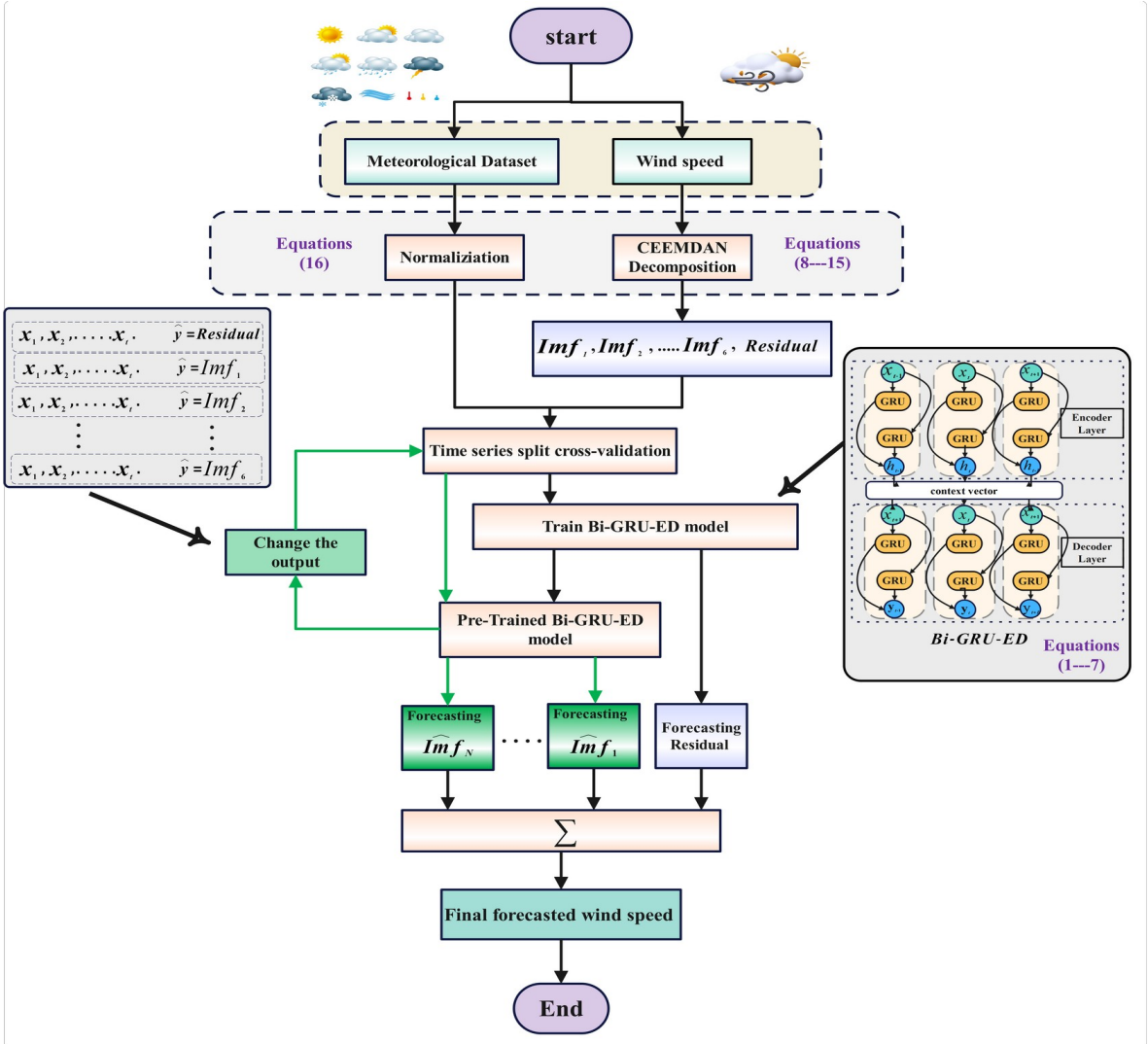



**Figure III.17:** Flowchart of the proposed forecasting architecture

## III.7 Computational Environment and Implementation Tools

The development and evaluation of the proposed forecasting frameworks were conducted in a computational environment centred on the Python programming language, chosen for its versatility, reproducibility, and rich ecosystem of scientific libraries. All experiments were carried out using the Spyder Integrated Development Environment (IDE) under the Anaconda distribution on Windows 11 (64-bit) which can be seen in Figure III.18. Spyder provided an efficient MATLAB-like interface with advanced debugging, variable exploration, and visualization utilities, thereby streamlining the workflow of implementing, testing, and refining the forecasting models.

Model implementation was primarily supported by TensorFlow and Keras, which enabled the design, training, and optimization of advanced deep learning architectures. Additional Python packages ensured a robust computational pipeline NumPy and Pandas for efficient data manipulation, Scikit-learn for pre-processing and statistical evaluation, and Matplotlib along with Seaborn for comprehensive visualization of time series behaviour, model predictions, and error distributions.

All simulations were executed on a Windows-based workstation equipped with a multi-core processor and sufficient memory resources to support large-scale training and multi-horizon forecasting tasks. This configuration allowed efficient handling of decomposition-based hybrid models while ensuring reproducibility and scalability.

## III.8 Conclusion

This chapter has presented the methodological frameworks developed for wind power and wind speed forecasting, employing advanced deep learning and hybrid modelling approaches. By utilizing real-world meteorological datasets from representative sites, the study implemented both standalone architectures such as LSTM and ConvLSTM and hybrid frameworks that integrate signal decomposition techniques, including EEMD and CEEMDAN, with neural network models.

For wind power forecasting, the integration of EEMD with LSTM and ConvLSTM enabled the models to better capture the nonlinear and non-stationary dynamics data. In contrast, wind speed forecasting adopted a more comprehensive strategy through the combination of CEEMDAN and a Bidirectional GRU Encoder-Decoder, further reinforced with transfer learning to enhance adaptability and cross-site generalization.

The methodological design encompassed rigorous pre-processing, systematic hyperparameter tuning, and robust training-validation procedures, all aimed at ensuring forecasting accuracy and stability under diverse conditions. These frameworks lay the groundwork for the detailed performance evaluation, benchmarking, and comparative analyses presented in the following chapter.

In summary, this chapter has established a strong methodological foundation describing how tailored AI-based forecasting schemes can be used to wind energy forecasting challenges.

# Chapter IV: Results and Discussions

# Results and Discussions

## IV.1 Introduction

This chapter presents and analyzes the simulation results for the two core contributions of this thesis: (1) the proposed hybrid wind power forecasting framework and (2) the advanced wind speed forecasting model. The analysis focuses on a comparative evaluation, robustness testing, and performance across multiple forecasting horizons.

For wind power forecasting, the effectiveness of the proposed hybrid model is validated through a rigorous comparative study. Its performance is benchmarked against a diverse set of models**,** including classical statistical approaches, conventional machine learning techniques, and state-of-the-art deep learning and hybrid architectures. Specifically, the evaluation includes comparisons with ARIMA, SVR, BP neural networks, LSTM (and its error-corrected variants)**,** ConvLSTM, as well as hybrid models such as EEMD-LSTM, CNN-LSTM, CNN-BiGRU, AE-GRU, and LSTM-ARIMA. By evaluating these models across multiple time horizons, this chapter demonstrates not only the superior accuracy of the proposed framework but also its crucial adaptability, a key requirement for the reliable integration of wind power into modern energy systems.

In parallel, for wind speed forecasting this chapter evaluates the proposed CEEMDAN-Bi-GRU-ED-TL framework, which leverages decomposition and transfer learning to enhance performance. The assessment is conducted from different analysis prospects:

1. Evaluating the Bi-GRU-ED model by comparing its one-hour-ahead forecasting performance with eight baseline approaches, namely SVR, RNN, CNN, LSTM, GRU, Bi-LSTM, Bi-GRU, and CNN-LSTM.
2. Comparing the proposed structure against other advanced hybrid models that integrate EEMD and CEEMDAN decomposition techniques.
3. Analyzing the contribution of the transfer learning (TL) component by contrasting the forecasted intrinsic mode functions (IMFs) with and without its application.
4. Evaluating the predictive accuracy of the Bi-GRU-ED, CEEMDAN-Bi-GRU-ED, and CEEMDAN-Bi-GRU-ED-TL models across multiple time horizons.

Through these comprehensive experiments, the chapter provides robust validation of the proposed methodologies, culminating in a critical discussion of their performance, limitations, and practical implications for wind energy integration.

## IV.2 Wind Power Forecasting: Results and Analysis

To evaluate the performance of the proposed forecasting framework, an extensive comparison was conducted against a range of benchmark models. The evaluation included traditional statistical methods, such as the Autoregressive Integrated Moving Average (ARIMA) and Support Vector Regression (SVR), as well as conventional machine learning and deep learning approaches, including Backpropagation (BP) neural networks, standard LSTM networks, and Convolutional LSTM (ConvLSTM). Furthermore, enhanced variants including an LSTM with ConvLSTM-based error correction were incorporated to establish a fair and rigorous basis for comparison.

Beyond these established baselines, the study extended the comparison to include several state-of-the-art hybrid architectures recognized in recent forecasting literature, such as CNN-BiGRU, Autoencoder-GRU (AE-GRU), EEMD-LSTM, CNN-LSTM, and LSTM-ARIMA. Evaluating the proposed model against this wide spectrum of approaches ensured a robust validation of its predictive capabilities.

To further assess its practical utility, the model's forecasting accuracy was rigorously evaluated across multiple time horizons. This multi-horizon evaluation not only highlights the model’s predictive accuracy under short-term conditions but also demonstrates its adaptability and stability when extended to longer forecasting intervals. Such a rigorous and layered analysis provides a holistic understanding of the model’s performance, conclusively establishing its robustness, reliability, and superior advantages over existing approaches.

### IV.2.1 Evaluation Against Benchmark Models

The performance of the proposed LSTM error-correction framework, integrating EEMD with ConvLSTM, was rigorously assessed through extensive benchmarking. As shown in Table IV.1 and illustrated in Figure IV.1 and Figure IV.2, the proposed model consistently outperformed all competing approaches across the primary evaluation metrics. It achieved the lowest MAE of 4.22 ± 0.31, RMSE of 9.31 ± 0.80, and MAPE of 0.19%, while attaining the highest $R^2$ at 0.9811. These results underscore the model’s capacity to minimize forecasting errors and effectively capture the complex temporal dynamics inherent in offshore wind power data.

**Table IV.1:** Comparative performance results of the proposed model against benchmark models (kW).

| Model | MAE (95% CI) | RMSE (95% CI) | $R^2$ | MAPE | p-value vs. Proposed Model |
|---|---|---|---|---|---|
| **LSTM Error-Correction (EEMD-ConvLSTM)** | **4.22 ± 0.31** | **9.31 ± 0.80** | **0.9811** | **0.19** | **-** |
| **LSTM Error-Correction (ConvLSTM)** | 4.60 ± 0.40 | 13.64 ± 1.20 | 0.9772 | 0.25 | 0.03 |
| **ConvLSTM** | 5.74 ± 0.56 | 18.33 ± 1.72 | 0.9496 | 0.36 | 0.02 |
| **LSTM** | 6.14 ± 0.64 | 22.69 ± 2.16 | 0.9355 | 0.53 | 0.015 |
| **BPNN** | 8.32 ± 0.82 | 31.90 ± 2.99 | 0.8937 | 0.83 | <0.001 |
| **ARIMA** | 11.36 ± 1.45 | 39.62 ± 3.85 | 0.8619 | 1.12 | <0.001 |
| **SVR** | 23.22 ± 2.20 | 48.61 ± 4.74 | 0.7835 | 1.58 | <0.001 |

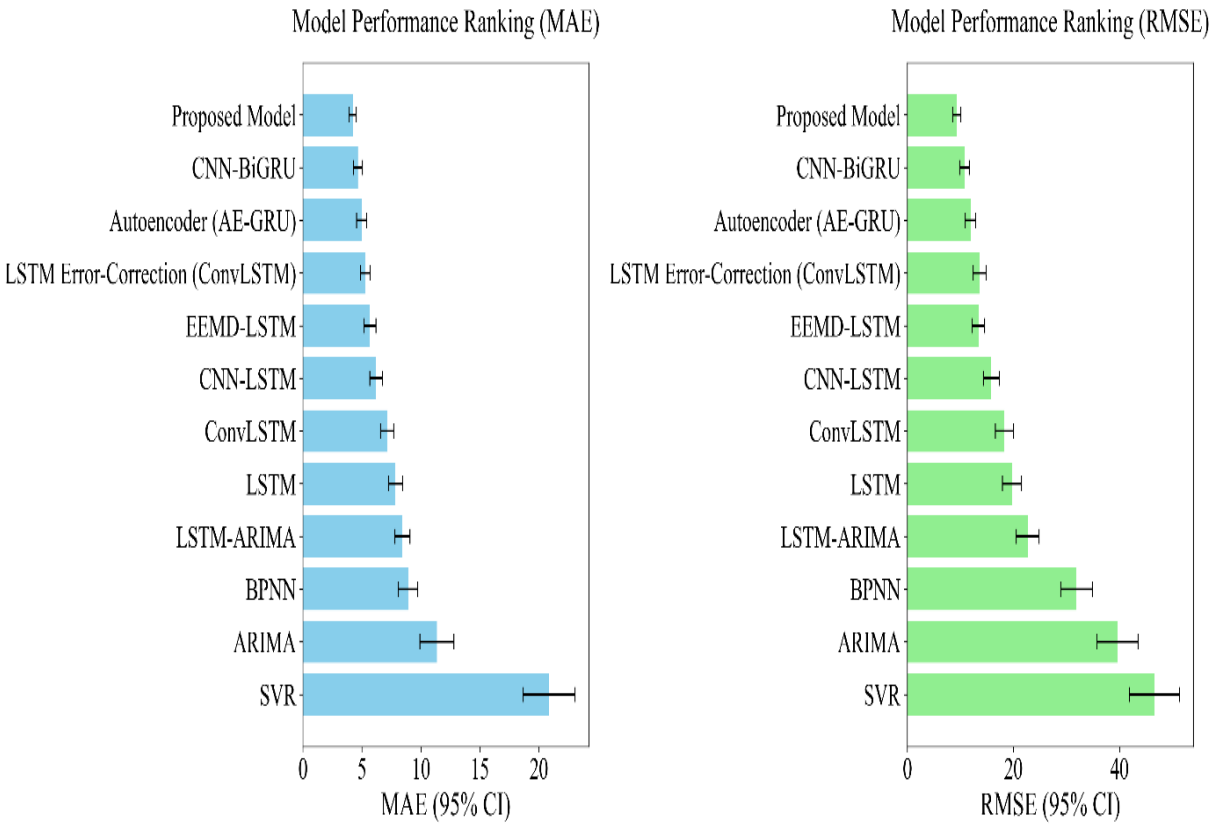



**Figure IV.1:** Comparison of forecasting models based on MAE and RMSE, including 95% confidence intervals.

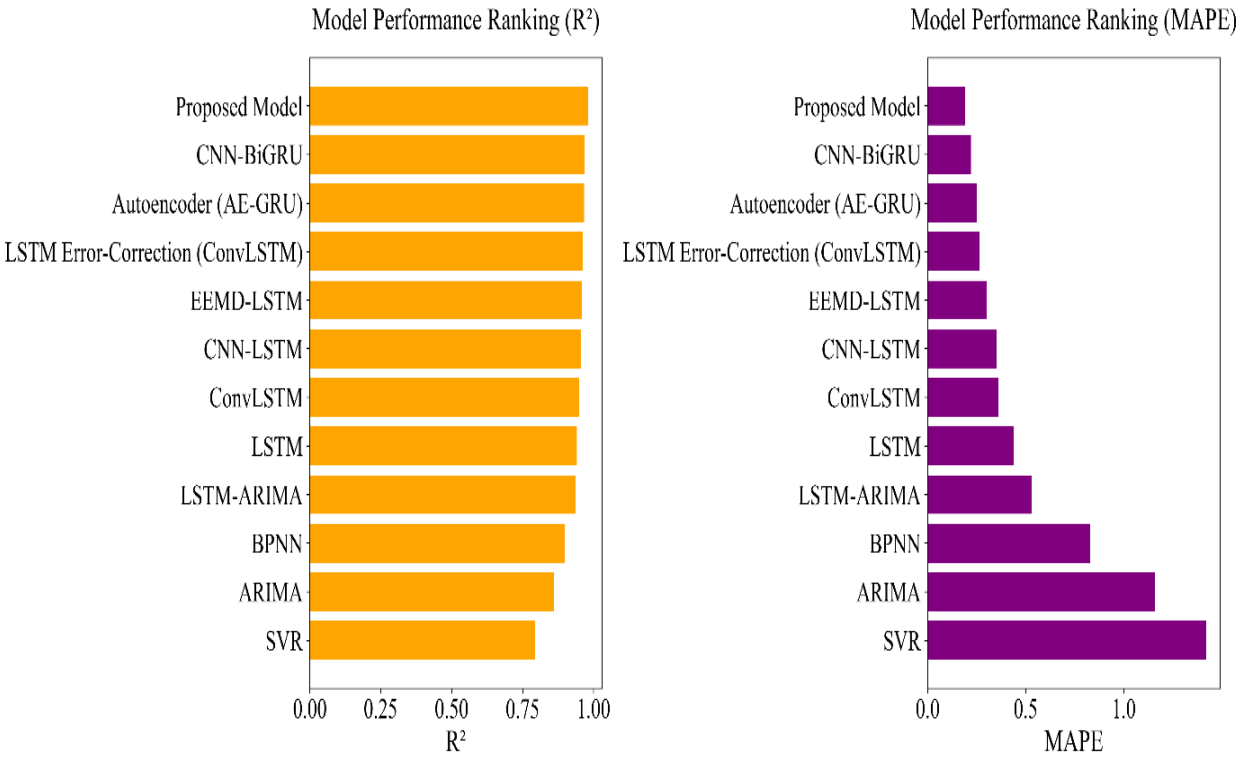



**Figure IV.2:** Comparative analysis of forecasting models in terms of MAPE and coefficient of determination ($R^2$)

A key advantage of the EEMD-ConvLSTM design lies in its decomposition strategy. This separation enables ConvLSTM to model local and global temporal-spatial patterns more effectively by focusing on individual components, thereby improving error correction. The statistical robustness of these results is validated through p-values presented in Figure IV.3, which show significance levels of 0.03 when compared to the ConvLSTM-based error correction model, 0.02 against standard ConvLSTM, and less than 0.001 relative to ARIMA, BPNN, and SVR confirming that improvements are not attributable to random variation.

Although the ConvLSTM error-correction model without EEMD also demonstrated strong results (MAE: 4.60 ± 0.40, RMSE: 13.64 ± 1.20, MAPE: 0.25%, $R^2$: 0.9772), it was slightly outperformed by the full EEMD-ConvLSTM approach. The inclusion of EEMD provided an 8.3% reduction in MAE and a 31.7% reduction in RMSE, evidencing its value in mitigating the nonlinear and non-stationary components of the forecasting error.

The standard ConvLSTM, applied without error correction, also yielded improvements compared to basic LSTM (MAE: 5.74 ± 0.56, RMSE: 18.33 ± 1.72, $R^2$: 0.9496 versus MAE: 6.14 ± 0.64, RMSE: 22.69 ± 2.16, $R^2$: 0.9355). These findings demonstrate that convolutional layers enhance the ability of recurrent networks to model spatial-temporal correlations,

delivering a 6.5% reduction in MAE and a 19.2% reduction in RMSE compared with conventional LSTM.

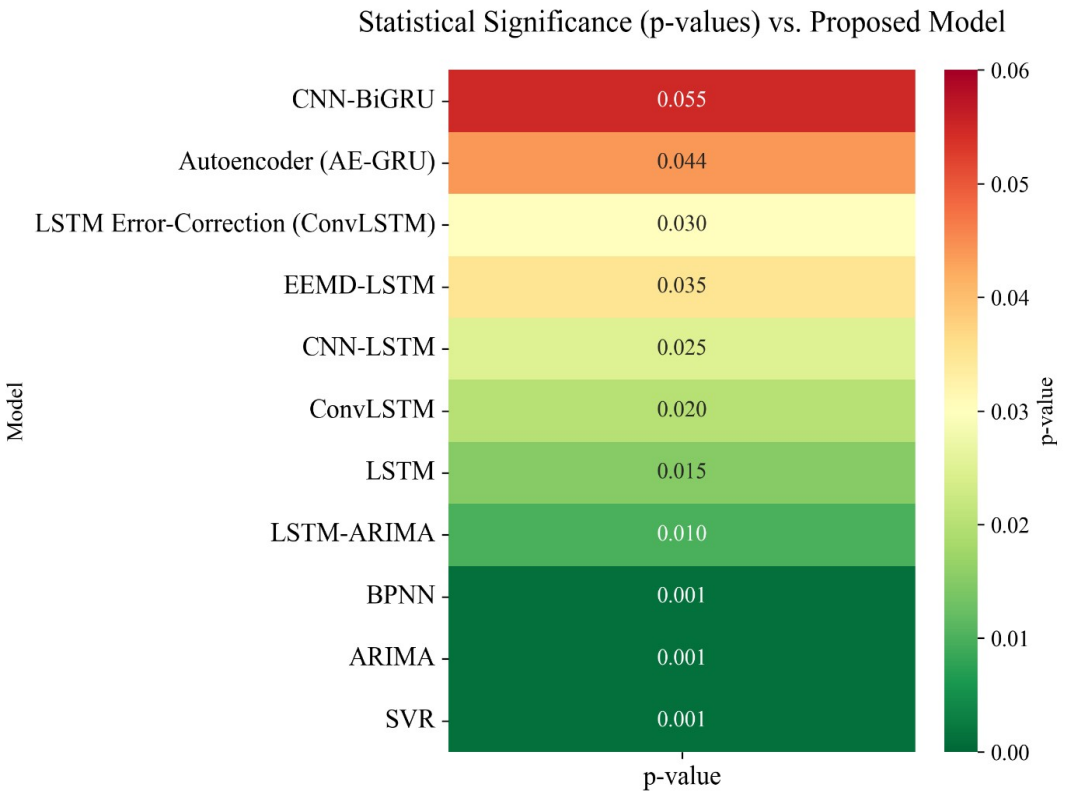



**Figure IV.3:** P-Values Comparing Benchmark Models to the Proposed Forecasting Model

By contrast, traditional forecasting methods such as ARIMA, SVR, and BPNN lagged considerably behind. The BPNN achieved moderate results (MAE: 8.32 ± 0.82, RMSE: 31.90 ± 2.99, $R^2$: 0.8937), while ARIMA (MAE: 11.36 ± 1.45, RMSE: 39.62 ± 3.85, $R^2$: 0.8619) and SVR (MAE: 23.22 ± 2.20, RMSE: 48.61 ± 4.74, $R^2$: 0.7835) lagged considerably in forecasting accuracy. These outcomes reaffirm the limitations of classical statistical and machine learning methods in handling the high variability and nonlinear behaviour of wind power time series, highlighting the clear advantages of deep learning-based frameworks with structured error modelling.

The superiority of the proposed model is further confirmed through visual analysis. As illustrated in Figure IV.4, the EEMD-ConvLSTM trajectories exhibit close alignment with ground-truth power values, even during sudden fluctuations. Although ConvLSTM with error correction produced strong results, minor deviations were observed during abrupt transitions. Traditional approaches and basic LSTM, by contrast, displayed larger discrepancies and weaker adaptability under dynamic conditions. To better highlight the performance gap among advanced methods, Figure IV.5 compares the top three models EEMD-ConvLSTM,

ConvLSTM with error correction, and standard ConvLSTM further validating the robustness and predictive reliability of the proposed approach across both smooth and volatile power variations.

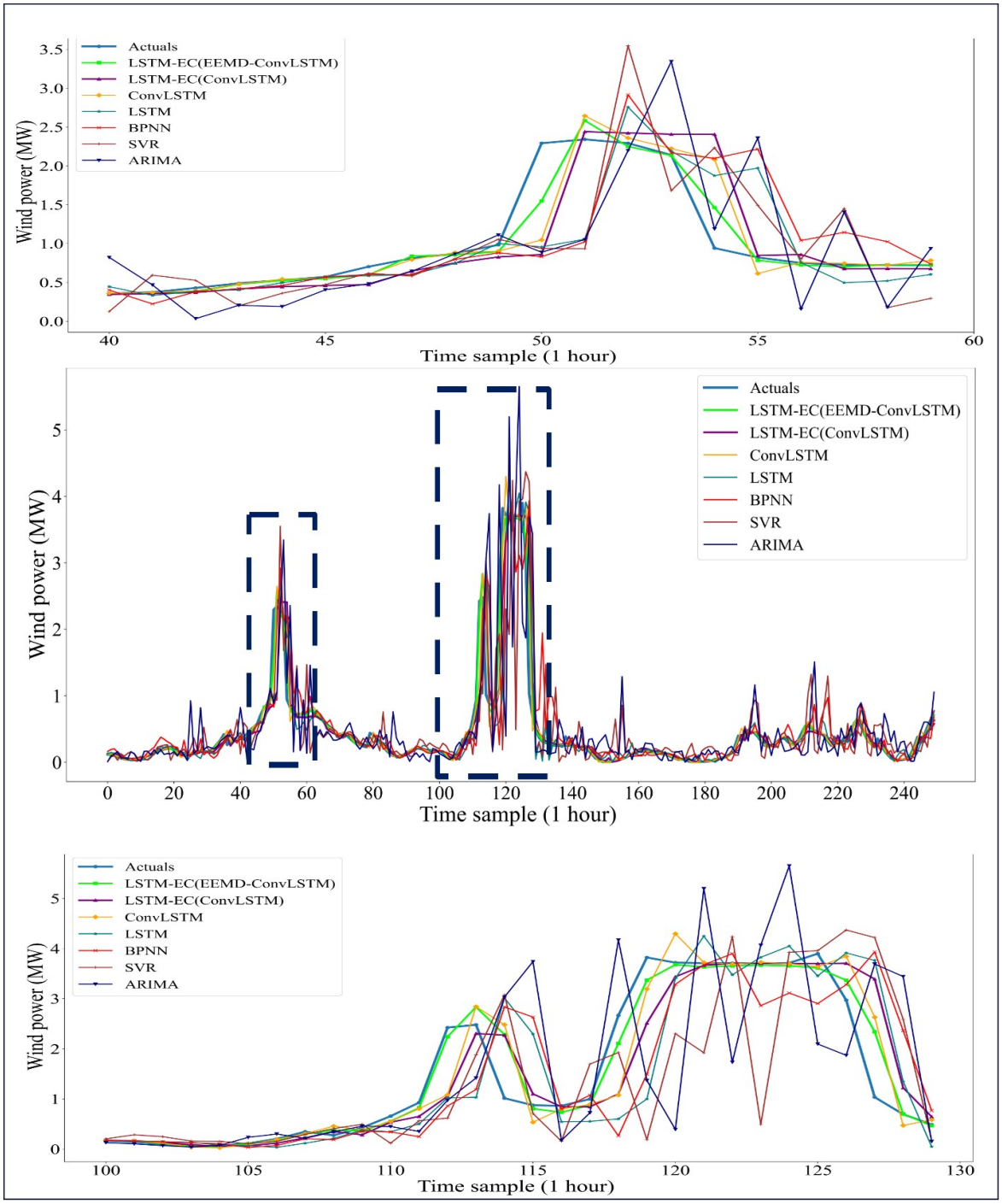



**Figure IV.4:** Forecasted wind power in comparison with benchmark approaches.

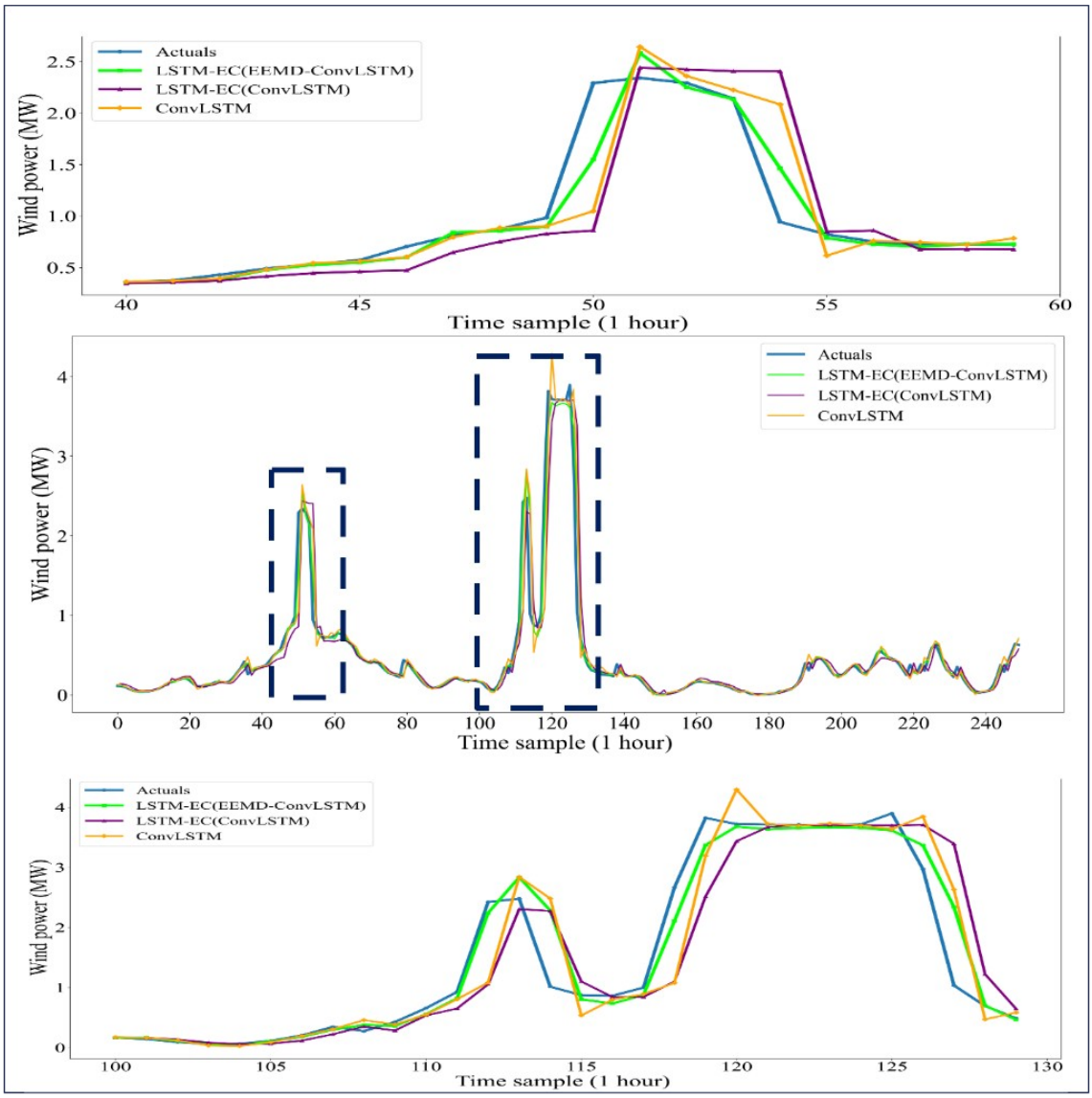



**Figure IV.5:** The proposed model compared with the top-performing models

### IV.2.2 Evaluation against Hybrid Models

To further validate the superiority of the proposed approach, a comparative analysis was performed against a range of advanced hybrid models. The results, presented in Table IV.2 and already illustrated in Figures IV.1 and IV.2, highlight the forecasting performance of models that integrate multiple architectures and techniques. Despite the competitive outcomes of these methods, the proposed EEMD-ConvLSTM framework consistently emerged as the top performer, attaining an MAE of $4.22 \pm 0.31$, RMSE of $9.31 \pm 0.80$, MAPE of 0.19%, and $R^2$ of 0.9811, thereby demonstrating the efficacy of integrating EEMD decomposition with ConvLSTM-based error correction.

Within the set of benchmark models, the CNN-BiGRU hybrid, which combines CNN for spatial feature extraction with Bidirectional GRU to capture temporal dependencies in both directions

demonstrated notably acceptable performance, with an MAE of 4.51 ± 0.38, RMSE of 10.75 ± 0.90, MAPE of 0.22%, and $R^2$ of 0.9768. Nevertheless, the proposed model outperformed these approaches, achieving a 6.4% reduction in MAE and a 13.4% decrease in RMSE. In comparison, the Autoencoder-GRU (AE-GRU) model, which combines an autoencoder for dimensionality reduction with a GRU for temporal sequence modelling, attained an MAE of 4.77 ± 0.42, RMSE of 11.90 ± 0.98, MAPE of 0.26%, and $R^2$ of 0.9724. The EEMD-LSTM model, integrating signal decomposition with a standard LSTM network, reported an MAE of 5.12 ± 0.50, RMSE of 13.42 ± 1.18, MAPE of 0.30%, and $R^2$ of 0.9659 showing clear gains over basic LSTM but remaining inferior to the EEMD-ConvLSTM approach.

Other hybrid strategies showed weaker performance. The CNN-LSTM model, which integrates convolutional layers for feature extraction with LSTM for temporal modelling, achieved an MAE of 5.58 ± 0.54, RMSE of 15.84 ± 1.50, MAPE of 0.35%, and $R^2$ of 0.9553. In contrast, the LSTM–ARIMA hybrid recorded an MAE of 6.35 ± 0.60, RMSE of 19.75 ± 1.80, MAPE of 0.44%, and $R^2$ of 0.9401, highlighting ARIMA limited capacity to model nonlinear dynamics. The statistical significance of the performance differences is confirmed by p-values in Figure IV.3, The error metrics for the models were 0.055 for CNN-BiGRU, 0.044 for AE-GRU, and 0.01 for LSTM-ARIMA, reflecting their relative predictive performance.

**Table IV.2:** Comparative Performance of the Proposed Model and Hybrid Approaches (kW)

| Model | MAE (95% CI) | RMSE (95% CI) | $R^2$ | MAPE | p-value vs. Proposed Model |
|---|---|---|---|---|---|
| LSTM Error-Correction (EEMD-ConvLSTM) | **4.22 ± 0.31** | **9.31 ± 0.80** | **0.9811** | **0.19** | **-** |
| CNN-BiGRU | 4.51 ± 0.38 | 10.75 ± 0.90 | 0.9768 | 0.22 | 0.055 |
| Autoencoder (AE-GRU) | 4.77 ± 0.42 | 11.90 ± 0.98 | 0.9724 | 0.26 | 0.044 |
| EEMD-LSTM | 5.12 ± 0.50 | 13.42 ± 1.18 | 0.9659 | 0.3 | 0.035 |
| CNN-LSTM | 5.58 ± 0.54 | 15.84 ± 1.50 | 0.9553 | 0.35 | 0.025 |
| LSTM-ARIMA | 6.35 ± 0.60 | 19.75 ± 1.80 | 0.9401 | 0.44 | 0.01 |

### IV.2.3 Forecasted Errors signal Analysis

Analyzing forecasted errors is essential for assessing the reliability of predictive models, particularly in wind power forecasting where the data is highly nonlinear and non-stationary. Figure IV.6 presents a comparison of residual errors from the ConvLSTM model and the proposed EEMD-ConvLSTM framework, plotted alongside the actual error values. This

visualization enables a closer evaluation of how well each model captures and corrects outputs deviations.

The residual trajectories highlight oscillations around the zero-error baseline, reflecting the inherent variability of wind power generation. The EEMD-ConvLSTM consistently produces smoother error curves with smaller deviations and fewer sharp spikes, indicating its superior ability to capture the underlying error dynamics and perform effective compensation. Conversely, the ConvLSTM model, while able to follow the general trend, displays larger fluctuations with higher amplitudes, suggesting limitations in capturing high-frequency or short-term error components.

The advantage of the EEMD-ConvLSTM framework arises from the decomposition of the error signal which isolates oscillatory components prior to ConvLSTM learning. This enhances the model's capacity to recognize error patterns across multiple frequency scales and improves correction accuracy. Consequently, the hybrid design achieves a marked reduction in both the magnitude and variance of forecasting errors, resulting in more stable and reliable forecasting.

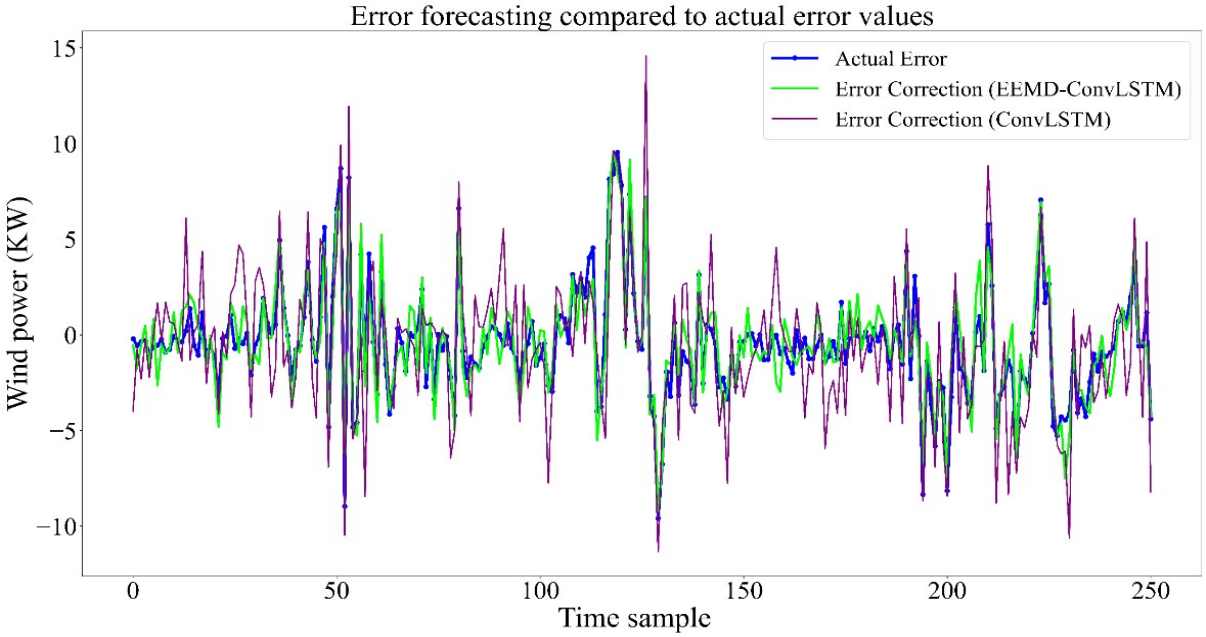



**Figure IV.6:** EEMD-ConvLSTM and ConvLSTM forecasted errors compared to actual

### IV.2.4 Multi-Horizon Forecasting Performance

Forecasting accuracy is strongly influenced by the prediction horizon. To comprehensively assess the robustness and adaptability of the proposed hybrid model, multi-horizon forecasting experiments were performed, spanning one-step to four-step ahead forecasts. The results, summarized in Table IV.3 and illustrated in Figure IV.7, compare the proposed LSTM Error-

Correction (EEMD-ConvLSTM) model with several alternatives, including LSTM-Error Correction (ConvLSTM), ConvLSTM, standard LSTM, BPNN, ARIMA, and SVR. Across all horizons, the EEMD-ConvLSTM consistently achieved the highest forecasting accuracy, demonstrating superior generalization and robustness.

**Table IV.3:** Quantitative comparison of prediction errors (kW) obtained from the EEMD-ConvLSTM model across several forecasting intervals.

| Models | MAE | RMSE | R2 | MAPE% |
|---|---|---|---|---|
| **Two steps** | | | | |
| LSTM-Error Correction (EEMD-CONVLSTM) | **7.605** | **13.66** | **0.9396** | **0.39** |
| LSTM-Error Correction (CONVLSM) | 8.801 | 21.05 | 0.9157 | 0.57 |
| CONVLSTM | 11.323 | 36.12 | 0.8533 | 0.9 |
| LSTM | 12.173 | 42.92 | 0.8337 | 1.14 |
| BPNN | 16.635 | 63.75 | 0.7764 | 1.65 |
| ARIMA | 24.279 | 84.53 | 0.7025 | 2.27 |
| SVR | 50.345 | 105.38 | 0.6807 | 2.98 |
| **Three steps** | | | | |
| LSTM-Error correction (EEMD-CONVLSTM) | **12.252** | **26.703** | **0.8936** | **0.76** |
| LSTM-Error correction (CONVLSM) | 14.745 | 43.648 | 0.8115 | 1.31 |
| CONVLSTM | 19.485 | 66.4159 | 0.7689 | 1.86 |
| LSTM | 20.899 | 77.146 | 0.7233 | 1.91 |
| BPNN | 29.973 | 114.862 | 0.6737 | 3.24 |
| ARIMA | 45.362 | 162.45 | 0.5892 | 4.59 |
| SVR | 97.538 | 209.023 | 0.5211 | 6.54 |
| **Four steps** | | | | |
| LSTM-Error correction (EEMD-CONVLSTM) | **25.505** | **56.815** | **0.7962** | **1.52** |
| LSTM-Error correction (CONVLSM) | 30.249 | 85.95 | 0.7193 | 2.06 |
| CONVLSTM | 40.236 | 130.15 | 0.6981 | 2.87 |
| LSTM | 43.634 | 162.98 | 0.6469 | 3.53 |
| BPNN | 67.716 | 261.31 | 0.5113 | 6.79 |
| ARIMA | 97.773 | 344.351 | 0.4521 | 9.81 |
| SVR | 217.836 | 437.018 | 0.3866 | 14.89 |

- Two-Step Forecasting

When the forecasting horizon is extended to two steps it introduces greater uncertainty, leading to a slight reduction in accuracy across all models. Nevertheless, the proposed EEMD-ConvLSTM maintains a clear advantage, achieving MAE 7.605 kW, RMSE 13.66 kW, $R^2$ 0.9396, with a MAPE of 0.39 %. In comparison, the ConvLSTM-based error correction model (without EEMD) produced a higher MAE of 8.801 kW, while the standalone ConvLSTM recorded 11.323 kW. These findings highlight the added value of EEMD decomposition in improving precision. Conventional approaches like ARIMA and SVR exhibited a pronounced performance drop, highlighting their limitations in multi-step forecasting.

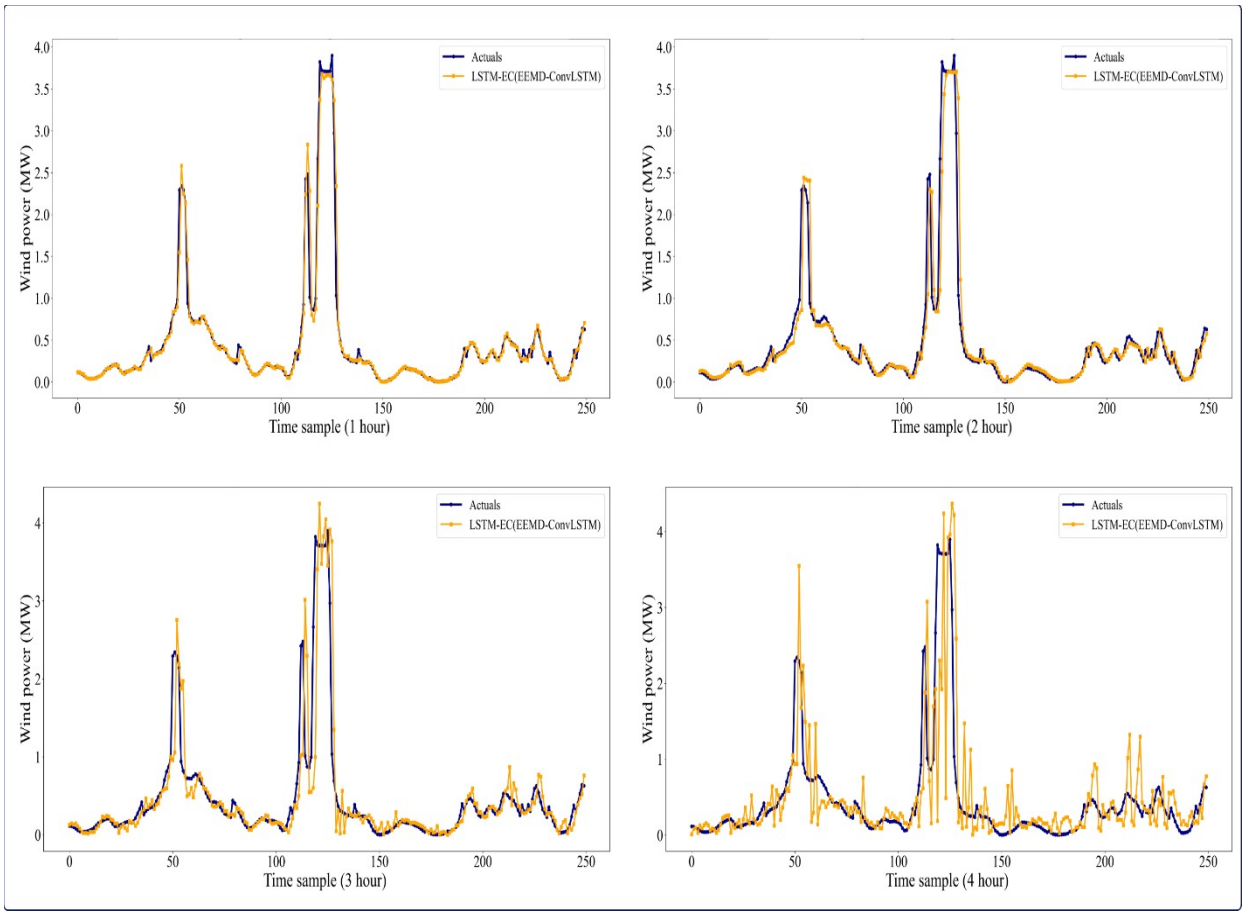



**Figure IV.7:** Performance assessment of the LSTM-based EEMD-ConvLSTM forecasting framework at different time horizons.

➢ Three-Step Forecasting

At three steps ahead, forecasting complexity rises due to error accumulation and the increased volatility of wind power data. Despite this, the EEMD-ConvLSTM remains robust, achieving with an MAE reaching 12.252 kW, RMSE at 26.703 kW, $R^2$ of 0.8936 and a MAPE of 0.76%. Other models exhibit a more pronounced decline: the basic LSTM reached an MAE of 20.899 kW, while ConvLSTM recorded 19.485 kW. Traditional approaches performed substantially worse under these conditions. As shown in Figure IV.7, while the proposed model continues to follow the main trends, deviations appear at local peaks and troughs, an expected effect of extending the prediction horizon. Nonetheless, it consistently outperforms alternatives in both stability and accuracy.

➢ Four-Step Forecasting

The four-step horizon represents the most challenging scenario, as cumulative errors and stochastic wind fluctuations significantly impact performance. Even under these conditions, the EEMD-ConvLSTM remains the leading model, reporting an MAE of 25.505 kW, RMSE of 56.815 kW, $R^2$ of 0.7962, and a MAPE of 1.52%. The ConvLSTM-based error correction model

yielded a higher MAE of 30.249 kW, while the pure ConvLSTM produced 40.236 kW. In contrast, ARIMA and SVR failed to cope with the extended horizon, registering MAEs of 97.773 kW and 217.836 kW respectively, accompanied by sharp drops in $R^2$ values. The forecasts visualized in Figure IV.7 highlight widening discrepancies between predictions and actual observations, particularly at abrupt changes and extrema.

In summary, while forecasting errors naturally increase with longer horizons, the proposed EEMD-ConvLSTM framework demonstrates consistent superiority. Its strength lies in the hybrid structure: the primary LSTM forecast is refined by error modelling through ConvLSTM applied to EEMD-decomposed components. This enables effective handling of both low- and high-frequency dynamics, ensuring reliable generalization from short to medium-term forecasting horizons.

### IV.2.5 Key Contributions and Insights

This study introduces a novel hybrid forecasting framework that integrates Ensemble Empirical Mode Decomposition (EEMD) with ConvLSTM-based error correction to enhance offshore wind power prediction. By leveraging the complementary strengths of signal decomposition and spatiotemporal deep learning, the model provides improved accuracy and robustness compared to conventional approaches. The research findings can be summarized as follows:

The proposed framework applies a robust error correction mechanism, effectively refining initial forecasts and reducing systematic biases.

- ✓ EEMD decomposition successfully captures the nonlinear and non-stationary components of the forecasting error, enabling a more structured representation of error dynamics.
- ✓ The integration of EEMD with ConvLSTM improves the capability to model spatiotemporal dependencies and detect extreme variations, outperforming standard LSTM-based methods.
- ✓ Comprehensive simulations based on offshore wind data recorded (2015-2023) from a Siemens SWT-3.6-120 turbine at the Amrumbank West offshore wind farm confirm the superiority of the proposed model across all evaluation metrics (MAE, RMSE, MAPE, and $R^2$). Significant improvements are observed across different forecasting horizons, highlighting the method's adaptability.

Together, these contributions demonstrate the effectiveness of hybrid deep learning frameworks in tackling the challenges of offshore wind power forecasting and provide a strong foundation for future research in AI-driven renewable energy integration.

## IV.3 Wind Speed Forecasting: Results and Analysis

To rigorously evaluate the performance the CEEMDAN-Bi-GRU-ED-TL forecasting framework, a comprehensive comparative analysis was conducted against a broad range of benchmark approaches. The evaluation included conventional statistical techniques such as SVR, as well as classical machine learning and deep learning models, including RNN, CNN, LSTM, GRU, Bi-LSTM, Bi-GRU, and CNN-LSTM. These baselines were implemented under identical conditions for one-step-ahead forecasting, ensuring a fair and consistent comparison.

In addition to single-step benchmarking, the study extended its analysis to decomposition-based strategies, comparing the proposed framework with leading hybrid approaches incorporating EEMD and CEEMDAN. This step highlighted the added value of signal decomposition in handling the nonlinear and non-stationary nature of wind speed dynamics. A further layer of analysis examined the role of transfer learning (TL) within the CEEMDAN-Bi-GRU-ED framework. By directly contrasting performance with and without TL, the study underscored how leveraging pre-trained residual forecasting models significantly enhances generalization, particularly when applied to Intrinsic Mode Function (IMF) components with limited data.

Finally, to establish robustness and adaptability, the proposed method was evaluated across multiple forecasting horizons. This multi-horizon analysis not only confirmed the framework's predictive accuracy in short-term settings but also demonstrated its stability and resilience as the forecasting window extended to longer time scales. Collectively, these evaluations provide strong evidence of the proposed model's superiority, offering a reliable and advanced solution for wind speed forecasting.

### IV.3.1 Evaluation Against Models Without Decomposition

Figure IV.8 and IV.9 presents the prediction outputs of eight models alongside the actual wind speed measurements. The suite of models considered comprises GRU, LSTM, Bi-GRU, Bi-LSTM, CNN-LSTM, and Bi-GRU-ED demonstrate strong predictive capabilities, with outputs that closely follow the true wind speed profiles. Their predictions are smooth and exhibit minimal deviation from the actual data. In contrast, SVR, CNN, and RNN display higher levels of forecasting error and instability, particularly noticeable in the zoomed sections of the figure. The CNN model struggles to preserve temporal coherence, a limitation of its architecture when applied to sequential data. RNN, as expected, suffers from vanishing and exploding gradient issues, which limit its ability to capture long-range dependencies. Similarly, SVR fails to effectively model the nonlinear patterns in wind speed, resulting in weaker predictions.

Among all models, Bi-GRU-ED stands out with superior forecasting accuracy, as confirmed both visually and statistically. According to Table IV.4 and both Figures IV.8 and IV.9, Bi-GRU-ED achieves, on Dataset 1, an MAE 0.4161, an RMSE 0.6451, a MAPE 10.27%, and an $R^2$ 0.9268. For Dataset 2, it maintains strong performance with an MAE of 0.4426, an RMSE of 0.6653, a MAPE of 11.13%, and an $R^2$ of 0.9278. These results emphasize the model's robustness and adaptability to temporal fluctuations.

In contrast, SVR records the weakest performance. For Dataset 1, it yields an MAE of 0.6573, an RMSE of 0.8107, a MAPE of 18.79%, and an $R^2$ of 0.8681. In Dataset 2, the degradation continues with an MAE of 0.6815, an RMSE of 0.8255, a MAPE of 21.24%, and an $R^2$ of 0.8577. These metrics confirm the model's limitations in capturing the stochastic and nonlinear dynamics of wind speed. The remaining comparative models display intermediate performance. Notably, the bidirectional-based models, particularly Bi-LSTM, Bi-GRU, and Bi-GRU-ED, along with the hybrid CNN-LSTM, consistently outperform their unidirectional counterparts. This performance gain reflects the advantage of processing information in both forward and backward directions, thereby enhancing the ability to capture complex temporal dependencies in wind speed forecasting.

**Table IV.4:** Evaluation of Bi-GRU-ED performance relative to benchmark models without applying decomposition

| | Dataset 1 | | | | Dataset 2 | | | |
|---|---|---|---|---|---|---|---|---|
| Models | MAE | RMSE | MAPE | $R^2$ | MAE | RMSE | MAPE | $R^2$ |
| SVR | 0.6573 | 0.8107 | 18.79 | 0.8681 | 0.6815 | 0.8255 | 21.24 | 0.8577 |
| RNN | 0.5702 | 0.7551 | 14.87 | 0.8704 | 0.5933 | 0.7703 | 15.6 | 0.8771 |
| CNN | 0.5399 | 0.7347 | 12.59 | 0.8906 | 0.5748 | 0.7581 | 14.46 | 0.8731 |
| LSTM | 0.4997 | 0.7069 | 11.25 | 0.9081 | 0.5654 | 0.7523 | 13.68 | 0.8994 |
| GRU | 0.4872 | 0.698 | 10.88 | 0.9126 | 0.5455 | 0.7386 | 12.97 | 0.9094 |
| Bi-LSTM | 0.4579 | 0.6764 | 10.59 | 0.9187 | 0.486 | 0.6972 | 12.53 | 0.9113 |
| Bi-GRU | 0.4469 | 0.6685 | 10.54 | 0.9208 | 0.473 | 0.6877 | 11.98 | 0.9184 |
| CNN-LSTM | 0.4346 | 0.6592 | 10.39 | 0.9293 | 0.4544 | 0.6741 | 11.47 | 0.9206 |
| Bi-GRU-ED | **0.4161** | **0.6451** | **10.27** | **0.9268** | **0.4426** | **0.6653** | **11.13** | **0.9278** |

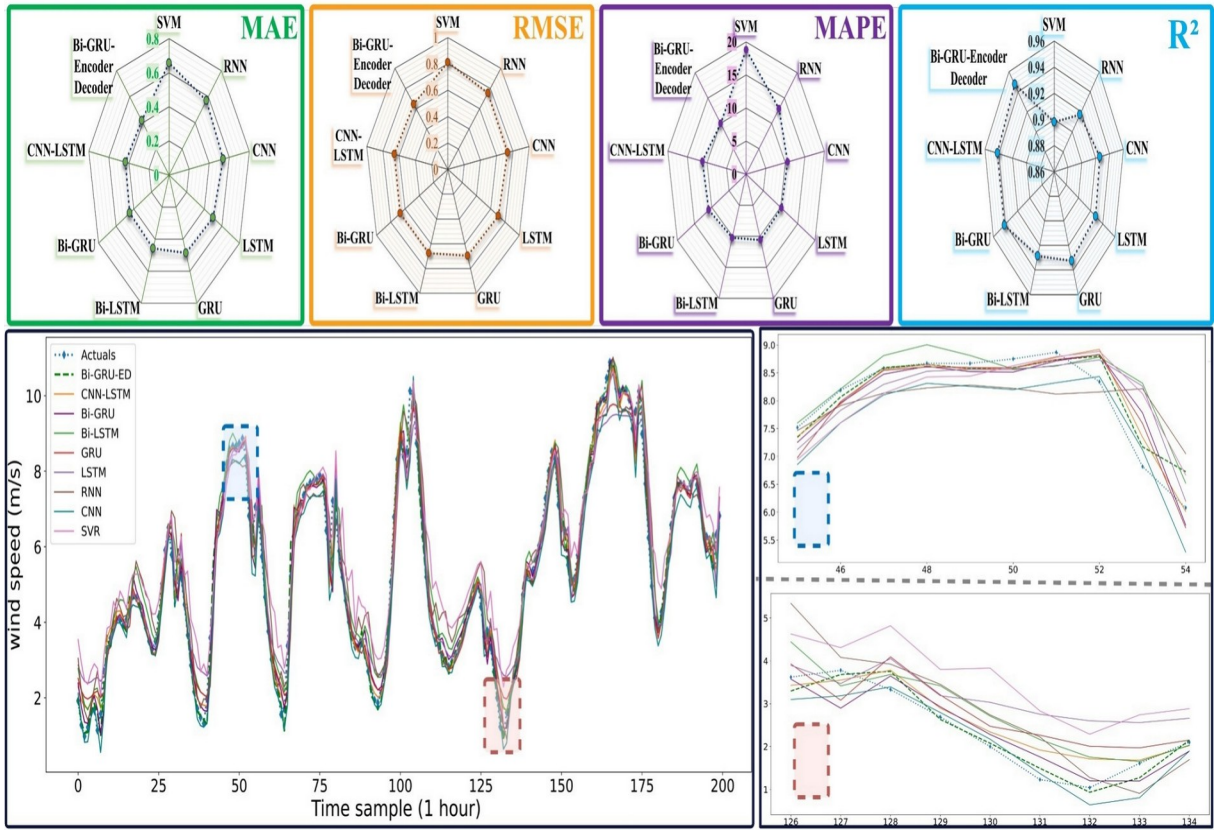



**Figure IV.8:** Comparative performance of Bi-GRU-ED and baseline models on Dataset 1

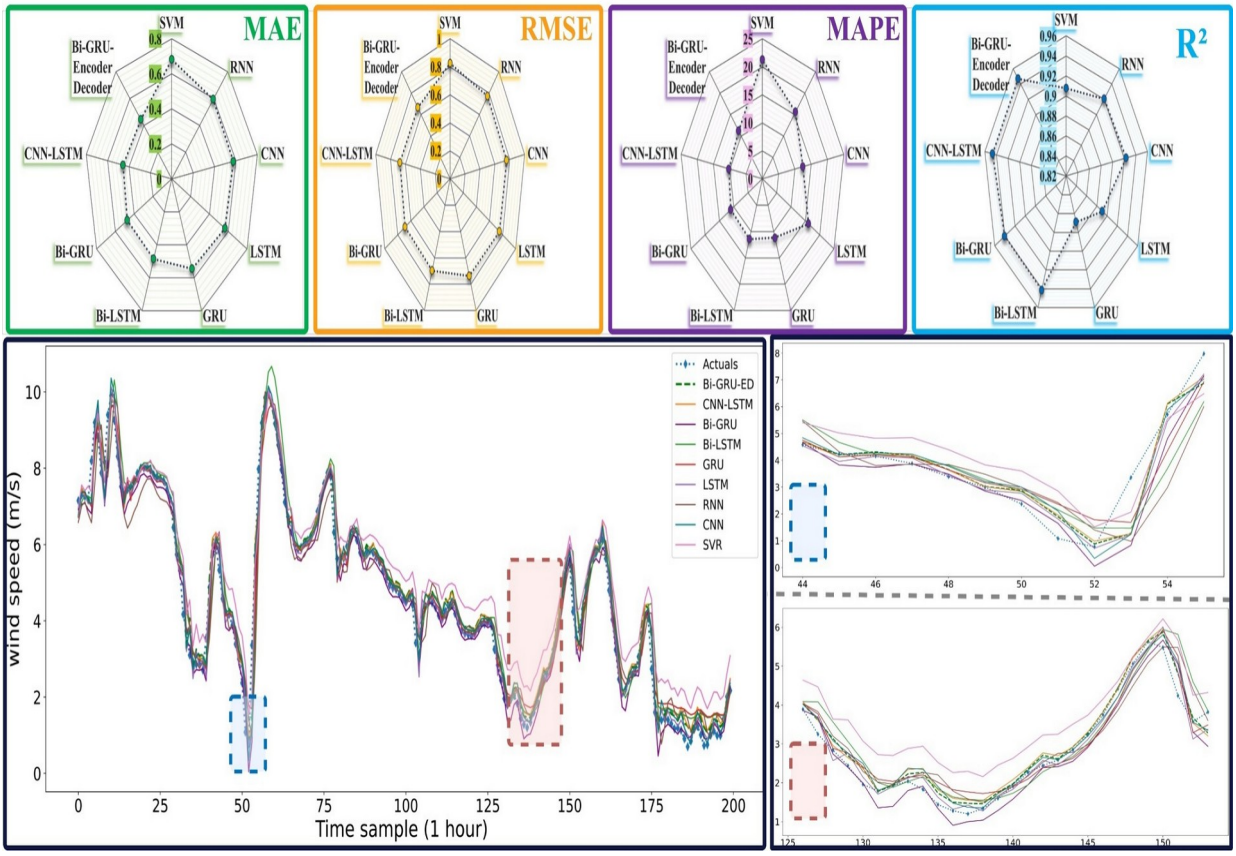



**Figure IV.9:** Wind speed forecasting results without decomposition for dataset 2

### IV.3.2 Models Enhanced with Signal Decomposition

The integration of signal decomposition techniques, particularly EEMD and CEEMDAN, resulted in significant improvements in forecasting performance, as illustrated in Figures IV.10, IV.11, IV.12, IV.13. The proposed framework consistently demonstrated the highest predictive accuracy across both datasets. This is evident in the local zoomed-in sections of the figures, where the predicted curves align closely with the actual wind speed values. Quantitatively, as reported in Table IV.5, the proposed framework achieved the lowest error values, with MAE, RMSE, and MAPE of 0.142, 0.3612, and 2.90 for Dataset 1, and 0.1305, 0.3549, and 2.832 for Dataset 2. In addition, the $R^2$ scores reached 0.9947 and 0.9955, confirming the model's robustness in capturing complex temporal dynamics while minimizing prediction deviations.

The CEEMDAN-Bi-GRU-ED model also delivered strong results, ranking as the second-best performing method. It obtained MAE, RMSE, and MAPE values of 0.1844, 0.4255, and 3.1 for Dataset 1, and 0.1765, 0.4125, and 2.991 for Dataset 2, alongside high $R^2$ values of 0.9896 and 0.9912. These results highlight the contribution of transfer learning in the CEEMDAN-Bi-GRU-ED-TL framework, where pre-trained weights are fine-tuned during IMF-specific learning. This mechanism enhances both learning efficiency and generalization, thereby improving predictive reliability.

An interesting observation emerges when comparing the results of models without decomposition (Table IV.4) to those enhanced with decomposition (Table IV.5). In the absence of decomposition, models generally performed better on Dataset 1 than Dataset 2, likely due to the larger sample size in Dataset 1, which supports stronger generalization. However, after applying decomposition, the relative trend reverses, with Dataset 2 exhibiting superior results. This shift underscores the ability of CEEMDAN to decompose signals into intrinsic mode functions (IMFs), which is particularly advantageous for smaller datasets. By isolating meaningful patterns and filtering noise, the decomposition process enhances training effectiveness and produces more accurate forecasts.

Among all decomposition-enhanced models, the EEMD-CNN-LSTM configuration produced the weakest results, recording the highest MAE, RMSE, and MAPE values: 0.3987, 0.6308, and 10.16 for Dataset 1, and 0.3854, 0.6137, and 9.98 for Dataset 2. Correspondingly, the $R^2$ values were the lowest, at 0.9221 and 0.9395. These outcomes suggest that although EEMD provides improvements compared to models without decomposition, CEEMDAN generally delivers superior performance. This superiority stems from CEEMDAN ability to more capture the nonlinear and non-stationary characteristics of data, correctly distinguishing signal components

from noise. Moreover, bidirectional structures, such as EEMD-Bi-GRU and CEEMDAN-Bi-GRU-ED, consistently outperform unidirectional configurations, confirming that leveraging both past and future temporal dependencies enhance predictive accuracy in wind speed forecasting.

**Table IV.5:** Comparative Forecasting Performance of the Proposed Model Against EEMD- and CEEMDAN-Based Approaches.

| | Dataset 1 | | | | Dataset 2 | | | |
|---|---|---|---|---|---|---|---|---|
| Models | MAE | RMSE | MAPE | $R^2$ | MAE | RMSE | MAPE | $R^2$ |
| EEMD-CNN-LSTM | 0. 3987 | 0.6308 | 10.16 | 0.9221 | 0.3854 | 0.6137 | 9.98 | 0.9395 |
| EEMD-GRU | 0.3328 | 0.5954 | 8.44 | 0.9481 | 0.3494 | 0.5824 | 8.78 | 0.9459 |
| EEMD-Bi-GRU | 0.3254 | 0.5526 | 7.64 | 0.958 | 0.3201 | 0.5363 | 7.823 | 0.9587 |
| EEMD-Bi-GRU-ED | 0.2873 | 0.536 | 7.17 | 0.9627 | 0.2741 | 0.5123 | 7.208 | 0.9698 |
| CEEMDAN-CNN-LSTM | 0.3281 | 0.5902 | 7.43 | 0.9529 | 0.3041 | 0.5423 | 7.122 | 0.9607 |
| CEEMDAN-GRU | 0.2581 | 0.5787 | 5.88 | 0.9709 | 0.2311 | 0.4806 | 4.598 | 0.9792 |
| CEEMDAN-Bi-GRU | 0.2202 | 0.4653 | 4.25 | 0.9852 | 0.2 | 0.4472 | 4.001 | 0.9894 |
| CEEMDAN-Bi-GRU-ED | **0.1844** | **0.4255** | **3.11** | **0.9896** | **0.1765** | **0.4125** | **2.991** | **0.9912** |

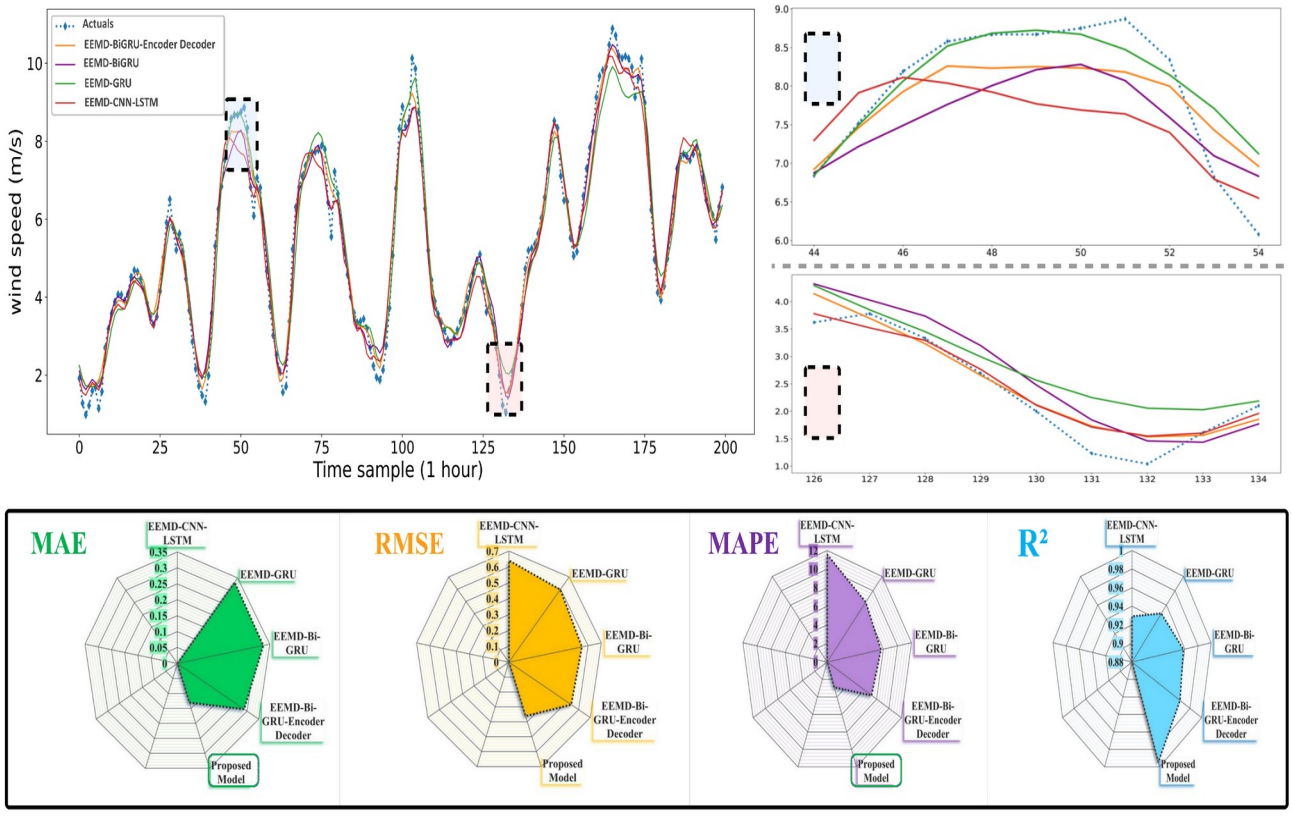



**Figure IV.10:** EEMD-Based Wind Speed Prediction Performance for Dataset 1

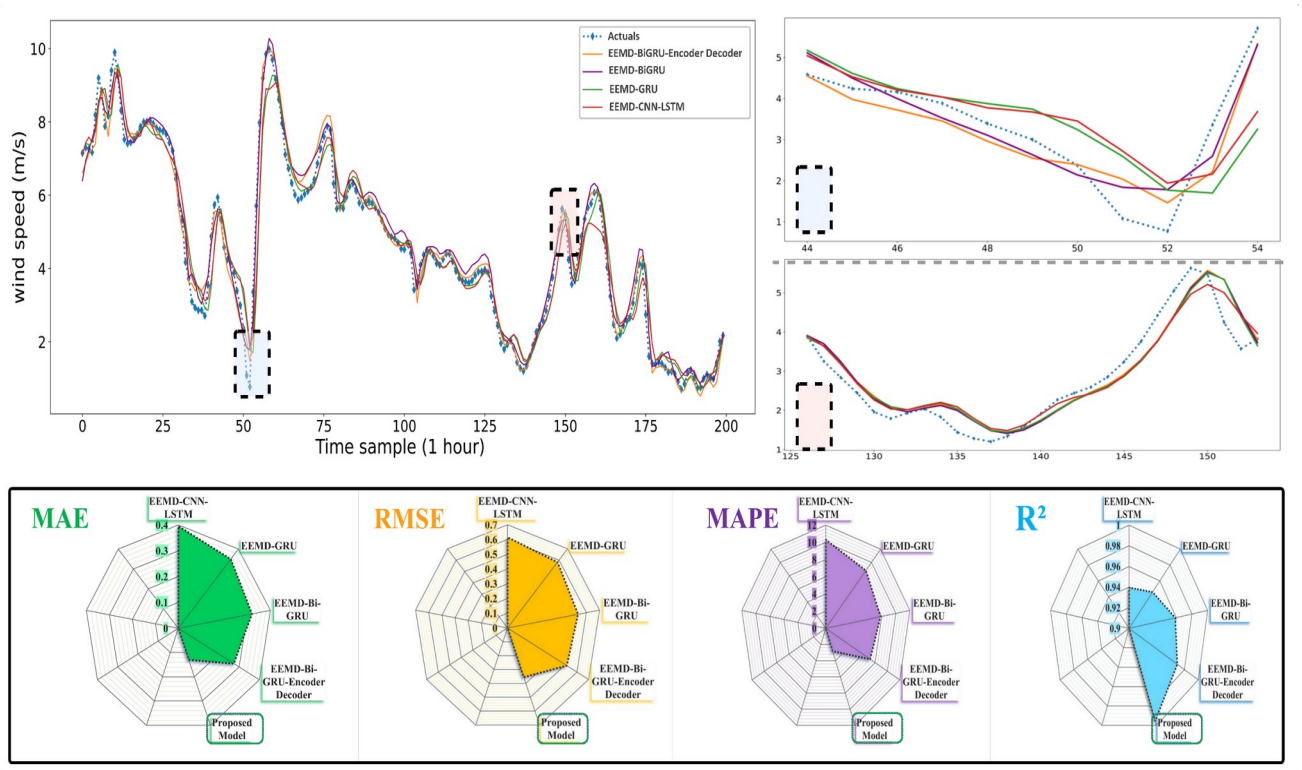



**Figure IV.11:** EEMD-Based Wind Speed Forecasting Performance for Dataset 2

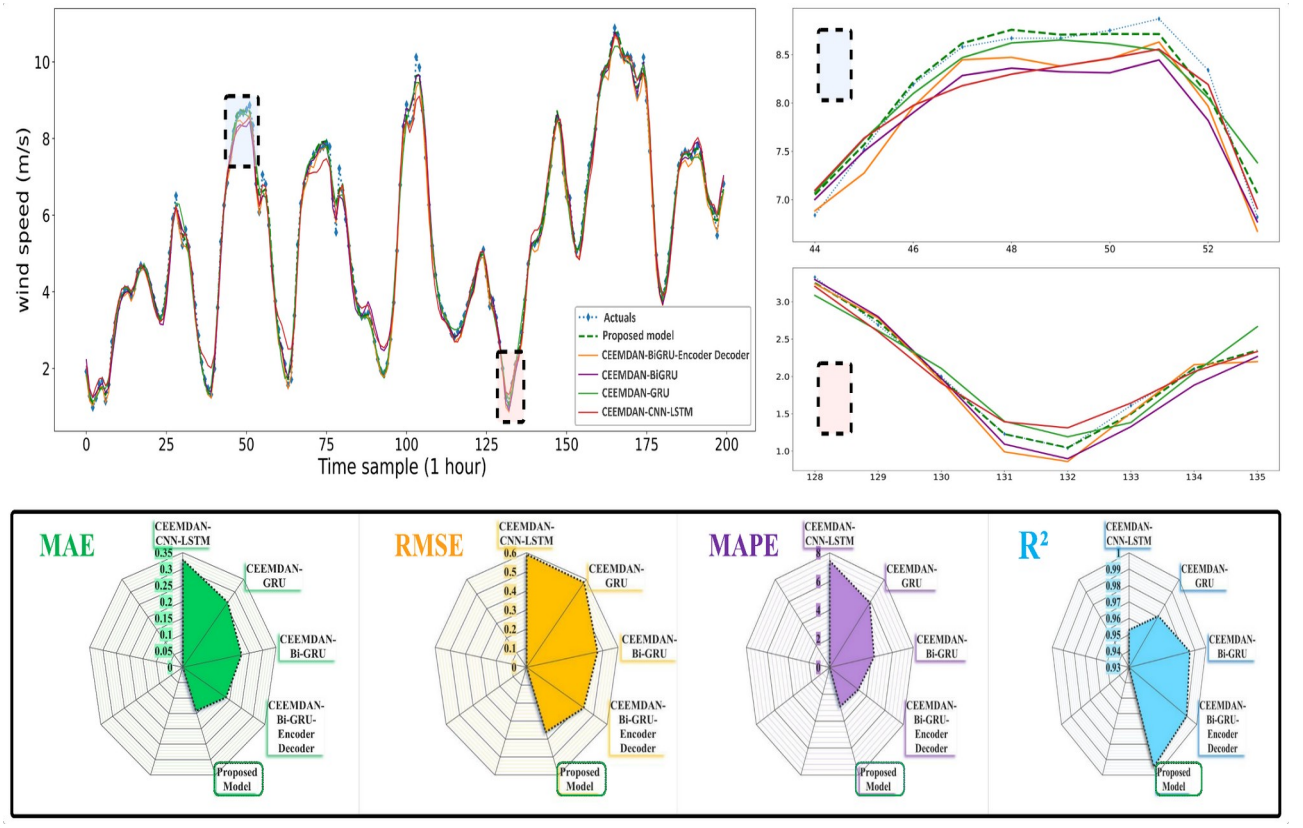



**Figure IV.12:** Forecasting outcomes with CEEMDAN-based decomposition for Dataset 1

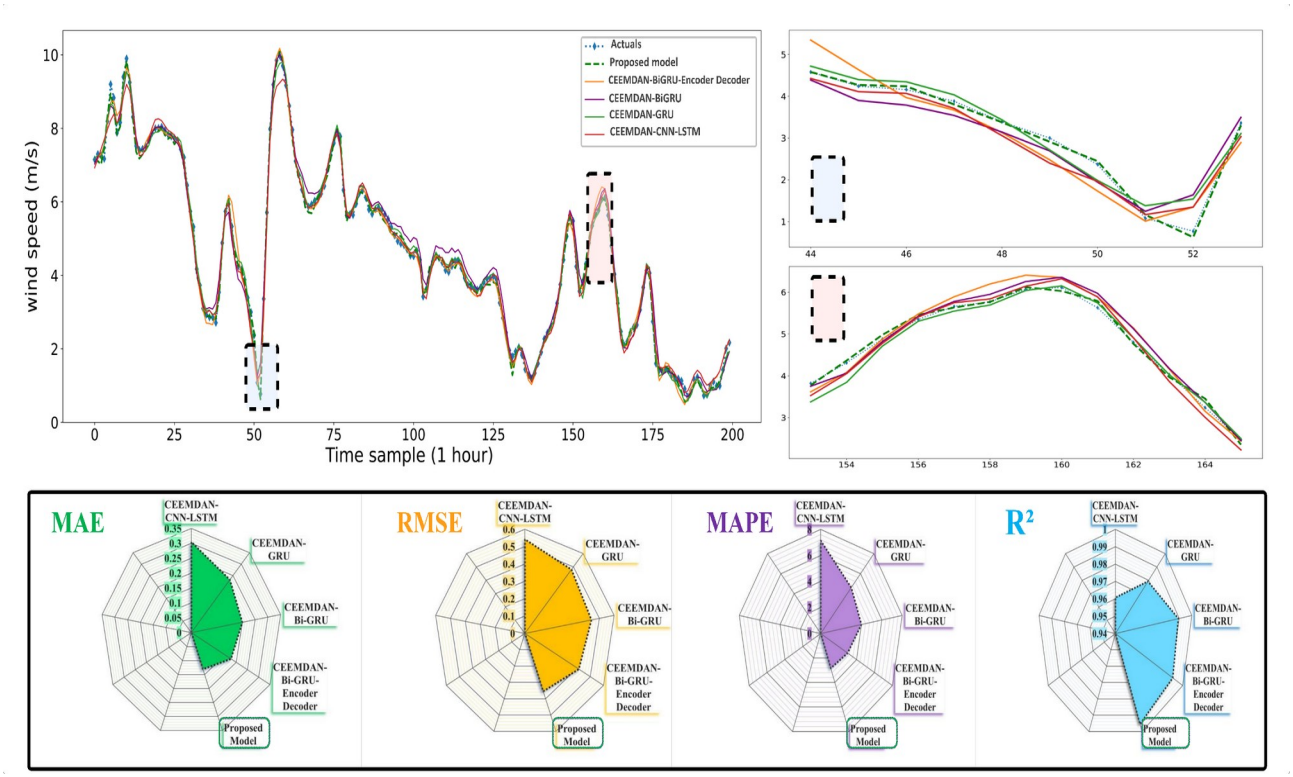



**Figure IV.13:** Forecasting outcomes with CEEMDAN-based decomposition for Dataset 2

### IV.3.3 Impact of Transfer Learning on Decomposed Forecasting

The role of transfer learning (TL) in enhancing decomposed forecasting performance was systematically evaluated, with results summarized in Table IV.6. Across both datasets, the integration of TL consistently improved predictive accuracy in most IMFs, particularly through noticeable reductions in MAE and RMSE. This demonstrates that transfer learning enhances learning generalization and stability within the CEEMDAN-Bi-GRU-ED framework.

For Dataset 1, notable enhancements were observed in IMF2 and IMF3, where the MAE decreased markedly from 0.1633 to 0.0453 for IMF2, and from 0.0199 to 0.0131 for IMF3. Similar improvements were observed in Dataset 2, where IMF2 achieved a reduction in MAE from 0.1578 to 0.0414, alongside a decline in RMSE from 0.3972 to 0.2028. These results highlight the robustness of TL in modelling low-frequency components of wind speed signals, which typically exhibit smoother patterns and less stochastic variability.

However, the impact of TL on high-frequency components is less consistent. For IMF1 in Dataset 1, performance slightly degraded, with MAE rising from 0.1021 to 0.1211 and RMSE from 0.320 to 0.348. By contrast, in Dataset 2, IMF1 demonstrated improved MAE performance (0.1224 to 0.0802), although accompanied by a minor increase in RMSE. These mixed outcomes suggest that while TL is highly beneficial for low and mid-frequency IMFs, its

advantage diminishes in high-frequency, noise-sensitive components, where learning transfer is less effective.

**Table IV.6:** Transfer Learning-Driven Accuracy Improvements Across IMFs and Residual Components

| Component | Dataset 1 | | | | Dataset 2 | | | |
|---|---|---|---|---|---|---|---|---|
| | CEEMDAN-Bi-GRU-ED without transfer learning | | CEEMDAN-Bi-GRU-ED with transfer learning (Proposed model) | | CEEMDAN-Bi-GRU-ED without transfer learning | | CEEMDAN-Bi-GRU-ED with transfer learning (Proposed model) | |
| | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE |
| Residual | 0.0828 | 0.1665 | 0.0828 | 0.1665 | 0.0493 | 0.1403 | 0.0493 | 0.1403 |
| IMF1 | 0.1021 | 0.32 | 0.1211 | 0.348 | 0.1224 | 0.2827 | **0.0802** | 0.3001 |
| IMF2 | 0.1633 | 0.4041 | **0.0453** | **0.2129** | 0.1578 | 0.3972 | **0.0414** | **0.2028** |
| IMF3 | 0.0199 | 0.1412 | **0.0131** | **0.1386** | 0.0192 | 0.1292 | **0.0167** | **0.1246** |
| IMF4 | 0.0304 | 0.1678 | **0.0281** | **0.1005** | 0.187 | 0.1358 | **0.0201** | **0.0856** |
| IMF5 | 0.0098 | 0.082 | **0.0067** | **0.0776** | 0.014 | 0.1184 | **0.0127** | **0.0931** |
| IMF6 | 0.0094 | 0.0912 | **0.0087** | **0.0892** | 0.0098 | 0.1214 | **0.0086** | **0.0902** |

### IV.3.4 Multi-Timestep Forecasting Comparison

To further evaluate the proposed forecasting framework, its performance was examined across multiple prediction horizons using three distinct configurations: the baseline Bi-GRU-ED model, the CEEMDAN-Bi-GRU-ED variant, and the CEEMDAN-Bi-GRU-ED architecture enhanced with TL. The comparative outcomes, illustrated in Figure IV.14 and summarized in Table IV.7, clearly demonstrate the advantages of the proposed approach.

At the one-hour forecasting, the CEEMDAN-Bi-GRU-ED-TL consistently outperforms all other configurations, achieving the lowest error values and highest determination coefficients. Extending the forecasting horizon to two hours results in a slight rise in the error metrics for all models a natural consequence of heightened temporal uncertainty. Nevertheless, the proposed framework maintains a clear edge. For Dataset 1, it delivers an MAE of 0.1822, RMSE of 0.4049, MAPE of 3.808%, and $R^2$ of 0.9894. On Dataset 2, performance remains comparably strong, with an MAE of 0.1658, RMSE of 0.3964, MAPE of 3.658%, and $R^2$ of 0.9915, highlighting the model's ability to sustain precision under extended horizons.

With a three-hour-ahead horizon, accuracy degradation becomes more evident, particularly in the CEEMDAN-Bi-GRU-ED model without transfer learning. This indicates an increased sensitivity to long-term uncertainty and error accumulation. Despite this, the CEEMDAN-Bi-GRU-ED-TL continues to outperform alternative frameworks, underscoring the stabilizing

impact of transfer learning in capturing temporal dependencies across multiple frequency bands.

Although forecasting errors naturally escalate as the prediction interval increases due to the stochastic wind speed nature, the hybrid integration of CEEMDAN with transfer learning substantially mitigates this decline. The decomposition stage facilitates more effective representation of low- and high-frequency components, while transfer learning enhances the reuse of prior knowledge, thereby improving generalization. Collectively, these mechanisms enable the proposed framework to balance short-term accuracy with medium-term robustness, establishing it as a reliable solution for multi-horizon wind speed forecasting [102].

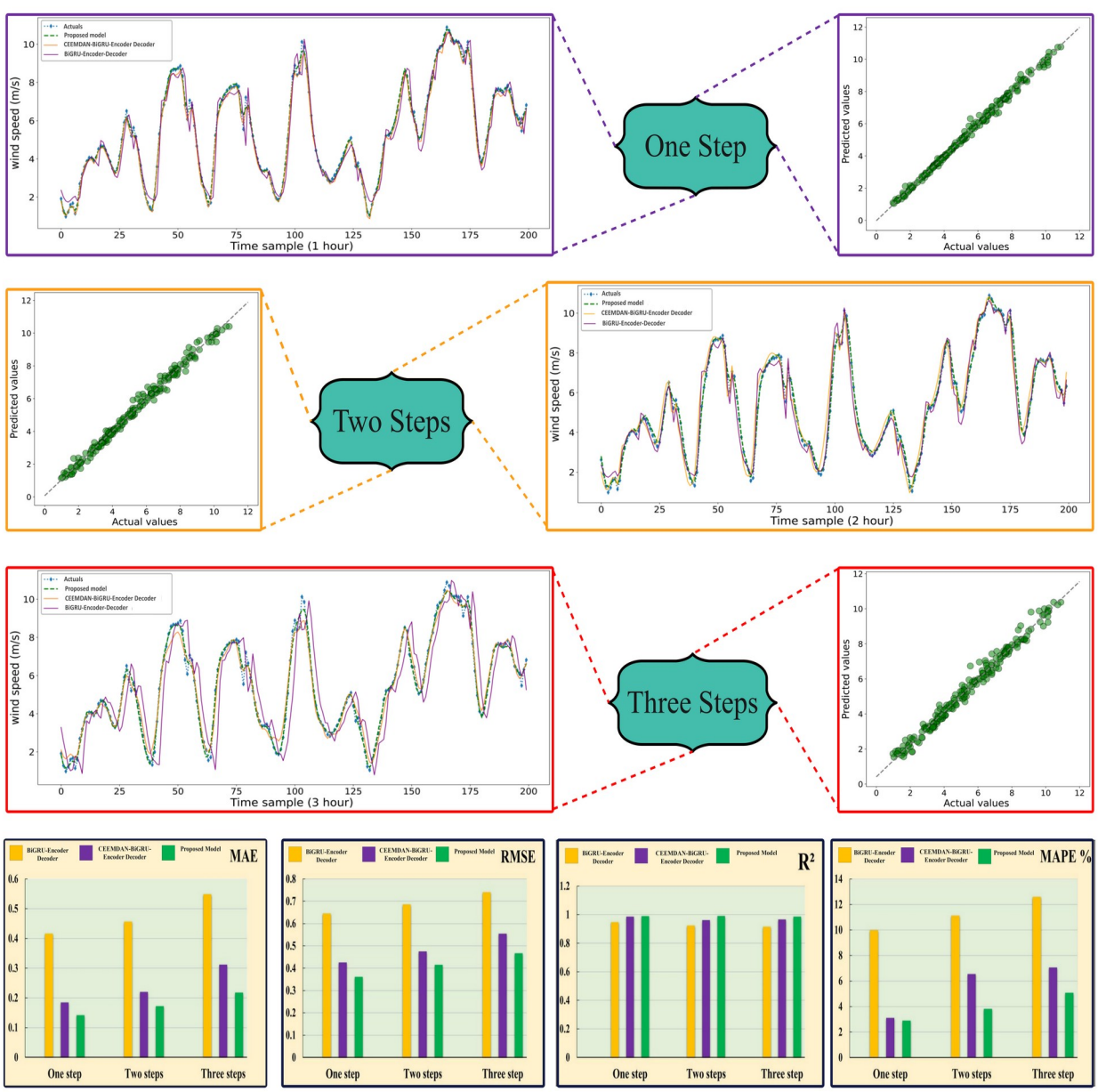



**Figure IV.14:** Comparative analysis of forecasts across multiple prediction horizons

**Table IV.7:** Forecasting accuracy across different forecasting horizons: Proposed Model vs. Models Without CEEMDAN or Transfer Learning

| | | Dataset 1 | | | | Dataset 2 | | | |
|---|---|---|---|---|---|---|---|---|---|
| steps | models | MAE | RMSE | MAPE | $R^2$ | MAE | RMSE | MAPE | $R^2$ |
| One step (t+1) | Bi-GRU-ED | 0.4161 | 0.6451 | 10.27 | 0.9268 | 0.4426 | 0.6653 | 11.13 | 0.9278 |
| | CEEMDAN-Bi-GRU-ED | 0.1844 | 0.4255 | 3.1 | 0.9896 | 0.1765 | 0.4125 | 2.991 | 0.9912 |
| | Proposed model | **0.142** | **0.3612** | **2.9** | **0.9947** | **0.1305** | **0.3549** | **2.832** | **0.9955** |
| Two steps (t+2) | Bi-GRU-ED | 0.4666 | 0.6857 | 10.65 | 0.9122 | 0.4802 | 0.6897 | 11.95 | 0.9068 |
| | CEEMDAN-Bi-GRU-ED | 0.2281 | 0.4749 | 5.15 | 0.9713 | 0.2201 | 0.4671 | 4.369 | 0.9848 |
| | Proposed model | **0.1822** | **0.4049** | **3.808** | **0.9894** | **0.1658** | **0.3964** | **3.658** | **0.9915** |
| Three steps (t+3) | Bi-GRU-ED | 0.5587 | 0.7408 | 15.01 | 0.8853 | 0.5402 | 0.7417 | 13.6 | 0.8756 |
| | CEEMDAN-Bi-GRU-ED | 0.312 | 0.5546 | 8.58 | 0.9606 | 0.2945 | 0.5422 | 6.47 | 0.9619 |
| | Proposed model | **0.2478** | **0.4767** | **6.678** | **0.9712** | **0.2195** | **0.4572** | **5.09** | **0.9803** |

### IV.3.5 Key Contributions and Insights

This research presents a sophisticated hybrid forecasting framework, CEEMDAN-Bi-GRU-ED-TL, which combines multi-scale signal decomposition, bidirectional recurrent deep learning models, and transfer learning to effectively tackle the complexities of short-term wind speed forecasting. By combining decomposition, temporal sequence modelling, and knowledge reuse across decomposed components, the framework establishes a new benchmark in wind speed forecasting. The main contributions of this research can be summarized as follows.

- ✓ The proposed framework applies a unified hybrid architecture that couples CEEMDAN decomposition with a Bi-GRU-based encoder-decoder, enabling effective modelling of nonlinear, non-stationary, and multi-frequency wind speed dynamics.
- ✓ By incorporating transfer learning, the framework effectively leverages and adapts a pre-trained residual forecasting model for predicting all individual IMFs, eliminating the need for independent training of each component. This significantly reduces computational complexity while enhancing learning generalization.
- ✓ Empirical validation on two real-world datasets confirms substantial performance gains. Compared with non-decomposed baselines, the proposed model achieves up to 65.9% reduction in MAE and 43.9% reduction in RMSE on Dataset 1, with similar improvements observed on Dataset 2. These results highlight its superior ability to minimize forecasting errors.

- ✓ The combination of CEEMDAN, Bi-GRU-ED, and transfer learning leads to improved robustness across multiple prediction horizons. Even under longer forecasting intervals, the model demonstrates high fidelity in capturing temporal patterns, maintaining a balance between short-term accuracy and medium-term stability.
- ✓ This work is among the first to integrate signal decomposition, bidirectional recurrent encoder-decoder structures, and transfer learning into a single forecasting pipeline. The demonstrated effectiveness Highlights its promise as a robust, scalable, and widely applicable approach for wind speed forecasting in renewable energy contexts.

Together, these contributions establish the CEEMDAN-Bi-GRU-ED-TL framework as a powerful tool for addressing the complexities of short-term wind speed forecasting, offering both methodological innovation and practical value for the integration of renewable energy into modern power systems.

## IV.4 Conclusion

This chapter has presented a rigorous comparative analysis and performance evaluation of two advanced hybrid deep learning frameworks designed for wind energy forecasting, each targeting distinct but complementary aspects of the renewable energy domain: offshore wind power prediction and short-term wind speed forecasting. Both studies sought to overcome the intrinsic challenges of renewable energy data namely non-linearity, non-stationarity, stochastic variability, and temporal-spatial complexity by combining signal decomposition, advanced deep neural networks as hybrid learning techniques.

For wind power forecasting, a novel hybrid model was developed that combines Ensemble Empirical Mode Decomposition (EEMD) with a ConvLSTM-based error correction mechanism. By refining LSTM forecasts through systematic error modelling, this framework successfully captured complex temporal dependencies within the error signal. Empirical testing on real-world data from the Amrumbank West offshore wind farm (2015–2023) demonstrated substantial accuracy gains over conventional deep learning and hybrid baselines. The integration of EEMD enabled multi-scale pattern extraction, while ConvLSTM contributed to robust spatiotemporal representation learning, leading to consistent improvements in MAE, RMSE, MAPE, and $R^2$ across varying forecast horizons.

In parallel, for short-term wind speed forecasting, an end-to-end architecture denoted CEEMDAN-Bi-GRU-ED-TL was introduced. This framework strategically integrates Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) for multi-scale signal decomposition, a Bidirectional GRU Encoder-Decoder for modelling

sequential data, along with a transfer learning strategy to efficiently reuse and fine-tune forecasting modules across decomposed components. Validation on two independent datasets confirmed the superiority of this approach, with the proposed model achieving state-of-the-art accuracy. Notably, transfer learning significantly reduced training overhead and enhanced model generalization, while CEEMDAN decomposition improved robustness against noise and non-stationarity. The framework demonstrated consistent performance across multiple forecasting horizons, underscoring its adaptability and scalability.

Taken together, the findings from these two complementary studies establish a clear demonstration of the value of hybridization in wind energy forecasting. By combining decomposition techniques with advanced deep learning architectures and augmenting them with intelligent learning strategies such as error correction and transfer learning, the proposed frameworks effectively address the major bottlenecks of wind energy forecasting data complexity, temporal variability, and computational efficiency.

Ultimately, this chapter underscores the transformative potential of AI-driven hybrid forecasting models. These contributions not only advance the scientific state-of-the-art but also provide a solid foundation for the integration of reliable forecasting tools into modern energy systems, thereby supporting more stable and efficient renewable energy integration.

# General Conclusion

This thesis contributes to addressing the critical challenge of wind energy forecasting by developing advanced artificial intelligence-based hybrid models. Given that the inherent non-linearity, non-stationarity, and stochastic nature of wind make accurate forecasting highly complex, this research integrates signal decomposition techniques with deep neural networks and innovative learning strategies. The resulting novel frameworks, validated through rigorous experimental analysis, enhance both the accuracy and reliability of predictions, offering significant methodological and computational contributions.

A primary contribution of this research is a hybrid model for offshore wind power forecasting that integrates Ensemble Empirical Mode Decomposition (EEMD) with a ConvLSTM-based error correction mechanism. This framework leverages decomposition to isolate complex oscillatory patterns within forecasting errors, while the ConvLSTM architecture captures the spatiotemporal dependencies often missed by conventional LSTM forecasters. Evaluated on data from the Amrumbank West wind farm, the model demonstrated superior performance against several benchmark deep learning and hybrid architectures. It consistently achieved lower error metrics (MAE, RMSE, MAPE) and higher $R^2$ values across multiple forecasting horizons, validating the efficacy of combining error decomposition with spatiotemporal memory networks to mitigate systematic biases and enhance forecast stability.

The second contribution introduced a novel hybrid pipeline for short-term wind speed forecasting, the CEEMDAN-Bi-GRU-ED-TL framework. This approach integrates CEEMDAN for multi-scale signal decomposition, a bidirectional GRU encoder-decoder to capture temporal relationships, and a homogeneous transfer learning scheme that applies pre-trained models across all decomposed components. Unlike traditional decomposition methods that train separate models for each component, this framework lowers computational costs while improving prediction accuracy. Extensive experiments conducted and confirmed its robustness, achieving MAE as low as 0.142 and $R^2$ values above 0.9947, clearly outperforming non-decomposed baselines and alternative hybrid models. These findings underline the value of integrating decomposition and transfer learning within a unified deep learning framework.

Together, these frameworks demonstrate that the strategic integration of hybridization. By effectively addressing fundamental challenges such as data non-stationarity, volatility,

complexity, and uncertainty, the proposed models establish a robust framework for developing more accurate, reliable, and stable forecasting systems.

Beyond its direct contributions, this thesis outlines several promising avenues for future research. A critical next step is the real-time implementation of the proposed models, dynamic grid conditions. Furthermore, embedding these advanced forecasting models into microgrid energy management systems could significantly improve operational efficiency by optimizing scheduling, load balancing, and storage utilization. Finally, extending the hybrid AI methodology to multi-source renewable forecasting, jointly predicting wind, solar, and other generation, presents a significant opportunity to enhance forecasting reliability and stability within increasingly complex and integrated energy networks.

Ultimately, this thesis has contributed to the scientific understanding and algorithmic development of AI-driven hybrid forecasting techniques for wind energy. Through the development and rigorous experimental validation of novel hybrid models, it lays a comprehensive foundation for future on-site implementations, directly supporting the advancement of smarter, more reliable, and sustainable power systems.

**References**


[1] B. Elshafei *et al.*, "A hybrid solution for offshore wind resource assessment from limited onshore measurements," *Appl. Energy*, vol. 298, no. May, p. 117245, 2021, doi: 10.1016/j.apenergy.2021.117245.

[2] J. Li and J. Wang, "Forcasting of energy futures market and synchronization based on stochastic gated recurrent unit model," *Energy*, vol. 213, p. 118787, 2020, doi: 10.1016/j.energy.2020.118787.

[3] N. Effenberger and N. Ludwig, "A collection and categorization of open-source wind and wind power datasets," *Wind Energy*, vol. 25, no. 10, pp. 1659–1683, 2022, doi: 10.1002/we.2766.

[4] H. Cai, Z. Wu, C. Huang, and D. Huang, "Wind Power Forecasting Based on Ensemble Empirical Mode Decomposition with Generalized Regression Neural Network Based on Cross-Validated Method," *J. Electr. Eng. Technol.*, vol. 14, no. 5, pp. 1823–1829, 2019, doi: 10.1007/s42835-019-00186-x.

[5] A. Schaffarczyk, "Aerodynamics and Blade Design," in *Understanding Wind Power Technology*, 2014, pp. 126–161. doi: https://doi.org/10.1002/9781118701492.ch4.

[6] P. Principles and W. E. Conversion, *Wind Turbines*, vol. V. Springer Cham, 2025. doi: 10.1007/978-3-031-87795-7.

[7] Y. Celik, "Of Mechanical Engineering . Aerodynamics and Self-Starting of Vertical Axis Wind Turbines with J-Shaped Aerofoils," The University of Sheffield, 2021. [Online]. Available: etheses.whiterose.ac.uk/id/eprint/29030/1/Yunus Celik-PhD Thesis.pdf

[8] D. Rekioua, *Wind Power Electric Systems: Modeling, Simulation, Control and Power Management Control*, vol. Part F2493. 2024. doi: 10.1007/978-3-031-52883-5.

[9] "Global Wind Report 2022 - Global Wind Energy Council." https://gwec.net/global-wind-report-2022/ (accessed Apr. 09, 2023).

[10] B. K. Saxena, S. Mishra, and K. V. S. Rao, "Offshore wind speed forecasting at different heights by using ensemble empirical mode decomposition and deep learning

models," *Appl. Ocean Res.*, vol. 117, no. November, p. 102937, 2021, doi: 10.1016/j.apor.2021.102937.

[11] J. Kiviluoma, *Managing wind power variability and uncertainty through increased power system flexibility*, no. June. 2013. [Online]. Available: http://www.vtt.fi/inf/pdf/science/2013/S35.pdf

[12] L. Melalkia, F. Berrezzek, Khaled khelil, A. Saim, and R. Nebili, "A hybrid error correction method based on EEMD and ConvLSTM for offshore wind power forecasting," *Ocean Eng.*, vol. 325, no. March, p. 120773, 2025, doi: 10.1016/j.oceaneng.2025.120773.

[13] F. Berrezzek, K. Khelil, and T. Bouadjila, "Efficient wind speed forecasting using discrete wavelet transform and artificial neural networks," *Rev. d'Intelligence Artif.*, vol. 33, no. 6, pp. 447–452, 2019, doi: 10.18280/ria.330607.

[14] E. Arslan Tuncar, Ş. Sağlam, and B. Oral, "A review of short-term wind power generation forecasting methods in recent technological trends," *Energy Reports*, vol. 12, no. November 2023, pp. 197–209, 2024, doi: 10.1016/j.egyr.2024.06.006.

[15] X. Huang, Y. Zhang, J. Liu, X. Zhang, and S. Liu, "A Short-Term Wind Power Forecasting Model Based on 3D Convolutional Neural Network–Gated Recurrent Unit," *Sustain.*, vol. 15, no. 19, 2023, doi: 10.3390/su151914171.

[16] N. Sihag and K. S. Sangwan, *Development of an electric-load intelligence system for component level disaggregation to improve energy efficiency of machine tools*. 2020. doi: 10.1007/978-3-030-44248-4_12.

[17] E. Erdem, J. Shi, and Y. She, "Comparison of two ARMA-GARCH approaches for forecasting the mean and volatility of wind speed," *Springer Proc. Phys.*, vol. 155, pp. 65–73, 2014, doi: 10.1007/978-3-319-05521-3_9.

[18] A. H. Bukhari, M. A. Z. Raja, M. Sulaiman, S. Islam, M. Shoaib, and P. Kumam, "Fractional neuro-sequential ARFIMA-LSTM for financial market forecasting," *IEEE Access*, vol. 8, pp. 71326–71338, 2020, doi: 10.1109/ACCESS.2020.2985763.

[19] F. Harrou, A. Dairi, A. Dorbane, and Y. Sun, "Enhancing wind power prediction with self-attentive variational autoencoders: A comparative study," *Results Eng.*, vol. 23, no. June, p. 102504, 2024, doi: 10.1016/j.rineng.2024.102504.

[20] F. Houndekindo and T. B. M. J. Ouarda, “LSTM and Transformer-based framework for bias correction of ERA5 hourly wind speeds,” *Energy*, vol. 328, no. April, p. 136498, 2025, doi: 10.1016/j.energy.2025.136498.

[21] S. Tasnim, A. Rahman, A. M. T. Oo, and M. E. Haque, “Wind power prediction in new stations based on knowledge of existing Stations: A cluster based multi source domain adaptation approach,” *Knowledge-Based Syst.*, vol. 145, pp. 15–24, 2018, doi: 10.1016/j.knosys.2017.12.036.

[22] F. Cassola and M. Burlando, “Wind speed and wind energy forecast through Kalman filtering of Numerical Weather Prediction model output,” *Appl. Energy*, vol. 99, pp. 154–166, 2012, doi: 10.1016/j.apenergy.2012.03.054.

[23] S. Hanifi, X. Liu, Z. Lin, and S. Lotfian, “A Critical Review of Wind Power Forecasting,” *Energies*, vol. 13, no. 15, pp. 1–24, 2020.

[24] L. B. V. de Amorim, G. D. C. Cavalcanti, and R. M. O. Cruz, “The choice of scaling technique matters for classification performance,” *Appl. Soft Comput.*, vol. 133, pp. 1–37, 2023, doi: 10.1016/j.asoc.2022.109924.

[25] R. Rosipal, M. Girolami, L. J. Trejo, and A. Cichocki, “Kernel PCA for feature extraction and de-noising in nonlinear regression,” *Neural Comput. Appl.*, vol. 10, no. 3, pp. 231–243, 2001, doi: 10.1007/s521-001-8051-z.

[26] P. K. G. J. Flusser, T. Ören, and V. R. S. Eds, *Advances in Computing and Data Sciences* Springer International Publishing, 2021. [Online]. Available: https://doi.org/10.1007/978-3-031-70906-7

[27] A. Tawakuli, B. Havers, V. Gulisano, D. Kaiser, and T. Engel, “Survey:Time-series data preprocessing: A survey and an empirical analysis,” *J. Eng. Res.*, vol. 13, no. 2, pp. 674–711, 2025, doi: 10.1016/j.jer.2024.02.018.

[28] J. Allgaier and R. Pryss, “Cross-Validation Visualized: A Narrative Guide to Advanced Methods,” *Mach. Learn. Knowl. Extr.*, vol. 6, no. 2, pp. 1378–1388, 2024, doi: 10.3390/make6020065.

[29] O. Pozdniakovych, “Hyperparameter optimization methods in machine learning,” *Model. Inf. Syst. Econ.*, vol. 104, pp. 135–143, 2025, doi: 10.33111/mise.104.12.

[30] T. Bashir, C. Haoyong, M. F. Tahir, and Z. Liqiang, “Short term electricity load

forecasting using hybrid prophet-LSTM model optimized by BPNN," *Energy Reports*, vol. 8, pp. 1678–1686, 2022, doi: 10.1016/j.egyr.2021.12.067.

[31] U. Singh, M. Rizwan, M. Alaraj, and I. Alsaidan, "A machine learning-based gradient boosting regression approach for wind power production forecasting: A step towards smart grid environments," *Energies*, vol. 14, no. 16, 2021, doi: 10.3390/en14165196.

[32] W. Wu and M. Peng, "A Data Mining Approach Combining K-Means Clustering with Bagging Neural Network for Short-Term Wind Power Forecasting," *IEEE Internet Things J.*, vol. 4, no. 4, pp. 979–986, 2017, doi: 10.1109/JIOT.2017.2677578.

[33] F. Shahid *et al.*, "1D Convolutional LSTM-based wind power prediction integrated with PkNN data imputation technique," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 10, p. 101816, 2023, doi: 10.1016/j.jksuci.2023.101816.

[34] L. P. Joseph, R. C. Deo, D. Casillas-Perez, R. Prasad, N. Raj, and S. Salcedo-Sanz, "Multi-Step-Ahead Wind Speed Forecast System: Hybrid Multivariate Decomposition and Feature Selection-Based Gated Additive Tree Ensemble Model," *IEEE Access*, vol. 12, no. April, pp. 58750–58777, 2024, doi: 10.1109/ACCESS.2024.3392899.

[35] A. Kiran and D. Vasumathi, *Data mining: Min–max normalization based data perturbation technique for privacy preservation*, vol. 1090. Springer Singapore, 2020. doi: 10.1007/978-981-15-1480-7_66.

[36] T. Bouadjila, K. Khelil, D. Rahem, and F. Berrezzek, "Hourly Solar Irradiance Forecasting Using Long Short Term Memory and Convolutional Neural Networks," *Smart Grids Sustain. Energy*, vol. 9, no. 2, 2024, doi: 10.1007/s40866-024-00224-2.

[37] Y. S. Kim, M. K. Kim, N. Fu, J. Liu, J. Wang, and J. Srebric, "Investigating the impact of data normalization methods on predicting electricity consumption in a building using different artificial neural network models," *Sustain. Cities Soc.*, vol. 118, no. June 2024, p. 105570, 2025, doi: 10.1016/j.scs.2024.105570.

[38] W. Sun and X. Wang, "Improved chimpanzee algorithm based on CEEMDAN combination to optimize ELM short-term wind speed prediction," *Environ. Sci. Pollut. Res.*, vol. 30, no. 12, pp. 35115–35126, 2023, doi: 10.1007/s11356-022-24586-1.

[39] L. Guo, C. Xu, T. Yu, T. Wumaier, and X. Han, "Ultra-short-term wind power forecasting based on long short-term memory network with modified honey badger

algorithm," *Energy Reports*, vol. 12, no. April, pp. 3548–3565, 2024, doi: 10.1016/j.egyr.2024.09.021.

[40] A. A. Ewees, M. A. A. Al-qaness, L. Abualigah, and M. A. Elaziz, "HBO-LSTM: Optimized long short term memory with heap-based optimizer for wind power forecasting," *Energy Convers. Manag.*, vol. 268, no. May, p. 116022, 2022, doi: 10.1016/j.enconman.2022.116022.

[41] A. R. S. Parmezan, V. M. A. Souza, and G. E. A. P. A. Batista, "Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model," *Inf. Sci. (Ny).*, vol. 484, pp. 302–337, 2019, doi: 10.1016/j.ins.2019.01.076.

[42] S. Liu and D. J. Zhou, "Using cross-validation methods to select time series models: Promises and pitfalls," *Br. J. Math. Stat. Psychol.*, vol. 77, no. 2, pp. 337–355, 2024, doi: 10.1111/bmsp.12330.

[43] Saad Hikmat Haji and Adnan Mohsin Abdulazeez, "Comparison Of Optimization Techniques Based On Gradient Descent Algorithm: A Review," *PalArch's J. Archaeol. Egypt / Egyptol.*, vol. 18, no. 4 SE-, pp. 2715–2743, 2021, [Online]. Available: https://archives.palarch.nl/index.php/jae/article/view/6705

[44] A. C. Wilson, B. Recht, and M. I. Jordan, "A Lyapunov Analysis of Momentum Methods in Optimization," 2016, [Online]. Available: http://arxiv.org/abs/1611.02635

[45] R. Kidambi, P. Netrapalli, P. Jain, and S. M. Kakade, "On the insufficiency of existing momentum schemes for stochastic optimization," *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.*, pp. 1–28, 2018.

[46] J. C. Duchi, P. L. Bartlett, and M. J. Wainwright, "Randomized smoothing for (parallel) stochastic optimization," *Proc. IEEE Conf. Decis. Control*, vol. 12, pp. 5442–5444, 2012, doi: 10.1109/CDC.2012.6426698.

[47] O. Hospodarskyy, V. Martsenyuk, N. Kukharska, A. Hospodarskyy, and S. Sverstiuk, "Understanding the Adam Optimization Algorithm in Machine Learning," *CEUR Workshop Proc.*, vol. 3742, pp. 235–248, 2024.

[48] Z. Yao, A. Gholami, S. Shen, M. Mustafa, and K. Keutzer, "A DA H ESSIAN : An Adaptive Second Order Optimizer for Machine Learning".

[49] A. Apicella, F. Donnarumma, F. Isgrò, and R. Prevete, “A survey on modern trainable activation functions,” *Neural Networks*, vol. 138, no. June, pp. 14–32, 2021, doi: 10.1016/j.neunet.2021.01.026.

[50] L. N. Smith, “A disciplined approach to neural network hyper-parameters: Part 1 -- learning rate, batch size, momentum, and weight decay,” pp. 1–21, 2018, [Online]. Available: http://arxiv.org/abs/1803.09820

[51] X. Ying, “An Overview of Overfitting and its Solutions,” *J. Phys. Conf. Ser.*, vol. 1168, no. 2, 2019, doi: 10.1088/1742-6596/1168/2/022022.

[52] Q. Meng, H. Qian, Y. Liu, Y. Xu, Z. Shen, and L. Cui, “Unsupervised Representation Learning for Time Series: A Review,” vol. 14, no. 8, pp. 1–27, 2023, [Online]. Available: http://arxiv.org/abs/2308.01578

[53] V. Nasteski, “An overview of the supervised machine learning methods,” *Horizons.B*, vol. 4, no. December 2017, pp. 51–62, 2017, doi: 10.20544/horizons.b.04.1.17.p05.

[54] N. R. Regmi, J. R. Giardino, E. V. McDonald, and J. D. Vitek, “A comparison of logistic regression-based models of susceptibility to landslides in western Colorado, USA,” *Landslides*, vol. 11, no. 2, pp. 247–262, 2014, doi: 10.1007/s10346-012-0380-2.

[55] F. Shahid, A. Zameer, and M. J. Iqbal, “Intelligent forecast engine for short-term wind speed prediction based on stacked long short-term memory,” *Neural Comput. Appl.*, vol. 33, no. 20, pp. 13767–13783, 2021, doi: 10.1007/s00521-021-06016-4.

[56] L. L. Li, L. N. Qu, G. Q. Lin, M. K. Lim, and M. L. Tseng, “Improved butterfly optimization algorithm-support vector machine: Short-term wind power forecasting model,” *Soft Comput.*, pp. 4857–4877, 2025, doi: 10.1007/s00500-025-10694-w.

[57] H. Ahmed Salman, A. Kalakech, A. Steiti, and A. History, “Random Forest Algorithm Overview A R T I C L E I N F O,” vol. 2024, pp. 69–79, 2024, [Online]. Available: https://doi.org/10.58496/BJML/2024/007;

[58] A. Ahmadi, M. Nabipour, B. Mohammadi-Ivatloo, A. M. Amani, S. Rho, and M. J. Piran, “Long-Term Wind Power Forecasting Using Tree-Based Learning Algorithms,” *IEEE Access*, vol. 8, pp. 151511–151522, 2020, doi: 10.1109/ACCESS.2020.3017442.

[59] Z. Zhou, X. Li, and H. Wu, “Wind Power Prediction based on Random Forests,” vol. 50, no. Iceeecs, pp. 352–356, 2016, doi: 10.2991/iceeecs-16.2016.73.

[60] Q. Yang, Y. Lin, S. Kuang, and D. Wang, “A novel short-term load forecasting approach for data-poor areas based on K-MIFS-XGBoost and transfer-learning,” *Electr. Power Syst. Res.*, vol. 229, no. May 2023, p. 110151, 2024, doi: 10.1016/j.epsr.2024.110151.

[61] J. Tang, J. Hu, J. Heng, and Z. Liu, “A novel Bayesian ensembling model for wind power forecasting,” *Heliyon*, vol. 8, no. 11, p. e11599, 2022, doi: 10.1016/j.heliyon.2022.e11599.

[62] Y. Zhang, J. Wang, and X. Wang, “Review on probabilistic forecasting of wind power generation,” *Renew. Sustain. Energy Rev.*, vol. 32, pp. 255–270, 2014, doi: 10.1016/j.rser.2014.01.033.

[63] A. Ghasemieh, A. Lloyed, P. Bahrami, P. Vajar, and R. Kashef, “A novel machine learning model with Stacking Ensemble Learner for predicting emergency readmission of heart-disease patients,” *Decis. Anal. J.*, vol. 7, no. May, p. 100242, 2023, doi: 10.1016/j.dajour.2023.100242.

[64] J. Wang, Y. Qian, L. Zhang, K. Wang, and H. Zhang, “A novel wind power forecasting system integrating time series refining, nonlinear multi-objective optimized deep learning and linear error correction,” *Energy Convers. Manag.*, vol. 299, no. October 2023, p. 117818, 2024, doi: 10.1016/j.enconman.2023.117818.

[65] M. Bilgili, B. Sahin, and A. Yasar, “Application of artificial neural networks for the wind speed prediction of target station using reference stations data,” *Renew. Energy*, vol. 32, no. 14, pp. 2350–2360, 2007, doi: 10.1016/j.renene.2006.12.001.

[66] Y. Yan, X. Wang, F. Ren, Z. Shao, and C. Tian, “Wind speed prediction using a hybrid model of EEMD and LSTM considering seasonal features,” *Energy Reports*, vol. 8, pp. 8965–8980, 2022, doi: 10.1016/j.egyr.2022.07.007.

[67] R. Solgi, H. A. Lo, and M. Kram, “Long short-term memory neural network ( LSTM-NN ) for aquifer level time series forecasting using in-situ piezometric observations,” vol. 601, 2021, doi: 10.1016/j.jhydrol.2021.126800.

[68] V. B. Kumar, V. M. Nookesh, B. S. Saketh, S. Syama, and J. Ramprabhakar, “Wind Speed Prediction Using Deep Learning-LSTM and GRU,” *Proc. - 2nd Int. Conf. Smart Electron. Commun. ICOSEC 2021*, pp. 602–607, 2021, doi: 10.1109/ICOSEC51865.2021.9591886.

[69] K. U. Jaseena and B. C. Kovoor, “Decomposition-based hybrid wind speed forecasting model using deep bidirectional LSTM networks,” *Energy Convers. Manag.*, vol. 234, no. November 2020, p. 113944, 2021, doi: 10.1016/j.enconman.2021.113944.

[70] Y. Deng, L. Wang, H. Jia, X. Tong, and F. Li, “A Sequence-to-Sequence Deep Learning Architecture Based on Bidirectional GRU for Type Recognition and Time Location of Combined Power Quality Disturbance,” *IEEE Trans. Ind. Informatics*, vol. 15, no. 8, pp. 4481–4493, 2019, doi: 10.1109/tii.2019.2895054.

[71] Z. Shen, X. Fan, L. Zhang, and H. Yu, “Wind speed prediction of unmanned sailboat based on CNN and LSTM hybrid neural network,” *Ocean Eng.*, vol. 254, no. March, p. 111352, 2022, doi: 10.1016/j.oceaneng.2022.111352.

[72] K. Moharm, M. Eltahan, and E. Elsaadany, “Wind speed forecast using LSTM and Bi-LSTM algorithms over gabal el-zayt wind farm,” *Proc. - 2020 Int. Conf. Smart Grids Energy Syst. SGES 2020*, pp. 922–927, 2020, doi: 10.1109/SGES51519.2020.00169.

[73] S. Harbola and V. Coors, “One dimensional convolutional neural network architectures for wind prediction,” *Energy Convers. Manag.*, vol. 195, no. May, pp. 70–75, 2019, doi: 10.1016/j.enconman.2019.05.007.

[74] H. Sun, T. Song, Y. Li, K. Yang, D. Xu, and F. Meng, “EEMD-ConvLSTM: a model for short-term prediction of two-dimensional wind speed in the South China Sea,” *Appl. Intell.*, vol. 53, no. 24, pp. 30186–30202, 2023, doi: 10.1007/s10489-023-05042-0.

[75] K. Trebing and S. Mehrkanoon, “Wind speed prediction using multidimensional convolutional neural networks,” *2020 IEEE Symp. Ser. Comput. Intell. SSCI 2020*, pp. 713–720, 2020, doi: 10.1109/SSCI47803.2020.9308323.

[76] M. Khodayar, O. Kaynak, and M. E. Khodayar, “Rough Deep Neural Architecture for Short-Term Wind Speed Forecasting,” *IEEE Trans. Ind. Informatics*, vol. 13, no. 6, pp. 2770–2779, 2017, doi: 10.1109/TII.2017.2730846.

[77] X. Wu, G. Jiang, X. Wang, P. Xie, and X. Li, “A Multi-Level-Denoising Autoencoder Approach for Wind Turbine Fault Detection,” *IEEE Access*, vol. 7, pp. 59376–59387, 2019, doi: 10.1109/ACCESS.2019.2914731.

[78] T. Lin, Y. Wang, X. Liu, and X. Qiu, “A survey of transformers,” *AI Open*, vol. 3, no. September, pp. 111–132, 2022, doi: 10.1016/j.aiopen.2022.10.001.

[79] R. Zhu, W. Liao, and Y. Wang, “Short-term prediction for wind power based on temporal convolutional network,” *Energy Reports*, vol. 6, pp. 424–429, 2020, doi: 10.1016/j.egyr.2020.11.219.

[80] K. P. Lin, P. F. Pai, and Y. J. Ting, “Deep belief networks with genetic algorithms in forecasting wind speed,” *IEEE Access*, vol. 7, pp. 99244–99253, 2019, doi: 10.1109/ACCESS.2019.2929542.

[81] S. Sinha and S. Tiwari, “An Improvement in Performance and Computational Cost of ANN Based Wind Speed Prediction System,” *Proc. 4th Int. Conf. Commun. Electron. Syst. ICCES 2019*, no. Icces, pp. 542–546, 2019, doi: 10.1109/ICCES45898.2019.9002315.

[82] K. Khelil, F. Berrezzek, and T. Bouadjila, “GA-based design of optimal discrete wavelet filters for efficient wind speed forecasting,” *Neural Comput. Appl.*, vol. 33, no. 9, pp. 4373–4386, 2021, doi: 10.1007/s00521-020-05251-5.

[83] T. Vinothkumar, S. N. Deepa, and F. V. A. Raj, “Adaptive probabilistic neural network based on hybrid PSO–ALO for predicting wind speed in different regions,” *Neural Comput. Appl.*, vol. 35, no. 27, pp. 19997–20011, 2023, doi: 10.1007/s00521-023-08807-3.

[84] H. Sun, Q. Cui, J. Wen, L. Kou, and W. Ke, “Short-term wind power prediction method based on CEEMDAN-GWO-Bi-LSTM,” *Energy Reports*, vol. 11, no. January, pp. 1487–1502, 2024, doi: 10.1016/j.egyr.2024.01.021.

[85] Q. Li, G. Wang, X. Wu, Z. Gao, and B. Dan, “Arctic short-term wind speed forecasting based on CNN-LSTM model with CEEMDAN,” *Energy*, vol. 299, no. March, p. 131448, 2024, doi: 10.1016/j.energy.2024.131448.

[86] L. Ji, C. Fu, Z. Ju, Y. Shi, S. Wu, and L. Tao, “Short-Term Canyon Wind Speed Prediction Based on CNN—GRU Transfer Learning,” *Atmosphere (Basel).*, vol. 13, no. 5, pp. 1–15, 2022, doi: 10.3390/atmos13050813.

[87] Z. Li, L. Ye, Y. Zhao, X. Song, J. Teng, and J. Jin, “Short-term wind power prediction based on extreme learning machine with error correction,” *Prot. Control Mod. Power Syst.*, vol. 1, no. 1, pp. 4–11, 2016, doi: 10.1186/s41601-016-0016-y.

[88] S. Yang, A. Yuan, and Z. Yu, “A novel model based on CEEMDAN, IWOA, and

LSTM for ultra-short-term wind power forecasting," *Environ. Sci. Pollut. Res.*, vol. 30, no. 5, pp. 11689–11705, 2023, doi: 10.1007/s11356-022-22959-0.

[89] E. J. Ahn and J. Hur, "A short-term forecasting of wind power outputs using the enhanced wavelet transform and arimax techniques," *Renew. Energy*, vol. 212, no. May, pp. 394–402, 2023, doi: 10.1016/j.renene.2023.05.048.

[90] E. Wheatcroft, "Interpreting the skill score form of forecast performance metrics," *Int. J. Forecast.*, vol. 35, no. 2, pp. 573–579, 2019, doi: 10.1016/j.ijforecast.2018.11.010.

[91] R. H. RIFFENBURGH, "Chapter 11 - Using the Reference Guide," in *Statistics in Medicine (Second Edition)*, R. H. B. T.-S. in M. (Second E. RIFFENBURGH, Ed. Burlington: Academic Press, 2006, pp. 187–193. doi: https://doi.org/10.1016/B978-012088770-5/50051-4.

[92] S. Zhang, M. Liu, M. Liu, Z. Lei, G. Zeng, and Z. Chen, "Day-ahead wind power prediction using an ensemble model considering multiple indicators combined with error correction[Formula presented]," *Appl. Soft Comput.*, vol. 148, no. February, p. 110873, 2023, doi: 10.1016/j.asoc.2023.110873.

[93] L. Melalkia, F. Berrezzek, and A. Saim, "A hybrid error correction method based on EEMD and ConvLSTM for offshore wind power forecasting," *Ocean Eng.*, vol. 325, no. February, p. 120773, 2025, doi: 10.1016/j.oceaneng.2025.120773.

[94] Y. Zhao, W. Zhang, and X. Liu, "Grid search with a weighted error function: Hyper-parameter optimization for financial time series forecasting," *Appl. Soft Comput.*, vol. 154, no. February, p. 111362, 2024, doi: 10.1016/j.asoc.2024.111362.

[95] S. Yang, A. Yuan, and Z. Yu, "A novel model based on CEEMDAN, IWOA, and LSTM for ultra-short-term wind power forecasting," *Environ. Sci. Pollut. Res.*, no. 0123456789, 2022, doi: 10.1007/s11356-022-22959-0.

[96] J. Wu, S. Li, J. C. Vasquez, and J. M. Guerrero, "A bi-level mode decomposition framework for multi-step wind power forecasting using deep neural network," *Energy Convers. Manag. X*, vol. 23, no. June, p. 100650, 2024, doi: 10.1016/j.ecmx.2024.100650.

[97] A. Rai, A. Shrivastava, and K. C. Jana, "A robust auto encoder-gated recurrent unit (AE-GRU) based deep learning approach for short term solar power forecasting," *Optik*

*(Stuttg).*, vol. 252, no. May 2021, p. 168515, 2022, doi: 10.1016/j.ijleo.2021.168515.

[98] V. Kosana, K. Teeparthi, and S. Madasthu, "Hybrid convolutional Bi-LSTM autoencoder framework for short-term wind speed prediction," *Neural Comput. Appl.*, vol. 34, no. 15, pp. 12653–12662, 2022, doi: 10.1007/s00521-022-07125-4.

[99] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," pp. 1–9, 2014, [Online]. Available: http://arxiv.org/abs/1412.3555

[100] F. Zhuang *et al.*, "A Comprehensive Survey on Transfer Learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, 2021, doi: 10.1109/JPROC.2020.3004555.

[101] T. Luo, Z. Tang, J. Liu, and B. Zhou, "A Review of Transfer Learning Approaches for Load, Solar and Wind Power Predictions," *Proc. - 2023 Panda Forum Power Energy, PandaFPE 2023*, pp. 1580–1584, 2023, doi: 10.1109/PandaFPE57779.2023.10141494.

[102] L. Melalkia, F. Berrezzek, and A. Saim, "Transfer learning based-hybrid model for short-term wind speed forecasting," *Energy Reports*, vol. 14, no. November 2024, pp. 3237–3253, 2025, doi: 10.1016/j.egyr.2025.10.007.

## Appendix

### Journal Publications


Melalkia, L., Berrezzek, F., Khelil. K ,Saim, A., Nebili. R, “A hybrid error correction method based on EEMD and ConvLSTM for offshore wind power forecasting,” *Ocean Eng.*, vol. 325, no. December 2024, p. 120773, 2025, doi: 10.1016/j.oceaneng.2025.120773.


Melalkia, L., Berrezzek, F., Khelil. K ,Saim, A., Nebili. R, “Transfer learning based-hybrid model for short-term wind speed forecasting,” *Energy Reports*, vol. 14, no. November 2024, pp. 3237–3253, 2025, doi: 10.1016/j.egyr.2025.10.007.