Wavelet-machine learning in wind prediction: A hybrid approach

by

KHATHUTSHELO STEVEN SIVHUGWANA

submitted in accordance with the requirements
for the degree of

DOCTOR OF PHILOSOPHY

In the subject

STATISTICS

at the

UNIVERSITY OF SOUTH AFRICA

SUPERVISOR: PROF E RANGANAI

JANUARY 2026

Declaration

Name: Sivhugwana Khathutshelo Steven

Student number: 50-400-568

Degree: PHD in Statistics

Exact wording of the title of the dissertation as appearing on the electronic copy submitted for examination:**Wavelet-Machine Learning in Wind Prediction: A Hybrid Approach.**

I declare that the above dissertation is my own work and that all the sources that I have used or quoted have been indicated and acknowledged by means of complete references.

I further declare that I submitted the dissertation to originality checking software and that it falls within the accepted requirements for originality.

I further declare that I have not previously submitted this work, or part of it, for examination at UNISA for another qualification or at any other higher education institution.

Sivhugwana Khathutshelo Steven

Signature: 

Date: 10 January 2026

Dedication

To my family, this project would not have been possible without their continuous and unwavering support. Their prayers, patience, and constant encouragement was always my source of my strength.

Acknowledgment

To my supervisor, Prof E. Ranganai, thank you for introducing me to the area of renewable energy. For extraordinary support, encouragement alongside valuable inputs throughout the development of this research output was superb.

List of Publications

The list of publications from this dissertation is provided below.

Article 1: Sivhugwana, K. S., & Ranganai, E. (2024). An ensemble approach to short-term wind speed predictions using stochastic methods, wavelets and gradient boosting decision trees. *Wind*, *4*(1), 44–67. https://doi.org/10.3390/wind4010003.

Article 2: Sivhugwana, K. S., & Ranganai, E. (2024). Short-term wind speed prediction via sample entropy: A hybridisation approach against gradient disappearance and explosion. *Computation*, *12*(8), 163. https://doi.org/10.3390/computation12080163.

Article 3: Sivhugwana, K. S., & Ranganai, E. (2025). Wind speed forecasting with differentially evolved minimum-bandwidth filters and gated recurrent units. *Forecasting*, *7*(2), 27. https://doi.org/10.3390/forecast7020027.

Article 4: Sivhugwana, K. S., & Ranganai, E. (2025). Short-term forecasting of unplanned power outages using machine learning algorithms: A robust feature engineering strategy against multicollinearity and nonlinearity. *Energies*, *18*(18), 4994. https://doi.org/10.3390/en18184994.

Abstract

Wind power relies heavily upon the availability of wind at a specific speed; ultimately, wind power forecasting relies on accurate wind speed forecasts. Although theoretically, wind power increases eightfold as its speed doubles, wind speed is often characterised by linear and nonlinear patterns, nonstationary behaviour as well as intermittency on both location and time scales such that it requires immediate and continuous adjustment to maintain power grid stability. Consequently, predicting wind power from the multidimensional wind speed resource can be a cumbersome and skillful task that cannot be performed effectively with a single forecasting model. However, literature has shown that hybrid approaches that simultaneously combine the strengths of data pre-processing, data optimisation, and data post-processing methods to efficiently, accurately, and reliably quantify wind data are very scant. Using high-resolution wind speed data from the Southern African Universities Radiometric Network (SAURAN) and Wind Atlas for South Africa (WASA), this work proposes improved wavelet machine learning hybrids to fill this gap. In the first module, an efficient and robust combination model leveraging wavelet transform (WT), autoregressive integrated moving average (ARIMA), extreme gradient boosting decision tree (XGBoost), and support vector machine (SVR) denoted by WT-ARIMA-XGBoost-SVR is developed and validated at three different SAURAN locations and at a short-to-long-term forecasting horizon. The proposed hybrid strategy efficiently reduces wind speed forecasting error accumulation caused by the use of linear models by reconciling nonlinear wavelet subseries forecasts. The second module proposed a highly advanced, robust hybrid approach blending WT, neural network autoregression (NNAR), stateless long short-term memory (LSTM) learning, gradient boosting machine (GBM), and sample entropy (SampEn) denoted by WT-NNAR-LSTM-GBM for short-term wind speed forecasting at four different SAURAN stations. With this approach, gradients were effectively mitigated from vanishing and exploding, while improving wind speed forecasting accuracy. Furthermore, this approach emphasised the classification and modelling of wavelet subsignals based on similar complex and deterministic features. The third module developed a hybrid approach by combining maximal-overlap discrete WT (MODWT) filters with gated recurrent units (GRUs) and differential evolution (DE) algorithm denoted by wavelet-MODWT-GRU to forecast wind speed at three different WASA locations in the medium to long-term forecast

horizon. The module provides a more efficient and reproducible method for selecting the most appropriate filters and wavelet decomposition levels to improve wind speed forecasts. As a secondary aim, the work also provided a comprehensive overview of the status of South Africa's electricity supply, focusing on load shedding and unplanned power outages. The review emphasised the importance of scaling up investment in wind power in the country. Moreover, the module also evaluated the generalisability and adaptability of the typical wavelet-machine learning (ML) hybrids in short-term power outage forecasting in order to enhance power grid reliability. Overall, the proposed wavelet-ML strategies demonstrated higher generalisability, reliability, and robustness, and were highly accurate than their counterparts based on various point and probabilistic error indicators, as well as various statistical tests. As a result, these approaches can help operate power grids in real-time, optimise wind power output, minimise energy losses, and distribute wind power effectively. As a result of the four modules, key contributions were summarised and potential directions for future research were highlighted to improve forecasting.

Keywords: Wind speed; Wind power forecasting; Wavelet transform; Hybrid model, Machine learning; Deep learning; Renewable energy; Wind energy; Sample entropy; Differential evolution; Eskom; Load-shedding; Power outage; Forecast accuracy

Contents

List of Tables

Chapter 2: Theoretical Background and Literature Gaps

Chapter 3: Research Methodology

Chapter 4: Multi-Horizon Wind Speed Forecasting Using Stochastic Methods, Wavelets and Gradient Boosting Decision Trees: A Hybrid Approach

Chapter 5: Short-Term Wind Speed Forecasting Using Sample Entropy-Based Hybrid Framework to Address Vanishing Gradients

Chapter 6: Wind Speed Forecasting Using Differentially Evolved Minimum-Bandwidth Filters and Gated Recurrent Units

Chapter 7: A Robust Wavelet Machine Learning Framework for Short-Term Forecasting of Unplanned Power Outages

Chapter 8: Conclusion and Future Research

List of Figures

Chapter 1: Introduction

Chapter 2: Theoretical Background and Literature Gaps

Chapter 3: Research Methodology

Chapter 4: Multi-Horizon Wind Speed Forecasting Using Stochastic Methods, Wavelets and Gradient Boosting Decision Trees: A Hybrid Approach

Chapter 5: Short-Term Wind Speed Forecasting Using Sample Entropy-Based Hybrid Framework to Address Vanishing Gradients

Chapter 6: Wind Speed Forecasting Using Differentially Evolved Minimum-Bandwidth Filters and Gated Recurrent Units

Chapter 7: A Robust Wavelet Machine Learning Framework for Short-Term Forecasting of Unplanned Power Outages

Abbreviations

Abbreviation	*Definition*
ABC	Artificial Bee Colony
ACF	Autocorrelation Function
AD	Anderson Darling Test
AIC	Akaike Information Criterion
ANN	Artificial Neural Network
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive Moving Average
BIC	Bayesian Information Criterion
bi-LSTM	Bi-directional Long Short-Term Memory Networks
CRPS	Continuous Rank Probability Score
CSIR	Council for Scientific and Industrial Research
CWT	Continuous Wavelet Transform
DB	Daubechies
DE	Differential Evolution
DFT	Discrete Fourier Transform
DM	Diebold Mariano Test
DSS	Dawid Sebastiani Score
DWT	Discrete Wavelet Transform
ED	Euclidean Distance
Eskom	Electricity Supply Commission
FFT	Fast Fourier Transform
FT	Fourier Transform
FWT	Fast Wavelet Transform
GA	Genetic Algorithm
GBDTs	Gradient Boosting Decision Trees
GBM	Gradient Boosting Machine
GHGs	Greenhouse Gases
GOSS	Gradient-Based One-Sided Sampling
GRU	Gated Recurrent Units
KF	Kalman Filter
KNN	K-Nearest Neighbour
LA	Least Asymmetric
LASSO	Least Absolute Shrinkage and Selection Operator
LGB	Light Gradient Boosting Machine
LSTM	Long Short-Term Memory Networks
MAE	Mean Absolute Error
MB	Morris Minimum Bandwidth
MD	Murphy Diagram
ML	Machine Learning
MLP	Multilayer Perceptron

Abbreviation	*Definition*
MODWT	Maximal Overlap Discrete Wavelet Transform
MRA	Multiresolution Analysis
MSE	Mean Square Error
MZ	Mincer–Zarnowitz Test
NILA	Niche Immune Lion Algorithm
NNAR	Neural Network Autoregression
NWP	Numerical Weather Prediction
OCLF	Other Capacity Loss Factor
PACF	Partial Autocorrelation Function
PI	Prediction Interval
PINAD	Prediction Interval Normalised Average Deviation
PINAW	Prediction Interval Normalised Average Width
PINC	Prediction Interval with Nominal Confidence
PIW	Prediction Interval Width
PIT	Probability Integral Transform
PL	Pinball Loss
PSO	Particle Swarm Optimisation
RBF	Radial Basis Function
R^2	Coefficient of Determination
RF	Random Forest
RMSE	Relative Mean Square Error
RNN	Recurrent Neural Networks
RSA	Republic of South Africa
RVM	Relevance Vector Machine
SA	Simulated Annealing
SampEn	Sample Entropy
SAURAN	Southern African Universities Radiometric Network
SGB	Stochastic Gradient Boosting Machine
SRM	Structural Risk Minimisation
Stateless LSTM	Stateless Long Short-Term Memory Networks
STFT	Short-Term Fourier Transform
SVR	Support Vector Regression
TPCLF	Total Planned Capacity Loss Factor
TUCLF	Total Unplanned Capacity Loss Factor
TUCLF.OCLF	Total Unplanned Capacity Loss Factor
UCLF	Unplanned Capacity Loss Factor
VAR	Vector Autoregressive Models
WASA	Wind Atlas South Africa
WD	Wavelet Decomposition
WPD	Wind Power Density
WRF	Weighted Random Forest
WT	Wavelet Transform
XGBoost	Extreme Gradient Boosting Machine

List of Symbols

P_{WT}	Wind power from a particular turbine
P_W	Wind power (Total available)
ρ	Density of air
C_p	Drag power coefficient of wind turbine
v	Wind speed
θ_t	Commitment of the unit
Ω_u	Unit operational cost
ϑ	Cost of electricity per unit
E_k	Kinetic energy
T	Local temperature
P_{air}	Air pressure
A	Wind turbine swept area
l_s	Turbine blade
r_s	Radius of the hub
L	Temperature lapse rate
η_a	Gearbox efficiency
η_b	Generator efficiency
η_c	Electrical efficiency
η_{tot}	Conventional efficiency
P_e	Effective power
C_F	Wind power efficiency
κ	Tip speed ratio
ω	Angular velocity
v_τ	Wake effect
$Y(\epsilon)$	CFT of a signal $Y(t)$
$S_g\, f(t,\epsilon)$	Continuous STFT
$Y(z)$	The z transform of the function $y(n)$
$\overline{\omega_l}(z)$	Polyphase components
$\psi_{s,h}(t)$	Wavelet child
$\psi(t)$	Mother wavelet
$K(.)$	Kernel function.
p	Order of non-seasonal AR process
q	Order of non-seasonal MA process
P	Order of seasonal AR process
Q	Order of seasonal MA process
S	Seasonal component
D	Seasonal differencing
d	Non-seasonal differencing
B	Backshift operator
$\phi_p(B)$	Non-seasonal AR polynomial
$\theta_q(B)$	Non-seasonal MA polynomial

$\Phi_P\ (B^s)$	Seasonal AR
$\Theta_Q(B^S)$	Seasonal MA
$\bar{y}$	Sample mean
σ_{y_t}	Standard deviation of the signal y_t
λ^L	Shrinkage parameter in LASSO
λ^s	Scaling factor in DE
$\boldsymbol{w}$	Weights
K	Kernel function
β^*	Regression weights
$\mathbf{\Gamma}_{\mathrm{i}}$	Donor Vector
$\boldsymbol{U}_i$	Trial vector
P_{cr}	Crossover probability
σ^f	Sigmoid function
φ^f	Tangent hyperbolic function
Ω_n	Training set
Ω_N	Complete data set, with $N > n$ such that $\Omega_{\mathrm{n}} \subset \Omega_{\mathrm{N}}$
$\mathcal{L}(.)$	Differential loss function
D_i	Decision tree
Υ_i, Υ_i^*	Slack variables

This page is intentionally blank

Chapter 1

Introduction

1.1 Research Context

Energy is pivotal to global socio-economic development. Over the past decades, energy demand (projected to grow by more than 100% by 2050) has been rapidly increasing as economic globalisation advances [1]. Due to the sheer scarcity of fossil energy and the growing severity of environmental degradation, the use of fossil energy has steadily declined over the years [1–4]. Accordingly, there has been a notable corresponding swing towards the consumption of renewable energy, particularly wind energy, which has experienced a more significant growth than other sources of clean energy for various reasons. These include its low installation and maintenance costs, environmental friendliness, the widespread availability of the wind resources, as well as its inexhaustible nature [2–4]. For example, the production of wind power has surged by 273 terawatt hours (*TWh*) (reaching 1 870 *TWh* in 2021) , marking a 50% increase from 2020, thereby establishing it as the leading renewable energy technology worldwide [5,6]. Concurrently, as of 2021 the worldwide installed capacity of wind power stood at 830 gigawatts (*GW*), with a significant portion, 93 %, originating from onshore wind installations, and the remainder, 7%, from offshore wind installations [5,6]. Consequently, CO_2 emissions in 2021 fell by about 220 metric tons (*Mt*) to 36.3 gigatons (*Gt*) [7]. By 2050, renewable energy, predominantly wind and solar, is expected to account for more than 25% of all global energy consumption [5,6].

South Africa has plentiful wind resources (see e.g., [8–12]), as such wind operational capacity has significantly grown from an immaterial 257 *MW* in 2013 to over 3 443 *MW* by 2022 [10]. As a result, annual wind power generation has increased significantly from 0.01 *TWh* to 9.7 *TWh* [10]. South Africa receives wind speeds between 10-11 ms^{-1}at 100 m altitude, which allows it to generate even larger volumes of wind power [9] (also see Figure 1.1) despite this low adoption of wind energy. Furthermore, authors of [9] estimated South African offshore wind annual energy production to be around 2 387.08 *TWh* available at depths less than 1 000 m. Theoretically, this would be sufficient to meet

South Africa's annual electricity demand (206 *TWh*) by over ten times. Whilst wind power is expected to constitute over 50% of the European energy mix by 2050, the South African Integrated Resource Plan (IRP) 2019 aims to increase wind share to 21% (17 742 *MW*) by 2030, up from 10% (9 200 *MW*) in IRP 2010 [13,14]. Indicating that South Africa still falls short of the global wind power generation level.

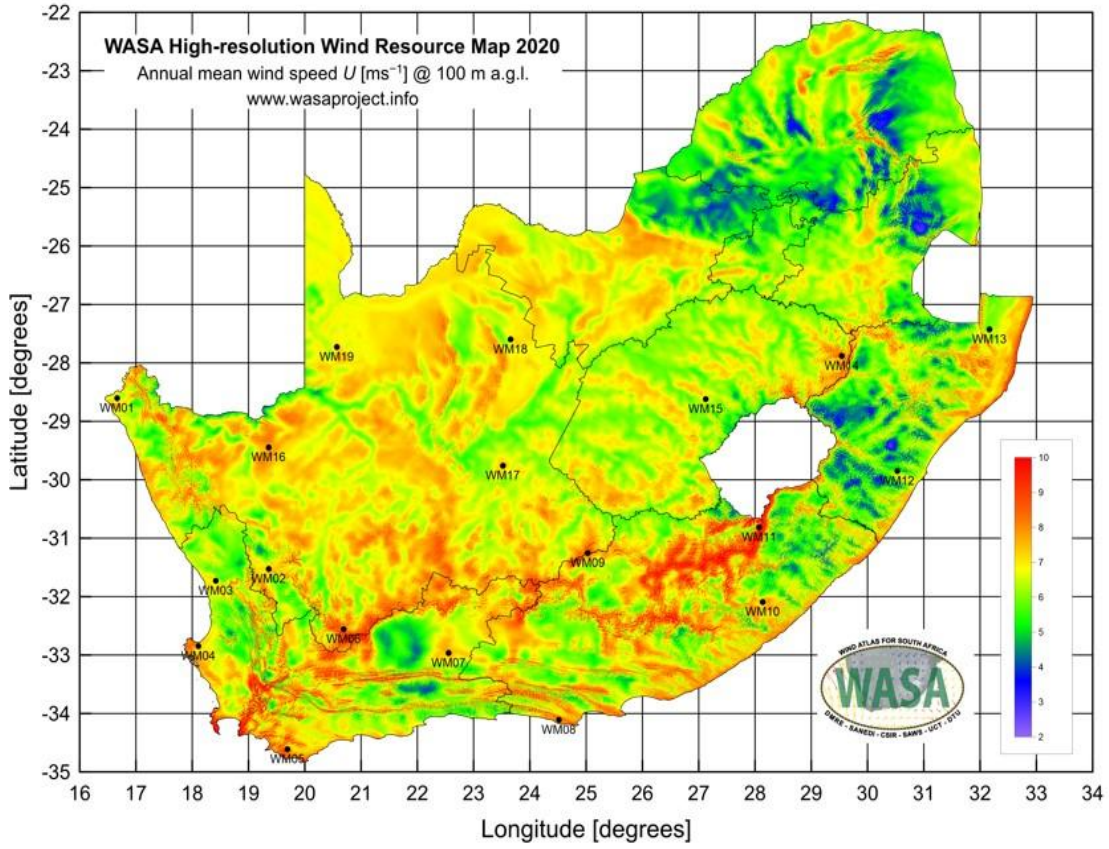
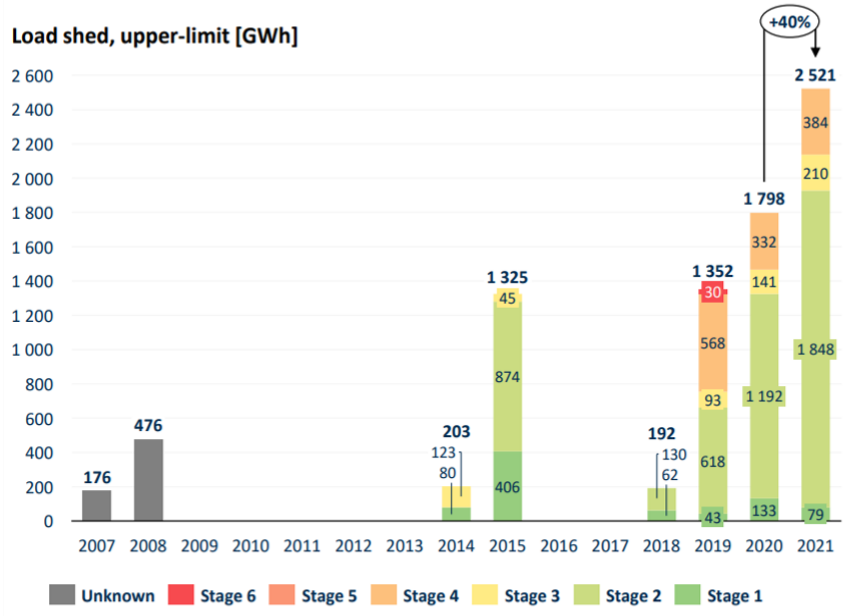


Figure 1.1. WASA high resolution wind resource map: mean wind speed [12].

South Africa continues to rely excessively on coal as a primary energy source [5,6,10,11,13-16]. In 2021, strained and old coal plants that are pushed to their limits and beyond produced over 90% of the overall electricity output, reaching 185 459 gigawatt hours (*GWh*), whereas renewable energy sources accounted for just 6% (of 215 337 *GWh*) [17]. With this consumption, South Africa is Africa's largest energy consumer and emits the most greenhouse gases (GHG) on the continent [18,19]. The decarbonisation of the energy sector, with a specific focus on wind power, is pivotal for achieving a sustainable energy supply. On the contrary, South Africa has reduced its CO_2 emissions target from a range of 398-614 *Mt* to 350-420 *Mt* by 2030, showing a step away from curtailing CO_2 emissions by 42% in 2025 [5] (also see e.g., [20]).

Overall, South Africa's energy sector is characterised by power grid inefficiencies [21] such that the Electricity Supply Commission (Eskom), responsible for supplying 90 % of electricity in the country, has resorted to load-shedding—a method of manually reducing power consumption across the nation to avoid a complete grid failure or collapse (see Figure 1.2). Since 2007, ongoing electricity supply disruptions have stifled foreign investment, raised the cost of production, and driven electricity prices up by more than 450% [5,6,21].



Year	Duration of outages (hours)	Energy Shed (GWh)
2007	-	176
2008	-	476
....	....	....
2014	121	203
2015	852	1325
2016	-	-
2017	-	-
2018	127	192
2019	530	1352
2020	859	1798
2021	1169	2521

Notes: Loadshedding assumed to have taken place for the full hours in which it was implemented. Practically, load shedding (and the Stage) may occassionally change/ end during a particular hour; Total GWh calculated assuming Stage 1 = 1 000 MW, Stage 2 = 2 000 MW, Stage 3 = 3 000 MW, Stage 4 = 4 000 MW, Stage 5 = 5 000 MW, Stage 6 = 6 000 MW;

Figure 1.2. Upper limit of loadshedding by year in South Africa [11].

The urgent need to reduce GHG emissions, alongside escalating electricity prices and an abundance of wind resources in South Africa, underlines the importance of investing in wind energy as a sustainable, and affordable strategy (also see [9,22]). Furthermore, the fact that wind power farms are typically smaller than traditional coal power stations, and thus requiring less transmission infrastructure costs, makes them ideal for utilisation at a local level, thereby enhancing their sustainability. Besides, wind power can provide electricity during Eskom's peak demand hours (beyond daylight times when solar power is unavailable), which means it is uniquely positioned to prevent the costly load-shedding if properly harnessed [10,11] (also see details in [15]).

1.2 Research Rationale

1.2.1 Motivation

While wind energy offers numerous advantages, integrating large volumes of wind power into the power grid is a complex task [22]. As wind power output varies over multiple time frames, the grid operator's daily operating procedures will also vary accordingly. In fact, the generation of wind energy is reliant upon the presence of wind at a specific velocity, and as such the process of harvesting wind power in a deliberate and scheduled manner becomes more complex [22]. Thus, the most influential variable for wind power, wind speed, is unpredictable, sporadic, and nonlinear [2–4]. Consequently, wind power is intermittent and inconsistent as theoretically depicted in the power curve equation below [3,4,22]:

$$P_{WT} = \frac{1}{2}\,\rho A C_p v^3, \tag{1.1}$$

where P_{WT} denotes the wind power from a particular turbine, A is the area intercepting wind, ρ is the density of air ($kg.m^{-3}$), C_p is the drag power coefficient of wind turbine, and v is the wind speed (ms^{-1}) [22–24]. Theoretically, from Equation (1.1), wind power will increase eightfold when the wind speed doubles [24]. As a result, the uncertainty in the predicted wind power will be proportional to the prediction error in the wind speed. In [25], authors also showed that wind turbine energy costs associated with prediction errors can reach 10% of total wind energy turnover. This can also be attributed to the fact that the power curve is very steep in the optimal wind power generation region (3 ms^{-1} to 15 ms^{-1}), which further exacerbates wind speed prediction errors. Consider the expected profit Equation (1.2) that an energy dispatcher seeks to optimise (also see e.g., [26,27] for details):

$$E_{\text{profit}} = \sum_{t=1}^{n} (\hat{y}_t . \vartheta)\,\theta_t - \Omega_u, \tag{1.2}$$

where $\hat{y}_t$ denotes the predicted power generation of an individual power plant (can be a mixture of fossil plant or wind farm), n is the number of power plants in the electricity grid, ϑ denotes the cost of electricity per unit, $\theta_t \in \{0, 1\}$ represents the commitment of the unit, and Ω_u is the operational costs parameter (e.g. fossil fuel costs, maintenance costs, etc.). Fundamentally, the goal is to maximise power generation such that Equation (1.2) satisfies the power demand equation given by:

$$E_{PD} = P_f + P_R = \sum_{t=1}^{n} (\hat{y}_t . \theta_t), \tag{1.3}$$

where P_f and P_R respectively denote the power generated from the combustion of fossil fuels and from renewable energy sources such as wind [26]. The optimisation is usually achieved by prioritising (in the feed-in process) and fully exploiting renewable energy such as wind power as it is cost effective and abundant thereby minimising the cost (Ω) associated with burning the fossils to generate energy. In this regard, accurate and reliable predictions of the wind power available from wind farm plants are critical and essential to planning and unit commitment (or economic dispatch). However, wind power's main input (wind speed), as a physical quantity, requires immediate and continuous adjustment to maintain power grid stability.

1.2.2 Problem Statement

A primary contributor to the generation of wind power is wind speed forecasting; consequently, the development of a reliable model to estimate wind power though complex is critical [28]. Besides, the optimal performance of the power grid and the uniform distribution of wind power pivots on robust, and accurate wind speed predictions [22,29,30]. In the wind forecasting framework, there are four main timescales/horizons [23], viz.; very short-term (seconds to 30 minutes), short-term (from 30 minutes to 6 hours), medium-term predictions (from 6 hours to 1 day), and long-term (from 1 day to weeks). Very short-term predictions are often reserved for real-time wind turbine management and load monitoring. For pre-load sharing, grid management, and market trading short-term forecasting scale are ideal and preferred. Medium-term predictions are often reserved for wind turbine maintenance and wind power system planning, whilst long-term predictions are reserved for the development of wind power plants.

In the literature, the wind forecasting model suffers from statistical, representational, and computational deficiencies [26]. The representational problem refers to the lack of model scalability, which usually limits the model's ability to capture the complexity characteristics (e.g. nonlinearity, wind gust, etc.) of the data; whilst the statistical relates to the deficiency found when the training dataset is insufficient to represent the actual process or phenomena under investigation. Computational deficiencies include the situation where model training is discontinued because of convergence to the local minima (often due to vanishing gradients and explosions). In wind

forecasting, physical models are often computationally intensive and reserved for the medium-term forecasting; statistical models (e.g. autoregressive moving average (ARMAs)) that are amenable to nonlinearity are reserved for the short-term; and traditional machine learning (ML) models (e.g. feed forward networks (FNNs)) although being highly accurate are data greedy and susceptible to vanishing gradients and are reserved for the short-term forecasting of complex data. To remedy these deficiencies and exploit the advantages of each model, a hybrid blend of at least two methods creates a more accurate predictive strategy [31]. Furthermore, hybrids are known for their best performance over ultra-short to long-term forecasting horizons; as such a comprehensive understanding of their principles can aid with the assessment of their suitability for use. Ultimately, this research work seeks to address the drawbacks of conventional statistical, ML, and hybrid models while improving wind speed prediction accuracy using enhanced tailor-made wavelet-ML-hybrids. *Thus, the overarching objective of this thesis is to improve wind speed prediction accuracy using new and enhanced tailor-made wavelet-ML hybrids. Although the main objective of the study is to enhance wind speed prediction accuracy, the generalisability and adaptability of these wavelet-ML frameworks is also tested in short-term power outage prediction to help enhance power grid reliability, given that wind variability affects the stability of the power system.*

1.2.3 Research Questions

The study would like to answer the following research questions, viz.;

- What is the current state of the electricity supply in South Africa, and what strategies can be employed to manage it?
- What are the limitations of traditional statistical, ML, and hybrid methods within the wind speed forecasting literature? What strategies can be employed to mitigate these deficiencies?
- How can hybrid frameworks be developed to reduce error, and increase accuracy in wind speed forecasting across different forecasting time scales?
- Can the generalisability and adaptability of these wavelet-ML hybrids be extended to other related fields?

1.2.4 Research Objectives

The current research work would like achieve the following objectives, viz.;

- To provide an extensive review of the limitations and benefits of the current statistical, ML, and hybrid methodologies within the framework of wind forecasting *(Chapter 2&3).*

- To develop and test a highly efficient and robust hybrid model that leverages WT, autoregressive integrated moving average (ARIMA), extreme gradient boosting decision trees (XGBoost), and support vector regression (SVR) methods. This strategy is designed to reduce error accumulation due to the common use of traditional linear combination models in wind speed forecast reconciliation *(Chapter 4).*

- To formulate and evaluate an advanced, robust hybrid methodology that incorporates WT, neural network autoregression (NNAR), sample entropy (SampEn), and gradient boosting machine (GBM). This approach aims to address the issues of vanishing gradients found to be prevalent in traditional neural network methods *(Chapter 5).*

- To develop and validate a hybrid model based on maximal-overlap discrete wavelet transform (MODWT) filters with gated recurrent units (GRUs) and differential evolution (DE) algorithm. This strategy aims to provide a more efficient and reproducible method for selecting the appropriate filters and wavelet decomposition levels in wind forecasting *(Chapter 6).*

- To assess the generalisability and adaptability of the typical wavelet-ML hybrids in short-term power outage forecasting, thereby improving power grid reliability. This objective culminated in the development and evaluation of a highly accurate stacked model based on least absolute shrinkage and selection operator (LASSO), wavelet transform (WT), random forest (RF), AdaBoost with regression and threshold (AdaBoostRT), and relevance vector machine (RVM) for unplanned outage predictions to facilitate proactive decision-making *(Chapter 7).*

- To conduct performance analysis of the proposed hybrid strategies across various datasets from different locations at varying forecast horizons using different performance metrics and statistical tests.

1.3 Research Novelty and Contribution

Literature has shown that hybrids that simultaneously combine data pre-processing, data optimisation, and data post-processing methods to efficiently, accurately, and reliably quantify wind data are very scant. Hence, the main contributions of this research work are summarised as follows:

- The use of individual models is often focused on prediction whilst overlooking other critical characteristics (i.e. breakdown points, signal trends, etc.) of the wind speed data. In practice, it is essential to discover useful information in the data via pre-processing and to characterise the data before prediction. Thus, pre-processing techniques, particularly WTs, are pivotal, as they aim to reduce random disturbances in the data sequence and increase prediction accuracy. Different from the Fourier Transform (FT), WTs provide excellent resolution in the time-frequency domain in that they can efficiently capture temporary resolution and have better signal representation (i.e. can detect signal trends and discontinuities). Hence, WTs are used to decompose the signal into different scale components with statistically more sound properties to improve the prediction accuracy of the proposed forecasting strategies (also see [22,28] for details). *This limitation is addressed in Chapters 4-7.*

- In the wind speed forecasting literature, hybrid models are generally implemented by first decomposing the original series into subsignals of different scales, modelling and forecasting the subsignals, and then combining subsignal forecasts linearly. However, this approach often disregards other critical elements of the subsignals. For instance, when decomposing signals through WTs, the complexity and variance of the subsignals at lower levels of decomposition is different from those at higher levels of decomposition. Consequently, similarly handling these subsignals often diminishes the accuracy of the final predictions. As such, it is important that each subsignal be individually inspected and modelled relative to its inherent complexity and

behaviour to improve the models' prediction accuracy. Hence, this study classified decomposed subsignals based on their identical complex and deterministic properties using information theory methods (i.e. SampEn). As a result, gradient explosion is managed and prediction accuracy is improved by applying the most appropriate modelling and forecasting approach. For example, in one of the proposed hybrid frameworks, WT-NNARLSTM-GBM, NNAR was used for low-variant (with low SampEn values) subsignals, whilst LSTM handled those subsignals with high SampEn values (more complex) to effectively manage gradient disappearance (also see [28] for details). *This limitation is addressed in Chapter 5.*

- In the literature, the majority of hybrid strategies implemented rely excessively on the linear combination approach of subsignal forecasts to generate the final forecast of the original signal under study. Even though this conventional approach is simple and efficient, it lacks accuracy and stability when combining wind subsignal forecasts that are inherently nonlinear, resulting in excessive error accumulation thereby compromising grid stability and reliability and thus compounding grid operating costs. To capture the complex nonlinear structure inherent in wind subsignal forecasts, it is imperative to use a nonlinear forecast combination method. It is also essential to optimise the proposed forecast combination method using other nonlinear methods at various stages of the prediction process, viz.; model parameter optimisation and output error correction. Hence, this study exploits the power of robust nonlinear SVRs and GBMs as they are characterised by improved accuracy and the ability to effectively capture inherent nonlinearity when combining nonlinear wind speed subsignal forecasts (also see [22,28] for details). *This limitation is addressed in Chapters 4-5.*

- Wind speed data are inherently chaotic, complex, and volatile and a single model may not possess sufficient learning capacity to fully explain and capture this behaviour. This character often leads to overestimation at low wind speed and underestimation at high wind speed which ultimately disrupts real-time grid operations, uniform wind power distribution, and output optimisation. However, ML methods (to an extent) and their hybrids can effectively and efficiently capture nonlinear characteristics inherent in wind speed data that are impossible to capture using simple conventional stochastic and linear methods

such as ARIMA. Ultimately, the following classes of ML methods are proposed, viz.; XGBoost, NNAR, SVR, GBM, and stateless LSTM for the development of two highly effective hybrids, viz.; WT-ARIMA-XGBoost-SVR and WT-NNAR-LSTM-GBM. The earlier hybrid seeks to efficiently reduce wind speed forecasting error accumulation caused by the use of linear models to reconcile nonlinear wavelet subsignal forecasts, whilst the latter aims to mitigate gradients from vanishing and exploding (also see [22,28] for details). *This limitation is addressed in Chapters 4-5.*

- The selection of the most suitable wavelet filter should be tailor-made for the specific problem at hand to enhance forecast accuracy. Literature has shown an excessive application of time-variant discrete wavelet transform (DWT) that cannot adequately capture random delays as opposed to time-invariant MODWT transforms with better localisation properties. Above other wavelet filters, the excessive application of Daubechies in the data pre-processing phase was observed in the literature. Despite great localisation features, these filters suffer from spectral leakage and are subject to signal distortion due to their asymmetric properties. Furthermore, the wavelet decomposition level has a significant influence on forecasting results, more so than the choice of a wavelet filter. However, the level of decomposition is often determined through a computationally intensive trial-and-error approach which is difficult to reproduce. Overall, the study combines time-invariant MODWT filters (i.e. the least asymmetric (LA), Daubechies (DB), and the Morris minimum bandwidth (MB)) with GRUs and DE algorithms aimed at providing an efficient and reproducible method for selecting the most appropriate filters and wavelet decomposition levels to improve wind speed forecasts (also see [29] for details). *This limitation is addressed in Chapter 6.*

- Though DE algorithms are effective and efficient in handling complex optimisation problems as well as nonlinear continuous data (e.g. wind speed), they are often overlooked as compared to the computationally expensive genetic algorithms (GA) in the wind forecasting literature. Consequently, these evolutionary algorithms which can be easily comprehended and are easy to reproduce, are used to calculate the optimal decomposition level. *This limitation is addressed in Chapter 6.*

- According to the assessment of the current energy situation in the South African energy sector, reliable and precise power outage prediction models are urgently needed to unmask and manage the effects of load-shedding as well as enhance proactive and informed decisions (e.g. investments in wind energy technology (e.g. wind farms)). In this regard, the study blends the LASSO, WTs, RF, AdaBoostRT, and RVMs for unplanned power outage forecasting. The model is designed in such a computationally efficient way, can handle multicollinearity and missing values, can capture nonlinearity, can avoid overfitting, has greater generalisability, has minimal bias, and can predict unplanned power outages with such a level of accuracy. Accordingly, the study successfully evaluated the generalisability and adaptability of the typical wavelet-ML hybrids in short-term power outage forecasting, thereby improving power grid reliability. *This limitation is addressed in Chapter 7* (also see [15] for details).

- In the literature, deterministic predictions have been the top priority for researchers and are constantly being improved to make them more accurate. This is because they are simple to train and efficient, and the principles behind them are easy to comprehend. However, there are instances where deterministic predictions struggle to adequately represent the chaotic characterisation of wind speed. As a result, they can pose risks to wind power market scenarios because of the sporadic gap between the observed value and the predicted value. Besides being computationally expensive, probabilistic forecasts can provide comprehensive information about wind speed behaviour that is crucial for assessing uncertain situations, and planning various strategies. As such, in each of the case studies, comprehensive wind speed both deterministic and probabilistic predictions were thoroughly evaluated. *This limitation is addressed in Chapters 4-7.*

1.4 Research Scope and Limitations

1.4.1 Research Scope

The study aims to accurately and reliably quantify South African wind resources to inform key stakeholders (e.g. wind power investors, utility companies, etc.) of their importance and untapped potential using high-resolution Southern African Universities Radiometric Network (SAURAN) (https://sauran.ac.za/) and Wind Atlas for South Africa (WASA) (https://www.wasaproject.info/) wind speed data.

Furthermore, Eskom's power grid data (https://www.eskom.co.za/dataportal/) will also be used in the study in order to predict unplanned outages for proactive decision-making. All the models were trained and tested on an Intel Corei5 (and i7) processor running in the HP (and DELL) notebook development environment using R program. The study has been conducted in a manner that is reliable, and reproducible, and has provided algorithms and comprehensive assessment metrics that are suitable for wind speed modelling and prediction.

1.4.2 Research Limitations

As this research focuses solely on wind speed forecasting in Southern Africa, it is limited by its use of small datasets, univariate wind speed series (does not consider numerical weather prediction (NWP) variables), and geographical focus. Besides, the proposed hybrids were solely applied to the energy space. It would be interesting to test the same model using time series data from the field of finance (i.e. the stock market). Further, only three types of wavelet filters were considered in the study, viz.; LA, DB, and MB. Other available mother wavelets such as Morlet and Mexican hat wavelets could be considered at varying vanishing moments.

1.5 Thesis Outline

This current work is structured into eight (8) chapters in which the research results are covered in Chapters 4 to 7 contributing to the overall objective (also see Figure 1.3).

Chapter 1 provides an overview of the study, including its context, motivation, objectives, purpose, limitations, and how the study will contribute to the wind energy literature in its conclusion.

Chapter 2: Section 2.1 provides fundamental concepts of wind energy alongside its challenges and advantages. The section further highlights the critical and influential parameters to consider when generating wind power using technology such as wind turbines. Section 2.2 lays the groundwork by covering fundamental theoretical concepts of wavelets and filtering methods. In the beginning, the chapter looked into the FT, short-time Fourier transform (STFT), and multi resolution analysis (MRA). The chapter also provides a detailed explanation of wavelet properties and the

mathematical concepts behind WTs. Additionally, the study delve into the theoretical framework and key components of filter banks, including analysis filter banks, downsampler, upsampler, and synthesis components, along with their essential properties. To conclude, the section briefly discuss filter bank properties such as polyphase filtering and perfect reconstruction concepts. In Section 2.3, a detailed review of classes of wind forecasting was conducted. The chapter further reviews and evaluates 30 hybrid wind forecasting research articles from more distinct geographical locations across different countries from different continents. The chapter concludes by summarising key findings derived and the gaps identified from the literature analysis. The findings (particularly the identified gaps) emanated from this chapter laid the foundation for Chapters 3-7.

Chapter 3 briefly describes the models fitted on the South African wind speed and power grid data, viz., the Box-Jenkins models, neural networks, information theory, variable selection and regularisation methods, metaheuristic algorithms, boosting decision trees, bagging methods, adaptive boosting methods, and support vectors. In fact, the chapter is divided into six main sections, viz., baseline models, feature engineering models, main models, proposed hybridisation framework, performance evaluation metrics, and conclusions.

Chapter 4 introduces a new hybrid approach, viz.; WT-ARIMA-XGBoost-SVR to improve wind speed predictions at short-term to long-term prediction scales by leveraging the strengths of WT, ARIMA, GBDTs, and SVR. The emphasis is placed on minimising error accumulation (mainly caused by the use of linear forecast combination approaches) in wavelet subsignal forecast reconciliation.

Chapter 5 proposed robust hybrid methodology that incorporates WT, NNAR, SampEn, and GBM. This approach aims to address the issues of vanishing gradients found to be prevalent in traditional neural network methods and wind gusts, which, to a certain extent, led to over and under-prediction of extreme values in Chapter 4.

Chapter 6 proposes a hybrid approach wavelet-MODWT-GRU to effectively assess the influence of wavelet filter and decomposition level on accuracy. As an alternative to only applying LA as in Chapters 3 and 4, this chapter implements and tests three wavelet filters, viz.; the LA, DB, and MB at varying wavelet decomposition levels and

vanishing moments for the medium to long-term forecast horizon. Moreover, the proposed framework addresses the accuracy and reliability of hybrids when the forecasting horizon is extended to long-term horizons, which was observed in the hybrids in Chapters 4 and 5.

In **Chapter 7,** a comprehensive overview of the current South African energy sector is provided, with a focus on the challenges faced by the sector such as load-shedding. Further, an advanced feature engineering strategy is proposed: RVM-WT-AdaBoostRT-RF for forecasting unplanned power outages. The rationale is to effectively evaluate whether the generalisability and adaptability of the typical wavelet-ML hybrids can be extended in short-term power outage forecasting, thereby improving power grid reliability.

Chapter 8 concludes the thesis with a discussion of future research work, as well as a comparison of the objectives to the results obtained in Chapters 4 to 7.

1.6 Thesis Style and Notation Disclaimer

This is a *"Doctoral Thesis with monograph"* compiled according to the University of South Africa (UNISA) Guidelines. Please note that some of the variables/symbols employed in this work were reused in different chapters. They represent (or carry) different concepts (or meanings) in different chapters. Hence, they should always be interpreted in the context of the particular chapter to get the most out of the current thesis. The terms "*subseries"* and "*subsignals"* are used interchangeably throughout this research. Also, the study synonymously used the words *"forecasting"* and *"prediction"* to imply estimating of future wind speeds using historic data. In the context of this study, models' ability to produce accurate results from test data is referred to as model efficacy, whilst model's efficiency is its ability to produce accurate results using minimal computational resources.

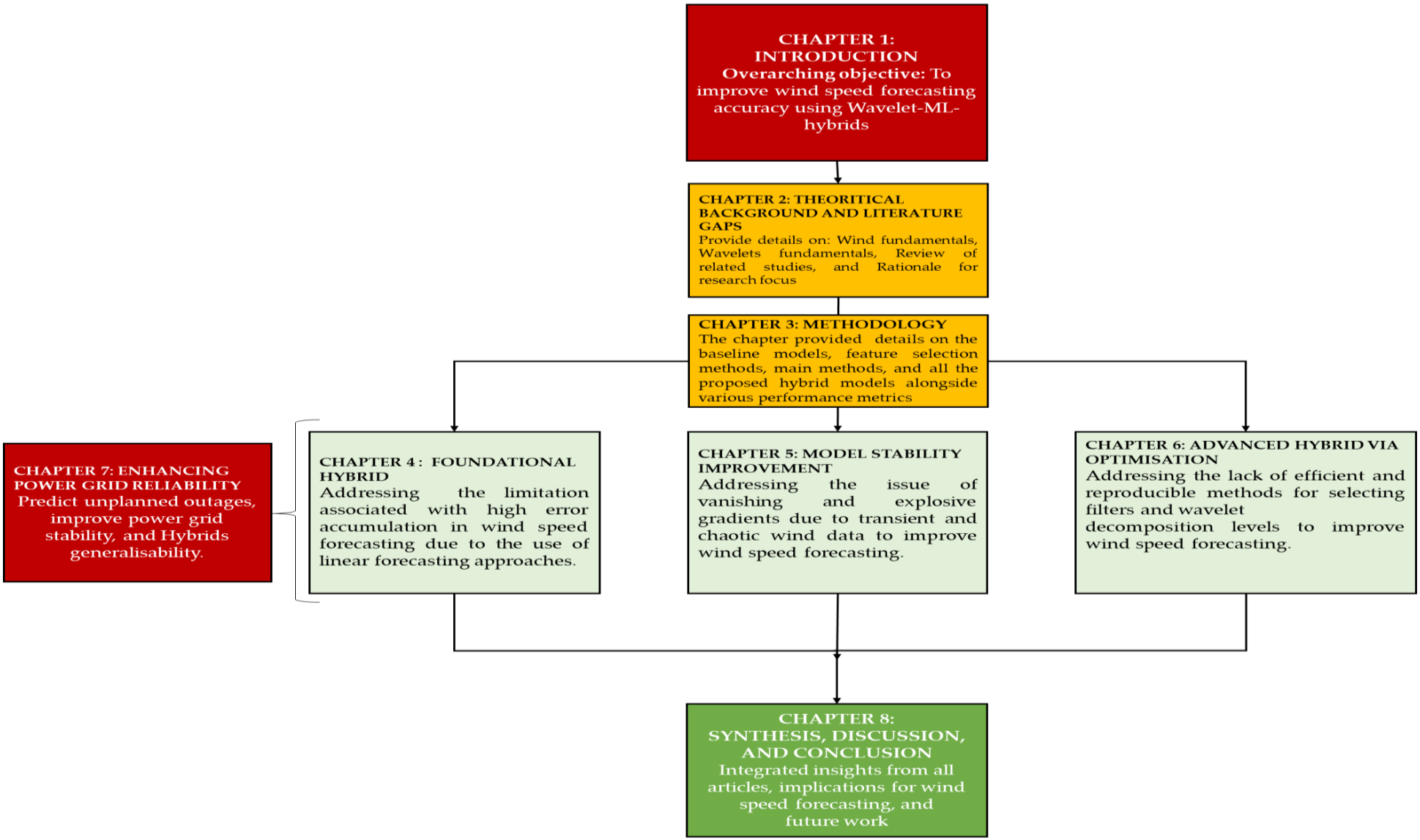


Figure 1. 3. Process flow of the thesis

This page is intentionally blank

Chapter 2

Theoretical Background and Literature Gaps

2.1 Wind Fundamentals

2.1.1 Introduction

By 2050, the electricity demand is anticipated to double [1]. Due to high dependency on fossil fuels (e.g. coal, oil, etc.) for electricity production, these non-renewable resources have depleted at an alarming rate. As a result, the demand for renewable energy is increasing, which brings wind power into the spotlight as a viable alternative to fossil fuels. A prime advantage of wind power is that it does not emit harmful wastes such as GHGs [2–4]. In addition to addressing various environmental challenges and being cost-effective, wind is abundant and can be harvested efficiently (with proper tools) to generate large volumes of wind power to diversify the electricity power grid. With the potential to harvest 1.266 billion MW of available wind energy, which could meet worldwide energy demand, wind power is becoming an increasingly important source of electricity [32]. Though wind is an abundant renewable energy resource, exploiting this physical quantity is a highly complicated exercise due to its inherent reliance on location and climate factors.

Despite the aforementioned benefits, wind energy still faces several challenges that minimises its generation and applications. Environmental challenges include, but are not limited to, wind turbines blocking bird migration routes, which can result in the harming of birds. Furthermore, as wind power plants development and construction continue to grow due to increasing adoption, the noise and visual pollution associated with this technology negatively impacts neighbouring communities. In terms of storage solutions, there are limited cost-effective solutions which is dent on the effective application or usage of wind power. Although wind turbine operational costs are relatively low, the high upfront costs remain a significant economic barrier (see [32] for details).

2.1.2 Wind Formation

As a physical quantity, wind is a by-product of deviations in atmospheric pressure [33]. On Earth, solar radiation from the sun is absorbed in different forms and the convection processes the movement of particles from an area with high heat to one of lower heat levels leads to unequal heat distribution, which in turn converts into air motion. This air motion ensures that equilibrium is maintained in the atmosphere by air moving from high-pressure to low-pressure locations [32]. Accordingly, wind speed is proportional to the speed of airflow, which is directly proportional to wind energy. Hence, wind speed (ultimately wind power) is higher where the atmospheric pressure gradient is also higher. Wind formation and speed are mostly affected by Solar radiation, Coriolis effect, and Topography (see [32] for details).

- **Solar Heating effect**
  The Equator receives most of the solar radiation per unit area. This is because the surface of the Earth is parallel to the solar radiation and perpendicular to the path of the Sun. This orientation results in temperature and gradient mismatch such that cold air from the poles will move to the equator, whilst hot air from the equator moves to the poles. Furthermore, as the earth rotates around its axis (at 23.5 degrees) solar radiation is received differently at different times of the day resulting in uneven heating and variations in weather patterns.

- **Coriolis effect**
  The Coriolis effect, which has to do with the deflection of the object from the surface of the earth to the atmosphere, is pivotal for the characteristics of wind speed (particularly its direction). In essence, this phenomenon, which is stronger at the poles than at the Equator, is such that it deflects wind to different directions depending on the hemisphere (i.e. Northern (to the right) and Southern (to the left)). The degree to which the wind is deflected relies on the earth's latitude and wind speed, with weaker winds being deflected less than stronger winds.

- **Topography effect**
  Earth's surface receives heat differently due to different factors, including the composition, absorption rate, and reflection of materials such as vegetation, rocks, etc. In fact, the topography of the earth's surface, including mountains, valleys and hills, also affects the solar radiation levels received. As a result,

different locations on Earth will experience different temperatures, even at the same altitude [32,33].

In summary, wind speed is often affected by obstacles and frictional drag, resulting in a process called wind shear[1]. Moderate wind speeds are best for wind power generation, whilst extreme wind speeds, such as hurricanes and tornadoes, can halt or limit wind power production [24]. For instance, high-speed winds can produce turbulence [2] and gusts [3] that can damage wind turbines. Thus, since wind speed is highly irregular, intermittent and variable on both a regional and time scale which cause the complexities of harvesting wind energy.

2.1.3 Fundamental Physical Concepts

Wind energy refers to the kinetic energy that is contained in airflow caused by convection on the surface of the earth. Kinetic energy is determined by the following expression (see [33] for details):

$$E_k = \frac{1}{2}\, mv^2, \tag{2.1}$$

where m and v, respectively denote the air mass and velocity of the air. A significant part of this quantity hinges on the volume, speed, and air mass [23,33]. Fundamentally, wind power is the flow of wind energy through an area during a unit rate of time. In essence, wind power is the rate of change of kinetic energy over a unit rate of time such that [33]

$$P_W = \frac{1}{2}\frac{dm}{dt}\, mv^2. \tag{2.2}$$

The mass flow rate (as the wind spins turbine's blades) is denoted by:

$$\frac{dm}{dt} = \rho A v. \tag{2.3}$$

Consequently, the mass flow rate is a function of the volume flow rate (Av), which is a function of area A under consideration over time t, and the density of the flow rate (ρ). Substituting Equation (2.3) in Equation (2.2) yields the total wind power denoted by [33]:

---

[1] Wind shear/gradient is a short-distance change in wind speed or/and direction in the atmosphere [34]

[2] Wind turbulence is a variation in wind speed with a small time interval. The turbulence intensity (T_I), which plays a paramount role when selecting a location for the wind turbines, is denoted by $T_I = \frac{\sigma_v}{\bar{v}}$ with $\bar{v}$ being the mean wind speed and the corresponding standard deviation given by σ_v [32]

[3] Wind gust, a type of turbulence, is associated with unexpected upsurge in wind speed within a short time scale [35]

$$P_W = \frac{1}{2}\,\rho A v v^2 = \frac{1}{2}\,\rho A v^3, \tag{2.4}$$

from Equation (2.4), it is apparent that wind power is proportional to the cube of the wind speed variable v (in ms^{-1}). In fact, the main influencing component is the wind speed variable to the extent that wind power will octuple when the wind speed doubles [24]. The density ρ (in $kg.m^{-3}$), which is proportional (nonlinearly) to wind power and highly dependent on the local temperature T (in K) and air pressure (denoted by P_{air}) is determined by:

$$\rho = \frac{P_{air}}{RT}, \tag{2.5}$$

with $R = 287\frac{J}{kg.K}$ being the gas constant. Based on Equation (2.5), the air density is reduced nonlinearly as the altitude rises. The nonlinearity is derived from the fact that ρ can be reduced further to [32]:

$$\rho = \rho_0\left(1 + \frac{T}{T_0}\right)^{\left\{\frac{-g}{RL}+1\right\}}, \tag{2.6}$$

where ρ_0 and T_0 respectively depict air density close to the ground and temperature at the ground level. Acceleration gravity is denoted by g, the temperature lapse rate by L, and the gas constant by R. The temperature variable has the highest positive influence on air density [36] and a negative influence on wind power output. On the other hand, the wind turbine swept area A (in m^2) (denoted by Equation (2.7)) is quadratically related to wind power, indicating the fact that turbines with longer blades are better (in terms of wind power generation) as they cover larger swept areas.

$$A = \pi[(l_s + r_s)^2 - {r_s}^2]. \tag{2.7}$$

In Equation (2.7), the length of the turbine blade is denoted by l_s while the radius of the hub is denoted by r_s. In essence, a wind turbine blade that is twice as long can sweep up to four times more area than a shorter blade.

2.1.4 Wind Power Fundamentals

2.1.4.1 Limits and Efficiency

Despite the inextinguishable nature of wind, turbine systems cannot fully extract all the available energy due to the Lanchester-Betz limit denoted by [37]:

$$C_p = \frac{P_{WT}}{P_W} \leq 59\%, \tag{2.8}$$

with power generated by the wind turbine P_{WT} denoted by:

$$P_{WT} = \frac{1}{2}\,\rho A C_p v^3, \tag{2.9}$$

where $P_W \geq P_{WT}$ always. Thus, a wind turbine located at the earth's location can only extract at most 59% of the available wind power. The justification of the Lanchester-Betz limit is that if all energy can be perfectly extracted, there will be no wind, hence, no wind speed, such that the turbine will eventually shutdown. Aerodynamic losses (mainly due to wake effects, blade tip, etc.) reduces turbine's efficiency to between 30% and 45% in practice, which is more substantial than the theoretical upper limit [37].

Hence, the conventional efficiency of wind turbines is heavily dependent on the efficiency of the gearbox (η_a), generator (η_b), and electrical efficiency (η_c). Generator efficiency is mostly influenced by mechanical losses; gearbox efficiency by load/no-load power losses; and electric efficiency by losses at various points in the electrical system such as converters, switches, controls, and cables [23,32,38,39]. The conventional efficiency is computed by the equation below:

$$\eta_{tot} = C_p \eta_a \eta_b \eta_c. \tag{2.10}$$

Multiplying Equation (2.10) by P_W yields the following result

$$P_W \eta_{tot} = P_W C_p \eta_a \eta_b \eta_c, \tag{2.11}$$

such that the effective power (P_e) representing the wind power output that can be integrated into the electrical power grid is given by:

$$P_e = \tfrac{1}{2}\,\rho A v^3 C_p \eta_{tot} = P_W \eta_{tot}. \tag{2.12}$$

Another measure of wind power efficiency is the capacity factor (C_F) (as shown in Equation (2.13)) of a wind turbine, which is the amount of energy a particular turbine can produce.

$$C_F = \frac{E_{actual}}{E_{ideal}} = \frac{t\bar{P}}{t\bar{P}_N}, \tag{2.13}$$

where t denotes time, the mean power is represented by $\bar{P}$, and P_N is the nominal power of the turbine. Wind turbine blades capture mechanical energy that needs to be converted into electrical energy by wind generators.

2.1.4.2 Wind Power Density

Wind power density (WPD) (watts per square meter (Wm^{-2})) is a metric that quantifies wind energy availability at a particular location. Wind power potential is determined by wind speed and power density levels such that locations characterised

by limited wind ($< 6\ ms^{-1}$) also have low WPD ($< 200\ Wm^{-2}$)), whilst the contrary is true for geographical locations with adequate or sufficient wind resources (wind speed $> 9\ ms^{-1}$and WPD$> 200\ Wm^{-2}$) [33] (see Table 2.1 below).

Table 2. 1. Description of wind power density (at 50 m height)

Category	Resource potential	WPD (Wm^{-2})	Wind speed (ms^{-1})
1.	Very low	< 200	0.0 - 5.9
2.	Low	200 - 300	5.9 - 6.7
3.	Moderate	300 - 400	6.7 - 7.4
4.	High	500 - 600	7.4 - 7.9
5.	Very high	500 - 600	7.9 - 8.4
6.	Excellent	600 - 800	8.4 - 9.3
7.	Outstanding	> 800	> 9.3

2.1.4.3 Wind Turbine Controls

A typical wind turbine (Horizontal Axis) [4] consists of a tower (often made of steel) and an upper nacelle (that can rotate 360 degree and is situated on the upper part of the tower)[5]. In the nacelle, there is a generator, whose main function is to provide power to the grid. The gear train in the hub links the generator to the blades such that the generator is spun (by the turbine at a certain speed) to produce wind power. The yaw control system directs the turbine in the direction of incoming wind. As the wind strikes the blades, the hub rotates. The pitch system controls the pitch of the blades, which ultimately determines how much lift (or drag) they generate [32,39,40]. A gearbox connects a rotor shaft to a generator shaft. And the generator's power is fed to the grid through a transformer. Some of the parameters with great influence on the performance of the wind turbine are discussed below (see e.g., [32,39-45] for details).

Pitch Control

The adjustment of blade pitch angle, known as pitch control, is an essential part of wind turbine systems. This process controls the forces acting on the blades, which directly influence the wind power output. By rotating the turbine on its longitudinal axis, the pitch angle (inversely proportional to the power output) can be altered to

[4] Wind turbines extract kinetic energy from wind and convert it into usable electricity. These instruments can be operationalised onshore (land location) or offshore (sea location) to generate power and can be divided into horizontal-axis and vertical-axis turbines. The earlier (which often feature three blades) is the most common turbines, whilst the latter is usually equipped with vertical rotor shafts.

[5] The nacelle, which is also the heart of the entire turbine, is also equipped with a cooling system to help regulate heat from the gearbox and generator

slow down the blades when required. For instance, in feathering mode, the pitch angle is utilised to protect the turbine from destructive winds by preventing the blades from catching any wind. The pitch control does not only improves stability and efficiency by managing blade pitch angle but also acts as a protection tool against high wind speeds (see e.g., [32,39,40,42,45] for details).

Stall Control

Stall control is a procedure employed to regulate and protect wind turbines and it involves stalling the blades once the rated or optimal speed has been reached. There are two types of stall control, viz.; passive and active control. Passive stalling (reserved for small turbines) protects the turbine from the wind by stalling the blades to limit the rotor speed and power output of the turbine. On the other hand, an active stall-controlled turbine employs a negative pitch angle to regulate power output. Passive stall control is simpler and more cost effective than a pitch controlling and active stalling. Though pricier and reserved for much larger turbines, active stalling is offers more reliability and accuracy for wind power regulation (see e.g., [32,39,40,45] for details).

Yaw Control

Wind turbines use a yaw control mechanism (which contains a motor and drive) to spin the nacelle and blades towards the direction of the wind, thereby ensuring maximum wind energy output. However, turning the nacelle can create cable twists inside the tower, leading to damage if the direction of the wind shifts and the nacelle continues spinning. Accordingly, there is a reader used to monitor cable twisting. This reader also alerts the controller to stop cable motion and to make it straight, thereby preventing damage [32,39,40,46].

2.1.4.4 Wind Turbine Aerodynamics

Tip-Speed Ratio

One of the most important parameters in wind turbine generator design is the tip speed ratio which is calculated as the ratio of wind speed and the velocity of the rotor tip and is given by [47,48]:

$$\kappa = \frac{v^*}{v} = \frac{\omega r_s}{v}, \tag{2.14}$$

where v^*denotes the rotor tip speed, v the wind speed, $\omega = 2\pi f$ represent angular velocity, r_s the radius of the hub, and f the rotation frequency. If the angular speed of

the blade is very low, the wind will pass through most of the area that the blades cover without allowing it to effectively catch wind energy. On the other hand, if the angular speed is very high, the blades spinning very fast will most likely obstruct the wind, resulting in less wind power produced. Therefore, determining the optimal tip speed is crucial for maximising the efficiency of wind turbines (see [47,48] for details).

Wake effects

Wake effects occur when two or more turbines are located close to each other along the path of the incoming wind. Wind turbines generate electricity by extracting energy from the incoming wind. As a result, the wind leaving the turbine, known as the "wake" and characterised by increased turbulence [41], will have less energy content. Ultimately, the wake will start to spread out as it moves downstream and gradually return to free-stream conditions. When the wake from one wind turbine reaches the swept area of another leeward turbine, the leeward turbine is considered to be affected by the wake. The efficiency of a wind farm can be significantly diminished by the wake effect. Additionally, the impact of the wake is influenced by the pitch and yaw control angles of the turbines [32,39,42].

Wind shear

Wind shear is highly influenced by a variety of factors, including terrain type, elevation, time of day and night, and season. It tends to be lower during the day and higher at night [32,34]. Wind shear is calculated by the Hellmann power equation which refers to the speed of wind at different heights [34,49].

$$v_\tau = v_{\tau_0}\left(\frac{\tau}{\tau_0}\right)^\gamma, \tag{2.15}$$

where τ denotes the height above the earth's surface associated with v_τ, τ_0 the reference height for which wind speed v_{τ_0} is known, and γ is the wind shear coefficient.

2.1.4.5 Wind Turbine Performance

The power curve is a vital tool to model and evaluate wind turbine performance. An example of a power curve for a pitch-regulated wind turbine is shown in Figure 2.1. There are four distinct regions that forms the power curve [23,24,43]. When the wind speed falls within the range of 0 ms^{-1} to 3 ms^{-1} (or Region A), the wind power output is immaterial or zero. In this range, the wind lacks sufficient kinetic energy to rotate

the wind turbine and produce electrical power. The second segment (Region B) of the curve (3 ms^{-1} to 15 ms^{-1}) represents the steep incline just above the cut-in speed and the rated speed (between 12 $ms^{-1}$to 15 ms^{-1}). In this segment, the wind turbine operates within standard parameters, resulting in a rapid increase in wind power output. In Region C, also referred to as the nameplate capacity (where wind speed range from 15 ms^{-1} to 22 ms^{-1}), the turbine's performance is restrained, and the total wind power output remains at an optimal and constant level. In the fourth segment (or Region D), known as the shutdown region (where wind speed exceed 22 ms^{-1}), the turbine is protected from high-speed winds (mostly through shutdown procedures). Thus, the turbine does not produce power in this region and beyond (see [23,24,43] for more details).

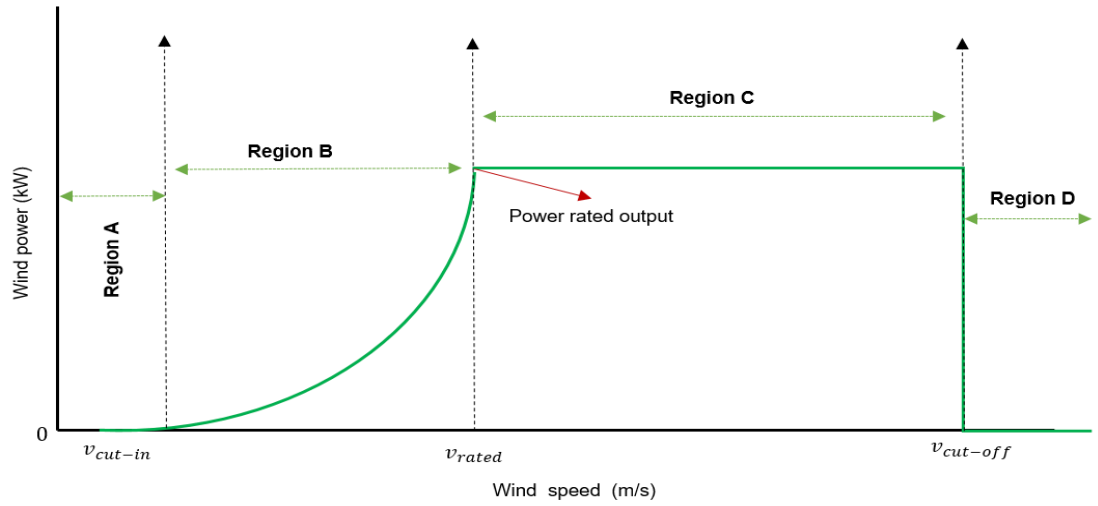


Figure 2. 1. Relationship between wind speed and wind power. The power curve plays a significant role in determining the average power output of a wind turbine needed for the wind turbine sizing and cost optimisation study, optimal turbine-site match, and the ranking of potential sites. Further, wind turbine power curve models estimate the capacity factor of a wind turbine [23,24,32,43].

It can be observed that the steepest segment (Region B) (3 ms^{-1} to 15 ms^{-1}) of the power curve can be significantly impacted by even small errors in wind speed prediction. Therefore, the unpredictability associated with estimating wind power output is directly related to the steepness of the power curve and the deviation from the fundamental forecast of wind speed. Besides, the deterministic relation of the typical power curve depicted in Figure 4 differs from the power curve observed in practice. This can be attributed to the fact that the curve was derived from testing a single turbine in an ideal environment (usually characterised by constant wind flow, free of obstacles and obstructions, no turbulence, and normal air pressure). In practice, wind is inherently unpredictable and random due to its heavy reliance on climatic factors.

2.1.5 Remarks

Wind speed depends on a variety of factors, such as atmospheric pressure changes, topography changes, seasonal changes, elevation above ground level, weather patterns, and land formations. As a result, wind speed is irregular and variable on both a regional and time scale. Essentially, the irregularity and chaotic feature of wind speed is due to the uneven spread of solar radiation (and wind) across the Earth's surface. When wind speed suddenly increases, causing wind turbulence and wind shear, efficient strategies (based on accurate and reliable wind speed forecasting) are pivotal to correct wind power imbalances and maintain stable power output from the wind turbine system. Besides, highly robust, accurate and reliable wind speed estimates based on appropriate modelling and forecasting approaches are also important for ensuring the safe functioning of wind farms (see e.g., [23]). Pivotal, wind speed prediction strategies are site-specific and subject to forecast horizon variation; model selection is a complicated and time-consuming task.

2.2 Wavelets Fundamentals

2.2.1 History of Wavelets

The history of wavelets dates back to the early 1900s when the concept of Haar families was introduced by Alfred Haar (the Hungarian mathematician). After 70 years, specifically in the early 1980s, the late Jean Morlet alongside Alex Grossmann introduced the term "wavelet" in the context of geophysics. The main idea behind the concept of a wavelet was to provide or develop a function that could effectively decompose signals into varying frequency components. Nonetheless, Morlet's contribution to the field dates back to the 1960s when he developed the continuous wavelet transform (CWT) to improve the Gabor transform (i.e. a type of STFTs) developed by Dennis Gabor in the early 1940s). Improving on the work of Morlet by handling redundancy in the selection of basis functions, Yves Meyer (a French Mathematician), proposed the concept of a wavelet localisation around 1985 through the development of the orthogonal wavelet basis functions with excellent time-frequency localisation properties. By 1986, Stephane Mallat and Meyer had introduced the concept of MRA for DWT. The idea involved filters (low and high pass) to decompose a signal into dyadic bands. In the late 1980s (specifically 1988), Ingrid Daubechies who had already worked with Mallat in 1986 on the discretisation of time and scale parameters of the WTs, later laid a foundation of compactly supported wavelets, the foundations of modern wavelet theory. In 1989, Mallat developed the fast wavelet transform (FWT). This decomposition and reconstruction algorithm relies on the concept of quadrature mirror filters (see [50] for more details).

2.2.2 Preliminary Concepts

2.2.2.1 Fourier Transform

In Fourier Transform (FT) (invented by Joseph Fourier in the early 1800s), signals are decomposed in components and presented as a sum of cosine functions. In essence, a signal is written as addition of its amplitude, frequency, and phase components, making these approaches effective at quantifying frequency content from fixed signals. To unmask inherent characteristics of a signal that are not visible in the time domain, FT (as a spectral approach) transforms signals from time domain to frequency domain representing the magnitude of frequencies (also see e.g., [51–56]). The continuous FT (CFT) of a function $Y(t)$ is given by (also see [52-54]):

$$Y(\epsilon) = \int_{-\infty}^{+\infty} y(t)\, e^{-i\epsilon t} dt, \tag{2.16}$$

where ϵ is the angular frequency. The inverse of the equation above is denoted by the following expression:

$$y(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} Y(\epsilon)\, e^{i\epsilon t} d\epsilon. \tag{2.17}$$

The CFT must satisfy the following integrability condition

$$\int_{-\infty}^{+\infty} y(t)\, dt < \infty. \tag{2.18}$$

In the Equations (2.16)-(2.17), $Y(\epsilon)$ is pivotal in computing the amplitude of $e^{i\epsilon}$ in the signal $y(t)$. The CFT utilises the equation below to handle discrete domain need for signal information in time

$$z_n = \frac{1}{T} \int_0^T y(t)\, e^{-in\epsilon_0 t} dt, \tag{2.19}$$

where ϵ_0 and z_n respectively denotes the fundamental (i.e. the smallest frequency of a sinusoid) frequency and n^{th} harmonic frequency. The corresponding inverse is then presented by the following mathematical expression:

$$y(t) = \sum_{n=-\infty}^{+\infty} z_n\, e^{i\epsilon_0 t}. \tag{2.20}$$

The aforementioned mathematical expressions are pivotal for the derivation of wavelets [52]. The discrete FT (DFT) of $Y(n)$ for N periodic signal $y(k)$ is denoted by:

$$Y(n) = \sum_{k=0}^{N-1} y(k)\, e^{\frac{-i2\pi kn}{N}}, n \in [0, N-1], \tag{2.21}$$

with the respective inverse denoted by the expression below:

$$y(k) = \frac{1}{N} \sum_{k=0}^{N-1} Y(n)\, e^{\frac{i2\pi kn}{N}}, k \in [0, N-1], \tag{2.22}$$

where $\frac{2\pi k}{N}$ are the harmonic frequencies. From the result above, the DFT maps the function $Y(n)$ from the time space to the frequency space $y(k)$ such that the signal ends in the complex domain. Furthermore, its basis functions $e^{\frac{i2\pi kn}{N}}$ provide a two-dimensional matrix which is of paramount importance when dealing with signal processing. In practice, the DFTs are associated with higher computational costs particularly when dealing with larger sample sizes (N) (see e.g., [53,54] for more details). As such, the Fast Fourier Transform (FFT) which is known to be computationally efficient and much more practical is often preferred (see Algorithm 2.1). Consider an $N = 2^n$ point DFT such that the signal $y(k)$ can be divided

into $y_a(k) = y(2k)$ (even) and $y_b(k) = y(2k + 1)$ (odd) with same length $\frac{N}{2}$. Then an FFT is given by:

$$Y(n) = \sum_{k=0}^{N-1} y(k)\, e^{\frac{-i2\pi kn}{N}},$$

$$= \sum_{k=0}^{\frac{N}{2}-1} y(2k)\, e^{\frac{-i2\pi(2k)n}{N}} + \sum_{k=0}^{\frac{N}{2}-1} y(2k+1)\, e^{\frac{-i2\pi(2k+1)n}{N}},$$

$$= \sum_{k=0}^{\frac{N}{2}-1} y(2k)\, e^{\frac{-i2\pi kn}{N/2}} + e^{\frac{-i2\pi n}{N/2}} \sum_{k=0}^{\frac{N}{2}-1} y(2k+1)\, e^{\frac{-i2\pi kn}{N/2}}. \quad (2.23)$$

The efficiency of the FFTs stems from the fact that these algorithms compute the $Nlog_2N$ operations to satisfy $N = 2^n$ point requirement [52]. Besides their efficiency, the approaches are amenable to nonstationarity or transient characteristics (mostly inherent in wind data). This can be attributed to the fact that the time-averaged amplitude spectrum of the DFTs (including the FFTs) is often not adequate as the spectral content is not localised in time [53,54]. Hence, STFT was introduced to remedy this (to some extent). Also see Algorithm 2.1 below.

Algorithm 2.1: FFT Algorithm

1. Deconstruct the DFT of size *N* into the addition of two DFTs of length $N/2$; one odd and the other even.
2. Again deconstruct each of these DFTs into the addition of four DFTs with length $N/4$.
3. Continue similarly until such time that the DFT calculation has been reduced to the calculation of DFTs of length 1.

2.2.2.2 Short Time Fourier Transform

The STFT (also referred to as Windowed Fourier Transform) extracts a small part of the signal f and then determines the frequency $f(x)$ using the FT at time $x = t$ [52]. In essence, STFT (as a type of FT of a windowed signal), imposes windows on the signals during segmented analysis to curtail nonstationarity [52]. A continuous STFT of f with reference to g is denoted by:

$$S_g\, f(t,\epsilon) = \int_{-\infty}^{+\infty} f(x)g(x - t)e^{-ix\epsilon}\, dx, t, \epsilon \in \mathbb{R},$$

$$= \int_{-\infty}^{+\infty} f(x)g_{t\epsilon}\, dx, t, \epsilon \in \mathbb{R}, \quad (2.24)$$

whilst the STFT discrete version is denoted by:

$$S_g\, f(t,\epsilon) = \sum_{-\infty}^{+\infty} f(x) g(x - t) e^{-ix\epsilon}, \tag{2.25}$$

with $g(x - t)e^{-ix\epsilon} \neq 0$ being a fixed, real and symmetric sliding window function. In the equation above, $S_g\, f(t,\epsilon)$ measures the frequency content of the signal f concerning a time point t and frequency ϵ by partitioning the time domain into equally spaced windows that are fixed in the time-frequency domain. However, STFTs being time localised, provide frequency information for instances where frequency components of the signal change over time, whilst the standard FTs only provide frequency information averaged across the entire time interval of the signal. It is pivotal to note that both time and frequency resolution cannot be accomplished simultaneously as alluded to by the uncertainty principle[6] and this drawback is efficiently and effectively remedied through the applications of wavelets (see e.g. [55,57–59]). The function g is said to be time-frequency localised if there exist $\sigma_x(g)$ such that (see e.g., [52,59])

$$\begin{aligned} {\sigma_x}^2(g) &= \int_{-\infty}^{+\infty} (x - t)^2 |g_{t,\epsilon}(x)|^2\, dx, \\ &= \int_{-\infty}^{+\infty} (x)^2 |g(x)|^2\, dx, \end{aligned} \tag{2.26}$$

with the frequency spread $\sigma_w(\hat{g})$ given by the following equation

$$\begin{aligned} {\sigma_w}^2(\hat{g}) &= \textstyle\int_{-\infty}^{+\infty} (\omega - \epsilon)^2 |g_{t,\epsilon}(\omega)|^2 d\omega, \\ &= \frac{1}{2\pi} \int_{-\infty}^{+\infty} (\omega)^2 |g(\omega)|^2 d\omega. \end{aligned} \tag{2.27}$$

2.2.2.3 z-Transformation

Pivotal for designing and analysis of signals, the z- transform (which generalises FTs) provide an efficient and effective approach to handling and manipulating mathematical and algebraic expressions. For instance, the z-transform of the function $y(n)$ is denoted by:

[6] Uncertainty Principle or the Heisenberg principle established by Heisenberg for problems encountered in quantum mechanics says that: "The position and the momentum of an electron within an atom cannot be both determined explicitly but only in a probabilistic sense under a certain "uncertainty". Applied to the FT this principle implies that the product of the duration and bandwidth of a signal $f(t)$ has a lower bound. Thus, the FTs cannot achieve both time and frequency resolution at the same instance (see e.g. [53] for more details and proofs).

$$Y(z) = \sum_{n=-\infty}^{+\infty} y(n)z^{-n}\,, z \in \mathbb{C}. \tag{2.28}$$

The results presented above are valid for z-values that converge, since the z-transform is an infinite series [56,60–62] (also see Table 2.2 below). Another convenient approach is so called ”polyphase filtering”, which involves splitting a signal or filter into its polyphase components. This approach is vital for multirate signal processing. Consider a filter given by Equation (2.28) such that when decimated by the factor of $N = 2$, it can be divided into even and odd coefficients components so that

$$\begin{aligned} Y(z) &= \sum_{n=-\infty}^{+\infty} y(2n)z^{-2n} + \sum_{n=-\infty}^{+\infty} y(2n+1)z^{-(2n+1)}\,, \\ &= \sum_{n=-\infty}^{+\infty} y(2n)z^{-2n} + z^{-1} \sum_{n=-\infty}^{+\infty} y(2n+1)z^{-2n}\,. \end{aligned} \tag{2.29}$$

Setting

$$\overline{\omega_0}(z) = \sum_{n=-\infty}^{+\infty} y(2n)z^{-n}\,, \tag{2.30}$$

and

$$\overline{\omega_1}(z) = \sum_{n=-\infty}^{+\infty} y(2n+1)z^{-n}\,, \tag{2.31}$$

where $\overline{\omega_0}(z)$ and $\overline{\omega_1}(z)$ denote polyphase components and $Y(z)$ can be expressed as follows (also see Table 2.2):

$$Y(z) = \overline{\omega_0}(z^2) + z^{-1}\overline{\omega_1}(z^2). \tag{2.32}$$

An extension for the decimation by any factor $M \in \mathbb{R}$ yields the following mathematical expression

$$\begin{aligned} Y(z) = \sum_{n=-\infty}^{+\infty} & y(Mn)z^{-Mn} \\ & + z^{-1} \sum_{n=-\infty}^{+\infty} y(Mn+1)z^{-(Mn)} \\ & + z^{-(M-1)} \sum_{n=-\infty}^{+\infty} y(Mn+M-1)z^{-(Mn)}\,. \end{aligned} \tag{2.33}$$

Hence, a polyphase equation (which ensures timely implementation of filter banks is denoted by (see e.g., [56,62])

$$Y(z) = \sum_{i=0}^{M-1} \overline{\omega_i}(z^M) z^{-i}, \tag{2.34}$$

with the subspace filter given by:

$$\overline{\omega_i}(z) = \sum_{n=-\infty}^{\infty} v_i(n) z^{-n}, i \in [0, N-1]. \tag{2.35}$$

Table 2.2. z-transform features

Property	**Time Domain**	**z-Domain**
Linearity	$\alpha h(n) + \beta g(n)$	$\alpha H(z) + \beta G(z)$
Time Shifting	$h(n - n_0)$	$z^{-n_0} H(z)$
Time Reversal	$h(-n)$	$H(z^{-1})$
Convolution	$h(n)g(n)$	$H(z)G(z)$

2.2.3 Filter Banks

Primarily, filter banks (i.e. a group of bandpass filters) extract or expose or enhance certain frequency components or bands embedded in a signal. In essence, filters suppress certain frequency bands of a signal (while others pass through) such that the output signal is the original signal minus these suppressed components (also see Figure 2.2 below). The filters facilitate signal decomposition and reconciliation, provides reliable frequency content extraction thereby enhancing analysis (see [51,59,62]). Hence, a finite impulse response (FIR) filter (also referred to as the discrete convolution summation) is given by [55,61–63]:

$$y(n) = \sum_{i} \sum_{k=0}^{N-1} f(k) h(n - k), \tag{2.36}$$

where $h(n)$ and $y(n)$ respectively denote the input and output signal, and $f(k)$ the FIR.

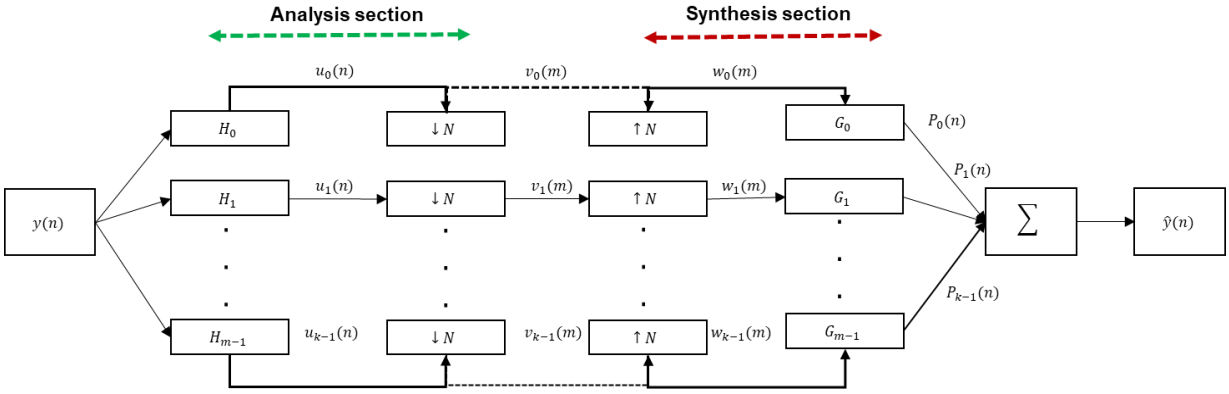


Figure 2. 2. Multiple channel digital filter bank

Distinguished by their functions, filters are commonly divided into two main types, viz.; analysis bank (responsible for decomposing input signals through downsamplers (i.e. deciminators)), and synthesis bank (which recomposes them using the decomposed signals through upsamplers (i.e. expanders)) as shown in Figure 2.2 (also see e.g., [56,62]). Though linear, both these sampling rate alteration operators (i.e. downsampler and upsampler) can be time-invariant [56,62]. Filter banks are fundamental to the successful application of wavelets as they ensure decomposition of the signal into different subseries with minimal complexity (and more statistically sound and predictable properties) than the original wind speed data.

2.2.3.1 Analysis Component and Downsampling

An analysis bank, $y(n)$ (denoting an original broadband signal) is partitioned into individual components with varying and distinct subbands to facilitate the excavation of specific frequency attributes of the input signal and assess them separately or individually. With the application of a bandpass filter the signal is decimated at a sample rate proportional to the reduced bandwidth (also see Figure 2.2). The first region (or branch) in the filter bank contains a low-pass filter denoted by $H_0(z)$. The modulated versions of this filter is denoted by $H_1(z), \ldots, H_{m-1}(z)$ (from low-pass to high-pass). The effect of each individual filter on $y(n)$ is such that $U_0(z) = Y(z)H_0(z), U_1(z) = Y(z)H_1(z), \ldots, U_{k-1}(z) = Y(z)H_{m-1}(z)$ [62–65]. The z-transformed $U_j(z)$ is the result of a convolution between $Y(z)$ and $H_j(z)$ (see [65] for details). Fundamentally, downsamplers reduce the sampling rate by some positive integer and this property facilitates computational needs within the filter bank [64]. Essentially, a downsampler decimates $y(n)$ (an input signal) by some factor N such

that only the N^{th} samples from $y(n)$ are preserved in the time domain (see [56,62,65] for details)

$$\hat{y}(n) = y(Nn), n \in \mathbb{Z}. \tag{2.37}$$

In the case where $N = 2$, the only even sample in the input signal are preserved by the deciminator. In the frequency domain we have

$$v_j(n) = \sum_i u_j(n) f(k) h(n - kN), k \in \mathbb{Z}, \tag{2.38}$$

with impulse $u_j(n)$ being the input sequence to the decimator. The $f(k)h(n-kN)$ represents the impulse response functions that are N samples apart and retain the kN^{th} samples. Fundamentally, the frequency is stretched by N such that

$$v_j(n) = \frac{1}{N} \sum_{k=0}^{N-1} u_j(n)\, e^{\frac{-i2\pi kn}{N}}, \tag{2.39}$$

with $n = 1$, implying that

$$v_j(n) = \frac{1}{N} \sum_{k=0}^{N-1} u_j(n)\, e^{\frac{-i2\pi k}{N}}. \tag{2.40}$$

Expressed in terms of z-transformation

$$\begin{aligned} V_j(z) &= \frac{1}{N} \sum_{k=0}^{N-1} U_j\left(z^{\frac{1}{N}} e^{\frac{-i2\pi k}{N}}\right), \\ &= \frac{1}{N} \sum_{k=0}^{N-1} U_j\left(z^{\frac{1}{N}} {W_N}^{-k}\right), z \in \mathbb{C}, \end{aligned} \tag{2.41}$$

where ${W_N}^{-k} = e^{\frac{i2\pi k}{N}}$. In this case, v_j (the downsampler outputs) are referred to as the polyphase signals. The main drawback of signal decimation is that the bandwidth is stretched by a factor of N, resulting in sample points that are not consistently reflective of the original input signal. This is attributed to the fact that the input frequency is greater than half the sample frequency [60,62,64]. To prevent aliasing, the sampling frequency should be $f_s^1 > 2f_{max}^1$ according to the Shannon sampling theorem[7]. In wind speed forecasting, the analysis component (responsible for downsampling) plays an important role since it ensures that the signal resolution is reduced (at various decomposition levels) in order to facilitate MRA. This supports reliable extraction of important features (i.e., trends, noise, etc.) from the wind speed signal.

[7] If we wish to recover an exact continuous-time signal with a maximum frequency of f_{max}, we must sample it periodically at a rate of $f_s^1 > 2f_{max}^1$. In this case $f_N^1 > 2f_{max}^1$ is called the Nyquist frequency, as it is the lower bound of f_s^1 [60,66]

2.2.3.2 Synthesis Component and Upsampling

The synthesis filter, which is essentially a parallel narrowband filter, connects the narrowband signals ($v_j(n),\ j \in [0, k-1]$) with one broadband output signal ($\hat{y}(n)$) [60,62,64,65] such that each broadband output signal passes through an upsampler into a high-pass filter. The effect of the narrowband filters ($G(n), j \in [0, m-1]$) on the respective input signal ($w_j(n),\ j \in [0, k-1]$) is such that $P_0(z) = W(z)G_0(z), P_1(z) = W(z)G_1(z), \ldots, P_{k-1}(z) = W(z)G_{m-1}(z)$. The upsampler stretches the input signal by factor N such that there is a padding of $N-1$ zeros between $y(n)$ and $y(n+1) \forall n$ (see e.g., [62,64,65] for details). In frequency domain, the input signal $y(n)$ will be stretched such that (see e.g., [56,62,65] for details)

$$\hat{y}(n) = \begin{cases} y\left(\frac{n}{N}\right), & N \text{ divides } n, \\ 0, & \text{otherwise}, \end{cases} \tag{2.42}$$

Upsampling by $N = 2$, every second adjacent sample in the given sequence will be pad with zero such that

$$w_j(n) = \frac{1}{N}\sum_{k=0}^{N-1} v_j(n)\, e^{\frac{-i2\pi k}{N}}. \tag{2.43}$$

Hence, the z-transform of the equation above is given by:

$$\begin{aligned} W_j(n) &= \frac{1}{N}\sum_{k=0}^{N-1} V_j\left(z^N e^{\frac{i2\pi k}{N}}\right), z \in \mathbb{C}, \\ &= \frac{1}{N}\sum_{k=0}^{N-1} V_j\left(z^N {W_N}^{-k}\right), z \in \mathbb{C}. \end{aligned} \tag{2.44}$$

Different from an analysis bank, the principal objective or role of a synthesis bank is to ensure perfect reconstruction of the original wind speed signal from decomposed subsignals.

2.2.3.3 Perfect Reconstruction

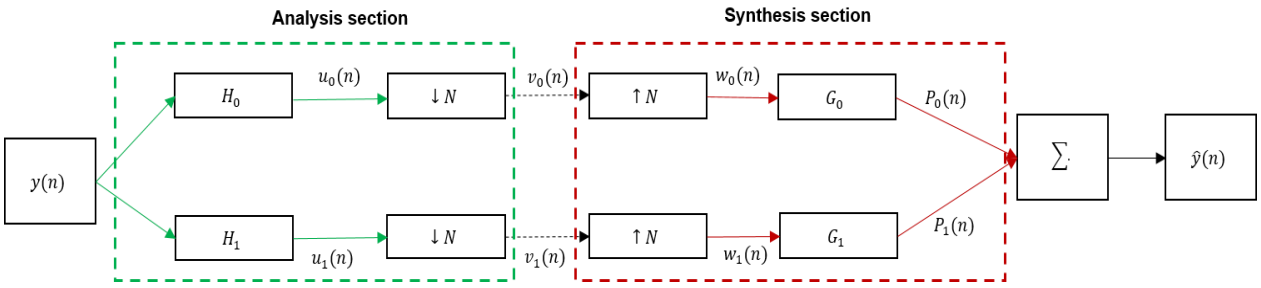


Figure 2.3. Two channel digital filter bank ($N = 2$)

A filter has been constructed perfectly if the original signal is equal to its reconstruction or its scaled version multiplied by some constant. Pivotal, the perfect reconstruction property eliminates the deficiency of sample aliasing due to either decimation or upsampling of a signal. Consider a two-channel filter bank (as depicted in Figure 2.3) with lowpass analysis filter $H_0(z)$, highpass analysis filter $H_1(z)$, lowpass synthesis filter $G_0(z)$, and highpass synthesis filter $G_1(z)$. Then, the resultant output of the analysis filter bank is given by (see e.g., [56,62,63,67] for details):

$$Y(z) = H_k k(z) Y(z), k \in [0, 1]. \tag{2.45}$$

After downsampling and then, upsampling the result above yields the following results (see e.g., [56,62,63,67] for details):

$$\hat{Y}(z) = \frac{1}{2}[H_0(z)G_0(z) + H_1(z)G_1(z)]Y(z) + \frac{1}{2}[H_0(-z)G_0(z) + H_1(-z)G_1(z)]Y(-z). \tag{2.46}$$

A perfect filter bank reconstruction requires that the conditions below be satisfied

$$H_0(-z)G_0(z) + H_1(-z)G_1(z) = 0, \tag{2.47}$$

$$H_0(z)G_0(z) + H_1(z)G_1(z) = 2z^{l_0}. \tag{2.48}$$

The solution to Equation (2.47) is given by the following polynomial terms $H_1(z) = G_0(-z)$ and $G_1(z) = -H_0(-z)$. The Equation (2.48) can be reduced to

$$V_0(z) - V_0(-z) = 2z^{l_0}, \tag{2.49}$$

through the product filter $V_0(z) = H_0(z)G_0(z)$. The Equation (2.49) play a pivotal role in perfect reconstruction [51,60,62]. It should be noted that the term z^{l_0} is odd such that the solution to Equation (2.49) extends to the following expressions:

$$h_0(T_0 - 1 - k) = (-1)^k g_1(k), \tag{2.50}$$

$$h_1(T_1 - 1 - k) = (-1)^k g_0(k). \tag{2.51}$$

with $T_i \in [0, 1]$ being the length of $h_i(n) \in [0, 1]$ so that the approximation and detailed component of the signal $y(n)$ is respectively denoted by:

$$p_0(k) = \sum_n y(n) g_0(2k - n), \tag{2.52}$$

$$p_1(k) = \sum_n y(n)h_1(2k-n), \tag{2.53}$$

$$\Longrightarrow \hat{y}(n) = \sum_n \{P_0(k)g_0(2k-n) + P_1(k)h_1(2k-n)\}. \tag{2.54}$$

Also see Algorithm 2.2 for summarised the steps to follow when designing filters [62,67].

Algorithm 2.2: Filter Design Algorithm [67]

1. Choose a polynomial $V_0(z) = H_0(z)G_0(z)$ that fulfils Equation (2.52).
2. Factor $V_0(z) = H_0(z)G_0(z)$ into two polynomials $H_0(z)$ and $G_0(z)$ that contains all zeros only at $z = -1$.
3. Choose $H_1(z) = G_0(-z)$ and $G_1(z) = -H_0(-z)$.

Maintaining perfect reconstruction of the original signal helps prevent distortion of the predicted subsignals which would compromise the accuracy of the final forecasting models.

2.2.3.4 Orthogonality

Orthogonality in filter banks facilitates energy preservation. Consider the two-channel filter bank that has been perfectly reconstructed, then reconstruction equation is denoted by (see e.g., [55,62,65] for details):

$$\begin{aligned}\widehat{Y_k}(z) = \frac{1}{2}[H_0(z)G_0(z) + H_1(z)G_1(z)]Y(z) \\ + \frac{1}{2}[H_0(-z)G_0(z) + H_1(-z)G_1(z)]Y(-z).\end{aligned} \tag{2.55}$$

In terms of matrix formation, the equation above can be rewritten as follows:

$$\widehat{Y_k}(z) = \frac{1}{2}\begin{bmatrix}G_0(z)\\G_1(z)\end{bmatrix}^T \begin{bmatrix}H_0(z) & H_0(-z)\\H_1(z) & H_1(-z)\end{bmatrix}\begin{bmatrix}Y(z)\\Y(-z)\end{bmatrix}, \tag{2.56}$$

with

$$\widetilde{\boldsymbol{H}_k}(z) = \begin{bmatrix}H_0(z) & H_0(-z)\\H_1(z) & H_1(-z)\end{bmatrix}, \tag{2.57}$$

being the alias component matrix such that $\widetilde{\boldsymbol{H}}_k(z) = \Delta\, \boldsymbol{H}_k{}^T(z)$ denotes the paraconjugate of $\boldsymbol{H}_k(z)$, and $\boldsymbol{I}$ the identity matrix. As a result, the filter bank will be orthogonal, and the alias component matrix is said to be lossless (see e.g., [60,62] for details) (also see Box 2.1).

Box 2.1. Lossless Property

A time variant and linear filter (say $H(z)$) is said to be lossless only if it preserves signal energy. Suppose $y(n)$ is an input signal corresponding to the output signal denoted by $\hat{y}(n)$, then $\hat{y}(n) = h * y(n)$ such that

$$\sum_{n=-\infty}^{\infty} |\hat{y}(n)|^2 = \sum_{n=-\infty}^{\infty} |y(n)|^2 .$$

Effecting the L_2 norm, the result above can be further expressed as follows

$$||\hat{y}(n)||_2^2 = ||y(n)||\,{}_2^2.$$

In essence, only stable filters can be lossless otherwise $||y(n)|| = \infty$.

2.2.4 Wavelet Transforms

A wavelet can be described as a collection of functions obtained from the prototype function by dilation and translation [55,57,59,60]. These functions have the capacity to alter their amplitudes and widths given a particular time frame. For instance a child wavelet denoted by $\psi_{s,h}(t) \in L_2(\mathbb{R})$ as result of a scaling (by $s \in \mathbb{R}\{\neq 0\}$) and time-shifting or translating mother wavelet $\psi(t)$ (by parameter $h \in \mathbb{Z}$) is given by [55,60]:

$$\psi_{s,h}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-h}{s}\right), \tag{2.58}$$

where scaling by $\sqrt{\frac{1}{s}}$ is critical for the independence of the wavelet $||\psi_{s,h}(t)||$ from h and s. Besides, normalising wavelets further mandates that $||\psi_{s,h}(t)|| = 1$. The result above assumes $\psi_{s,h}(t)$ satisfies the admissibility criteria:

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < \infty, \tag{2.59}$$

with $\Psi(\omega) = \int_{-\infty}^{\infty} \psi(t) e^{-i\omega t} dt$. The satisfaction of the admissibility condition implies that a wavelet is a normalised finite, short-lived, zero mean function such that

$$\Longrightarrow \int_{-\infty}^{\infty} \psi(t)\, dt = \Psi(0) = 0. \tag{2.60}$$

The trade-off characteristics of the time-frequency resolution can be achieved through wavelets which are appropriately defined in the MRA framework. The MRA framework also guarantees that wavelets have a better signal representation in terms of revealing transient behaviour and trends compared to their counterparts (e.g. FTs), which (to some degree) presents better statistical characteristics and is easier to model

[60,68,69]. Fundamentally, MRA is a collection of closed subspaces $M_j \subset L_2\mathbb{R}$ such that (see e.g., [55,60,68-71] for details):

a) $M_j \subset M_{j+1}, \forall j \in \mathbb{Z}$
b) $f(t) \in M_j \Leftrightarrow f(2t) \in M_{j+1}, \forall j \in \mathbb{Z}$
c) $f(t) \in M_j \Leftrightarrow f(t-k) \in M_j, \ \forall j,k \in \mathbb{Z}$
d) $\overline{\cup_j M_j} = L^2(\mathbb{R})$
e) $\cap_j M_j = \{0\}$
f) $\exists, \psi \in M_0$ such that $\{\psi(y - k)\}_{\{k \in \mathbb{Z}\}}$ is an orthonormal basis for M_0

MRA approach seeks to decompose a signal $y(t) \in L_2(\mathbb{R})$ such that $y_i \in M_j$ and the complementary subspaces $D_j = M_j - M_{j+1}$are such that M_j can be written as follows

$$M_j = M_j \otimes D_j. \tag{2.61}$$

Suppose the $\phi(t)$ and $\psi(t)$ are the scaling and wavelet functions $\ni \phi(t) \in M_1$ and $\psi(t) \in D_1$. If $\phi(t) \in M_1 \subset M_2$ such that $\phi(t)$ can be expressed in terms of $\phi(2t)$. Then, the dilation/scaling equation can be expressed by [55,59,60,68]:

$$\phi(t) = 2^{\frac{j}{2}} \sum_k h_\phi(k)\phi(2t - k), \tag{2.62}$$

with $h_\phi(k)$ denoting a lowpass filter coefficients. Furthermore, $M_2 \subset D_1$ implies that wavelet equation is given by:

$$\psi(t) = 2^{\frac{j}{2}} \sum_k g_\psi(k)\psi(2t - k), \tag{2.63}$$

where $g_\psi(k)$ are high-pass filter coefficients. The high pass and lowpass filters are interconnected by the expression below:

$$g_\psi(k) = (-1)^k h_\phi\big((N-1) - k\big), \tag{2.64}$$

where *N* denotes the sample size. For a bandwidth-constrained signal the function $y(t)$ can further be reduced to (see [55,60] for details)

$$y(n) = \sum_j \sum_k 2^{\frac{j}{2}} g_j(k)\psi(2^j t - k) + \sum_k 2^{\frac{j}{2}} h_j(k)\phi(2^j t - k), \tag{2.65}$$

where $g_j(k)$ and $h_j(k)$ respectively denote the detail and approximate coefficients.

2.2.4.1 Continuous Wavelet Transform

A CWT of a $y(t) \in L_2(\mathbb{R})$ is denoted by the inner product of the function and the child wavelets so that (see [55,60] for details)

$$C_f(s,h) = < y(t), \psi_{s,h} > = \int_{-\infty}^{+\infty} y(t) \frac{1}{\sqrt{s}} \psi^* \left(\frac{t-h}{s}\right) dt, t \in \mathbb{R}, \tag{2.66}$$

where $C_f(s,h)$ are the wavelet coefficients, ${\psi_{s,h}}^*(t)$ denotes the conjugate of the mother wavelet $\psi_{s,h}(t)$. The scaling (s) and translation (h) vary continuously in $\mathbb{R}\{\neq 0\} \times \mathbb{R}$.

The value $\frac{1}{\sqrt{s}}$ further ensures energy invariant (see [60]). Higher values of $\psi_{s,h}(t)$ are associated with longer duration and lower frequency, whilst smaller values of $\psi_{s,h}(t)$ are associated with shorter duration and high frequency. However, ψ cannot explain the entire spectrum (see e.g., [59,60,64]). The Morlet and Mexican hat are some of the well known CWTs (see e.g., [72,73]).

2.2.4.2 Discrete Wavelet Transform

The DWT of a signal $y(t)$ is determined by first discretising translation ($s = a^m$) and dilation ($h = a^m n$) such that (see e.g., [60])

$$\psi_{m,n}(t) = \frac{1}{\sqrt{a^m}}\psi\left(\frac{t-a^m n}{a^m}\right), m > 0, t, n \in \mathbb{Z}, \tag{2.67}$$

$$= a^{-\frac{m}{2}}\psi(a^{-m}t - n). \tag{2.68}$$

The discretisation in the result above is pivotal as it facilitates perfect reconstruction of wavelets. In practice, the function $\psi_{m,n}(t)$ is often computationally expensive. Nonetheless, $a = 2$ and $m = j$ provides a dyadic WT which enhances analysis through efficient decomposition of a signal into varying scales such that (see e.g., [55,60 ,64,70,71])

$$\psi_{j,n}(t) = 2^{\frac{-j}{2}}\psi\left(2^{-j}t - n\right), j, n \in \mathbb{Z}, \tag{2.69}$$

where j denotes the decomposition level and n is the time-shift or translation parameter. Despite improved efficiency, the scaling and translation function $\psi_{j,n}(t)$ still cannot cover the entire spectrum [55,57]. Hence, the introduction of the lowpass scaling functions donated by $\phi_{jo,n}(t) = 2^{\frac{-jo}{2}}(2^{-jo}t - n)$. Hence, the DWT of the signal $y(t)$ is given by [55,60]:

$$D_f(j,n) = \sum_{j=j_0}^{J} < y(t).\psi_{j,n}(t) > + < y(t), \phi_{j_o,n} >, j_o \le j \le J\,. \tag{2.70}$$

The portions $\sum_{j=j_0}^{J} < y(t).\psi_{j,n}(t) >$ and $< y(t).\phi_{jo,n} >$ respectively represent the high and low frequency content signals. The $D_f(j,n)$ is ideal for applications of wavelets in signal processing and can be reconstructed using the MRA through the inverse function (inverse DWT) (see [60] for details)

$$y(t) = \sum_{n}\sum_{j=j_0}^{J} < y(t).\psi_{j,n} > \hat{\psi}_{j,n} + \sum_{n} < y(t), \phi_{j_o,n} > \hat{\phi}_{j_o,n}, j_o \le j \le J. \tag{2.71}$$

The summarised process for DWT is presented in Algorithm 2.3. Common DWTs include but not limited to the Haar and Daubechies wavelets (also see [57,69] for details). DWT can be efficiently applied by using the FWT algorithm. The FWT

employs filter banks to deconstruct data into approximation and detail components. Details on FWT can be found in the work of [60,69].

Algorithm 2.3: The DWT Algorithm

Begin: Project the signal $y(t)$ on the subspace M_J with J being the sampling frequency.

1. Separate the approximation coefficient into detailed ($h(k)$) and approximate ($g(k)$) components.
2. Rescale the approximation coefficients.
3. Reprocess the approximation part to create a new approximation and detail component.
4. Repeat (2) and (3) until satisfactory results are obtained.

2.2.5 Remarks

Different from the WTs, FTs assume a stationary signal and they cannot reflect time series features in the time domain. As a result, FTs cannot adequately capture and explain non-stationary signals. Though, to some extent, STFTs solves this drawback by using localised shortwave to extract spectral features embedded in the nonstationary signals, these methods have problems associated with the use of fixed-width window function. Consequently, STFTs' application in the time-frequency domain has drawbacks. Hence, the current section delved into the fundamentals of wavelets which are more advantageous. This section begins with the detailed history on the origin of wavelets. Thereafter, a thorough discussion on the core building blocks of FT and STFT is provided alongside the concept of z-transformation which plays a critical role in handling algebraic expression when working with wavelets. Relative to wavelets, the strength and limitations of both FTs and STFT were also discussed (briefly). The section also discuss the concept of Filters, with specific emphasis on the Analysis and Synthesis components. The subsection of filters also delved into aspects of perfect reconstruction which plays a critical role when developing valid wavelets. In the last subsection, which also forms conclusion of Section 2.2, the study provided details in the concept of MRA before thoroughly discussing on the different types of wavelets; viz.; the CWT and DWT wavelets. The section provide significant details and adequate derivation of the aforementioned wavelet types as considering their characteristics, limitations, and strengths.

2.3 Wind Forecasting: Review

2.3.1 Introduction

2.3.1.1 Overview of Wind Forecasting

Although wind power is abundant in most parts of the world and can be produced at efficient costs, its volatility and randomness are major hindrances to the efficient planning, operation and stability of electricity systems, often leading to imbalances in wind power supply and demand. As intermittent wind power output varies over multiple time frames, the grid operator's daily operating procedures will vary from day-ahead, hourly-ahead and real-time. A direct factor in the generation of wind power is wind speed forecasting; consequently, the building of a reliable predictive model to estimate this data is becoming increasingly important but difficult. Several model combinations have been developed and applied in recent years to reduce uncertainty and improve model prediction performance but with inherent drawbacks based on the training algorithms. Therefore, a comprehensive understanding of their principles, strengths and shortcomings can promote the assessment of their suitability for use. Hence, a review of models using hybrid learning methods for predicting wind speed is presented in this chapter. This review facilitates wind power optimal integration into the electrical grid through improved utility services.

2.3.2 Wind Forecasting Fundamentals

2.3.2.1 Forecasting Horizons

Energy consumers need continuous power supply, and the power grid must maintain an optimal supply-demand equilibrium at all times. In this regard, in wind power, the wind speed resource is an influential random physical quantity that requires continuous adjustment to ensure power grid stability. In fact, the theoretical cubic relationship between wind power and wind speed is evident that an accurate forecast of wind speed data would, in turn, improve the accuracy and reliability of wind power forecasts. At different forecast horizons, wind speed forecasts provide valuable information about future wind speed and power output which is important for optimal power grid management. According to [4], there are four main time-scale categories for wind forecasting [8], viz.; very short-term predictions (from seconds to 30

[8] Different forecast horizon scales can also be found in the work of [74]

minutes); short-term predictions (from 30 minutes to 6 hours); medium-term predictions (from 6 hours to 1 day); and long-term predictions (from 1 day to weeks) based on power system operation requirements. Very short-term forecasts are often used for turbine management and tracking load, whilst short-term forecasts are used for effective management and operation of power grids, and market trading. Medium-term forecasts are pivotal for power system planning/wind turbine maintenance, whereas long-term predictions are for the wind turbine maintenance schedule and establishment of wind farms. Various studies use different forecasting methods based on data elements, available resources, and the need for use, including physical, statistical, ML, and hybrid methods.

2.3.2.2 Classification of Wind Forecasting Methods

In the literature, wind forecasting models are often divided into two main classes, viz.; deterministic and probabilistic methods (also see [4]). Contrary to complex probabilistic methods, deterministic methods have been extensively explored over the years to improve their reliability and elevate their accuracy mostly due to their simplicity and computational efficiency [75,76]. Deterministic methods are also advantageous in that they are consistent with the general principles of standard ML methods [26,75,76]. However, deterministic methods are unable, to some extent, to adequately capture the inherently uncertain nature of wind data. With a huge move towards renewable energy alongside an increase in the penetration of wind power into conventional power grids, point forecasts remain paramount for optimal power grid management frameworks. In essence, a single-point forecast for a random variable such as wind is often not adequate for effective decision-making since the error between the actual value and the point forecast could under or overestimate wind power output thereby compromising investment and trading decisions. Thus, the uncertainty of wind speed predictions cannot be entirely and adequately explained by a single-point forecast. In this case, the most suitable strategy is the one that outputs predictive distributions rather than single-point forecasts. Different from deterministic methods, probabilistic methods provide prediction in the form of probability density functions (PDFs), confidence or prediction intervals, and quantiles of the distribution, which cater for uncertainty when dealing with wind data [26,75,76]. For example, wind power probabilistic forecasts are pivotal for trading, load dispatch, unit commitment, reserve estimation, and operating cost assessment. While wind power can be predicted directly using wind speed time series as the main

input data, it can also be predicted indirectly using numerical weather prediction (NWP) as input data albeit at higher cost as they require huge amounts of data.

2.3.2.3 Wind Performance Metrics

Error measures should distinguish or discriminate between a robust predictive model and an unstable predictive model. The discrimination component allows one to select better models from a collection of comparable forecasting models [26,28]. The abstraction, which is the normalisation of error scores, is also critical for assessing the performance of predictive models over forecasting horizons and terrains. In fact, when errors are normalised (e.g. MAPE), multiple locations can be compared independently of their rated capacities (see Table 2.3).

Table 2.3. Commonly utilised error metrics based on the reviewed studies (also see e.g., [26,75,76])[9]

Metric	Implication	Features
RMSE	Smaller values closer to 0 are desired as they mean better predictive strength.	RMSE penalises large errors more than smaller ones. RMSE is scale-dependent, quadratic-based, and sensitive to data outliers.
MAE	MAE varies between 0 and infinity. Lower (closer to 0) error values denote more reasonable predictions.	Differences are linear. High sensitivity to extreme values than other indices. Linear absolute error measure proportional to the weighting of errors.
MAPE	MAPE varies between 0 and 100%. The lower the MAPE rate the better the predictions.	MAPE can be easily comprehended as compared to other indicators. Sensitive to model bias and skewness. As the error distribution increases, MAPE increases linearly.
R^2	The coefficient of determination varies between 0 and 1, with values closer to 1 indicating a better prediction results.	R^2 is the proportion of the variation in the dependent variable explained by the linear model.
MSE	MSE varies between 0 and infinity plus lower values mean better forecast results.	MSE is a quadratic-based error and penalises larger errors. Less weighting for small errors.

[9] In Table 2.3, the metric that is often mentioned the most in the literature is listed first in the table, followed by the second most mentioned metrics, and so on.

2.3.3 Point Forecasting Methods

A variety of deterministic wind speed forecasting methods have been developed and applied for different forecasting purposes based on a variety of meteorological data and varying forecast time horizons. Among deterministic wind speed forecasting models, there are four main types, viz.; physical, statistical, MLs, and hybrid [4].

2.3.3.1 Physical Methods

Physical methods such as Diagnostic and Computational Fluid Dynamics, rely heavily on climatic data (such as air temperature, surface coarseness, airflow, humidity, pressure, etc.) and power curves to accurately quantify wind power [75,77–80]. These methods, which are governed by atmospheric physical laws, utilise hydrodynamic and thermodynamic principles, through solving complex partial differential equations, to accurately forecast wind speed [79,80]. These advanced and computationally intensive methods require large amounts of training data [75] and are often inaccessible in underdeveloped regions (such as South Africa). Their predictive ability is good under favourable weather conditions, whilst challenging under unfavourable conditions [75] and are reserved for medium and long-term forecasting horizons [79,80].

2.3.3.2 Statistical Methods

Besides their simplicity, efficiency, and ease of modelling, statistical methods produce exceptional results (in the short-term horizon) while remaining cost-effective. To produce forecasts, statistical methods (i.e. ARMAs) use linear combinations of the historical data to identify patterns and trends [30,81-85]. In essence, the interdependence between the past and future values of the time series is assumed in forecasting [82-85]. As ARMA (and its exogenous version ARMAX) models stationary data, ARIMA (and its variants, like ARIMAX and seasonal ARIMA (SARIMA)) models non-stationary data, and these methods are often reserved for short-term forecasting [3,77,86,87]. These models are one of the most commonly used statistical methods in wind forecasting. Heavily dependent on prior mathematical model assumptions, the accuracy or precision of ARMA models decreases as the forecast time scale increases [3,78]. Accordingly, these processes are (to a large extent) unable to sufficiently capture nonlinearities engraved in wind speed. For example, the work of [30] employed a stationary ARMA to capture wind speed characteristics. The study found that ARMA

could not adequately capture the nonlinearities and nonstationarities inherent in wind speed data. To address the nonstationary and random character of wind data, SARIMA [88] and SARIMA coupled to exogenous variables (SARIMAX) [81] were applied in short-term wind speed forecasting. There are instances where ARMA variants outperform advanced models (especially for short-term forecasts) depending on the data features. For example, in the work of [89], ARMA and artificial neural networks (ANN) were compared in three case studies for wind speed forecasting. The predictive performance of ARMAs were more satisfactory than that of the ANN model. Similar results were obtained in the work of [88] where the ARIMA demonstrated superiority over LSTM algorithms in short-term wind speed forecasting. As an alternative to the naive model, ARMAs are often reserved for benchmarking in wind speed forecasting studies (see e.g., [22,75,83]). As one of the simplest statistical models, the naive or persistence is such that prediction at time $t + \Delta t$ is the same as the last observed value at time t.

2.3.3.3 Machine Learning Methods

Fundamentals

Being a subgroup of artificial intelligence (AI) methods, MLs are capable of capturing nonlinear and complex wind speed features that cannot be adequately explained by time series-based statistics methods [91,92]. The application of these methods is associated with increased wind speed prediction efficiency and accuracy. For example traditional ANNs [3,22,89,90], LSTMs [90,93–95], GBDTs [96–100] and SVRs [92,101] have been successfully (to some extent) implemented in wind forecasting at various locations to predict wind speed data. Despite their complex design, these methods are highly efficient, have a tolerance for missing values and/or outliers, capture nonlinearity, minimise overfitting, have high accuracy, and have better generalisation capabilities in forecasting high-resolution wind data (see e.g., [96,98–100]). Yet, these approaches are often black-boxed such that their sophistication in processing data cannot be understood with ease. Descriptive, prescriptive, and predictive [102], are the three main functions which ML methods are centred around. The descriptive exercise involves the use of MLs to describe occurrences, whilst the predictive role involves the utilisation of data to make predictions based on ML methods. The prescriptive role employs MLs to determine what should be done to achieve particular predetermined results.

Supervised Learning

Supervised learning predicts an output based on the input (see e.g., [103] for details). This process involves finding an estimated function $\hat{f}(\boldsymbol{y}_t)$ of the function $f(\boldsymbol{y}_t)$ such that $f : \boldsymbol{y}_t \rightarrow y_t$. The goal of supervised learning is to predict the output value of a function (e.g. $y_t = f(\boldsymbol{y}_t)$) such that

$$\xi_t + \hat{y}_t = y_t = \hat{f}(\boldsymbol{y}_t), \tag{2.72}$$

where $\boldsymbol{y}_t = [y_1, y_2, \ldots, y_{t-1}]$, $\xi_t = y_t - \hat{y}_t$ is the error term, $\hat{f}$ is unknown and it estimates the nonlinear relations between a dependent variable y_t and the input vector $\boldsymbol{y}_t$. The function f is estimated by optimising MLs. The rationale is to minimise the loss or error function between predicted $\hat{y}_t$ and actual value y_t denoted by:

$$\Lambda(y_t, \hat{y}_t) = \frac{1}{n}\sum_{t=1}^{n} {\xi_t}^2, \tag{2.73}$$

where ξ_t is the error term and n represents the training sample size. In the training phase, the function is adapted based on the error between actual and predicted values. For regression purposes, the equation above is the simplest loss function, however, it does not always provide the most desirable generalisation results. Nevertheless, other advanced and effective optimisation methods such as the Euclidean distance (which is similar to minimising squared errors), cross-entropy, hinge loss, and smooth MAE [104] exist in the literature. A structure of risk minimisation, which corresponds to support vector machines (SVM), is commonly used to mitigate model overfitting or underfitting.

$$\min_{\boldsymbol{w}} \frac{1}{n}\sum_{t=1}^{n} \Lambda(\hat{y}_t, f(\boldsymbol{y}_t) + \iota_1 ||\boldsymbol{w}||_2^2, \tag{2.74}$$

where Λ is the loss function. As a compromise parameter, ι_1 is determined through cross-validation and represents the penalty loss function. Besides performance measurement and provision of the direction for improvement during the training phase, a well-balanced loss function should account for model bias and variance, which is critical for model scaling to new datasets [104].

Model Fitting[9]

An underfitting model cannot sufficiently learn from the training data (i.e. characterised high bias and lower variance), such that it performs poorly (lacks a good approximation) of the overall data. This is mostly attributed to the use of simple and linear models (which cannot handle nonlinearity), small input features and training datasets, and too much regularisation. This drawback can be averted by, for instance, increasing model complexity, using feature engineering methods (such as SVMs), removing noise (using WT), and increasing epochs (in deep leaner such as GRU) or training duration [105]. As complexity increases, the probability of underfitting declines, while the likelihood of overfitting increases. An overfitted model excessively comprehends patterns of the training data (with high variance) but performs poorly (in terms of generalisation ability) on the overall dataset. Attributed to high variability, low bias, high complexity, and size of the training data, overfitting can be managed by improving training data quality, increasing training data size, reducing model complexity (through regularisation methods such as LASSO), early training termination, and increasing dropout rates (in deep learning methods) (see e.g., [105,106]). A good fitted model generates predictions with minimal errors such that it generalises the data under study well without succumbing to overfitting or underfitting.

Feature Selection

The goal of feature selection is to select a subset of features from the original dataset to minimise the number of features, based on pre-set criteria. Thus, given set $\Omega_{(m)} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$ oftentimes $\boldsymbol{x}_i \in \mathbb{R}^N$ is a set of predictor vectors (input) whilst $y_i \in \mathbb{R}$ denotes the output. In feature selection, the objective is to utilise $\{\boldsymbol{x}_{(1)}, \boldsymbol{x}_{(2)}, .., \boldsymbol{x}_{(m)}\} \subset \{\boldsymbol{x}_1, \boldsymbol{x}_2, .., \boldsymbol{x}_N\}, m < N$ in establishing input-output association (see e.g., [107] for more details). Feature selection or engineering does not only enhance dimensionality reduction in ML modelling, but promotes optimised learning, improved predictive accuracy, and the interpretability of learning results. In this review, emphasis is placed on information theory or entropy as a feature selection method and its application to wind data (also see e.g., [108]).

Entropy refers to the degree to which a physical system is disordered (see e.g., [108–112] for details). The more chaotic a system is, the more information is contained

[9] Details on the subject of model fitting are thoroughly outlined in the work of [105]

within it. Thus, large entropy values suggest that a time series signal is complex (or noisy), whilst low entropy levels indicate a predictable or deterministic time series (with less complexity) (see e.g., [109–114] for more details). Types of entropy range from SampEn, permutation entropy (PE), Shannon entropy, statistical entropy to residual entropy. In short-term wind speed forecasting, [115] used multiscale fuzzy entropy (MFE) to determine the level of complexity in multiple intrinsic mode functions (IMFs). To overcome the weaknesses of traditional metrics such as MAE or RMSE, which do not account for forecasting error distributions, authors of [116] proposed the Renyi′ entropy as an alternative metric to evaluate different forecasting techniques. The authors of [108] employed SampEn to study the complexity and compared of wind speed temporal series and its connection to wind power prospects in Brazil. The rationale behind entropy in time series analysis is to uncover the complex and hidden dynamic patterns in the data which, in turn, enhances accuracy through the application of the most suitable modelling and forecasting approaches.

Gradient Boosting Decision Trees

Due to recent innovations in MLs, GBDTs are becoming increasingly popular due to their short training time, high accuracy, scalability, and the ability to handle large datasets with efficiency [22,97–100]. The support for the central processing unit (CPU) used by GBDTs is better than the graphics processing unit (GPU) utilised in traditional ANNs. Different GBDTs have been (to an extent) successfully applied ranging from stochastic gradient boosting (SGB), and XGBoost to light-GBM (LGB) in wind power forecasting. The work of [98], employed an advanced XGBoost model to improve wind speed prediction accuracy. Compared with backpropagation neural networks (BPNN) and linear regression (LR) models, XGBoost was highly efficient and accurate. In [100] authors compared ANN, SGB, and a generalised additive model (GAM) for short-term wind speed forecasting. The SGB outcompeted other models based on MAE and MAPE. Besides computational efficiency and robust model tuning, GBDTs are also beneficial in that they have a user-friendly interface. However, there are some drawbacks to using GBDTs. For instance, XGBoost often overfits small datasets, LGB creates more complex trees and can easily overfit small datasets, and SGB requires a lot of trees and is data greedy.

Deep Learning

In the literature, ANN architectures vary from the simplest FFNs to the more complex Auto-encoder (AE), with the difference being the way and direction in which the data moves within the network. Different from advanced LSTM and GRU, conventional FFNs (e.g. NNAR) move data in the forward direction within the network [28,83,86]. The deep learners (such as RNNs, LSTMs, GRUs, AE, etc.) are built in such that they can store and analyse historical sequence data. For instance, RNNs employ recurrent neural connections to explain and handle highly sequential data [90]. However, RNNs (analogous to FNNs) often suffer from vanishing gradients during deep propagation, hence, LSTMs were proposed by [95] to circumvent this drawback. The LSTM memory cell, which explains the dependency relationship in the input time series data [93,95,117], is regulated by different gates, viz.; input gate, output gate, and forget gate. These gates determine the addition, removal, and output of information from the memory cell. As evidenced by studies such as [94,117–119], LSTM and its variant GRU deliver effective reserve optimisation, increased learning ability, and mitigates gradient explosion in forecasting time series data. In fact, these methods are also designed in such a way that they are able to selectively discard irrelevant information. In [94], bi-directional LSTMs (bi-LSTMs) dominated persistence models and SVMs in short-term multi-scale wind speed predictions based on MAE, MAPE, and RMSE. The study further found that bi-LSTMs' superiority was due to the fact that these methods consider both historical and future sequences when forecasting. The work of [120] compared the LSTM with the standard ANN in short-term wind speed forecasting. Due to their inherent ability to selectively remember patterns over a long period, the LSTM model produced the lowest day-ahead error and the best performance based on MAPE. Using historical meteorological variables (i.e. temperature, humidity, and air pressure), the work of [93] developed a multivariate LSTM (MV-LSTM) to predict wind speed in the next hour. Based on the RMSE, MAE, mean bias error (MBE), and MAPE, the proposed demonstrated superiority over ARMA and single-variable LSTMs. The study results further showed that the performance of the proposed model varies with terrain. A comparative analysis of LSTM and SVM in short-term wind speed forecasting was conducted in the work of [121]. Based on the MSE and RMSE, the study results showed that LSTM was more effective when compared to SVM mainly due to its inherent pattern recognition for longer durations. The authors further highlighted that LSTMs and their variants can be used on much larger datasets in order to enhance prediction accuracy and reliability. From the review, the standard

or stateful LSTMs are oversubscribed for time series prediction tasks (such as wind forecasting), though stateless LSTMs (which resets cell memory every batch) are better in terms of efficiency, stability, and accuracy [28,118,119]. It should be noted further that deep learners are often computationally intensive, making them unsuited for environments with minimal resources.

Support Vector Methods

Aside from robust kernel tricks and a solid mathematical foundation, SVRs (an extension of SVM) exhibit a high convergence speed, has good sparse and generalisation properties, and can effectively handle smaller datasets [22,92,122–127]. In the work of [126], SVR compares better with the multilayer perceptron (MLP) model based on the MSE in short-term wind speed forecasting using mean daily wind speed data. Based on South African NWP data from the Alexander Bay region, authors of [101] showed that SVR performance in short-term wind speed forecasting can be (to an extent) improved by the systematic selection and integration of appropriate input features. Despite their effectiveness at short-term wind speed forecasting, SVRs suffer from the need for too many support vectors plus their forecasts are deterministic. On the other hand, Bayesian-based RVMs offer accuracy similar to that of the SVRs alongside small support vectors (as they use fewer kernel functions), reduced training time, probabilistic structure and sparseness, and greater generalisation strengths. In [128], RVM compared better than BPNN and SVM for wind speed forecasting. The RMSE and mean relative error (MRE) showed RVM to be more efficient, robust, and accurate. Similarly, authors [129] compare short-term wind power forecasts from the three wind farms using RVM, wavelet and radial basis function (RBF) methods. The RMSE revealed that RVM had superior predictive strength, and better generalisability when handling small samples, and the probabilistic forecasts produced could explain the uncertainty and the change in wind power output. The main benefit of RVM is that they produce both mean (deterministic) and variance (probabilistic forecast) estimates (see e.g., [127,130]). RVM's drawback appears to be training time for large datasets [131].

Ensemble Learning

The RF algorithm uses fewer parameters, overcomes the overfitting problem of ANNs, provides efficient classification and processing for both large and small samples, accurately estimates the importance of factors, and is noise resistant [132–135].

According to [132–135], tree-building with some randomness and variability increases generalisability and diminishes the variance of the RF model. Using 10- minute samples, the authors of [136] showed that RF improves accuracy (based on RMSE) in short-term wind speed forecasting. Similarly, the proposed RF approach in the work of [137] proved to be a flexible solution for capturing the inherent features of variant wind speed data over short-term horizons. Using coastal wind speed data, the work of [138] showed that RF compares better than ANN, and the persistence model in terms of MAE, MAPE, and RMSE in short-term wind speed predictions. Despite RF complexity at the level of the tree, decision tree algorithms have low sensitivity to missing values and outliers, minimise overfitting, handle small and large data with efficiency and accuracy, and are (to some extent) noise resistant [135].

2.3.4 Probabilistic Forecasting Methods

Although probabilistic predictions are complex and require a lot of computing power, they are effective at predicting the future and can be effectively used to plan different strategies for a variety of possible outcomes. For instance, the tails of the wind power PDFs can be used for the study of extreme errors which may be of interest in ensuring optimal reliability of the power grid system [23,76,139]. Furthermore, for operators to avoid costly backup facility activation, they rely on the certainty of the prediction model. Thus, prediction errors or uncertainty help system operators with effective risk management strategies. Probabilistic forecasting can be categorised into parametric and non-parametric methods.

2.3.4.1 Parametric Methods

Parametric models have fully defined structure, distributions, and only parameters are to be estimated. Parametric methods take into account the probability distribution of wind fluctuations and their prediction errors, resulting in different confidence intervals or residual distributions. However, these techniques cannot guarantee the precision of the predictions, as they cannot control the accuracy of the expected probability distributions. In the context of highly variable weather phenomena such as wind speed (ultimately wind power) , it is not always appropriate to assume the form of the output distribution, since wind speed error distributions are often dependent on terrain and specific forecast horizons (see e.g., [139]). In essence, the selection of a distribution function that represents the real distribution of wind speed at a particular site is crucial in order to accurately predict wind power at that specific

location. A Weibull distribution, as illustrated by the equation below, has been widely used for representing wind speed variations over a long period of time, mostly due to its flexibility and simplicity [75,140].

$$f_{WB}(v) = \frac{\gamma^*}{\beta'}\left(\frac{v-\alpha^*}{\beta'}\right)^{\gamma^*-1} e^{-\left(\frac{v-\alpha^*}{\beta'}\right)^{\gamma^*}}, \tag{2.75}$$

In the equation above, v is the wind speed, α^* denotes the location, β' is the scale, and γ^* represents the shape parameters. Wind speed variations, however, have a thicker tail than wind power fluctuations [139] due to short-term transient behaviour (such as wind turbulence and wind gusts) such that neither the conventional Weibull distribution nor its variants can accurately capture and explain this complex behaviour.

2.3.4.2 Non-parametric Methods

Non-parametric methods are highly scalable, do not presuppose a predefined distribution form, and produce output values directly from data [75,76]. Non-parametrics are commonly divided into two main categories, viz.; traditional non-parametric methods (including but not limited to quantile regression-based methods) and prediction interval methods (such as the lower upper bound estimate (LUBE)) [75,76]. In [141], regression quantiles (based on generalised additive model (GAM)) were used to predict wind power production probabilities. The least mean absolute deviation (MAD) and RMSE revealed that the proposed quantile-based approach was ideal for handling wind uncertainty. In a similar study, [142] proposed a quantile regression bidirectional minimal gated memory network (QRBiMGM) and kernel density estimation approach, which was found to be computationally efficient, accurate (based on RMSE), and possesses strong adaptability, making them suited for forecasting wind data. Based on three Chinese wind farm data, the authors of [143] proposed quantile regression closed-form continuous-time neural networks (QRCfC). The MAPE and CRPS values demonstrated the superiority of QRCfC over quantile regression time convolution networks (QRCNN), quantile regression LSTM (QRLSTM), quantile regression GRU (QRGRU), and quantile regression neural network (QRNN). Furthermore, the proposed approach could effectively explain the risks associated with wind speed, which is valuable for wind energy optimisation strategies. The work of [144] integrated LUBE and Quasi-RNN for wind speed forecasting. The model used a stochastic gradient descent training approach, and

when evaluated using eight datasets, the model reduced prediction width by 33% compared to traditional models. An improved LUBE model was proposed based on gradient descent training methods for better efficiency and wind speed prediction performance in the work of [145]. The approach was compared with other LUBEs using various probabilistic loss functions such as normalised average deviation (NAD), prediction interval coverage probability (PICP), etc. The proposed model was found to improve learning time and prediction interval quality. A fully connected layer GRU learned the long-term dependence of wind power data and merged the learned features to calculate the upper and lower bounds of the target prediction intervals in the work of [146]. The proposed method performed better than standard LUBE, LSTM-based LUBE, mean-variance estimation, and a newly developed gradient descent method. However, these approaches often require a large amount of data to achieve a certain level of accuracy, hence, they are often complex and computationally intensive.

2.3.5 Hybrid Forecasting Methods: A Comprehensive Review

2.3.5.1 Significance of Hybrid Models

In recent years, various model combinations (or hybrids) have been developed and employed to reduce uncertainty and enhance prediction performance (see e.g., [31, 147-153]). To alleviate deficiencies and take advantage of each model's strengths, hybrid models usually combine more than one forecasting method to form a new model. Hybridisation can effectively model and forecast the complex autocorrelation structures inherent in time series data such as wind speed [31]. Differently from individual models that are only suitable for specific applications, hybrids can eliminate the need for testing different models thereby making it easier to identify the best-performing model [153]. Nonetheless, hybrid/combined approaches are excellent in handling and overcoming frequent statistical, computational, and representational drawbacks in the forecasting arena (see e.g., [26,153]). The latter problem (i.e. representational) refers to the lack of model scalability, which may limit the model's ability to capture the complexity of the problem; whilst the earlier (statistical) relates to the downsides that are found when the training data set is insufficient to represent the actual process. Computational disadvantages include the consequences of discontinuing model training in local optimisation (due to gradient explosion) which can be remedied more effectively by recurrent model training. However, even

selecting the appropriate hybrid prediction model can be a complex and time-consuming task due to the site-specific nature and varying forecast horizons in the wind data forecasting scenario.

2.3.5.2 Classification of Hybrid Models

In the literature, combined/hybrid approaches blend techniques from a number of disciplines, including the following (see [153] for details):

- *Weighting approach*
  This approach involves assigning weights based on the model's predictive performance. Moreover, this approach is known to be easy to implement and highly adaptive to new set of data, but does not guarantee forecast accuracy plus it requires an extra model for determining the weights.

- *Pre-processing or data decomposition approach*
  This method has to do with breaking down signals for prediction purposes. Overall, this approach has higher performance and is associated with robust adaptations to changes or variation in wind speed data, but requires detailed mathematical knowledge and provides a slow response to new data.

- *Parameter selection or optimisation approach*
  This technique looks at optimising the hyperparameters of the prediction model. These approaches are often complex to develop (or program), computationally expensive and their performance is reliant on the user knowledge.

- *Data post-processing technique/error processing approach*
  This approach involves combining predictions based on predicting the residual error (i.e. stacking) generated by the forecast model (s). This approach has high accuracy and reduces systematic error, but is often computationally expensive.

2.3.5.3 Review of the Related Hybrid Literature

In [147], authors proposed Kalman Filter (KF)-ARIMA (i.e. KF-ARIMA) and ARIMA-ANN models to accurately predict wind speed. The study results showed that hybrids captured and explained linear and nonlinear wind speed data components accurately. A combined approach based on ARIMA and EMD (empirical mode decomposition) (ARIMA-EMD) proposed by [148] was employed to accurately predict wind speed

over the short-term forecasting horizon. However, the approach is associated with computational complexity due to EMD and forecast error accumulation due to the inability of ARIMA to capture nonlinear components. In [154], particle swarm optimisation (PSO) was used to optimise the ARIMA parameters. Thereafter, the optimised ARIMA was blended with KF models for robust and accurate wind speed forecasting. Besides high level of accuracy, the proposed approach is associated with high sensitivity to parameter setting due to the PSO algorithm. In another study, [155] combined WT, genetic algorithm (GA), and SVM. Besides its high computation cost due to the use of GA algorithm, the proposed method offered greater generalisation strength and produced accurate and robust forecasts than the individual models. The LGB and the Gaussian process regression (GPR) model was proposed in the work of [96] for accurate short-term wind speed predictions. The LGB-GPR model produced (despite its complexity) the most accurate, robust, and reliable probabilistic forecasts when compared with individual models. Proposed by [156], a hybrid of adaptive boosting (AdaBoost) and extreme learning machine (ELM) model (AdaBoost-ELM) was used to improve wind speed predictions based on the MAE, RMSE, coefficient of correlation (r), and Nash-Sutcliffe efficiency coefficient (NSEC). The proposed approach proved to be less sensitive to the extension of the forecasting horizon. It should note that Adaboost is highly sensitive to outliers and demarcation parameters (with poor selection leading to overfitting or underfitting). In [157], improved variational mode decomposition (VMD) combined with GA and permutation entropy (PE) (IVMD-PE-GA-stacking) was proposed for short-term wind power forecasting. The proposed method outperformed other methods such as SVM, LSTM, XGBoost, deep neural network (DNN), GBDTs, ARIMA, and convolutional neural networks (CNN)-LSTM based on the smallest MAE, MAPE, and RMSE. Though robust to noise, it is paramount to note that VMD's computation cost is slightly higher (when compared to WT) plus its performance is reliant on the decomposition modes selected. Hence, the authors articulated the need to improve the computational time and complexity of the approach in the future. In [158], authors proposed an LSTM-entropy (specifically SampEn) wind predictive method to improve the model's adaptability to varying wind conditions. Two Chinese wind farms were utilised to show the superiority of the proposed method based on MAE, RMSE and r. Furthermore, the application of SampEn reduced model input redundancy, thereby enhancing model generalisation abilities. Similar to the work of [158], [159] combined GA-VMD, SampEn, and Bi-LSTM in short-term wind speed prediction. The original wind speed

data was decomposed (through GA-VMD) into several modal subseries, SampEn reconstructed similar subseries, whilst Bi-LSTM was superimposed to generate accurate forecasts. Overall, GA prioritises high accuracy and robustness over computation cost and complexity as also seen in [155]. The work of [160] combined NN and WT to create NNWT for short-term wind power forecasting in Portugal. The proposed model (based on the least MAPE values) dominated individual models such as persistence, ARIMA, and ANN. Though highly accurate, the proposed approach often suffer from gradient explosions (due to reliance on the ANN) when dealing with high variant detailed subsignals at low levels. The work of [161] leveraged grey relational analysis (GRA), least SVM (LSVM), and RBF neural network (RBFNN) to effectively reduce (to some extent) model complexity (through GRA) thereby accurately capturing nonlinearity (through LSVM-RBFNN) at very short-term wind power forecast horizon. In another study, [151] blended wavelet decomposition (WD) and weighted RF (WRF) based on the niche immune lion algorithm (NILA) (WD-NILA-WRF). Based on MAPE, RMSE, MAE, and goodness of fit, the proposed approach outcompeted BPNN, SVM, RF and NILA-RF in ultra-short-term wind power prediction. The study result further showed that the proposed approach could withstand noise and unstable factors in the wind data thereby enhancing the models' generalisation and robustness. Using 10-minute averaged wind speed data, the work of [92] combined filtering methods (i.e. KF and/or WT), ARIMA, and ML algorithms (i.e. SVR and RF). The ARIMA model was used to capture linear components and the resultant residuals (representing nonlinearity) were decomposed using level 5 WT. These were utilised as input features into SVR and RF. Based on R^2 and RMSE, KF-WT-ML produced the best results. In [152], the authors employed a repeated WT-ARIMA model (RWT-ARIMA) in very short-term wind speed forecasting. The proposed RWT-ARIMA model demonstrated better accuracy than the ARIMA, persistence, and WT-ARIMA models. As seen in [147], ARIMAs cannot entirely capture the nonlinear wind component plus the repeated wavelet approach renders the model computationally intensive. A Legendre multiwavelet-based neural network (LMWNN) was proposed in the work [162] to capture the inherent nonlinear component wind speed data at a short-term forecast horizon. Based on MAE, MAPE, MSE, RMSE, and R^2, the proposed adaptive approach (characterised by excellent nonlinear learning) yielded a much more robust multi-step forecast as compared to ARIMA, BP, DWT-LMWNN, EMD-LMWNN, and Morlet-NN. Despite the complexity and parameter sensitivity of the approach, when compared to WT, the LM algorithm

struggles (to some extent) when dealing with a highly variant large dataset. In [163], the proposed WT-Linear Neural Networks with Tapped Delay (WT-LNNTD) outperformed the Naïve, FNN, LNNTD, PSO neural network (PSONN), GA neural network (GANN), WT-PSONN, WT-GANN, and several other models in short-term wind speed prediction by achieving lower MAE and MAPE values, as well as superior PI indices and higher skill scores, including the Likelihood, Brier, and Heidke scores.

2.3.6 In-Depth Analysis and Synthesis

The study employed the following selection criteria to identify the most appropriate and relevant research papers/studies. A paper should be written in English from a peer-reviewed journal or conference focusing on wind power/speed forecasting and must have been published between 2010 and 2023. Papers published in journals with high-impact factors alongside clear and detailed novel contributions to the wind forecasting literature are prioritised. Furthermore, each paper should present an evaluation of hybrid/combined methods aimed at enhancing forecast accuracy, with a bias towards wavelet-ML hybrid methods. The paper should (to an extent) give enough details (in the abstract) on the characteristics of the data, including granularity, the country where research was conducted, and the variables involved, as well as the forecasting horizon employed. Papers that the researcher could not access were also not included in the synthesis review. Furthermore, *"wind speed forecasting"*, *"wavelet transform"*, *"wind power forecasting"*, and *"wind speed predictions*" were the frequently searched phrases. The synthesized results are presented below (also see Tables 2.4 to 2.6).

Table 2.4. Summary of the 30 reviewed hybrid models for wind forecasting[10]

#	Ref	Model	Resolution	Wind Input	Forecast Scale Horizon[11]	Wind Output	Metrics	Future work	Country of research (year)
1	[160]	NN-WT	Hourly	TS	SH	P	MAPE, SSE, SDE		Portugal (2011)
2	[164]	WTNN	15 min	TS	SH	S	SSE; MAE; RMSE; MAPE; MSE; R^2; R		Canada (2013)
3	[154]	PSO-ARIMA-KF	Day	TS	SH	S	ARE; MAE; RMSE		China (2014)
4	[165]	WT-TNN	Hourly	TS	SH	S	MAE; RMSE; MAPE		China (2014)
5	[166]	AWNN	10-min	TS	SH	S	APE, MAPE; MAE, RMSE, R		USA (2014)

[10] S=Speed; P=Power; TS= Time Series; NN= Neural Network; USH=Ultra Short-Term Horizon; SH= Short-term Horizon; MH=Medium Horizon; LH= Long Horizon; SSE=Sum of square error; APE= Absolute Percentage Error; SDE= standard deviation of error; MSE=Mean Square Error; CC/r/R=Correlation Coefficient; sMAPE=Scaled MAPE; IA=Index of Agreement; ICPC=Interval Coverage Probability Coefficient; FI=Forecast interval U_1=Unbiased Statistic; AWNN=Adaptive Wavelet Neural Network; TNN=two-hidden-layer Neural Network; RT=Run Test; FICP=Forecasting Interval Coverage Percentage; FIAW= Forecasting Interval Average Width; SK=Skill Score; ANFIS= Adaptive Neural Network based Fuzzy Inference System; LSSVM= Least Square Support Vector Machine; LNNTD=Linear Neural Networks with tapped delay; KELM= Improved Extreme Learning Machine with Kernel; HS=Harmony Search; DWT=Discrete Wavelet Transform; CS=Cuckoo Search; BGA =Binary Genetic Algorithm; BBFWA=Bare-Bones Fireworks Algorithm; LSSVR= Least Squares Support Vector Regression; CEEMD= Complementary Ensemble Empirical Mode Decomposition ;WOA= Whale Optimisation Algorithm; IOWA=Induced Ordered Weighted Average; CNN=Convolutional Neural Networks ; SE= Sample Entropy; WA=Wavelet analysis; BP= back Propagation; SRCV= Grid Search with rolling; Cross-Validation; ROPSO=Randomness Operator-based Particle Swarm Optimization; NARMAX= Non-linear Autoregressive Moving Average with External Inputs

[11] This is based on the context and how it was stated on the study under review

#	Ref	Model	Resolution	Wind Input	Forecast Scale Horizon[11]	Wind Output	Metrics	Future work	Country of research (year)
6	[130]	EEMD-RT-RVM	15 min	TS	USH	P	MAPE; FICP;FIAW		China (2016)
7	[167]	EEMD-ANFIS-SARIMA	10 min	TS	SH	S	MAE; RMSE; MAPE		USA (2017)
8	[168]	EEMD-LSSVM	10 min	TS	SH	S	MAE; RMSE; MAPE, SSE, SDE,CC		Spain (2017)
9	[163]	WT-LNNTD	Hourly	TS	SH	S	MAE; MAPE; FIs; SKs		Canada (2018)
10	[151]	WD-NILA-RF	5 min	NWP	USH	P	MAPE; RMSE; MAE; R^2		China (Mongolia) (2018)
11	[169]	CEEMD-SE-HS-KELM	5 min	TS	SH	P	RMSE; MAE; R^2		China (2019)
12	[152]	RWT-ARIMA	1 min; 3 min; 5 min; 7 min; 10 min	TS	USH	S	MAE; RMSE; MAPE; SSE	Quantification of wind power generation forecast errors.	Ireland (2019)
13	[90]	DWT-LSTM	15 min	TS	SH	P	MAE; MAPE; RMSE	Optimiser algorithm of LSTM will be improved for better time series prediction.	China/Mongolia/ Netherlands (2019)
14	[170]	CS-WD-WNN	15 min	TS	SH	S	MAE; RMSE; MAPE		China (2019)
15	[171]	CNN-LSTM	Day	NWP	SH to MH	P	MAE; RMSE; MAPE;MSE		Ethiopia (2022)

#	Ref	Model	Resolution	Wind Input	Forecast Scale Horizon[11]	Wind Output	Metrics	Future work	Country of research (year)
16	[172]	DWT-ANN	Day	TS	SH	S	MSE		Algeria (2020)
17	[94]	WT-bi-LSTM	15 min	TS	SH	S	MAE; MAPE; RMSE		China (2020)
18	[123]	VMD-BGA-BBFWA-LSSVR	10 min	TS	SH	S	MAE; MAPE; RMSE	Enhance multistep framework for multistep wind speed prediction.	USA (2021)
19	[150]	CEEMD-ARIMA-SVM	10 min	TS	SH	S	MAE; MAPE; RMSE		China (2022)
20	[173]	CEEMD–WOA–ELMAN	15 min	NWP	USH	P	RMSE; MAE; MAPE; IA; *R*	Ultra-short wind power prediction: effects of different time scales. A faster running time and improved accuracy could be achieved by optimising the model.	China (2022)
21	[80]	IOWA-CNN-LSTM	15 min	NWP	SH	P	RMSE; MAE	Model extension to long-term forecast horizon.	China (2022)
22	[96]	LGB-GPR	Hourly	NWP	SH	P	MAPE; RMSE; R^2; ICPC; CRPS		China (2022)
23	[92]	KF-WT-ML	10 min; 30 min; 60 min	TS	SH	S	MAE, R^2; RMSE	Use feature selection and hyperparameter fine tuning to improve model accuracy. Also assess accuracy at	Portugal/USA (2022)

#	Ref	Model	Resolution	Wind Input	Forecast Scale Horizon[11]	Wind Output	Metrics	Future work	Country of research (year)
								longer-term forecasting horizon.	
24	[157]	IVMD–PE–GA	15 min	NWP	SH	P	MAE; RMSE; MAPE	Improve wind power prediction accuracy by reducing algorithm complexity and analysing the impact of multiple wind farms on each other.	China (2023)
25	[159]	GA-VMD-SE-BiLSTM	10 min	TS	SH	S	RMSE; MAE; MAPE; sMAPE; U_1; R^2		Spain (2023)
26	[174]	wavelet-ANN	Day	TS	MH to LH	S	MSE; RMSE; MAPE; MAE; R^2		Iraq (2023)
27	[162]	EMD-LMWNN	5 min; 15 min; 30 min	TS	SH	S	MAE, MSE; MAPE; RMSE; R^2	Improve self adaptiveness and training speed of the model	USA (2023)
28	[175]	BiLSTM-LGB	10 min	NWP	SH	P	MAE; RMSE; MAPE	Data analysis for dynamic wind speed division, wind direction clustering, model parameter optimisation, and downscaling of NWP data from multiple stations to improve accuracy.	China (2023)

#	Ref	Model	Resolution	Wind Input	Forecast Scale Horizon[11]	Wind Output	Metrics	Future work	Country of research (year)
29	[176]	VMD-GRUGSRCV	Hourly	TS	SH	S	MAE; RMSE; MAPE; sMAPE	Concurrent application of multiple data processing methods can further enhance decomposition performance. Methodology for optimising the unique lag order of each decomposed subseries such as the swarm intelligence algorithms can be employed to fully extract their temporal characteristics.	USA (2023)
30	[177]	WT-ROPSO-NARMAX	15 min	NWP	SH	P	MAE; RMSE	Include methods for model selection to further enhance the proposed approach.	China (2023)

Table 2.5. Model combination strategies employed in the 30 reviewed hybrid studies.[12]

No.	Ref	Model	SM	ML	SD	OA	EPP	Entropy	Decomposition/Filter method	Forecast combination	LSTM Type	Simple benchmark
1	[160]	NN-WT		✓	✓				DWT(DB4) (L=3)	Linear		Naive & ARIMA
2	[164]	WTNN		✓	✓				WT(DB3)(L=3)	Linear		Other (BP)
3	[154]	PSO-ARIMA-KF	✓	✓	✓	✓ (PSO)			KF	Unclear		ARIMA
4	[165]	WT-TNN		✓	✓				DWT (DB3)(L=3)	Linear		Naive
5	[166]	AW-NN		✓	✓				CWT (Morlet)(L=7-9))	Linear		None
6	[130]	EEMD-RT-RVM		✓	✓	✓ (Bayesian)			EEMD	Linear		Other (RVM)
7	[167]	EEMD-ANFIS-SARIMA	✓	✓	✓				EEMD	Linear		ARIMA
8	[168]	EEMD-LSSVR		✓	✓				EEMD	Linear		ARIMA
9	[163]	WT-LNNTD		✓	✓				DWT (DB 10)(L=7)	Linear		Naive
10	[151]	WD-NILA-RF		✓	✓	✓ (NILA)			DWT (?) (L=5)	Linear		Other (BP)
11	[169]	CEEMD-SE-HS-KELM**		✓	✓	✓ (HS)		SampEn	CEEMD	Unclear		Other (ELM)
12	[152]	RWT-ARIMA	✓		✓		✓		MODWT(DB2)(L=3)	Linear		ARIMA
13	[90]	DWT-LSTM		✓	✓				DWT (DB7) (L=?)	Linear	Stateful	Other (BP)
14	[170]	CS-WD-WNN		✓	✓	✓ (CS)			DWT (?) (L=6)	Linear		Naive &ARIMA
15	[171]	CNN-LSTM		✓		✓ (Bayesian)				Nonlinear	Unclear	Other (ANN)

[12] SM=Statistical Methods; ML=Machine Learning Methods; SD= Signal Decomposition; OA= Optimisation Approach; EPPM= Error Post-Processing; LC=Linear combination; NLC=Non Linear Combination, UC=Unclear; SF=Stateful; BP=Back propagation Network; ENN =Elman neural network; LR=Linear Regression; L=Decomposition Level; DB=Daubechies

No.	Ref	Model	SM	ML	SD	OA	EPP	Entropy	Decomposition/Filter method	Forecast combination	LSTM Type	Simple benchmark
16	[172]	DWT-ANN		✓	✓				DWT (DB1-DB10)(L=5)	Linear		None
17	[94]	WT-bi-LSTM		✓	✓				DWT (DB4) (L=3)	Linear	Stateful	Naive
18	[123]	VMD-BGA-BBFWA-LSSVR		✓	✓	✓(BGA)			VMD	Unclear		Other(LSSVR)
19	[150]	CEEMD-ARIMA-SVM	✓	✓	✓		✓		CEEMD	Linear		ARIMA
20	[173]	CEEMD-WOA-ELMAN		✓	✓	✓ (WOA)			CEEMD	Linear		Other (ELMAN)
21	[80]	IOWA-CNN-LSTM		✓		✓ (IOWA)				Nonlinear	Unclear	Other (SVM/RBF)
22	[96]	LGB-GPR		✓		✓ (Bayesian)				Nonlinear		Other (LR)
23	[92]	KF-WT-ML		✓	✓	✓ (GSRCV)	✓		DWT (DB3) (L=5)	Linear		Other (SVR\RF)
24	[157]	IVMD-PE-GA		✓	✓	✓ (GA)		PE	VMD	Linear		ARIMA
25	[159]	GA-VMD-SE-BiLSTM		✓	✓	✓ (GA)		SampEn	VMD	Linear	Unclear	Other
26	[174]	WT-ANN		✓	✓				CWT (L=?)	Linear		Other (ANN)
27	[162]	EMD-LMWNN		✓	✓	✓ (Legendre)			EMD	Linear		ARIMA
28	[175]	BiLSTM-LGB		✓			✓			Nonlinear	Unclear	Other (BP)
29	[176]	VMD-GRU-GSRCV		✓	✓	✓ (GSRCV)			VMD	Linear		Other (BP)
30	[177]	WT–ROPSO–NARMAX		✓	✓	✓ (ROPSO)			DWT (DB4)(L=3)	Nonlinear		Naive
Total			**4**	**29**	**26**	**16**	**4**	**3**	**WT=14; VMD=4; C\EEMD=7; KF=1**	**NLC=5; UC=3; LC=22**	**SF=2; NC=4**	**Naive=6; ARIMA=9; Other=15**

2.3.6.1 Current Trends and Insights

The predictive performance (in terms of accuracy, reliability, etc.) of wind forecasting models is heavily dependent on the characteristics of the datasets, the structure and the architect of the algorithm, and the hyperparameters utilised. Below, the comprehensive evaluation of the 30 hybrid studies from different countries is provided (also see Table 2.4 to 2.6).

- Over 50% (15 out of 30) of these studies were conducted in China, indicating a significant contribution of Chinese research to the evaluated corpus. In total, only two (2) out of 30 studies (6%) were conducted in Africa (Algeria and Ethiopia), while six (6) out of 30 studies were in the USA. None of the 30 reviewed hybrid studies were from the Southern Africa region. Hence, there is a need for wind forecasting models to accurately and reliably quantify available wind resources in the Southern Africa region to promote continuous investment in wind power and its technologies.

- Based on the analysis of the input variables, eight (8) of the 30 synthesised hybrid studies utilised NWP variables as input data, while 22 of the 30 reviewed studies used time series data. A significant finding from the synthesis is that 40% (12 out of 30) of the studies focused on wind power predictions, whilst 60% concentrated specifically on forecasting wind speed. Further, the synthesis showed that one (1) out of 30 (3%) of the studies used wind speed data with granularity (or resolution) averaged at 1 minute, whilst more than 90% used data with resolution of atleast 10 minutes. To adequately capture rapid changes (mostly due to wind gusts and shear) in wind data, high resolution data, particularly 1-minutely averaged resolution, are of significant importance (see Table 2.4).

- In terms of the use of feature selection methods (specifically information theory), only 10% or three (3) out of the 30 synthesised studies applied entropy for feature selection during the pre-or-post-processing stage to assess the complexity or randomness of wind data signals/subsignals. Demonstrating the scant applications of information theory to improve wind speed prediction in the literature (see Tables 2.4 and 2.5).

- Weighing on the models employed in hybridisation approach, among the 30 hybrid methods that were assessed, it was found that 13% (4 out of 30) of studies employed, blended some form of statistical method with other models; whilst the majority (29 out of 30 or 97%) used ML methods (in combination with other methods). Furthermore, a substantial proportion of these models (more than 86% or (26 out of 30)) integrated some form of pre-processing approach, whereas 53% (16 out of 30) adopted an optimisation approach (including but not limited to PSO, NILA, GA, CS, HS, WOA, etc.). Atleast thirteen per cent (13%) (4 out of 30) of the models made use of some form of post-processing or error-processing approach (see Tables 2.4 and 2.5).

- Studying the status (or type) of the LSTM network applied, very few studies disclosed the state of LSTM applied during time series modelling and forecasting. In fact, of the six (6) hybrid models that incorporated LSTM, only two (2) were disclosed to have used the standard or stateful LSTM, while the remaining four (4) were unclear. Hence, there is a lack of disclosure of the LSTM network state and very scant application of stateless LSTMs in wind forecasting (see Table 2.4).

- Reflecting on the utilisation of the data or signal decomposition methods, a significant portion of the combined strategies, specifically around 47% (14 out of 30), opted to employ wavelets, with the Daubechies being the most utilised filter among the studies. However, most (at least 10 of the 14) of these studies did not consider the performance of the proposed hybrid model at varying wavelet decomposition levels, let alone examine the concept of wavelets in detail, nor provide an optimised reproducible method for selecting the optimal level of decomposition (see Table 2.5). On the other hand, atleast 23% (7 out of 30) of strategies used the C/EEMD data decomposition method, while 13% (4 out of 30) applied VMD. The C/EEMD and VMD approaches are known to be effective but more complex and might be computationally expensive when compared to the more efficient and simpler WT (see Tables 2.4 and 2.5).

- Focusing on the forecast combination methods, about 22 (about 73%) of the 30 hybrid studies evaluated employed some form of linear reconciliation approaches of different subsignals predictions to arrive at the final forecast

value, whilst five (5) out of 30 (about 17%) used nonlinear reconciliation approaches (e.g. SVR, RF, etc.). In practice, linear forecast reconciliation models (by their design) lead to error accumulation in the final forecast as these approaches cannot account for the nonlinearity inherent in wind speed signals or subsignals (see Tables 2.4 and 2.5). In 10% (three (3) out of 30) of the reviewed hybrids, the forecast combination approach was not clearly stipulated. Hence, there is a need for details in this area in the future to enhance the reproducibility of the proposed methods.

- The review results further reveal that only four (4) out of 30 (or 13%) of the studies evaluated were categorised under ultra-short forecasting, with the majority (25 out of 30) (or 86%) falling into the short-term forecast horizon. Studies with a medium-to-long-term forecast horizon constituted one (1) (or 3%) of the total of 30 studies. Overall, the synthesis revealed the prioritisation of short-term forecasting time scales over medium and long-term forecasting time scales. Furthermore, most studies focus on one specific horizon such that either ultra-short or short-term or medium-term or long-term horizon (see Tables 2.4 and 2.5).

- In benchmarking, the commonly used benchmark model is ARIMA. In fact, of the 30 hybrid studies reviewed nine (9) (or 30%) employed the ARIMA model as a baseline, whilst the Naive model was applied in the six (6) studies. On the other hand, 15 of the 30 (50%) of the reviewed studies used different other models (i.e. MLs) for benchmarking, with seven (7) of these 15 models (almost 50%) being the standard ANN or BP, whilst support vectors were utilised in four (4) of these 14 studies. Only one (1) of the 30 studies benchmarked against a simple linear regression model. Overall, there is a clear move away from model comparison with the traditional benchmarks. This can be concerning since failure to benchmark against simple models (particularly naive or ARMAs) can lead to misleading conclusions or spurious claims of ML superiority (see [178] for more details).

- Reflecting on the model performance metrics, the most frequently utilised error metric measures across the 30 evaluated studies are deterministic and encompass, but are not limited to, RMSE, MAPE, MAE, R^2, and an MSE.

Among the indices previously mentioned, the RMSE is the most frequently utilised error metric indicator, whereas the MSE seems to be the least utilised. It is also evident that the application of probabilistic error indicators (e.g. CRPS, and prediction interval width indices (e.g. prediction interval normalised average width (PINAW)), is notably scant across the 30 studies under consideration. Furthermore, the use of appropriate skill score metrics (such as the Dawid-Sebastiani (DS), and probability integral transform (PIT)) is very limited. According to [178], the use of inadequate evaluation methods makes it difficult to differentiate truly competitive techniques from flawed ones by bypassing spurious results (see Tables 2.4 and 2.5).

- Finally, the suggested future research work emanating from the 30% reviewed hybrid studies include but is not limited to, improving wind power prediction accuracy by reducing algorithm complexity and analysing the impact of multiple wind farms. Use feature selection and hyperparameter fine-tuning to improve model accuracy. Enhance a multistep framework for multistep wind speed prediction, quantifying wind power generation forecast errors. There is also a need to assess wind hybrid forecasting methods over the long-term forecast horizon and improve (reduce) training time. As observed from Table 2.4, there is underreporting regarding this area as most studies (at least 60%) did not clearly provide future research areas.

Table 2.6. Key strengths and limitations of the classes wind speed/power forecasting methods (also see e.g., [77,153])

Model	Strength	Weakness	Suitable Horizon	Best input data
Physical	Medium to long-term forecasting scale effectiveness, wide spatial, and parameter range.	Computationally expensive, unsuitable for a short-term forecast horizon, the calculation are often highly complex, and limited accuracy.	Medium to long-term horizon.	NWP yield best results.
Statistical	Easy to learn and has a wide range of applications, high predictive strength (especially for short-term horizon), massive literature available, and requires fewer data modelling (e.g. ARIMA)	Cannot adequately explain nonlinearity and nonstationarity (e.g. ARMA but other such SARIMAX can capture this nonstationarity behaviour), and prior model assumption required. Reduced accuracy in medium to longer forecasting scales.	Ultra-short to short-term.	Historical time series data produce higher accuracy.
ML	Better generalisation capabilities, a wide range of applications, can capture nonlinearity effectively, high noise tolerance (RF), highly accurate (LSTM), and efficient (e.g. LGB)	Mostly data greedy, complex (i.e. require skill comprehension and application), some algorithms (e.g. LSTM) can be computationally expensive.	Ultra-short to short term.	Historical time series provides better accuracy.
Hybrid	Can effectively capture nonlinear, nonstationary signals, robust, better generalisation strengths, and outperforms individual models across varying time scales.	Might require large amounts of data, can be highly complex, and computationally expensive.	Ultra-short, short-term, medium-term, and long-term	Historical time series data produce higher accuracy.

2.4 Conclusion and Identified Research Areas

Globally, there is an urgent need to resolve complex power grid management problems due to the penetration of large volumes of complex and unpredictable wind power into the existing power grid system. In essence, accurate and reliable wind predictions will minimise the monetary and technical risks attached with the inherent unpredictability of wind power. Accordingly, this chapter has provided a thorough review and discussion of the current trends among the state-of-the-art hybrid models with specific focus on the architect, forecasting horizon, input data granularity/resolution, performance metrics, output data, the state of the LSTM network, and benchmarking. Overall, the study unmasked the limitations of the current traditional hybrids and seek to address the identified short-comings. Hence, the following conclusions could be drawn from the synthesised literature review:

- Despite the abundance of wind resources in the Southern African region, there are limited wind speed hybrid forecasting studies that have been conducted. Hence, a hybrid model that accurately predicts wind output at multiple locations within this region would be the most effective tool for informing decisions regarding future investment in wind power technology thereby enhancing the exploitation of the untapped wind power potential. Besides, a hybrid model that leverages the advantages of data pre-processing, data optimisation, and data post-processing approaches to accurately predict wind speed characteristics due to linearity, nonlinearity, and nonstationarity is very scant in the literature. *This gap is covered in Chapters 4-6.*

- In a traditional hybrid model, signals are decomposed into subsignals of different scales, individually forecasted, and then their forecasts are combined using linear combination models. This often leads to error accumulation in the final forecast. As deduced from the literature, most of the hybrid studies rely on linear combination approaches that are often incapable of capturing the nonlinearity inherent in wind signals and subsignal forecasts. Hence, there is an urgent need to prioritise nonlinear forecast reconciliation methods to improve the predictive strength and reliability of the hybrids in wind forecasting. *This gap is covered in Chapters 4-7.*

- Additionally, the literature indicates the lack of research efforts to improve wind power prediction accuracy by mitigating vanishing gradients through stateless LSTM. Much emphasis has been placed on the standard or traditional stateful LSTM which is not always best for handling weather-dependent physical quantities such as wind speed. In fact, a hybrid that employs stateless LSTM and leverages the SampEn criterion to select wavelet signals based on their analogous complexity properties to ensure that the most appropriate modelling and forecasting approach is applied to improve prediction accuracy is unavailable. *This gap is covered in Chapter 5.*

- The reviewed work showed that most of the studies did not (to a major extent) consider the potential influence of different wavelet decomposition levels on the analysis of wind speed data. Moreover, no reproducible method (with over reliance on the trial and error) was provided for selecting the optimal level of decomposition, and the concept of WTs is often not thoroughly detailed. This oversight emphasises the necessity for further research and analysis to better understand the impact of varying wavelet decomposition levels and wavelet filters in wind forecasting. *This gap is covered in Chapter 6.*

- The literature showed negligible swing towards long-term forecasting scale on the latest research papers. Overall, the main focus of the hybrid methods is on short-term forecasting scale. While these are important for maintaining the stability of the microgrid, medium-to-long-term forecasts are pivotal for assessing the economic feasibility of the integration of wind power energy into the power grid and investment in wind turbine technology. As a result, there is need for more emphasis on medium-to-long-term forecasting to improve the integration of wind power into power grids. *This gap is covered in Chapter 4 and Chapter 6.*

- Probabilistic predictions can improve electricity network planning by accurately quantifying fluctuations and uncertainties in wind data. Furthermore, a probabilistic forecasting system provides complete information about intermittent wind speed behaviour, which is crucial for assessing uncertain situations, and planning various strategies. However, these predictions are

under-reported in the literature, making probabilistic predictions of wind speed a top priority. *This gap is covered in Chapter 4-6.*

Despite researchers' efforts in developing several hybrid methods to improve wind forecasts, there seems to be very minimal progress away from the traditional hybrid model as seen in the synthesis provided above. In fact, many research areas (as shown above) and questions remain unknown and unanswered, which lays an agenda for current research work.

This page is intentionally blank

Chapter 3

Research Methodology

3.1 Introduction

This chapter gives an in-depth description of the WT-ML hybrid modelling framework in wind speed prediction. As a foundational starting point, models used which include the Box-Jenkins ARIMA models, neural networks, information theory, variable selection and regularisation methods, metaheuristic algorithms, boosting decision trees, bagging methods, adaptive boosting methods, and support vectors on Southern Africa wind speed data and South African power grid data are briefly described. In essence, the chapter is divided into six main sections: baseline models, feature engineering models, main models, proposed hybridisation framework, performance evaluation metrics, and conclusions. Section 3.2 discusses the baseline models used to benchmark the predictive performance of the proposed hybrid frameworks. These include, but are not limited to, the ARIMA models, NNAR models, Naïve or Persistence models, and vector autoregressive (VAR) models. In Section 3.3, the feature engineering strategies, including MODWT, SampEn, and LASSO, are thoroughly outlined. DE optimisation algorithms are also discussed. Section 3.4 presents the core models, which include the highly accurate stateless LSTM, GRU, XGBoost, light gradient boosting machine (LGB), GBM/SGB, RF, AdaBoostRT, SVR, and RVM. These played a key role in the development and building of the proposed hybrid frameworks, viz., WT-ARIMA-XGBoost-SVR (see Chapter 4), WT-NNAR-LSTM-GBM (see Chapter 5), wavelet-MODWT-GRU (MB) (see Chapter 6), and RVM-WT-AdaBoostRT-RF (Chapter 7). Section 3.5 outlines the primary processes for developing these strategies, the rationale behind each model, and its contributions. Towards the end, the chapter discusses deterministic and probabilistic performance indicators in Section 3.6. Concluding remarks are provided in Section 3.7. Full details on these methodologies are provided in the published work of [15,22,28,29].

3.2 Baseline Models

It is pivotal to employ baselines when evaluating forecasts. In fact, comparison with appropriate benchmarks, especially simpler ones, is crucial. There is, however, a lack

of rigorous comparisons between new algorithms and the simpler or relevant benchmarks in the forecasting literature [178] (also see Chapter 2). Among many models, the following models were used as baselines for evaluating the efficacy of the proposed hybrid frameworks.

3.2.1 Naive Model

The Naive model assumes that the forecasted wind speed at the time "$\Delta t + t$" is equal the most recent observed value at time "t" [83]. The Persistence model is often utilised as a benchmark model when working with meteorological data, and it yields accurate results for very-short and short-term forecasting horizons. The Persistence or Naive model assumption is given by:

$$\hat{y}_{\Delta t+t|t} = y_t. \tag{3.1}$$

Thus, this model offers no sophisticated theory and has been widely used as a benchmark model.

3.2.2 Autoregressive Integrated Moving Average

ARIMAs are statistical techniques for modelling both nonstationary and stationary univariate time series and were introduced and advanced by [82]. These statistical approaches are the most commonly employed in time series forecasting and are often reserved for short-term forecasting [82-87]. In general, the multiplicative seasonal ARIMA $(p,d,q)(P,D,Q)s$ which account for seasonality is expressed by the equation below:

$$\phi_p(B)\Phi_P(B^s)(1-\mathrm{B})^d(1-B^s)^D y_t = C_y + \theta_q(B)\Theta_Q(B^S)e_t, \tag{3.2}$$

where $\phi_p(B), \theta_q(B)$, $\Phi_P(B^s), \Theta_Q(B^S)$ respectively, represents the non-seasonal autoregressive (AR), non-seasonal moving average (MA), seasonal AR (SAR), and seasonal MA (SMA) components. The seasonal and non-seasonal differences are respectively denoted by positive integers D and d, B is the back shift operator such that $B^k y_t = y_{t-k}$, and $S \in \mathbb{N}\backslash\{0,1\}$ is the seasonality. The powers, P, p, Q, and q respectively denote the SAR, non-seasonal AR, SMA, and non-seasonal MA orders, while the parameters $\Phi_i, i \in \{1,\dots,P\} \subset \mathbb{N}\backslash\{0\}$; $\phi_j, j \in \{1,..,p\} \subset \mathbb{N}\backslash\{0\}$; $\Theta_m, m \in \{1,\dots,Q\} \subset \mathbb{N}\backslash\{0\}$, and $\theta_k, k \in \{1,\dots,q\} \subset \mathbb{N}\backslash\{0\}$ are real numbers. The residuals are assumed to be white noise such that $e_t \sim N(0,\sigma^2)$. It is primarily the autocorrelation function (ACF) along with the partial autocorrelation function (PACF) used to determine the evolution of a stochastic process which include stationarity, short/long

memory, and model identification (see e.g., [82-87]). The conventional non-seasonal ARIMA (p, d, q) does include the seasonal components $P, D, Q,$ and s.

3.2.3 Neural Network Autoregression

The NNAR approach, which is non-parsimonious and belongs to the class of FFNNs, is implemented in a recursive manner. This MLP network consists of an input, hidden, and output layers alongside an activation function that regulates the effect of outliers on predictions [28,83,86]. A typical MLP is denoted by the following mathematical expression:

$$y_t = g\left(\sum_{j=0}^{z} {w_{tj}}^{(2)} f\left(\sum_{i=0}^{u} x_i \,.\, {w_{ij}}^{(1)} + {b_j}^{(1)}\right) + {b_t}^{(2)}\right), \tag{3.3}$$

where u and z denote the number of neurons in the hidden layer, f denotes an activation function of the hidden layer, g represents the activation function of the output layer, ${w_{ij}}^{(1)}$ denotes the weight from the i^{th} input to the j^{th} hidden neuron, ${b_j}^{(1)}$represents the bias for the j^{th} hidden neuron, ${w_{tj}}^{(2)}$ represents weight from the j^{th} hidden neuron to the t^{th}output neuron, and ${b_t}^{(2)}$ represents the bias for the t^{th} output neuron.

For an NNAR approach, stochastic weights are firstly assigned and iteratively updated with training observations. Additionally, the NNAR approach utilises lagged observations to generate step-ahead predictions [28,83,86]. The NNAR approach consisting of input ($\{y_{t-1}, y_{t-2}, \ldots, y_{t-p}\}$) and output ($y_t$) can be calculated by the following mathematical expression:

$$y_t = w_0 + \sum_{j=1}^{z} w_j . f\left(w_{0j} + \sum_{i=1}^{u} y_{t-i} \,.\, w_{ij}\right) + b, \tag{3.4}$$

where z denotes the number of input nodes; u denotes number of hidden nodes, $w_{ij} (i = 0,1, \ldots, u;\ j = 0,1, \ldots, z)$, $w_i\ (i = 0,1, \ldots, u)$; f is an activation function, and b is the bias term.

3.2.4 K-Nearest Neighbour

K-Nearest Neighbour (KNN) is a robust non-parametric ML method widely used for classification or regression problems [179,180]. KNN involves three steps, viz.;

determining the distance between the training and test datasets; choosing the nearest neighbour with the minimal distance; and prediction of wind speed values based on the weighted approach. The study employed Euclidean distance (ED) given by:

$$\text{ED} = \sqrt{\sum_{i=1}^{n}(T_i^j - \varrho_i)^2}, \tag{3.5}$$

where T_i^j is the j^{th} training instance from a vector $\tilde{V} = (T_1^j, T_2^j \ \ldots, T_n^j)$ of length n, and ϱ_i denotes a new instance from the vector of new instances $\tilde{\varrho} = (\varrho_1, \varrho_2, \ldots, \varrho_l)$ whose target is unknown but whose features are known (also see Algorithm 3.1 below). These algorithms can be slow when dealing with very large dataset, thus computationally expensive.

Algorithm 3.1: KNN

1. Compute the ED (y, y_i)
2. Sort the n distances in ascending order.
3. Given $k \in \mathbb{Z}_+$ select the first k smallest distances.
4. Identify the k data points corresponding to these distances.
5. Let k_i denote the number of points belonging to class i^{th} among the k^{th} neighbors.
6. If $k_i > k_j$ for all $i \neq j$ then put y in class i.

3.2.5 Vector Autoregressive Models

The VARs are simple data-driven methods capable of handling high-dimensional time series and can effectively capture structural changes [83]. Nonetheless, these models have shortcomings, such as reduced interpretability of coefficients due to many variables; selection of lag highly influences model performance; number of parameters increases with dimension; and in high-dimensional spaces, sparsity is required to mitigate multicollinearity [83]. A typical VAR (p) model is denoted by:

$$y_{i,t} = \tau_i + \sum_{k=1}^{p} \phi_{i1,k}, y_{1,t-k} + \sum_{k=1}^{p} \phi_{i2,k} y_{2,t-k} + \cdots + \sum_{k=1}^{p} \phi_{in,k} y_{n,t-k} + e_{i,t} \tag{3.6}$$

where $\tau_i (i = 1{:}\, n)$ are the constants or intercepts terms of the i^{th} time series; $y_{i,t}$ $(i = 1{:}\, n)$ denotes the i^{th} time series at time t; p represents the maximum lag for the model; $\phi_{ij,k}$ is the effect of region j on region i with a lag of k time points, $e_{i,t}$ $(i = 1{:}\, n)$ is the uncorrelated noise or residual terms is the error term for the i^{th} time series at time t. The VAR model is fitted through the least squares approach, where parameters are calculated by solving the sum of squares for each equation. These approaches work best with stationary time series data; otherwise, data must be transformed (commonly via "VAR" differencing) to stationarity.

3.3 Feature Engineering Methods

3.3.1 Maximal Overlap DWT

The WT effectively and efficiently extracts significant details at the same time filtering noise and random trends from the signals (see details in Chapter 2). The non-orthogonal MODWT (which is a class of the orthogonal and time-variant discrete DWT), deconstructs signals into asymptotic wavelet coefficients, and is redundant and well-defined for all sample sizes [181–184]. Furthermore, time-invariant MODWT employs zero-phase-filter-associated coefficients that are time-inverse which ensures that variations in time series data do not distort the signal structure [182–184]. Hence, the MODWT decomposition of the original signal y_t constructed from the vectors corresponding to the j^{th} order vectors of the low-pass $\widetilde{G_{j,l}}(.)$ and high-pass $\widetilde{H_{j,l}}$ (.) filters yield the following respective scaling and detailed expressions:

$$\tilde{\varsigma}_{jt} = \sum_{l=0}^{L_j-1} \tilde{h}_{j,l} y_{t-l} \ mod\ N\ t \in [0, N-1] \tag{3.7}$$

and

$$\tilde{s}_{jt} = \sum_{l=0}^{L_j-1} \tilde{g}_{j,l} y_{t-l}\, mod\ N\ t \in [0, N-1], \tag{3.8}$$

where two functions $\widetilde{G_{j,l}}(.)$ and $\widetilde{H_{j,l}}$ (.), which form functions that are quadrature mirrors, are of length $L_j = (2^j - 1)(L-1) + 1$, where L denotes the filter at $j = 1$. Without loss of generality, the original signal y_t can be reconstructed using the equation below:

$$y_t = \sum_{k=1}^{J} D_k + A_J \tag{3.9}$$

where $D_k = \sum_{l=0}^{N-1} \widetilde{h^p}_{k,l} \tilde{\varsigma}_{k,t+l}$ and $A_J = \sum_{l=0}^{N-1} \widetilde{g^p}_{j,l} \tilde{s}_{j,t+l}$, with $\widetilde{h^p}_{j,l}$ and $\widetilde{g^p}_{j,l}$ being the respective periodised $\tilde{h}_{J,l}(.)$ and $\tilde{g}_{j,l}(.)$ to size N. The detailed coefficients are calculated for each level, whilst the approximate coefficients are computed for the $J \leq \text{int}(\log_2(N))$. The expression above is analogous to the DWT-based multi-resolution analysis. The characteristics of the filters employed in this research work are outlined in Table 3.1 below.

Table 3.1. Characteristics of wavelet filters applied in the current study.

DB4	LA8	MB8	Citation
• Can (to some extent) capture sharp transition features in the dataset.	• LA8 yields better results when handling high-variant data.	• Highly capable of handling high-frequency and transient features in wind data.	[57,69,183, 185-190]
• Compactly supported. Offers good localisation features but spectral leakage can be a problem when dealing with highly non-stationary data.	• Compactly supported. Good localisation and reconstructs signals much better. Excellent for handling nonstationary data.	• Offers excellent localisation properties alongside minimal spectral leakage.	[57,69,183,185-190]
• DB4 are asymmetric and have a non-linear phase response. Signal distortion is possible due to inadequate boundary handling (compared to LA8)	• LA8 are least asymmetric (compared to DB4) and has linear phase response. Signal distortion is possible, but minimal. Adequately handles boundaries.	• Despite a lack of symmetrical properties, MB8 is less prone to signal distortion due to narrow bandwidth.	[57,69,183,185-190]
• DB4 is orthogonal and has (to some extent) good energy preservation properties. Efficient perfect reconstruction.	• Offer optimal trade between orthogonality and symmetry. Also possesses efficient perfect reconstruction	• Orthogonal and provides a great time-frequency trade-off. Efficient computation and perfect reconstruction.	[57,69,183,185-190]

In Table 3.1, the DB performs optimally on signals with abrupt transitions, the LA delivers a balanced performance for mixed signals, and the MB delivers better results when handling high frequency signals whilst minimising spectral leakage.

3.3.2 Sample Entropy

The SampEn is intended to quantify or identify the complexity of the sequential data without pre-existing knowledge or information of the source generating the dataset (see e.g., [109, 191-195] for details). Given sequential data of length n, $SampEn\ (m, r, n)$ can be expressed as the negative logarithm of the conditional probability that two sequential data points are identical over m points given the tolerance threshold r, excluding any self-matches (see e.g., [109, 191-195] for details). Thus,

$$SampEn\ (m, r, n)\ = -ln\ \left(\frac{A^*}{B^*}\right), \tag{3.10}$$

where A^* and B^*denote the number of template vector pairs given by: $A^* = d[\boldsymbol{y}_{m+1}(i), \boldsymbol{y}_{m+1}(j)] < r$ and $B^* = d[\boldsymbol{y}_m(i), \boldsymbol{y}_m(j)] < r$, respectively. Furthermore,

$$d[\boldsymbol{y}_m(i), \boldsymbol{y}_m(j)] = \max_k\{|y_{i+k}, y_{j+k}|\} \leq\ r, \tag{3.11}$$

where $k \in [0, m-1]$ and $r \approx 0.2\sigma_{y_t} \geq 0$ with $\sigma_{y_t}(.)$ representing the standard deviation of the signal y_t. The expression $max\{||.||\}$ denotes the Chebyshev norm given by:

$$max\{|y_{i+k}, y_{j+k}|\} = \underset{0 \leq k \leq m-1}{max}\{||y_{i+k} - y_{j+k}||\}. \tag{3.12}$$

The SampEn algorithm produces higher values when the tolerance r is small, indicating that a smaller r corresponds to a more random series. Conversely, a larger r relaxes the similarity condition, allowing more patterns to be recognised and increasing the likelihood that the distance between sequences is less than r. Essentially, SampEn decreases monotonically as r increases. Time series with high SampEn values exhibit a low probability of repeated sequences, indicating lower regularity and greater complexity [191]. Often, SampEn ranges within 0 and 1, though it may exceed 1 based on the number of classes in the dataset under study. Yet, in this context, the meaning of entropy stays the same, with values more than 1 indicative of chaotic series, unstable patterns, abrupt spikes, and turbulence characteristics [194]. A lower SampEn value indicates low randomness or complexity, whereas a higher SampEn value indicates high complexity.

3.3.3 Least Absolute Shrinkage and Selection Operator

LASSO encourages sparse models and performs both variable selection and regularisation to increase the interpretability and prediction accuracy of the model [196]. Furthermore, these models can handle data characterised by multicollinearity (see [196] for details). The LASSO approach seeks to solve the loss function denoted by the Lagrangian form below:

$$\underset{\boldsymbol{\beta}^* \in \mathbb{R}^p}{min} \left\{ \frac{1}{2 \times N} \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{p} x_{ij} {\beta_j}^* \right)^2 + \lambda^L \sum_{j=1}^{p} |{\beta_j}^*| \right\}, \tag{3.13}$$

where N pair of predictor variables ($\boldsymbol{X}$) and response variables ($\boldsymbol{Y}$) are denoted by $\{(x_{ij}, y_i), i = 1{:}N; j = 1{:}p\}$, p denotes the number of predictor variables, $\boldsymbol{\beta}^* = \left({\beta_1}^*, {\beta_2}^*, \dots, {\beta_p}^*\right)$ are the regression weights such that $\sum_{j=1}^{p} |{\beta_j}^*| \leq \eta$, $\eta > 0$ is a tuning parameter controlling sparsity, and λ^L (controlling the amount of shrinkage) is a tuning parameter. If $\lambda^L = 0$, we revert to the ordinary least squares equation. The larger values of λ^L, the more the coefficients are shrunk towards zero. As a result, the model will under-fit the data. For an increasing λ^L, the coefficients are set to zero and eliminated, thereby leading to model bias. A decreasing tuning parameter λ^L results in a variance increase. LASSO seeks to strike the best balance between model over-

fitting and sparsity by avoiding the extreme values of the constraint η. As a result, the coordinates of the lesser significant features are truncated towards zero according to LASSO regression, while statistically insignificant features are entirely eliminated.

3.3.4 Differential Evolution

DE is a stochastic algorithm that can be applied to complex optimisation problems without explicit adaptation to each particular problem [197,198]. Furthermore, DE is effective and efficient at handling non-differentiable and nonlinear continuous functions [197-199]. Accordingly, DE is robust, efficient, reliable, and scalable than artificial bee colony (ABC), simulated annealing (SA), GA, and PSO (also see [197-199]). In spite of this, parameter settings and mutation strategies significantly affect the performance of the DE algorithm. DE's implementation involves four main steps, viz.,

Step 1: Initialisation

The DE starts by initialising the population such that the initial population $R_{ij} \in P^{\tau} \subseteq \mathbb{R}^{D \times NP}$ with random initialisation $L \leq R_{i,j} \leq U$ is given by [197-198]:

$$R_{i,j} = L + (U - L)\, rand_{ij}(0,1], j = 1, 2, \ldots, D;\ i = 1, 2, \ldots, NP, \quad (3.14)$$

where L and U, respectively, denote the lower and upper limit, NP is the population size, D denotes the number of variables, and $rand_{ij}(0,1]$ is a uniform distribution. Thereafter, for each generation, the algorithm will undergo three main three evolutionary processes, viz., mutation, crossover, and selection to reach global minimum.

Step 2: Mutation

In this step, a mutant or donor vector denoted by $\mathbf{\Gamma}_i$ is produced through scale differencing distances any two of the three vectors and then adding to the third one [198,199] yielding the following expression:

$$\Gamma_{i,g} = y_{r1,g} + \lambda^s (y_{r3,g} - y_{r2,g}), r1 \neq r2 \neq r3 \quad (3.15)$$

where $\Gamma_{i,g}$ represents a mutant vector for the i^{th} individual in the next generation (g); and $y_{r1,\mathrm{g}}, y_{r2,\mathrm{g}}, and\ y_{r3,\mathrm{g}}$ are randomly and independently selected from the population generation $g \in [0,1, \ldots, G_{max}]$; and λ^s denotes the scaling factor.

Step 3: Crossover

The crossover operator creates offspring by mixing components of the current element and those generated by mutation [198–200]. Fundamentally, the binomial crossover parameter combine either the characteristics of the mutant ($\boldsymbol{\Gamma}_i$) and target vector ($\boldsymbol{y}_i$) to develop a trial vector strategy ($\boldsymbol{U}_i$) such that [197-202]

$$U_{i,j,g} = \begin{cases} \Gamma_{i,j,g} \; if \; r_{ij} \leq P_{cr} \; or \; j = j_{rand}, \\ . \\ y_{i,j,g} \;\; otherwise \, , \end{cases} \tag{3.16}$$

where $r_{ij} = rand_{ij}(0,1]$ are uniformly distributed random numbers generated for each j and P_{cr} represents the crossover probability. The parameter j_{rand} is vital as it assures that $\Gamma_{i,j,g} \neq y_{i,j,g}$.

Step 4: Selection

Finally, $U_{i,g}$ is compared with the target $y_{i,g}$, and the best value is carried into the next generation [200-202] such that

$$y_{i,g+1} = \begin{cases} U_{i,g}, \text{if } \mathcal{F}(U_{i,g}) \leq \mathcal{F}(y_{i,g}), \\ . \\ y_{i,g} \;\; \text{otherwise} \, , \end{cases} \tag{3.17}$$

where $\mathcal{F}$ is the objective function. If the new trial vector is the same or smaller than the target vector, it becomes the new target. If not, the target vector stays the same, thereby ensuring that the population remains the same or improves [197-203]. The mutation, crossover and selection process repeats until termination condition is satisfied (e.g. maximum iteration) (see Algorithm 3.2). Also see Appendix A.

Algorithm 3.2: Standard DE

1. Initialise algorithm within the limits [L, U]
2. Set and evaluate the objective function
3. Until stopping criterion is reached (i.e. maximum iteration), repeat the following process
    i. For each individual
        - Mutate to generate a donor vector.
        - Create a trial solution via crossover
        - Evaluate the trial and target and preserve the superior one
4. Preserve the optimal solution ($\boldsymbol{y}_{best}$)

3.4 Main Predictive Models

3.4.1 Long Short-Term Memory Networks

Sequence modelling algorithms within LSTMs (which is a variant of RNN) can be divided into stateful or stateless depending on the training configuration adopted [118,119]. A stateful LSTM preserves information across batches, which is crucial for capturing long-term dependencies on the input data. On the other hand, the stateless LSTM is characterised by an internal state that resets after each batch such that each unit is processed separately or independently [118,119]. It is pivotal to note that statelessness is between sequences, not within batches or sequences such that dependencies and useful data in the sequences are memorised or preserved [119] since the memory unit is still fully functional and effective within the sequences. A stateless LSTM can learn patterns in unstable random time series data, such as wind speed, more effectively and accurately than a stateful LSTM. In addition, stateless LSTMs are more stable, simpler, and accurate than stateful LSTMs [28,118]. The typical (stateful) LSTM unit constitutes an input node ($\boldsymbol{g}_t$), input gate ($\boldsymbol{i}_t$), output gate ($\boldsymbol{o}_t$), and forget gate ($\boldsymbol{f}_t$) as shown in the equations below (also see e.g., [93-95]) (also see Figure 3.1):

$$\boldsymbol{f}_t = \sigma^f(\boldsymbol{W}_f[\boldsymbol{h}_{t-1}, \boldsymbol{y}_t] + \boldsymbol{b}_f),\ \boldsymbol{f}_t \in [0,1], \tag{3.18}$$

$$\boldsymbol{i}_t = \sigma^f(\boldsymbol{W}_i[\boldsymbol{h}_{t-1}, \boldsymbol{y}_t] + \boldsymbol{b}_i),\ \boldsymbol{i}_t \in [0,1], \tag{3.19}$$

$$\boldsymbol{g}_t = \varphi^f(\boldsymbol{W}_g[\boldsymbol{h}_{t-1}, \boldsymbol{y}_t] + \boldsymbol{b}_g),\ \boldsymbol{g}_t \in [-1,1], \tag{3.20}$$

$$\boldsymbol{o}_t = \sigma^f(\boldsymbol{W}_o[\boldsymbol{h}_{t-1}, \boldsymbol{y}_t] + \boldsymbol{b}_o),\ \boldsymbol{o}_t \in [0,1], \tag{3.21}$$

$$\boldsymbol{s}_t = \boldsymbol{s}_{t-1} \odot \boldsymbol{f}_t + \boldsymbol{g}_t \odot \boldsymbol{i}_t, \tag{3.22}$$

$$\boldsymbol{h}_t = \varphi^f(\boldsymbol{s}_t) \odot \boldsymbol{o}_t, \tag{3.23}$$

where $\boldsymbol{W}_f$, $\boldsymbol{W}_i$, $\boldsymbol{W}_g$, and $\boldsymbol{W}_o$ are the weight matrices connecting the input signal $[\boldsymbol{h}_{t-1}, \boldsymbol{y}_t]$, where $\boldsymbol{h}_{t-1}$ is the previous cell output, and $\boldsymbol{y}_t$ denotes the input vector. The vectors $\boldsymbol{b}_f, \boldsymbol{b}_i, \boldsymbol{b}_g$, and $\boldsymbol{b}_o$ are bias vectors. The $\sigma^f(.)$ is the logistic sigmoid activation function, $tanh$=$\varphi^f(.)$ is the tangent hyperbolic function, and $\boldsymbol{s}_t$ is the memory cell state that recalls historical information over arbitrary time intervals. The $\sigma^f(.)$ and the $\varphi^f(.)$ are respectively denoted by:

$$\sigma^f(\boldsymbol{y}_t) = \frac{1}{1 + e^{-\boldsymbol{y}_t}} \in [0,1], \tag{3.24}$$

and

$$\varphi^{f}\ (\boldsymbol{y_t}) = \frac{e^{\boldsymbol{y_t}} - e^{-\boldsymbol{y_t}}}{e^{\boldsymbol{y_t}} + e^{-\boldsymbol{y_t}}} \in [-1,1]. \tag{3.25}$$

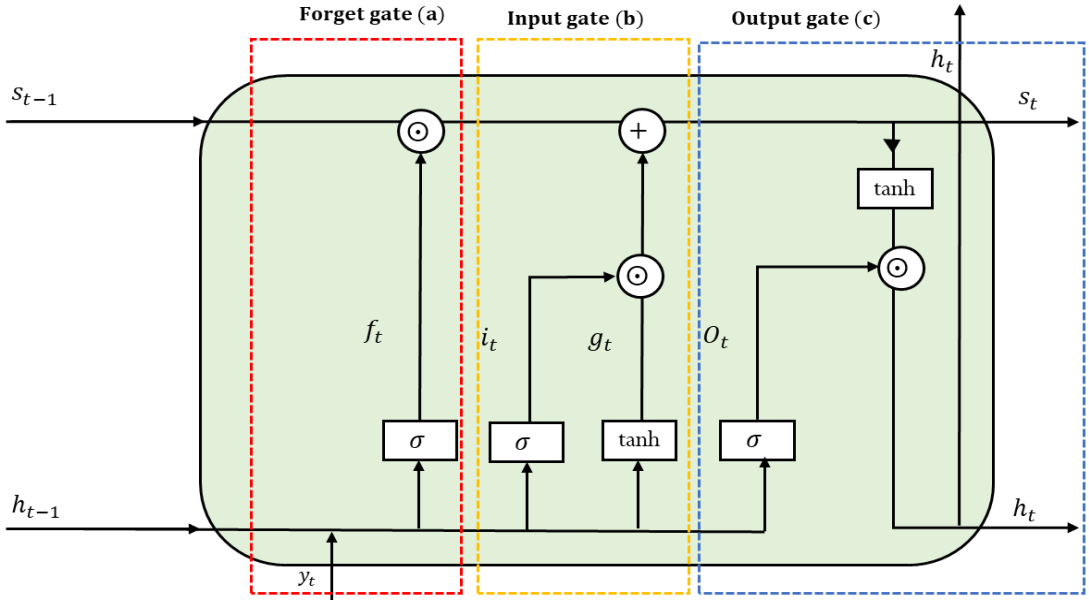


Figure 3.1. Conventional stateful LSTM cell.

Below a brief discussion on the main LSTM network gates as indicated in Figure 3.1 above is provided. *The forget gate* removes unnecessary information from a memory cell state by multiplying $\boldsymbol{y}_t$ and $\boldsymbol{h}_{t-1}$, by weight matrices before bias is added. The resulting binary value is sent via an activation function, with the output either 0 (lost) or 1 (stored for future use), depending on the cell state. This process involve Equation (3.18).

The input gate gives important information to the memory cell state by regulating it with a sigmoid function and filtering data with a forget gate. The $tanh$, which outputs values between -1 and 1, regulates the nonlinearity of the network**.** This function ($tanh \in (-1,1)$) generates a vector containing all possible values from $\boldsymbol{h}_{t-1}$ and $\boldsymbol{y}_t$. The vector and regulated values are multiplied to generate meaningful data. The candidate values are updated by combining $\boldsymbol{g}_t \odot \boldsymbol{f}_t$ and the current state $\boldsymbol{i}_t \odot \boldsymbol{g}_t$ such that $\boldsymbol{s}_t = \boldsymbol{s}_{t-1} \odot \boldsymbol{f}_t + \boldsymbol{g}_t \odot \boldsymbol{i}_t$. This process involve Equations (3.18) to (3.20).

Output gate employs the sigmoid function to determine the part of data or information from the current cell state should be realised as output. Thus, this gate extracts information from the current state of the cell by creating a vector using tanh $\in (-1,1)$ function, controlling it using sigmoid function by filtering it with $\boldsymbol{h}_{t-1}$ inputs and $\boldsymbol{y}_t$

inputs, multiplying the vector and controlled values to output and input the next cell. This process involve Equations (3.21) and (3.23). Also see the work of [93,94,95, 117,118,120,121] for more details on the gates system.

With the introduction of a gate mechanism that controls the flow of information through the network, LSTM addresses the vanishing gradient problem, a common limitation of the traditional neural networks, which makes it well-suited for handling sequential or time series data with long-term dependencies. However, LSTM networks are known to be computationally expensive and time-consuming, making them less suitable for resource constrained environments.

3.4.2 Gated Recurrent Units

Different from LSTMs, GRUs are characterised by simpler structural architecture, high convergent speed, less or reasonable computational time, and (to some extent) better generalisation abilities [95,204,205]. Comparable to LSTM, GRU (which also mitigates vanishing gradients), interpolates the previous hidden state ($\boldsymbol{h}_{t-1}$) and candidate hidden state ($\widetilde{\boldsymbol{h}}_t$) to build an activation function such that [204]

$$\boldsymbol{h}_t = \boldsymbol{\zeta}_t \odot \boldsymbol{h}_{t-1} + (1 - \boldsymbol{\zeta}_t) \odot \widetilde{\boldsymbol{h}}_t, \tag{3.26}$$

where $\boldsymbol{\zeta}_t$ denotes an update gate which regulates how much information from the hidden state should be retained and is given by:

$$\boldsymbol{\zeta}_t = \sigma^f(\boldsymbol{W}_\zeta \boldsymbol{y}_t + \boldsymbol{U}_\zeta \, \boldsymbol{h}_{t-1}), \tag{3.27}$$

where $\boldsymbol{h}_{t-1}$ denotes the hidden state from the previous time step, $\boldsymbol{W}_\zeta$ and $\boldsymbol{U}_\zeta$ represents weight matrices, $\sigma^f(.)$ defined as an activation function. It should be noted that $\boldsymbol{\zeta}_t \to 1$ will retain information from $\boldsymbol{h}_{t-1}$, otherwise will retain information $\boldsymbol{h}_t$. The candidate activation ($\widetilde{\boldsymbol{h}}_t$) is given by:

$$\widetilde{\boldsymbol{h}}_t = tanh(\boldsymbol{W}\boldsymbol{y}_t + \boldsymbol{U}(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1})), \tag{3.28}$$

where the reset gate $\boldsymbol{r}_t$ is given by the equation below which is analogous to that used by the update gate:

$$\boldsymbol{r}_t = \sigma^f(\boldsymbol{W}_r \boldsymbol{y}_t + \boldsymbol{U}_r \, \boldsymbol{h}_{t-1}), \tag{3.29}$$

where $\boldsymbol{W}_r$ and $\boldsymbol{U}_r$ represents weight matrices corresponding to the reset gate. The results above regulate the extent to which the previous hidden state is disregarded when calculating the new candidate's hidden state [204]. Although GRUs can capture

temporal trends and dependencies in varying datasets, they can fall behind LSTMs in long-term dependency tasks involving complex sequential data (also see [204]).

3.4.2 Gradient Boosting Decision Trees

As a step-wise ensemble method, GBDTs utilise data from historically developed weak classifiers to enhance model performance [96–100,132,206,207]. The prediction is achieved by splitting the training data, utilising each portion to train various models, and finally aggregating the predictions [97].

3.4.2.1 Gradient Boosting Machine

GBMs are widely employed ML approaches that build an ensemble of shallow and weak successive trees, where each successive tree improves upon the errors of its predecessor. The algorithm is founded upon three key elements: weak learners, an additive model, and a loss function [207,208]. The GBM is fitted additively to the model such that it is a stage-wise process. The gradient boosting tree is denoted by the following expression:

$$\mathrm{F}_n(y_t) = \sum_{i=1}^{n} D_i\,(y_t), \tag{3.30}$$

where $D_i\,(y_t)$ denotes a decision tree and n represents the total number of trees. The trees are built or developed sequentially such that the $(n+1)^{th}$ decision tree can be calculated by the expression below:

$$\arg\min_{D_{n+1}} \sum_t \mathcal{L}\,(y_t, \mathrm{F}_n(y_t) + D_{n+1}\,(y_t)), \tag{3.31}$$

where $\mathcal{L}(.)$ is a differential loss function. The gradient descent is employed to solve equation above. Although GBM is flexible and accurate, it is also computationally inefficient, requires large datasets (to guarantee accuracy), and it is susceptible to overfitting. GBM is given by (see [207,208]):

$$F(y) = \sum_{m=1}^{M} \beta_m\, \theta(y; \gamma_m), \tag{3.32}$$

where $\theta(y;\gamma_m) \in \mathbb{R}$ are functions of y characterised by the expansion parameters β_m and γ_m, which are fitted in stage-wise way to delay model overfitting. Hence, when implementing GBM, these hyperparameters are pivotal for optimal model performance: number of trees, interaction depth, and learning rate.

3.4.2.2 Extreme Gradient Boosting Machine

XGBoost algorithms are highly accurate, flexible, computationally efficient, and versatile boosting methods that enable parallel computing, effectively handle sparse data, and improve CPU performance [97,99,206,207]. XGBoost (built on CPU devices similar to LGBs) is developed by applying data greedy algorithms to the objective function; sequentially constructing decision trees leads to a complete model [97, 206,207]. This method can be considered an additive model that comprises M decision trees expressed by the mathematical equation below [207]:

$$y_i = \sum_{m=1}^{M} D_m(x_i), \quad D_m \in F, \tag{3.33}$$

where D_m is a decision tree, F denotes the function of the decision tree. This is a reliable and resilient approach characterised by quicker optimisation and learning, as it employs a regularisation approach. It is this regularisation that enables XGBoost to regulate model overfitting and it is given by [97,206,207]:

$$L(\theta) = \sum_{i=1}^{n}(L(Y_i.\, y_i) \; + \sum_{m=1}^{M} R(D_m)\,, \; \theta = (D_1, D_{2,\ldots,} D_m), \tag{3.34}$$

where L represents the loss function and R representing the regularisation function, is denoted by:

$$R(f) = \alpha' |f^{BR}| + 0.5\beta' ||\boldsymbol{w}'||^2 \tag{3.35}$$

where $|f^{BR}|$ is the number of branches; α' and β' are the regularisation terms; and $\boldsymbol{w}'$ represents a vector indicating the value of each leaf. XGBoost utilises a step-by-step forward approach to simplify model complexity [97,206,207]. Each time the model adds a decision tree, it learns a new function and its coefficients to match the residuals predicted in the last step. The XGBoost employs the following gain function to determine algorithm improvement per leaf split:

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L+\beta'} + \frac{G_R^2}{H_R+\beta'} - \frac{(G_R+G_L)^2}{H_L+H_R+\beta'}\right], \tag{3.36}$$

with

$$G_j = \sum_{j\in I_j} g_j = \sum_{j\in I_j} \partial_{\hat{y}_i} L(\hat{y}_i.\, y_i), \tag{3.37}$$

and

$$H_j = \sum_{j\in I_j} h_j = \sum_{j\in I_j} \partial^2{}_{\hat{y}_i} L(\hat{y}_i.\, y_i), \tag{3.38}$$

where G and H respectively denote the left and right score of the child. The key parameters that regulate overfitting in the XGBoost model include, but are not limited to, learning rate, maximum tree depth, and minimum child weight.

3.4.2.3 Light Gradient Boosting Machine

The LGB is a variant of GBDTs that employs a gradient-based one-sided sampling (GOSS) framework. This framework downsamples the cases based on the gradients [97,206]. This approach enables the LGB model to train and work faster compared to XGBoost [206], whilst preserving high predictive accuracy. Unlike conventional GBDTs, LGB grows a decision tree leaf by leaf, rather than checking all previous leaves for each new leaf [96,97,206]. This method is intended to reduce training time and to lower memory consumption. LGB also supports CPU learning and is highly computationally efficient when handling large-scale datasets [96,97,206]. Suppose that there is a dataset $\Omega_n = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \subset \Omega_{\mathrm{N}}, n < \mathrm{N}$, such that $\boldsymbol{x} \in \mathbb{R}^n$ and N are the number of samples, then an LGB algorithm is given by [96]:

$$L_M(\boldsymbol{x}) = \sum_{m=1}^{M} V(\boldsymbol{x}; \boldsymbol{\varepsilon}_m), \tag{3.39}$$

where $V(\boldsymbol{x}; \boldsymbol{\varepsilon}_m)$ is the a single binary regression tree, $\boldsymbol{\varepsilon}_m$ denotes the parameters of the tree, and M denotes the number of trees. In the LGB model, the overall prediction result is obtained by combining the prediction results of multiple decision trees [96,206]. Since this approach is prone to overfitting when dealing with small datasets [96,97,206], setting a optimal maximum number of tree depth parameters is essential to address this limitation [96].

3.4.3 Random Forest

RF is an ensemble flexible ML approach that relies on the aggregation of weak predictors (i.e. regression trees) to provide accurate and reliable predictions [132,133]. This approach can be employed to resolve both regression and classification problems as it can provide high levels of accuracy with minimal hyperparameter tuning. The bootstrapping or bagging technique, which reduces overfitting and bias, is employed to build each tree independently using a subset of the training data [133,209]. Each tree must be grown on an independent bootstrap sample from the training data. From all possible M variables, select m variables at random and find the most optimal split at each node. In the end, by averaging the forecasts from all trees, forecasts are computed using the equation below:

$$\hat{y} = \frac{1}{M}\sum_{m=1}^{M} D_m(\boldsymbol{x}) \tag{3.40}$$

Contrary to boosting, RF results do not gradually change the training set [132,133]. Aside from its complex architecture, RF is not prone to outliers and noise, can handle highly nonlinear interactions and missing values, and can determine important variables in the data [209]. Below is the pseudo code describing the key steps involved in the RF algorithm (see Algorithm 3.3). Also see Appendix A.

Algorithm 3.3: Random forest

1. **Input:** $\Omega_n \subset \Omega_N, n < N$, M trees

$For\ m = 1$ to M

a) Select sample (K) of size N from the training dataset

b) Built a random forest (D_m) for each node of the decision tree using the bootstrapped sample by repeatedly following these steps, until you reach a minimum node size:

i. Randomly select Z variables from the n available.

ii. Select the best variable among Z.

iii. Partition the node into two portions.

2. **Output:** Ensemble of the decision tree (D_m), $\hat{y} = \frac{1}{M}\sum_{m=1}^{M} D_m(\boldsymbol{x})$

3.4.4 AdaBoostRT Algorithm

AdaBoost is an ensemble learning boosting algorithm that aims to enhance model robustness and generalisation abilities by combining predictions from multiple weak learners [210-213]. AdaBoostRT is a type of Adaboost algorithm designed to handle regression problems. Fundamentally, AdaBoost groups the observations or values as either correct or incorrect on the basis of the ARE value, computed through the expression below:

$$\rho_i = \left|\frac{\hat{y}_i - y_i}{y_i}\right|, \tag{3.41}$$

where $\hat{y}_i$ denotes the predicted value and y_i is the actual observation. The AdaBoostRT algorithm effectively discriminates between correct and incorrect predictions by establishing a distinct threshold from the training data. Afterwards, the algorithm places greater emphasis (increased weights) on observations that are incorrectly predicted. The primary shortcoming of AdaBoostRT is that it is sensitive to the selected threshold. In fact, when the threshold exceeds 0.4 more observations are categorised as accurate and the algorithm will converge slowly [211]. Conversely, if the threshold is very small, algorithm accuracy levels will drop, leading to

diminished robustness, reliability, and stability of the ensemble [211]. The pseudo code outlining the main steps involved in the implementation of the AdaBoostRT are presented in Algorithm 3.4 (also see [211-213] for details). Also see Appendix A.

Algorithm 3.4: AdaBoostRT

1. **Input:** $\Omega_n \subset \Omega_{\mathrm{N}}, n < N$
2. Determine the number of iterations n and set threshold $\Psi \leq 0.38$.
3. The sample weight $w_{(t)i} = \frac{1}{n}, i \in (1,2,\dots,n)$ is initialised with i and t being respectively the number of the training dataset and current iteration of the algorithm.
4. Train weak predictors and determine ρ_i for each training set
5. Compute the error index ε_t such that $\rho_i > \Psi$
6. Sample weights w are updated so that iterations at $t + 1$:
   $w_{(t+1)i} = \left[\frac{w_{(t)i}}{M_i}\right]\varepsilon_t$ when $\rho_i \leq \Psi$ or $w_{(t+1)i} = \frac{w_{(t)i}}{M_i}$ when $\rho_i > \Psi$ such that $\sum w_{(t+1)i} = 1$.
7. Repeat until robust predictors and results are obtained.

3.4.5 Support Vector Regression

Along with robust kernel tricks and mathematical basis, SVRs are nonlinear approaches characterised by fast convergence rate and can effectively handle smaller data [123,150,214]. In fact, the Gauss radial basis function employed in this study demonstrates strong adaptability and high convergence across low and high dimensional spaces [124,214]. Introduced by [215], the SVR is founded on the idea of structural risk minimisation (SRM), which minimises the upper limit of generalisation error as a function of the sum of training error and confidence [150,214,215]. Consider a training dataset given by $\Omega_n = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \subset \Omega_{\mathrm{N}}, n < N$, then the regression formula can be expressed as

$$f(\boldsymbol{x}) = \sum \boldsymbol{w}_i \boldsymbol{K}_i(\boldsymbol{x}_i) + b,\ \mathrm{K}_i\colon \mathbb{R}^n \to F^s,\ w_i \in F^s, b \in \mathbb{R}, \tag{3.42}$$

where $\boldsymbol{w}_i$ denotes the weights (or support vector) estimated from the training data, b represents the threshold value, and K_i denotes nonlinear mapping functions which map the sample datasets to high-dimensional feature space F^s [123,124,214]. Following the SRM principle, the weights w_i can be determined from the sample data by solving the quadratic programming problem below [214-216]:

$$\min_{w,b,\Upsilon,\Upsilon^*} \frac{1}{2}\left|\left|\boldsymbol{w}\right|\right|^2 + \mathrm{C}\sum_{i=1}^{n}(\Upsilon_i + \Upsilon_i{}^*), \tag{3.43}$$

so that

$$|y_i - \boldsymbol{w}_i.\boldsymbol{K}_i(\boldsymbol{x}_i) - b| \leq \varepsilon + \Upsilon_i, \Upsilon_i, \Upsilon_i{}^* \geq 0, i = 1,2,\dots,n, \tag{3.44}$$

where $\|.\|$ represent a Euclidean norm, a constant positive C denotes the cost coefficient, also known as the penalty factor, and it regulates the empirical risk degree of the SVR method. The number features are denoted by n, whilst $Y_i, Y_i^* \geq 0$ represent the slack variables [214,215]. $Y_\varepsilon\ (\cdot)$ denotes the ε-intensive loss function and is given by:

$$\Upsilon_\varepsilon\,(y_i) = \begin{cases} 0, & \text{if } |y_i - f(\boldsymbol{x}_\mathrm{i})| \leq \varepsilon, \\ |f(\boldsymbol{x}_i) - y_i| - \varepsilon, & \text{otherwise.} \end{cases} \tag{3.45}$$

By resolving the optimisation problem, the resulting estimation function is given by:

$$f(\mathbf{x}) = \sum_{i=1}^{n}(\delta_i^* - \delta_i)K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) + b, \tag{3.46}$$

such that the constraints $\delta_i \geq 0$, $\delta_i^* \leq C$, $\sum_{i=1}^{n}(\delta_i^* - \delta_i) = 0$, and $K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)$ denotes the kernel function. In the current study, a Gaussian kernel function, which is employed to handle the nonlinear regression problem, is determined by the following expression:

$$K\left(\mathbf{x}_i\,, \mathbf{x}_j\right) = exp(-\frac{1}{2}\sigma^2{}_{rbf}\left|\left|\boldsymbol{x}_i - \boldsymbol{x}_j\right|\right|^2) \tag{3.47}$$

The parameters C and $\gamma_{rbf} = -\frac{1}{2}\sigma^2{}_{rbf}$ are tuning parameters that regulate the empirical risk level of the SVR method and the width of the kernel function, respectively.

3.4.6 Relevance Vector Machine

In RVM, which is a type of sparse linear model based on the hierarchical prior distribution [127,217], sparseness is achieved through the assumption of sparse distribution over the weights of the regression model [127,217,218]. The independent Gamma prior distribution is usually assumed on the weight parameters, whilst the Gamma hyperprior distributions are assumed on the variance parameters [217], resulting in a student t prior distribution over the weights thereby achieving sparseness. The RVMs have the same function form as the well-known SVMs; however, the kernel functions that form the basis of the RVMs do not have to comply with Mercer's criteria (i.e., continuous symmetric positive integral operator). Additionally, RVMs have a smaller number of relevance vectors (compared to support vectors used by the SVMs) and minimal sensitivity to hyperparameter settings [127,217-219]. Nonetheless, RVMs typically involve highly nonlinear optimisation processes [217]. The RVMs compute predictions based on the following mathematical expression (see [127,217,219] for details):

$$f(\boldsymbol{X}, \boldsymbol{w}) = \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{w}_i \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{x}_i) + \omega_0 \,, \tag{3.48}$$

where $\boldsymbol{w} = (w_1, w_2, \dots, w_N)^T$ denotes the weights of the model, $K(.,.)$ is the kernel function centred at different training data observations, and ω_0 is the bias term. The kernel function defines one basis function for each observation in the training dataset. To automatically select the right kernel at each location, the sparse component or element of RVMs prunes all irrelevant kernels [217]. Suppose that we are given a training dataset of input-output denoted by $\Theta = \{\boldsymbol{x}_n, t_n\}_{n=1}^{N}$ and assume that the outputs or targets t_n are from the model defined by the following mathematical expression [127,217,219]:

$$t_n = f(\boldsymbol{x}_n, \boldsymbol{w}) + \varepsilon_n, \tag{3.49}$$

where additive noise $\varepsilon_n \sim N(0,\ \beta^{-1})$ is a set of independent samples, and β^{-1} is the precision of the variance of the noise term. Thus, the likelihood distribution of t_n is given by the following expression (see e.g., [127,217,219] for details):

$$p(t_n|\boldsymbol{X}, \boldsymbol{w}, \beta) = \Pi_{n=1}^{N}\, P(t_n|\boldsymbol{x}_n, \boldsymbol{w}, \beta) = \Pi_{n=1}^{N}\, N(t_n|f(\boldsymbol{x}_n, \boldsymbol{w}), \beta^{-1}). \tag{3.50}$$

For each weight hyperparameter w_i, the RVM model introduces a separate hyperparameter π_i (which represents the precision of the weight parameter), such that the weight parameter will have a prior distribution concentrated around zero with the following form [127,217,219]:

$$p(\boldsymbol{w}|\boldsymbol{\pi}) = \prod_{i=1}^{N} N(w_i|0, {\pi_i}^{-1}), \tag{3.51}$$

where a vector $\boldsymbol{\pi} = (\pi_1, \pi_2, \dots \pi_N)$ and as mentioned before π_i denotes the precision of the weight hyperparameter w_i which controls the variability (or shrinkage). Since the resultant basis functions do not contribute to predictions, the final model will be sparse. Different from SVM, which uses sequential minimal optimisation (SMO), RVMs use expectation maximisation algorithms [127,217,219]. According to [219], RVM updates its hyperparameters iteratively until a threshold condition is satisfied. The pseudo presented by Algorithm 3.5 below provide the key steps involved in the implementation of the RVM. Also see Appendix A.

Algorithm 3.5: Relevance Vector Machine

1. Select an appropriate kernel function (with optimal parameters) for the training dataset $\Omega_n \subset \Omega_{\mathrm{N}}, n < N$
2. Create a design matrix $\mathbf{\Phi}$ using a kernel function.
3. Determine an appropriate convergence condition for the hyperparameters α and β.
4. Establish a threshold parameter α_χ such that $\alpha_i \to \infty$ upon reaching it.
5. Select initial values for α and β.
6. Compute the variance Σ and mean μ of the posterior distribution over the respective weights.
7. Update α_i and β.
8. For $\alpha_i > \alpha_\chi$, prune the respective weights and basis functions.
9. Steps (5) to (8) are repeated until the convergent condition is satisfied.

Output: α and β

Table 3.2 below summarises the advantages and disadvantages of the models involved in the building of the proposed hybrid framework.

Table 3.2. Merit and demerits of the main methods blended for proposed wind forecasting hybrid framework

Category	**Model**	**Merit**	**Demerit**	**Citation**
Statistical	ARMA	Excellent in handling linearity and easy to comprehend	Failure in capturing nonlinearity.	[82-87,147,148]
ML	SVM/SVR	High convergence speed and handles small data better.	Inefficiency in handling large-scale dataset plus Kernel have to comply with Mercer's criteria	[92,101,123,125,127,150,215]
ML	XGBoost	Faster and robust model tuning, highly scalable, flexible, and versatile.	Can easily overfit small datasets, and often data greedy.	[97-99,132,206,207]
ML	LGB	High training speed, better accuracy, supports CPU learning, and highly capable of handling large-scale data.	Complex trees, data greedy, and small data overfitting	[22,96,97,206,]
ML	GBM	Highly accurate and flexible plus they can handle both categorical and numerical data.	Data greedy, overfitting, computationally expensive, and require a lot of trees for accuracy.	[22,28,100]
ML	LSTM/GRU	These deep learners are mostly robust, remedy vanishing or exploding gradient problem. Though complex by design, GRU is much simpler and efficient than LSTM while providing similar accuracy results	Require large amounts of data and can be computationally expensive.	[88,90,93,94,95,118-120]
ML	NNAR	Captures nonlinearity and are efficient (to some extent).	Have slow convergence rate and can overfit unstable and smaller data. Likely to suffer from gradient explosions when	[28,83,86]

Category	**Model**	**Merit**	**Demerit**	**Citation**
			dealing with high variant data (e.g. wind data).	
Signal processing /Filters	WT (MODWT)	Better capabilities in the time-frequency domain (as compared to FTs). Handles non-stationary and noise very well.	Difficult to determine optimal decomposition level.	[125,152,181-190]
Information Theory	SampEn	Efficient (to some extent) and estimate signal complexity without any prior knowledge of the source data.	High sensitivity to input parameter choice.	[109-114]
Variable selection	LASSO	Besides being computationally efficient, LASSO is effective at regularisation, variable selection, and dimension reducibility.	Selects only a subset of correlated predictors and shrinks the rest to zero. The number of predictors selected is limited to the number of samples.	[196]
ML	RF	Through the bagging technique, RF effectively handles nonlinearities, outliers, and missing values thereby avoiding model overfitting and minimising variance. Can effectively handle both classification and regression problems.	Requires more training time than other decision tree-based algorithms. Complex compared to other decision tree-based algorithms.	[132-134,136,138,209]
ML	AdaBoostRT	Leveraging boosting, these methods enhance generalisation capabilities. Can avoid overfitting and minimise bias. Do not require a large training dataset.	AdaBoostRT's convergence speed depends on the threshold selected.	[210-213]

Category	**Model**	**Merit**	**Demerit**	**Citation**
ML	RVM	Founded on the Bayesian framework, RVMs are sparse, probabilistic and require fewer support vectors. Handles high-dimensional data well, offering greater generalisation, and preventing overfitting (high variance). Does not have to comply with Mercer's criteria and performs very well on smaller datasets.	Requires more training time for large datasets.	[127,217-219]
Metaheuristic/ Evolutionary algorithm	DE	DEs are robust and efficient optimisation algorithm. Can effectively solve non-linear and non-differentiable	High sensitivity to parameter setting.	[197-203]

3.5 Hybrid Modelling Process Flow

The combination of various forecasting methods, including but not limited to statistical methods, wavelets, and ML models, is referred to as hybrid methods. The rationale behind hybridisation is to enhance overall prediction accuracy by leveraging the strengths of each predictive approach.

3.5.1 WT-ARIMA-XGBoost-SVR

The originality of the proposed WT-ARIMA-XGBoost-SVR ensemble method are premised on the basis that wind speed is characterised by inherent linearity, nonlinearity, and nonstationarity phenomena that cannot be simultaneously captured by a single class of models. Hence, the approach leveraging an ensemble of stochastic methods, wavelets, and GBDTs in wind speed prediction is briefly outlined in this subsection. Also see details in the work of [22].

Summary of the Design

The main steps involved in the development of the proposed approach are as follows (see Appendix A for details):

a) Data preprocessing: Wind speed data of interest are preprocessed to detect and handle anomalies, outliers (over 22 m/s) and missing data. In instances where outliers and missing data are found, mean imputation will be used.
b) Data partition: To preserve temporal dependencies, a chronological split was used to partition data into two sets, namely; the training set (80%) (for model training) and the testing set (20%) (preseved for final model evaluation).
c) Train and fit ARIMA model: Identify optimal parameters (using "auto.rima", ACF, and PACF) for the ARIMA. Based on the identified optimal parameters, apply ARIMA to capture the linear component and generate residuals.
d) Data decomposition: The ARIMA residuals are decomposed into less noisy subseries using level 3 & 4 MODWT. Chronologically partition the decomposed subseries into training dataset (80%) (for model training) and test dataset (20%) (preserved only for testing)
e) XGBoost training and prediction: Determine model hyperparameters using an iterative grid search with 5 fold cross validation. The hyperparameters with the lowest average RMSE were considered optimal. The search range is as follows: maximum tree depth (3-15), learning rate (0.05-0.95); and minimum child

weight (1) (see Table 4.3). To capture the nonlinear component, decomposed subseries are utilised as input features into XGBoost. Thereafter, we evaluate the model performance using the subseries test dataset based on MAE and RMSE.

f) Forecast reconciliation through SVR: Using only the training dataset, employ the search grid (using "*tune.svm*" function) to identify optimal SVR hyperparameters, namely; cost (1-50) and gamma (0.5-10), based on the least RMSE (see Table 4.3). Reconcile the ARIMA and XGBoost forecasts via SVR to form the WT-ARIMA-XGBoost-SVR.

g) Model evalauation: Evaluate final model based on MAE, RMSE, R^2 prediction interval normalised average deviation (PINAD), and PINAW using only the original wind speed test dataset.

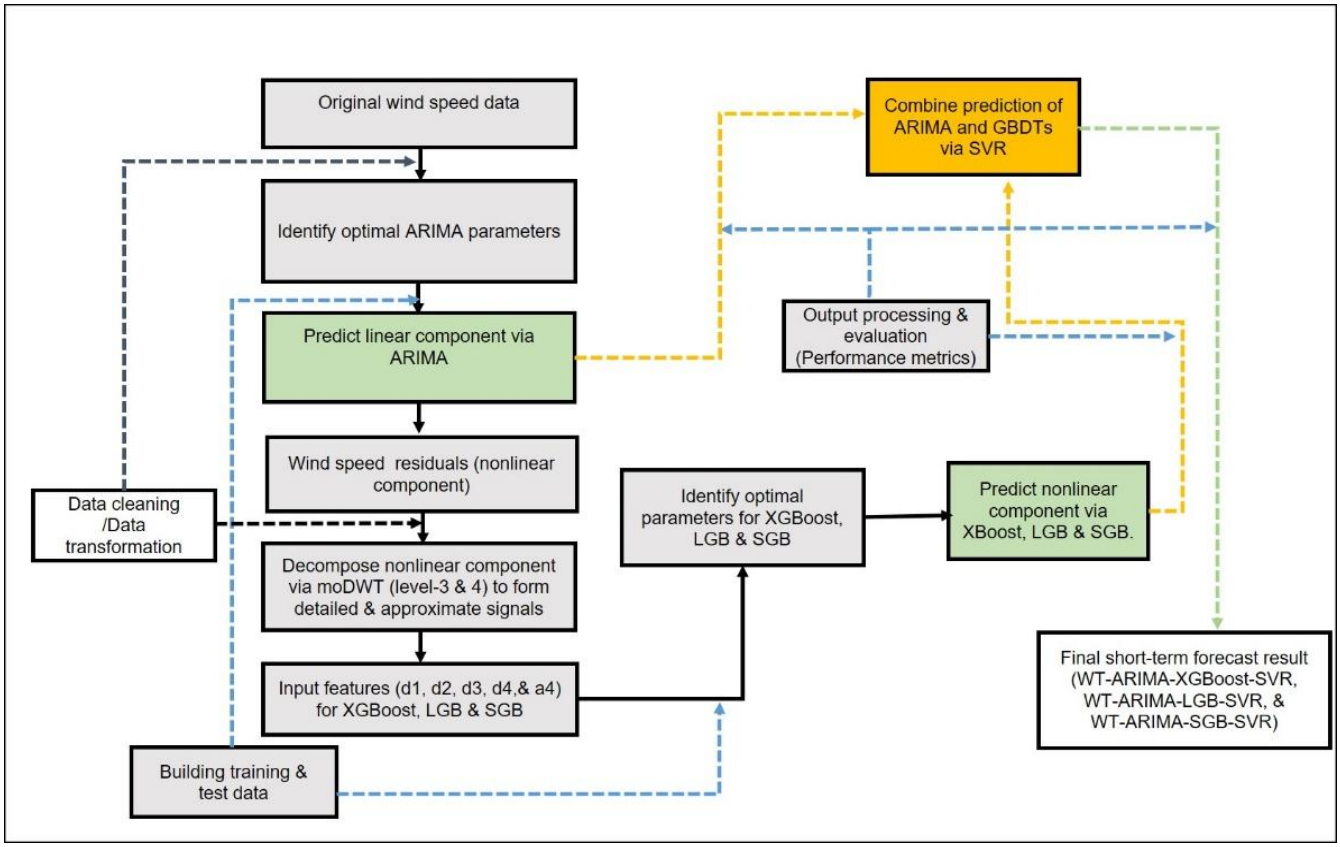


Figure 3.2. The typical WT-ARIMA-GBDTs-SVR framework for wind speed prediction

Rationale

In the literature, wavelet decomposition of a particular signal is often followed by separate modelling for each subseries using an appropriate technique. Finally, a

reconciliation (through summation) of subseries predictions will follow. Besides its simplicity, this conventional approach (summation of subseries predictions) incorporates errors from each subseries into the final predictions. Hence, compromising the accuracy and robustness of the final predictions. The proposed strategy harnesses the advantages of WT (i.e. excellent at denoising high variant signals), ARIMA (i.e. captures linearity very well), XGBoost (i.e. high accuracy, robust model tuning, highly scalable and sparse, and computational efficiency) and SVR algorithms (i.e. high convergence speed). Essentially, the proposed framework accurately and efficiently captures both linear and nonlinear components associated with wind speed turbulence and gusts thereby minimising error accumulation in final wind speed predictions.

Model Contribution

The specific role of each method involved in the proposed WT-ARIMA-XGBoost-SVR hybrid framework is summarised below (see Table 3.3).

Table 3.3. Model contribution to the proposed WT-ARIMA-XGBoost-SVR model.

Model		Contribution to the Strategy
ARIMA	✓	ARIMA was selected due to its strength in capturing linear characteristics in the wind speed data, especially at short-term forecast horizon. Besides, ARIMA models are easy to implement, flexible, and can identify trends and patterns in time series data.
MODWT	✓	Besides being an excellent time-frequency domain approach, WT is also used to deconstruct wind speed residuals as it is computationally efficient and can effectively handle nonstationary variations. Thus, the WT approach improves models' predictive performance, due to its ability to extract noise and expose patterns in the time series signal.
XGBoost	✓	To mitigate ARIMA's limitations in explaining nonlinear behaviour (e.g., wind turbulence) embedded in wind speed data, a highly robust, computationally efficient, fast, and adaptable nonlinear XGBoost model is implemented to forecast the deconstructed nonlinear wind speed residuals.
SVR	✓	Besides its fast convergence rate, SVR is selected for forecast reconciliation over a linear combination approach (e.g., direct summation), as it accounts for nonlinearity when combining forecasts, thus reducing error accumulation.

3.5.2 WT-NNAR-LSTM-GBM

The inherent complex wind speed characteristics, to an extent, can be reliably and effectively explained or captured via the implementation of special types of hybrid frameworks that simultaneously uncover complex, irregular, deterministic, and random trends in wind speed data prior to implementing appropriate predictive approaches. The aforementioned challenges founded originality in the proposed hybrid WT-NNAR-LSTM-GBM framework. Also see details in the work of [28].

Summary of the Design

The key steps involved in the development of the proposed WT-NNAR-LSTM-GBM approach are as follows (also see Figure 3.3 and Appendix A for details):

a) Data preparation: This step involved cleaning, formatting, handling inconsistences, and outliers. Observations exceeding 22 m/s are excluded due to their impracticality for wind power generation and applications. In instances where outliers and missing data are found, mean imputation will be used.
b) Data decomposition: The MODWT LA8 (at decomposition level 3) is applied to decompose the wind speed data into three high-frequency components (D1, D2, and D3) and one approximation component (A3).
c) Data complexity assessment: Data complexity is assessed using SampEn. A SampEn value of 0.9 or higher indicates a complex time series, while lower values suggest a more deterministic and less random time series sequence structure.
d) Data processing: For data processing, a min-max scaler is effected to normalise the data after to partitioning for the application of stateless LSTM approach. The Box-Cox transformation is employed to normalise and stabilise variance before superimposing the NNAR approach.
e) Data partition: Data is partitioned chronologically to preserve temporal order, resulting in a training set (for model training) comprising 80% of the data and a test set comprising the remaining 20% (preserved for model evaluation).
f) Model parameters: Model parameters for LSTM (i.e., activation function, number of layers, learning rate, optimiser, epochs, batch size, time steps, features, network dimensions) and NNAR (p, k) are identified using only the training dataset (also see Tables 5.5-5.6 for details). For NNAR (p, k), parameters are automatically selected through the "*nnetar()*" function. The LSTM parameters are iteratively tuned using only the training dataset to minimise MSE.

g) Model testing: Generate subseries forecast using the best-fitted NNAR and stateless LSTM. Based on RMSE and MAE, compare NNAR and denormalised stateless LSTM predictions with the test dataset (20%) of the original subseries.
h) Forecast reconciliation: Train and validate GBM using the training dataset (80% of the original wind speed data). Hypeparamaters are iteratively optimised through cross-valiadation to minimise RMSE on the training data. Use the trained GBM model to calculate the final predicted value by combining the predictions of all decomposed subseries data.
i) Final model evaluation: The reconciled forecasts are evaluated against the test dataset (20% of the original wind speed data) based on MAE, RMSE, median absolute deviation (MAD), continuous ranked probability score (CRPS), and prediction interval width (PIW).

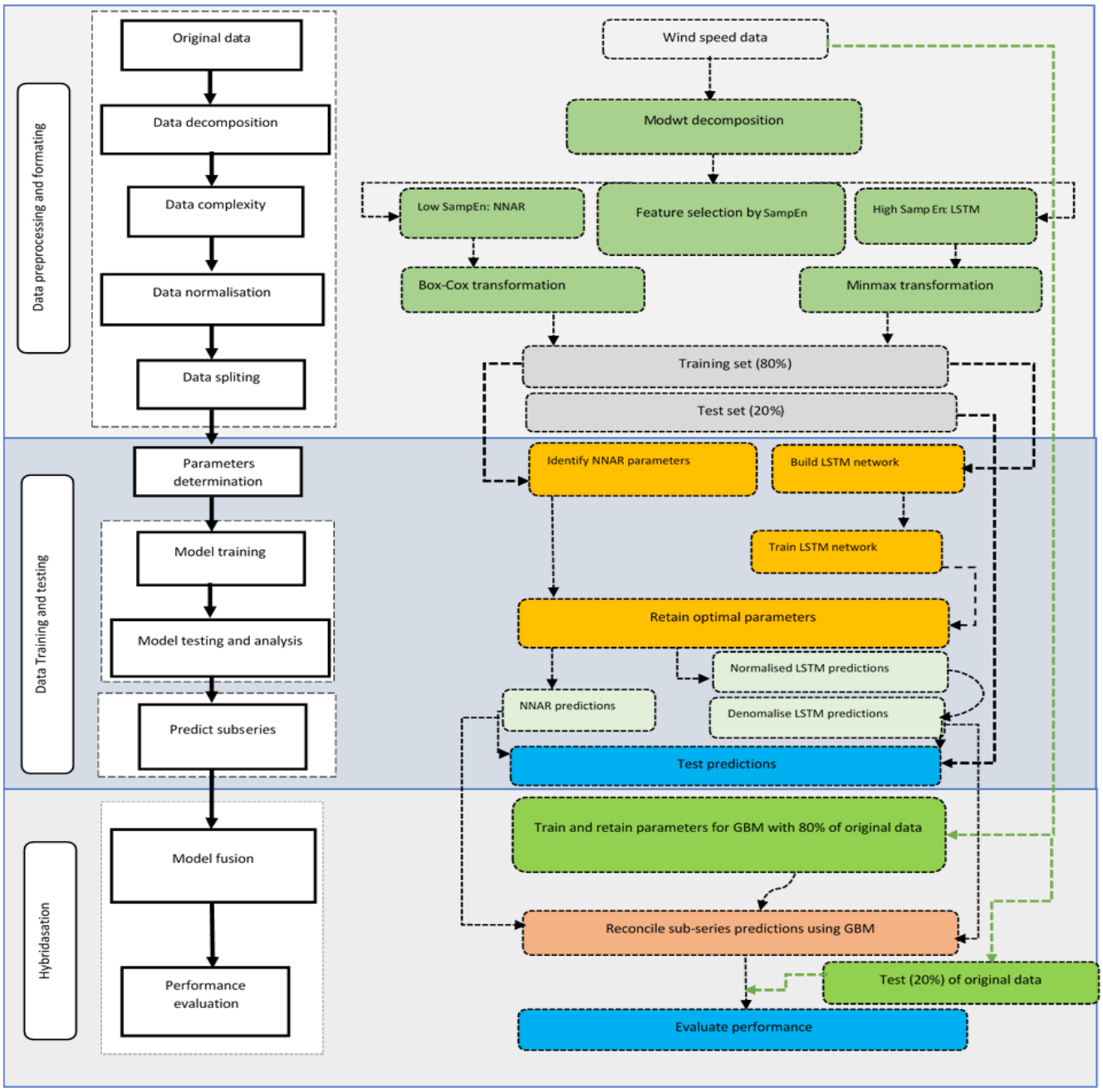


Figure 3.3. Proposed WT-NNAR-LSTM-GBM model

Rationale

In the evaluated studies (see Chapter 2), no research work examined the complexity of wavelet subsignals using information theory methods to identify the most appropriate modelling technique. Besides, wind speed prediction models are rarely optimised and enhanced through the blending of SampEn, WT. The forecast combination methods used are often linear (direct summation) and cannot account for the nonlinear features of wind speed predictions. This study proposes a new hybrid framework that leverages the benefits of WT, SampEn, NNAR, LSTM, and GBM

methods to construct the WT-NNAR-LSTM-GBM. In this framework, the WT is employed to deconstruct wind speed data into multiple subseries (approximate and detailed subseries with different and improved statistical properties), whereas the SampEn characterise the complexity feature in each of the subsignals. The NNAR and LSTM are employed to separately forecast the deterministic (low-variant) and complex (high-variant) subseries, respectively. The GBM, being nonlinear, scalable, and accurate, is employed to effectively and efficiently combine subseries predictions to arrive at final forecasts.

Model Contribution

The contribution of each model towards the proposed WT-NNAR-LSTM-GBM framework is presented in Table 3.4 below.

Table 3.4. Model contribution to the proposed WT-NNAR-LSTM-GBM model.

Model	Contribution to the Strategy
WTs	✓ WTs are of paramount importance as a denoising and transform technique, as they are designed to minimise random fluctuations in the data sequence and enhance prediction accuracy. Consequently, these techniques are endorsed for the deconstruction of wind speed data into low-frequency and several high-frequency signals.
SampEn	✓ Besides being highly efficient and simple, SampEn can determine the randomness of a wind speed series of data without any previous knowledge of the source data. Hence, the study unleashes the power of SampEn and employ it to determine the level of complexity for each of the decomposed signals, thereby ensuring that the most appropriate modelling and forecasting approach is employed for improved prediction accuracy.
NNAR	✓ NNARs are not only employed to detect trends, but they are not prone to non-stationarity and outlier effects. Consequently, these nonlinear approximators are employed to precisely detect and model nonlinear features within those subseries that have been recognised as less random (or deterministic) through the application of the SampEn criterion.
LSTM	✓ To circumvent the drawbacks of gradient disappearance and explosion to which the NNARs are vulnerable, those subseries that are considered to be more complex (or highly random) based on the SampEn criterion (i.e., SampEn values closer or greater than 1 (i.e., at least 0.9)), are modelled using a more robust and reliable stateless LSTM. This time series prediction task is best performed using stateless LSTMs over stateful LSTMs due to their stability and accuracy.
GBM	✓ In addition to its robustness and scalability, a nonlinear GBM model is preferred over a linear combination (such as conventional direct summation) model for prediction fusion because it is highly accurate. In addition, it takes into account the nonlinear structure of subseries forecast in the combination of predictions, thus enhancing predictive performance.

3.5.3 wavelet-MODWT-GRU (MB)

The originality of the proposed wavelet-MODWT-GRU (MB) framework are founded on the basis that wind data exhibit cyclostationarity owing to cyclic variations, non-uniform variance caused by variations in meteorological conditions, and structural discontinuities that originates transient wind behaviour. Consequently, a single class of models cannot sufficiently capture this behaviour. Moreover, literature (see Chapter 2) rarely investigated MODWT, nor did they thoroughly outline the mathematical approach employed to determine the wavelet decomposition level. Also see details in the work of [29].

Summary of the Design

The main steps involved in the built of the proposed wavelet-MODWT-GRU (MB) approach are as follows (see Appendix A for details):

a) Data cleaning, processing, and partition: Clean, format, handle inconsistences, and outliers. Only retain wind speed less than 22 m/s for practical wind power applications. In instances where outliers and missing data are found, mean imputation will be used. Also, chronologically partition the original data into a training set (80%) (for model training) and a test set (20%) (preserved for model evaluation).
b) DE hyperparameter search: Using the training dataset (80%), the MB filter was initialised, the objective function was defined, and the DE parameters were set. Thereafter, the DE was run to identify the optimal decomposition level through minimising MSE. The DE hyperparameters search space is as follows: Number of iterations: 50-60; population size (45-60); crossover probability (0.75-0.85); weights (0.5-0.6); and parameter bounds (1-10) (also see Table 6.4 for details).
c) Signal deconstruction and partition: Deconstruct the original wind speed data using MODWT based on MB (at $L = 3$) to yield varying subseries with different frequency bands. Partition, in a chronologically manner, each subseries into a training set (80%) (for model training) and a test set (20%) (preserved for model evaluation). Thereafter, use the min-max scalar to normalise these data.
d) GRU hyperparameter search: Format subseries data for compatibility with the GRU network, initialise parameters, train the model, and preserve optimal parameters. Using the training dataset (80%), early stopping procedure together with dropout regularisation were employed to reduce model

overfitting and select the best model. The GRU search space is given by: dropout rate (0-0.5), time steps (1-10), epochs (1-100), validation split (0.1) (of the training dataset), learning rate (0-0.1), activation function (tanh), loss function (MSE), and optimiser (Adam) (also see Table 6.4 for details). GRU was iteratively trained on the training dataset with the main objective of minimising the MSE indicator.

e) Test GRU performance: Generate normalised subseries forecasts using GRU, denormalise the subseries forecasts, and evaluate against the preseved test set (20%) of the original subseries data based on RMSE, MAE, and MAPE

f) Signal reconstruction and prediction evaluation: Reconstruct predictions from all subseries using MRA (inverse MODWT) approach and compare the results with the test set (20%) of the original wind speed data.

g) Final model evalaution: Evaluate and preserve performance metrics and tests (i.e. RMSE, MAE, MAPE, R^2, pinball loss (PL), murphy diagram (MD), Mincer-Zarnowitz (MZ), Dawid–Sebastiani (DS), and probability integral transform (PIT)).

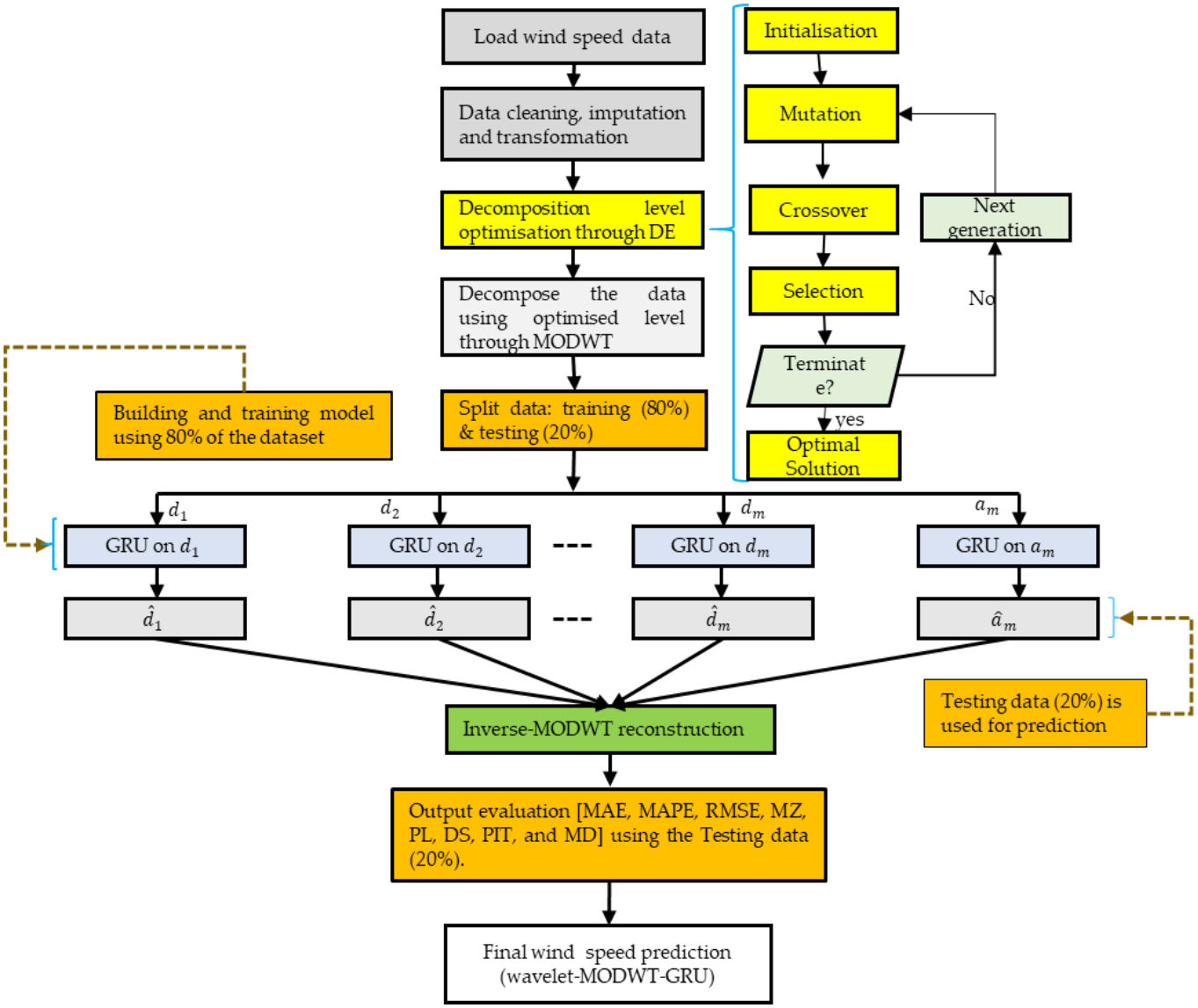


Figure 3.4. Flow chart of the proposed wavelet-MODWT-GRU

Rationale

The proposed framework exploits the advantages of MODWT, which successfully filters noise and reveals (to some extent) transient wind characteristics, combined with DE, known for its robust global search efficiency, rapid convergence speed, and ease of implementation. Therefore, the proposed framework employs DE to optimise the decomposition level of the MB filter. This allows for effective and efficient decomposition of wind speed signals into frequency bands with enhanced statistical features, aiding the detection of cyclical variations in the mean and variance through MODWT. Each subsignal derived through MODWT decomposition is distinctly modelled and predicted with the GRU to effectively handle the complex characteristics of wind speed characteristic such as vanishing gradients and model overfitting. The final prediction is calculated by reconstructing the original signal using the inverse MODWT, which employs the MRA approach that integrates each subsignal prediction as input.

Model Contribution

The proposed framework leverages from the strength of DE, MB, MODWT, GRU, and MRA to form wavelet-MODWT-GRU (MB). In Table 3.5, the summary of the role for each of the model involved in the development of wavelet-MODWT-GRU (MB) model is presented.

Table 3.5. Model contribution to the proposed wavelet-MODWT-GRU (MB).

Model	Contribution to the Strategy
DE	✓ The study employed the DE algorithm for the optimisation the wavelet decomposition level since it is simple, efficient, and capable of handling complex continuous non-differentiable problems.
MODWT	✓ A time-invariant MODWT resistant to boundary effects is used, utilising wavelet filter MB8, to deconstruct the wind data into detailed and approximate frequency components with reduced noise and are easy to model and predict.
GRU	✓ GRU is renowned for its ability to handle vanishing gradients, a characteristic particularly suited to handling detailed signals. Hence, these techniques are employed to individually model and predict each subsignal with such a level of accuracy and reliability while addressing the issue of vanishing gradients.
MRA	✓ Finally, MRA, recognised for its effective data extraction, denoising, and efficient reconstruction, was employed to reconstruct the original wind data and produce the final forecast from all subsignal forecasts.

3.5.4 RVM-WT-AdaBoostRT-RF

In terms of the framework contributions, the proposed RVM-WT-AdaBoostRT-RF improves upon the inadequacies of the previous work in the sense that it provides computational efficiency, effectively captures nonlinearity, avoids overfitting, and has high accuracy in the prediction of unplanned power outages using power grid parameters. Also see details in the work of [15].

Summary of the Design

The main step steps involved in the development of the proposed approach are as follows (see Appendix A for details):

1. *Data cleaning, formatting, dimension reduction, and partition*
   a) Data cleaning and formatting: Check completeness, correctness, consistency, handle structural errors, drop irrelevant features, and create new features
   b) Multicollinearity and data partition: Detect the presence multicollinearity through variable inflation factor (VIF);
   c) Feature selection (LASSO): Partition data into 80% training and 20% testing sets. Train LASSO and then conduct variable selection.
   d) Data partitioning: Divide the complete dataset (of only the variables selected via LASSO) into four season-based datasets. Extract data from the first two months of each season to represent that season (e.g., Autumn: March to April).
   e) Future selection (RF): In each of the season-based datasets, partition the data into training (80%) and test (20%) sets. Thereafter, train and select the top 10 most influential variables per season using RF.
2. *Base Model forecast*
   f) Base model forecasts: Partition season-based data into training (80%) (train =60% +val=20%) and test (20%) datasets. Train, validate, and predict data using RVM, RF, and AdaBoostRT. Thereafter, assess performance via MAE, MAPE, and RMSE for each model.
3. *Decomposition of RVM residuals*
   g) Residual generation: Fit the RVM model on each of the seasoned-based datasets, generate RVM residuals, and decompose the residuals via MODWT

h) Residual forecasts: Partition subseries into training (80%) (train=60% +val=20%) and test (20%) datasets. Train, validate, and predict residuals using AdaBoostRT. Thereafter, assess performance via MAE, MAPE, and RMSE.

4. *Stacking through RF*

i) Forecast fusion (stacking): Utilise base models' validation predictions (i.e., RVM, AdaBoostRT, RF, and residual validation forecasts) as input into RF to train, stack, and produce final model forecasts.

j) Model evaluation: Model performance evaluation through MAE, MAPE, RMSE, PINAW, MZ test, and DM test

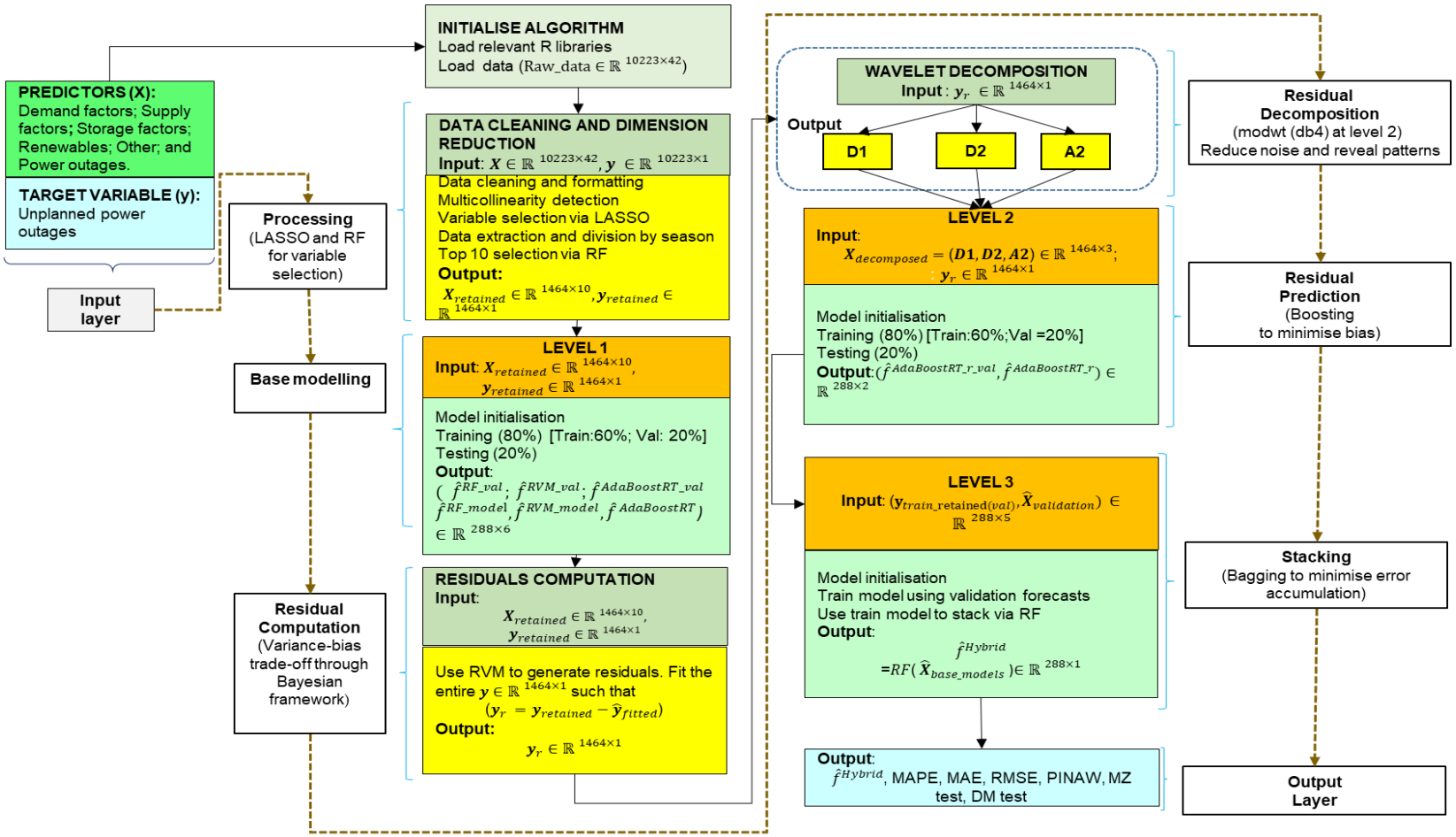


Figure 3.5. Schematic representation of the proposed stacking hybrid RVM-WT-AdaBoostRT-RF model

Rationale

In stacking, multiple predictive models are combined via a meta-model. Fundamentally, each successive leaner in the stack corrects the errors of the prior model, thus minimising error accumulation, overfitting, and ultimately improving predictive performance of the model. In the proposed approach, LASSO and RF detect and manage general aspects, including dimensionality reduction; RVM enhances nonlinear probabilistic learning; WTs along with AdaBoostRT dwell on more specific aspects of the residuals to minimise bias and noise. RF algorithm further ensures that combined forecasts are more stable by effectively modelling nonlinearity as well as averaging errors with speed and accuracy within the stacked framework. In essence, the stacking technique minimises the errors of the base model, improves forecast accuracy, and increases the seasonal robustness of the model.

Model Contribution

The contribution of each model employed in the proposed framework is presented in Table 3.6 below.

Table 3.6. Model contribution to the proposed RVM-WT-AdaBoostRT-RF.

Model	Contribution to the Strategy
LASSO	✓ LASSO is employed for regularisation, variable selection, and dimension reduction. Consequently, unplanned power outages are accurately forecasted by employing the most relevant and significant features.
RVM	✓ These sparse Bayesian-based methods are probabilistic frameworks which require smaller number of support vectors whilst delivering accuracy and generalisation paralleled to that of SVMs. In fact, RVMs can effectively characterise complex data patterns (such as random fluctuations, nonlinearity, intermittence, etc.) while avoiding overfitting. Therefore, RVMs are a top choice for regression on heterogeneous power grid data.
WT	✓ The application of the frequency and time-domain compatible WTs, effectively minimise the effect of noise and expose complex data characteristics that exist in power outage data. Hence, RVM residuals are best decomposed with a WT, as it is efficient and can handle nonstationary fluctuations well. Therefore, these signals become statistically reliable and easy to predict, thereby enhancing the predictive power of the model.
AdaBoostRT	✓ Our solution for high volatile residuals leverages AdaBoostRT capabilities to minimise model bias, and accurately forecast residual subseries while using decomposed subseries as input. As a result, bias in the forecast is minimised.
RF	✓ Besides providing top-10 most importance variables (which are pivotal for robust season-specific modelling), RFs are highly efficient at capturing nonlinearity while preventing overfitting and minimising variance. We, therefore, utilise RF as a meta-model to accurately

and efficiently ensemble RVM, RF, AdaBoostRT, and residual forecasts to arrive at the forecast value, while minimising error accumulation and enhancing overall model robustness.

3.6 Performance Evaluation Metrics

The current section focuses on performance evaluation metrics and statistical tests employed in Chapters 4-7 to compare and select the best models.

3.6.1 Point Prediction Evaluation Metrics

The point or deterministic predictive accuracy of the models employed in this study were evaluated using MAE, RMSE, MAPE, and R^2. Models that attained the least value of MAE, RMSE, MAPE and highest value of R^2 were considered to be the best (also see e.g., [26,75-77,83]). Furthermore, Chapter 2 (Table 2.3) presents the detailed characteristics of each of the aforementioned indicators. The respective mathematical expressions of the aforementioned performance metrics are given by:

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n} |\xi_t|, \tag{3.52}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n} {\xi_t}^2}, \tag{3.53}$$

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n} \frac{|\xi_t|}{y_t} \times 100, \tag{3.54}$$

$$R^2 = 1 - \frac{\sum_{t=1}^{n} {\xi_t}^2}{\sum_{t=1}^{n} (y_t - \bar{y}_t)^2}, \tag{3.55}$$

where $\xi_t = y_t - \hat{y}_t$, y_t and $\bar{y}_t$ respectively represent the actual and mean wind speed value, and n is the sample size. This is a pivotal quantity employed in the assessment of residual predictions.

3.6.2 Probabilistic Evaluation Metrics

Noting that wind resources (e.g., wind speed) are physical quantities that are highly reliant on climatic conditions, measuring uncertainty in wind power is critical for power grid stability. Probabilistic predictions provide a range of scenarios, thereby improving operational decisions in the power system management (also see [26]).

Probabilistic metrics evaluate the model's spread by considering the deviation of the conditional mean distribution from actual observations. The study employed the following probabilistic indices to assess models' sharpness, reliability, generalisation, and calibration of predictions, viz., PL, MAD, DS, PIT histograms, CRPS, and PI indices (see e.g., [26, 220-223] for details).

Pinball Loss Function

Apart from its ability to deal with non-symmetric errors, the PL can be used to assess the *reliability* of interval predictions. The PL is given by [83]:

$$PL_{\tau}(y_t, \hat{y}_{\tau t}) = \begin{cases} (y_t - \hat{y}_{\tau t})\tau, y_t \geq \hat{y}_{\tau t}, \\ (\hat{y}_{\tau t} - y_t)(1-\tau), y_t < \hat{y}_{\tau t}, \end{cases} \tag{3.56}$$

where $\tau \in (0,1)$ is the target quantile, y_t the actual wind speed value, $\hat{y}_{\tau t}$ the quantile forecast. The lower the PL value, the more accurate the quantile forecast.

Median Absolute Deviation

The aspect of *sharpness* involves the model's ability to generate predictions within a narrow probability distribution [220]. To measure the *sharpness* of the model, the study employs the MAD given by (see [220]):

$$\text{MAD} = 1.4826 \times \text{median}(|y_t - \tilde{y_t}|), \tag{3.57}$$

where y_t and $\tilde{y_t} = \text{median}(y_t)$, respectively, denote the actual and the median values at time t. The MAD is advantageous in that it is robust as it is less sensitive to outliers. However, if the distribution is normal, the method loses efficiency by not utilising all the information available in the data. The lesser the value of MAD, the better the model.

Dawid-Sebastiani Score

Model *reliability or calibration* refers to its ability to detect *uncertainty* when predicting. In this scoring rule broad distributions are penalised, whilst narrower distributions are incentivised, emphasising distribution *sharpness*. The DS is calculated by the following equation [220,221]:

$$DS = 2log(\sigma) + \left(\frac{y_t - \mu}{\sigma}\right)^2, \tag{3.58}$$

where y_t is the actual wind speed observation, μ denotes the mean and σ is the standard deviation of the forecast distribution. Lower values of the DS score imply

more accurate and reliable probabilistic forecasts. Fundamentally, a better-calibrated and sharp DS score for wind speed predictions is particularly useful to system operators as it provides an improved representation of sudden and extreme wind speed events which supports effective planning and risk management during wind power uncertainty. A similar interpretation can be applied to smaller values of PL, MAD, and PIs.

Probability Integral Transform

In an ideal forecast distribution, there should be no bins with extremely high or low levels on the PIT histogram. The PIT is computed using the equation below [220]:

$$PIT = \beta = F_{\delta}(y_t), \tag{3.59}$$

where β is the transformed variable and $F_{\delta}(y_t)$ is the forecast distribution F_{δ} evaluated at the actual wind speed value y_t. Smaller and uniformly distributed PIT values imply *well-calibrated* probabilistic forecasts; otherwise, the models' probabilistic forecasts are miscalibrated or skewed.

Continuous Ranked Probability Score

Similar to the DS core, CRPS is also used to measure *calibration* alongside detecting uncertainty (i.e. models' reliability) when forecasting [220-223]. The CRPS between x and F is defined as (see [220,222])

$$\text{CRPS}\ (F, x) = \int_{-\infty}^{+\infty} (F(y) - H\{x \geq y\})^2 dy, \tag{3.60}$$

where F is the cumulative distribution function (CDF) of the random variable X, and H is the Heaviside step function ($H\{x \geq y\}$ equals 1 if $x \geq y$ and 0 if $x < y$). A CPRS closer to 0 indicates *reliable, better calibrated* and accurate predictions, while a CRPS away from 1 indicates inaccurate predictions.

Prediction Intervals

PI shows a broad range of possible probabilistic values within which the actual values of wind speed should lie with a certain specified probability. The PI is essential when assessing uncertainty in point forecasts as it handles uncertainties [26]. The PI indicators employed in the current study are as follows:

$$\text{PIW}_t = \text{UL}_t - \text{LL}_t, \tag{3.61}$$

$$\text{PINAD} = \frac{1}{nR}\sum_{i=1}^{n} Z_t, \tag{3.62}$$

where

$$\text{Z}_t = \begin{cases} \text{LL}_t - \text{y}_t, \text{y}_t < \text{LL}_t, \\ 0, \text{y}_t \in (\text{LL}_t, \text{UL}_t), \\ \text{y}_t - \text{UL}_t, \text{y}_t \geq \text{UL}_t, \end{cases} \tag{3.63}$$

$$\text{PINAW} = \frac{1}{nR_{y_t}}\sum_{t=1}^{n}(\text{UL}_t - \text{LL}_t), \tag{3.64}$$

where UL_t and LL_t are the upper and lower limits of the PI, respectively, and R_{y_t} represents the range (y_t) indicator. The smaller PI indicator value is preferred since it resembles a forecast that is narrower, more *reliable*, and better *calibrated*. Another basic but essential PI index is the prediction interval coverage probability (PICP) denoted by the expression below:

$$\text{PICP} = \frac{1}{n}\sum_{t=1}^{n} \text{I}_t \ , \tag{3.65}$$

where I_t is the binary function such that

$$\text{I}_t = \begin{cases} 1, \ y_t \ \in (\text{LL}_t, \text{UL}_t), \\ 0, \ \text{if otherwise.} \end{cases} \tag{3.66}$$

For instance, a 95% PI for wind speed shows that about 95% of the observations are anticipated to fall within the interval. This value assists system operators to minimise risk by planning for different scenarios and it facilitates the timely distribution of resources.

3.6.3 Predictive Accuracy Assessment

Murphy Diagram

The MD was proposed by [224], and it graphically plots to compare the predictive strength of the models. Thus, MD displays forecast skill over various scoring functions, thereby enhancing comprehensive comparison and assessment of probabilistic model accuracy. Considering the loss function $S(\hat{y}_t, y_t) = (\hat{y}_t - y_t)^2$, the work of [224] showed that any consistent scoring function can be written in the following form:

$$S(\hat{y}_t, y_t) = \int s_\theta\,(\hat{y}_t, y_t)\mathcal{H}d(\theta), \tag{3.67}$$

where $\mathcal{H}$ denotes a non-negative measure and

$$S_\theta(\hat{y}_t, y_t) = \begin{cases} |y_t - \theta| & if\ min\ (\hat{y}_t, y_t) \le \theta < max(\hat{y}_t, y_t), \\ 0 & \text{otherwise,} \end{cases} \tag{3.68}$$

where $\theta \in \mathbb{R}$. For n events point forecasts, the average, denoted by $S_\theta(\hat{y}_t, y_t)$ can be calculated using the following mathematical expression:

$$s(\hat{y}_t, y_t) = \frac{1}{n}\sum_{i=1}^{n} S_\theta\left(\hat{y}_{t_i}, y_{t_i}\right). \tag{3.69}$$

Diebold-Mariano Test

The DM test evaluates the forecasting strength of the models. Consider the two forecasted values donated $\hat{y}_{ti}, \hat{y}_{tj}$, for the value y_t from models i and j, respectively. Let $L(\zeta_{tr}) = y_{tr} - \hat{y}_{tr}$ for $r = 1,2$ be the loss function associated with the two forecasts with d_r=$L(error_i)$- $L\left(error_j\right)$. The DM test the following hypothesis H_0: $E(d_r) = 0$, vs. H_a: $E(d_r) \neq 0\ \forall\ r$, with the statistic given by (see e.g., [225]):

$$DM = \frac{\frac{1}{n}\sum_{r=1}^{n} d_r}{\sqrt{\frac{S^2}{n}}}, \tag{3.70}$$

where S^2 is the estimated variance of d_r. The calculated DM value is then compared to the critical value. In both cases, the original hypothesis will be rejected if the DM statistic is greater than the upper critical value ($Z_{\frac{\alpha}{2}}$) or less than the lower critical value ($-Z_{\frac{\alpha}{2}}$).

3.6.4 Model Biasedness

Consider the residual terms denoted $\xi_t = y_t - \hat{y}_t$, $t = 1, \dots, n$, if $\xi_t \neq 0$, then the model is either overestimating ($\xi_t > 0$) or underestimating ($\xi_t < 0$) the actual wind speed observation. Fundamentally, if $\mathrm{E}\,(\xi_t) = \frac{1}{n}\sum \xi_t = 0$ then the forecast results are unbiased. The MZ test regression function is given by the following equation (see e.g., [226] for details):

$$y_t = \omega + \phi\hat{y}_{t-1} + \xi_t. \tag{3.71}$$

The MZ test evaluates the unbiasedness and consistency of the predictions, by testing the null hypothesis that the intercept (ω) and slope (ϕ) terms are respectively 0 and 1. If parameter $\omega = 0$ and $\phi = 1$, it indicates that the model's predictions are unbiased and the prediction errors are minimal, otherwise, the model is considered biased. In essence, the rejection rule is such that if the p-value > 0.05 then the model is unbiased; otherwise the model is biased.

3.7 Conclusions

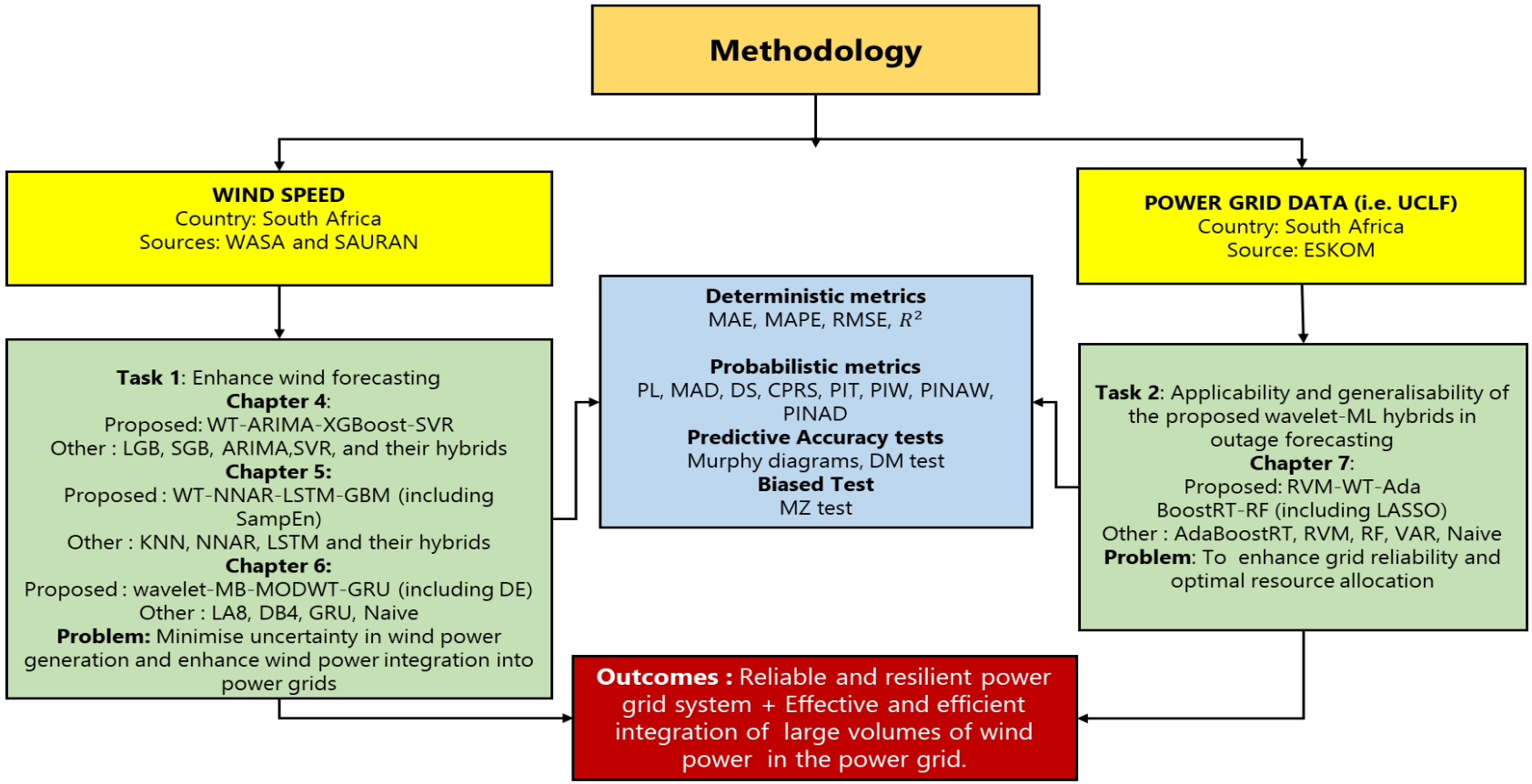


Figure 3.6. Methodology flowchart

This page is intentionally blank

Chapter 4

Multi-Horizon Wind Speed Forecasting Using Stochastic Methods, Wavelets and Gradient Boosting Decision Trees: A Hybrid Approach

4.1 Introduction

The earnestness for decarbonisation, together with a rise in electricity prices and the abundant availability of wind resources in South Africa, renders investments in wind technologies a clear decarbonisation approach [9]. In addition, South Africa has yet to exploit its abundance of wind power potential [9]. The study aims to assess these wind energy resources in order to inform key decision-makers of their value and unexploited potential. Thus, this chapter delves into wind speed predictions as a way of providing accurate forecasts to policymakers, thereby enhancing the effective incorporation of significant volumes of wind power into existing electric grids. To overcome significant challenges to operations and planning practices in the incorporation of the electric system due to wind energy's inherent discontinuity and limited predictability, wavelet-ML hybrids are available in the literature, but to a lesser degree. Hence, the rationale for the proposed WT-ARIMA-XGBoost-SVR ensemble method is premised on the basis that wind speed is characterised by inherent linearity, nonlinearity, and nonstationarity phenomena that cannot be simultaneously captured by one single class of model. In fact, previous applications of single classes of models focused exclusively on predictions while neglecting to unmask the aforementioned aspects of wind speed time series. The efficacy of the proposed WT-ARIMA-XGBoost-SVR approach is benchmarked against ARIMA and two other hybrid models, namely, the one that substituted XGBoost with an LGB component to create a WT-ARIMA-LGB-SVR and the other, which replaced it with an SGB to create a WT-ARIMA-SGB-SVR. The feasibility and efficacy of the proposed hybrid framework model were confirmed empirically through prediction metrics.

4.2 Empirical Results

4.2.1 Data Description

The current chapter employs minute-averaged wind speed data measured using an

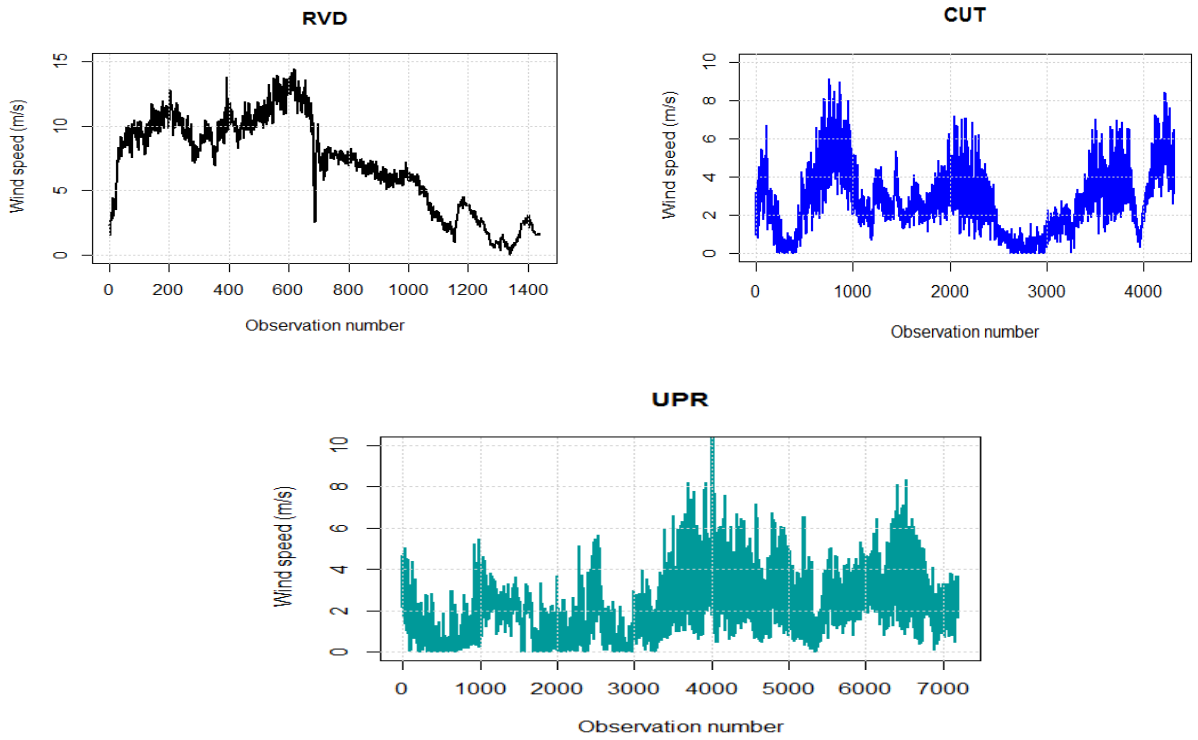


Figure 4.1. Minute wind speed data for RVD (top left panel), CUT (top right panel), and UPR (bottom centre panel).

R.M. Young (05103 or 03001) anemometer. These data were obtained from the different SAURAN radiometric stations, viz., Richtersveld (RVD), Central University of Technology (CUT), and University of Pretoria (UPR) in South Africa (see Chapter 1 for the link). The CUT station is located on the roof of a building at the CUT University, in the Free State province, at latitude −29.121337, longitude 26.215909, and an elevation of 1397 m. The RVD station is located in the desert region of the Northern Cape at latitude −28.56084061 and longitude 16.76145935, with an elevation of 141 m. The UPR station is located on the roof of a building at the University of Pretoria, in the Gauteng province, at latitude −25.75308037, longitude 28.22859001, and an elevation of 1410 m (also see Figure 4.1). We deliberately choose these locations to test the robustness of the proposed hybrid approach under different terrains with varying weather conditions, thereby enhancing the generalizability of the proposed approach.

Table 4.1. Details of sampled data division.

Station	Number of Days	Month	Sample Size	Training Set	Testing Set
RVD	1	7 September 2019	1440	1152	288
CUT	3	15–19 August 2019	4320	3456	864
UPR	5	1–5 June 2021	7200	5760	1440

These data were partitioned into a training set (80%) and a testing set (20%). To measure the predictive performance of the proposed hybrid framework, sites characterised by different meteorological conditions were selected over different days, months, and years (see Table 4.1 and Figure 4.1).

4.2.2 Summary Statistics

Table 4.2 below presents the descriptive statistics for wind speed observations at the three radiometric stations under study. The UPR exhibits the smallest variation, while RVD demonstrates the highest variability. Additionally, the RVD dataset is negatively skewed whilst the other datasets are positively skewed.

Table 4.2. Descriptive statistics for wind speed data (m/s).

Station	RVD	CUT	UPR
Min	0.036	0	0
Mean	7.125	2.770	2.229
St. Dev.	3.603	1.651	1.488
Max	14.400	9.130	10.790
Skewness	−0.218	0.479	0.646

4.2.3 Model Settings

In this Chapter, the methods employed for analysis were tuned and evaluated on an HP notebook (in R package 4.2.2) equipped with an Intel Core i5 processor. The "forecast" package was used to implement the ARIMA model, whilst the "waveslim" package through "modwt" deconstructed the wind speed data. To tune the SVM algorithm, the "svm" function found in the "e1071" package is employed. XGBoost and SGB were built using the packages "xgboost" and "gbm", respectively, whilst LGB was trained through "lightgbm". A sequential grid searching technique was employed to determine the best hyperparameters for the regression models. The resulting best parameter intervals are shown in Table 4.3. We modified the model

hyperparameters to adjust the forecast horizon, thereby increasing (to some extent) model forecasting accuracy.

Table 4.3. Model hyperparameter optimisation interval.

Model	Hyperparameter	Optimisation Interval
ARIMA	Autoregressive term	0–3
	Moving average term	0–3
	Integrated term	0–1
WT	wf	'la8'
	n.levels	3–4
	boundary	'periodic'
SVR	RBF kernel: Cost	1–50
	RBF kernel: Gamma	0.5–10
XGBoost	Max Tree depth	3–15
	Learning rate	0.05–0.95
	Min child	1
LGB	Max Tree depth	3–15
	Learning rate	0.05–1
SGB	Interaction depth	3–7
	Learning rate	0.005–0.3
	Number of trees	6–59

The computational time across all models is shown in Table 4.4. The WT-ARIMA-LGB-SVR proved to be the most efficient, followed by the WT-ARIMA-XGBoost-SVR and ARIMA models. The WT-ARIMA-SGB-SVR yielded the lowest (yet acceptable) execution time across all models.

Table 4.4. Implementation time (in seconds) for the fitted models on the wind speed data.

Model	Training and Testing Dataset (s)
ARIMA	~7–15
WT-ARIMA-XGBoost-SVR	~4–11
WT-ARIMA-LGB-SVR	~3–9
WT-ARIMA-SGB-SVR	~7–30

The three residual wind speed datasets were deconstructed into several high-frequency signals and one low-frequency signal using a level 3 (for RVD) and level 4 MODWT (for CUT and UPR), as presented in Figure 4.2. The three datasets display a rise in variability as decomposition levels declines.

4.2.4 Discussion of the Results

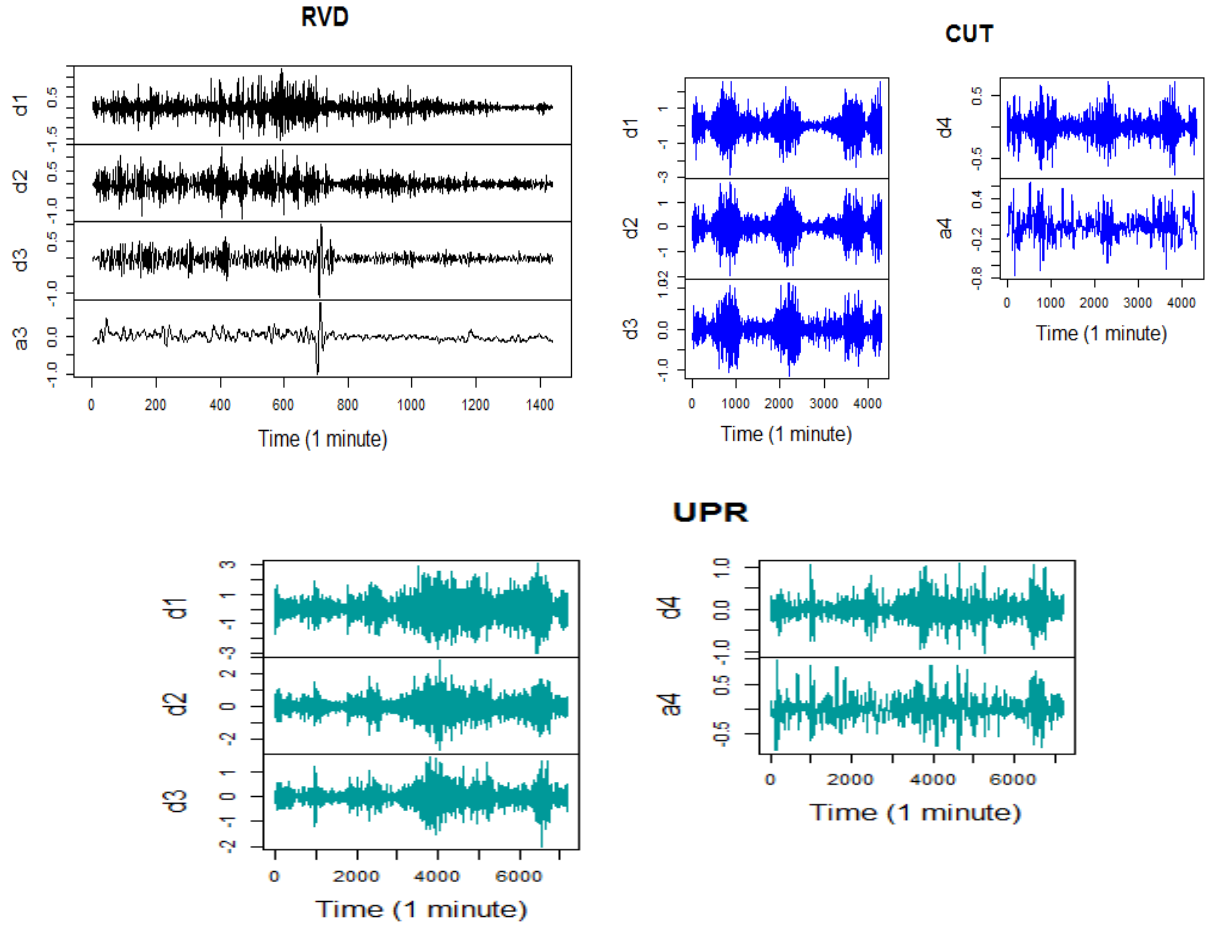


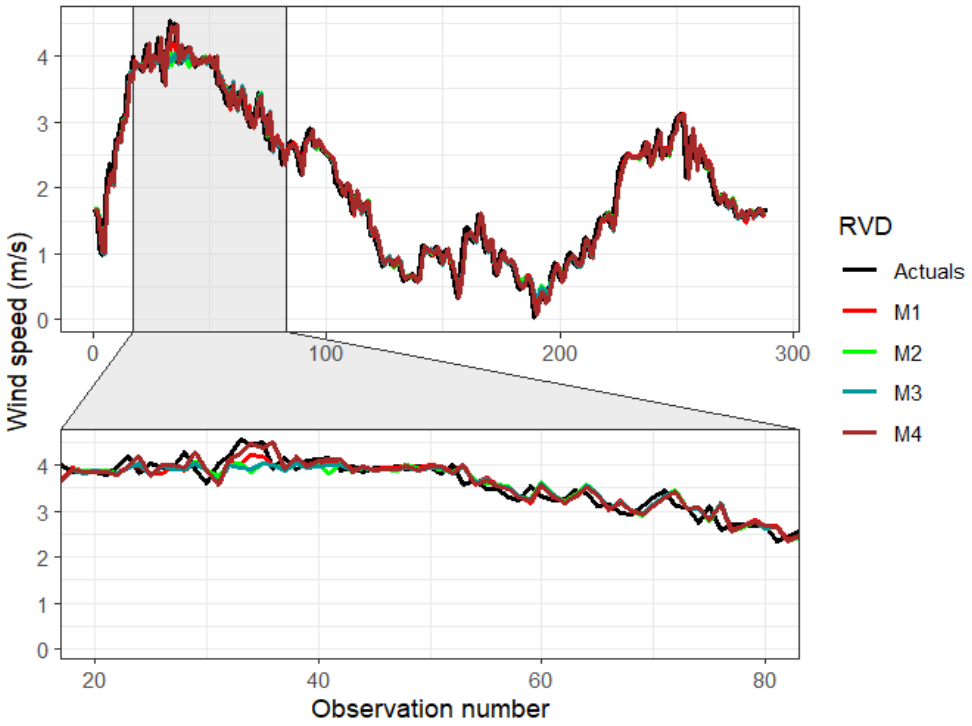
Figure 4.2. MODWT results for minutely averaged wind speed data for RVD (top left panel), CUT (top right panel), and UPR (bottom centre panel).

Table 4.5 presents the point predictions for the four approaches applied to the three datasets. M1, M2, M3, and M4 correspond to WT-ARIMA-XGBoost-SVR, WT-ARIMA-LGB-SVR, WT-ARIMA-SGB-SVR, and ARIMA, respectively. The best performing model values are highlighted in Table 4.5. Across all three datasets, M1 consistently achieved the best results for RMSE, MAPE, MAE, and R^2. For the CUT and UPR datasets, M2 performed second-best, followed by M3 and M4. For RVD data, M4 showed the second-best results in RMSE and MAE, while M2 outperformed M3 in RMSE. Model performance was stronger for smaller datasets but declined with larger datasets, especially as the forecast horizon increased. Performance varied based on dataset size, geographical context, terrain complexity, and forecasting horizon. Overall, M1 (followed by M2) demonstrated superior forecast accuracy across the three wind speed datasets (see Figure 4.3).

Table 4.5. Comparative analysis using error metrics.

Indicator	M1	M2	M3	M4
		RVD		
RMSE (m/s)	**0.174**	0.180	0.181	0.179
MAE (m/s)	**0.132**	0.135	0.135	0.134
MAPE (%)	**8.6**	8.8	8.8	9.2
R^2	**0.976**	0.974	0.974	0.974
		CUT		
RMSE (m/s)	**0.813**	0.871	0.894	0.912
MAE (m/s)	**0.549**	0.624	0.659	0.697
MAPE (%)	**14.4**	16.7	17.4	18.3
R^2	**0.693**	0.648	0.628	0.613
		UPR		
RMSE(m/s)	**0.934**	0.958	0.968	0.979
MAE(m/s)	**0.694**	0.720	0.727	0.752
MAPE(%)	**23.1**	24.0	24.4	24.7
R^2	**0.416**	0.386	0.373	0.359

Bold = Best model.



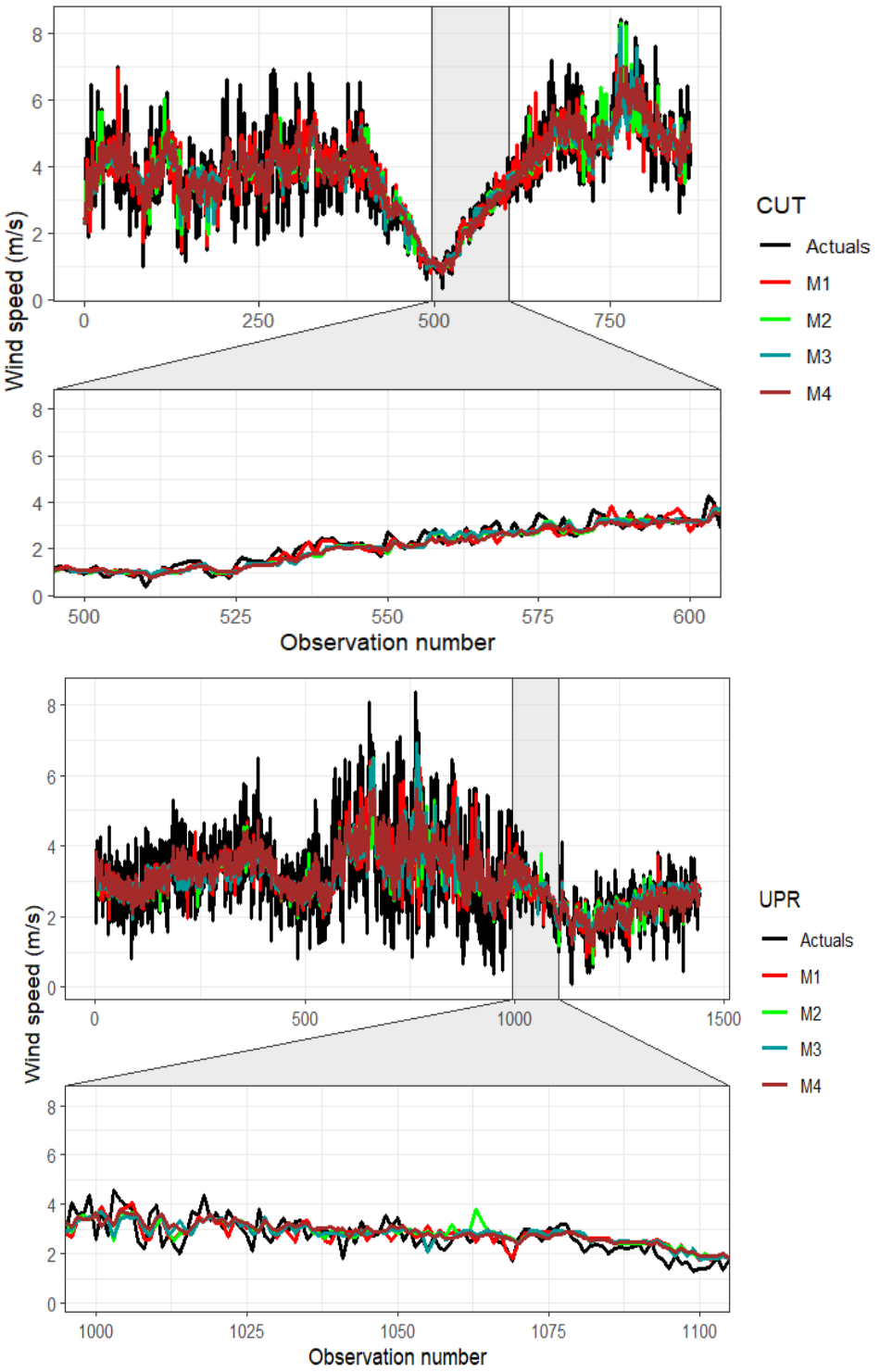


Figure 4.3. Comparison of predicted wind speeds and actual wind speed data for RVD (top panel), CUT (middle panel), and UPR (bottom panel) datasets.

The percentage improvement in forecasting accuracy between M1 and the other three models is shown in Table 4.6. Model M1 lowered RMSE by 3.2% for RVD data, 9.8% for CUT data, and 3.6% for UPR data. For the RVD and CUT datasets, M1 reduced

MAE by 2.3% and 20.3%, respectively, while MAE for the UPR data was reduced by 5.6%. Considering MAPE, M1 reduced MAPE by a mean of 3.3% for RVD, 20.9% for CUT, and 5.2% for UPR. The greatest mean gain for the R^2 metric was observed for UPR (10.3%), followed by CUT (9.1%) and RVD (0.2%) data. Overall, from evaluating percentage improvement metrics, it can be deduced that M1 improved M4 the greatest, followed by M3 for larger datasets (CUT and UPR). This suggests that M2 has the second-highest forecasting ability behind M1 for CUT and UPR. Additionally, the lowest enhanced model for RVD data is M4, followed by M2 based on RMSE, MAE, and R^2. While all models are hysteretic (see Figure 4.3), the proposed M1 produces better and steadier results when compared to the other models. Furthermore, there still exists some lag in the forecasted values when wind speed erupts, especially for the CUT and UPR.

Table 4.6. Percentage improvement rates (%).

Indicator	M1:M2	M1:M3	M1:M4	Mean
		RVD		
RMSE	3.3	3.7	2.5	3.2
MAE	2.5	2.7	1.7	2.3
MAPE	1.4	1.5	6.9	3.3
R^2	−0.2	−0.2	−0.1	−0.2
		CUT		
RMSE	7.1	10	12.2	9.8
MAE	13.7	20.1	26.9	20.3
MAPE	15.8	20.4	26.6	20.9
R^2	−6.5	−9.4	−11.5	−9.1
		UPR		
RMSE	2.5	3.6	4.8	3.6
MAE	3.7	4.8	8.4	5.6
MAPE	3.7	5.2	6.6	5.2
R^2	−7.1	−10.3	−13.6	−10.3

Table 4.7 presents the residuals of the trained models for RVD, CUT, and UPR data. The values of the best-performing models are bolded. Except for M4, errors for all models are skewed to the right for all three datasets, showing small but frequent underestimation with lesser chances of extreme gains. For RVD, residuals for model M4 are approximately normally distributed (skewness = −0.020). As expected, the ARIMA model appears to model the smaller dataset better on a shorter forecast time scale. Kurtosis analysis for all datasets and models shows positive values below 3. Therefore, the distributions are mesokurtic relative to the normal distribution. All the models have light-tailed residuals with fewer outliers. The small variability in the data implies that the data are closely spread around the mean as well (also see Figure

4.4). Overall, M1 forecasts RVD, CUT, and UPR data with greater accuracy than any other model.

Table 4.7. Comparison of models' residuals (m/s).

Statistic	M1	M2	M3	M4
		RVD		
Std.Dev	**0.175**	0.180	0.181	0.179
Skewness	0.156	0.239	0.273	−0.020
Kurtosis	1.510	1.524	1.556	1.498
		CUT		
Std.Dev	**0.812**	0.870	0.894	0.912
Skewness	0.302	0.315	0.292	0.236
Kurtosis	1.939	1.081	0.925	0.594
		UPR		
Std.Dev	**0.934**	0.957	0.968	0.980
Skewness	0.308	0.353	0.344	0.197
Kurtosis	1.015	0.732	0.907	0.574

Bold = Best model.

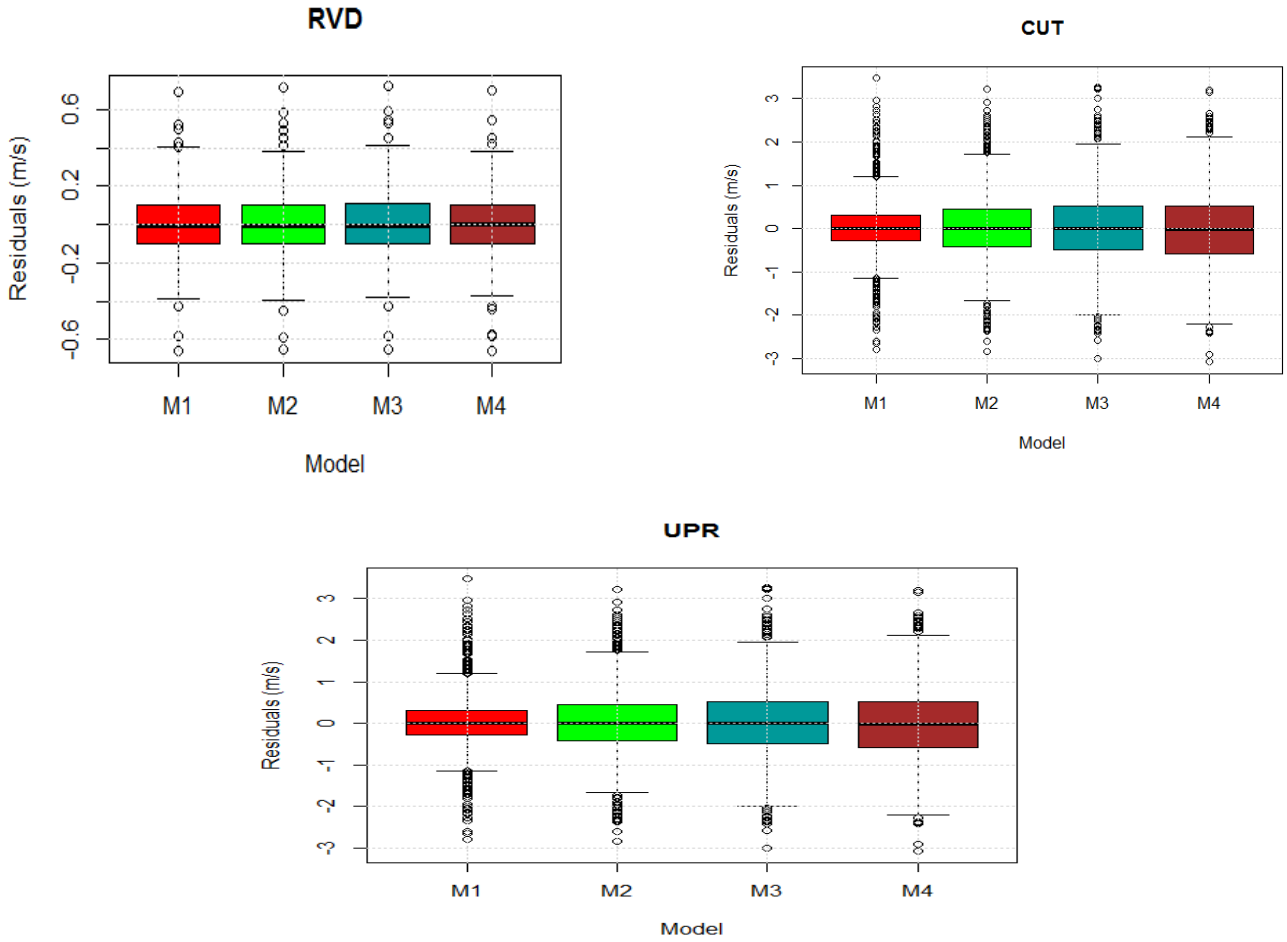


Figure 4.4. Boxplots of the residuals for RVD (top left panel), CUT (top right panel), and UPR (bottom centre panel).

Table 4.8 evaluates model efficacy using 90% PI metrics, PINAW, and PINAD. The results of the best model are reported in bold. The PINAD values were the least for the CUT and UPR datasets with M3 and M4, respectively, and the lowest for the RVD dataset using model M3. The least PINAW was produced by M1 for the CUT and UPR datasets, followed by M4 for the same datasets. PINAW was the least for RVD datasets with M4. For all three datasets, M1 produced the lowest values outside the 90% PI. Overall, M1 captures CUT and UPR data characteristics with greater reliability and with more certainty compared to other models.

Table 4.8. Comparative analysis of models using PI indices.

Indicator	M1	M2	M3	M4
		RVD		
PINAW (%)	12.464	12.496	13.208	**12.355**
PINAD (%)	0.203	0.229	**0.197**	0.215
OL (count)	**27**	29	28	30
OL (%)	**9.4**	10.1	9.7	10.4
		CUT		
PINAW (%)	**33.790**	36.535	37.373	36.468
PINAD (%)	0.616	0.550	**0.522**	0.569
OL (count)	**84**	89	86	**84**
OL (%)	**9.9**	10.5	10.2	**9.9**
		UPR		
PINAW (%)	**37.245**	38.166	38.707	38.105
PINAD (%)	0.517	0.499	0.516	**0.486**
OL (count)	143	**142**	**142**	145
OL (%)	9.9	**9.9**	**9.9**	10.1

OL = Number of predictions outside limits. Bold = Best model.

4.4 Conclusions

Considering the linear and nonlinear components as well as nonstationary behaviour inherent in wind speed data, this chapter presents WT-ARIMA-XGBoost-SVR which combines WT, ARIMA, GBDTs, and SVR to effectively and efficiently enhance short-term wind speed predictions. The predictive strength of the model was assessed using one-minute-averaged wind speed data from RVD, CUT, and UPR radiometric stations in South Africa. From the comparative study, the decomposition (through MODWT) of the highly variable and nonlinear components of the wind speed data reduced noise and variability, thereby enhancing the forecasting accuracy of all three hybrid

techniques. ARIMA was effectively applied to all three datasets to capture the linear portion of wind speed, while the GBDTs captured the irregular and erratic nonlinear portion. Both XGBoost and LGB demonstrated high efficiency and enhanced the predictive accuracy of WT-ARIMA-XGBoost-SVR and WT-ARIMALGB-SVR, respectively. Based on RMSE, MAE, MAPE, and R^2, the WT-ARIMA-XGBoost-SVR demonstrated superiority across the three datasets. Based on the overall comparative study, it can be concluded that the WT-ARIMA-XGBoost-SVR hybrid framework addresses the single models' inherent drawbacks and delivers improved accuracy, efficiency, robustness, and reliability over the three datasets under study. Despite the models' ability to accurately forecast wind speed over multi-forecast horizons on varying terrains (with different climatic conditions) in South Africa, the proposed framework shows certain limitations with sudden changes in wind speed and when forecast horizon is increased. In the future, it will be pivotal to evaluate the predictive performance of these approaches on high-variant and large wind speed datasets (outside South Africa).

4.5 Contributions

This chapter contributed to the wind forecasting literature in the following ways: Firstly, in an effort to enhance wind speed forecasting accuracy, nonlinear boosting and support vector ML methods were introduced. Secondly, computationally efficient and highly accurate GBDTs were employed instead of conventional ARIMA models, which are susceptible to nonlinearity. Thirdly, the effect of each subseries forecast error (often leading to error accumulation) on the final wind speed forecast is reduced (to some extent) by using all decomposed subseries as input features into highly accurate and nonlinear XGBoost method. Fourthly, the proposed hybrid framework accurately and efficiently captures the nonlinear components associated with wind speed turbulence and gusts. Finally, the proposed model was successfully evaluated and demonstrated satisfactory performance in accuracy, reliability, and generalisation. It was tested on multiple datasets from various locations and terrain complexities, and used for forecasting over different time spans within the multi-horizon wind forecasting framework. The results of this study are significant for enhancing the reliability of wind speed predictions and facilitating the development of effective wind power management strategies. Chapter 5 expands the hybrid framework by incorporating combining FFNs, deep learning (stateless LSTM), GDBTs (GBM) and information theory (SampEn) models to effectively uncover complex, irregular, deterministic, random, chaotic trends, and vanishing gradients associated

with wind speed data. In essence, the Chapter aims to improve stability of the hybrid frameworks in wind speed forecasting. Also see identified gaps in the literature, Chapter 2.

This page is intentionally blank

Chapter 5

Short-Term Wind Speed Forecasting Using Sample Entropy-Based Hybrid Framework to Address Vanishing Gradients

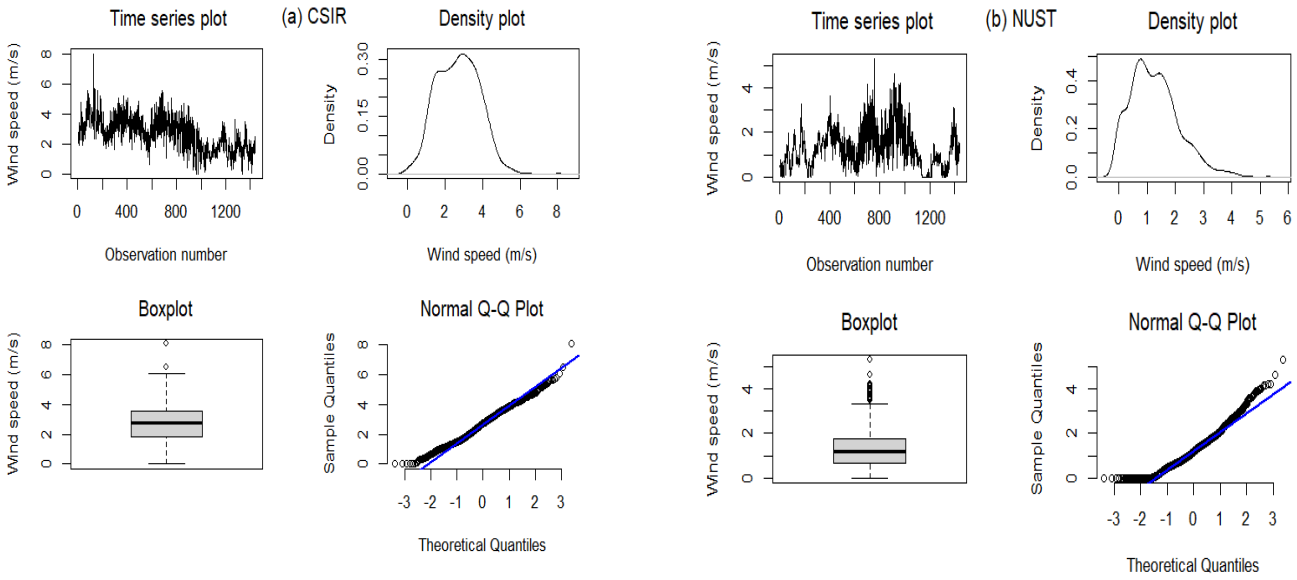
5.1. Introduction

Although wind is a renewable energy resource abundantly available in Southern Africa, exploiting its power is a multifaceted process and a specialised exercise. Reliable and accurate wind power predictions are pivotal to increasing the adoption of wind energy in electric grids. Therefore, the incorporation of wind power into electric grids pivots around the accuracy of wind power forecasts. Because wind power is strongly reliant on wind speed, accurate wind power predictions can therefore be obtained by accurately predicting wind speed. However, a single model simply cannot capture the inherent complexity and randomness of wind speed data. In the existing literature, wind speed prediction models are rarely optimised and enhanced through blending SampEn and WT. In fact, the use of stateless LSTMs to address the challenge of vanishing gradients in wind speed forecasting is not widely available in the literature (see Chapter 2 for details). A very limited, if not none, number of studies have combined the aforementioned techniques to predict wind speed reliably, accurately, and robustly over the short-term forecast horizon. The efficacy and reliability of the proposed framework, which, to our knowledge, remains unexplored in the literature on wind speed forecasting, is assessed against the NNAR (benchmark), LSTM, WT-NNAR-KNN-GBM, and WT-LSTM-KNN-GBM models using deterministic and probabilistic prediction indicators (also see Chapter 3 for details).

5.2 Empirical Results

5.2.1 Data Description

This research work employs minute-averaged wind speed data sourced from the Council for Scientific and Industrial Research (CSIR) energy centre, RVD, USAID Venda (Venda), and USAID Namibian University of Science and Technology (NUST) radiometric stations accessed via SAURAN website (see Chapter 1 for the link). The study uses three (3) South African data stations (CSIR, RVD, and Venda), and one Namibian (NUST) station. An R.M. Young anemometer was employed to recode these high-resolution minute-based wind speed data. This anemometer, which is robust, resistant to corrosion, and lightweight, features a four-blade helicoid propeller to accurately monitor wind speed and a vane to capture wind direction. The stations under investigation were carefully selected to effectively evaluate the accuracy and robustness of the proposed hybrid forecasting approach in various seasons and at sites with varying meteorological patterns (see Figure 5.1 and Table 5.1).



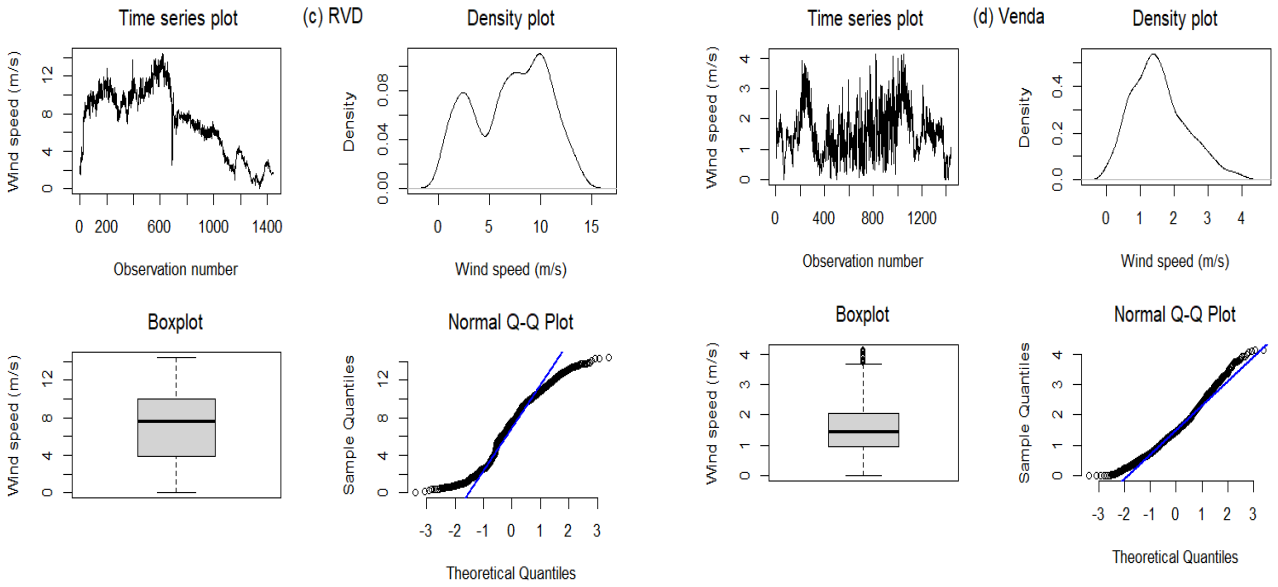


Figure 5.1. The time series and Q-Q plots of minutely averaged wind speed data for the CSIR (a), NUST (b), RVD (c), and Venda (d) stations. Blue lines represent QQ lines, while grey boxes indicate interquartile ranges.

Table 5.1. Location coordinates of the stations.

Station	Latitude (°N)	Longitude (°E)	Elevation (m)	Topography
CSIR	25.746519	28.278739	1400	The roof of a building
NUST	22.565000	17.075001	1683	The roof of the engineering building
RVD	28.560841	16.761459	141	Inside enclosure in a desert region
Venda	23.131001	30.423910	628	Vuwani Science Research Centre

As shown in Table 5.2, the one-minute averaged wind speed data spans four months, corresponding to four seasons in 2019. A total of 1440 observations (covering a full day) were sampled per station. The data for each station are partitioned into a training set (80% or 1152 samples) and a testing set (20% or 288 samples). Models were built and trained using the training dataset, whereas their efficacy was assessed based on the testing dataset.

Table 5.2. Details of minutely averaged wind speed datasets under experimentation.

Station	Season	Month	Sample Size	Training Set	Testing Set
CSIR	Winter	15 August 2019	1440	1152	288
NUST	Autumn	15 May 2019	1440	1152	288
RVD	Spring	7 September 2019	1440	1152	288
Venda	Summer	31 January 2019	1440	1152	288

5.2.2 Summary Statistics

Table 5.3 provides descriptive statistics of the wind speed data for the four (4) stations of interest. NUST and Venda are leptokurtic (kurtosis>3), whereas CSIR and RVD are platykurtic (kurtosis < 3). The RVD exhibits high variability (as indicated by a larger standard deviation value) compared with other datasets. Moreover, the RVD data are skewed to the left (negative skewness), whereas the other datasets are skewed to the right (positive skewness). Furthermore, time series plots, boxplots, quantile-to-quantile (Q-Q) plots, and density plots indicate that data from all stations are non-normal and noisy.

Table 5.3. Summary of the descriptive statistics of the wind speed data sets (in m/s).

Station	Min	Q1	Median	Mean	St. Dev.	Q3	Max	Skewness	Kurtosis
CSIR	0.000	1.812	2.725	2.692	1.116	3.512	8.050	0.143	2.792
NUST	0.000	0.6465	1.1760	1.2772	0.861	1.7820	5.2890	0.769	3.641
RVD	0.036	3.832	7.588	7.125	3.603	10.030	14.400	−0.218	1.947
Venda	0.000	0.962	1.454	1.561	0.812	2.048	4.136	0.579	3.008

5.2.3 Model Settings

Wavelet Analysis

Using decomposition ($L = \text{int}(\log(N)) \approx 3 \leq \text{int}(\log_2(N))$), it was observed that the level of variability declines with increasing decomposition levels. In fact, subseries at lower decomposition levels (i.e., D1 and D2) showed increased variability compared to those at higher decomposition levels (i.e., D3 and A3) (see Figure 5.2).

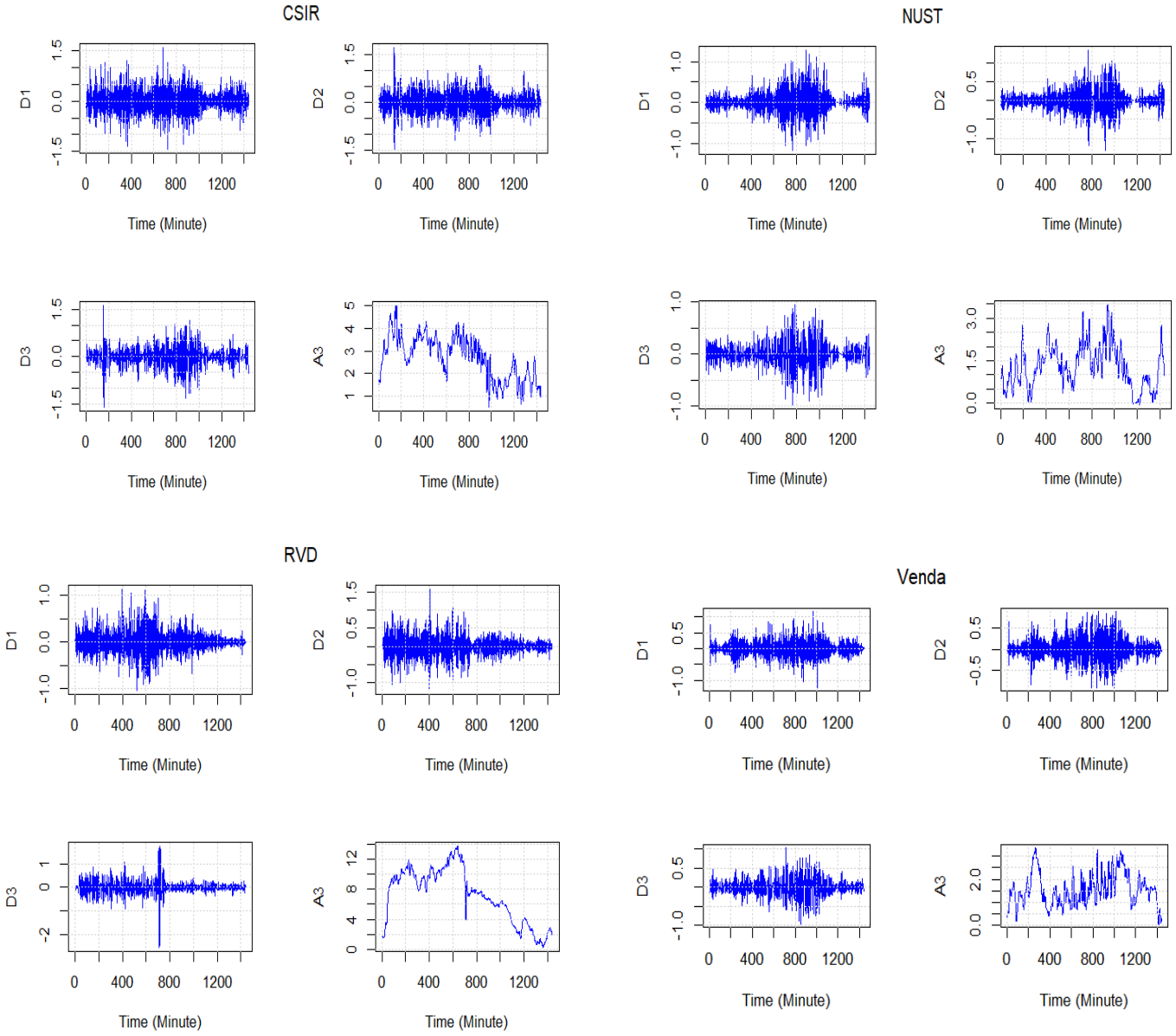


Figure 5.2. Level three MODWT results for minutely averaged wind speed data for CSIR (top left panel), NUST (top right panel), Venda (bottom left panel) and RVD (bottom right panel). D1-D3 denote the detailed coefficients at different decomposition levels and A3 denotes the approximate signal of y_t.

Sample Entropy

To quantify the complexity of each deconstructed subseries the "sample_entropy" function in the "pracma" package in the R program was used. We employed an embedded dimension of $m = 2$ to compute sample entropy. The subseries D1 and D2 showed higher variability (more chaotic) since their SampEn values are much closer to or above 1 (i.e., with a selected threshold ≥0.9) (also see Figure 5.2). Therefore, these signals show greater complexity of time series feature patterns. Conversely, D3 and

A3 have a smaller SampEn, indicative of less random and deterministic time series signals (Table 5.4).

Table 5.4. Computed SampEn values for the wavelet subseries.

Station	D1	D2	D3	A3
CSIR	1.7127	1.4321	0.7617	0.3292
NUST	0.9138	0.9038	0.7038	0.3309
RVD	1.2615	1.1175	0.6430	0.0716
Venda	1.3662	1.1764	0.7288	0.3743

Neural Network Autoregression

The original wind speed data, together with their deconstructed subseries, do not follow the normal distribution (see Tables 5.3 and 5.5). Hence, the following Box–Cox transformation was employed to normalise and stabilise data variability, ultimately enhancing the forecasting performance of the NNAR approach.

$$y_t(\lambda) = \begin{cases} \frac{{y_t}^{\lambda}-1}{\lambda}, \ if\ \lambda \neq 0, \\ ln(y_t), \ if\ \lambda = 0, \end{cases} \tag{5.1}$$

where $ln(.)$ represents the natural logarithm. The parameters for NNAR were automatically selected using the "nnetar" function in the "forecast" R package.

Table 5.5. Standard deviation ($= \sigma$) and skewness ($= \vartheta_{sk}$) for the wind speed subseries datasets.

Station	D1 ($\sigma; \vartheta_{sk}$)	D2 ($\sigma; \vartheta_{sk}$)	D3 ($\sigma; \vartheta_{sk}$)	A3 ($\sigma; \vartheta_{sk}$)
CSIR	(0.3455, 0.0549)	(0.3324, 0.0946)	(0.3287, −0.1607)	(0.9525, −0.1555)
NUST	(0.2550, 0.1959)	(0.2579, 0.2359)	(0.2412, −0.0412)	(0.7424, 0.3316)
RVD	(0.2430, 0.2374)	(0.2794, 0.0740)	(0.3026, −0.5158)	(3.5706, −0.2514)
Venda	(0.2361, 0.0793)	(0.2537, 0.1026)	(0.2388, −0.0025)	(0.6928, 0.4834)

Stateless LSTM

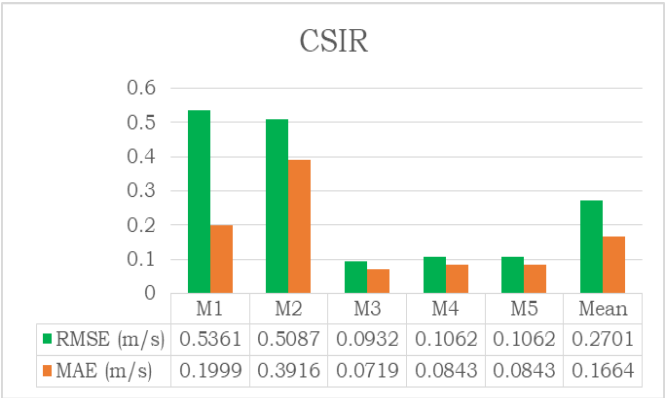
The stateless LSTM was implemented in R using the "keras" library. Different from NNAR, the performance of the LSTM model is highly influenced on the structure of the data. Hence, the LSTM approach was implemented using the parameters outlined in Table 5.6. Also see Appendix A for details.

Table 5.6. Hyperparameter search space for LSTM network for the four datasets.

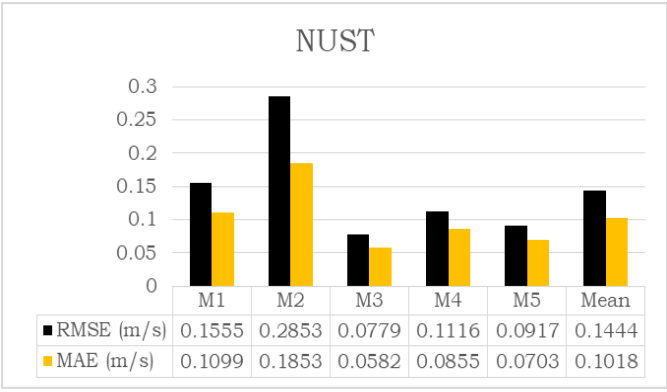
Hyperparameter	Values
Activation function	Hyperbolic
Number of layers	3
Loss function	MSE
Optimiser	ADAM
Stateful	False
Shuffle	False
Learning rate	~1%–2%
Epochs (D1, D2, D3, A3, y_t)	~25–30

5.2.4 Discussion of the Results

Figure 5.3 depicts the step-ahead point forecast performance indicators for the five models trained and fitted on the CSIR, NUST, Venda, and RVD datasets. The objective is to evaluate the prediction accuracy of the proposed hybrid WT-NNAR-LSTM-GBM (M3) model relative to the other four models, viz., benchmark NNAR (M1), LSTM (M2), WT-LSTM-KNN-GBM (M4), and WT-NNAR-KNN-GBM (M5). The proposed M3 outcompetes all other models across all evaluation metrics for CSIR, NUST, and Venda wind speed data. Considering the RVD wind speed data, M4 achieved superior performance in terms of the smallest RMSE. Meanwhile, M3 outperformed M4 in terms of MAE from RVD data. M5 yielded the second-best performance behind M3 for the NUST and Venda stations across all evaluation metrics. For NUST predictions, all models achieved the smallest RMSE, whereas RVD recorded the smallest MAE.



	M1	M2	M3	M4	M5	Mean
RMSE (m/s)	0.5361	0.5087	0.0932	0.1062	0.1062	0.2701
MAE (m/s)	0.1999	0.3916	0.0719	0.0843	0.0843	0.1664



	M1	M2	M3	M4	M5	Mean
RMSE (m/s)	0.1555	0.2853	0.0779	0.1116	0.0917	0.1444
MAE (m/s)	0.1099	0.1853	0.0582	0.0855	0.0703	0.1018

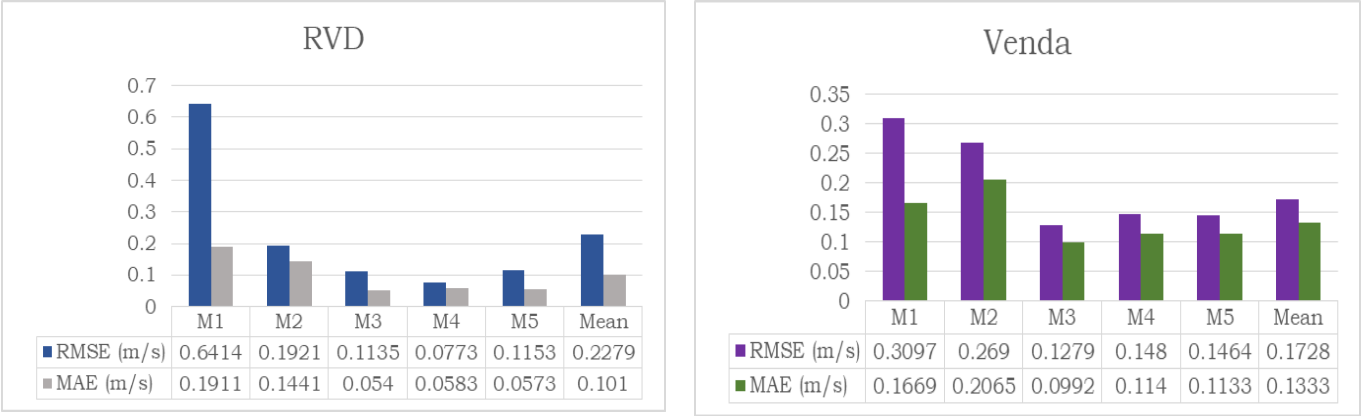


Figure 5.3. Model comparisons using performance metrics for CSIR (top left panel), NUST (top right panel), RVD (bottom left panel), and Venda (bottom right panel)

Between the single model approaches M1 and M2, M2 demonstrated superiority over M1 on the basis of RMSE and MAE values for RVD wind speed data. In contrast, for NUST, M1 outcompeted M2 based on all evaluation metrics. Regarding the CSIR and Venda wind speed data, however, M1 outperformed M2 on the basis of MAE. However, considering the same CISR data, M2 yielded better results than M1 in terms of a smaller RMSE. Notably, results showed that M1 produced better outcomes for low-variance data (NUST and Venda) as compared to high-variance data (CSIR and RVD). Furthermore, while single models (especially M1) exhibited efficiency in terms of computational time, their forecasting performance is inferior to hybrid models.

WT improved the model predictive strength for the four wind speed data of interest. For instance, in the case of the Venda data station, the RMSE values for M1 and M3 are 0.3097 m/s and 0.1279 m/s, respectively. Additionally, SampEn enhances forecasting accuracy by reducing the complexity of hybrid methods. Classifying subseries complexity behaviour through SampEn improved the forecasting performance of the hybrid approach. Thus, WT and SampEn are integral parts of the proposed framework. They help ensure model stability in different locations and seasons of the year. The recommended predictive approach (M3) differs from other models because it effectively captures maximum and minimum wind speed observations (see Figure 5.4). The forecasting accuracy of all models fluctuates depending on the geographical site and the approach used. Based on the evaluation indicators, M3 produced the most stable and reliable predictions at all four sites. Overall, model M3 successfully captured rapid fluctuations in the wind speed data relative to alternative models (see Figure 5.4).

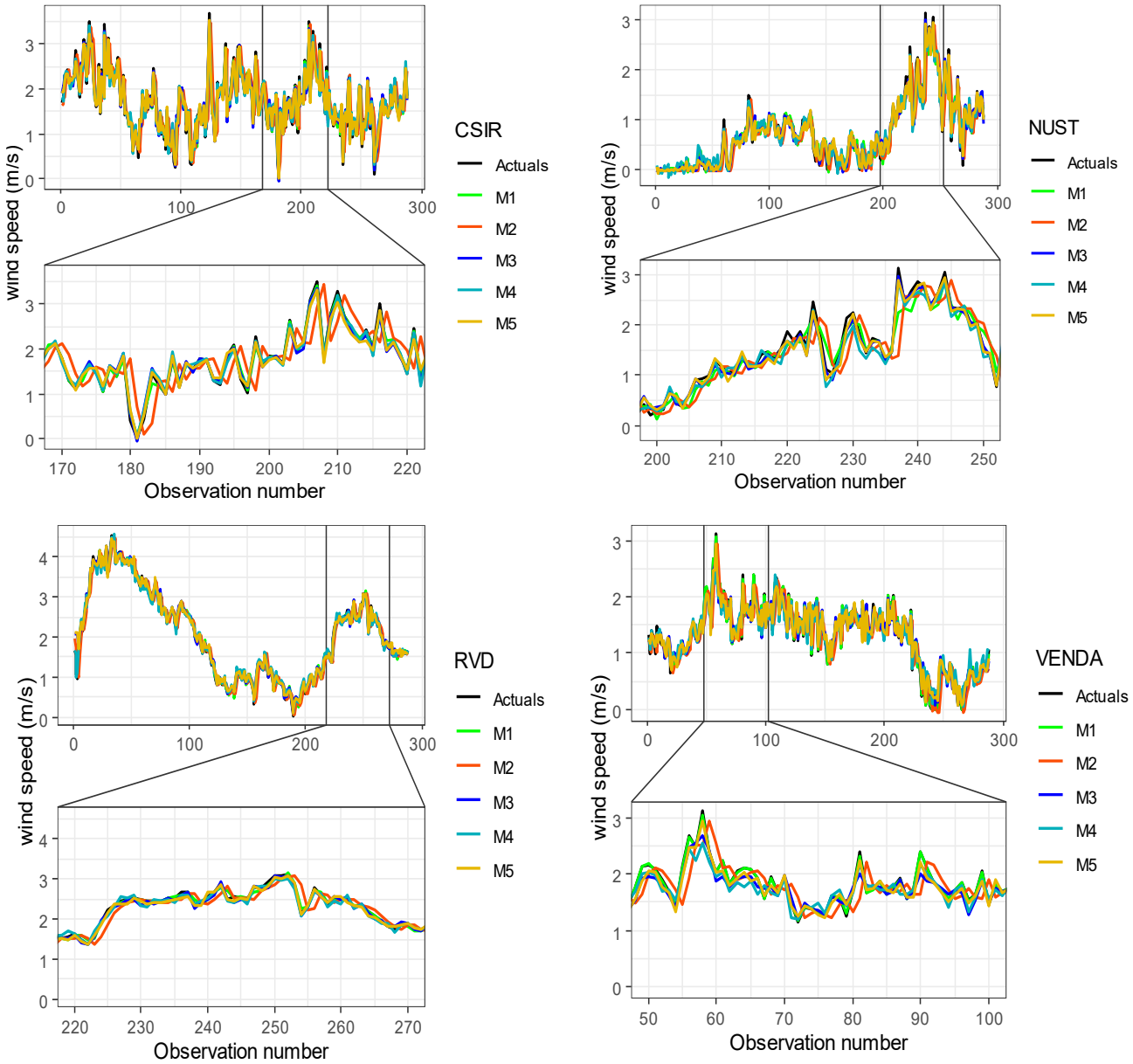


Figure 5.4. Comparison of 288 min predictions and actual wind speed data for CSIR (Top panel), NUST (Second top panel), RVD (Second bottom panel) and Venda (Bottom panel).

Table 5.7 presents the prediction errors of the models implemented on the CSIR, NUST, RVD, and Venda wind speed data, with the superior model bolded. The forecast errors of all models for the NUST and Venda wind speed data are skewed to the right (positively skewed) and their distributions are heavy tailed. This implies predominance of positive errors over negative ones. On the contrary, M3 and M4 overpredict the CSIR and RVD data, as indicated by the negative skewness values. M1 produced the highest positive skewness for CSIR (3.6226), RVD (4.3280), and Venda (3.4968) data, whereas M3 produced the highest negative skewness values (heavy

right rail) for RVD (−6.2854) wind speed data. Using the bias test, the study assessed whether the applied method consistently overestimated or underestimated the observed wind speed data. A balance between 50% underestimation and 50% overprediction is desired. M1 showed the highest bias for CSIR (55%→ $\hat{y}_t > y_t$) and RVD (56%→ $\hat{y}_t > y_t$) data, while M2 produced the highest bias for both the NUST (65% → $\hat{y}_t > y_t$) and Venda (58%→ $\hat{y}_t > y_t$) datasets. Differently, M3 yielded the least bias for CSIR (51%→ $\hat{y}_t > y_t$), RVD (48%→ $\hat{y}_t > y_t$), Venda (50%→ $\hat{y}_t > y_t$), and NUST (50%→ $\hat{y}_t > y_t$) wind speed data (see Figure 5.5).

Table 5.7. Residual analysis of the fitted models for four datasets.

Test	M1	M2	M3	M4	M5
CSIR					
$\xi_t < 0$ (%)	54.8610	**48.6111**	**51.3889**	47.5694	52.0833
$\xi_t > 0$ (%)	45.1389	**51.3889**	**48.6111**	52.4306	47.9167
Std.Dev (m/s)	0.5229	0.5093	**0.0934**	0.1064	0.0954
Skewness (m/s)	3.6226	0.0048	−0.0472	0.0679	0.0242
AD* ($\alpha = 0.05$)	<0.0001	0.06522	0.1147	0.6743	0.05911
NUST					
$\xi_t < 0$ (%)	46.1806	64.5833	**50.0000**	49.3056	49.3056
$\xi_t > 0$ (%)	53.8194	35.4167	**50.0000**	50.6944	50.6944
Std.Dev (m/s)	0.1558	0.2829	**0.0781**	0.1118	0.0918
Skewness (m/s)	0.7140	0.7507	0.2284	0.3877	0.2161
AD* (α=0.05)	<0.0001	<0.0001	0.0043	0.0059	0.0522
RVD					
$\xi_t < 0$ (%)	55.5556	52.4306	**47.5694**	**47.5694**	46.8750
$\xi_t > 0$ (%)	44.4444	47.5694	**52.4306**	**52.4306**	53.1250
Std.Dev (m/s)	0.6257	0.1913	0.1132	**0.0774**	0.1150
Skewness (m/s)	4.3280	0.1282	−6.2854	- 0.0108	−5.8903
AD* (α=0.05)	<0.0001	0.01416	<0.0001	0.01404	<0.0001
VENDA					
$\xi_t < 0$ (%)	51.7361	57.6389	**49.6528**	48.2638	46.8750
$\xi_t > 0$ (%)	48.2639	42.3611	**50.3472**	52.4306	53.1250
Std.Dev (m/s)	0.2996	0.2668	**0.1281**	0.1483	0.1466
Skewness (m/s)	3.4968	0.0625	0.0669	0.1281	0.2871
AD* (α=0.05)	<0.0001	0.00654	0.5505	0.1815	0.1412

Std.Dev= Standard deviation; AD*=Anderson–Darling test; $\xi_t = y_t - \hat{y}_t$

Considering the standard deviation values, M3 yielded the narrowest residual spread for the CSIR, NUST, and Venda datasets. For the RVD dataset, model M4 achieved the narrowest residual speed among all models, followed by M3, M5, M2, and M1. With the exception of the NUST dataset, where M2 produced the widest residual spread,

M1 exhibited the largest residual spread for the CSIR, RVD, and Venda datasets (also see Figure 5.5).

Using the Anderson–Darling (AD) test, the study evaluated whether the forecasting performance of the applied approaches differs for the four different datasets of interest. In essence, AD test enables one to assess the normality and homoscedasticity behaviour of residuals. For the NUST dataset, the AD test indicated that residuals from M5 were normally distributed (i.e. p-value> 0.05) from the model, while those residuals from M1 for the CSIR and Venda data did not pass the AD normality test (i.e. p-values< 0.05). Overall, residual analysis revealed that M3 yielded the least biased and most accurate predictions for the CSIR, NUST, and Venda data relative to other models. On the other hand, M4 produced the most biased and least accurate predictive results in the RVD dataset (see Figure 5.5).

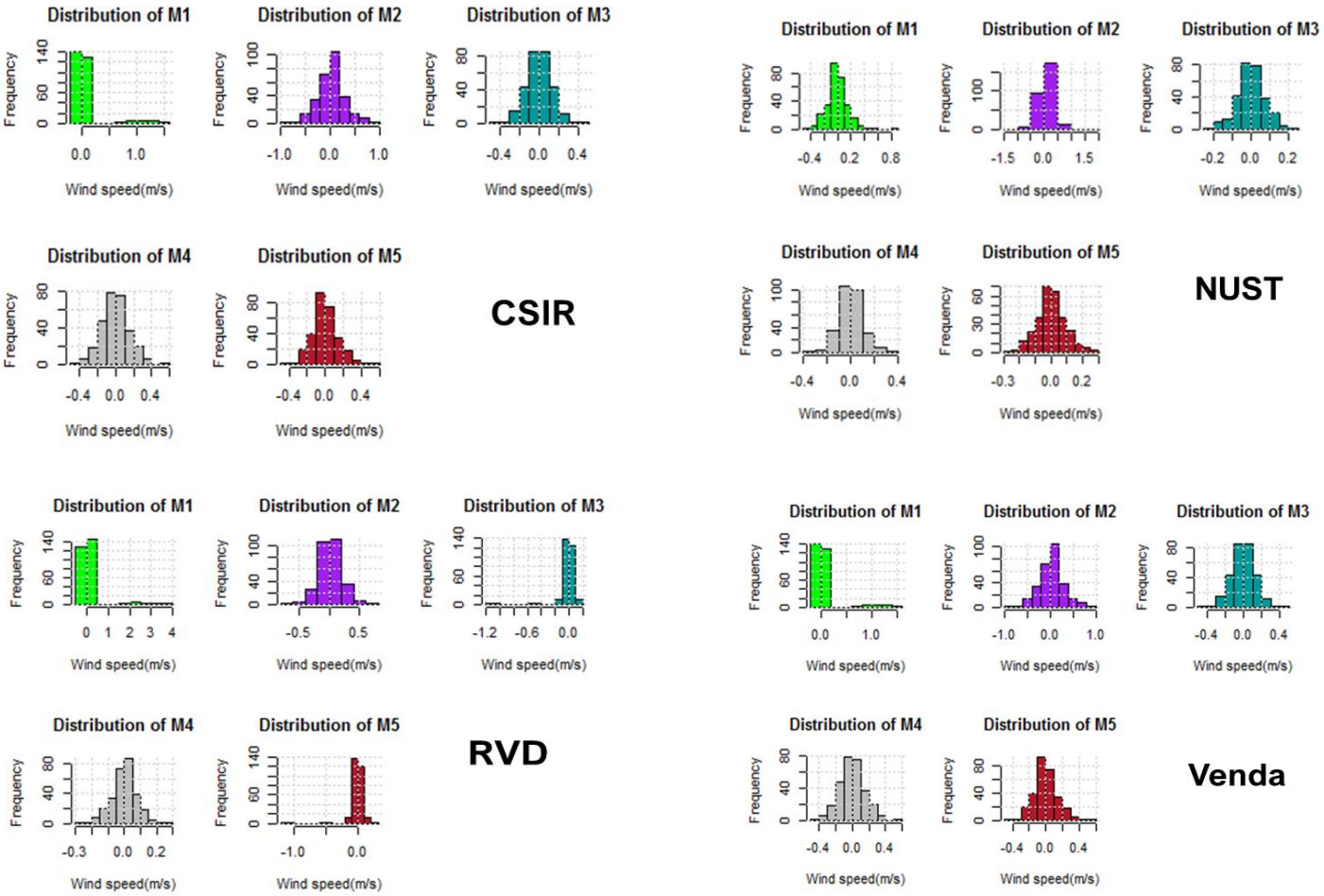


Figure 5.5. Distributions of the residuals for CSIR (top left panel), NUST (top right panel), RVD (bottom left panel), and Venda (bottom right panel).

Table 5.8 summarises the MAD, CRPS, and PIW results for the 90% PIs for all the models applied to the wind speed datasets under study. The results of the superior model are bolded. In this study, MAD is employed to evaluate the sharpness of the applied approaches. M3 yielded the lowest MAD value for the CSIR, NUST, and RVD data, whereas M1 produced the smallest MAD value for the Venda data. Thus, the model M3 produced the narrowest probability distributions for the CSIR, NUST, and RVD datasets. For Venda data, M1 produced forecasts that exhibited the narrowest distribution. Consequently, M3 is the sharpest for the CSIR, NUST, and RVD wind speed data, whereas M1 is the sharpest for the Venda data.

Table 5.8. Comparative analysis of models using scoring rules and PIW.

	M1	**M2**	**M3**	**M4**	**M5**	**Mean**
			CSIR			
MAD (m/s)	0.0886	0.4808	**0.0778**	0.1006	0.0834	0.1662
St.Dev PIW (m/s)	0.5903	0.1373	0.0012	0.0118	**0.0005**	0.1482
OL (count)	**25**	30	27	29	27	28
CRPS (m/s)	0.4017	0.3541	0.3439	**0.3358**	0.3423	0.3555
			NUST			
MAD (m/s)	0.1106	0.1527	**0.0595**	0.0950	0.0829	0.1021
St.Dev PIW (m/s)	0.0950	0.4450	**0.0074**	0.0118	0.0247	0.1178
OL (count)	28	28	28	29	28	28
CRPS (m/s)	0.3551	0.3634	0.3540	0.3503	**0.3412**	0.3528
			RVD			
MAD (m/s)	0.0582	0.1664	**0.0521**	0.0640	0.0575	0.0796
St.Dev PIW (m/s)	0.7608	0.0569	**0.0075**	0.0283	0.0075	0.1722
OL (count)	28	**26**	**26**	30	28	28
CRPS (m/s)	0.6869	0.6368	0.6247	**0.6141**	0.6238	0.6373
			VENDA			
MAD (m/s)	**0.0187**	0.2390	0.1110	0.1316	0.1354	0.1045
St.Dev PIW (m/s)	0.3564	0.1485	0.0143	**0.0066**	0.0284	0.1108
OL (count)	28	27	27	27	32	28
CRPS (m/s)	0.3740	0.3138	0.2873	**0.2746**	0.2754	0.3050

OL = Outside PI limits.

According to 90% PIW (standard deviation values), M3 produced the narrowest PIW for NUST and RVD data. Models M5 and M4 achieved the narrowest PIW for CSIR and Venda data, respectively. On the contrary, M1 yielded the broadest PIW for CSIR, RVD, and Venda data. Likewise, M2 achieved a wider PIW than the other models for NUST data. An average of 260 predicted values (i.e., over 90%) were contained within 90% PI. For the CSIR dataset, M1 produced the most predictions within 90% PIs. In

the case of RVD data, M2 and M3 produced the most forecasts within 90% of PIs. Conversely, for the NUST dataset, M4 yielded the most values outside the PIs. For the Venda dataset, M5 recorded the largest forecast count outside of the 90% PI.
The CRPS evaluates the calibration or reliability of the applied approaches. The CRPS metric demonstrated the superiority of M4 over other models in forecasting CSIR, RVD, and Venda data. Hence, models M5 and M4 are evidently better calibrated to predict NUST and Venda data, respectively. Model M4 emerged as the best-calibrated approach. The overall comparison indicates that hybrid frameworks (M3 and M4), which incorporate stateless LSTM, are well-calibrated and can effectively explain the transient features inherent in the wind speed data. Consequently, M3 and M4 produced more accurate and stable forecasts that better generalises to all four datasets under study.

Table 5.9. Percentage (%) error reduction due to M3 using NUST data.

Metric	M3:M1	M3:M2	M3:M4	M3:M5
RMSE	49.90	72.70	30.20	15.05
MAE	47.04	68.59	31.93	17.21

Table 5.9 presents error reduction analysis due to M3 based on the NUST dataset. The results show a substantial reduction in error metrics across all models. For instance, for the selected hybrid model (M3), the error reduction in RMSE for models M1 and M2 increased by at least 49.90% and 72.70% respectively, whilst that for models M4 and M5 increased by 30.20% and 15.05% respectively. These results are pivotal for making well-informed decisions about resource allocation such as power grid stability management and wind turbine operations.

5.4 Conclusions

The proposed WT-NNAR-LSTM-GBM framework aims to mitigate inherent wind speed turbulence and chaotic behaviour as well as enhancing short-term wind speed forecasting. To assess the efficacy and robustness of the proposed strategy relative to WT-NNAR-KNN-GBM, WT-LSTM-KNN-GBM, LSTM, and baseline NNAR, minute-averaged wind speed data sourced from CSIR, NUST, RVD, and Venda radiometric stations in Southern Africa were employed. The wavelet transformation of turbulent and chaotic wind speed data into more statistically improved subseries enhanced the

forecasting accuracy of the hybrid approach for all four sites, whilst SampEn minimised forecasting computational complexity and enhanced predictive accuracy by aligning and adjusting the predictive approaches to the unique and complex characteristics of the wind speed subseries. Due to the stateless LSTM, the hybrid framework was able to forecast more accurately, reliably, and robustly by capturing extreme wind speed variations associated with turbulence. Though computationally expensive, LSTMs proved effective and beneficial for wind speed prediction due to their inherent ability to recognise patterns and handle vanishing gradients. Forecast reconciliation via nonlinear GBM proved effective, efficient, and accurate over various datasets and seasons. Through assessment of point and probabilistic data metrics, along with residual analysis, findings confirmed that WT-NNAR-LSTM-GBM (with the least sensitivity to season and location) shows superior accuracy, sharpness, robustness, and reliability across diverse terrain and weather conditions. These results could be used for even wind power distribution and optimisation, thereby ensuring smooth grid operations in real-time.

5.5 Contributions

This Chapter proposed an advanced hybrid WT-NNAR-LSTM-GBM framework for enhancing wind speed prediction at the short-term forecast horizon. This approach is based on a multi-model combination approach, integrating data denoising and deconstruction methods, complexity evaluation, separate subseries modelling and forecasting, and nonlinear forecast combination. The WTs were endorsed for breaking down chaotic and erratic wind speed data into subsignals characterised by clear patterns and trends with sound statistical features compared to the original wind speed signals. To a certain degree, this reduces computational costs and improves prediction accuracy. The SampEn showed efficiency and effectiveness in the classification of deconstructed signals based on their complex and deterministic attributes. As a result, the most appropriate models were utilised to enhance forecasting accuracy. In fact, NNAR models leveraged the SampEn approach to accurately forecast subseries characterised by low randomness. Robust stateless LSTMs successfully mitigated the challenge of vanishing gradients (of which NNAR is prone to) by accurately predicting subseries classified as highly complex through SampEn. A highly scalable, robust, nonlinear GBM approach, preferred over a conventional linear combination approach, successfully reconciled nonlinear wind

speed subseries predictions with such a level of accuracy and speed. In an effort to improve wind speed forecasting, Chapter 6 proposes a hybrid approach, wavelet-MODWT-GRU, aimed to address the influence of the wavelet filter and decomposition level on the forecasting accuracy of the wavelet-ML hybrids (Also see Chapter 2 for details).

This page is intentionally blank

Chapter 6

Wind Speed Forecasting Using Differentially Evolved Minimum-Bandwidth Filters and Gated Recurrent Units

6.1 Introduction

The unparalleled prediction accuracy and, to some extent, computational efficiency of hybrid methods have attracted considerable interest from researchers in recent times. Besides, hybrid methods are capable of handling and mitigating the inherent statistical, computational, and representational deficiencies present in the prediction tasks. In the reviewed work, the studies often employed the standard DWT, which (though effective to some extent) is shift-variant so that small variations in the signal influence wavelet coefficients, thereby compromising their ability to effectively undertake time series tasks as compared to the more effective and time-invariant MODWT (see Chapter 2 for more details). In fact, the reviewed work rarely investigated MODWT, nor did they thoroughly outline the mathematical approach employed to determine the decomposition level. Instead of applying a reproducible technique, the decomposition level is frequently computed through trial and error. Moreover, the majority of reviewed work applied single-wavelet filters (mainly DB) in the data preprocessing stage, with minimal adoption and implementation of MB filters. Also, the application of the DE as a metaheuristic approach for optimising wavelet decomposition levels in wind speed prediction is also underreported in the literature, relative to GA and other metaheuristic approaches. Overall, very few studies provided a clear rationale for choosing a particular wavelet approach and decomposition levels in wind forecasting, thereby limiting the comparability and reproducibility of the results. Hence, this chapter presents a new class of hybrids, namely; wavelet-MODWT-GRU framework, to effectively evaluate the effect of the wavelet filter and decomposition level on the accuracy and robustness of wind speed forecasts. The efficacy of the three types of wavelet-MODWT-GRU strategies, namely;

LA-MODWT-GRU, DB-MODWT-GRU, MB-MODWT-GRU, GRU, and the baseline Naïve model is thoroughly assessed using both deterministic and probabilistic predictions.

6.2 Empirical Results

6.2.1 Data Description

This study employs 10-minute averaged wind speed data sourced from the three WASA stations, viz., Alexander Bay, Humansdorp, and Jozini (see Chapter 1 for the link). A detailed overview of the site for each of the data stations of interest is presented in Table 6.1. The wind speed series is divided into two subsets, viz., the training set (80%) and the testing set (20%). Methods were tuned and developed on the training set, while the testing set served to test and assess forecasting abilities and reliability of the models in different years, seasons, patterns, terrains, and forecasting time scales.

Table 6.1. Training and testing dataset.

Station	Month	N	Granularity	Training	Testing
Alexander Bay	1–31 August 2022	4462	10 min	3570	892
Humansdorp	1–25 April 2021	3600	10 min	2800	720
Jozini	1–17 December 2020	2400	10 min	1920	480

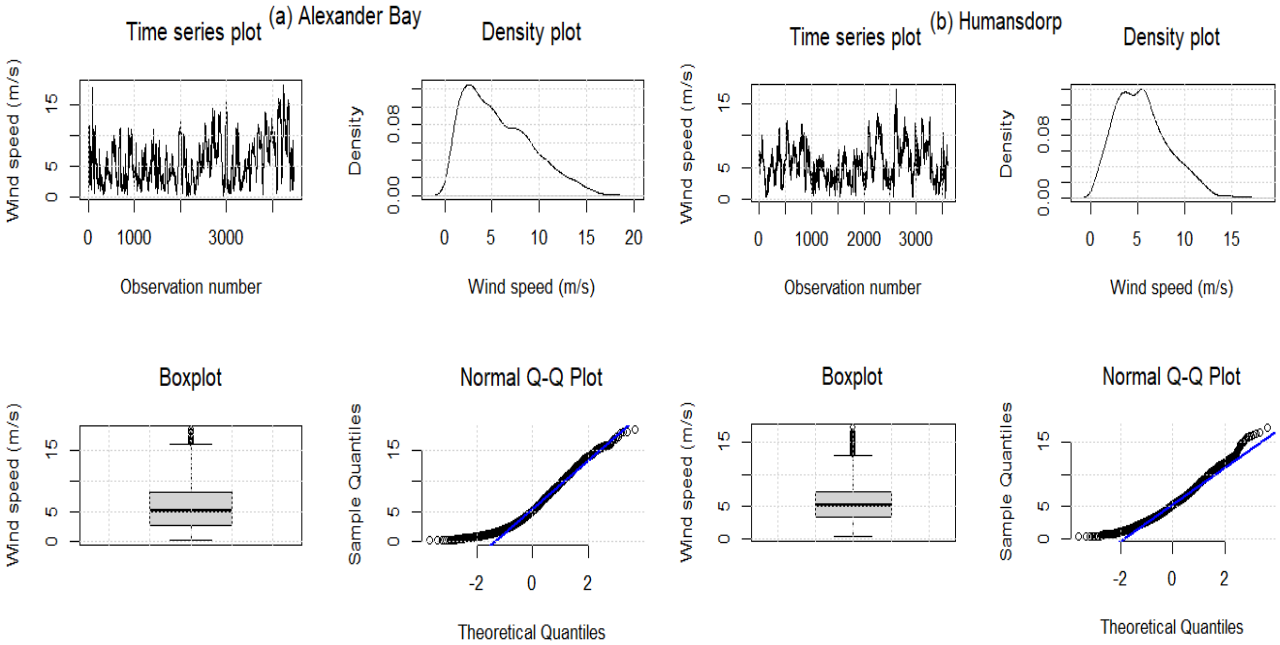
Positioned at latitude of −28.601882, longitude 16.664410, and with an elevation of 152 m, the first station identified by Mast ID WM01 is located in Alexander Bay, a desert region of the Northern Cape. A second station is situated at Humansdorp in the Eastern Cape with Mast ID WM08, longitude 24.514360, latitude −34.109965, and elevation 110 m (also see Figure 5.2). A third site, identified by Mast ID WM13 is located in the Jozini region of KwaZulu-Natal at longitude 32.16636, latitude −27.42605, and elevation 80 m (see Table 6.2). Each of the data points of each station consists of 10-minute averaged wind speed with varying features depending on the location (see Table 6.2).

Table 6.2. Location description for the stations.

Station	Mast ID	Longitude (°E)	Latitude (°N)	Elevation (m)	Anemometer Height (m)
Alexander Bay	WM01	16.664410	28.601882	152	61.85
Humansdorp	WM08	24.514360	34.109965	110	61.84
Jozini	WM13	32.16636	27.42605	80	61.75

6.2.2 Summary Statistics

The summary statistics for wind speed data for the three WASA stations of interest are presented in Table 6.3. The positive skewness values show that wind data at each station are skewed to the right (see Figure 6.1). Additionally, the Jarque–Bera (JB) test (with p-values<0.05) demonstrates that all three datasets are not normally distributed. This is further apparent in the density plots which are positively skewed; the data of interest exhibit time-varying variance, resembling the features of cyclostationary. When considering standard deviation values, the wind data recorded at Alexander Bay are more variable relative to other stations. Moreover, Alexander Bay and Jozini's wind speed data are platykurtic (i.e. kurtosis< 3). The wind speed data for Humansdorp have a kurtosis value greater than 3, which means the distribution is leptokurtic.



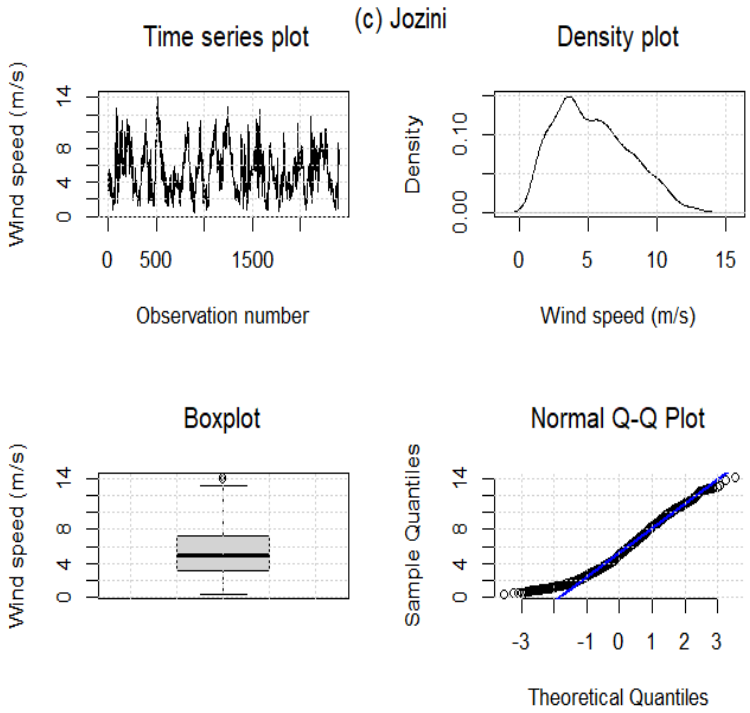


Figure 6.1. Wind speed data for Alexander Bay (**a**), Humansdrop (**b**), and Jozini (**c**). Lines in blue represent QQ lines and boxes in grey indicate interquartile ranges.

Table 6.3. Descriptive statistics for wind speed data (in m/s) at the three stations of interest.

Station	Min	Q1	Median	Mean	Q3	Max	Std.dev	Skewness	Kurtosis	JB (*p*-Value)
Alexander Bay	0.210	2.790	5.030	5.727	8.140	18.360	3.5739	0.6861	2.7581	$<2.2 \times 10^{-16}$
Humansdorp	0.2349	3.3596	5.2340	5.5351	7.2610	17.3089	2.8532	0.6385	3.1533	$<2.2 \times 10^{-16}$
Jozini	0.4045	3.2022	5.0106	5.3241	7.2137	14.1427	2.6980	0.4722	2.5046	$<2.2 \times 10^{-16}$

6.2.3 Model Settings

To accommodate the DE algorithm, the study employed the "DEoptim" library, while the "wavelism" and "wavelets" libraries were utilised for the selection of appropriate wavelet filters, including "d4", "mb8", and "la8". The "modwt" function from the "waveslim" library was instrumental in decomposing wind speed data sets into various subsignals with varying frequency components. The GRU model was established and subsequently fitted using the "keras" library, a Python package (version 3.12). A systematic approach involving metaheuristic, early stopping, and dropout regularisation was employed to identify the best hyperparameters for the models. The resulting best range of parameters is outlined in Table 6.4.

Table 6.4. Model hyperparameters.

Model	Main Hyperparameter	Search Space
DE	Number of iterations	50–60
	Population size	45–60
	Crossover probability	0.75–0.85
	Weights	0.5–0.6
	Bounds	1–10
MODWT	filters	"la8", "d4", "mb8"
GRU	Dropout rates	0–0.5
	Time steps	1–10
	Epochs	1–100
	Learning rate	0–0.1
	Activation function	tanh
	Loss function	MSE
	Optimiser	Adam

6.2.4 Discussion of the Results

Table 6.5 presents the findings of the point performance metrics for the methods applied at the three WASA stations of interest, viz., Alexander Bay, Humansdorp, and Jozini. Ideally, this section would like to evaluate the influence of wavelet filters, level decomposition, station site, and prediction horizon on the proposed hybrid framework in terms of point and probabilistic forecasts.

Table 6.5. Predictive performance indicators for the three wavelet filter models.

Data	Model	Performance Indicator				
		RMSE	MAE	MAPE (%)	R^2	MZ Bias Test
Alexander Bay (August,WM01) N = 4462 h = 892				$L = 3$		
	M1 (LA8)	0.5256	0.3968	14.1156	0.9810	Unbiased
	M2 (DB4)	0.5562	0.4050	14.3321	0.9792	Biased
	M3 (MB8)	**0.5102**	**0.3787**	**12.2258**	**0.9824**	Biased
				$L = 4$		
	M1 (LA8)	**0.7557**	**0.5847**	37.2397	**0.9607**	Unbiased
	M2 (DB4)	0.8676	0.6520	**15.4997**	0.9505	Biased
	M3 (MB8)	0.7882	0.6007	26.4022	0.9585	Biased
				$L = 5$		
	M1 (LA8)	1.2909	0.9729	15.8073	0.8872	Biased
	M2 (DB4)	1.3119	0.9923	16.8702	0.8839	Biased
	M3 (MB8)	**1.2150**	**0.9081**	**14.7143**	**0.8983**	Unbiased
Humansdorp (April, WM08) N = 3600 h = 720				$L = 3$		
	M1 (LA8)	0.4767	0.3580	**8.8326**	0.9718	Unbiased
	M2 (DB4)	0.5482	0.4059	10.4977	0.9631	Biased
	M3 (MB8)	**0.4678**	**0.3544**	9.1258	**0.9729**	Unbiased
				$L = 4$		
	M1 (LA8)	0.8640	0.6027	14.3734	0.9073	Unbiased

	M2 (DB4)	**0.7322**	**0.5198**	**12.2237**	**0.9335**	Unbiased
	M3 (MB8)	0.7634	0.5512	12.5808	0.9276	Unbiased
			$L = 5$			
	M1 (LA8)	1.0015	0.7537	16.5959	0.8754	Unbiased
	M2 (DB4)	0.9003	0.7154	16.6009	0.8994	Unbiased
	M3 (MB8)	**0.8619**	**0.6750**	**15.2706**	**0.9077**	Unbiased
Jozini (December, WM13) N = 2400 h = 480			$L = 3$			
	M1 (LA8)	0.7092	0.4924	10.6034	0.9116	Unbiased
	M2 (DB4)	0.7167	0.5002	10.9543	0.9098	Unbiased
	M3 (MB8)	**0.6703**	**0.4685**	**10.0479**	**0.9210**	Unbiased
			$L = 4$			
	M1 (LA8)	0.7789	0.5652	12.7938	0.8935	Unbiased
	M2 (DB4)	**0.7566**	0.5512	12.3017	**0.8995**	Unbiased
	M3 (MB8)	0.7693	**0.5350**	**11.5570**	0.8962	Unbiased
			$L = 5$			
	M1 (LA8)	0.8927	0.7088	15.5410	0.8599	Unbiased
	M2 (DB4)	**0.8096**	**0.6266**	**13.2615**	**0.8848**	Unbiased
	M3 (MB8)	1.0169	0.8060	18.7082	0.8194	Unbiased

To thoroughly assess the forecasting performance of the recommended decomposition level (i.e., $L = 3$), other decomposition levels (i.e., $L = 4$ and $L = 5$) were employed to accomplish the same exercise through the wavelet-MODWT-GRU framework. In addition, LA8 (referred to as M1), MB8 (referred to as M3), and DB4 (referred to as M2) are utilised for effective benchmarking. Further, the study evaluated the predictive strength of the superior (or best) wavelet-MODWT-GRU framework against the individual GRU and the baseline Naïve model.

Considering the Alexander Bay wind speed data, M3 (at $L = 3$) outcompeted other models in terms of the smallest RMSE, MAE, MAPE, and highest R^2 values. According to the MZ test, model M1 (at $L = 3$ and $L = 4$) showed unbiasedness; whilst M3 remained unbiased at $L = 5$. Based on the same performance metric, M1 (at $L = 3$), yielded the second-highest performance, whereas M2 (at $L = 5$) produced the poorest results. Generally, forecasting accuracy is the highest at $L = 3$ and diminishes with the increase in the level of decomposition.

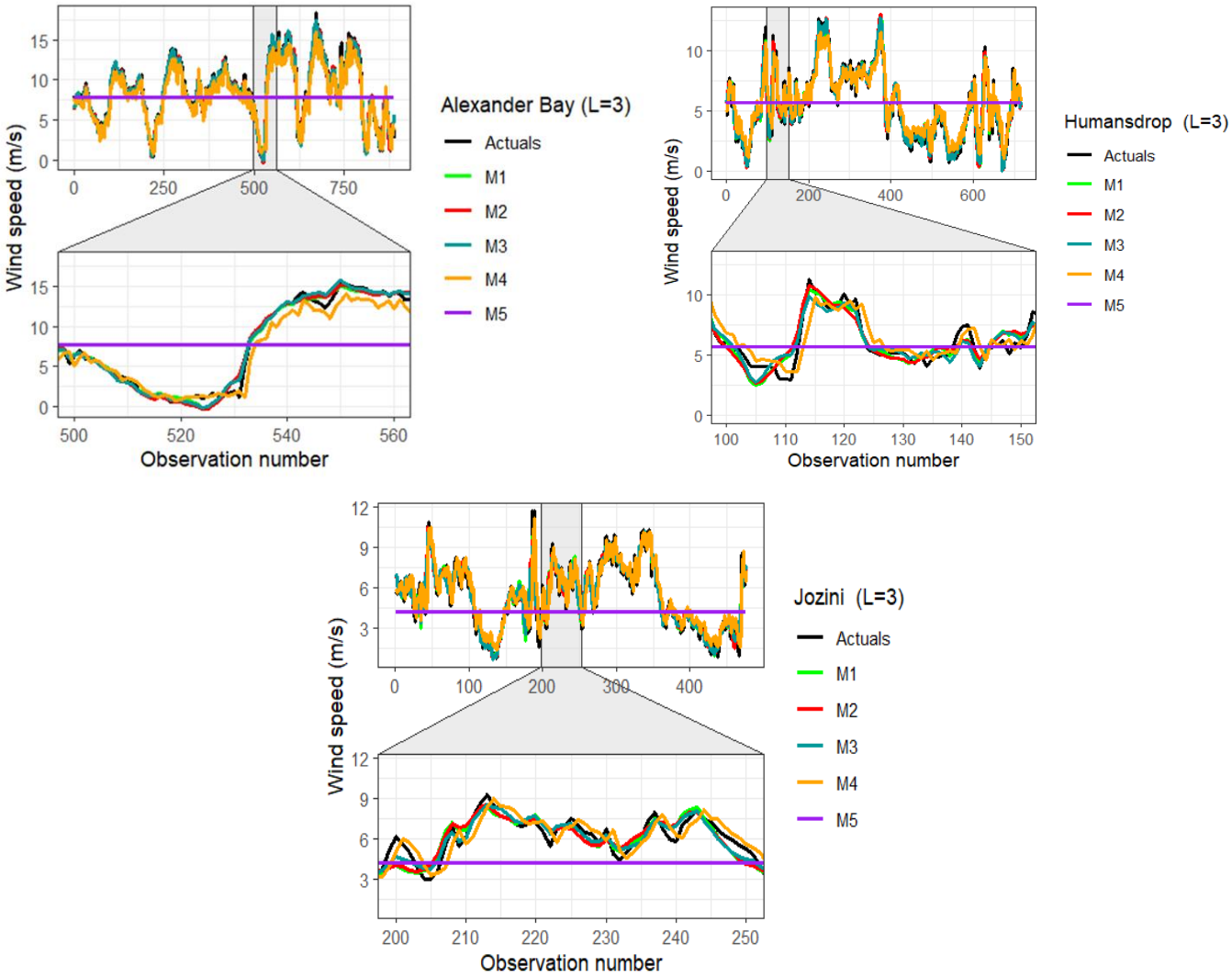


Figure 6.2. Comparison of wind speed predictions against actual wind speed data for Alexander Bay (left panel), Humansdrop (right panel), and Jozini (bottom centre panel).

Model M3 (at $L = 3$ and $L = 5$) demonstrated superiority over other approaches when considering the least RMSE, MAE, and the highest R^2 at the Humansdorp site. Similar to the Alexander Bay site, the decomposition level optimised at $L = 3$ yielded the most accurate prediction relative to decomposition levels (at $L = 4$ and $L = 5$). Overall, the predictive ability of models declined with the increase in the level of decomposition. With the exclusion of M2 (at $L = 3$), the MZ test showed that all approaches exhibited unbiasedness at all decomposition levels. Overall, M3 (at $L = 3$) yielded the most accurate performance for the Humansdorp wind speed data.

At the Jozini station, M3 (evaluated at $L = 3$) showed overall superiority over other approaches and decomposition levels (i.e., $L = 4$ and $L = 5$). For instance, M3 (examined at $L = 3$) outcompeted all other models based on the least MAE, MAPE,

RMSE, and highest R^2. Furthermore, forecasts from all models exhibited unbiasedness based on the MZ test (see Figure 6.2).

The comparative assessment demonstrated that the forecasting strength of the hybrid framework was influenced by site location, the filter function employed, the level of decomposition, and the prediction horizon. Furthermore, models yielded the most satisfactory predictive performance on shorter forecast horizons than at longer forecast horizons. For example, M2 (at $L = 5$) recorded the largest RMSE = 1.3119 for the Alexander Bay dataset with a forecast horizon ($h = 892$), whereas M3 (at $L = 3$) (RMSE = 0.4678) achieved the smallest RMSE for the Humansdorp dataset with $h = 720$. The MODWT decomposition at higher levels yields more statistically sound subsignals, but also produces larger error accumulations.

Table 6.6. Point performance indicators for the best wavelet filters model (at $L = 3$) against the GRU and naïve model.

Model	Performance Indicator			Skilled Indicator			Bias
	RMSE	**MAE**	**MAPE (%)**	**RMSE**	**MAE**	**MAPE**	**MZ Test**
			Alexander Bay				
M3 (MB8)	**0.5102**	**0.3787**	12.2258	**0.8705**	**0.8819**	0.7045	Biased
M4 (GRU)	0.7147	0.5099	**8.2808**	0.8186	0.8410	**0.7999**	Biased
M5 (Naive)	3.9391	3.2069	41.3795				Biased
			Humansdorp				
M3 (MB8)	**0.4768**	**0.3544**	**9.1258**	**0.8321**	**0.8493**	**0.7793**	Unbiased
M4 (GRU)	0.7557	0.5376	12.7266	0.7340	0.7714	0.6923	Unbiased
M5(Naive)	2.8406	2.3516	41.3546				Biased
			Jozini				
M3 (MB8)	**0.6703**	**0.4685**	**10.0479**	**0.7482**	**0.7885**	**0.8093**	Unbiased
M4 (GRU)	0.8497	0.6377	13.5132	0.6808	0.7121	0.7435	Biased
M5 (Naive)	2.6616	2.2151	52.6837				Biased

Bold = Best model

In Table 6.6, the study presents the results of the best wavelet filter model at level 3 in comparison to M4 (GRU) and the baseline model M5 (Naïve). We also employed the skilled score indicator to determine the forecast performance gain relative to M5. The proposed hybrid framework demonstrated superiority to the M4 and M5 approaches in terms of the least RMSE and MAE across the three datasets (also see Figure 6.2). Furthermore, the developed hybrid framework yielded the best performance for the Humansdorp and Jozini data, as indicated by the least MAPE values, whereas M4 achieved superior results for Alexander Bay based on MAPE. Based on the MZ test, M3 results exhibited unbiasedness for the Humansdorp and Jozini data, whereas M5

results exhibited bias for all datasets. The point performance indicators demonstrated that M3 achieved the best predictive performance across the three sites.

A comparative assessment of the methods based on the probabilistic error indicators, viz., PL, DS, and PIT, is presented in Table 6.7. The PL and DS scores were employed to assess prediction accuracy and uncertainty, while the PIT scores were utilised to evaluate calibration of the models. Smaller values of the PL, DS, and PIT scores indicate better probabilistic predictions. Moreover, a uniform PIT histogram indicates a model with unbiased predictions.

Table 6.7. Distributional forecast accuracy indicators for the best wavelet filter model (at $L = 3$) against the naïve (M5) and GRU (M4) model.

Model	PL Score	DS Score	PIT Score	
	τ = 0.95	Mean	KS Test (D)	KS (p-Value)
		Alexander Bay		
M3 (MB8)	0.7029	**3.6753**	**0.0309**	**0.3633**
M4 (GRU)	**0.6635**	3.6798	0.0503	0.0223
M5 (Naive)	0.7187	3.7441	0.1199	1.615×10^{-11}
		Humansdorp		
M3 (MB8)	0.6361	**3.0853**	0.0481	**0.0729**
M4 (GRU)	0.6020	3.0905	**0.0428**	**0.1451**
M5 (Naive)	**0.5204**	3.0881	0.0677	0.0028
		Jozini		
M3 (MB8)	0.4286	**2.7419**	**0.0689**	0.0217
M4 (GRU)	**0.4248**	2.7618	0.0902	0.0009
M5 (Naive)	0.5156	2.9833	0.2192	$<2.2 \times 10^{-16}$

Model M4 yielded the smallest PL for the Jozini and Alexander Bay data, whereas M5 achieved the smallest PL value for the Humansdorp data. For all datasets, M3 outperformed M4 and M5 based on the smallest DS score. Based on the same metric, M4 dominated the baseline M5 for the Alexander Bay and Jozini data. Across the three datasets, M3 delivered the most accurate and reliable predictions.

For the Alexander Bay and Jozini data, the PIT test revealed that M3 achieved better results than M4 and M5, as indicated by the least deviation between the actual and predicted results. The Kolmogorov-Smirnov (KS) test for M3, with p-values more than 0.05 for the Alexander Bay and Humansdorp data, indicates that the distribution of the PIT values does not significantly deviate from uniformity. This suggests that the forecasts from M3 are unbiased and well-calibrated for the Alexander Bay and Jozini datasets. Despite having the p-value < 0.05 at the Jozini site, M3 still delivered the best results when compared to M4 and M5 (see Figure 6.3). Overall, model M3

achieved the most accurate, reliable, and better-calibrated results for all three datasets. Across all datasets, preprocessed methods through MODWT (at $L = 3$) demonstrated superiority in terms of forecast accuracy, reliability, and calibration over those that had not been preprocessed.

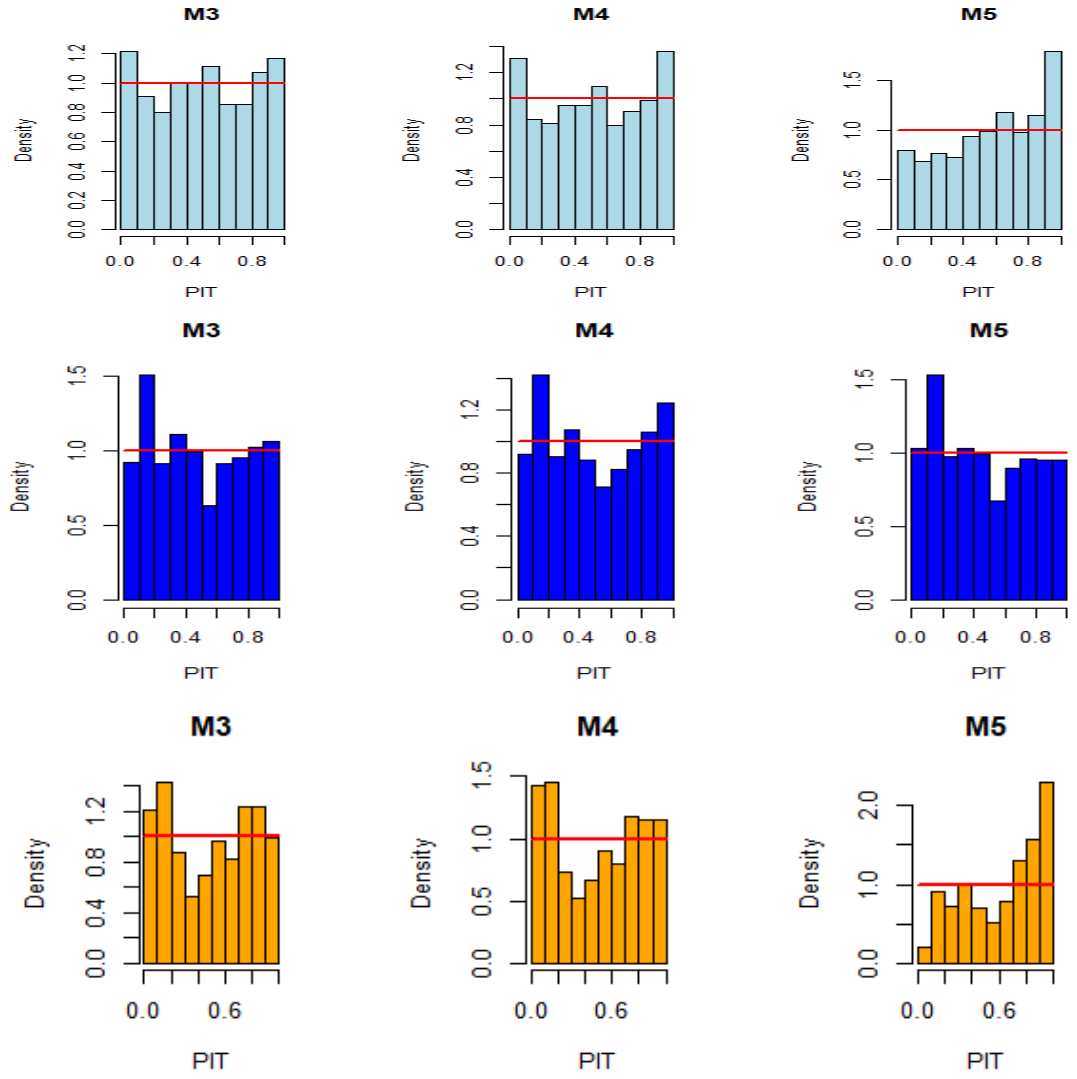


Figure 6.3. PIT Histograms for the Alexander Bay (top panel), Humansdorp (middle panel), and Jozini (bottom panel) comparing models M3, M4, and M5.

Figure 6.4 presents the MDs for the effective comparison of the predictive performance of the suggested hybrid framework against M4. In the top panel (Alexander Bay dataset), M3 dominates M4. This is further confirmed by the blue line, which is consistently below the red line. Furthermore, the difference in scores, which is overall negative, exhibited a wider and broader 95% PI. For the Humansdorp dataset (middle panel), M3 again displayed clear dominance over M4. In this case, the 95% PI of the difference in scores is slightly broader than that of the Alexander Bay dataset.

Considering the Jozini dataset or bottom panel, M3 demonstrated superior predictive strength to M4. Additionally, the scores exhibited slightly broader and wider 95% PIs relative to the other two datasets. In summary, the MD validated the results deduced from the other performance indicators, and M3 emerged as the most effective forecasting model for all three datasets.

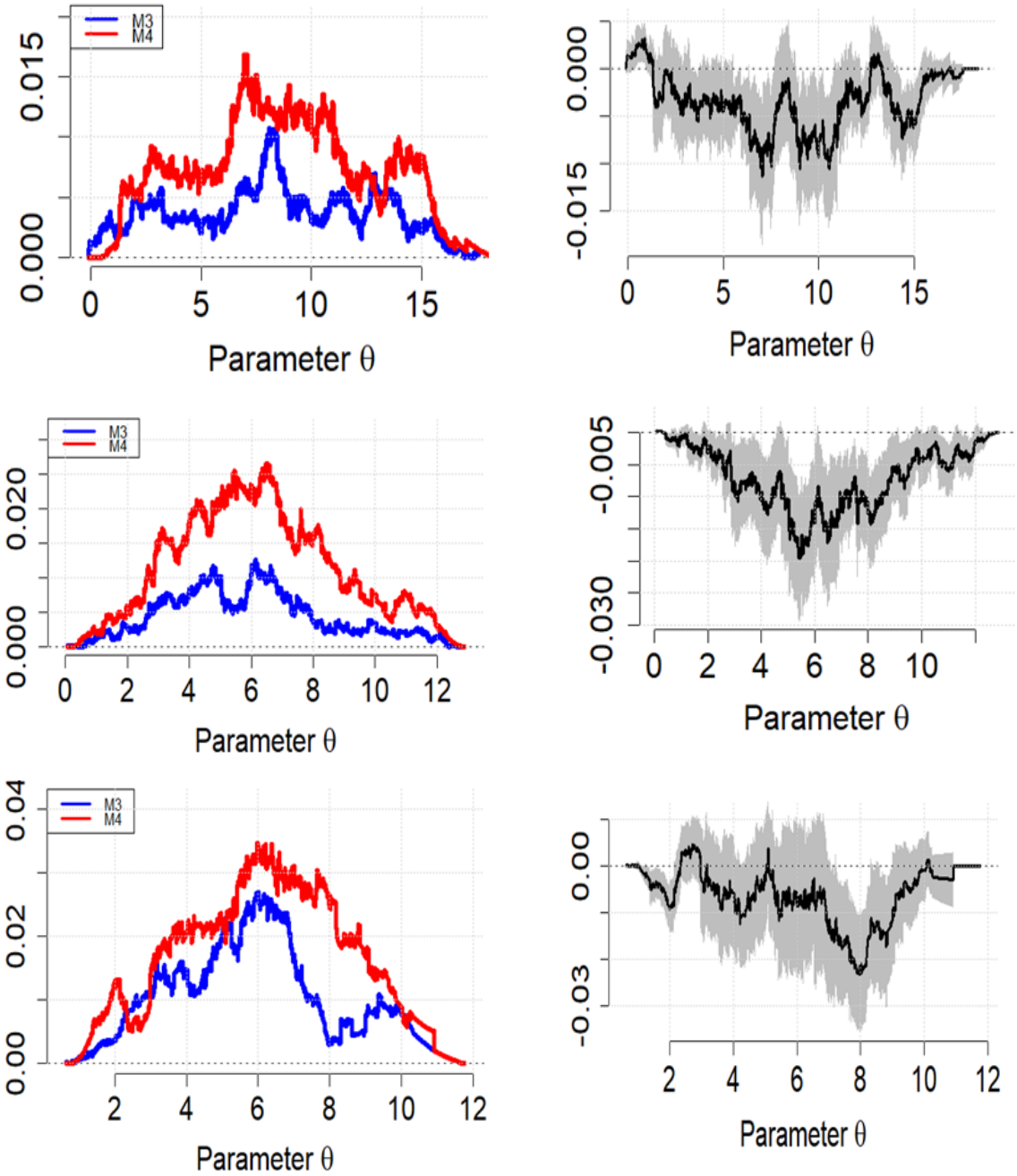


Figure 6.4. Murphy diagrams with 95% confidence intervals: M3 and M4 (upper panel, Alexander Bay), (centre panel, Humansdorp), and (bottom panel, Jozini). Shaded regions indicates 95% confidence intervals for the difference between the two functions.

Table 6.8. Effect of the lead times on model performance using the Alexander Bay dataset (at $L = 3$).

Method	Lead Time (Minutes)	Performance Indicator			Skilled Indicator		
		RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE
M3	10	0.5102	0.3787	12.2258	0.8705	0.8819	0.7045
	60	0.5898	0.4330	7.9098	0.8507	0.8657	0.8099
M4	10	0.7147	0.5099	8.2808	0.8186	0.8411	0.7999
	60	2.3845	1.8621	39.4635	0.3964	0.4225	0.0514
M5	10	3.9391	3.2069	41.3795			
	60	3.9503	3.2243	41.6032			

Keynote: time interval = 10 min.

Table 6.8 presents the predictive performance of the hybrid framework (M3) in comparison with M4 and M5 at various lead times using highly variable Alexander Bay wind speed data. Ideally, the study would like to assess the accuracy, generalisability, and robustness of the proposed approach as lead time increases from 10 min to 60 min. From Table 6.8, the predictive performance of the models diminishes with longer lead times, evidenced by higher error values in MAE and RMSE. Still, M3 demonstrates clear superiority and dominance over the other models, even when the lead time is increased to one hour.

6.3 Conclusions

The Chapter introduced a hybrid framework for wind speed prediction that combines MODWT filters, DEs, and GRUs, referred to as wavelet-MODWT-GRUs. To assess the efficacy of the proposed hybrid framework, 10-minute averaged high-resolution wind speed data sourced from the three WASA stations, viz., Alexander Bay, Humansdorp, and Jozini, were used. The chapter first compared the performance of the wavelet-MODWT-GRUs using LA8, DB4, and MB8 filters at decomposition levels $L = 3, L = 4$, and $L = 5$. Subsequently, the efficacy of the best-performing wavelet-MODWT-GRU at all three stations is evaluated relative to the GRU and the baseline Naïve model. According to RMSE, MAE, MAPE, and R^2, the proposed hybrid frameworks demonstrated superior predictive power at the decomposition level $L = 3$, optimised via DE, for all three datasets and wavelet filters. Comparison based on wavelet filter functions showed that wavelet-MODWT-GRU (MB) (at $L = 3$) framework achieved the most reliable and unbiased wind speed predictions for the three datasets. The probabilistic predictions scores (i.e., DS and PIT) showed that the same hybrid wavelet-MODWT-GRU (MB) framework delivered the most reliable and better-

calibrated results relative to the GRU and the Naïve for all three datasets. These results were further affirmed by the MD analysis. Overall, the DE approach provided an efficient, reproducible, and simpler technique to identifying the best decomposition level (at $L = 3$). As a result, the deconstructed signals using MODWT filters were (to an extent) easy to train and predict using the GRU model. Overall, the predictive performance of the hybrid framework was influenced by site location, filter function employed, level of decomposition, and the prediction horizon. The use of MODWT filters facilitated the effective and efficient extraction of noise and exposed significant data trends and patterns, thus enhancing the forecasting accuracy of the models. As a result, GRU models accurately and reliably predict wind speed data. As lead times increased, error increased and predictive accuracy diminished. However, the proposed hybrid framework continued to outperform all other models. The proposed framework will enable utility managers to incorporate large volumes of wind power into electric systems, enhance their knowledge of MODWT-based hybrid frameworks. However, it should be noted that this research evaluated only three wavelet filter functions using smaller South African wind speed datasets. Hence, future research work could apply other filters, such as Morlet or Meyer on larger and more variable data from different terrain in and out of South Africa.

6.4 Contributions

This chapter contributed to the wind forecasting literature in the following ways: Wavelets were efficiently and successfully used to identify structural discontinuities, thus improving the precision and reliability of the wind speed prediction framework. The Chapter successfully implemented the MODWT approach to deconstruct the original wind speed signal into high and low frequency subsignals exhibiting lower complexity compared to the original series. Different from the standard DB4 and LA8, the study further exploited (effectively and successfully) MB filters that are characterised by great frequency localisation, narrow bandwidths that minimises spectral leakage, while effectively isolating and filtering of noise outside a specific frequency band. Different from the computationally expensive GA, the DE approach, which is able to solve complex optimisation problems and optimise continuous functions that are both non-differentiable and nonlinear, with high convergence speed, was effectively and successfully applied to determine the best decomposition level for the selected wavelet filter. The nonlinear GRUs have the ability to capture the

fluctuations in variance over time that wind data usually present in the forecasting arena. Consequently, the simpler, robust, and efficient GRU was successfully employed to accurately predict each of the deconstructed subsignals. Most studies in the review literature (see Chapter 2) focus on short-term wind prediction. They neglect medium-to-long-term predictions, which are pivotal for wind turbine maintenance and wind farm construction. Accordingly the study assessed probabilistic prediction indicators in the medium-to-long-term forecast horizon using PL, DS, and PIT. Chapter 7 extends the methodological innovations in earlier chapters (e.g., feature engineering, support vector methods, feature decomposition, and boosting) to predict unplanned outages, thus demonstrating the applicability and generalisability of the typical Wavelet-ML approaches in power grid management.

This page is intentionally blank

Chapter 7

A Robust Wavelet Machine Learning Framework for Short-Term Forecasting of Unplanned Power Outages

7.1 Introduction

Owing to the persistent and ongoing power disequilibrium, national utility company (i.e. Eskom), which supplies more than 90% of South Africa's total electricity output, has been enforcing nation-wide load-shedding or manual load reduction (a controlled and enforced reduction in electricity supply) since late 2007 to prevent total national grid failure [227]. The recurring power outages stemming from constrained power generation in country has discouraged FDI inflows and raised production costs, thereby intensifying socio-economic challenges [6,228]. To overcome the persistent structural lack of adequate electricity supply, the authors [22,228] recommended swiftly switching to renewables as the country has abundant solar and wind energy resources [22,229,230,231]. However, coal (a non-renewable resource) remains the primary energy source in South Africa [16,17,232]. A total of 15 over-aged and over-used strained coal plants generated more than 85% of the electricity produced in 2021 (185 459 GWh), while renewable energies generated only 6% (or 215 337 GWh of the total electricity) [16,17]. As a result, South Africa (the largest energy consumer on the continent) emits the most GHGs in Africa [18]. In [233], authors pointed to the lack of academic research (which ultimately compromises informed decision-making processes and effective policy development) in the energy space as one of the main contributing factors to the ongoing energy crisis in South Africa. While research in renewable energy (especially solar energy) has gained traction recently in the country, there has been very little work done on power grid management and wind energy [22] (also see [234,235]). In [9], authors also indicated that the lack of studies on how large amounts of wind power can be integrated into South Africa's electric grid hinders wind energy growth since investors and decision-makers cannot quantify how much wind power South Africa can produce [230]. Besides reducing carbon emissions, the

advantages of wind power are its low maintenance costs and abundance (available day and night) [231].

Several variables affect the reliability and stability of power grid systems. These include intrinsic factors such as equipment life span, equipment defects, and internal maintenance; external factors such as weather fluctuations; and human error factors such as vandalism [236]. In ref [19], authors indicated that maintenance delays lead to increased unplanned capacity loss factor (UCLF) (i.e., unplanned outages) leading to increased load-shedding stages [237]. Furthermore, [234,236,238] revealed that businesses, the general public, and utility companies often lacked accurate power outage predictions in advance, making it difficult to plan ahead for the economic and maintenance impacts. As a result of the complex and limited predictability of power outages, hybrids are available in the literature, albeit to a lesser degree. In a pursuit of a well-balanced hybrid model, considering complexity, efficiency, and accuracy, this chapter combines WT, LASSO, RVM, AdaBoostRT, and RF in short-term unplanned outage forecasting. LASSO regression handles multicollinearity through variable selection to enhanced model interpretability. Meanwhile, RVM counts for nonlinearity and bias based on the original dataset, and WT deconstruct RVM residuals into more predictable subsignals. An AdaBoostRT make use of RVM residual data to produce residual predictions. The RF algorithm is used to both select the best top 10 variables per season and as a meta-model that combines RVM, AdaBoostRT, RF, and residual forecasts with speed and accuracy (minimal error variance). Hence, this chapter proposed an advance and highly accurate RVM-WT-AdaBoostRT-RF hybrid framework. The study evaluated the efficacy of the proposed framework against the benchmark Naïve model, VAR, RF, AdaBoostRT, and RVM using hourly power grid data sourced from the Eskom data portal.

7.2 Empirical Results

7.2.1 Data Description

This chapter evaluates the feasibility of developing a forecasting approach for power outages using hourly measurements of the power grid data downloaded from the Eskom website (see Chapter 1 for the link). The data spans from 1 March 2021 to 30 April 2022, as presented in Figure 7.1. The final dataset incorporated 43 variables as outlined in Table 7.1.

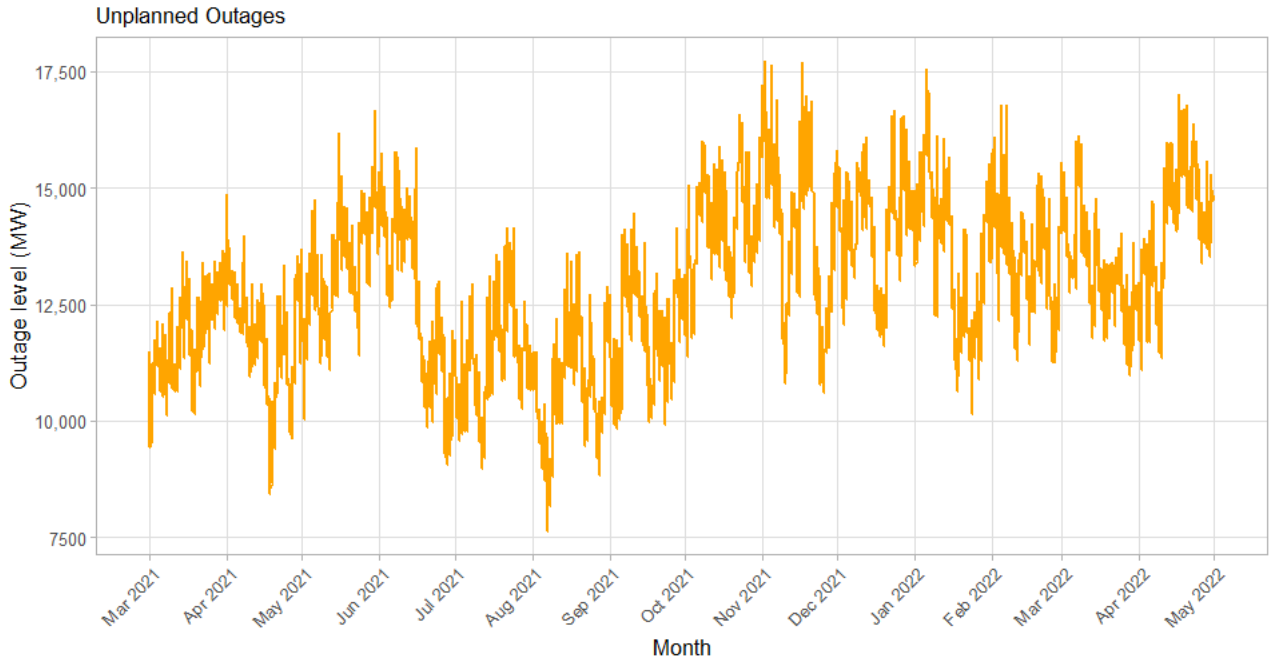


Figure 7.1. Hourly unplanned outage levels plot for the period 1 March 2021 to 30 April 2022.

In Figure 7.1, the unplanned outages (in MW) (i.e., $\boldsymbol{y}_{TUCLF.OCLF}$) are used as a dependent variable, while other variables are considered to be independent (see Table 7.1). The data details presented in Table 7.1 are supplemented by a comprehensive glossary and individual distributions in Appendix A. These help us understand the physical or operational meaning of the variables and their significance in power outage forecasting.

Table 7.1. Power grid data description.

Variable	Keynote
$\boldsymbol{x; y}$	Independent Variable; Dependent Variable
$\boldsymbol{x}_{ORFL}; \boldsymbol{x}_{RF}; \boldsymbol{x}_{RSA.CF}; \boldsymbol{x}_{DG}; \boldsymbol{x}_{IE}; \boldsymbol{x}_{RD}; \boldsymbol{x}_{RSA.CD}.$	ORFL = Original Residual Forecast Before Lockdown; RF = Residual Forecast; RSA.CF = Republic Of South Africa (RSA) Contracted Forecast; DG = Dispatchable Generation; IE = International Exports; RD = Residual Demand; RSA.CD = RSA Contracted Demand.
$\boldsymbol{x}_{IM}; \boldsymbol{x}_{TG}; \boldsymbol{x}_{NG}; \boldsymbol{x}_{EGG}; \boldsymbol{x}_{E.OCGT.G}; \boldsymbol{x}_{HWG}; \boldsymbol{x}_{ILSU}; \boldsymbol{x}_{MLR}; \boldsymbol{x}_{IOS}; \boldsymbol{x}_{D.IPP.OCGT}; \boldsymbol{x}_{E.GSCO} \boldsymbol{x}_{E.OCGT.SCO}; \boldsymbol{x}_{PWSCO.P}; \boldsymbol{x}_{PS}; \boldsymbol{x}_{IEC}.$	IM = International Imports; TG = Thermal Generation; NG = Nuclear Generation; EGG = Eskom Gas Generation; E.OCGT.G = Eskom Open Cycle Gas Turbine Generation; HWG = Hydro Water Generation; PWG = Pumped Water Generation; ILSU = Interruptible Load Shed Usage; MLR = Manual Load Reduction; IOS = Interruption of Supply Excl ILS and MLR; D.IPP.OCGT = Dispatchable Independent Power Producers Eskom Open Cycle Gas Turbine; E.GSCO = Eskom Gas Synchronous Condenser Operation; E.OCGT.SCO = Eskom Open Cycle Gas Turbine Synchronous Condenser Operation; PWSCO.P = Pumped Water Synchronous Condenser Operation Pumping; PS = Pump Storage; IEC= Installed Eskom Capacity.

$\boldsymbol{x}_{DGUH}$; $\boldsymbol{x}_{PGUH}$; $\boldsymbol{x}_{IGUH}$.	DGUH = Drakensberg Generation Unit Hours; PGUH = Palmiet Generation Unit Hours; IGUH = Ingula Generation Unit Hours.
$\boldsymbol{x}_{WIND}$; $\boldsymbol{x}_{PV}$; $\boldsymbol{x}_{CSP}$; $\boldsymbol{x}_{ORE}$; $\boldsymbol{x}_{TRE}$; $\boldsymbol{x}_{WIC}$; $\boldsymbol{x}_{PVIC}$; $\boldsymbol{x}_{CSPIC}$; $\boldsymbol{x}_{OREIC}$; $\boldsymbol{x}_{TREIC}$.	PV = Photovoltaic; CSP = Concentrated Solar Power; ORE = Other Renewable; TRE = Total Renewable; WIC = Wind Installed Capacity; PVIC = PV Installed Capacity; CSPIC = CSP Installed Capacity; OREIC = Other Renewable Installed Capacity; TREIC = Total Renewable Installed Capacity.
$\boldsymbol{x}_{TPCLF}$; $\boldsymbol{x}_{TUCLF}$; $\boldsymbol{x}_{TOCLF}$; $\boldsymbol{x}_{E.GSCO}$; $\boldsymbol{x}_{lag\ 1}$; $\boldsymbol{x}_{lag\ 2}$; $\boldsymbol{x}_{lag\ 24}$; $\boldsymbol{x}_{NCS}$.	TPCLF = Total Planned Capability Loss Factor of Eskom plant; TUCLF = Total Unplanned Capability Loss Factor of Eskom plant; TOCLF = Total Other Capability Loss Factor of Eskom plant; lag 1 = TUCLF. OCLF 1 h ago (i.e., to capture immediate fluctuations); lag 2 = TUCLF. OCLF 2 h ago (i.e., to capture short-term trends); lag 24 = TUCLF. OCLF 24 h ago (i.e., to capture daily patterns); NCS = Non-comm sentout (NCS).
$\boldsymbol{y}_{TUCLF.OCLF.}$	TUCLF. OCLF = Total unplanned power outage including TOCLF

Data Partition

The data were divided into four seasons with different variability, viz., autumn, winter, spring, and summer. An effective predictive framework must be tested and validated across all conditions throughout the year. In this study, the first two months of each season (around 1464 hours) were selected to represent that specific season. For each season, the data are partitioned into a training set (80%) and a testing set (20%) (see Table 7.2). The study preserved the results for the Autumn 2022 dataset to evaluate models' applicability and robustness over the years. By assessing the four distinct seasons within a single year, we can examine patterns that impact the model and ensure its robustness across different seasonal conditions.

Table 7.2. Sample breakdown for model training and testing.

Dataset	Date	Sample	Training (80%)	Test (20%)
Autumn	1 March–30 April 2021	1464	1176	288
Winter	1 June–31 July 2021	1464	1176	288
Spring	1 September–31 October 2021	1464	1176	288
Summer	1 December 2021–31 January 2022	1488	1195	293
Autumn 2022	1 March 2022–30 April 2022	1464	1176	288

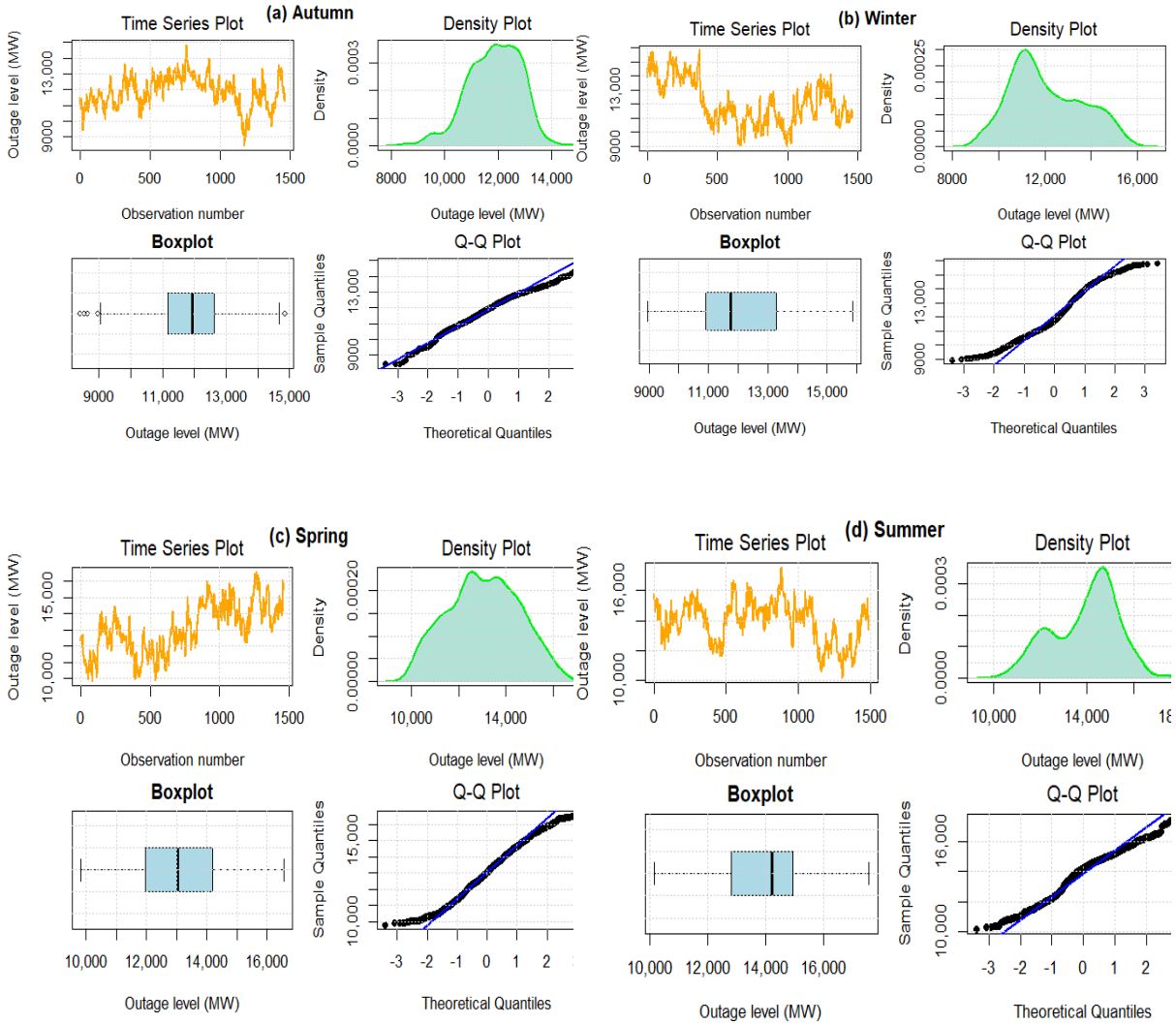
7.2.2 Summary Statistics

Table 7.3 presents summary statistics for the five datasets under study. The highest power outage levels were recorded in the summer (17,558 MW), whilst the lowest were observed in the autumn (8410 MW). On average, the summer power outages (13,928 MW) were significantly higher than in any other season. In contrast, winter had the highest variance (1562.242 MW) relative to any other season, indicating further seasonal differences. Power outage levels varied by season, and this variation was

affirmed by the Kruskal–Wallis test at the 5% significance level, which resulted in a p-value <0.05, confirming the presence of seasonality effects (also see Figure 7.2). All datasets have a kurtosis below 3, hence they are platykurtic. Deviance from the normal distribution was noted in autumn, winter, and summer.

Table 7.3. Summary statistics for the datasets (in MW).

Dataset	Min	Q1	Median	Mean	Q3	Max	Std.Dev	Kurtosis	Skewness
Autumn	8410	11,184	11,931	11,863	12,619	14,867	986.3464	0.0874	−0.4222
Winter	8957	10,914	11,754	12,076	13,303	15,862	1562.242	−0.7764	0.3735
Spring	9819	11,966	13,044	13,055	14,193	16,573	1503.281	−0.7384	0.0026
Summer	10,144	12,823	14,219	13,928	14,924	17,558	1396.025	−0.5362	−0.3862
Autumn 2022	10,981	12,829	13,749	13,793	14,676	17,022	1245.443	−0.6943	0.1651



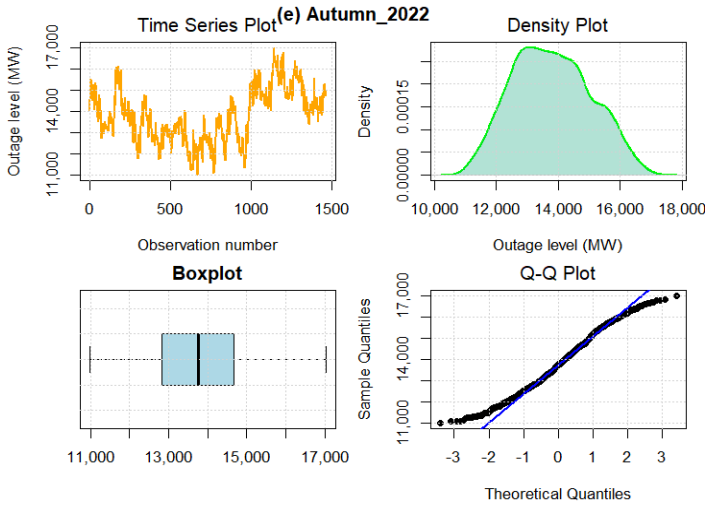


Figure 7.2. The time plot, density plot, boxplot, and Q-Q plot for power outage data for Autumn (top left panel), Winter (top right panel), Spring (middle left panel), Summer (middle right panel), and Autumn 2022 (bottom centre panel) datasets. Blue lines represent Q-Q lines and sky-blue boxes in indicate interquartile ranges.

7.2.3 Model Settings

Models were trained through cross-validation, grid search, and heuristic approaches. Table 7.4 presents the optimal parameter intervals and the respective R libraries used. The average implementation time per dataset was approximately 4 to 5 minutes.

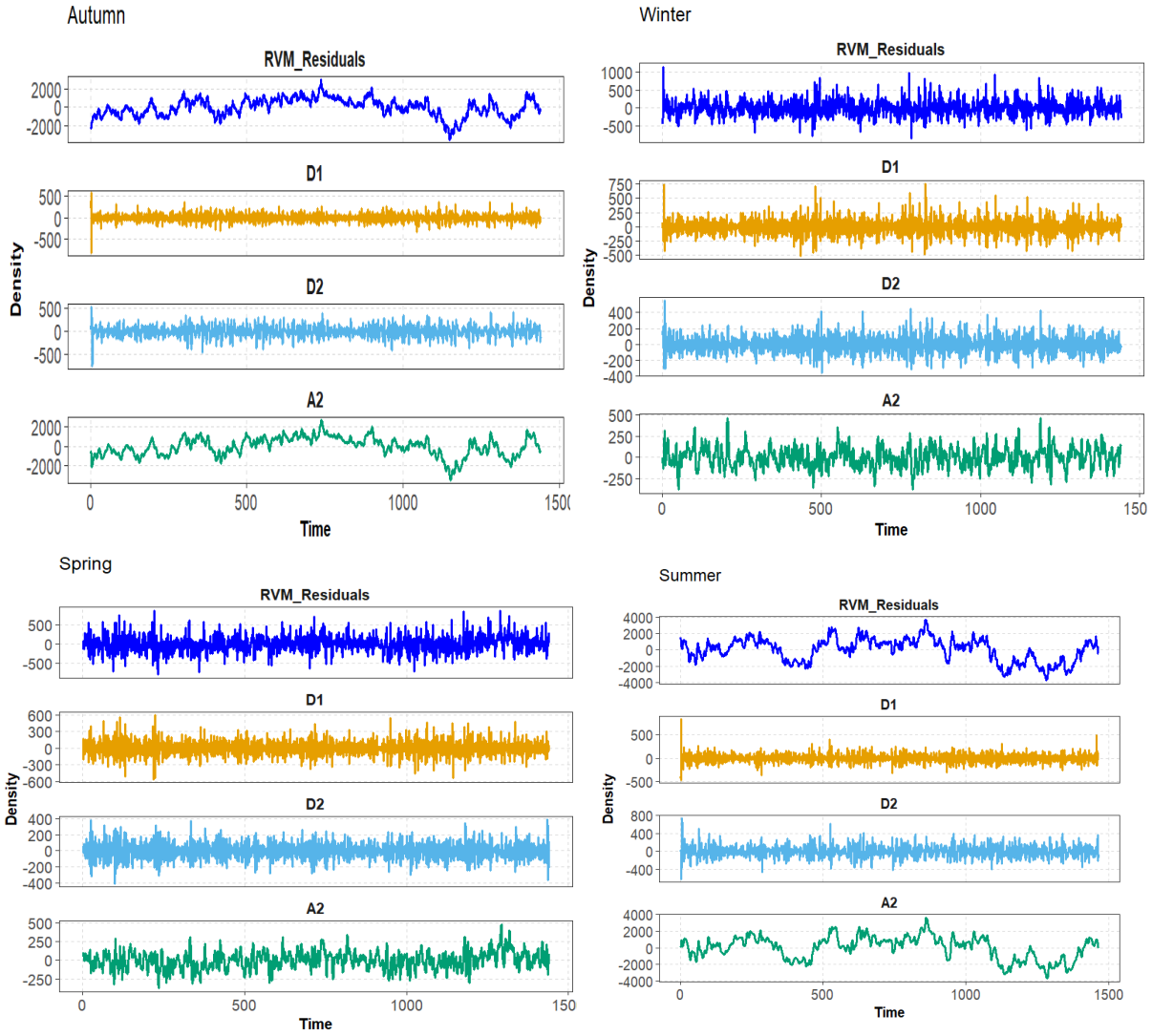
Table 7.4. Model parameters settings

Model	Libraries	Method	Parameter	Optimal range
LASSO	glmnet	Variable Selection	lambda family nlambda	$0-2$ "guassian" $100-500$
RF	Caret, ranger, randomForest	Bagging ensemble	mtry ntree nodesize	$1-10$ $100-1000$ $1-15$
RVM	kernlab (rvm)	Bayesian inference	kernel sigma degree	("anovadot","rbfdot") $0-2$ $1-2$
AdaBoostRT	ReBoost (AdaBoostRT)	Boosting ensemble	thr power t_final	$0.001-0.3$ $0-2$ $30-500$
WT	Waveslim (modwt)	Signal decomposition (noise reduction)	wf n.levels boundary	'db4' 2 'periodic'
VAR	Vars (var)	Autoregression	lag order p	$1-3$
Hybrid	-	Stacked		

7.2.4 Discussion of the Results

Wavelet Analysis

Evaluation using RMSE and R^2 (at decomposition level $L = 2$ to 3), showed that the DB4 filter outperformed LA8 filter with the best results at $L = 2$. Hence, the properties of the deconstructed RVM residuals using DB4 filter are shown in Figure 7.3. The models' predictive strength was high at lower decomposition levels.



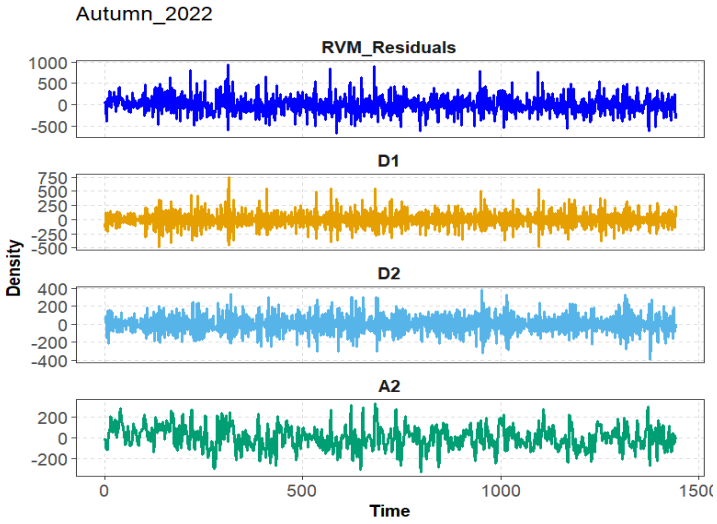


Figure 7.3. Level 2 DB4 wavelet decomposition of the RVM residuals for Autumn (top left panel), Winter (top right panel), Spring (middle left panel), Summer (middle right panel), and Autumn 2022 (bottom centre panel) datasets.

Comparative Analysis

In Table 7.5, various performance metrics (point and probabilistic) and statistical tests used to compare the stacked hybrid (i.e., RVM-WT-AdaBoostRT-RF) against RF, RVM, AdaBoostRT, VAR, and benchmark Naive model using the five datasets are presented. The Autumn 2022 dataset was preserved to assess the seasonal robustness of the developed approach. The best models are bolded.

Table 7.5. Performance indicators for the developed models.

	Model	Autumn	Winter	Spring	Summer	Autumn 2022
Point forecasts evaluation						
RMSE (MW)	**Hybrid**	**262.6653**	**264.5506**	**394.6098**	**379.0801**	**260.6709**
	RF	403.1206	326.5305	740.894	678.7496	383.0104
	RVM	414.4714	302.2173	549.6189	608.635	301.7799
	AdaBoostRT	390.2795	318.0112	714.8113	637.2507	359.4349
	VAR	2491.183	942.7002	1100.073	1201.092	812.6005
	Naive	1214.92	1027.169	1075.973	3252.646	993.0945
MAE (MW)	**Hybrid**	**201.1949**	**198.5543**	**288.1561**	**273.6507**	**190.9103**
	RF	287.4192	253.7929	538.0478	519.6454	289.1329
	RVM	298.4295	232.7954	385.2204	543.8472	227.4744

	AdaBoostRT	252.9515	239.721	500.6655	469.6258	264.6619
	VAR	2294.874	800.4576	893.751	990.4461	695.3962
	Naive	986.413	908.9877	902.3139	3013.444	781.4761
MAPE (%)	**Hybrid**	**1.81973**	**1.6408**	**1.9897**	**2.2190**	**1.2744**
	RF	2.5850	2.1058	3.7770	4.0968	1.9376
	RVM	2.9030	1.9925	2.6684	4.6424	1.5185
	AdaBoostRT	2.2760	1.9778	3.4994	3.7291	1.7754
	VAR	25.1661	6.7485	6.3391	7.9346	4.5550
	Naive	8.2233	7.4672	6.2169	19.2540	5.4756
Residual analysis						
Standard deviation (MW)	**Hybrid**	**257.5268**	**257.9521**	**376.4643**	379.1566	**261.116**
	RF	368.4571	322.784	584.5953	474.7601	326.8649
	RVM	369.7274	283.4306	445.0868	**279.8739**	301.1337
	AdaBoostRT	374.4322	318.5555	606.9506	533.957	325.5703
	VAR	970.9904	931.584	1080.136	1188.798	722.2145
	Naive	1056.375	1007.025	1070.423	1226.374	767.169
Skewness/Error direction	Hybrid	Underestimate	Underestimate	Underestimate	Overestimate	Underestimate
	RF	Overestimate	Underestimate	Underestimate	Overestimate	Underestimate
	RVM	Underestimate	Underestimate	Underestimate	Underestimate	Underestimate
	AdaBoostRT	Overestimate	Overestimate	Underestimate	Overestimate	Underestimate
	VAR	Underestimate	Underestimate	Underestimate	Underestimate	Underestimate
	Naive	Overestimate	Underestimate	Underestimate	Underestimate	Underestimate
Bias test (Conclusion)						
MZ *	Hybrid	Biased	Biased	Biased	Biased	Unbiased
	RF	Biased	Biased	Biased	Biased	Biased
	RVM	Biased	Biased	Biased	Biased	Biased
	AdaBoostRT	Biased	Biased	Biased	Biased	Biased
	VAR	Biased	Biased	Biased	Biased	Biased
	Naive	Biased	Biased	Biased	Biased	Biased
Prediction intervals evaluation						
95% PINAW	**Hybrid**	**21.2277**	**24.2517**	**30.2351**	30.7052	**30.0080**

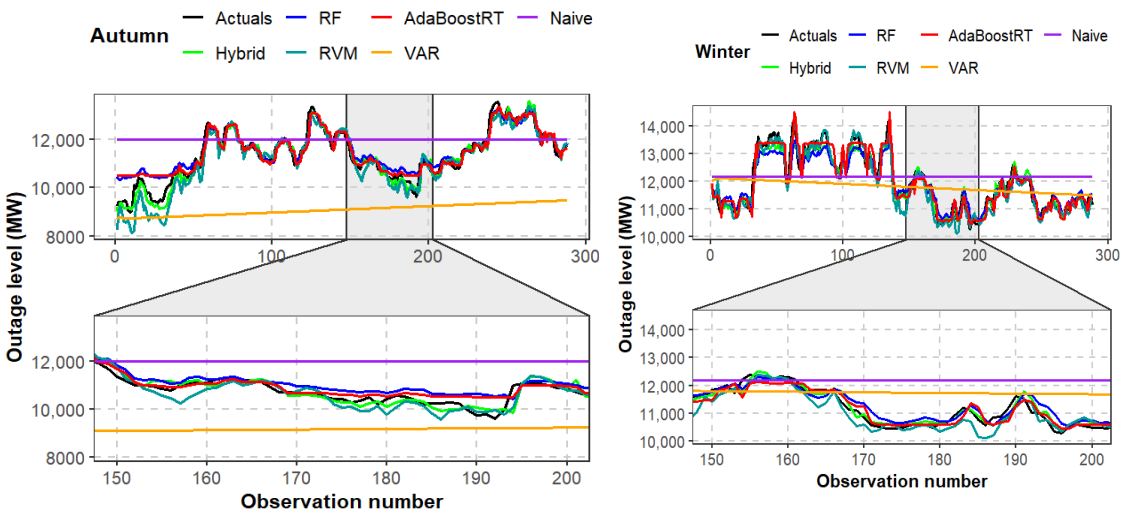
	RF	25.8908	27.8732	40.7178	32.6159	35.2706
	RVM	25.3539	27.8480	32.4444	**17.4738**	37.7469
	AdaBoostRT	27.9150	31.2775	43.6040	38.3266	36.9853
	VAR	68.7904	70.4972	79.6423	70.5827	57.4267
	Naive	86.1431	84.3254	92.8462	82.2576	79.2907

Predictive accuracy evaluation: Hybrid vs. individual models.

DM **	RF	H_0 Rejected	H_0 Rejected	H_0 Rejected	H_0 Rejected	H_0 Rejected
	RVM	H_0 Rejected	H_0 Rejected	H_0 Rejected	H_0 Rejected	H_0 Rejected
	AdaBoostRT	H_0 Rejected	H_0 Rejected	H_0 Rejected	H_0 Rejected	H_0 Rejected
	VAR	H_0 Rejected	H_0 Rejected	H_0 Rejected	H_0 Rejected	H_0 Rejected
	Naive	H_0 Rejected	H_0 Rejected	H_0 Rejected	H_0 Rejected	H_0 Rejected

Bold = Best model.

Based on RMSE, MAE, and MAPE, the comparative analysis showed that the proposed stacked hybrid model outperformed all other models, followed by RVM, AdaBoostRT, RF, VAR, and Naive. Except for the autumn dataset, RVM consistently exhibited the highest predictive power relative to other individual models based on the aforementioned metrics. All evaluated models, including the hybrid, exhibited sensitivity to seasonal fluctuations as shown in Figures 7.4 and 7.5.



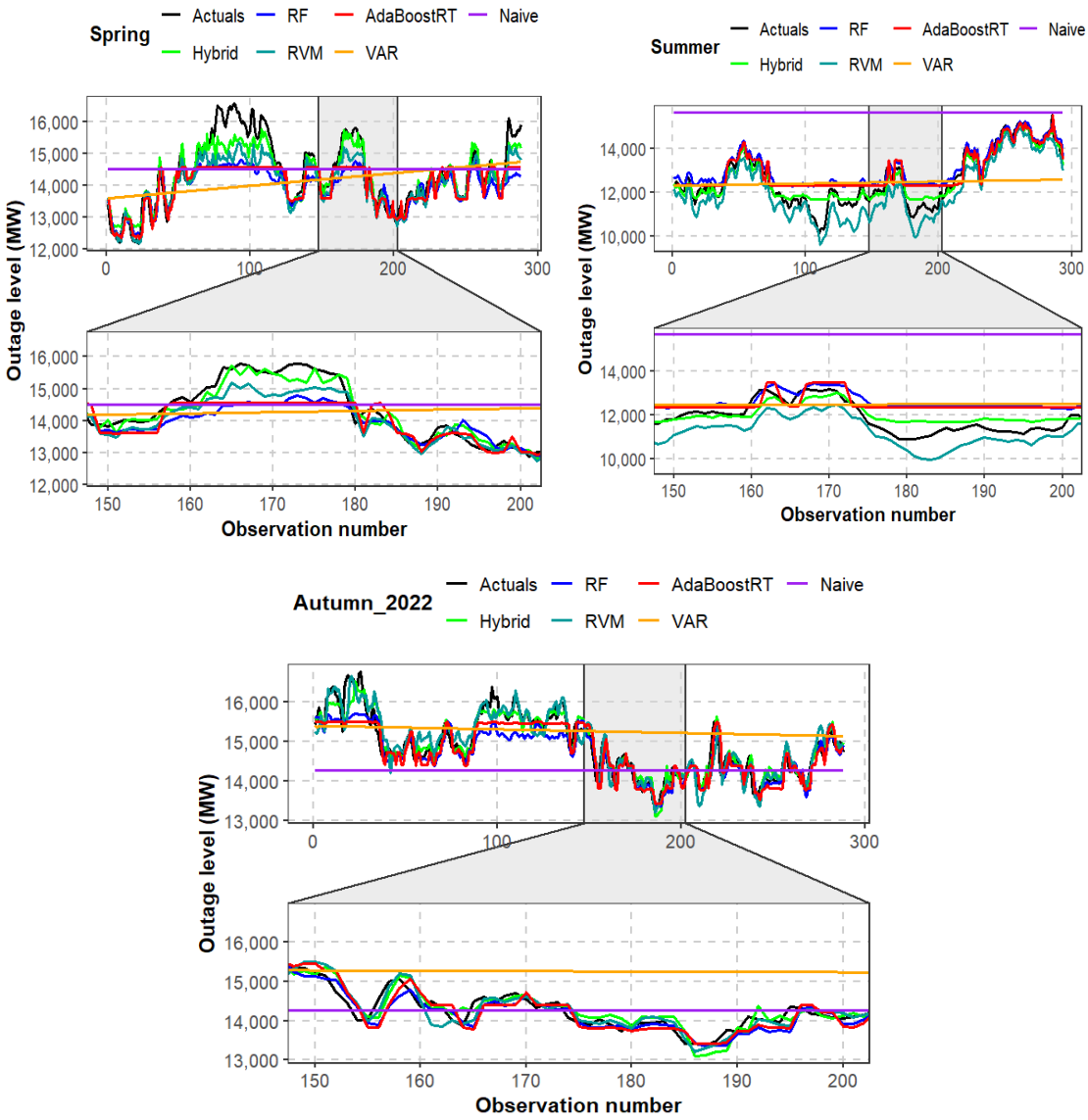
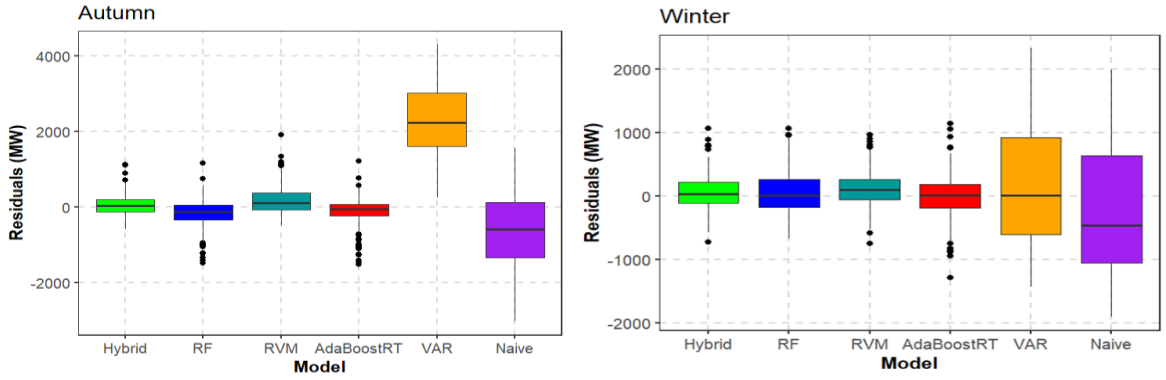


Figure 7.4. Comparison of models' predictions and actual power outage levels for Autumn (top left panel), Winter (top right panel), Spring (middle left panel), Summer (middle right panel), and Autumn 2022 (bottom centre panel) datasets.



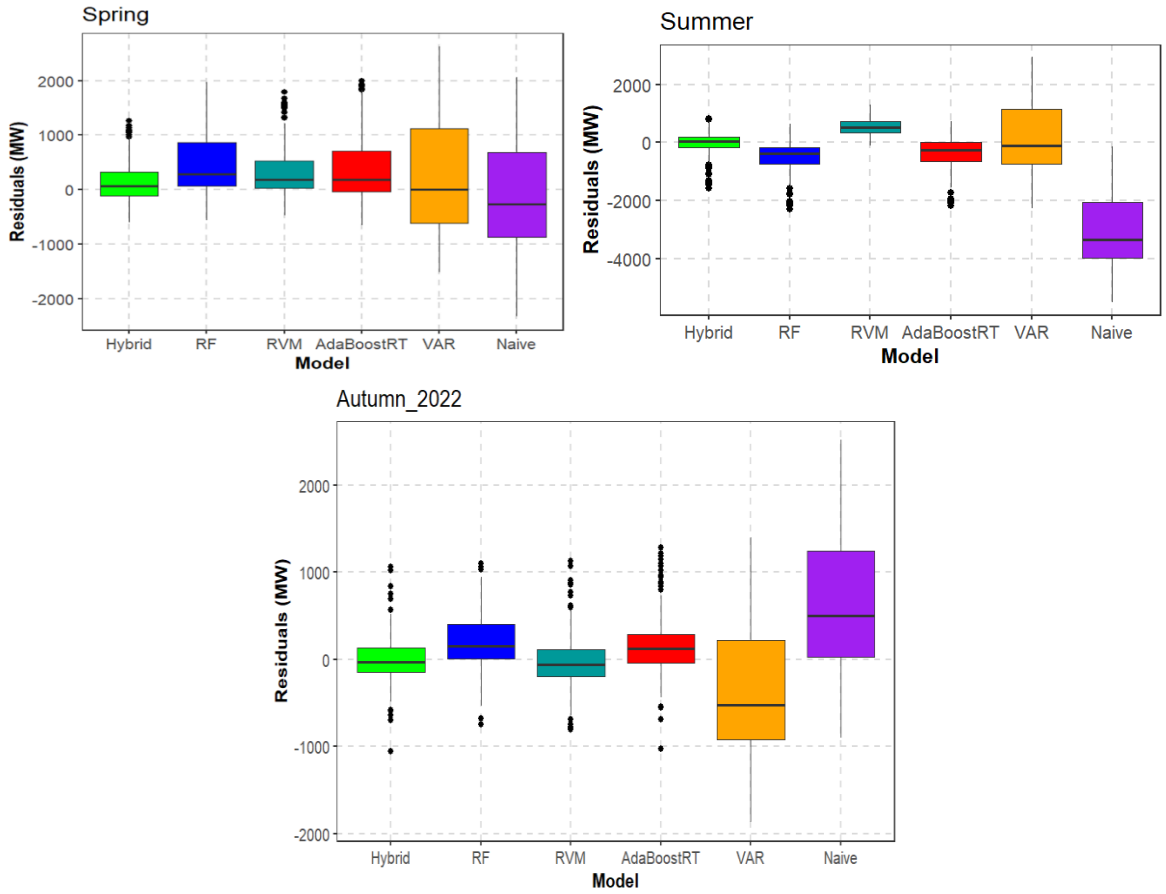


Figure 7.5. Box plot comparison of models' residuals for Autumn (top left panel), Winter (top right panel), Spring (middle left panel), Summer (middle right panel), and Autumn 2022 (bottom centre panel) datasets.

With the exception of the summer data, the hybrid model underestimated all power outage datasets, as its residuals were positively skewed. A similar trend was noted for RVM and VAR predictions for all seasons under examination. Apart from the spring dataset, the AdaBoostRT overestimated the four datasets. The Naive forecasting model overestimated the autumn data and underpredicted all remaining datasets. Excluding the summer dataset, the stacked approach produced the least standard deviation values relative to the individual models (see Figure 7.4). The most accurate performance of the stacked hybrid model was observed in the autumn season compared to other seasonal datasets. Except for the autumn dataset, RVM produced the smallest residual spread based on the smallest standard deviation values among the individual models. According to the MZ test, all models were found to be biased in each of the four datasets considered.

At the 95% confidence coverage, the hybrid yielded the lowest PINAW (i.e., better calibration) for autumn, winter, and spring datasets, whereas the RVM outperformed

the stacked hybrid for the summer dataset. In general, the hybrid PIs were significantly narrower for the autumn dataset and relatively broader for the spring dataset. The DM test was rejected at 5% level of significance implying enhanced predictive capability for the stacked hybrid approach relative to other models. Based on Autumn 2022, the comparative study assessing seasonal model adaptability and robustness demonstrated the superiority of the stacked hybrid approach (followed by RVM) in terms of unbiasedness and accuracy relative to other approaches across all performance assessment metrics and DM test. The stacked hybrid method showed stability and robustness to seasonal effects. Overall, the stacked model effectively accounts for higher peaks and variations inherent in power outage data better than all models (see Figure 7.4). Therefore, the stacked approach delivers better short-term point and probabilistic predictions characterised by less uncertainty and robust seasonal adaptability (see Figure 7.4).

Accuracy–Complexity Trade-Off

With the stacked hybrid approach, the accuracy of predictions for each model goes up by at least 40%, which is pivotal for making well-informed decisions about resource allocation, power grid risk management, and cost minimisation in the energy space. The computational cost (as shown in Table 7.6) can be accounted for by the utilisation advanced computer hardware. To some degree, the processing time (60s) restricts the model's applicability in real-time power grid control. Nevertheless, the proposed stacked approach remains pivotal for short-term power grid management planning.

Table 7.6. Trade-off between accuracy and complexity (excluding Autumn 2022 dataset).

Model	Computational Time Intervals (s)	Average Computational Time (s)	Hybrid vs. Single Model Time Difference (s)	−%ΔRMSE
RF	20–30	25	30	61
RVM	30–40	35	20	43
AdaBoostRT	30–40	35	20	55
VAR	5–10	7.5	47.5	375
Naive	5–10	7.5	47.5	395
Hybrid	50–60	55	-	-

Ablation Study

As shown in Table 7.7, the ablation analysis based on the summer dataset assesses the contribution of each component to the full stacked approach. The findings showed that each model included in the full stacked model enhanced predictive performance, affirming the synergetic architecture of the proposed framework. The full stacked hybrid model achieved the best and most balanced performance compared to the partial hybrids.

Table 7.7. Ablation study using the summer dataset.

Model	Blender	RMSE/MW	MAPE/%	PICP/%	PINAW/%	MAD/MW	−%ΔRMSE
RVM+AdaBoosRT	RF	458.6394	2.6951	95.5631	31.4419	376.9945	21
RVM+AdaBoosRT+ $\hat{\xi}_f$	RF	430.3829	2.5669	95.2218	31.7904	352.7375	14
RF+AdaBoosRT	RF	466.5318	2.6454	95.9044	33.4339	304.2609	23
RF+AdaBoosRT+ $\hat{\xi}_f$	RF	436.3922	2.4485	94.8806	31.6265	**287.6163**	15
RVM+RF+ $\hat{\xi}_f$	RF	399.4977	2.7087	95.5631	**25.5820**	440.9243	5
RVM+AdaBoosRT+RF+ $\hat{\xi}_f$	Average	3337.881	35.689	95.2218	26.8902	5001.131	781
Full stacked (Hybrid)	**RF**	**379.0801**	**2.2190**	**95.9044**	30.7052	292.8671	

Keynote: $\hat{\xi}_f$ = residual forecast; Bold = Best model.

7.4 Conclusions

The current chapter introduced a stacked hybrid learning model, RVM-WT-AdaBoostRT-RF to accurately and reliably predict unplanned power outages in South Africa using power grid data provided by Eskom. The study employed variables comprising electricity demand, supply, renewables, storage, and outages. The performance of the proposed hybrid approach was examined based on RMSE, MAPE, MAE, residual analysis, 95% PINAW, the MZ test, and the DM test against the benchmark Naive, VAR, RF, RVM, and AdaBoostRT. The overall findings demonstrated that the proposed stacked hybrid outcompeted all other models across all datasets based on the RMSE, MAPE, MAE, residual analysis, 95% PINAW, and the DM statistic. Furthermore, proposed stacked framework delivered high predictive accuracy and most reliable PI estimates with less uncertainty alongside seasonal robustness. Additionally, the hybrid delivered better results on unscaled data than on scaled data. However, the proposed strategy showed varying results influenced by the season of the year. Moreover, the hybrid framework consistently underestimated outages in the spring dataset, this can be attributed to the fact that wind power, a key

predictor, is highly variable and irregular during this season in South Africa. In essence, the primary shortcoming of this study is that it does not fully account for weather variables including rainfall, storms, and other weather events plus the chapter on dwell on short-term predictive modelling. The season-based results will contribute to effective resource management as well as enhancing power grid management strategies tailored to the needs of each season. Thus, the finding from the study can be used by grid management teams, utility operators, and other energy infrastructure developers in South Africa.

7.5 Contributions

The inherent complex behaviour of the power grid requires the use of an appropriate modelling approach to effectively characterise the multi-faceted power grid characteristics, as relying on individual models frequently lead to inaccurate and unreliable forecasts. Consequently, the current chapter introduced a stacked hybrid framework which leverages the strength of various ML methods to deliver efficiency, accuracy, minimal bias, and robustness in generating point and interval forecasts that are vital for informed decision-making process. The initial analysis through VIF successfully detected the presence multicollinearity among the variables, facilitating the process of variable selection and regularisation. The LASSO and RF were successfully deployed to minimise high data dimensional complexity and enhance the feature engineering process. The study also leveraged the sparse and probabilistic Bayesian learning of RVM to capture complex data behaviour (such as nonlinearity, random fluctuations, etc.), while curtailing model overfitting. WTs (through MODWT) efficiently decomposed and smoothed residuals. In handling residuals, bias was minimised through a weak learner boosting approach inherent in the AdaBoostRT model. Finally, RF is also successfully used as a meta-model that combines RVM, AdaBoostRT, RF, and residual predictions with efficiency and accuracy. The dependent variable TUCLF.OCLF integrates OCLF and UCLF and better accounts for power outage behaviour across seasons as compared to the use of only UCLF. To a certain extent, proposed stacked approach successfully captured the seasonality effects, nonlinearity, random fluctuations, and nonstationarity patterns inherent in the power grid data.

This page is intentionally blank

Chapter 8

Conclusion and Future Research

8.1 Conclusions

Primarily, this research work aims to develop wavelet ML hybrids that simultaneously combine and leverage the strengths of data pre-processing, data optimisation, and data post-processing methods to efficiently, accurately, and reliably quantify high-resolution wind data over a short-to-long-term forecast horizon using varying datasets from different locations with varying meteorological characteristics. As such, the second chapter of this study presents an in-depth analysis alongside a synthesis of the shortcomings and strengths of statistical, ML, and hybrid methodologies within the context and framework of wind forecasting. The main issues identified in wind speed forecasting literature, which shape the focus of this research work, are as follows:

- high wind speed forecasting error accumulation (which forms part and the core of the fourth chapter);
- vanishing and explosive gradients due to transient and chaotic wind data (which are the focal point of the fifth chapter); and
- the lack of efficient and reproducible methods for selecting filters and wavelet decomposition levels (which is the core and centre of the sixth chapter).

All these issues if not treated appropriately, often compromise the accuracy, reliability, and robustness of wind forecasting models and associated forecasts thereby undermining investment in wind power. As a secondary objective, the seventh chapter seeks to test the generalisability of the wavelet-ML hybrids in short-term power outage forecasting using South African power grid data. Overall, the novelty and the main contributions of the current research work are summarised as follows (also see Table 8.1):

- Chapter 4 introduced the WT-ARIMA-XGBoost-SVR hybrid method for improving wind speed predictions, combining WT for noise reduction; ARIMA for linear modelling; XGBoost for high efficiency and accuracy; and SVR for efficient nonlinearity forecast combination. The proposed hybrid strategy efficiently reduced wind speed forecasting error accumulation caused by the use of linear models in reconciling nonlinear wavelet subseries forecasts.
- Chapter 5 outlined the WT-NNAR-LSTM-GBM hybrid model, which enhanced performance by leveraging WT to decompose and expose the underlying trends and patterns of high variant wind data, SampEn for complex feature detection, NNAR for deterministic predictions, LSTM for nonlinear component and vanishing gradients management, and GBM for nonlinear optimal forecasting reconciliation (including reduction of error accumulation). With this approach, gradients were effectively prevented from vanishing and exploding, while improving short-term wind speed forecasting accuracy. Also, the approach emphasised the classification and modelling of wavelet sub-signals based on their complex and deterministic characteristics which further helped manage the issue of gradient disappearance effectively.
- Chapter 6 presents a wavelet-MODWT-GRU approach for assessing the impact of wavelet filters and decomposition levels on wind prediction accuracy, utilising MODWT for denoising and GRU for capturing both linear and nonlinear components (including avoiding gradient disappearance). Primarily, the chapter provides a more efficient and reproducible approach to select the most appropriate wavelet filters and associated decomposition levels to improve wind speed forecasts (by reducing prediction error). Additionally, the framework demonstrated improved forecast accuracy as forecasting horizons increased.
- Across chapters 4-6, performance analysis was conducted across multiple datasets from different locations at different forecast horizons (short-to-long-term) using various probabilistic and deterministic metric error indicators, and statistical tests. The results show that the proposed wavelet-based hybrids out-competed conventional statistical, ML, and hybrid models.
- Underscoring the importance of the careful integration of WTs with ML methods, the wavelet-ML hybrids implemented in this thesis improved wind speed prediction accuracy at different forecasting horizons using high-resolution wind speed datasets from varying terrains with diverse climatic conditions (including different seasons) as detailed in Chapters 4–6. Given that wind variability directly impacts power grid stability, the proposed approaches

were then extended to short-term power outage forecasting, thereby showing great generalisability, flexibility, and adaptability in real-world applications (beyond wind forecasting), specifically in power grid management.

Table 8.1. Summary of Contributions

Area	Contribution	Significance
Theory	Alongside the developed taxonomy to classify hybrid methods, this research work clearly and extensively showed the ability of wavelets to extract noise, handle nonstationarity, reveal data patterns, and facilitate accuracy in wind speed forecasting.	Provided a solid foundation for the wavelet-based hybrid methods implementation and processing.
Methodology	The research work established and employed four (4) different wavelet-ML hybrid approaches, provided a thorough approach for the optimal selection of hyperparameters, wavelet decomposition levels, and assessed various wavelet families or filters for accurate and robust wind speed and power outage forecasting.	The methods applied are pivotal in that they enhanced model strength, accuracy, robustness, and generalisation in predicting highly complex wind data and power grid data.
Practical applications	This research work applied highly accurate wavelet-ML hybrid models to several real-world wind datasets (from different locations with varying data characteristics) and highly complex heterogeneous power grid data. Furthermore, the work also conducted comprehensive model performance evaluation using appropriate performance metrics alongside valid statistical tests.	The proposed approaches demonstrated potential for practical and real world applications renewable energy space and power grid management.
Published research works	Author has published a total of four (4) journal papers and have gained several citations from highly recognised journals.	Research increased visibility and contribution to the knowledge.

8.2 Reconciling Parsimony, Accuracy, and Adequacy

In model selection, there are three (3) key aspects that are recognised, namely; accuracy, adequacy, and parsimony. Accuracy is concerned with the model's ability to predict observations based on unseen data (i.e., test data). In contrast, adequacy looks at how well the model captures the underlying data patterns and characteristics, whilst parsimony advocates for a simpler model with minimal parameters (also see [239,240]).

Both aspects of parsimony and adequacy were not the primary focus of the proposed wavelet-ML hybrids, but were considered through controlled model design. This

includes minimising wavelet decomposition levels, using feature selection, applying structured hyperparameter tuning, and regularisation techniques. The accuracy of the aforementioned hybrids was evaluated using point evaluation metrics (e.g., MAE, MAPE, RMSE), whilst the forecast reliability (including calibration and sharpness) was evaluated using probabilistic evaluation metrics (e.g., PIW, PINAW, CRPS). Model evaluations were carried out on datasets spanning diverse terrains, climatic conditions, seasons, and forecasting horizons, against several benchmark models. Regardless of regime changes, the proposed hybrid frameworks consistently outperformed other models. Accordingly, structural adequacy[13] is reflected in the robustness of model performance across diverse datasets and operating conditions [241,242].

In line with the predictive framework[14] of Shmueli (2010) [239] and algorithmic modelling approach[15] of Breiman (2001) [240], this study prioritises predictive accuracy using an algorithmic modelling approach, namely; wavelet-ML hybrids. In complex tasks such as wind speed forecasting, the emphasis shifts from parsimony towards models with a high predictive capability [240,243]. As a result, adequacy and parsimony played a supportive or secondary role in the proposed predictive framework.

8.3 Future Research

Although the proposed hybrids demonstrated high accuracy, robustness, and generalisability (to some extent), there remain several limitations and room for improving model complexity, efficiency, generalisability and application to much more complex and larger datasets beyond the Southern African region.

- In Chapter 4, the proposed WT-ARIMA-XGBoost-SVR displayed some gaps when capturing wind trends and patterns and transient behaviour when the forecast horizon was increased from short-term to medium and the long-term forecast horizon was increased. In the future, it would be interesting to replace

[13] In structural adequacy, a model remains valid under counterfactuals and regime change [241,242].

[14] "In predictive modeling, the focus is on predictive accuracy or predictive power, which refer to the performance of $\hat{f}$ on new data. Measures of predictive power are typically out-of-sample metrics or their in-sample approximations, which depend on the type of required prediction." [239]

[15] "There are two cultures in the use of statistical modeling to reach conclusions from data. One assumes that the data are generated by a given stochastic data model. The other uses algorithmic models and treats the data mechanism as unknown. The statistical community has been committed to the almost exclusive use of data models." [239,240]

ARIMA with a robust and nonlinear framework such as RF to form WT-RF-XGBoost-SVR to remedy the deficiencies of the linear ARIMA model beyond short-term forecasting.

- In Chapter 5, by replacing SampEn with a simpler but robust and faster PE and replacing LSTMs with a faster and simpler GRU, and more efficient LGB for efficient and accurate forecast reconciliation such that we have WT-NNAR-GRU-LGB, the complexity and computation intensiveness of the proposed WT-NNAR-LSTM-GBM model could be reduced whilst improving accuracy and reducing computational efficiency.
- In Chapter 6, the proposed wavelet-MODWT-GRU hybrid framework applied only LA, DB, and MB wavelet filters based on the MODWT wavelet transformation approach. In the future, studies could use other wavelet filters, specifically, the Morlet wavelet with rapid and drastic structure which could be effective at capturing the inherent transient features of wind data. Furthermore, MODWT could also be replaced by more efficient wavelet packet transformation (WPT) with higher frequency resolution, more balanced frequency analysis, and optimal noise extraction. For instance, studies could evaluate the performance of wavelet-WPT-Mortlet-XGBoost in the ultra-short and long-term horizon to further improve accuracy, robustness, and efficiency in wind speed forecasting.
- Considering Chapters 4-6, future research work could focus on improving wind speed prediction accuracy and generalisability using more complex and larger datasets to test the influence of NWP variables (such as wind direction, humidity, air pressure, temperature) on wind speed from varying regions outside the Southern African region at an medium and long-term horizon.
- The proposed framework wavelet-ML hybrids are currently limited to wind energy (i.e. wind speed) and power outage forecasting. It would be interesting to measure their performance in another branch of renewable energy, in particular, solar forecasting and other different fields founded on time series forecasting such as financial markets.

This page is intentionally blank

References

1. Chaturvedi, D. K., & Isha, I. (2016). Solar power forecasting: A review. *International Journal of Computer Applications*, *145*(6), 28-50.
2. Shafiullah, M., Ahmed, S. D., & Al-Sulaiman, F. A. (2022). Grid integration challenges and solution strategies for solar pv systems: A review. *IEEE Access*, *10*, 52233-52257.
3. Chang, W. Y. (2013). Short-term wind power forecasting using the enhanced particle swarm optimization based hybrid method. *Energies*, *6*(9), 4879-4896.
4. Soman, S. S., Zareipour, H., Malik, O., & Mandal, P. (2010, September). A review of wind power and wind speed forecasting methods with different time horizons. In *North American power symposium 2010* (pp. 1-8).
5. IEA, I. (2021). World energy balances: Overview. IEA: Paris, France. *Retrieved from*: https://www.iea.org/reports/key-world-energy-statistics-2021
6. IEA,I. (2022). Wind Electricity. IEA: Paris, France. *Retrieved from*: https://www.iea.org/reports/wind-electricity
7. IEA, G. (2022). Energy Review: CO2 Emissions in 2021. *Retrieved from:* https://www.iea.org/reports/global-energy-review-co2-emissions-in-2021-2, Licence: CC BY 4.0
8. Zwane, N., Tazvinga, H., Botai, C., Murambadoro, M., Botai, J., De Wit, J., ... & Mabhaudhi, T. (2022). A bibliometric analysis of solar energy forecasting studies in Africa. *Energies*, *15*(15), 5520.
9. Rae, G., & Erfort, G. (2020). Offshore wind energy-South Africa's untapped resource. *Journal of Energy in Southern Africa*, *31*(4), 26-42.
10. Wright, J. G., & Calitz, J. R. (2021). Statistics of utility-scale power generation in South Africa H1-2021.
11. Pierce, W. T., & Ferreira, B. A. (2022). Statistics of utility-scale power generation in South Africa in 2021.
12. WASA Project. (n.d.). WASA high-resolution wind resource map

13. Department of Energy. (2019). Integrated Resource Plan (IRP 2019).
14. Department of Energy. (2010). Integrated Resource Plan (IRP 2010).
15. Sivhugwana, K. S., & Ranganai, E. (2025). Short-term forecasting of unplanned power outages using machine learning algorithms: A robust feature engineering strategy against multicollinearity and nonlinearity. *Energies*, *18*(18), 4994.
16. Akinbami, O. M., Oke, S. R., & Bodunrin, M. O. (2021). The state of renewable energy development in South Africa: An overview. *Alexandria Engineering Journal*, *60*(6), 5077-5093.
17. Stats SA. (2022). Electricity, Gas And Water Supply Industry Report 2021.
18. Oladunni, O. J., Mpofu, K., & Olanrewaju, O. A. (2022). Greenhouse gas emissions and its driving forces in the transport sector of South Africa. *Energy Reports*, *8*, 2052-2061.
19. Pretorius, I., Piketh, S. J., & Burger, R. P. (2015). The impact of the South African energy crisis on emissions. *WIT Trans. Ecol. Environ*, *198*, 255-264.
20. Marquard, A., Ahjum, F., Bergh, C., Von Blottnitz, H., Burton, J., Cohen, B., ... & Winkler, H. (2023). Exploring net zero pathways for South Africa-An initial study.
21. Loewald, Chris. (2023).South African Reserve Bank Special Occasional Bulletin of Economic Notes Special OBEN/23/01. *Retrieved from:* https://www.resbank.co.za/content/dam/sarb/publications/special-occasional-bulletins/2023/special-occasional-bulletin-of-economic-notes-2301-august-2023-combined.pdf
22. Sivhugwana, K. S., & Ranganai, E. (2024). An ensemble approach to short-term wind speed predictions using stochastic methods, wavelets and gradient boosting decision trees. Wind, *4*(1), 44–67.

23. Sohoni, V., Gupta, S. C., & Nema, R. K. (2016). A critical review on wind turbine power curve modelling techniques and their applications in wind based energy systems. *Journal of Energy*, *2016*(1), 8519785.
24. Powell, W. R. (1981). An analytical expression for the average output power of a wind machine. *Solar Energy*, *26*(1), 77-80.
25. Fabbri, A., Roman, T. G., Abbad, J. R., & Quezada, V. M. (2005). Assessment of the cost associated with wind generation prediction errors in a liberalized electricity market. *IEEE Transactions on Power Systems*, *20*(3), 1440-1446.
26. Gensler, A. (2019). Wind Power Ensemble Forecasting: Performance Measures and Ensemble Architectures for Deterministic and Probabilistic Forecasts (Vol. 12). Kassel university press GmbH.
27. Mallipeddi, R., & Suganthan, P. N. (2014). Unit commitment–a survey and comparison of conventional and nature inspired algorithms. *International Journal of Bio-Inspired Computation*, *6*(2), 71-90.
28. Sivhugwana, K. S., & Ranganai, E. (2024). Short-term wind speed prediction via sample entropy: A hybridisation approach against gradient disappearance and explosion. *Computation*, *12*(8), 163.
29. Sivhugwana, K. S., & Ranganai, E. (2025). Wind speed forecasting with differentially evolved minimum-bandwidth filters and gated recurrent units. *Forecasting*, *7*(2), 27.
30. Zhang, Y., Zhao, Y., Kong, C., & Chen, B. (2020). A new prediction method based on VMD-PRBF-ARMA-E model considering wind speed characteristic. *Energy Conversion and Management, 203*, 112254.
31. Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, *50*, 159-1.
32. Tong, W. (2010). *Fundamentals of wind energy* (Vol. 44, p. 112). Southampton, UK: WIT press.
33. Kalmikov, A. G. (2017). *Wind Power Fundamentals*.

34. Stival, L. J. L., Guetter, A. K., & de Andrade, F. O. (2017). The impact of wind shear and turbulence intensity on wind turbine power performance. *Espaço Energia, 27*, 11-20.

35. Gutierrez, ´ A., Porrini, C., & Fovell, R. G. (2020). Combination of wind gust models in convective events. *Journal of Wind Engineering and Industrial Aerodynamics*, *199*, 104118.

36. Schlechtingen, M., Santos, I. F.,& Achiche, S. (2013). Using data-mining approaches for wind turbine power curve monitoring: A comparative study. *IEEE Transactions on Sustainable Energy*, *4(*3), 671-679.

37. Betz, A. (2013). The maximum of the theoretically possible exploitation of wind by means of a wind motor. *Wind Engineering*, *37*(4), 441-446.

38. Huleihil, M., & Mazor, G. (2012). Wind turbine power: The betz-limit and beyond. *In Advances in wind power.* IntechOpen.

39. Carriveau, R. (Ed.). (2012). Advances in wind power. BoD–Books on Demand.

40. Wei, T., Haque, M. M. M., Yong, S., & Yan, L. (2019) A Basic Approach for Designing Pitch, Yaw and Supervisory Control System Wei of Wind Turbines. *International Journal of Scientific and Research Publications (IJSRP)*, *9*.

41. Gonzalez-Longatt, ´ F., Wall, P., & Terzija, V. (2012). Wake effect in wind farm performance: Steady-state and dynamic behavior. *Renewable Energy*, *39*(1), 329-338.

42. Adaramola, M. S., & Krogstad, P. A.˚ (2011). Experimental investigation of wake effects on wind turbine performance. *Renewable energy*, *36*(8), 2078-2086.

43. Borunda, M., Garduno, R., de la Cruz Soto, J., & Figueroa Díaz, R. A. (2024). Intelligent control of an experimental small-scale wind turbine. *Energies, 17*(22), 5656.

44. Aranizadeh, A., Mirmozaffari, M., & Khalatabadi Farahani, B. (2025). Maximizing Wind Turbine Power Generation Through Adaptive Fuzzy Logic Control for Optimal Efficiency and Performance. Wind, *5*(1), 4.

45. Apata, O., & Oyedokun, D. T. O. (2020). An overview of control techniques for wind turbine systems. *Scientific African*, *10*, e00566.

46. Zhang, T., Xu, X., Wang, S., Xing, Y., Dou, P., Ji, R., ... & Yang, P. (2025). Investigating the influence of yaw control on wind farm dynamic characteristics for three conceptual floating offshore wind turbines. *Ocean Engineering*, *339*, 122218.

47. Bakırcı, M., & Yılmaz, S. (2018). Theoretical and computational investigations of the optimal tip-speed ratio of horizontal-axis wind turbines. *Engineering Science and Technology, an International Journal*, *21*(6), 1128-1142.

48. Ragheb, M. (2014). Optimal rotor tip speed ratio. Lecture notes of Course no. NPRE, 475.

49. Laz′ ar,′ I., Hadnagy, I., Bertalan-Balazs,′ B., Bertalan, L., & Szegedi, S. (2024). Comparative examinations of wind speed and energy extrapolation methods using remotely sensed data–A case study from Hungary. *Energy Conversion and Management*, *24*, 100760.

50. Polikar, R. (1999). The story of wavelets. *Physics and modern topics in mechanical and electrical engineering*, 192-197.

51. Penedo, S. R., Netto, M. L., & Justo, J. F. (2019). Designing digital filter banks using wavelets. EURASIP. *Journal on Advances in Signal Processing*, *2019*, 1-11.

52. Mallat, S. (2009). *A Wavelet Tour of Signal Processing: The Sparse Way (3rd ed.).* Academic Pres.

53. Bachman, G., Narici, L., & Beckenstein, E. (2000). *Fourier and wavelet analysis* (Vol. 586). New York: Springer.

54. Boggess, A., & Narcowich, F. J. (2015). *A first course in wavelets with Fourier analysis*. John Wiley & Sons.

55. Vetterli, M., & Kovacevic, J. (1995). *Wavelets and subband coding* (No. BOOK).

56. Skodras, A. N. (2003). Discrete wavelet transform: an introduction. *Hellenic Open University Technical Report*, *2*(1), 1-26.

57. Daubechies, I. (1992). *Ten lectures on wavelets. Society for industrial and applied mathematics*.

58. Grochenig, K. (2001). *Foundations of time-frequency analysis*. Springer Science & Business Media.

59. Chun-Lin, L. (2010). A tutorial of the wavelet transform. *NTUEE, Taiwan*, *21*(22), 2.

60. Hussain, R. (2011, July). *A concise introduction to wavelets*.

61. Kovacevic, J., Goyal, V. K., & Vetterli, M. (2013). Fourier and wavelet signal processing. *Fourier Wavelets. org*, 1-294.

62. Banks, C. S. P. R. F. (2020). MUS421/EE367B Lecture 9 Multirate, Polyphase, and Wavelet Filter Banks.

63. Strang, G., & Nguyen, T. (1996). *Wavelets and filter banks*, SIAM.

64. Giron-Sierra, J. M. (2016). *Digital Signal Processing with Matlab Examples, vol 2: Decomposition, Recovery, Data-Based Actions*. Springer.

65. Dibal, P. Y., Onwuka, E., Agajo, J., & Alenoghena, C. (2019). Analysis of wavelet transform design via filter bank technique. *In Wavelet transform and Complexity*. London, UK: IntechOpen.

66. Por, E., Van Kooten, M., & Sarkovic, V. (2019). Nyquist–Shannon sampling theorem. *Leiden University, 1*(1), 1-2.

67. Strang, G. (1996). Creating and comparing wavelets. In *ICAOS'96: 12th International Conference on Analysis and Optimization of Systems Images, Wavelets and PDEs Paris, June 26–28, 1996* (pp. 143-153). Springer Berlin Heidelberg.

68. Gonzalez, R. C. (2009). *Digital image processing*. Pearson education india.

69. Mallat, S. G. (1989). A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, *11*(7), 674-693.

70. Mertins, A. (1999). *Signal Analysis: Wavelet Filter Banks, Time-Frequency Transforms and applications*, John Wiley & Sons Ltd.

71. Antoniadis, A. (1997). Wavelets in statistics: a review. *Journal of the Italian Statistical Society*, *6*, 97-130.

72. Sadowsky, J. (1994). The continuous wavelet transform: A tool for signal investigation and understanding. *Johns Hopkins APL Technical Digest*, *15*, 306-306.

73. Hou, T., & Qin, H. (2012). Continuous and discrete Mexican hat wavelet transforms on manifolds. *Graphical Models*, *74*(4), 221-232.

74. Piotrowski, P., Rutyna, I., Baczyński, D., & Kopyt, M. (2022). Evaluation metrics for wind power forecasts: A comprehensive review and statistical analysis of errors. *Energies*, *15*(24), 9657.

75. Bazionis, I. K., & Georgilakis, P. S. (2021). Review of deterministic and probabilistic wind power forecasting: Models, methods, and future research. *Electricity*, *2*(1), 13-47.

76. Xie, Y., Li, C., Li, M., Liu, F., & Taukenova, M. (2023). An overview of deterministic and probabilistic forecasting methods of wind energy. *Iscience*, *26*(1).

77. Yousuf, M. U., Al-Bahadly, I., & Avci, E. (2019). Current perspective on the accuracy of deterministic wind speed and power forecasting. *IEEE Access*, *7*, 159547159564.

78. Zhao, X., Wang, S., & Li, T. (2011). Review of evaluation criteria and main methods of wind power forecasting. *Energy Procedia*, *12*, 761-769.

79. Zhao, X., Liu, J., Yu, D., & Chang, J. (2018). One-day-ahead probabilistic wind speed forecast based on optimized numerical weather prediction data. *Energy Conversion and Management*, *164*, 560-569.

80. He, B., Ye, L., Pei, M., Lu, P., Dai, B., Li, Z., & Wang, K. (2022). A combined model for short-term wind power forecasting based on the analysis of numerical weather prediction data. *Energy Reports*, *8*, 929-939.

81. Ahn, E., & Hur, J. (2023). A short-term forecasting of wind power outputs using the enhanced wavelet transform and arimax techniques. *Renewable Energy*, *212*, 394-402.

82. Box, G.E.P.; Jenkins, G.M. Time Series Analysis: Forecasting and Control; Holden-Day: San Francisco, CA, USA, 1976.

83. Hyndman, R. J., & Athanasopoulos, G. (2012). *Forecasting: Principles and practice.* An online textbook. Scientific Research Publishing.

84. Wei, W. (2006). *Time Series Analysis: Univariate and Multivariate Methods (2nd Edition.)*. Boston: Addison-Wesley.

85. Box, George EP, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

86. Sivhugwana, K. S., & Ranganai, E. (2020). Intelligent techniques, harmonically coupled and SARIMA models in forecasting solar radiation data: A hybridisation approach. *Journal of Energy in Southern Africa*, *31*(3), 14-37.

87. Ranganai, E., & Sigauke, C. (2020). Capturing long-range dependence and harmonic phenomena in 24-hour solar irradiance forecasting: A quantile regression robustification via forecasts combination approach. *IEEE Access*, *8*, 172204-172218.

88. Liu, X., Lin, Z., & Feng, Z. (2021). Short-term offshore wind speed forecast by seasonal ARIMA-A comparison against GRU and LSTM. *Energy, 227,* 120492.

89. Gomes, P., & Castro, R. (2012). Wind speed and wind power forecasting using statistical models: autoregressive moving average (ARMA) and artificial neural networks (ANN). *International Journal of Sustainable Energy Development*, *1*(1/2).

90. Liu, Y., Guan, L., Hou, C., Han, H., Liu, Z., Sun, Y., & Zheng, M. (2019). Wind power short-term prediction based on LSTM and discrete wavelet transform. *Applied Sciences*, *9*(6), 1108.

91. Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM* Journal *of research and development*, *3*(3), 210-229.

92. Patel, Y., & Deb, D. (2022). Machine Intelligent Hybrid Methods Based on Kalman Filter and Wavelet Transform for Short Term Wind Speed Prediction. *Wind*, *2*(1), 37-50.

93. Xie, A., Yang, H., Chen, J., Sheng, L., & Zhang, Q. (2021). A short-term wind speed forecasting model based on a multi-variable long short term memory network. *Atmosphere*, *12*(5), 651.

94. Xiang, J., Qiu, Z., Hao, Q., & Cao, H. (2020). Multi-time scale wind speed prediction based on WT-bi-LSTM. In *MATEC Web of Conferences* (Vol. 309, p. 05011). EDP Sciences.

95. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

96. Liu, G., Wang, C., Qin, H., Fu, J., & Shen, Q. (2022). A novel hybrid machine learning model for wind speed probabilistic forecasting. *Energies*, *15*(19), 6942.

97. Al Daoud, E. (2019). Comparison between XGBoost, LightGBM and CatBoost using a home credit dataset. *International Journal of Computer and Information Engineering*, *13*(1), 6-10.

98. Cai, R., Xie, S., Wang, B., Yang, R., Xu, D., & He, Y. (2020). Wind speed forecasting based on extreme gradient boosting. *IEEE Access*, *8*, 175063-175069.

99. Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

100. Daniel, L. O., Sigauke, C., Chibaya, C., & Mbuvha, R. (2020). Short-term wind speed forecasting using statistical and machine learning methods. *Algorithms*, *13*(6), 132.

101. Botha, N., & van der Walt, C. M. (2017, November). Forecasting wind speed using support vector regression and feature selection. In *2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech)*

(pp. 181-186). IEEE.

102. Raschka, S. (2020). *STAT 451: Machine Learning Lecture Notes*.
103. Nasteski, V. (2017). An overview of the supervised machine learning methods. *Horizons. b*, *4*(51-62), 56.
104. Sun, S., Cao, Z., Zhu, H., & Zhao, J. (2019). A survey of optimization methods from a machine learning perspective. *IEEE transactions on cybernetics*, *50*(8), 36683681.
105. Bilmes, J. (2020). Underfitting and overfitting in machine learning. *UW ECE course notes*, 5.
106. Ying, X. (2019, February). An overview of overfitting and its solutions. In *Journal of physics: Conference series* (Vol. 1168, p. 022022). IOP Publishing.
107. Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, *3*(Mar), 1157-1182.
108. da Silva, G. F., da Silva, A. S. A., & Stosic, T. (2019). Using Sample Entropy to assess complexity of wind speed dynamics. *Acta Scientiarum. Technology*, *41*, e38954.
109. Delgado-Bonal, A., & Marshak, A. (2019). Approximate entropy and sample entropy: A comprehensive tutorial. *Entropy*, *21*(6), 541.
110. Bandt, C., & Pompe, B. (2002). Permutation entropy: a natural complexity measure for time series. *Physical review letters*, *88*(17), 174102.
111. Bhavsar, R., Helian, N., Sun, Y., Davey, N., Steffert, T., & Mayor, D. (2018). Efficient methods for calculating sample entropy in time series data analysis. *Procedia computer science*, *145*, 97-104.
112. Celeux, G., & Soromenho, G. (1996). An entropy criterion for assessing the number of clusters in a mixture model. *Journal of classification*, *13*(2), 195-212.
113. Henry, M., & Judge, G. (2019). Permutation entropy and information recovery in nonlinear dynamic economic time series. *Econometrics*, *7*(1), 10.

114. Popovic, M. (2017). Researchers in an entropy wonderland: A review of the entropy concept. *arXiv preprint arXiv:1711.07326*.

115. Xinxin, W., Xiaopan, S., Xueyi, A., & Shijia, L. (2023). Short-term wind speed forecasting based on a hybrid model of ICEEMDAN, MFE, LSTM and informer. *Plos one*, *18*(9), e0289161.

116. Hodge, B. M., Orwig, K., & Milligan, M. (2012). *Examining information entropy approaches as wind power forecasting performance metrics* (No. NREL/CP-5500-53515). National Renewable Energy Lab. (NREL), Golden, CO (United States).

117. Liu, Y., Guan, L., Hou, C., Han, H., Liu, Z., Sun, Y., & Zheng, M. (2019). Wind power short-term prediction based on LSTM and discrete wavelet transform. *Applied Sciences*, *9*(6), 1108.

118. Yadav, A., Jha, C. K., & Sharan, A. (2020). Optimizing LSTM for time series prediction in Indian stock market. *Procedia Computer Science*, *167*, 2091-2100.

119. Saha, S. (2024). Comprehensive Forecasting-Based Analysis of Hybrid and Stacked Stateful/Stateless Models. *arXiv preprint arXiv:2404.19306*.

120. Prema, V., Sarkar, S., Rao, K. U., & Umesh, A. (2019). LSTM based Deep Learning model for accurate wind speed prediction. *Data Sci. Mach. Learn*, *1*, 6-11.

121. Gangwar, S., Bali, V., & Kumar, A. (2020). Comparative analysis of wind speed forecasting using LSTM and SVM. *EAI Endorsed Transactions on Scalable Information Systems*, *7*(25), e1-e1.

122. Rodr´ıguez-Perez,´ Raquel, and Jurgen¨ Bajorath. "Evolution of support vector machine and regression modeling in chemoinformatics and drug discovery." *Journal of Computer-Aided Molecular Design* 36, no. *5* (2022): 355-362.

123. Quan, J., & Shang, L. (2021). An Ensemble Model of Wind Speed Forecasting Based on Variational Mode Decomposition and Bare-Bones Fireworks Algorithm. *Mathematical Problems in Engineering*, *2021*(1), 6632390.

124. Yang, Z.J. (2008). Kernel-based support vector machines. *Comput. Eng. Appl.*, *44*, 1–6.

125. Dhiman, H. S., Anand, P., & Deb, D. (2018). Wavelet transform and variants of SVR with application in wind forecasting. *In Innovations in infrastructure: Proceedings of ICIIF 2018* (pp. 501-511). Singapore: Springer Singapore.

126. Mohandes, M. A., Halawani, T. O., Rehman, S., & Hussain, A. A. (2004). Support vector machines for wind speed prediction. *Renewable energy*, *29*(6), 939-947.

127. Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of machine learning research, 1*(Jun), 211-244.

128. Sun, G., Chen, Y., Wei, Z., Li, X., & Cheung, K. W. (2014). Day-ahead wind speed forecasting using relevance vector machine. *Journal of Applied Mathematics*, *2014*.

129. Jinhua, Z., Jie, Y., Wenjing, W. U., & Yongqian, L. I. U. (2019). Research on short-term forecasting and uncertainty of wind turbine power based on relevance vector machine. *Energy Procedia*, *158*, 229-236.

130. Zang, H., Fan, L., Guo, M., Wei, Z., Sun, G., & Zhang, L. (2016). Short-term wind power interval forecasting based on an EEMD-RT-RVM model. *Advances in Meteorology*, 2016.

131. Kecman, V. (2005). Support vector machines–an introduction. In *Support vector machines: theory and applications (pp. 1-47)*. Berlin, Heidelberg: Springer Berlin Heidelberg.

132. Natras, R., Soja, B., & Schmidt, M. (2022). Ensemble machine learning of random forest, AdaBoost and XGBoost for vertical total electron content forecasting. Remote Sensing, *14*(15), 3547.

133. Breiman, L. (2001). Random forests. *Machine learning*, 45, 5-32.

134. Louppe, G. (2014). *Understanding random forests: From theory to practice* (Doctoral dissertation), Universite de Liege (Belgium).

135. Zhou, Z., Li, X., & Wu, H. (2016, December). Wind power prediction based on random forests. In *2016 4th International Conference on Electrical & Electronics Engineering and Computer Science (ICEEECS 2016)* (pp. 352-356). Atlantis Press.

136. Drisya, G. V., Asokan, K., & Kumar, K. S. (2022). Wind speed forecast using random forest learning method. *arXiv preprint arXiv:2203.14909*.

137. Samuel, G. G., Salimath, G. F., Porselvi, T., & Karthikeyan, V. (2021, July). Improved Prediction of Wind Speed Using Machine Learning. In *Journal of Physics: Conference Series* (Vol. 1964, No. 5, p. 052005). IOP Publishing.

138. Ho, C. Y., Cheng, K. S.,& Ang, C. H. (2023). Utilizing the Random Forest Method for Short-Term Wind Speed Forecasting in the Coastal Area of Central Taiwan. *Energies*, *16*(3), 1374.

139. Chen, H., Anfinsen, S. N., Birkelund, Y., & Yuan, F. (2021). Probability distributions for wind speed volatility characteristics: A case study of Northern Norway. *Energy Reports*, 7, 248-255.

140. Qin, S., & Liu, D. (2023). Distribution Characteristics of Wind Speed Relative Volatility and Its Influence on Output Power. *Journal of Marine Science and Engineering*, *11*(5), 967.

141. Pessanha, J. F., Castellani, V. L., & Andrade, V. A. (2017). Short-Term Wind Power Forecasting Based On Quantile Regression. *Brazil Windpower*.

142. Qin, X., Sheng, H., & Dong, X. (2023). Interval Wind-Speed Forecasting Model Based on Quantile Regression Bidirectional Minimal Gated Memory Network and Kernel Density Estimation. *Arabian Journal for Science and Engineering*, *48*(2), 1625-1639.

143. Qiu, W., Zhang, W., Wang, G., Guo, Z., Zhao, J., & Ma, K. (2024). Combined wind speed forecasting model based on secondary decomposition and quantile regression closed-form continuous-time neural network. *International Journal of Green Energy*, *21*(8), 1793-1814.

144. Saeed, A., Li, C., & Gan, Z. (2022, February). Short-Term Wind Speed Interval Prediction using LUBE based Quasi-Recurrent Neural Network. *In Journal of Physics: Conference Series* (Vol. 2189, No. 1, p. 012015). *IOP Publishing.*

145. Liu, F., Li, C., Xu, Y., Tang, G., & Xie, Y. (2021). A new lower and upper bound estimation model using gradient descend training method for wind speed interval prediction. *Wind Energy*, *24*(3), 290-304.

146. Liu, F., Tao, Q., Yang, D., & Sidorov, D. (2021). Bidirectional gated recurrent unitbased lower upper bound estimation method for wind power interval prediction. *IEEE Transactions on Artificial Intelligence*, *3*(3), 461-469.

147. Liu, H., Tian, H. Q., & Li, Y. F. (2012). Comparison of two new ARIMA-ANN and ARIMA-Kalman hybrid methods for wind speed prediction. *Applied Energy*, *98*, 415-424.

148. Liu, H., Tian, H. Q., & Li, Y. F. (2015). An EMD-recursive ARIMA method to predict wind speed for railway strong wind warning system. *Journal of Wind Engineering and Industrial Aerodynamics*, *141*, 27-38.

149. Feng, F., Si, A., & Rao, G. Early fault diagnosis of bearing based on wavelet correlation permutation entropy. *J. Engineering*, *48*, 73-79, 2012.

150. Chen, N., Sun, H., Zhang, Q., & Li, S. (2022). A Short-Term Wind Speed Forecasting Model Based on EMD/CEEMD and ARIMA-SVM Algorithms. *Applied Sciences*, *12*(12), 6085.

151. Niu, D., Pu, D., & Dai, S. (2018). Ultra-short-term wind-power forecasting based on the weighted random forest optimized by the niche immune lion algorithm. *Energies*, *11*(5), 1098.

152. Singh, S. N., & Mohapatra, A. (2019). Repeated wavelet transform based ARIMA model for very short-term wind speed forecasting. *Renewable energy*, *136*, 758-768.

153. Tascikaraoglu, A., & Uzunoglu, M. (2014). A review of combined approaches for prediction of short-term wind speed and power. *Renewable and Sustainable Energy Reviews*, *34*, 243-254.

154. Su, Z., Wang, J., Lu, H., & Zhao, G. (2014). A new hybrid model optimized by an intelligent optimization algorithm for wind speed forecasting. *Energy conversion and management*, *85*, 443-452.

155. Liu, D., Niu, D., Wang, H., & Fan, L. (2014). Short-term wind speed forecasting using wavelet transform and support vector machines optimized by genetic algorithm. *Renewable energy*, *62*, 592-597.

156. Wang, L., Guo, Y., Fan, M., & Li, X. (2022). Wind speed prediction using measurements from neighboring locations and combining the extreme learning machine and the AdaBoost algorithm. *Energy Reports*, *8*, 1508-1518.

157. Qu, Z., Hou, X., Hu, W., Yang, R., & Ju, C. (2023). Wind power forecasting based on improved variational mode decomposition and permutation entropy. *Clean Energy*, *7*(5), 1032-1045.

158. Yang, M., Dai, B., Wang, J., Chen, X., Sun, Y., & Li, B. (2021). Day-ahead wind power combination forecasting based on corrected numerical weather prediction and entropy method. *IET Renewable Power Generation*, *15*(7), 1358-1368.

159. Liu, Z., & Liu, H. (2023). A novel hybrid model based on GA-VMD, sample entropy reconstruction and BiLSTM for wind speed prediction. *Measurement*, *222*, 113643.

160. Catalao, ˜ J. D. S., Pousinho, H. M. I., & Mendes, V. M. F. (2011). Short-term wind power forecasting in Portugal by neural networks and wavelet transform. *Renewable energy*, *36*(4), 1245-1251.

161. Shi, J., Ding, Z., Lee, W. J., Yang, Y., Liu, Y., & Zhang, M. (2013). Hybrid forecasting model for very-short term wind power forecasting based on grey

relational analysis and wind speed distribution features. *IEEE Transactions on Smart Grid*, *5*(1), 521-526.

162. Zheng, X., Jia, D., Lv, Z., Luo, C., Zhao, J., & Ye, Z. (2023). Short-time wind speed prediction based on Legendre multi-wavelet neural network. *CAAI Transactions on Intelligence Technology*, *8*(3), 946-962.
163. Saroha, S., & Aggarwal, S. K. (2018). Wind power forecasting using wavelet transforms and neural networks with tapped delay. *CSEE Journal of Power and Energy Systems*, *4*(2), 197-209.
164. Ramesh Babu, N., & Arulmozhivarman, P. (2013). Improving forecast accuracy of wind speed using wavelet transform and neural networks. *Journal of Electrical Engineering and Technology*, *8*(3), 559-564.
165. Wang, J. (2014). A Hybrid Wavelet Transform Based Short-Term Wind Speed Forecasting Approach. *The Scientific World Journal*, *2014*(1), 914127.
166. D Chandra, R., Sailaja Kumari, M., Sydulu, M., Grimaccia, F., & Mussetta, M. (2014). Adaptive wavelet neural network based wind speed forecasting studies. *Journal of Electrical Engineering & Technology*, *9*(6), 1812-1821.
167. Zhang, J., Wei, Y., Tan, Z. F., Wang, K., & Tian, W. (2017). A hybrid method for short-term wind speed forecasting. *Sustainability*, *9*(4), 596.
168. Kang, A., Tan, Q., Yuan, X., Lei, X., & Yuan, Y. (2017). Short-Term Wind Speed Prediction Using EEMD-LSSVM Model. *Advances in Meteorology*, *2017*(1), 6856139.
169. Wang, K., Niu, D., Sun, L., Zhen, H., Liu, J., De, G., & Xu, X. (2019). Wind power short-term forecasting hybrid model based on CEEMD-SE method. *Processes*, *7*(11), 843.
170. Zhang, Y., Yang, S., Guo, Z., Guo, Y., & Zhao, J. (2019). Wind speed forecasting based on wavelet decomposition and wavelet neural networks optimized by the Cuckoo search algorithm. *Atmospheric and Oceanic Science Letters*, *12*(2), 107-115.

171. Tefera, E., Martínez-Ballesteros, M., Troncoso, A., & Martínez-Álvarez, F. (2023, August). A new hybrid cnn-LSTM for wind power forecasting in ethiopia. *In International Conference on Hybrid Artificial Intelligence Systems* (pp. 207-218). Cham: Springer Nature Switzerland.

172. Khelil, K., Berrezzek, F., & Bouadjila, T. (2020, May). DWT-based Wind Speed Forecasting Using Artificial Neural Networks in the region of Annaba. In *2020 1st International Conference on Communications, Control Systems and Signal Processing (CCSSP)* (pp. 508-512).

173. Zhu, A., Zhao, Q., Wang, X., & Zhou, L. (2022). Ultra-short-term wind power combined prediction based on complementary ensemble empirical mode decomposition, whale optimisation algorithm, and elman network. *Energies*, *15*(9), 3055.

174. Mohammed, M. A., & Ahmed, L. A. (2023). Forecasting wind speed using the proposed wavelet neural network. *Discrete Dynamics in Nature and Society*, 2023.

175. Liu, Z., Li, X., & Zhao, H. (2023). Short-Term Wind Power Forecasting Based on Feature Analysis and Error Correction. *Energies*, *16*(10), 4249.

176. Lv, S., Wang, L., & Wang, S. (2023). A hybrid neural network model for shortterm wind speed forecasting. *Energies*, *16*(4), 1841.

177. Shah, A. A., Aftab, A. A., Han, X., Baloch, M. H., Honnurvali, M. S., & Chauhdary, S. T. (2023). Prediction Error-Based Power Forecasting of Wind Energy System Using Hybrid WT–ROPSO–NARMAX Model. *Energies*, *16*(7), 3295.

178. Hewamalage, H., Ackermann, K., & Bergmeir, C. (2023). Forecast evaluation for data scientists: common pitfalls and best practices. *Data Mining and Knowledge Discovery*, *37*(2), 788-832.

179. Huang, Huang, C. M., Chen, S. J., Yang, S. P., & Chen, H. J. (2023). One-day-ahead hourly wind power forecasting using optimized ensemble prediction methods. *Energies*, *16*(6), 2688.

180. Martínez, F., Frías, M. P., Charte, F., & Rivera, A. J. (2019). Time Series Forecasting with KNN in R: the tsfknn Package.

181. Dghais, A. A. A., & Ismail, M. T. (2013). A comparative study between discrete wavelet transform and maximal overlap discrete wavelet transform for testing stationarity. *Int J. Math. Comput. Sei. Eng*, *7*, 1184-1188.

182. Cornish, C. R., Bretherton, C. S., & Percival, D. B. (2006). Maximal overlap wavelet statistical analysis with application to atmospheric turbulence. *Boundary-Layer Meteorology*, *119*(2), 339-374.

183. Rodrigues, D. V., Zuo, D., & Li, C. (2021). A modwt-based algorithm for the identification and removal of jumps/short-term distortions in displacement measurements used for structural health monitoring. *IoT*, *3*(1), 60-72.

184. Paramasivam, S., PL, S. A., & Sathyamoorthi, P. (2022). Maximal overlap discrete wavelet transform-based power trace alignment algorithm against random delay countermeasure. *Etri Journal*, *44*(3), 512-523.

185. Zhang, Z., Telesford, Q. K., Giusti, C., Lim, K. O., & Bassett, D. S. (2016). Choosing wavelet methods, filters, and lengths for functional brain network construction. *PloS one*, *11*(6), e0157243.

186. Morris, J. M., & Peravali, R. (1999). Minimum-bandwidth discrete-time wavelets. *Signal Processing, 76*(2), 181-193.

187. Merry, R. J. E. (2005). Wavelet theory and applications: a literature study.

188. Gröchenig, K. (2001). *Foundations of time-frequency analysis*. Springer Science & Business Media.

189. Kovacevic, J., Goyal, V. K., & Vetterli, M. (2013). Fourier and wavelet signal processing. *Fourier Wavelets. org*, 1-294.

190. Alarcon-Aquino, V., & Barria, J. A. (2009). Change detection in time series using the maximal overlap discrete wavelet transform. *Latin American applied research*, *39*(2), 145-152.

191. Wei, Q., Liu, D. H., Wang, K. H., Liu, Q., Abbod, M. F., Jiang, B. C., ... & Shieh, J. S. (2012). Multivariate multiscale entropy applied to center of pressure signals analysis: an effect of vibration stimulation of shoes. *Entropy*, *14*(11), 2157-2172.

192. Lake, D. E., Richman, J. S., Griffin, M. P., & Moorman, J. R. (2002). Sample entropy analysis of neonatal heart rate variability. *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, *283*(3), R789-R797.

193. Richman, J. S., & Moorman, J. R. (2000). Physiological time-series analysis using approximate entropy and sample entropy. *American journal of physiology-heart and circulatory physiology*, *278*(6), H2039-H2049.

194. Wesley, J., Rhodes, S., Zeitler, D. W., & Alderink, G. (2025). Approximate and Sample Entropy of the Center of Pressure During Unperturbed Tandem Standing: Effect of Altering the Tolerance Window. *Applied Sciences*, *15*(2), 576.

195. Olbrys, J., & Majewska, E. (2022). Approximate entropy and sample entropy algorithms in financial time series analyses. *Procedia Computer Science*, *207*, 255-264.

196. Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *67*(2), 301-320.

197. Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, *11*(4), 341-359.

198. Zaharie, Zaharie, D. (2007, October). A comparative analysis of crossover variants in differential evolution. In *Proceedings of IMCSIT* (Vol. 2007, pp. 171-181).

199. Wang, Wang, Y., Cai, Z., & Zhang, Q. (2011). Differential evolution with composite trial vector generation strategies and control parameters. *IEEE transactions on evolutionary computation*, *15*(1), 55-66.

200. Eltaeib, T., & Mahmood, A. (2018). Differential evolution: A survey and analysis. *Applied Sciences*, *8*(10), 1945.

201. Leon, M., & Xiong, N. (2014, June). Investigation of mutation strategies in differential evolution for solving global optimization problems. In *International conference on artificial intelligence and soft computing* (pp. 372-383). Cham: Springer International Publishing.

202. Eiben, Á. E., Hinterding, R., & Michalewicz, Z. (2002). Parameter control in evolutionary algorithms. *IEEE Transactions on evolutionary computation*, *3*(2), 124-141.

203. Mullen, K. M., Ardia, D., Gil, D. L., Windover, D., & Cline, J. (2011). DEoptim: An R package for global optimization by differential evolution. *Journal of Statistical Software*, *40*, 1-26.

204. Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

205. Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

206. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, *30*.

207. Zheng, H., & Wu, Y. (2019). A xgboost model with weather similarity analysis and feature engineering for short-term wind power forecasting. *Applied Sciences*, *9*(15), 3019.

208. Friedman, J. H. (2002). Stochastic gradient boosting. *Computational statistics & data analysis*, *38*(4), 367-378.

209. Singh, U., Rizwan, M., Alaraj, M., & Alsaidan, I. (2021). A machine learning-based gradient boosting regression approach for wind power production forecasting: A step towards smart grid environments. *Energies*, *14*(16), 5196.

210. Bühlmann, P. (2011). Bagging, boosting and ensemble methods. In *Handbook of computational statistics: Concepts and methods* (pp. 985-1022). Berlin, Heidelberg: Springer Berlin Heidelberg.

211. Solomatine, D. P., & Shrestha, D. L. (2004, July). AdaBoost. RT: a boosting algorithm for regression problems. In *2004 IEEE International Joint Conference on Neural Networks* (IEEE Cat. No. 04CH37541) (Vol. 2, pp. 1163-1168).

212. Zhang, P., & Yang, Z. (2015). A robust AdaBoost. RT based ensemble extreme learning machine. *Mathematical Problems in Engineering*, *2015*(1), 260970.

213. Li, R., Sun, H., Wei, X., Ta, W., & Wang, H. (2022). Lithium battery state-of-charge estimation based on AdaBoost. Rt-RNN. *Energies*, *15*(16), 6056.

214. Chen, J., Xue, X., Ha, M., Yu, D., & Ma, L. (2014). Support vector regression method for wind speed prediction incorporating probability prior knowledge. *Mathematical Problems in Engineering*, *2014*(1), 410489.

215. Vapnik, V. N. (1995). The nature of statistical learning theory. Springer: New York, NY, USA.

216. Zhang, B., Yang, T., Hong, H., Cheng, G., Yang, H., Wang, T., & Cao, D. (2021). Research on long short-term decision-making system for excavator market demand forecasting based on improved support vector machine. *Applied Sciences, 11*(14), 6367.

217. Tzikas, D. G., Wei, L., Likas, A., Yang, Y., & Galatsanos, N. P. (2006). A tutorial on relevance vector machines for regression and classification with applications. *EURASIP News Letter*, *17*(2), 4.

218. Karatzoglou, A., Smola, A., Hornik, K., & Zeileis, A. (2004). kernlab—An S4 package for kernel methods in R. *Journal of Statistical Software, 11*, 1–20.

219. Fletcher, T. (n.d.). (2008). Relevance vector machines explained. *Retrieved from*: https://www.di.fc.ul.pt/~jpn/r/PRML/chp7/Fletcher_RVM_Explained.pdf

220. Funk, S., Camacho, A., Kucharski, A. J., Lowe, R., Eggo, R. M., & Edmunds, W. J. (2019). Assessing the performance of real-time epidemic forecasts: A case study of Ebola in the Western Area region of Sierra Leone, 2014-15. *PLoS computational biology*, *15*(2), e1006785.

221. Gneiting, T., & Katzfuss, M. (2014). Probabilistic forecasting. *Annual Review of Statistics and Its Application*, *1*(1), 125–151.

222. Jordan, A., Krüger, F., & Lerch, S. (2019). Evaluating probabilistic forecasts with scoringRules. *Journal of Statistical Software*, 90(12), 1–37.

223. Gneiting, T., & Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, *102*(477), 359–378.

224. Werner, E., Tilmann, G., Alexander, J., & Fabian, K. (2016). Of quantiles and expectiles: Consistent scoring functions, Choquet representations, and forecast rankings. *Journal of the Royal Statistical Society: Series B (Statistical Methodology*), *78*(3), 505–562.

225. Diebold, F. X., & Mariano, R. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, *13*(3), 253–265.

226. Mincer, J. A., & Zarnowitz, V. (1969). The evaluation of economic forecasts. In Economic forecasts and expectations: Analysis of forecasting behavior and performance (pp. 3-46). NBER.

227. Nowakowska, M., & Tubis, A. (2015). Load shedding and the energy security of Republic of South Africa. *Journal of Polish Safety and Reliability Association*, *6*(3), 99-108.

228. Inglesi-Lotz, R. (2021). The impact of electricity shortage on South Africa's economy. National Science and Technology Forum (NSTF). *Retrieved from*: https://nstf.org.za/wp-content/uploads/2022/05/NSTF-2021-Loadshedding-Roula-Inglesi-Lotz.pdf

229. Olatayo, K. I., Wichers, J. H., & Stoker, P. W. (2018). Energy and economic performance of small wind energy systems under different climatic conditions of South Africa. *Renewable and Sustainable Energy Reviews, 98*, 376-392.

230. Fluri, T. P. (2009). The potential of concentrating solar power in South Africa. *Energy Policy*, 37, 5075–5080.

231. Bosch, J., Staffell, I., & Hawkes, A. D. (2018). Temporally explicit and spatially resolved global offshore wind energy potentials. *Energy, 163*, 766–781.

232. Chikobvu, D., & Mamba, M. (2023). Modelling emissions from Eskom's coal fired power stations using Generalised Linear Models. *Journal of Energy in Southern Africa, 34*(1), 1-14.

233. Pouris, A. (2008). Energy and fuels research in South African universities: A comparative assessment. *Open Information Science Journal*, 1, 1–9.

234. Onaolapo, A. K., Carpanen, R. P., Dorrell, D. G., & Ojo, E. E. (2022). A comparative assessment of conventional and artificial neural networks methods for electricity outage forecasting. *Energies*, *15*(2), 511.

235. Onaolapo, A. K., Pillay-Carpanen, R., Dorrell, D. G., & Ojo, E. E. (2021, January). A comparative evaluation of conventional and computational intelligence techniques for forecasting electricity outage. *In 2021 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA)* (pp. 1-6). IEEE.

236. Pahwa, A. (2004, June). Effect of environmental factors on failure rate of overhead distribution feeders. In *IEEE Power Engineering Society General Meeting, 2004.* (pp. 691-692). IEEE.

237. Pombo-van Zyl, N. (2020, February). Warning: Stage 2 loadshedding returns states Eskom. ESI Africa, Africa's Power Journal. Retrived 9 October 2023, from https://www.esi-africa.com/energy-efficiency/warning-high-risk-of-loadshedding-returns-states-eskom/.

238. Rakotonirainy, R. G., Durbach, I., & Nyirenda, J. (2019). Considering fairness in the load shedding scheduling problem. *ORiON*, *35*(2), 127-144.

239. Shmueli, G. (2010). To explain or to predict? *Statistical science*, *25* (3), 289-310.

240. Breiman, L. (2003). Statistical modeling: The two cultures. *Quality control and applied statistics, 48*(1), 81-82.

241. Katz, H. (2026). Coupled Supply and Demand Forecasting in Platform Accommodation Markets. *arXiv preprint arXiv:2603.00422.*

242. Gupta, H. V., Clark, M. P., Vrugt, J. A., Abramowitz, G., & Ye, M. (2012). Towards a comprehensive assessment of model structural adequacy. *Water Resources Research, 48*(8).

243. Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling* (Vol. 26, p. 13). New York: Springer.

This page is intentionally blank

Publications

Research articles (title pages and abstracts) that emanated from this thesis are provided below.

wind

MDPI

Article

An Ensemble Approach to Short-Term Wind Speed Predictions Using Stochastic Methods, Wavelets and Gradient Boosting Decision Trees

Khathutshelo Steven Sivhugwana and Edmore Ranganai *

Department of Statistics, University of South Africa, Florida Campus, Johannesburg 1709, South Africa; 50400568@mylife.unisa.ac.za

* Correspondence: rangae@unisa.ac.za; Tel.: +27-11-670-9257

Abstract: Considering that wind power is proportional to the cube of the wind speed variable, which is highly random, complex power grid management tasks have arisen as a result. Wind speed prediction in the short term is crucial for load dispatch planning and load increment/decrement decisions. The chaotic intermittency of speed is often characterised by inherent linear and nonlinear patterns, as well as nonstationary behaviour; thus, it is generally difficult to predict it accurately and efficiently using a single linear or nonlinear model. In this study, wavelet transform (WT), autoregressive integrated moving average (ARIMA), extreme gradient boosting trees (XGBoost), and support vector regression (SVR) are combined to predict high-resolution short-term wind speeds obtained from three Southern African Universities Radiometric Network (SAURAN) stations: Richtersveld (RVD); Central University of Technology (CUT); and University of Pretoria (UPR). This hybrid model is termed WT-ARIMA-XGBoost-SVR. In the proposed hybrid, the ARIMA component is employed to capture linearity, while XGBoost captures nonlinearity using the wavelet decomposed subseries from the residuals as input features. Finally, the SVR model reconciles linear and nonlinear predictions. We evaluated the WT-ARIMA-XGBoost-SVR's efficacy against ARIMA and two other hybrid models that substitute XGBoost with a light gradient boosting machine (LGB) component to form a WT-ARIMA-LGB-SVR hybrid model and a stochastic gradient boosting machine (SGB) to form a WT-ARIMA-SGB-SVR hybrid model. Based on mean absolute error (MAE), mean absolute percentage error (MAPE), root mean square error (RMSE), coefficient of determination (R^2), and prediction interval normalised average width (PINAW), the proposed hybrid model provided more accurate and reliable predictions with less uncertainty for all three datasets. This study is critical for improving wind speed prediction reliability to ensure the development of effective wind power management strategies.

Keywords: wavelet transform; short term; wind speed; XGBoost; support vector regression; ARIMA

Citation: Sivhugwana, K.S.; Ranganai, E. An Ensemble Approach to Short-Term Wind Speed Predictions Using Stochastic Methods, Wavelets and Gradient Boosting Decision Trees. *Wind* **2024**, *4*, 44–67. https://doi.org/10.3390/wind4010003

Academic Editors: Francesco Castellani and Zhe Chen

Received: 29 July 2023
Revised: 22 September 2023
Accepted: 19 October 2023
Published: 4 February 2024

1. Introduction

1.1. Motivation

Globally, the continuous increase (which is expected to more than double by 2050) in electricity demand is constantly depleting the Earth's non-renewable resources, such as coal, natural gas, and oil [1]. With the current impetus towards renewable energy, wind power generation is growing in popularity [2,3], as it is a cost-effective and sustainable alternative to generating electricity. In addition to mitigating the increase in carbon footprint by curbing fossil fuel use, wind energy also contributes to sustainable economic progress [4]. The literature shows that adequate energy supplies improve economic stability [1,4]. Furthermore, economic stability, infrastructure development, and improved quality of life are inextricably linked to a sufficient supply of clean and renewable energy [1]. As wind power has attained high penetration on power grids, complex management tasks have emerged due to the high randomness and intrinsic character of wind energy resources [3,5]. Wind

computation

MDPI

Article

Short-Term Wind Speed Prediction via Sample Entropy: A Hybridisation Approach against Gradient Disappearance and Explosion

Khathutshelo Steven Sivhugwana * and Edmore Ranganai

Department of Statistics, University of South Africa, Florida Campus, Johannesburg 1709, South Africa; rangae@unisa.ac.za

* Correspondence: 50400568@mylife.unisa.ac.za; Tel.: +27-11-670-9257

Abstract: High-variant wind speeds cause aberrations in wind power systems and compromise the effective operation of wind farms. A single model cannot capture the inherent wind speed randomness and complexity. In the proposed hybrid strategy, wavelet transform (WT) is used for data decomposition, sample entropy (SampEn) for subseries complexity evaluation, neural network autoregression (NNAR) for deterministic subseries prediction, long short-term memory network (LSTM) for complex subseries prediction, and gradient boosting machine (GBM) for prediction reconciliation. The proposed WT-NNAR-LSTM-GBM approach predicts minutely averaged wind speed data collected at Southern African Universities Radiometric Network (SAURAN) stations: Council for Scientific and Industrial Research (CSIR), Richtersveld (RVD), Venda, and the Namibian University of Science and Technology (NUST). For comparison purposes, in WT-NNAR-LSTM-GBM, LSTM and NNAR are respectively replaced with a k-nearest neighbour (KNN) to form the corresponding hybrids: WT-NNAR-KNN-GBM and WT-KNN-LSTM-GBM. We assessed WT-NNAR-LSTM-GBM's efficacy against NNAR, LSTM, WT-NNAR-KNN-GBM, and WT-KNN-LSTM-GBM as well as the naïve model. The comparative study found that the WT-NNAR-LSTM-GBM model was the most accurate, sharpest, and robust based on mean absolute error, median absolute deviation, and residual analysis. The study results suggest using short-term forecasts to optimise wind power production, enhance grid operations in real-time, and open the door to further algorithmic enhancements.

Keywords: wind speed forecasting; SampEn; LSTM; NNAR; WT-NNAR-LSTM-GBM

Citation: Sivhugwana, K.S.; Ranganai, E. Short-Term Wind Speed Prediction via Sample Entropy: A Hybridisation Approach against Gradient Disappearance and Explosion. *Computation* **2024**, *12*, 163. https://doi.org/10.3390/computation12080163

Academic Editors: Maria Trigka and Elias Dritsas

Received: 25 May 2024
Revised: 7 August 2024
Accepted: 7 August 2024
Published: 12 August 2024

1. Introduction

1.1. Overview

While wind is a clean energy resource abundant in Southern Africa, harnessing its power is a complex and specialised task. Even so, the high-variant behaviour of wind speed causes aberrations in the wind power system, which compromises the effective operation of wind farms and the integration of large volumes of wind power into the power grid [1–5]. On the other hand, reliable and accurate wind power forecasts are crucial to increasing the penetration of wind power into electric grids. For instance, energy utilities and system operators require accurate short-term wind power forecasting information for real-time grid operations and regulation actions [1]. Thus, effective integration of wind energy into existing power grids is heavily reliant on the precision of wind power predictions. Since wind power is highly dependent on wind speed, accurate wind power forecasts can in turn be achieved by predicting wind speed accurately [2].

1.2. Literature Review

Forecasting strategies for wind power are divided into four categories: physical, statistical, machine learning, and hybrid methods [3,6–11]. However, physical methods (such as numerical weather prediction (NWP)) are computationally expensive and inaccurate at

forecasting

MDPI

Article

Wind Speed Forecasting with Differentially Evolved Minimum-Bandwidth Filters and Gated Recurrent Units

Khathutshelo Steven Sivhugwana * and Edmore Ranganai

Department of Statistics, University of South Africa, Florida Campus, Johannesburg 1709, South Africa; rangae@unisa.ac.za

* Correspondence: 50400568@mylife.unisa.ac.za; Tel.: +27-11-670-9257

Abstract: Wind data are often cyclostationary due to cyclic variations, non-constant variance resulting from fluctuating weather conditions, and structural breaks due to transient behaviour (due to wind gusts and turbulence), resulting in unreliable wind power supply. In wavelet hybrid forecasting, wind prediction accuracy depends heavily on the decomposition level (L) and the wavelet filter technique selected. Hence, we examined the efficacy of wind predictions as a function of L and wavelet filters. In the proposed hybrid approach, differential evolution (DE) optimises the decomposition level of various wavelet filters (i.e., least asymmetric (LA), Daubechies (DB), and Morris minimum-bandwidth (MB)) using the maximal overlap discrete wavelet transform (MODWT), allowing for the decomposition of wind data into more statistically sound sub-signals. These sub-signals are used as inputs into the gated recurrent unit (GRU) to accurately capture wind speed. The final predicted values are obtained by reconciling the sub-signal predictions using multiresolution analysis (MRA) to form wavelet-MODWT-GRUs. Using wind data from three Wind Atlas South Africa (WASA) locations, Alexander Bay, Humansdorp, and Jozini, the root mean square error, mean absolute error, coefficient of determination, probability integral transform, pinball loss, and Dawid-Sebastiani showed that the MB-MODWT-GRU at $L = 3$ was best across the three locations.

Keywords: wind speed; wind forecasting; MODWT; differential evolution; GRU; Morris minimum bandwidth

Academic Editor: Sonia Leva

Received: 14 April 2025
Revised: 26 May 2025
Accepted: 3 June 2025
Published: 10 June 2025

Citation: Sivhugwana, K.S.; Ranganai, E. Wind Speed Forecasting with Differentially Evolved Minimum-Bandwidth Filters and Gated Recurrent Units. *Forecasting* **2025**, *7*, 27. https://doi.org/10.3390/forecast7020027

1. Introduction

1.1. Research Motivation

Wind power is clean and environmentally friendly. Furthermore, wind power has multiple economic and societal advantages [1–4]. For instance, wind power is economical, sustainable, and inexhaustible [1–5]. In fact, wind power resources are abundant, and can be captured day and night (when solar energy is unavailable). Consequently, there has been a rapid increase in the volume of wind power penetrating existing electric power grids. Provided that the primary and most significant input or resource to wind power, namely, wind speed, is highly complex and unpredictable, the integration of substantial amounts of wind power into the power grid frequently compromises wind power management strategies [5].

The complex nature of wind speed originates from the fact that wind as a physical quantity is dependent on a variety of complex factors, such as atmospheric pressure fluctuations, topography changes, seasonal variations, elevation above ground level, weather patterns, and land formations, such that it is irregular and variable in both location and time

energies

MDPI

Article

Short-Term Forecasting of Unplanned Power Outages Using Machine Learning Algorithms: A Robust Feature Engineering Strategy Against Multicollinearity and Nonlinearity

Khathutshelo Steven Sivhugwana * and Edmore Ranganai

Department of Statistics, University of South Africa, Florida Campus, Johannesburg 1709, South Africa; rangae@unisa.ac.za

* Correspondence: 50400568@mylife.unisa.ac.za; Tel.: +27-11-670-9257

Abstract

Efficient power grid operations and effective business strategies require accurate prediction of power outages. However, predicting outages is a difficult task due to the large amount of heterogeneous, random, intermittent, and non-linear power grid data characterised by highly complex variable relationships. Attempting to simultaneously quantify these characteristics using a conventional single (linear or nonlinear) model may lead to inaccurate and costly results. To address this, we propose a hybrid RVM-WT-AdaBoostRT-RF framework using power grid data from the Electricity Supply Commission (Eskom) of South Africa. To achieve model interpretability, the least absolute shrinkage and selection operator (LASSO) is first applied to remedy the adverse effects of multicollinearity through regularisation and variable selection. Secondly, a random forest (RF) is used to select the top 10 most influential variables for each season for further analysis. A relevance vector machine (RVM) captures complex nonlinear relationships separately for each season, while the wavelet transform (WT) decomposes residuals generated from RVM into different frequency subseries (with reduced noise). These subseries are predicted with minimal bias using AdaBoost with regression and threshold (AdaBoostRT). Finally, we stack RVM, AdaBoostRT, RF, and residual individual predictions using RF as a meta-model to produce the final forecast with minimal error accumulation and efficiency. The comparative study, based on point forecast metrics, the Diebold-Mariano test, and prediction interval widths, shows that the proposed model outperforms vector autoregressive (VAR), RF, AdaBoostRT, RVM, and Naïve models. The study results can be utilised for optimising resource allocation, effective power grid management, and customer alerts.

Keywords: machine learning; forecasting; power outage; load-shedding; South Africa; Eskom

Academic Editors: Renato Procopio, Rodolfo Antônio Ribeiro De Moura and Alice La Fata

Received: 20 August 2025
Revised: 17 September 2025
Accepted: 18 September 2025
Published: 19 September 2025

Citation: Sivhugwana, K.S.; Ranganai, E. Short-Term Forecasting of Unplanned Power Outages Using Machine Learning Algorithms: A Robust Feature Engineering Strategy Against Multicollinearity and Nonlinearity. *Energies* **2025**, *18*, 4994. https://doi.org/10.3390/en18184994

1. Introduction

1.1. Context

In developing regions such as Africa, an adequate and planned electricity supply should be at the core of economic development strategies [1,2]. Since 2008, South Africa has experienced energy demand growth compared to other African regions, leading to power imbalances [3,4]. The work of [5,6] attributes this to several factors. Firstly, there

This page is intentionally blank

Appendix A

A. Algorithms

Algorithm A1. WT-ARIMA-GBDTs-SVR

INPUT: Wind speed time series data $\boldsymbol{y_t}$

A. Data cleaning

1. The original wind speed data from three datasets of interest are cleaned to handle anomalies, such as invalidities and missing data that might occur due to environmental factors or instability of the data collection system. All observed wind speeds greater than 15 m/s are treated as outliers (and are removed if found). Over 15 m/s, the wind turbine's blades spin rapidly, which might cause the turbine to break down; thus, its operation is usually restricted. In some instances, wind turbines are switched off when the velocity exceeds 22 m/s, which is also referred to as feathering.

B. Data partition

2. Each dataset is divided into two sets, namely the training set (80%) and the testing set (20%).
3. In the proposed strategy, the training set is utilised to build the model, while the testing set evaluates each of the established models.

C. Train and predict using the ARIMA model

4. Determine ARIMA orders using the "auto.arima" function in the R program using the training dataset.
5. Predict wind speed data to capture (predict) linear components using the optimal ARIMA model such that the predictions are denoted by $\widehat{\boldsymbol{y}}_{\boldsymbol{\tau}}$.
6. Generate the ARIMA residuals using the entire wind speed dataset such that residuals are calculated by $\boldsymbol{R}_{\boldsymbol{y_t}} = \boldsymbol{y_t} - \widehat{\boldsymbol{y}}_{\boldsymbol{\varepsilon}}$ with $\widehat{\boldsymbol{y}}_{\boldsymbol{\varepsilon}}$ being the fitted values.
7. Check the efficacy of ARIMA predictions based on root mean square error (RMSE) and MAE.

D. Data decomposition

8. The ARIMA residuals (or nonlinear components) are decomposed into less noisy subseries using level 3 and 4 MODWT.
9. Divide decomposed subseries ($\boldsymbol{R}_{\boldsymbol{y_t}}$) into training set (80%) ($\boldsymbol{R}_{\boldsymbol{train}}$) and testing set (20%) ($\boldsymbol{R}_{\boldsymbol{test}}$)

E. Train and predict using the XGBoost model

10. Using a grid search, determine and validate model hyperparameters such as the interaction depth, learning rate, maximum number of trees, and minimum child using the training dataset ($\boldsymbol{R}_{\boldsymbol{train}}$) of the decomposed subseries as input features. The objective is to obtain those parameters that minimise the RMSE and MAE.
11. To capture the nonlinear component, the testing set ($\boldsymbol{R}_{\boldsymbol{test}}$) of the decomposed subseries is utilised as input features into the optimal XGBoost model for prediction.
12. The efficacy of the predictions ($\widehat{\boldsymbol{R}}_{\boldsymbol{test}}$) from the XGBoost model is evaluated using the decomposed subseries based on RMSE and MAE.

F. Combination of predictions via SVR

13. Use a grid search to identify hyperparameters, such as the Cost and Gamma, before the SVR is utilised to combine sub-series predictions.
14. To arrive at the final prediction, the ARIMA and XGBoost predictions are combined through the SVR algorithm to form the WT-ARIMA-XGBoost model such that

$$\widehat{\boldsymbol{y}}_{\boldsymbol{final}} = \boldsymbol{SVR}_{\boldsymbol{rbf}}\left(\boldsymbol{y_t}^{\boldsymbol{T}}, \widehat{\boldsymbol{y}}_{\boldsymbol{\tau}}^{\boldsymbol{T}}, \widehat{\boldsymbol{R}}_{\boldsymbol{test}}\right) \quad (22)$$

G. Final prediction evaluation

15. The efficacy of the final predictions is evaluated using error metrics (MAE, MAPE, R^2, and RMSE) and prediction interval indices (PINAD and PINAW) against the original wind speed testing dataset.

OUTPUT: Predictions $\widehat{\boldsymbol{y}}_{\boldsymbol{final}}$ performance metrics (MAE, MAPE, RMSE, and $\boldsymbol{R}^2$), prediction interval indices (PINAD and PINAW).

Algorithm A.2. WT-NNAR-LSTM-GBM

INPUT: Wind speed data ($\boldsymbol{y_t}$)

1. ***Data preparation***

The wind speed data is checked for invalidities and missing data mainly due to collection system malfunction and adverse environmental conditions. Wind speeds over 22 m/s are considered outliers and removed (if found) as they are not practical for wind power generation.

2. ***Data preprocessing***

The original wind speed data is decomposed using a 3-level MODWT, resulting in three detailed signals (D1, D2, and D3) and one low-frequency approximate signal (A3).

3. ***Data complexity assessment***

In step 3, SampEn is used to estimate the level of randomness or complexity of all four decomposed subseries. Subseries with a SampEn value greater than or equal to 0.9 are considered to be complex (high-variant) while those with a SampEn value of less than 0.9 are considered to be deterministic (less-variant or random).

4. ***Data formatting***

In LSTM, complex subseries are normalised through the MINIMAX technique before they are divided into train and test sets such that they fall within [-1, 1]. In NNAR, the Box-Cox transformation is used to eliminate non-normality from datasets and to ensure roughly homoscedastic residuals.

5. ***Data partition***

In step 5, the data is split into a training set (to build the model) (80%) and a testing set (to test the strength and robustness of the model) (20%). Note that the data split was performed in a way that the structure of the subseries was preserved (no reshuffling) as that would impact model performance.

6. ***Parameter identification and model building***

After splitting the data, the parameters for building an effective and efficient LSTM network are determined. These include batch size, time step, features, neural network layers, network dimensionality, learning rate, activation function, return type, network state, number of epochs, and error function. The "*nnetar*" function in the 'forecast' package is employed to automatically select optimal parameters (*p*,*k*) for the NNAR model.

7. ***Model training***

Using the training data set (80%) as input to the model, the prediction error of the LSTM network is computed and the model's performance is evaluated. Thereafter, the model parameters are fine-tuned according to the results to further improve the predictive power of the model.

8. ***Model testing (predictions and evaluation)***

Both NNAR and LSTM use the best-fitting model to generate predictions and compare them with the corresponding subseries test dataset (20%) based on the performance metrics (RMSE, MAE). Note that the predictions from the LSTM network are first denormalised to restore the original state of the subseries before performing the comparison with the subseries test dataset.

9. ***Prediction ensemble***

i. Training GBM: Before the GBM is employed to combine subseries predictions, the model hyperparameters (number of regression trees, interaction depth, learning rate, and error function) are identified and used to train the model using 80% of the original wind speed data. Prediction error (RMSE) is computed, model parameter settings are fine-tuned, and the prediction strength of the model is improved.

ii. Forecast combination: The resultant GBM model from training is used in computing the final predicted value by combining the predictions for all subseries. Finally, the hybrid predictions are compared to 20% of the original wind speed testing data using performance metrics (MAE, RMSE, MAD, CRPS, and PIW).

OUTPUT: $\hat{\mathbf{y}}_t$, MAE, RMSE, MAD, CRPS, and PIW

Algorithm A3: Wavelet-MODWT-GRU

1. ***Input**: Wind speed data ($\boldsymbol{y_t}$)*

A. Data Cleaning and Preprocessing

2. *data_cleaning_and_preprocessing*
3. Load original wind speed data $\boldsymbol{y}_t \in \mathbb{R}^M$ into R program environment.
4. Clean and format data inconsistencies and anomalies caused by environmental factors and instrument instability.
5. Retain $y_t \in (0, 22\frac{m}{s}]$ as wind turbines resort to feathering beyond this limit and are switched off.
6. Divide data into 80% training ($\boldsymbol{y_t}^{train} \in \mathbb{R}^{M-h}$) and 20% testing sets ($\boldsymbol{y_t}^{test} \in \mathbb{R}^h$) with $M > h$ such that $M \in \mathbb{R}^+, h \in \mathbb{R}^+$.
7. *output*

B. DE hyperparameter search

8. *de_hyperparameter_search*
9. Initialise the wavelet filter.
10. Define the objective function based on the original wind data (y_t) and the reconstructed series ($\widehat{y_t}$) such that the mean sum of square (MSE) error is given by $\text{MSE}_\text{R} = \frac{\sum_{t=1}^{M}(y_t - \widehat{y_t})^2}{M} \in \mathbb{R}^+$. The function is specific to the wavelet filter and is used to evaluate performance.
11. Set parameter bounds within which DE will search. Thus, set population size, number of iterations, crossover probability, parameter bounds, and weights. This is vital for DE to search a relevant interval and to improve search efficiency.
12. Run the DE until the predetermined termination criterion (i.e., number of runs) is reached.
13. *output*

C. Signal denoising and processing

14. *signal_denoising_and_formating*
15. In MODWT, the optimised decomposition level (L) is used alongside the conditions, filters and boundary parameters to decompose the $\mathbf{y}_t \in \mathbb{R}^M$ into less noisy and more statistically sound subsignals $\boldsymbol{\Gamma} = \{\boldsymbol{d}_1, \boldsymbol{d}_2, \dots, , \boldsymbol{d}_L, \boldsymbol{A}_L\}$ with $\boldsymbol{d}_i (i = 1{:}\,L) \in \mathbb{R}^M$ and $\boldsymbol{A}_L \in \mathbb{R}^M$.
16. Each subsignal is divided into two sets, namely; the training set (80%) ($\boldsymbol{\vartheta_t}^{train} \in \mathbb{R}^{M-h}$) and the testing set (20%) ($\boldsymbol{\vartheta_t}^{test} \in \mathbb{R}^h$).
17. Normalise subsignals using the min-max criterion such that ${\vartheta_{t(norm)}}^{train}(t = 1{:}\,M-h) = \frac{{\Gamma_t}^{train} - {\Gamma_{min}}^{train}}{{\Gamma_{max}}^{train} - {\Gamma_{min}}^{train}} \in \mathbb{R}^{M-h}$ and ${\vartheta_{t(norm)}}^{test}(t = (M-h+1){:}\,M) = \frac{{\Gamma_t}^{test} - {\Gamma_{min}}^{test}}{{\Gamma_{max}}^{test} - {\Gamma_{min}}^{test}} \in \mathbb{R}^h$. This ensures that ${\vartheta_{t(norm)}}^{test} \in \mathbb{R}_{\{0,1\}}$ and ${\delta_{t(norm)}}^{train} \in \mathbb{R}_{\{0,1\}}$ are compatible with the hyperbolic tangent function and minimise noise/variance effects on the predictions.
18. *output*

D. GRU hyperparameter search

19. *gru_hyperparameter_search*
20. Array data into a 3D format (i.e., samples, time steps, feature) for compatibility with the GRU network.

21. Initialise parameters: input shape, batch size, dropout rates, epochs, activation function, loss function, learning rate, and optimiser.
22. Train GRU model and evaluate performance based on $MSE = \frac{1}{M-h}\sum_{t=1}^{M-h}({y_t}^{train} - {\hat{y}_t}^{train})^2 \in \mathbb{R}^+$) using the normalised training dataset ${\boldsymbol{\vartheta}_{t(norm)}}^{train} \in \mathbb{R}^{M-h}$.
23. Preserve model parameters with optimal performance.
24. *output*

E. Test GRU performance

25. *test_gru_performance*
26. Superimpose the GRU model with optimal parameters on ${\vartheta_{t(norm)}}^{test}$ to generate normalised forecasts ${\hat{\vartheta}_{t(norm)}}^{test}$.
27. Return to the original subsignal forecast via ${\hat{\Gamma}_t}^{test} = ({\Gamma_{max}}^{test} - {\Gamma_{min}}^{test}) * {\hat{\vartheta}_{t(norm)}}^{test} + {\Gamma_{min}}^{test}$,
where ${\hat{\boldsymbol{\Gamma}}_t}^{test}(t = (M-h+1):M) \in \left({\widehat{\boldsymbol{d}}_1}^{test}, {\widehat{\boldsymbol{d}}_2}^{test}, \cdots, {\widehat{\boldsymbol{d}}_L}^{test}, {\widehat{\boldsymbol{A}}_L}^{test}\right) \in \mathbb{R}^h$ are the original subsignal predictions and ${\widehat{\boldsymbol{\vartheta}}_{t(norm)}}^{test} \in \left({\widehat{\boldsymbol{d}}_{1(norm)}}^{test}, {\widehat{\boldsymbol{d}}_{2(norm)}}^{test}, \cdots, {\widehat{\boldsymbol{d}}_{L(norm)}}^{test}, {\widehat{\boldsymbol{A}}_{L\,norm}}^{test}\right) \in \mathbb{R}^h$ are the normalised subsignal predictions.
28. Evaluate the performance of the GRU predictions for each subsignal using RMSE, MAE, and MAPE.
29. *output*

F. Signal reconstruction and output evaluation

30. *signal_reconstruction_and_output_evaluation*
31. All subsignals predictions are used to reconstruct ${\boldsymbol{y}_t}^{test} \in \mathbb{R}^h$ such that $\widehat{\boldsymbol{y}_t}^{test} = inverse\text{-}MODWT({\widehat{\boldsymbol{d}}_1}^{test}, {\widehat{\boldsymbol{d}}_2}^{test}, \cdots, {\widehat{\boldsymbol{d}}_L}^{test}, {\widehat{\boldsymbol{A}}_L}^{test})$.
32. Use performance metrics and statistical tests (i.e., RMSE, MAE, MAPE, coefficient of R^2, PL, MD, MZ, PIT, and DS) to compare $\widehat{\boldsymbol{y}_t}^{\text{test}} \in \mathbb{R}^h$ and ${\boldsymbol{y}_t}^{test} \in \mathbb{R}^h$.
33. *output*

34. ***Output***: $\widehat{\boldsymbol{y}}_t$, *RMSE*, *MAE*, *MAPE*, R^2, *PL*, *MD*, *MZ*, *PIT*, and *DS*

Algorithm A4: Variable Selection (through LASSO and RF)

1. Load relevant R libraries *(glmnet, caret, randomForest)*

2. Data cleaning

Input: Raw_data $\in \mathbb{R}^{10,223\times42}$,

Check completeness, correctness, consistency, handle structural errors, drop irrelevant features, and create new features.

Output: $\boldsymbol{X}_{new} \in \mathbb{R}^{10,223\times42}$, $\boldsymbol{y}_{new} \in \mathbb{R}^{10,223\times1}$

3. Detect multicollinearity through VIF

Input: $\boldsymbol{X}_{new}$, $\boldsymbol{y}_{new}$

Check for multicollinearity using VIF

Output: $\boldsymbol{X}_{vif} \in \mathbb{R}^{10,223\times42}$, $\boldsymbol{y}_{vif} \in \mathbb{R}^{10,223\times1}$

4. Data division for LASSO training

Input: $\boldsymbol{X}_{vif}, \boldsymbol{y}_{vif}$

Partition data into an 80% training set and a 20% test set

Output: ($\boldsymbol{X}_{train} \in \mathbb{R}^{8178\times42}$, $\boldsymbol{y_{train}} \in \mathbb{R}^{8178\times1}$) ← training set, ($\boldsymbol{X}_{test} \in \mathbb{R}^{2045\times42}$, $\boldsymbol{y}_{test} \in \mathbb{R}^{2045\times1}$) ← test set

5. Variable selection (LASSO)

Input: $\boldsymbol{X}_{train}$, $\boldsymbol{y}_{train}$, α_{lasso}

Through cross validation, find optimal λ and fit LASSO

Retain variables with non-zero coefficients ($\xi_j \neq 0$)

Output: 31 predictors; 1 dependent variable

6. Extract data to represent each season

Input: $\boldsymbol{X}_{selected\_var} \in \mathbb{R}^{10,223\times31}$, $\boldsymbol{y}_{selected\_var} \in \mathbb{R}^{10,223\times1}$

Extract data to represent each season

Output: $\boldsymbol{X}_{season} \in \mathbb{R}^{1464\times31}$, $\boldsymbol{y}_{season} \in \mathbb{R}^{1464\times1}$

7. Select top 10 variable per season (RF)

Input: $\boldsymbol{X}_{season} \in \mathbb{R}^{1464\times31}$, $\boldsymbol{y}_{season} \in \mathbb{R}^{1464\times1}$

Partition data into an 80% training set [60% train +20% val] and a 20% test set; Select the top ten variables for each season through RF

Output: ($\boldsymbol{X}_{train\_retained} \in \mathbb{R}^{1176(=888\ \boldsymbol{train}+\ 288\ val)\times10}$, $\boldsymbol{y}_{train\_retained} \in \mathbb{R}^{1176\ (=888\ train+\ 288\ val)\times1}$) ← training set; ($\boldsymbol{X}_{test\_retained} \in \mathbb{R}^{288\times10}$, $\boldsymbol{y}_{test\_retained} \in \mathbb{R}^{288\times1}$) ← test set; ($\boldsymbol{X}_{retained} \in \mathbb{R}^{1464\times10}$, $\boldsymbol{y}_{retained} \in \mathbb{R}^{1464\times1}$) ← Full retained set

Algorithm A5: RF (through bagging)

1. Load relevant R libraries *(caret, randomForest, ranger)*

2. Train (60%) and validate (20%) using 80% of the retained data

2.1. **Tuning hyperparameters**

Input: $\boldsymbol{X}_{train\_retrained} \in \mathbb{R}^{888 \times 10}$, $\boldsymbol{y}_{train\_retained} \in \mathbb{R}^{888 \times 1}$

Tune RF hyperparameters and find the optimal number of trees (M) and featues (m)

Output: Optimised M, m

For each tree i= 1 to M

a. Bootstrapped sample generation

Input: $\boldsymbol{X}_{train\_retained}$, $\boldsymbol{y}_{train\_retained}$

Draw each sample with a replacement from ($\boldsymbol{X}_{train\_retained}$, $\boldsymbol{y}_{train\_retained}$) to create a bootstrapped sample ($\boldsymbol{X}_{bstraped}$, $\boldsymbol{y}_{bstraped}$) = $\boldsymbol{B}^*$

Output: $\boldsymbol{B}^*$

b. Build decision tree

Input: $\boldsymbol{B}^*$, m

Build a decision tree; randomly select m at each node; grow a decision tree

Output: Decision tree T_i

c. Building forest

Input: T_i

Built forest using all the tress $(T_1, T_2, \dots, T_M)$

Output: $(T_1, T_2, \dots, T_M) \leftarrow$ RF^{optimal}

2.2. **Model validation on the 20% of the data**

Input: RF^{optimal}, $\boldsymbol{X}_{train\_retained(val)}$, $\boldsymbol{y}_{train\_retained(val)}$

Aggregate predictions from all trees such that $\hat{\boldsymbol{y}}_{train\_retained(val)} = \hat{f}\left(\boldsymbol{X}_{train\_retained(val)}\right) = \frac{1}{M}\sum_{i=1}^{M} T_i\ \left(\boldsymbol{X}_{train\_retained(val)}\right) = \hat{f}^{RF\_val} \in \mathbb{R}^{288 \times 1}$

Compute MAE, MAPE, and RMSE between $\hat{\boldsymbol{y}}_{train\_retained(val)}$ and $\boldsymbol{y}_{train\_retained(val)}$

Output: {MAE, MAPE, RMSE} ← Performance metrics

3. Predicting using the test data

Input: RF^{optimal}, $\boldsymbol{X}_{test\_retained}$, $\boldsymbol{y}_{test\_retained}$

Use RF^{optimal} on $\boldsymbol{X}_{test\_retained}$ to predict $\boldsymbol{y}_{test\_retained}$ such that $\hat{f}\left(\boldsymbol{X}_{test\_retained}\right) = \hat{\boldsymbol{y}}_{test\_retained}$

Output: $\hat{\boldsymbol{y}}_{test\_retained} \leftarrow \hat{f}^{RF\_model} \in \mathbb{R}^{288 \times 1}$

4. Model performance assessment using test data

Input: $\boldsymbol{y}_{test\_retained}$, $\hat{f}^{RF\_model}$

Calculate MAE, MAPE, RMSE, PINAW, MZ test, and the DM test

Output: {MAE, MAPE, RMSE, PINAW, MZ test, DM test} ← Performance metrics

5. Final output

Output: $\hat{f}^{RF\_model}$, Performance metrics

Algorithm A6: Wavelet transform (through MODWT)

1.Load relevant R libraries *(waveslim, forecast, caret,kernlab)*

2.Fit the entire retained data

Input: $\boldsymbol{X}_{retained} \in \mathbb{R}^{1464 \times 10}$, $\boldsymbol{y}_{retained} \in \mathbb{R}^{1464 \times 1}$,

Compute $\hat{\boldsymbol{y}}_{fitted} = \hat{f}(\boldsymbol{X}_{retained})$ using RVM model

Output: $\hat{\boldsymbol{y}}_{fitted} \in \mathbb{R}^{1464 \times 1}$

3.Calculate residuals

Input: $\boldsymbol{y}_{retained}$, $\hat{\boldsymbol{y}}_{fitted}$

Compute residuals $\boldsymbol{y}_r = \boldsymbol{y}_{retained} - \hat{\boldsymbol{y}}_{fitted}$

Output: $\boldsymbol{y}_r \in \mathbb{R}^{1464 \times 1}$

4.Set wavelet parameters

Input: $\boldsymbol{y}_r$, db4 ← wavelet_filter, 2 ← n_level, periodic ← boundary

Output: $\boldsymbol{y}_r$, wavelet_filter, n_level, boundary

5.Perform wavelet decomposition using MODWT

Input: $\boldsymbol{y}_r$, wavelet_filter, n_level, boundary

Decompose $\boldsymbol{y}_r$ into detailed and approximate signals

Output: $\boldsymbol{A_2} \in \mathbb{R}^{1464 \times 1} \leftarrow \boldsymbol{A}$ (Approximate subseries);

$(\boldsymbol{D_1}, \boldsymbol{D_2}) \in \mathbb{R}^{1464 \times 2} \leftarrow \boldsymbol{D}$ (Detailed subseries)

Algorithm A7: AdaBoostRT (through boosting)

1. Load relevant R libraries *(ReBoost)*

2. Initialise parameters

$\tau_i = \left(\frac{1}{n}\right) \in \mathbb{R}$ ← weights, $V \in \mathbb{R}$ ← number of weak learners, $\delta \leq 0.38 \in \mathbb{R}$ ← error threshold

Output: Initialised τ_i, V, δ

3. Train (60%) and validate (20%) using 80% of the retained data

3.1. **Tuning hyperparameters**

For each i = 1 to V

a. Training weak leaner

Input: $\boldsymbol{X}_{train\_retained}$, $\boldsymbol{y}_{train\_retained}$, τ_i

Fit a weak learner q_i to the weighted $\boldsymbol{X}_{train\_retained}$

Predict for $\boldsymbol{y}_{train\_retained}$

Output: $q_i\ (\boldsymbol{X}_{train\_retained})$

b. Calculate error function

Input: $\boldsymbol{y}_{train\_retained}$, $q_i\ (\boldsymbol{X}_{train\_retained})$

Compare $q_i(\boldsymbol{X}_{train\_retained})$ with $\boldsymbol{y}_{train\_retained}$:

If error $\rho_i < \delta$, correct; otherwise, incorrect

Output: $\boldsymbol{\rho_i}$ (incorrectly classified)

c. Update weights

Input: $\boldsymbol{y}_{train\_retained}$, $q_i(\boldsymbol{X}_{train\_retained})$, τ_i, ρ_i

Increase incorrectly classified weights

Normalise the updated weights

Output: τ_{i+1}

3.2. **Calculate model weights**

Input: ρ_i

Compute model weight ψ_i

Output: ψ_i (for the i^{th} weak learner)

3.3. **Preserve trees and weights**

Input: Weak trees $\{q_1, q_2, \ldots, q_V\}$, weights $\{\psi_1,\psi_2,...,\psi_V\}$

Store $\{q_1, q_2, \ldots, q_V\} \leftarrow \boldsymbol{q}$ and $\{\psi_1,\psi_2,...,\psi_V\} \leftarrow \boldsymbol{\psi}$

Output: $(\boldsymbol{q}, \boldsymbol{\psi}) \leftarrow AdaBoostRT^{optimal}$

3.4. **Model validated on the 20% of the data**

Input: $AdaBoostRT^{optimal}$, $\boldsymbol{X}_{train\_retained(val)}$, $\boldsymbol{y}_{train\_retained(val)}$

Weighted ensemble of the predictions from all trees such that $\hat{f}\left(\boldsymbol{X}_{train\_retained(val)}\right) = \sum_{i=1}^{V} \psi_i\, q_i\left(\boldsymbol{X}_{train\_retained(val)}\right) = \hat{\boldsymbol{y}}_{train\_retained(val)} = \hat{f}^{AdaBoostRT\_val} \in \mathbb{R}^{288\times 1}$

Compute MAE, MAPE, and RMSE

Output: {MAE, MAPE, RMSE} ← Performance metrics

4. Predicting using test data

Input: $AdaBoostRT^{optimal}$, $\boldsymbol{X}_{test\_retained} \in \mathbb{R}^{288\times 10}$, $\boldsymbol{y}_{test\_retained} \in \mathbb{R}^{288\times 1}$

Compute $\hat{f}\left(\boldsymbol{X}_{test\_retained}\right) = \sum_{i=1}^{V} \psi_i\, q_i\left(\boldsymbol{X}_{test\_retained}\right) = \hat{\boldsymbol{y}}_{test\_retained}$

Output: $\hat{\boldsymbol{y}}_{test\_retained} \leftarrow \hat{f}^{AdaBoostRT} \in \mathbb{R}^{288\times 1}$

5. Model performance assessment using test data

Input: $\boldsymbol{y}_{test\_retained}$, $\hat{f}^{AdaBoostRT}$

Calculate MAE, MAPE, RMSE, PINAW, MZ test, and DM test

Output: {MAE, MAPE, RMSE, PINAW, MZ test, DM test } ← Performance metrics

6. Final output

Output: $\hat{f}^{AdaBoostRT}$, Performance metrics

Algorithm A8: RVM (through Bayesian framework)

1. Load relevant R libraries (*kernlab*, *caret*)

2. Initialise Parameters

Set $\boldsymbol{\pi} \in \mathbb{R}^{888\times1}$(precision weights), $\boldsymbol{w} \in \mathbb{R}^{888\times1}$(weights), $\beta^{-1} = \sigma^2 \in \mathbb{R}$ (noise term in regression), $\omega_0 \in \mathbb{R}$ (bias term)

Output: $\boldsymbol{\pi}$, σ^2, $\boldsymbol{w}$, ω_0

3. Train (60%) and validate (20%) using 80% of the retained data

3.1. **Tuning hyperparameters**

a. Choose a basis and transform the data

Input: $\boldsymbol{X}_{train\_retained} \in \mathbb{R}^{888\times10}$, $\boldsymbol{y}_{train\_retained} \in \mathbb{R}^{888\times1}$

Define a basis function such that $K\,(\boldsymbol{X}_{train\_retained}) \in \mathbb{R}^{888\times888}$

Output: $K\,(\boldsymbol{X}_{train\_retained})$

b. Fit the model on training data

Input: $\boldsymbol{\pi}$, $\boldsymbol{w}$, ω_0, $\boldsymbol{X}_{train\_retained}$, $K\,(\boldsymbol{X}_{train\_retained})$

Compute $\hat{f}(\boldsymbol{X}_{train\_retained}) = \sum_{i=1}^{888} w_i K(\boldsymbol{X}_{train\_retained}, \boldsymbol{x}_i) + \omega_0$

Output: $\hat{f}(\boldsymbol{X}_{train\_retained})$

c. Update hyperparameters

Input: $\hat{f}(\boldsymbol{X}_{train\_retained})$, $\boldsymbol{y}_{train\_retained}$, ω_0

Through marginal likelihood, optimise $\boldsymbol{w}$, update both σ^2 and $\boldsymbol{\pi}$

Using bias ω_0, prune excessive weights, and remove non-relevant vectors

Adjust precision to ∞

Output: updated $\boldsymbol{\pi}, \sigma^2, \boldsymbol{w}, \omega_0$

d. Check for convergence

Input: $\epsilon = \sum_{i=1} w_i^{n+1} - w_i^n < \epsilon_{Thresh} \leftarrow$ threshold (based on weights)

While convergence is not achieved, fit the model and update hyperparameters (repeat from step 3)

If convergence is achieved, stop the process.

Output: Convergence decision (True or False), optimised parameters $(\boldsymbol{\pi}, \sigma^2, \boldsymbol{w}, \omega_0) \leftarrow RVM^{optimal}$

3.2. **Model validation on the 20% of the data**

Input: $\boldsymbol{X}_{train\_retained(val)}$, $RVM^{optimal}$

Use $RVM^{optimal}$ to fit $\hat{f}(\boldsymbol{X}_{train\_retained(val)}) = \hat{\boldsymbol{y}}_{train\_retained(val)} = \hat{f}^{RVM\_val} \in \mathbb{R}^{288\times1}$

Compute MAE, MAPE, and RMSE

Output: $\hat{\boldsymbol{y}}_{train\_retained(val)}$, {MAE, MAPE, RMSE} $\leftarrow$ Performance metrics

4. Predicting using test data

Input: $RVM^{optimal}$, $\boldsymbol{X}_{test\_retained} \in \mathbb{R}^{288\times10}$, $\boldsymbol{y}_{test\_retained} \in \mathbb{R}^{288\times1}$

Compute $\hat{\boldsymbol{y}}_{test\_retained} = \hat{f}(\boldsymbol{X}_{test\_retained}) = \sum_{i=1}^{288} w_i K(\boldsymbol{X}_{test\_retained}, \boldsymbol{x}_i) + \omega_0$ using $RVM^{optimal}$

Output: $\hat{\boldsymbol{y}}_{test\_retained} \in \mathbb{R}^{288\times1}$

5. Evaluate model performance on test data

Input: $\boldsymbol{y}_{test\_retained}$, $\hat{\boldsymbol{y}}_{test\_retained}$

Compute MAE, MAPE, RMSE, PINAW, MZ test, DM test

Output: $\hat{\boldsymbol{y}}_{test\_retained} \leftarrow \hat{f}^{RVM\_model}$, {MAE, MAPE, RMSE, PINAW, MZ test, DM test } $\leftarrow$ Performance metrics

6. Final output

Output: $\hat{f}^{RVM\_model}$, Performance metrics

B. Some Selected R Codes

```
#=====================================================================
TYPICAL STATELESS LSTM
#=====================================================================
#==============================Load libraries=========================
library(tidyverse)
library(vip)
library(dplyr)
library(tensorflow)
library(tsibble)
library(lubridate)
library(fable)
library(forecast)
library(h2o)
library(thief)
library(keras)
library(waveslim)
library(forecast)
library(gbm)
library(caret)
library(moments)
library (reticulate)
library(conflicted)
library(tsfknn)
.
.
.

#============================d2 forecasts=============================
#Lagging d2
d2.frame <- data.frame(x = d2)
d2.lag <- d2.frame %>% mutate(xlag = dplyr::lag(x, 1))
d2.lag[is.na(d2.lag)] <- 0
head(d2.lag)

#Data split d2
N = nrow(d2.lag)
n = base::round(N *0.8, digits = 0)
train.d2.lag = d2.lag[1:n, ]
test.d2.lag  = d2.lag[(n+1):N,  ]
head(train.d2.lag)
head(test.d2.lag)

#Data normalisation
data.norm.func <- function(train.d2.lag, test.d2.lag) {
  # Normalize x
  min.x <- min(train.d2.lag$x)
  max.x <- max(train.d2.lag$x)
  train.norm.x <- (train.d2.lag$x - min.x) / (max.x - min.x)
  test.norm.x <- (test.d2.lag$x - min.x) / (max.x - min.x)
```

```
  # Normalise xlag
  min.xlag <- min(train.d2.lag$xlag)
  max.xlag <- max(train.d2.lag$xlag)
  train.norm.xlag <- (train.d2.lag$xlag - min.xlag) / (max.xlag -
min.xlag)
  test.norm.xlag <- (test.d2.lag$xlag - min.xlag) / (max.xlag -
min.xlag)

  list(list(x = train.norm.x, xlag = train.norm.xlag),
       list(x = test.norm.x, xlag = test.norm.xlag),
       c(min.x, max.x, min.xlag, max.xlag))
}

# Normalise the data
norm.d2 <- data.norm.func(train.d2.lag, test.d2.lag)

train.x <- norm.d2[[1]]$x
train.lagx <- norm.d2[[1]]$xlag
test.x <- norm.d2[[2]]$x
test.lagx <- norm.d2[[2]]$xlag

# Reshaping the data
xtrain.shape.x <- array(train.x, dim = c(length(train.x), 1, 1))
xtrain.shape.lagx <- array(train.lagx, dim = c(length(train.lagx), 1,
1))

# Check dimensions
dim(xtrain.shape.x)
dim(xtrain.shape.lagx)

# Model training
lstm.model <- keras_model_sequential() %>%
layer_lstm(units = 1, batch_input_shape = c(1, 1, 1),activation =
"tanh", stateful = FALSE,  return_sequences = TRUE, dropout = 0.1) %>%
layer_dense(units = 1, activation = "tanh")

lstm.model %>% compile(
  loss = 'mean_squared_error',
  optimizer = optimizer_adam(learning_rate = 0.01),
  metrics = c('accuracy')
)

# Training the model
conflicts_prefer(base::`&&`)
lstm.model %>%
  keras::fit(xtrain.shape.lagx, xtrain.shape.x,
             epochs = 30,
             batch_size = 1,
             verbose = 1,
             shuffle = FALSE)

# Make forecast
xtest.shape.x <- array(test.lagx, dim = c(length(test.lagx), 1, 1))
norm.d2.fc <- lstm.model %>% predict(xtest.shape.x, batch_size = 1)
```

```
str(norm.d2.fc)

# Denormalisation function
denorm.forecast <- function(forecast, minmax) {
  min.x <- minmax[1]
  max.x <- minmax[2]

  # Reverse the min-max scaling
  denorm.y <- min.x + forecast * (max.x - min.x)

  return(denorm.y)
}

# Apply denormalisation
fc.d2 <- denorm.forecast(norm.d2.fc, norm.d2[[3]])

# Check accuracy
d2<-as.matrix(d2)
N = nrow(d2)
n = base::round(N *0.8, digits = 0)
train.d2= d2[1:n, ]
test.d2 = d2[(n+1):N,  ]
accuracy(ts(fc.d2), ts(test.d2))

# Result plot
par(mfrow=c(1, 1))
plot(ts(test.d2),type = "l", lwd=2, ylim=range (-1, 1), col='black',
xlab="Observation number", ylab=" Wind speed(m/s)", main="CSIR")
lines(ts(fc.d2),type = "l", lwd=2, col="blue")
legend ("topright", legend=c("Actuals", "M1"),
        fill=c("black","blue"), col = 1:1,  adj = c(0, 0.2), cex=0.85)
grid()
#=================================================================
```

```
#=====================================================================
SAMPLE ENTROPY
#=====================================================================
#==============================Load libraries=========================
library(pracma)

sample_entropy(d1, edim = 2, r = 0.2*sd(d1), tau = 1)
sample_entropy(d2, edim = 2, r = 0.2*sd(d2), tau = 1)
sample_entropy(d3, edim = 2, r = 0.2*sd(d3), tau = 1)
sample_entropy(a3, edim = 2, r = 0.2*sd(a3), tau = 1)
#=====================================================================
```

```
#=========================================================================
STANDARD DE ALGORITHM
#=========================================================================
#================================Load library=============================
library(forecast)
library(waveslim)
library(DEoptim)
library(wavelets)


#================================Load data================================
data.alex<-read.csv("Augustwm01.csv")
data.alex<-data.alex$WS_60_mean

#========================== data split ===================================

train.size <- round(0.8 * length(data.alex))
data.train <- data.alex[1:train.size]
data.test <- data.alex[(train.size + 1):length(data.alex)]

#============================DE Optimasation==============================

# Objective function
obj.function <- function(dlevel) {
  L <- as.integer(round(dlevel[1]))
  wt <- wavelets::modwt(data, filter = "la8", n.levels = L)
  forecast <- wavelets::imodwt(wt)
  mse <- mean((data - forecast)^2)
  return(mse)
}

# For reproducibility
set.seed(100)
data <- ts(data.train)

lower_bound <- 1
upper_bound <- 10

bounds <- matrix(c(lower_bound, upper_bound), nrow = 1, byrow = TRUE)

# Function to run DE and return the best level
DE.result <- function() {
  result <- DEoptim(
    fn = obj.function,
    lower = bounds[1, 1],
    upper = bounds[1, 2],
    DEoptim.control( itermax = 50,  NP = 20, CR = 0.7, F = 0.5)
  )
  return(round(result$optim$bestmem))  level
}

# The DE n=30 times
Decom.L <- replicate(30, DE.result())

# Count the frequency of each decomposition level
counts <- table(Decom.L)
counts
```

```
#=====================================================================
TYPICAL WAVELET- MODWT-GRU (MB)
#=====================================================================
library(tidyverse)
library(tsibble)
library(lubridate)
library(fable)
library(forecast)
library(keras)
library(tensorflow)
library(waveslim)
library(wavelets)
library(caret)
.
.
.
#======================Load data==================================
tensorflow::set_random_seed(100)
April.data.1<-read.csv("aprilwm08.csv")
str(April.data.1)
#============================= MODWT decomposition=================

April.data.wt<- waveslim::modwt(April.data.1, wf="mb8", n.levels = 3,
boundary = "periodic")
str (April.data.wt)

y1<- April.data.wt

d1.r<-y1$d1
plot(d1.r, type="l")
d1=ts(d1.r, frequency=1,start=c(1,1))

d2.r<-y1$d2
plot(d2.r, type="l")
d2=ts(d2.r, frequency=1,start=c(1,1))

d3.r<-y1$d3
plot(d3.r, type="l")
d3=ts(d3.r, frequency=1,start=c(1,1))

s3.r<-y1$s3
plot(s3.r, type="l")
a3=ts(s3.r, frequency=1,start=c(1,1))

#==========================d1 forecasts============================

tensorflow::set_random_seed(100)
data <- ts(d1)
data.length <- length(data)

# Split the data into training and testing sets
```

```
train.length <- round(0.8 * length(data))
train.data <- data[1:train.length]
test.data <- data[(train.length + 1):length(data)]

# Normalise the data
X.train.min <- min(train.data)
X.train.max <- max(train.data)
X.train.norm <- (train.data - X.train.min) / (X.train.max -
X.train.min)
X.test.norm <- (test.data - X.train.min) / (X.train.max - X.train.min)

# Create sequences for GRU input
seq.f <- function(data, time_steps, lead_time = 1) {
  X.append <- NULL
  y.append <- NULL
  max.idx <- length(data) - lead_time
  for (i in seq(time_steps, max.idx)) {
    X.append <- rbind(X.append, data[(i - time_steps + 1):i])
    y.append <- c(y.append, data[i + lead_time])
  }
  return(list(X.append = X.append, y.append = y.append))
}

# Prepare sequences for training and testing
X.train_seq <- seq.f(X.train.norm, time_steps, lead_time)
y.test_seq <- seq.f(X.test.norm, time_steps, lead_time)


# Reshape data (samples, time steps, features)
X.train.s <- array(X.train_seq$X.append, dim =
c(nrow(X.train_seq$X.append), time_steps, 1))
X.test.s <- array(y.test_seq$X.append, dim =
c(nrow(y.test_seq$X.append), time_steps, 1))

model <- keras_model_sequential() %>%
  layer_gru(units = 64, input_shape = c(time_steps, 1), activation =
"tanh", return_sequences = TRUE) %>%
  layer_dropout(rate = 0.3) %>%
  layer_gru(units = 64, activation = "tanh") %>%
  layer_dropout(rate = 0.3) %>%
  layer_dense(units = 1)

# Compile the model
model %>% compile(
  optimizer = optimizer_adam(learning_rate = 0.01),
  loss = 'mean_squared_error'
)

y.train.s <- X.train_seq$y.append
y.test.s <- y.test_seq$y.append
```

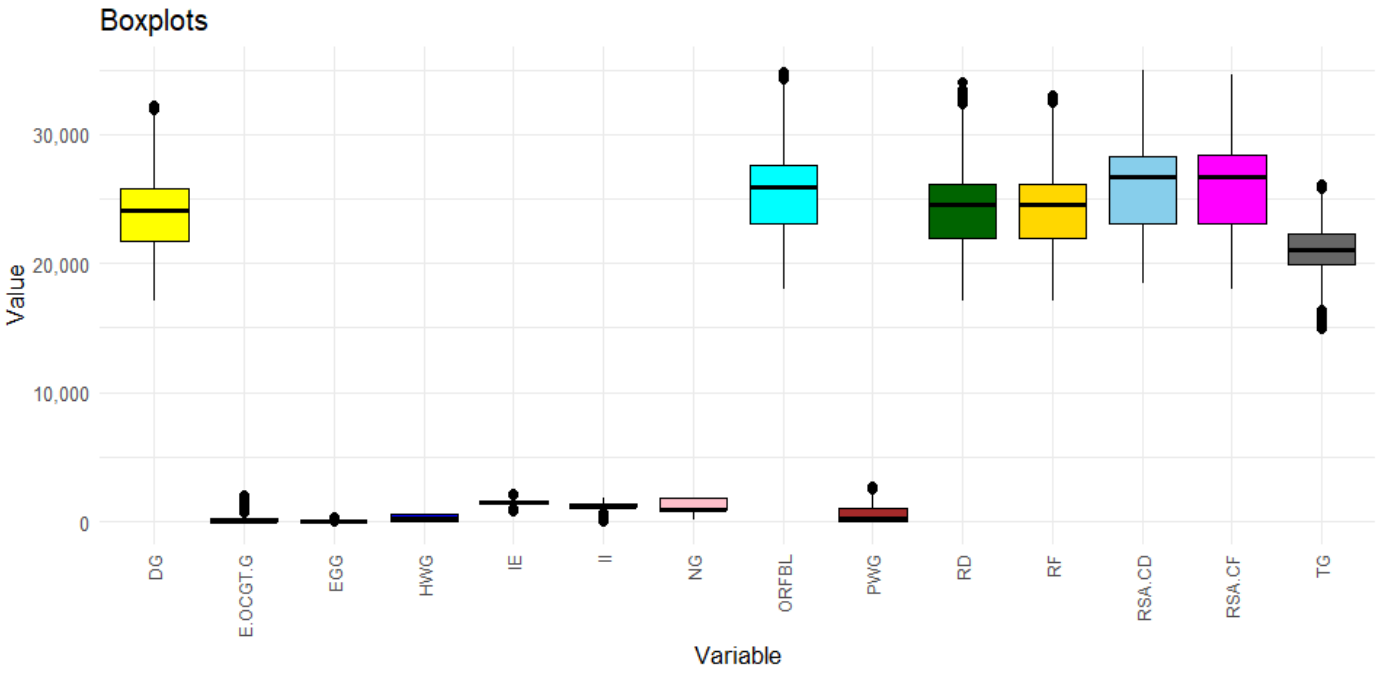
```
# Train the GRU model
history <- model %>% fit(
  X.train.s, y.train.s,
  epochs = 50,
  batch_size = 32,
  validation_split = 0.1,
  callbacks = list(callback_early_stopping(monitor = "val_loss",
patience = 10, restore_best_weights = TRUE)),
  verbose = 1, shuffle = FALSE
)

# Make predictions
Pred.norm.d1 <- model %>% predict(X.test.s)

# Rescale predictions back to original scale
fc.dedorm.d1 <- ts(Pred.norm.d1 * (train.max - train.min) + train.min)
y.test <- ts(y.test.s * (train.max - train.min) + train.min)

# Evaluate predictions
forecast::accuracy (y.test, fc.dedorm.d1)

# Plot actual values against predicted
plot(y.test, type = 'l', col = 'black', lwd = 2, main = "GRU
forecasts", ylab = "Wind Speed")
lines(fc.dedorm.d1, col = 'blue', lwd = 2, lty = 2)
legend("topright", legend = c("Actual", "Predicted"), col = c("black",
"blue"), lty = 1:2, lwd = 2)

#=========================d2 forecasts==============================
tensorflow::set_random_seed(100)
data <- ts(d2)
data.length <- length(data)
.
.
. Repeat steps
.
.
.

#=========================d3 forecasts==============================
tensorflow::set_random_seed(100)
data <- ts(d3)
data.length <- length(data)
.
.
.
.Repeat steps
.
.
```

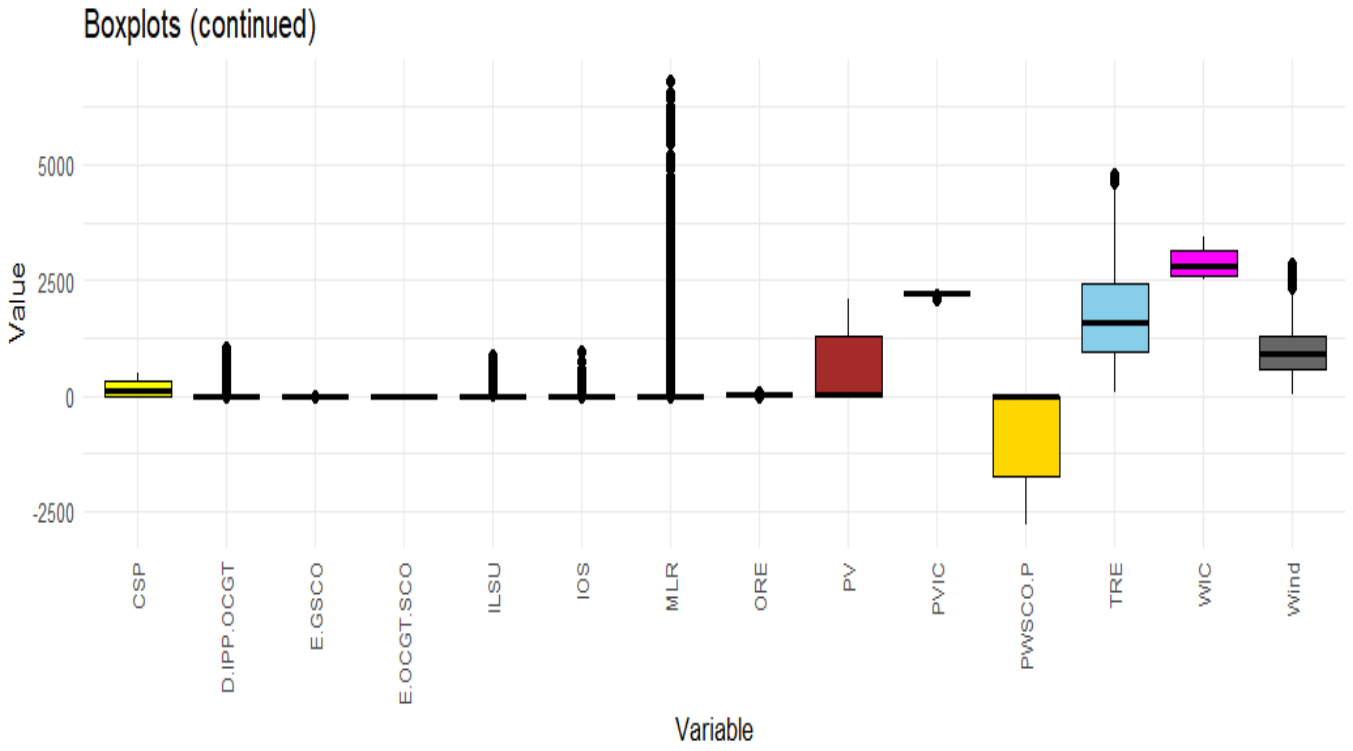
```
.
#=========================a3 forecasts==============================
tensorflow::set_random_seed(100)
data <- ts(a3)
data.length <- length(data)

# Split the data into training and testing sets
train.length <- round(0.8 * length(data))
train.data <- data[1:train.length]
test.data <- data[(train.length + 1):length(data)]

# Normalise the data
X.train.min <- min(train.data)
X.train.max <- max(train.data)
X.train.norm <- (train.data - X.train.min) / (X.train.max -
X.train.min)
X.test.norm <- (test.data - X.train.min) / (X.train.max - X.train.min)

# Create sequences for GRU input
seq.f <- function(data, time_steps, lead_time = 1) {
  X.append <- NULL
  y.append <- NULL
  max.idx <- length(data) - lead_time
  for (i in seq(time_steps, max.idx)) {
    X.append <- rbind(X.append, data[(i - time_steps + 1):i])
    y.append <- c(y.append, data[i + lead_time])
  }
  return(list(X.append = X.append, y.append = y.append))
}

# Prepare sequences for training and testing
X.train_seq <- seq.f(X.train.norm, time_steps, lead_time)
y.test_seq <- seq.f(X.test.norm, time_steps, lead_time)

# Reshape data (samples, time steps, features)
X.train.s <- array(X.train_seq$X.append, dim =
c(nrow(X.train_seq$X.append), time_steps, 1))
X.test.s <- array(y.test_seq$X.append, dim =
c(nrow(y.test_seq$X.append), time_steps, 1))

model <- keras_model_sequential() %>%
  layer_gru(units = 64, input_shape = c(time_steps, 1), activation =
"tanh", return_sequences = TRUE) %>%
  layer_dropout(rate = 0.3) %>%
  layer_gru(units = 64) %>%
  layer_dropout(rate = 0.3) %>%
  layer_dense(units = 1)

# Compile the model
model %>% compile(
```

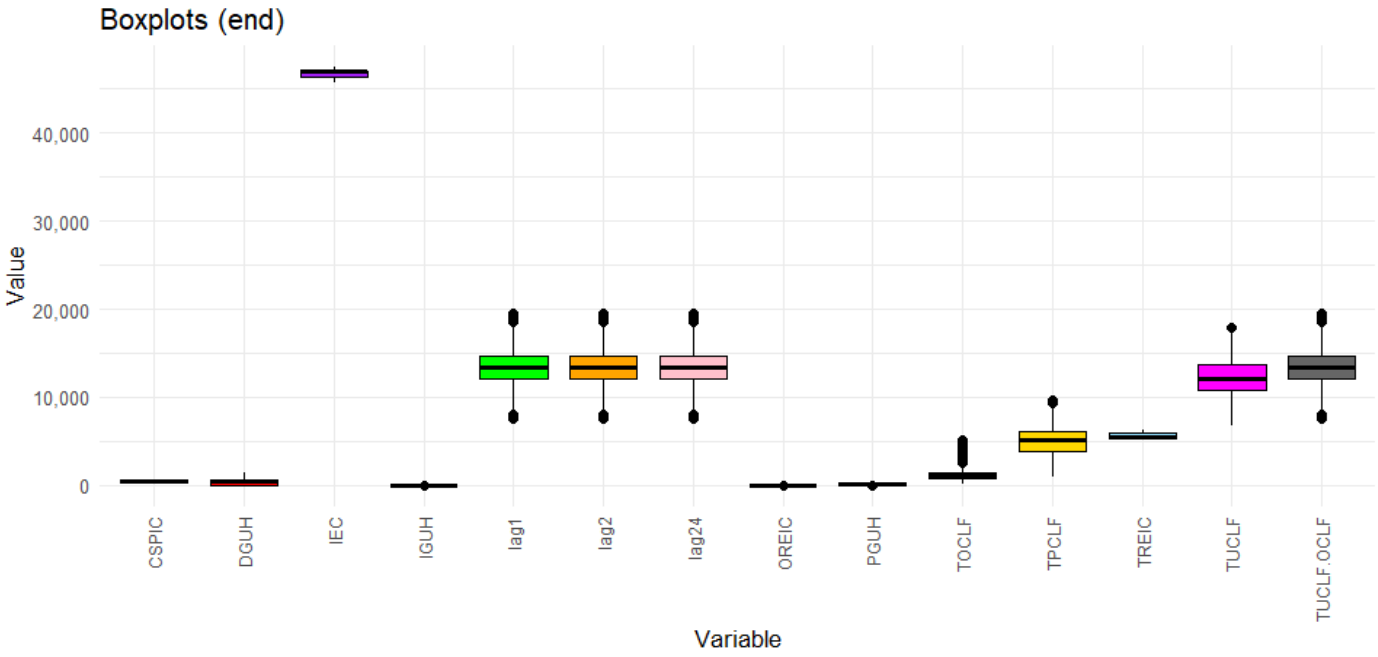
```
  optimizer = optimizer_adam(learning_rate = 0.01),
  loss = 'mean_squared_error'
)

y.train.s <- X.train_seq$y.append
y.test.s <- y.test_seq$y.append

# Train GRU
history <- model %>% fit(
  X.train.s, y.train.s,
  epochs = 30,
  batch_size = 32,
  validation_split = 0.1,
  callbacks = list(callback_early_stopping(monitor = "val_loss",
patience = 10, restore_best_weights = TRUE)),
  verbose = 1,
  shuffle = FALSE
)

# Make predictions
fc.norm.a3 <- model %>% predict(X.test.s)

# Demormalise predictions back to original scale
fc.gru.a3 <- fc.norm.a3 * (X.train.max - X.train.min) + X.train.min
y.test <- y.test.s * (X.train.max - X.train.min) + X.train.min
y.test<-ts(y.test)
fc.dedorm.a3<-ts(fc.gru.a3)

# Evaluate predictions
forecast::accuracy (y.test, fc.dedorm.a3)

# Plot actual values against predicted
plot(y.test, type = 'l', col = 'black', lwd = 2, main = "GRU
forecasts", ylab = "Wind Speed")
lines(fc.dedorm.a3, col = 'blue', lwd = 2, lty = 2)
legend("topright", legend = c("Actual", "Predicted"), col = c("black",
"blue"), lty = 1:2, lwd = 2)


#============Subseries forecasts reconstruction=======================
# Forecast values for each wavelet subseries
d1.fc <- fc.dedorm.d1
d2.fc <- fc.dedorm.d2
d3.fc <- fc.dedorm.d3
a3.fc <- fc.dedorm.a3

# Perform MODWT for mb8
modwt_MB<- waveslim::modwt(test.data.1, wf = "mb8", n.levels = 3,
boundary = "periodic")
str(modwt)
```

```
# Reconstruct the signal using inverse MODWT

fc.original.data.mb8 <- waveslim::imodwt(modwt_MB)

#=====================================================================
```

C. Boxplots of Power Grid Variables






Boxplots (end)
Value
40,000
30,000
20,000
10,000
0
CSPIC
DGUH
IEC
IGUH
lag1
lag2
lag24
OREIC
PGUH
TOCLF
TPCLF
TREIC
TUCLF
TUCLF.OCLF
Variable

D. Variable Definitions

Available Dispatchable Capacity (Incl Non-Comm Units)—The capacity that is available from all dispatchable generation resources, and includes non-commercial generation, as it is dispatchable energy available to support the system.

CSP—Total contracted Concentrated Solar Power generation.

Dispatchable IPP OCGT—OCGT plant that is owned by an IPP and is dispatched by Eskom National Control.

Gen Unit Hours—The number of hours that one unit at pump storage stations can generate based on the amount of water still available in the dams or the number of hours that one unit at an OCGT power station can generate based on the fuel available at that power station.

GW—Gigawatt = 1000 megawatts.

GWh—Gigawatt-hour = 1000 MWh.

Hydro Generation—Generation from large hydropower stations, and sent out onto the Transmission network.

ILS—Interruptible Load Shed. This is consumer load(s) that can be contractually interrupted without notice or reduced by remote control or on instruction from Eskom National Control. Individual contracts place limitations on usage.

International Exports—Energy that is exported from RSA to neighbouring countries.

International Imports—Energy that is imported into RSA from neighbouring countries.

IOS—Interruption of Supply. It is all contracted as well as mandatory demand reduction resources utilised by Eskom National Control. This includes interruption of supply due to Transmission network faults.

IPP—Independent Power Producers that Eskom has contracts with.

kWh—Kilowatt-hour = 1000 watt-hours.

Load Factor—The ratio of the energy generated over a specific time versus the maximum generating capability over the same period.

MLR—Manual Load Reduction. It is an estimation of the demand that has been reduced due to load shedding and/or curtailment.

MW—Megawatt = 1 million watts.

MWh—Megawatt-hour = 1000 kWh.

Non-Dispatchable Conventional IPP—IPP that uses conventional fuel sources to generate energy. These IPPs are contracted with Eskom but not dispatched by Eskom National Control.

Nuclear Generation—Generation from nuclear power stations, and sent out onto the Transmission network.

OCGT—Open Cycle Gas Turbine. Generation from open cycle gas turbine power stations, and sent out onto the Transmission network. These power stations use diesel as their primary resource.

OCLF—Other Capability Loss Factor of Eskom plant. It is the ratio between the unavailable energy of the units that cannot be dispatched, due to constraints out of the power station management control, over a period compared to the total net installed capacity of all units over the same period.

Other RE—Generation from other smaller contracted renewables (small hydro, biomass, landfill gas, etc.).

PCLF—Planned Capability Loss Factor of Eskom plant. It is the ratio between the unavailable energy of the units that are out on planned maintenance over a period compared to the total net installed capacity of all units over the same period.

Pumped Water Generation—Generation from pumped storage power stations, and sent out onto the Transmission network.

Pumping—During off-peak periods and when the system allows, water is pumped from the bottom dams at pumped storage stations to the top dams so that this water is available to generate again. During this process, energy is used from the Transmission network.

PV—Total contracted Photovoltaic generation.

Residual Demand—The hourly average demand that needs to be supplied by all resources that can be dispatched by Eskom National Control. It includes Eskom generation, international imports, dispatchable IPPs and IOS. Normally expressed in MW.

Residual Energy—The total residual demand that is summated over a period of time. Normally expressed in MWh or GWh.

Residual Forecast—The forecast of what the expected residual demand will be in the future.

RSA Contracted Demand—The hourly average demand that needs to be supplied by all resources that Eskom has contracts with. It is the residual demand including demand supplied by self-dispatched generation (such as the renewables).

RSA Contracted Energy—The total RSA contracted demand that is summated over a period of time. Normally expressed in MWh or GWh.

RSA Contracted Forecast—The forecast of what the expected RSA contracted demand will be in the future.

SCO—Synchronous Condenser Operation. The energy used (MW per hour) to overcome the frictional losses when the plant is used to assist in stabilizing the network by supplying or absorbing reactive power.

Thermal Generation—Generation from coal-fired power stations, and sent out onto the Transmission network.

Total Available Capacity (Incl Non-Comm Units and Renewables)—The capacity that is available from all generation resources that Eskom has contracts with, and includes non-commercial generation, as it is energy available to support the system.

UCLF—Unplanned Capability Loss Factor of Eskom plant. It is the ratio between the unavailable energy of the units that are out on unplanned outages over a period compared to the total net installed capacity of all units over the same period.

Wind—Total contracted Wind generation.