SHORT-TERM WIND SPEED AND POWER FORECASTING: A COMPREHENSIVE CASE STUDY FOR THREE OPERATIONAL WIND FARMS

A Thesis Submitted to the
Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Energy Engineering

by
İrem Selen YOLDAŞ

December 2022

İZMİR

ACKNOWLEDGEMENT

Firstly, I would like to express my deepest gratitude to my advisor Asst. Prof. Dr. Ferhat BİNGÖL for his inspiring scientific point of view, guidance, mentorship, support, kindness, tolerance, patience, and trust. I am very grateful to him for allowing me to be a part of project number 120N498 TUBITAK as a scholar, which is also an ERA-NET Smart Energy System project.

I would like to thank Faruk Tuna and Gülsün Gizem Taylan for their endless support, contribution, and belief in me.

Also, I would like to express my most significant appreciation to my mother, Hülya YOLDAŞ, my father, Feridun YOLDAŞ, and my brother Mert Berkin YOLDAŞ for their endless support and belief, encouragement, and self-sacrifices up throughout my entire life.

ABSTRACT

SHORT-TERM WIND SPEED AND POWER FORECASTING: A COMPREHENSIVE CASE STUDY FOR THREE OPERATIONAL WIND FARMS

Wind energy is gradually growing with the increasing energy demand. However, the rising wind power penetration into modern grids could seriously affect the safe operation of power systems and power quality due to the intermittence and randomness of wind characteristics. Several effective ways could be considered to mitigate these issues: a robust power grid, energy storage, and wind power forecasting. Optimal integration of wind energy into power systems calls for high-quality wind power predictions. This research focuses on the short-term forecast of wind speed and power generation. Firstly, wind speed forecasting is studied. A case study is performed to analyze the forecasting performance of five approaches: the multivariate Facebook Prophet, seasonal autoregressive integrated with moving average (SARIMA), SARIMA with exogenous variable (SARIMAX), gated recurrent units (GRU) and long short-term memory (LSTM). The performance indicators are applied to verify the effectiveness of models, which are R-square (R^2), mean square error (MSE), root mean square error (RMSE), and mean absolute error (MAE). The predictions obtained by the LSTM model almost coincide with the real-time wind speed, which is also supported by the performance indicators, which indicate that the LSTM model outperforms the other methods for the real-time dataset of IZTECH meteorological mast. The second part of the study is to forecast the wind power generation using the LSTM model and the wind speed forecasts and wind speed power curve of wind turbines in the wind farms. The proposed model is validated using the real-time wind power generation data from the EPIAS Transparency Platform. Due to the unavailable meteorological dataset, an ERA5 dataset of the location is used to predict wind speed and power generation. Also, each wind farm's daily forecasts are obtained to investigate the results for Day-ahead Market. The results indicate that using the LSTM model with the ERA5 dataset could give better forecasts than wind farms’ own forecasts. Additionally, it is understood that if the SCADA data could be obtained, the forecasting performance might be increased.

ÖZET

KISA DÖNEM RÜZGÂR HIZI VE GÜÇ TAHMİNİ: ÜÇ OPERASYONEL RÜZGÂR TARLASI İÇİN KAPSAMLI BİR VAKA ÇALIŞMASI

Rüzgâr enerjisi, artan enerji talebi ile giderek büyümektedir. Bununla birlikte, modern şebekelerde artan rüzgâr enerjisi entegrasyonu, rüzgarın doğası gereği aralıklı ve rastgele olması nedeniyle güç sistemlerinin güvenli çalışmasını ve güç kalitesini ciddi şekilde etkileyebilir. Rüzgâr enerjisinin güç sistemlerine optimum entegrasyonu, yüksek kaliteli rüzgâr enerjisi tahminlerini gerektirir. Bu araştırma, rüzgâr hızı ve güç üretiminin kısa vadeli tahminine odaklanmaktadır. İlk olarak rüzgâr hızı tahmini incelenmiştir. Beş farklı yaklaşımın tahmin performansını analiz etmek için kapsamlı bir vaka çalışması yapılmıştır: çok değişkenli Facebook Prophet, hareketli ortalama ile entegre mevsimsel otoregresif (SARIMA), dış değişkenli SARIMA (SARIMAX), kapılı tekrarlayan birimler (GRU), ve uzun kısa süreli bellek (LSTM) modelleri kullanılmıştır. Performas göstergeleri, R-kare (R^2), ortalama kare hata (MSE), ortalama karakök hata (RMSE), ve ortalama mutlak hata (MAE), modellerin etkinliğini doğrulamak için uygulanmıştır. LSTM modeli ile elde edilen tahminler, gerçek zamanlı rüzgâr hızı ile neredeyse örtüşmekte olup, aynı zamanda performans göstergeleri tarafından da desteklenmektedir. Bu da, LSTM modelinin IYTE ölçüm direğinin veri seti için diğer yöntemlerden daha iyi performans sergilediğini göstermektedir. Çalışmanın ikinci kısmı, rüzgâr tarlalarında rüzgâr hızı tahmini ve rüzgâr rütbinlerine ait güç eğrilerini, ve LSTM modelini kullanarak rüzgâr enerjisi üretiminin tahmin edilmesidir. Önerilen model, EPİAŞ Şeffaflık Platformu'ndan alınan rüzgâr enerjisi üretim verileri kullanılarak doğrulanmıştır. Mevcut olmayan meteorolojik veri seti nedeniyle, rüzgar hızını ve güç üretimini tahmin etmek için konumun ERA5 veri seti kullanılmıştır. Ayrıca, Gün Öncesi Piyasası için her bir rüzgar santralinin günlük tahminleri elde edilmiştir. Sonuçlar, ERA5 veri seti ile LSTM modelininin kullanılmasının, rüzgar tarlalarının kendi tahminlerinden daha iyi tahminler verebileceğini göstermektedir. Ayrıca SCADA verilerinin elde edilmesi durumunda tahmin performansının arttırılabileceği anlaşılmıştır.

Dedicated to my family…

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

Figure **Page**

LIST OF TABLES

Table **Page**

LIST OF ABBREVIATIONS

AC: Autocorrelation

ANN: Artificial Neural Network

AR: Autoregressive Model

ARMA: Autoregressive Moving Average Model

ARIMA: Autoregressive Integrated with Moving Average Model

BP: Back Propagation

CF: Capacity Factor

CFD: Computational Fluid Dynamics

DAM: Day-ahead Market

DTU: University of Denmark

ECMWF: European Centre for Medium-range Weather Forecast

ELM: Extreme Learning Machine

EKF: Extended Kalman Filter

GP: Gaussian Process

GRU: Gated Recurrent Units

HIRLAM: High-Resolution Limited Area Model

ICS: Improved Cuckoo Search

IEA: International Energy Agency

IZTECH: Izmir Institute of Technology

IMM: Informatics and Mathematical Modelling

MA: Moving Average Model

MAE: Mean Absolute Error

MAPE: Mean Absolute Percentage Error

MOS: Model Output Statistics

MSE: Mean Square Error

NWP: Numerical Weather Prediction

LSSVM: Least Squares Support Vector Machine

LSTM: Long Short-term Memory

PAC: Partial Autocorrelation

PPA: Pre-combined Prediction Value

PSO-DBN: Deep Belief Networks based on Particle Swarm Optimization

R^2: R-square

RMSE: Root Mean Square Error

RMLP: Recurrent Multilayer Perceptron

RNN: Recurrent Neural Network

SARIMA: Seasonal ARIMA Model

SARIMAX: SARIMA with Exogenous Variable Model

SC: Spatial Correlation

SVM: Support Vector Machine

VMD: Variational Mode Decomposition

WAsP: Wind Atlas Analysis and Application Program (WAsP)

WN-LSTM: Wavelet Kernels LSTM

WPPT: Wind Power Prediction Tool

WRF: Weather Research and Forecasting

CHAPTER 1

INTRODUCTION

Wind energy has been one of the most rapidly growing renewable energy sources in recent years (IEA, 2021). As is well known, it is an environmentally friendly and cost-effective energy source that contributes to pollution reduction and economic development. Because of the randomness and intermittency of wind characteristics, increasing wind power penetration into modern grids can impact power system operation safety and quality. Energy storage and accurate wind speed forecasts can help to solve these issues.

Limitations of excess energy storage lead to wind speed forecasting. For the usage of wind energy safely and efficiently, it is necessary to improve the accuracy of wind speed prediction. Since wind speed affects the wind energy produced, accurate wind speed prediction models can enhance the safety of energy systems. However, compared to other traditional power plants, wind speeds are highly dependent on various meteorological factors (e.g., temperature, relative humidity, barometric pressure, and wind direction). Wind power is not easily predictable due to its highly probabilistic and fluctuating properties.

Research and contributions are currently being made on wind speed prediction. Many methods have been proposed in the literature to improve the accuracy and accuracy of forecasts. These methods can be divided into four groups: physical, statistical, artificial intelligence, and hybrid approaches. In addition, forecasting algorithms can be categorized based on very short-term (several seconds and 30 minutes ahead), short-term (30 minutes and 72 hours ahead), medium-term (72 hours and one week ahead), and long-term (more than a week and a year ago) forecast periods (Meka et al., 2021). Recent studies have focused primarily on short-term wind forecasts due to the importance of wind forecasts for energy systems. In particular, the day before market, regulation, disposal, planning, load tracking, and other system operations occur during these periods. Therefore, a group of short-term prediction methods is examined in this study.

The study is organized as follows: a literature survey of wind energy forecasting methods and the purpose of the thesis are given in the first chapter; Chapter 2 describes the sites under study; the forecasting methods are compared in Chapter 3; the theory and method of the study are explained in Chapter 4, two different approaches in Chapter 5 predict the wind power production of three operational wind farm.

1.1 Literature Survey

Before the 1990s, the significance of short-term prediction to power systems started to be attempted by the Pacific Northwest Laboratory (Wendel et al., 1978). Their conclusions showed that sufficiently reliable predictions might be used for the operation stages, such as weekly predictions for maintenance scheduling, daily predictions for load scheduling, and hourly forecasts to dispatch decisions. Wegley et al. (1984) investigated three methods: the persistence, autoregressive and generalized equivalent Markov models for 10-, 30-, and 60-min ahead. While the persistence model performed well for the shortest time ahead, the generalized equivalent Markov model was adequate for the longest. Geerts (1984) established ARMA and Kalman filter models for the wind energy integration to the grid with an hourly time-step for a 24 h forecast horizon. While both models outperformed the persistence model up to 16h, ARMA (2,1) performed better results than the Kalman filter. The persistence model is the most frequently utilized benchmark method in the literature. His results also remarked that using other variables could improve the forecast accuracy (e.g., temperature, pressure, and wind direction).

In the 1990s, the installed capacity of wind energy was increasing worldwide, which caused a necessity to integrate an increase in wind power fluctuation into the grid. Motivated by this necessity, the electricity markets' members and researchers concerted their attention on short-term forecasting. Watson et al. (1992) worked on decreasing fossil fuel costs, utilizing numerical weather prediction (NWP) and model output statistics (MOS) to forecast wind speed and direction up to 18 h forecast horizon with an hourly time-step. Using the forecasts, they performed a case study on the UK grid system. They concluded that the predictions obtained by NWP and MOS could produce a significant improvement in fossil fuel savings compared to the persistence model. Jensen et al. (1994) proposed a wind power prediction tool (WPPT) enhanced by the Department of Informatics and Mathematical Modelling (IMM) from the Technical University of Denmark (DTU). The tool was built based on an autoregressive model using power as the primary variable and wind speed as the exogenous variable and verified with seven wind farms. The predictions were obtained up to 36 h with a half-hourly time-step oriented to dispatch decisions.

Since the year 2000, Zhang (2003) combined a seasonal ARIMA (SARIMA) and the least squares support vector machine (LSSVM) model. LSSVM improved results by forecasting the residuals of SARIMA outputs. Wang et al. (2004) described a mathematically nonlinear neural network model. In this algorithm, ANNs capture the short-term pattern in wind speed data. The long-term pattern is categorized as increasing, decreasing, and almost stable. The process is divided into short-term forecasting and adapting results to long-term forecasting. The results were compared to other linear regression approaches. This model improves the forecasting accuracy in short-term and long-term predictions. Damousis et al. (2004) developed a fuzzy logic model based on the spatial correlation model for wind speed and power generation forecasting.

In contrast, it performs good results on flat terrain, but its performance declines in a complex landscape. Guoyang et al. (2005) stated that statistical methods use pattern identification, parameter estimation, and model checking to make a mathematical model of a problem based on a historical data set. The methods proposed by Jenkins can be divided into an autoregressive (AR), a moving average (MA), an autoregressive moving average (ARMA), and an autoregressive integrated moving average (ARIMA) model. Louka et al. (2008) indicated that the Kalman filter could eliminate systematic errors in the forecasting results of the NWP model. Jung et al. stated that physical methods outperform statistical methods in long-term wind prediction. As the NWP model slowly updates and lags behind the historical data, it might cause significant errors in the forecasting results. Due to the computational complexity, their applicability in short-term wind prediction is limited.

Cassola et al. (2012) used the Kalman filter, which evaluates the predictions and observations obtained from the NWP model with statistical methods and establishes a regression between the predictions and observations. It corrects erroneous forecasts from the model based on the observations. Li et al. (2015) defined a dynamic SC model with a tracking framework created by the Kalman filter between the geographically distributed wind farms for short-term wind predictions. Filik et al. (2017) proposed ANN-based models that differentially combine multiple local meteorological measurements such as wind speed, temperature, and pressure values. This method could improve wind speed forecasting for various situations. Zhu et al. (2019) studied methods that could predict wind speed in multiple regions by adding the SC model. Spatial characteristics were obtained by long short-term memory (LSTM) and a convolutional neural network. The forecasting limitations caused by a rapid change in wind speed were overcome by considering the geographical characteristics of the wind farms in the dynamic SC model. Shahid et al. (2020) developed a hybrid prediction model for nonlinear mapping using long short-term memory with wavelet kernels (WN-LSTM), which enriches deep learning for vanishing gradient and wavelet transformations. Compared with the well-known existing models, a percentage improvement of up to 30% was obtained. Liu et al. (2020) proposed a model which consists of three stages. In the first step, the empirical wavelet transform reduces the nonstationarity of wind speed data by decomposing the data into subarrays. In the second step, three types of deep networks are applied to construct the forecasting model and calculate the results of all sub-series. In the last step, the reinforcement learning method combines the three deep networks. The results of each series are combined to get the final prediction results. Compared with nineteen alternative models, it provides the best accuracy. Hu et al. (2021) proposed a hybrid short-term forecasting method that integrates the corrected NWP and SC models into a Gaussian process (GP). Compared with the primary GP, the forecasting accuracy in different seasons is developed at 7.02%-29.7% using the corrected NWP, 0.65-10.23% after integrating SC, and 10.88-37.49% using the proposed hybrid model.

1.2 The Purpose of the Thesis

Renewable energy sources are the mainstay of any energy transition to reach the net zero target. While the countries gradually shift away from conventional resources, it is critical to understand the significant role of renewables in decarbonizing multiple areas to guarantee a smooth pathway to the net zero targets (IEA, 2021). Regarding renewables, wind energy has become one of the most important sources, with huge reserves and high commercial development value. However, the wind power series is highly nonlinear and nonstationary due to the inherent characteristics of wind energy. That can cause serious power imbalance issues due to voltage and frequency fluctuations and seriously affects the power system dispatching, especially in the case of large-scale wind power integration into the grid. Most scholars focus on developing a highly accurate wind power prediction method to provide a more secure and stable power system (Jiandong et al., 2022).

The forecasting models used in today's research and engineering are being investigated to find the best-performing model for an operational wind farm with real-time production data. This research focuses on the short-term prediction of wind power generation, analyzing the different forecasting methods. Based on the earlier review and analysis, a model will be proposed to forecast wind speed. Datasets for four seasons collected from a meteorological mast and ERA5, mentioned in Section 2.1, will be utilized to verify the performance of forecasting models. For each experiment, all models will be evaluated separately. Four performance evaluation metrics will be used to compare the actual and prediction values. According to the analysis, the model gives better results, which indicates a higher accuracy, and will be selected.

Considering the proposed model's effectiveness and efficiency, it is applied to wind power forecasting of three nearby wind farms as case studies: Urla, Kores Kocadag, and Germiyan wind farms, introduced in Chapter 2. The prediction results will be tested using the wind farms' real-time power generation. Finally, it will be determined whether nearby wind farms' power outputs could be predicted.

CHAPTER 2

SITES DESCRIPTIONS

2.1 The IZTECH Meteorological Mast

The IZTECH meteorological mast was established at N 38º19'60" and E 26º37'58" in 2017. Its location is displayed in Figure 2.1.

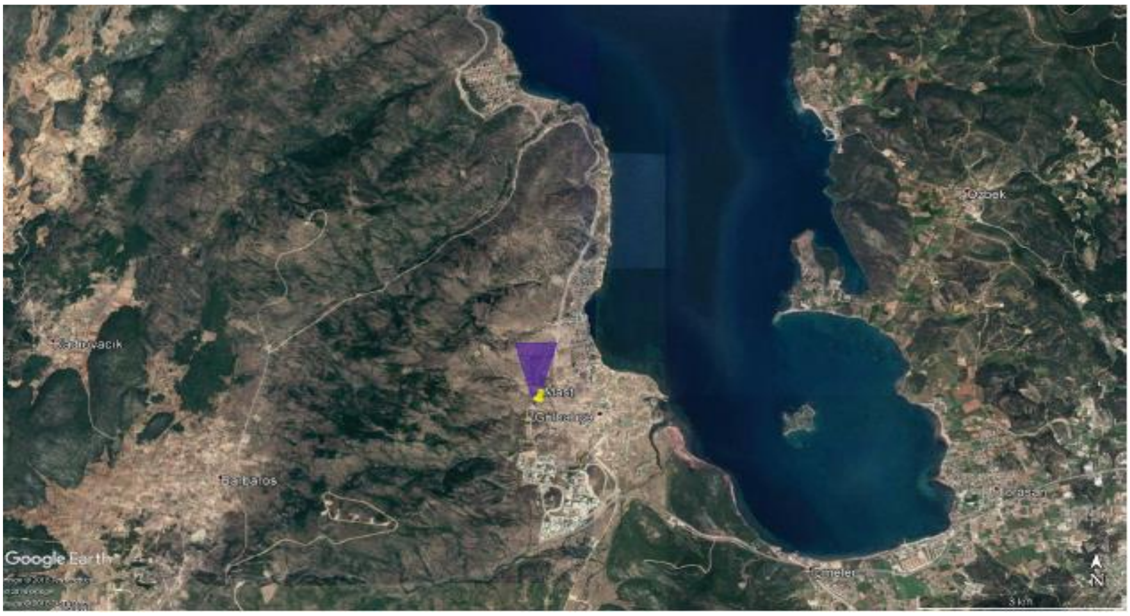


Figure 2.1. The IZTECH Meteorological Mast location

The height of the meteorological mast shown in Figure 2.2 is 101 m, and the elevation from sea level is 52 m. It contains several instruments to measure the different quantities. These are one lightning rod, a cup anemometer, a flashlight mounted, three wind vanes, backup, other anemometers, humidity, temperature sensors, two ultrasonic 3D anemometers, an air pressure sensor, and a data logger. Velocity and direction data are obtained from anemometers and wind vanes. Also, the mast has a solar panel and aviation light which maintains the mast from any accident. All the instruments' locations are respectively shown in Figure 2.3. In addition, IEC 61400-12 is applied as a standard for its setup (Tuna et al., 2018).

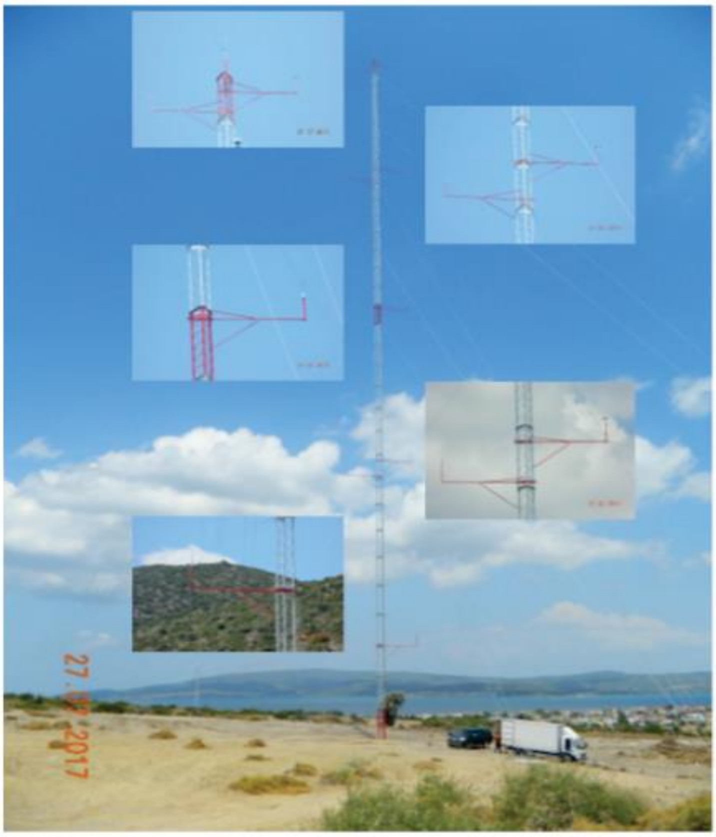


Figure 2.2. The IZTECH Meteorological Mast (Source: Tuna et al., 2018)

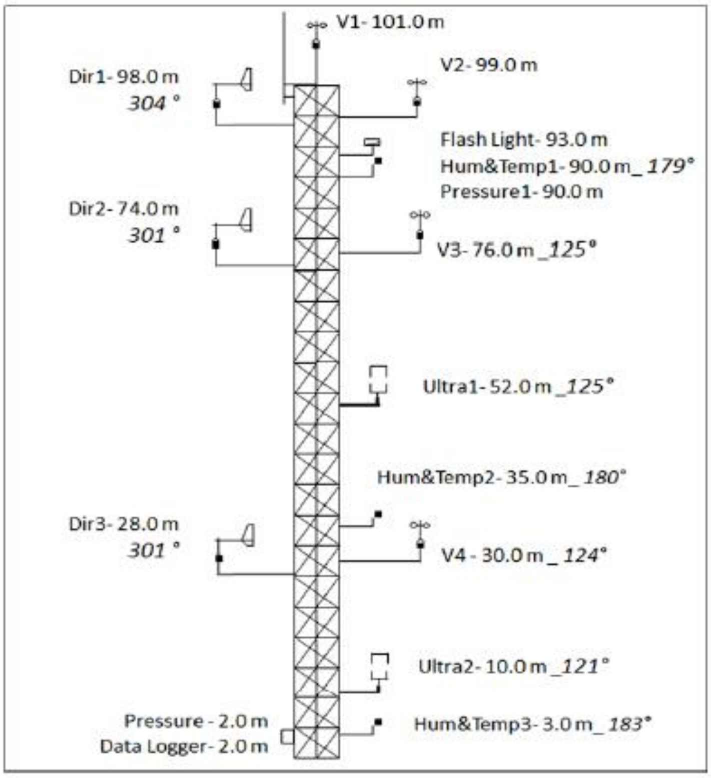


Figure 2.3. Technical drawing of the IZTECH Meteorological Mast and instruments (Source: Tuna et al., 2018)

The detailed information about the met mast is respectively tabulated in Table 2.1. Also, the derived values and their related parameters are classified in Table 2.2.

Table 2.1. The IZTECH Mat Properties (Source: Tuna et al., 2018)

Height	Channel	Sensor	Unit
101 m	WS_{101}	Thies First Class Adv. Anemometer	m/s
99 m	WS_{99}		
76 m	WS_{76}		
30 m	WS_{30}		
52 m	WS_{52}, WD_{52}, θ_{vir52}	Gill WindMaster 3D Anemometer	m/s, ^{0}, ^{0}C
10 m			
98 m	WD_{98}	Thies First Class Wind Vane	^{0}
74 m	WD_{74}		
28 m	WD_{28}		
90 m	RH_{90}, T_{90}	Galtec KPC 1.S/6-ME	%, ^{0}C
35 m	RH_{35}, T_{35}		
3 m	RH_{3}, T_{3}		
90 m	P_{90}	Thies 3.1157.10.000 Pressure	hPa
2 m	P_{2}		
2 m	-	Ammonit Meteo40 Data Logger	-

Table 2.2. Derived parameters (Source: Tuna et al., 2018)

Derived Values	Related Parameters
$\boldsymbol{\rho}_{10}$	P_{10}, RH_{10}, T_{10}
$\boldsymbol{\rho}_{52}$	P_{52}, RH_{52}, T_{52}
$\boldsymbol{u}_{*,10}$	u_{10}, v_{10}, w_{10}
$\boldsymbol{u}_{*,52}$	u_{52}, v_{52}, w_{52}
$\boldsymbol{Q}_{,10}$	WS_{52}, WD_{52}, θ_{vir52}
$\boldsymbol{Q}_{,52}$	WS_{10}, WD_{10}, θ_{vir10}
$\mathbf{L}_{10}$	w_{10}, T_{10}, $u_{*,10}$, z
$\mathbf{L}_{52}$	w_{52}, T_{52}, $\boldsymbol{u}_{*,52}$, z

2.2 Capacity Factor

A capacity factor (CF) of a wind farm indicates how much energy is produced over a given period relative to the theoretical maximum possible it could provide, that is, operating full time (24 hours a day, 365 days a year):

$$CF = \frac{Actual\ power\ production}{Rated\ power\ operating\ full\ time} \times 100\% \tag{2.1}$$

CF can be calculated for a single wind turbine, a wind farm, or a region consisting of many wind farms. It depends on factors such as geographical location, turbine design, etc. For example, combining a larger rotor with a smaller generator can achieve a higher CF. Even a higher CF does not indicate higher efficiency or vice versa; it can be said that a higher CF is particularly more economical because the average CF of a wind turbine is proportional to the present net return over its lifetime. A wind farm's typical CF varies between 20% and 40%. That's why the CF value plays an essential role in deciding about the wind farm (Zhang, 2015).

2.3 Three Operational Wind Farms

Three operational wind farms under study in this work are Urla, Kores Kocadag, and Germiyan Wind Farms, located in the Urla region of Izmir. Onshore wind farms are composed of five, ten, and six wind turbines. The Urla and Germiyan wind farms started to operate in 2016, while the Kores Kocadag Wind Farm has been operating since 2013. The wind farms are displayed in Figure 2.4.

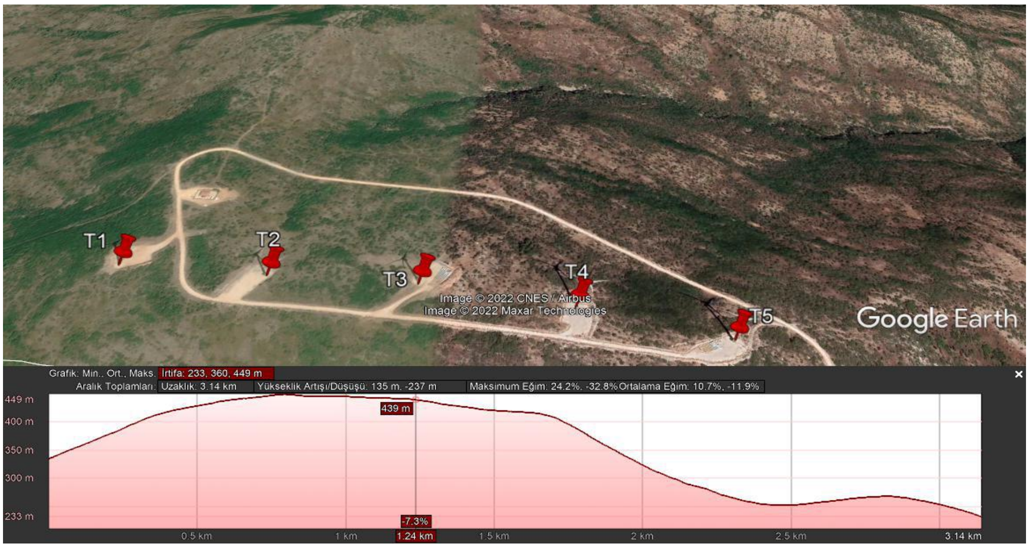
Figure 2.4. Three operational wind farms (a) The Urla Wind Farm (Source: Egenda, n.d.) (b) The Kores Kocadag Wind Farm (Source: Dost Energy, n.d.) (c) The Germiyan Wind Farm (Source: Egenda, n.d.)

The technical data with its main specifications and the total installed capacity of each wind farm are tabulated in Table 2.3.

Table 2.3. The technical data and total installed capacities of wind farms

Wind Farm	**Turbine Type**	**Rated Power (kW)**	**Cut-in wind speed (m/s)**	**Cut-out wind speed (m/s)**	**Rotor Diameter (m)**	**Hub Height (m)**	**Total Installed Capacity MWm/MWe**	
Urla	Enercon E-82 E4	3000	3.0	34	82	84	15	13
Kores Kocadag	Nordex N90	2500	3.0	25	90	100	25	
	Nordex N100	2500	3.0	20	99.8	100		
Germiyan	Enercon E82 E2	2000	2.0	34	82	108	12	10.8

The wind farms are on a complex terrain; their average elevations above sea level are 450 m, 345 m, and 147 m, as seen in Figure 2.5.


T1
T2
T3
T4
T5
Image © 2022 CNES / Airbus
Image © 2022 Maxar Technologies
Google Earth
Grafik: Min., Ort., Maks. İrtifa: 233, 360, 449 m
Aralık Toplamları: Uzaklık: 3.14 km
Yükseklik Artışı/Düşüşü: 135 m, -237 m
Maksimum Eğim: 24.2%, -32.8%
Ortalama Eğim: 10.7%, -11.9%
449 m
400 m
350 m
300 m
233 m
439 m
-7.3%
0.5 km
1 km
1.24 km
1.5 km
2 km
2.5 km
3.14 km

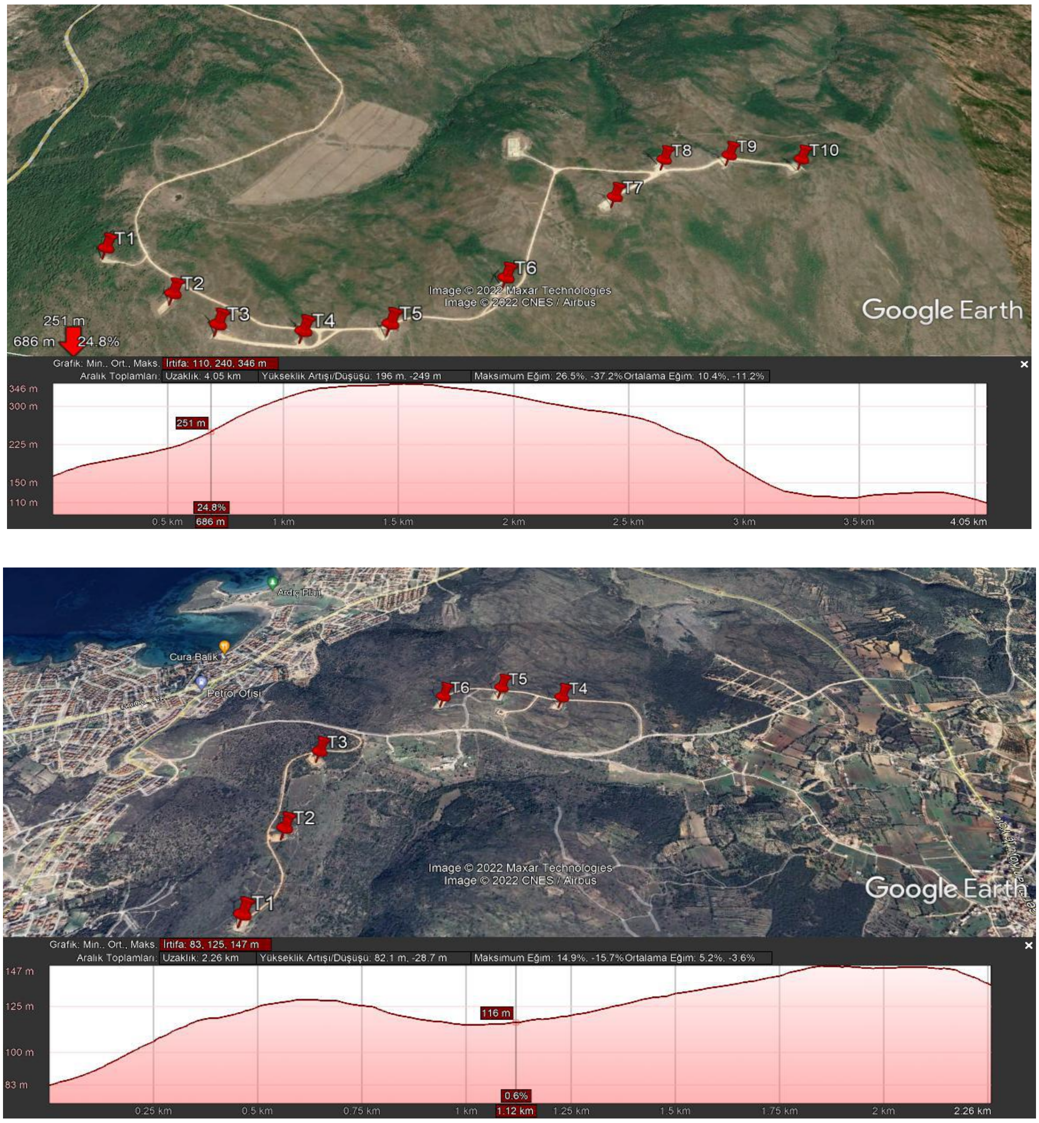

T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
251 m
686 m
24.8%
Image © 2022 Maxar Technologies
Image © 2022 CNES / Airbus
Google Earth
Grafik: Min., Ort., Maks. İrtifa: 110, 240, 346 m
Aralık Toplamları: Uzaklık: 4.05 km
Yükseklik Artışı/Düşüşü: 196 m, -249 m
Maksimum Eğim: 26.5%, -37.2%
Ortalama Eğim: 10.4%, -11.2%
346 m
300 m
225 m
150 m
110 m
0.5 km
1 km
1.5 km
2 km
2.5 km
3 km
3.5 km
4.05 km
Ardıç Plajı
Cura Balık
Petrol Ofisi
T1
T2
T3
T4
T5
T6
Image © 2022 Maxar Technologies
Image © 2022 CNES / Airbus
Google Earth
Grafik: Min., Ort., Maks. İrtifa: 83, 125, 147 m
Aralık Toplamları: Uzaklık: 2.26 km
Yükseklik Artışı/Düşüşü: 82.1 m, -28.7 m
Maksimum Eğim: 14.9%, -15.7%
Ortalama Eğim: 5.2%, -3.6%
147 m
125 m
100 m
83 m
116 m
0.6%
0.25 km
0.5 km
0.75 km
1 km
1.12 km
1.25 km
1.5 km
1.75 km
2 km
2.26 km

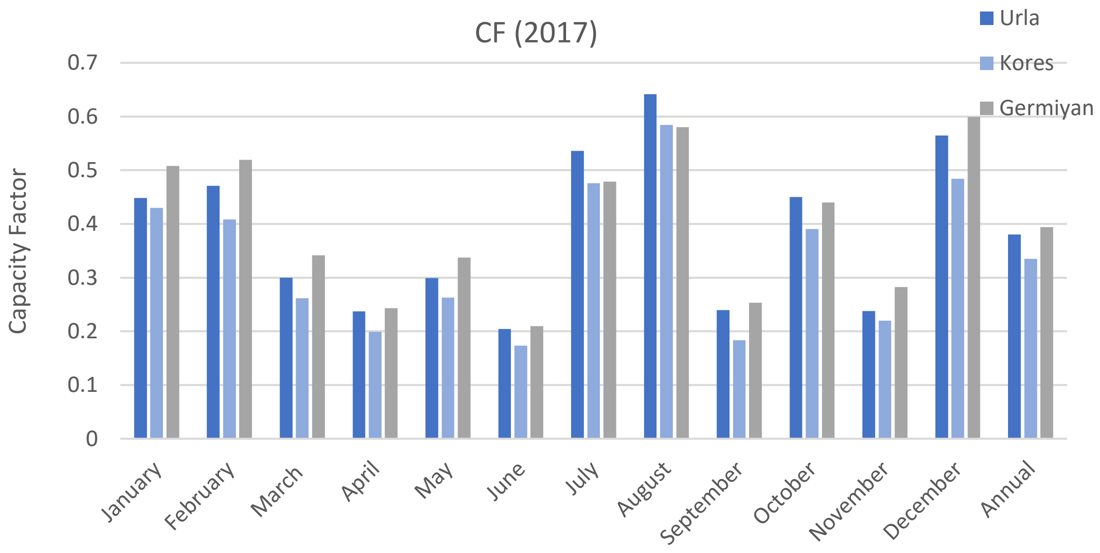
Figure 2.5. The locations and elevations of three operational wind farms (a) The Urla Wind Farm, (b) The Kores Kocadag Wind Farm, (c) The Germiyan Wind Farm

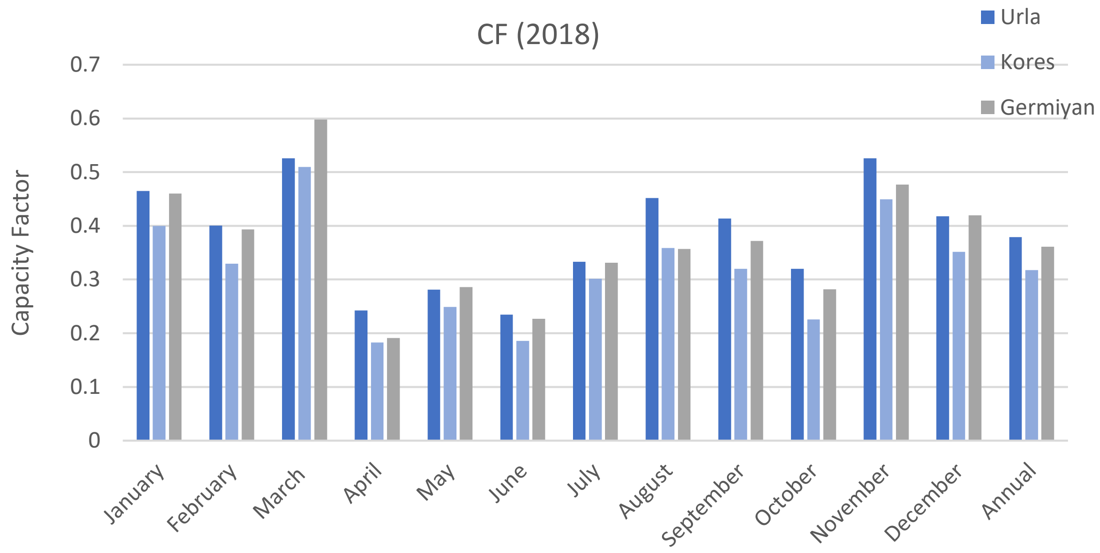
The turbines are arranged in one row for each wind farm, with a length of 930 m, 2410 m, and 1855 m, respectively. The Urla Wind Farm array is regularly spaced, and the spacing between turbines is 230 m (~2.8D). The Kores Kocadag Wind Farm array is irregularly spaced, and the spacing between turbines varies from 210 m (~2.33D) to 410 m (~4.14D). The Germiyan Wind Farm arrangements are also irregularly spaced, and the spacing between turbines is 310 m (~3.78D) and 500 m (~6.1D). The coordinates of each turbine in the wind farms are tabulated in Table 2.4.

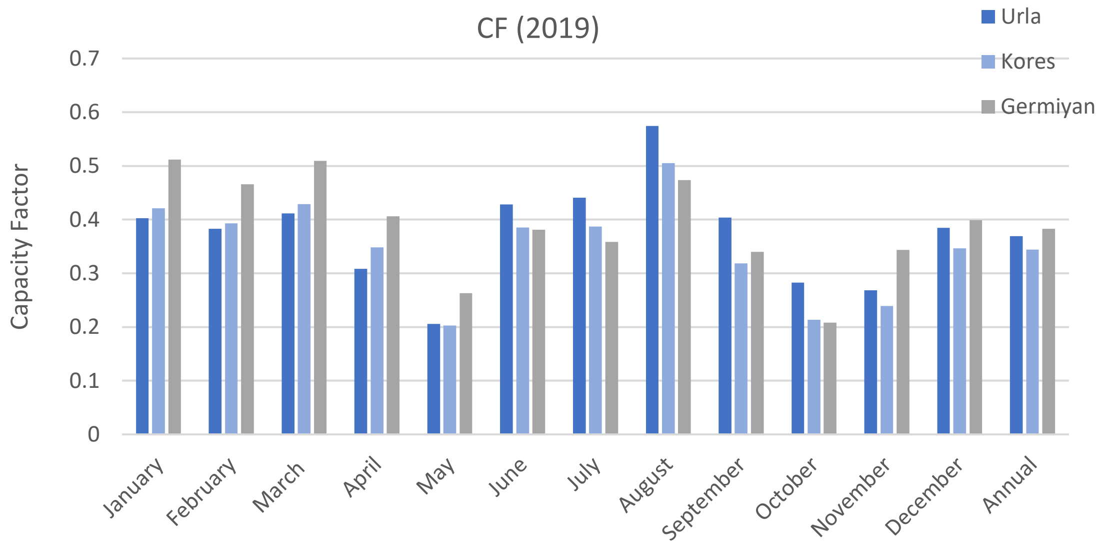
Table 2.4. The coordinates of each turbine in the wind farms

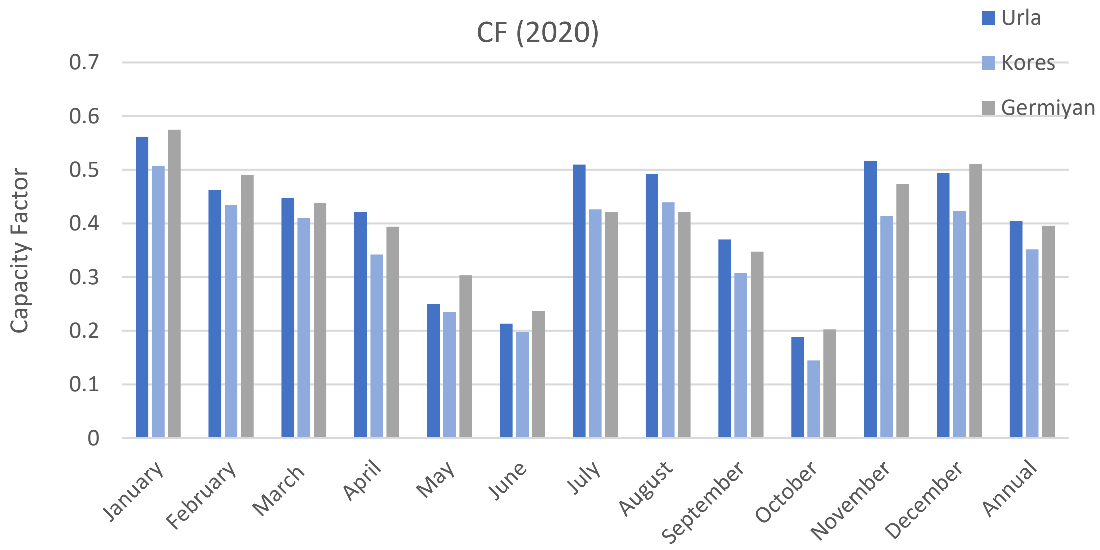
Wind Farm/Turbines Coordinates	Urla	Kores Kocadag	Germiyan
T1	38° 19' 2.676" 26° 36' 9.2052"	38° 17' 26.682" 26° 35' 27.024"	38° 19' 6.2544" 26° 26' 18.7794"
T2	38° 19' 2.676" 26° 36' 9.2052"	38° 17' 26.682" 26° 35' 27.024"	38° 19' 6.2544" 26° 26' 18.7794"
T3	38° 19' 2.676" 26° 36' 9.2052"	38° 17' 26.682" 26° 35' 27.024"	38° 19' 16.5822" 26° 26' 3.2712"
T4	38° 19' 2.676" 26° 36' 9.2052"	38° 17' 26.682" 26° 35' 27.024"	38° 19' 36.6888" 26° 26' 27.909"
T5	38° 19' 2.676" 26° 36' 9.2052"	38° 17' 26.682" 26° 35' 27.024"	38° 19' 36.6888" 26° 26' 27.909"
T6	-	38° 17' 26.682" 26° 35' 27.024"	38° 19' 36.6888" 26° 26' 27.909"
T7	-	38° 17' 32.748" 26° 36' 15.1698"	-
T8	-	38° 17' 32.748" 26° 36' 15.1698"	-
T9	-	38° 17' 32.748" 26° 36' 15.1698"	-
T10	-	38° 17' 32.748" 26° 36' 15.1698"	-

The monthly and annual capacity factors (CF) of three operational wind farms from 2017 to 2021 are displayed in Figure 2.6 (EPIAS, n.d.)




CF (2018)
Capacity Factor
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
January
February
March
April
May
June
July
August
September
October
November
December
Annual
Urla
Kores
Germiyan


CF (2019)
Capacity Factor
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
January
February
March
April
May
June
July
August
September
October
November
December
Annual
Urla
Kores
Germiyan


CF (2020)
Capacity Factor
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
January
February
March
April
May
June
July
August
September
October
November
December
Annual
Urla
Kores
Germiyan

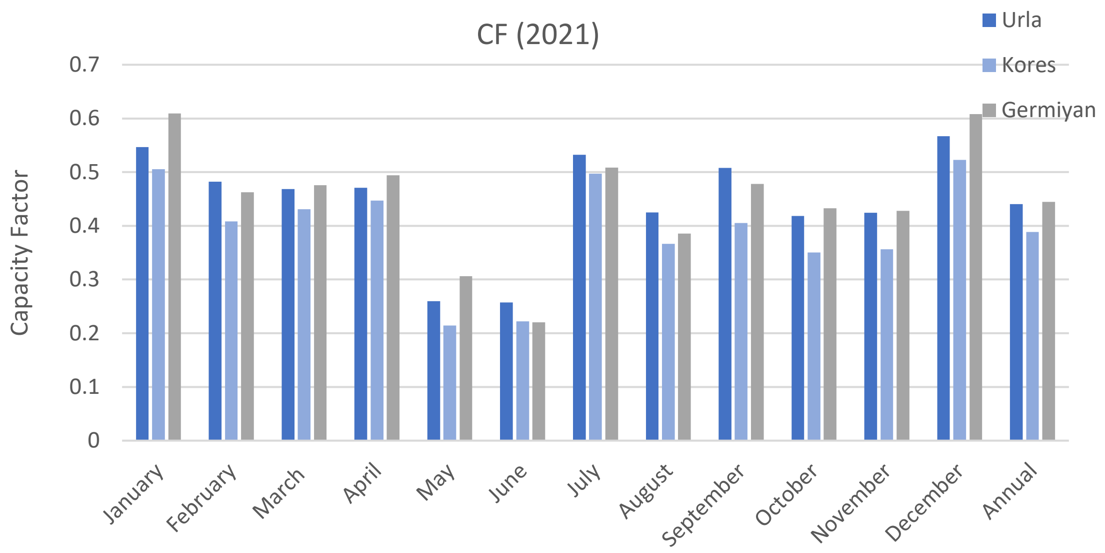


Figure 2.6. The monthly and annual CF for three operational wind farms from 2017 to 2021 (Source: EPIAS, n.d.)

The monthly and annual power production of each wind farm from 2017 to 2021 are tabulated in Table 2.5.

Table 2.5. The monthly and annual power productions of three operational wind farms (Source: EPIAS, n.d.)

Year	**Month**	**Urla (MWh)**	**Kores (MWh)**	**Germiyan (MWh)**
2017	January	4195.1	7735	3951.32
	February	4406.99	7352	4036.51
	March	2807.23	4707	2657.07
	April	2220.23	3580	1890.26
	May	2803.05	4730	2625.74
	June	1910.31	3125	1629.07
	July	5016.31	8562	3722.91
	August	6005.12	10523	4510.05
	September	2243.76	3304	1971.84
	October	4215.51	7028	3421.96
	November	2226.83	3953	2195.16
	December	5285.65	8719	4660.28
	Annual	43336.09	73318	37272.17
2018	January	4351.59	7193	3576.99
	February	3746.14	5935	3058.45
	March	4919.73	9170	4651.46
	April	2270.19	3287	1486.8
	May	2634.17	4477	2221.46
	June	2193.72	3348	1764

(Cont. of the next page)

(Cont. of the next page)

	July	3115.01	5429	2576.07
	August	4228.24	6454	2777.94
	September	3869.15	5760	2890.64
	October	2992.63	4063	2189.58
	November	4919.91	8088	3707.51
	December	3912.02	6331	3264.6
	Annual	43152.5	69535	34165.5
2019	January	3768.09	7576	3977.31
	February	3585.26	7074	3620.5
	March	3850.34	7721	3960.96
	April	2888.62	6266	3156.87
	May	1926.88	3648	2044.82
	June	4009.13	6940	2961.5
	July	4125.41	6964	2789.52
	August	5377.46	9096	3681.25
	September	3777.86	5731	2645.53
	October	2647.23	3842	1616.94
	November	2512.92	4300	2672.83
	December	3599.11	6239	3101.3
	Annual	42068.31	75397	36229.33
2020	January	5256.6	9125	4470.02
	February	4325.82	7819	3814.73
	March	4188.29	7382	3405.12
	April	3944.84	6154	3062.58
	May	2344.64	4220	2356.81
	June	1995.04	3564	1844.57
	July	4773.32	7669	3274.03
	August	4606.12	7908	3274.06
	September	3463.88	5538	2699.47
	October	1761.49	2605	1574.31
	November	4838.75	7447	3679.72
	December	4619.7	7621	3972.87
	Annual	46118.49	77052	37428.29
2021	January	5115.82	9098	4735.68
	February	4514.34	7351	3595.23
	March	4385.97	7756	3698.37
	April	4406.23	8050	3843.97
	May	2428.63	3861	2379.4
	June	2409.16	3996	1714.34
	July	4985.29	8948	3952.6
	August	3979.3	6595	2996.3
	September	4755.54	7296	3715.05
	October	3913.44	6305	3363. 27

(Cont. of the next page)

(Cont. of the next page)

	November	3974.68	6409	3328.2
	December	5307.39	9413	4730.09
	Annual	50175.79	85078	42052.5

CHAPTER 3

THE COMPARISON OF FORECASTING METHODS

3.1 Forecasting Methods

3.1.1. Physical Methods

Physical models consider physical factors such as terrain, obstacles, temperature, and pressure to predict wind speed. The physical models can be investigated in Computational Fluid Dynamics (CFD) and the Diagnostic Model. While CFD methods are employed to simulate the wind flow over complex terrain, Diagnostic models utilize the parametrizations of the boundary layer for wind flow over flat terrain (Wang et al., 2018; Castellani et al., 2016).

Another usage of these methods is an auxiliary input for the first step of other forecasting methods. Numerical weather prediction (NWP) is one of the most used physical methods meteorologists develop. Generally, it is utilized for large-scale weather prediction. Physical methods are mainly based on NWP, and the manufacturer's power curves are used in case of missing historical data. These methods may not provide accurate results for short-term wind prediction. NWP numerically solves the conservation equations at the specified region to increase the accuracy. Simultaneously, digital elevation models to describe the topography should be employed in NWP to obtain better results. Model output statistics (MOS) can be applied to minimize residual error (Tascikaraoglu et al., 2014; Lei et al., 2009; Foley et al., 2011; Hu et al., 2021).

Several physical methods have been developed from now on. Risoe National Laboratory in Denmark developed The Predictor, which uses the NWP prediction grom High-Resolution Limited Area Model (HIRLAM) and Wind Atlas Analysis and Application Program (WAsP) to consider the local conditions (Landberg, 1998; Landberg, 1999; Landberg, 2001).

A model is developed, which has a modular composition that enables the integration of several physical process modules, and each module has been built by a different group (Dudhia, 2014). The method is a new-generation mesoscale NWP model called Weather Research and Forecasting (WRF) (Skamarock et al., 2008). Recently, the WRF model has been employed for more studies evaluating turbine-height wind speed.

3.1.2. Statistical Methods

Statistical methods use pattern identification, parameter estimation, and model checking to make a mathematical model of a problem based on a historical data set (Lei, 2009). The statistical approach describes the relations between wind speed (or power) prediction and online measured data. Numerous data are analyzed, and meteorological variables are not characterized in a statistical approach.

The statistical models are relatively simple compared to the physical ones, and large-scale supplementary monitoring equipment is not essential, decreasing the cost. It can be specified that statistical approaches are commonly used for short-term prediction models since physical models need several computational sources and take a long lead time. However, the accuracy of statistical models depends on the reliability of historical data and the number of observations (Di et al., 2019). Additionally, these models are usually linear, with limited capability to forecast wind speed series containing highly nonlinear and non-stationary characteristics (Nie et al., 2021).

The statistical approach contains several autoregressive methods. The linear autoregressive method's prediction error is larger than the polynomial method due to the wind power fluctuations. The polynomial autoregressive model is a nonlinear regression model that shows a better fit for wind power prediction (Li et al., 2021).

An autoregressive moving average (ARMA) model is established with different orders based on an order determination method and the wind power curve (Dong et al., 2011). The weighted average method is adapted to the ARMA models to obtain the forecast values, which improves the forecast result to a certain extent. Still, the running time and calculation amount get more prominent due to the step calculation.

Another approach is the Markov chain method, which aims to generate good synthetic wind speed data to obtain more accurate models. An artificial time series is developed utilizing Monte Carlo simulations for wind speed. Three semi-Markov models are proposed, and these models' statistical properties are compared with actual data and a synthetic time series produced over a simple Markov chain (D’Amico et al., 2013).

Kalman filters have also been widely utilized for meteorological variables in data assimilation and improving weather forecasting performance. While using classical Kalman filters improves air temperature forecasting, similar work for wind speed forecasting may produce poor results. A polynomial Kalman filter is proposed to improve the results according to the correction of the 10-m wind speed prediction (Cassola et al., 2012).

Current statistical approaches mainly focus on machine learning-based models to better understand the relation between linear and nonlinear characteristics of wind speed time series.

3.1.3. Artificial Intelligence Methods

Since the development of artificial intelligence technology, scientists have constructed intelligent prediction methods to employ them for wind energy prediction, containing artificial neural networks (ANN), extreme learning machines (ELM), and support vector machines (SVM) (Nie et al., 2021).

ANN is a commonly used method that consists of many layers. It predicts the wind speed by learning from a data set with input-output mapping. Input and output data are required to train and test these networks. The features of ANN, such as being fault-tolerant, fast, and straightforward, being able to learn and generalize, and being adaptable to different situations, are significant. Another method is the fuzzy logic model, which is used when it is difficult to model a system (Lei et al., 2008).

Deep learning methods have been focused on wind speed and power production because of their three characteristics: strong generalization skills, unsupervised feature learning, and big data training to develop the performance of prediction (Shadid et al., 2020; Jiandong et al., 2022). Because of the prediction dependency on historical information, RNN employs complex vector values (Olaofe, 2020). The long-term dependencies cause the vanishing gradient problem. Long short-term memory (LSTM) is proposed to prevent this problem, which uses the loopback memory to save the gradient along the long-term dependencies (Hochreiter et al., 2017).

SVM is one of the powerful machine learning methods which can be effectively employed for time-series forecasting with good results in different areas (Zhou et al., 2010). Such as, daily air pollution is predicted using SVM and wavelet decomposition (Osowski et al., 2007). SVM could obtain a feed-forward network design with a single hidden layer of cells that are primarily nonlinear (Sreelakshmi et al., 2008). It is featured using a kernel trick technique for nonlinear classification issues. SVM can deal with high dimensional data even with small training data and could sufficiently handle the generalization of complex methods (Belousov et al., 2002).

A control algorithm is supplied to predict the wind speed and power based on the ANN method using the back-propagation approach. The results show that this method could help obtain better economic benefits (Flores et al., 2002). A fuzzy method based on spatial correlation is proposed to forecast wind speed and power production. The technique performs better for flat terrain than complex landscapes (Damousis et al., 2004). Apart from these approaches, different hybrid forecasting algorithms have been developed to get the advantage of the unique ability of individual methods.

3.1.4. Hybrid Methods

Hybrid methods combine the final prediction performance of individual forecasting models and ensure significant advantages compared to the unique models. Hybrid models usually consist of both linear and nonlinear models. Combining a linear and nonlinear model to forecast the hidden components embedded in the wind speed could show better performance to improve prediction accuracy (Tascikaraoglu et al., 2014).

A combination method is put forward to get the weight coefficient of individual methods (persistent, ARIMA, and ANN-based models), which uses the maximum entropy basis. RMSE of every single and combination model are compared for different forecasting horizons (1-6h) (Han et al., 2010).

A new model is proposed based on the combination of the variational mode decomposition (VMD) approach. The model decomposes the original wind power series to remove local features. The long short-term memory (LSTM) and deep belief networks based on particle swarm optimization (PSO-DBN) establish sub-series forecasting methods. Then, the multiple sub-series methods are merged by a nonlinear weighted combination approach based on PSO-DBN to create a hybrid method (Jiandong et al., 2022).

The characteristic of NWP and historical wind power data are extracted to obtain an accurate result using the feature-extracting approach. The model combines ELM and least squares SVM (LSSVM) methods. Then, critical parameters of models are optimized by developing a cuckoo search (ICS) to achieve a reliable result, specified as the pre-combined prediction value (PPA). Finally, the pre-combined forecasting method weights are allocated using a variance strategy to get the final predictions (Lu et al., 2021).

A hybrid deep learning method is proposed to get more accurate forecasts for a wind farm's very short-term wind power generation. The gated recurrent units and fully connected neural networks are combined to improve the performance using the Harris Hawks Optimization to tune the hyperparameters (Hossain et al., 2021).

Kalman filter-based approaches have an effective recursive algorithm to estimate the states of the system, reducing the MSE values, and developing the combined wind speed and power prediction methods. A recurrent multilayer perceptron (RMLP) method is put forward to predict one-step power production, training by an extended Kalman filter (EKF)- based back propagation (BP) through a time algorithm (Li, 2003; Tascikaraoglu et al., 2014).

3.2 The Forecasting Methods Under Study

3.2.1. Facebook Prophet

Facebook Prophet is a decomposable time series forecasting model to deal with the standard features of business time series, which was developed by the core data science team of Facebook in 2017 (Chung et al., 2014; Vishwas et al., 2020). Significantly, it is also constructed to consist of intuitive parameters that can be determined with fewer details of the underlying model. It is essential for the analyst to efficiently tune the model (Taylor et al., 2018). It has three main model components: trend, seasonality, and holidays, which are combined in the following equation:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \tag{3.2}$$

Here, the *g(t)* function is the trend function which is a piecewise linear or logistic growth curve to model non-periodic variations. Seasonality of the s(t) function represents periodic variations (e.g., weekly/yearly). The *h(t)* function is the impacts of the holiday, which happen on potentially irregular schedules. ϵ_t is an error term that accounts for idiosyncratic variations the model does not accommodate. The Prophet tries to fit numerous linear and nonlinear functions of time as components utilizing time as a regressor. Two trend models are used for Facebook applications: a nonlinear saturating growth model and a piecewise linear model. A nonlinear model, in its most basic form of the logistic growth model, is represented as:

$$g(t) = \frac{C}{1 + \exp\,(-k(t-m))} \tag{3.2}$$

Where C is the carrying capacity, that is, the maximum value of the curve, k represents the growth rate, which means the curve's steepness, and m is an offset parameter. If the k is tuned, m must also be adjusted to connect the endpoints of segments. A piecewise linear model with a constant rate of growth is then:

$$g(t) = \frac{C(t)}{1 + \exp\,(-(k + a(t)^T\delta)(t - (m + a(t)^T\gamma))} \tag{3.3}$$

Where δ and γ are vector rate correction that describes the variation in the rate that happens at the time s_j. The variation points result in the growth rate will change, and the trend model is:

$$g(t) = \;(k + a(t)^T\delta)t(m + a(t)^T\gamma) \tag{3.4}$$

γ_j is adjusted to $-s_j\gamma_j$ to make the function continuous. The seasonal effect could be represented with the following equation:

$$s(t) = \;\sum_{n=1}^{N}(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right)) \tag{3.5}$$

Where P represents a regular period, usually, because holidays and events do not follow a periodic pattern, their effects could not correctly be modeled by a smooth cycle (Caraka et al., 2018). Prophet allows the analyst to supply a custom list of past and future events. A window consisting of such days is considered distinctly, and extra parameters are fitted to model the impact of holidays and events. Before the analysis, the data should be split into training and testing (Taylor et al., 2018; Oo et al., 2019; Toharudin et al., 2020; Thiyagarajan et al., 2020; Asha et al., 2020).

3.2.2. Seasonal Autoregressive Integrated with the Moving Average (SARIMA)

Autoregressive (AR), autoregressive with moving average (ARMA), and autoregressive integrated with moving average (ARIMA) are the main statistical models which are used to predict time series (Garcia et al., 2019). Box and Jenkins introduced ARIMA in 1976 (Box et al., 1976). ARIMA can predict future values based on its historical values, which are lagged, and prediction errors lag (Fathi, 2019; Dubey et al., 2021). ARIMA has three main components: *p* is the number of lags observed values, d is the degree of differencing to make the predictors independent so the series can turn stationary, and q is the moving average (MA) degree. *d* should be chosen in what order autocorrelation (AC) reaches zero. p could be decided using the order of AR, which should be equivalent to the lags in the partial autocorrelation (PAC), which significantly cuts the limit set (Dubey et al., 2021; Liu et al., 2021). Non-seasonal ARIMA methods are demonstrated as ARIMA (p, d, q), and to choose these coefficients, mainly Box–Jenkins methodology is applied (Sharma et al., 2016; Garcia et al., 2019). The following equation defines linear expression:

$$y_t = \sum_{i=1}^{p} (\Phi_i y_{t-i}) + \sum_{j=1}^{q} (\theta_j y_{t-j}) + \varepsilon_t \tag{3.6}$$

Where y_t is the observation of time series at time t $\{y_t|$t=1, 2, …, N$\}$, Φ_i is the i^{th} autoregressive coefficient, θ_j is the j^{th} moving average coefficient, ε_t is the error term at time t (Fang et al., 2016; Garcia et al., 2019).

When the requirement of seasonal patterns arises in the time series, a seasonal term is added to the ARIMA model, which makes the model SARIMA SARIMA can be expressed as the following equation:

$$ARIMA\ (p, d, q) \times (P, D, Q)_S \tag{3.7}$$

While (p, d, q) represents the non-seasonal part, $(P, D, Q)_S$ represents the seasonal parts of the model. S refers to the number of periods per season. For instance, s can be defined as 12 for monthly observations because of 12 months a year. For hourly observations, s is generally set as 24 because of 24 hours a day (Chen et al., 2018; Siami-Namini et al., 2018; Dubey et al., 2021; Liu et al., 2021). The SARIMA model can be mathematically expressed as the following formula:

$$\varphi_p(B)\Phi_P(B^s)\nabla^d\nabla_s^D y_t = \theta_q(B)\Theta_Q(B^s)\varepsilon_t \tag{3.8}$$

Here, y_t is the observation of time series at time t $\{y_t|$t=1, 2, …, N$\}$. The expressions $\varphi_p(B)$ and $\theta_q(B)$ denote the order of the characteristic polynomial of non-seasonal AR and MA components. $\Phi_p(B^s)$ and $\Theta_Q\left(B^s\right)$ denote the seasonal AR and MA polynomial. ∇^d and ∇_s^Dindicate the non-seasonal and seasonal time series are differentiating operators, eliminating the non-seasonal and seasonal non-stationarity, respectively. B operates on y_t by shifting it at one point, denoted as the backshift operator. All operators and the polynomials are expressed as follows (Fıskın et al., 2019; Fathi et al., 2019; Dutta et al., 2021; Manigandan et al., 2021):

$$\varphi_p(B) = 1 - \sum_{i=1}^{p} \varphi_i B^i \qquad \Phi_P(B^s) = 1 - \sum_{i=1}^{P} \Phi_i B^{s,i} \tag{3.9}$$

$$\theta_q(B) = 1 - \sum_{i=1}^{q} \theta_i B^i \qquad \Theta_Q(B^s) = 1 - \sum_{i=1}^{Q} \Theta_i B^{s,i} \tag{3.10}$$

$$\nabla^d = (1 - B)^d \qquad \nabla_s^D = (1 - B^s)^D \tag{3.11}$$

3.2.3. SARIMA with Exogenous Factor (SARIMAX)

SARIMAX is an advancement of the SARIMA model, improved with the capability to incorporate exogenous (external features) variables (X) to improve its prediction performance. SARIMAX is usually defined as:

$$\varphi_p(B)\Phi_P(B^s)\nabla^d\nabla_s^D y_t = \beta_k x'_{k,t} + \theta_q(B)\Theta_Q(B^s)\varepsilon_t \tag{3.12}$$

Where $x_{k,t}$ are the vector with the k^{th} exogenous variables at time t and β_k represents the coefficient value of the k^{th} exogenous (X) time-series variables (Fıskın et al., 2019; Fathi et al., 2019; Dutta et al., 2021; Manigandan et al., 2021).

3.2.4. Gated Recurrent Unit (GRU)

GRU network model is an improved variation of the LSTM network based on optimizing the three-gate functions, which was first proposed by Cho et al. (2014). Besides the LSTM model, it deals with nonlinear time series problems and has a more compact and straightforward structure than the LSTM network. A single update gate is obtained by integrating the forget gate and input gate, and the memory cell and hidden state are mixed simultaneously. Consequently, the number of parameters decreases, and the training time is immensely shortened. The following equations show the governing equations of a GRU unit:

$$z_t = \sigma(W_Z x_t + U_Z h_{t-1} + b_Z) \tag{3.13}$$

Where z_t is the update gate and the $\sigma(\cdot)$ represents the sigmoid function expressed by:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{3.14}$$

r_t is the forget gate which is determined along with corresponding activation functions $\sigma(\cdot)$ by:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{3.15}$$

Where x_t is the input at time t, and h_{t-1} is the memory cell vector for time t-1. Then, by using the new memory cell vector $\tilde{h}_t$, the final output, that is, the current memory cell vector h_t at time t, is updated by:

$$h_t = (1 - z_t)h_{t-1} + z_t\tilde{h}_t) \tag{3.16}$$

Where $\tilde{h}_t$ is computed by:

$$\tilde{h}_t = tanh(Wx_t + U(r_t \otimes h_{t-1}) + b) \tag{3.16}$$

Where $\otimes$ represents the element-wise multiplication and $tanh(\cdot)$ is an activation function which is called the hyperbolic tangent function expressed by:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.14}$$

Where W_Z, W_r, W and b_Z, b_r, b are the assigned weights and bias, respectively.

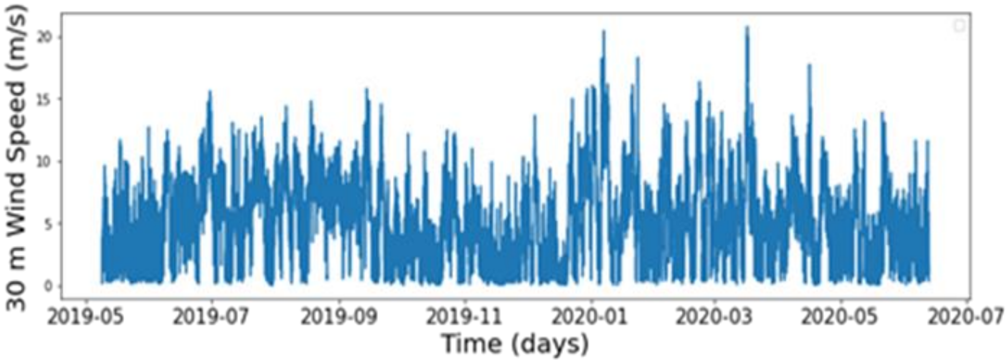
The update gate has a curial role in computing how much of the historical information will go through the current state, and a combination is adjusted between the new input and the previous information by the forget gate (Hosseini et al., 2020; Liu et al., 2021; Kisvari et al., 2021; Wu et al., 2022; Ji et al., 2022).

3.2.5. Long Short-term Memory (LSTM)

This study proposes a multivariate LSTM model, which is why the LSTM method is comprehensively explained in Section 4.1.

3.3 Dataset

The meteorological data to be used are collected from a 100 m IZTECH meteorological mast. The proposed model aimed to be validated using the production dataset from Urla, Kores Kocadag, and Germiyan Wind Farm. The collected wind time series are between 05/2019-06/2020 with 10 min temporal resolution from 30, 76, and 101 m. Figure 3.1 shows the wind speed time series. The time series is divided into the training and test data sets to verify the forecasting results.



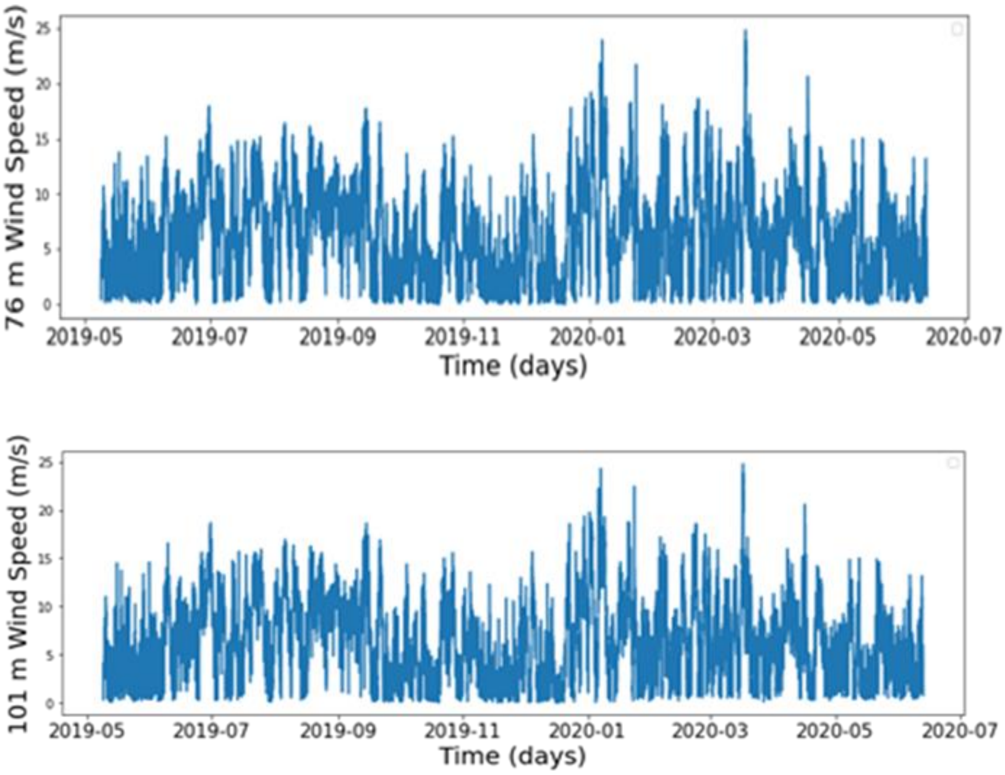


Figure 3.1. Wind speeds data sets at (a)30, (b)76, and (c)101 m

On the other hand, determining prevailing wind directions is essential in wind energy studies. Figure 3.2 shows the wind roses obtained based on the blow frequency of data taken from the meteorological mast's 28 m and 74 m heights. Although most winds come from the North (0º), intensity is observed in the range of 330-30º. Slightly winds are also seen between 180º and 210º. For this reason, the prevailing winds are from the North and South directions.

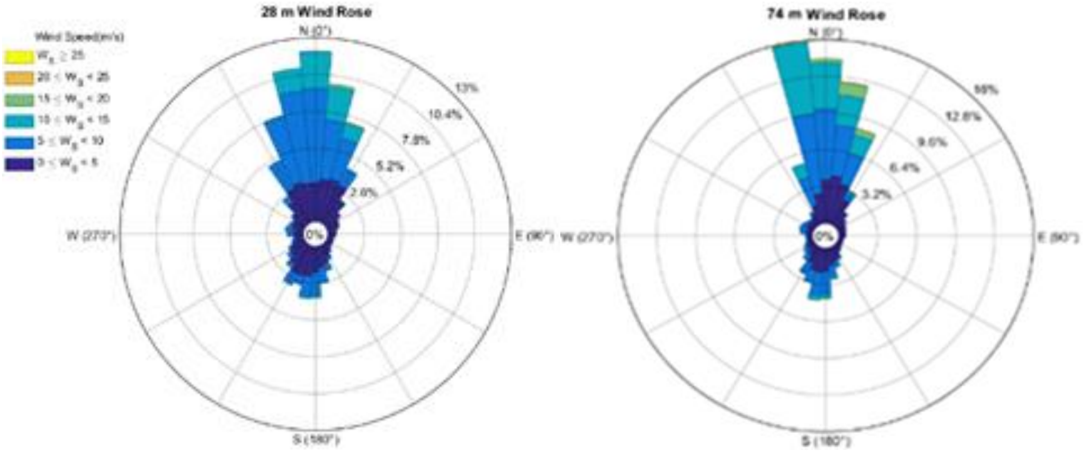


Figure 3.2. Wind roses at 28 m and 74 m

In addition to wind speed and direction, topography and meteorological variables also mainly affect the improvement of forecasting accuracy. Factors affecting wind speed, such as temperature, relative humidity, and barometric pressure, should also be considered (Liu et al., 2020). Temperature (°C), relative humidity (%), and barometric pressure (hPa) data are shown in Figure 3.3.

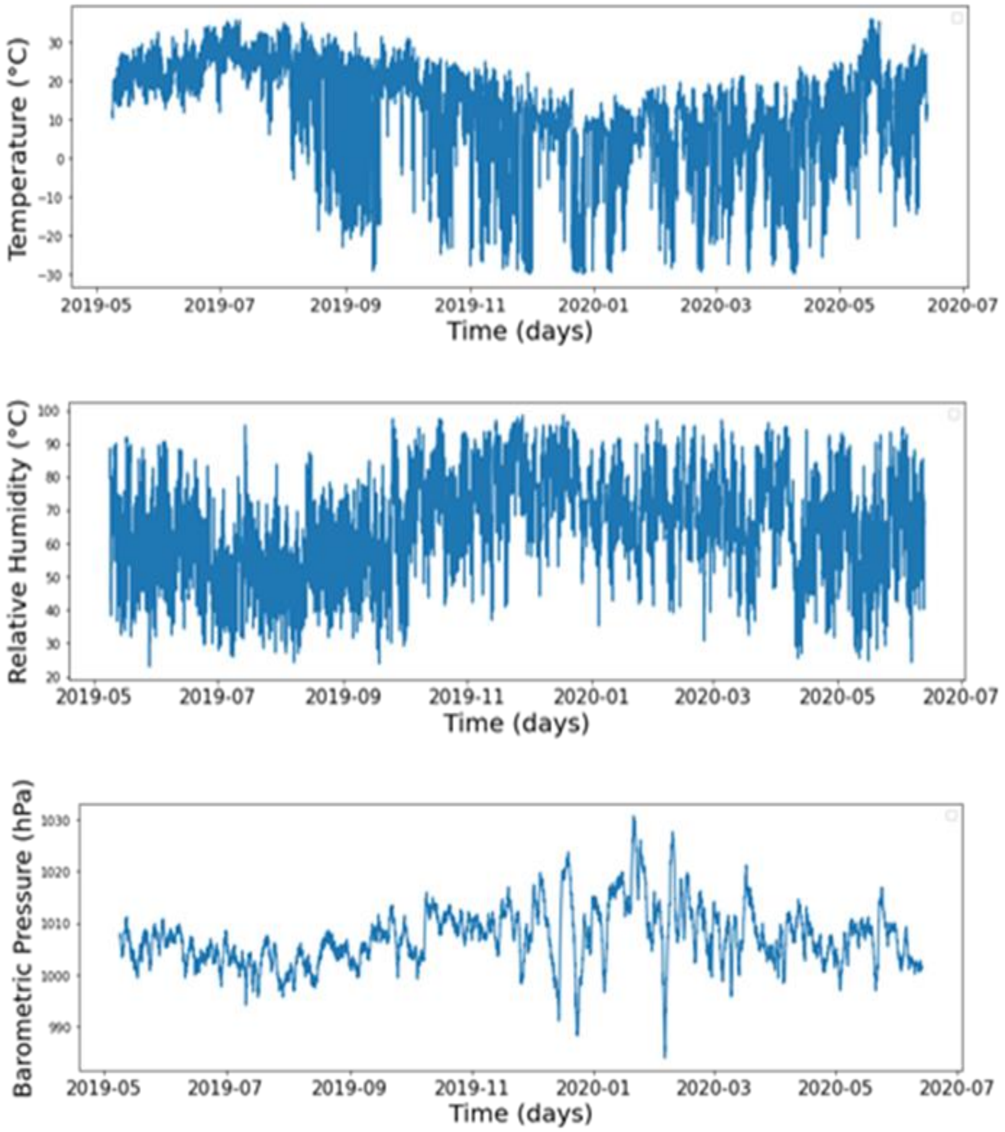


Figure 3.3. Temperature, relative humidity, and barometric pressure data sets

Here, the data is statistically analyzed and visualized for clear understanding. After visualizing the data, inspecting the nature of the data is the next step in the initial data analysis. The collected data are not naturally homogenous, and the distribution of these data features is not always normal. Using a histogram is an ideal option for understanding the data thoroughly. By looking at Figure 3.4, the real-time collected data are not normally distributed; the presence of skewness is in the data.

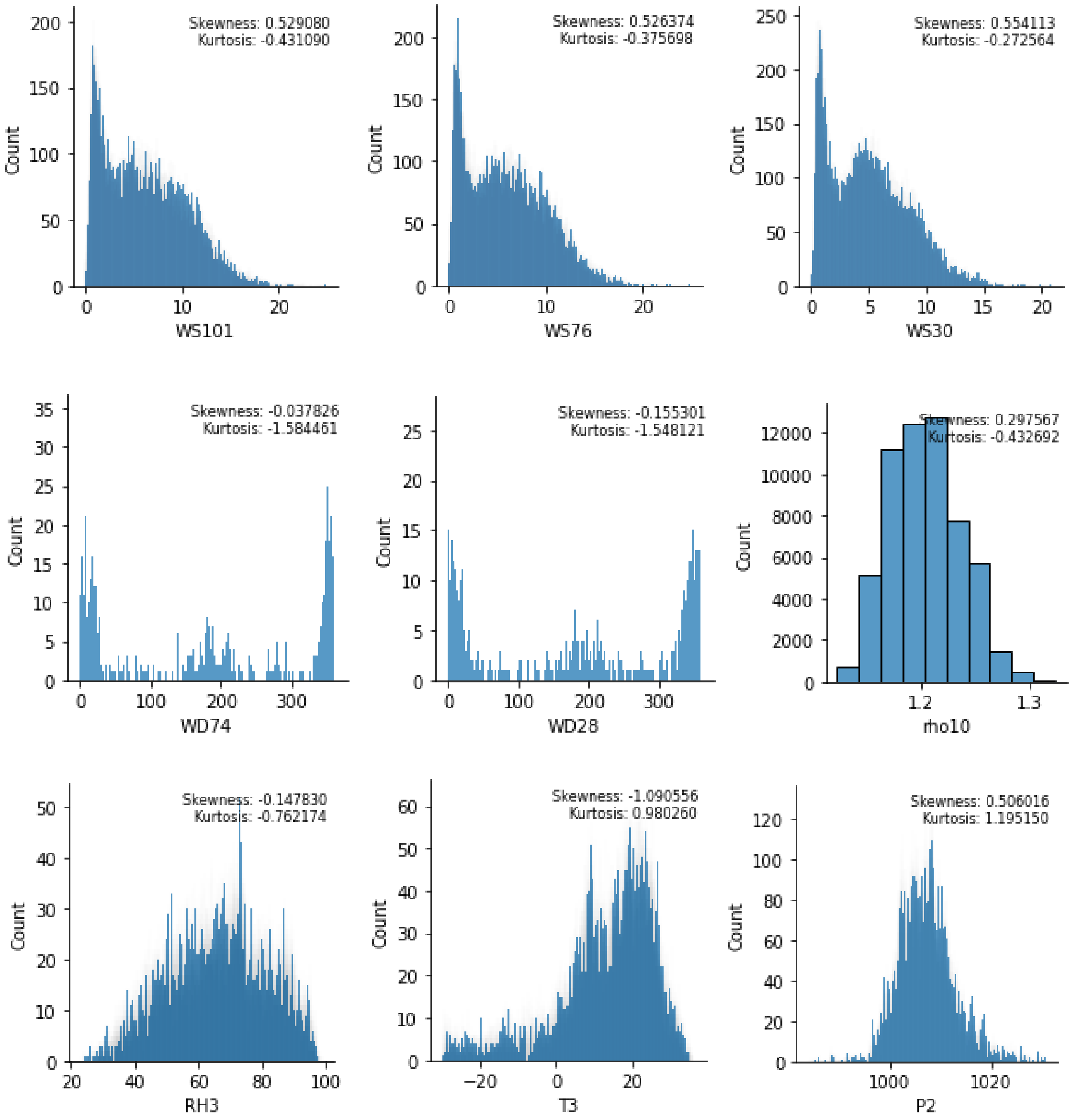


Figure 3.4. Histograms of selected parameters

From the descriptive statistics of the above figure, the positive and negative skewness and kurtosis indicate that distributions have an asymmetric characteristic. While the skewness's absolute value of wind speed data is more than 0.5, the data are positively skewed. And their kurtosis is lower than three, which is slightly flatter than a normal distribution. The wind direction data are relatively symmetrical, with skewness values between -0.5 and 0.5. Their kurtosis values are 1.58 and 1.55, which gives a flatter distribution than normal, where the values are moderately spread out. Air density, relative humidity, and pressure also show similar behavior. If the temperature data are considered, its skewness is -1.09, negatively skewed as the trail drags towards the left. Its kurtosis is 0.98, which is flatter than a normal distribution.

Table 3.1. Statistical descriptions of the time series dataset

Parameter	Count	Mean	Std	Min	25%	50%	75%	Max
WS101	57744	6.21	4.20	0.00	2.48	5.74	9.23	24.81
WS76	57744	6.01	4.07	0.00	2.39	5.60	8.91	24.84
WS30	57744	5.09	3.41	0.00	2.08	4.76	7.48	20.78
WD74	57744	355.34	138.72	0.00	21.28	187.61	343.81	360.00
WD28	57744	352.37	135.01	0.00	29.86	198.21	335.50	360.00
RH3	57744	66.00	15.82	22.92	54.05	66.78	78.18	98.78
T3	57744	12.62	13.09	-29.81	6.72	15.42	22.17	35.92
P2	57744	1007.21	5.62	983.87	1003.36	1006.86	1010.38	1030.65
rho10	57744	1.20	0.03	1.12	1.18	1.20	1.23	1.31

The statistical description of the count, mean, standard deviation and percentiles of selected features are reported in Table 3.1. The mean wind speeds at 30 m, 76 m, and 101 m are 5.09 m/s, 6.01 m/s, and 6.21 m/s, respectively. The mean ambient temperature is 12.62 °C with minimum and maximum values of -29.81 °C and 35.92 °C, respectively.

3.4 Performance Evaluation Metrics

To quantitatively investigate the prediction performance of forecasting models, the mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), and R square (R^2) as evaluation metrics which are expressed as follows:

$$MSE = \frac{1}{N}\sum_{t=1}^{N}(w(i) - \overline{w}(i))^2 \tag{1}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(w(i) - \overline{w}(i))^2} \tag{2}$$

$$MAE = \frac{1}{N}\sum_{t=1}^{N}|w(i) - \overline{w}(i)| \tag{3}$$

$$R^2 = 1 - \frac{\sum_{t=1}^{N}(w(i)-\overline{w}(i))^2}{\sum_{t=1}^{N}(w(i)-w\_m)^2} \tag{4}$$

where N is the number of samples, w(i) is the actual wind speed value, $\overline{\mathrm{w}}(\mathrm{i})$ is the forecast value of wind speed, and w_m is the average wind speed value (Tian et al., 2021).

3.5 Wind Speed Forecasting: A Case Study

In this study, an analysis of the forecasting performance of five approaches based on the results is performed. According to the literature review, the selected models will be investigated to obtain the best approach for wind speed prediction. After choosing the proposed model, the remaining methods will be used as benchmarks. A brief of forecasting results is provided to analyze the observations further. After determining the necessary parameters for each model, wind speeds are forecasted.

Figure 3.4 provides the absolute difference between the real-time data and prediction results for the wind speeds based on the multivariate Facebook Prophet, SARIMA, SARIMAX, GRU, and LSTM models. Facebook Prophet is a method that utilizes the general additive model to fit the nonlinear trends for time series with daily, weekly, and yearly seasonality (Asha et al., 2020). SARIMA is an ARIMA model including seasonal effects, and SARIMAX has an additional exogenous factor (Mangayarkarasi et al., 2021). They are the conventional statistical methods and are generally used as benchmarks. A deep neural network is a promising approach based on machine learning, and GRU and LSTM are the types of recurrent neural networks (RNN) (Hossain et al., 2021).

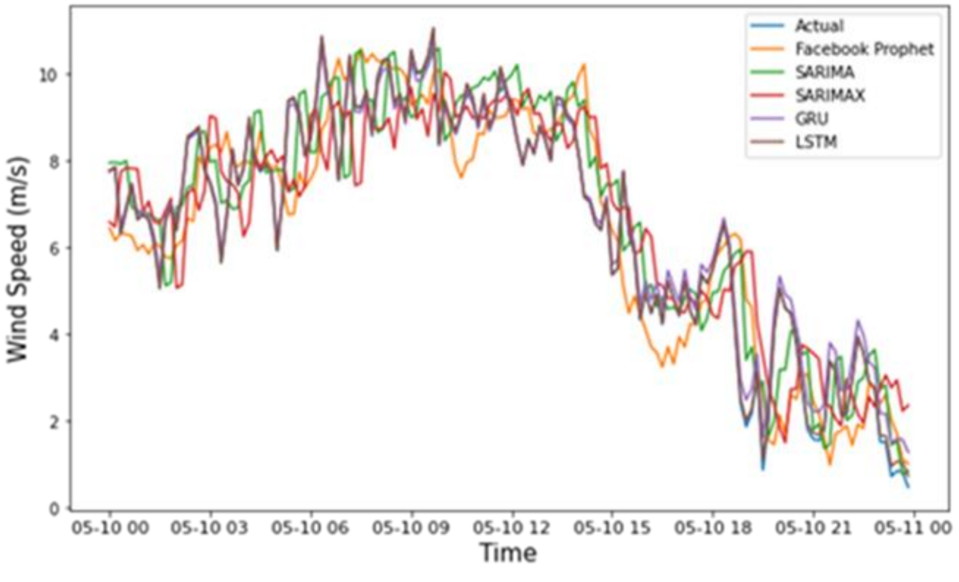


Figure 3.5. Daily wind speed forecasting results

The forecasting results are presented daily in Figure 3.5, and Figure 3.6 summarize the functional forecasting evaluation criteria to investigate the performance of each model. As seen in Figure 3.5, the prediction results of LSTM almost coincide with the real-time wind speed, which is also supported by the performance metrics. On the other hand, the predictions of other models follow the actual wind speed pattern for the day with accurate results. According to the adjusted R^2 scores, which indicate the goodness-of-fit, the LSTM model with the highest score of 0.9809 shows the best performance. That means the regression line of the LSTM model is fitted closest to the actual values compared with the other models. However, only the R^2 score may not be meaningful for the time series models. The LSTM model also shows the best performance with the lowest errors: MSE value of 0.2932, RMSE value of 0.4358, and MAE value of 0.3089, although the GRU model has relative values to the LSTM model.

On the other hand, the SARIMAX model cannot deeply understand the inherently chaotic nature of wind speed time series. The model shows the worst performance with the R^2 value of 0.7498, MSE value of 1.8584, RMSE value of 1.3632, and MAE value of 1.1004. Since wind speed affects wind power generation in the third order, SARIMAX cannot give good forecasting results.

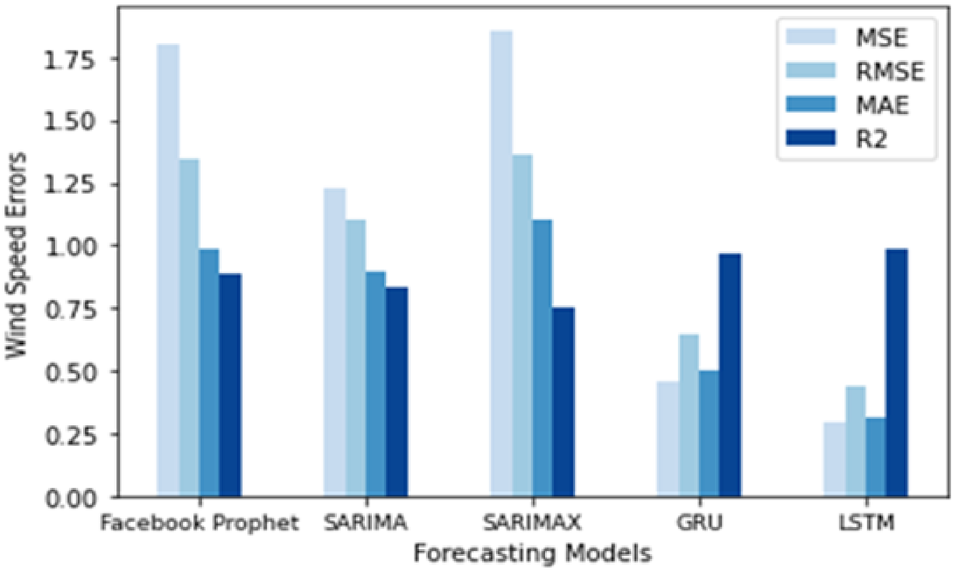


Figure 3.6. The evaluation criteria of wind speed forecasting for each model

Table 3.2. The evaluation criteria of wind speed forecasting for each model

Evaluation Criteria	**Facebook Prophet**	**SARIMA**	**SARIMAX**	**GRU**	**LSTM**
MSE	1.8035	1.2245	1.8584	0.4518	**0.2932**
RMSE	1.343	1.1065	1.3632	0.6407	**0.4358**
MAE	0.9814	0.8916	1.1004	0.5036	**0.3089**
R^2	0.8911	0.8351	0.7498	0.9718	**0.9809**

CHAPTER 4

THEORY AND METHOD

4.1 Long Short-Term Memory Networks (LSTM)

4.1.1 Recurrent Neural Network (RNN)

A long short-term memory network is a variant model of a recurrent neural network (RNN). On the other hand, the RNN is one of the artificial neural networks (ANN) with a deep learning structure especially adapted to sequence processing tasks. Its approach is to forecast the next element in a sequence of observations relative to previous steps. The compact illustration of an RNN is shown in Figure 4.1, where x_t is the input and element of the sequence, and y_t is the output value.

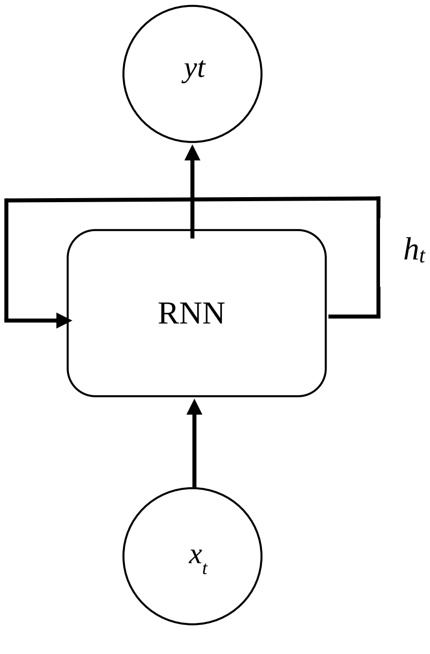


Figure 4.1. The compact illustration of RNN

A hidden state acting as memory is calculated when x_t is supplied to the RNN, denoted as h_t. Each time a new input updates its hidden state using this new input value and its previous hidden form to feed the RNN. In this way, the past information calculated from earlier elements of the sequence is effectively employed to notify the output created for the next segment of the sequence. Figure 4.2 shows how an RNN efficiently replicates the memory by using historical information to make an output for the next sequence element.

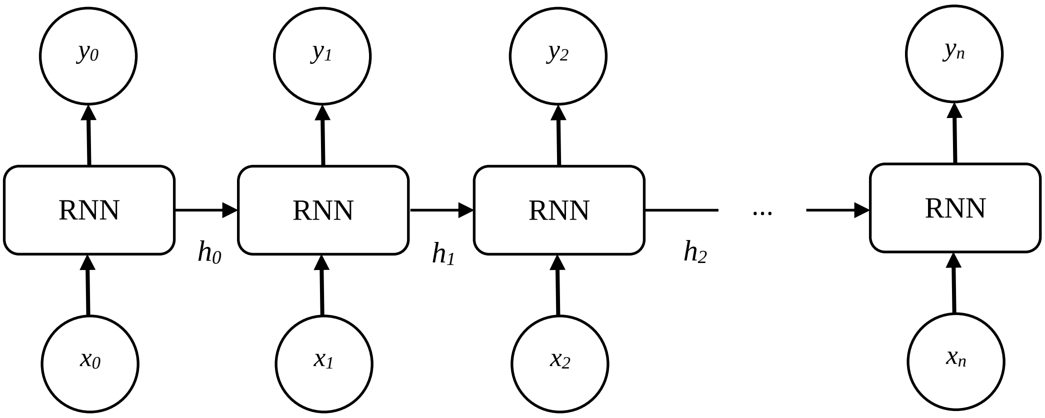


Figure 4.2. The expanded illustration of RNN

However, the network must be capable of memorizing information created as input in many steps before computing its final hidden state. This problem is due to the vanishing gradient, which is the function that asks the network how to adjust weights. That causes some weights to become too small or too large when the network unfolds for too many time steps. Thus, the vanishing gradient becomes very small, sometimes close to 0, so the network's weight is not changed. That causes the loss of information in the long term. The LSTM is a network structure designed to solve the long-term dependency problem (Graves et al., 2005; Cao et al., 2012; Li et al., 2017; Liu et al., 2018; Duan et al., 2020; Peixeiro, 2022).

4.1.2 The LSTM Architecture

As mentioned in Section 4.1.1, the LSTM network is a particular version of RNN developed to avoid the vanishing gradient problem, which could affect the network's learning. Currently, the method is widely utilized in the forecasting problem of time series with success. The LSTM is designed by developing a memory cell that stores the historical information for a longer time, while RNN has a short-term memory using the hidden state. As can be seen from Figure 4.2, the architecture of LSTM is more complicated than the basic RNN. The memory cell is denoted as *C*, resolving the vanishing problem. In this case, both the memory cell C_t and the hidden state h_t are passed on to the next element of the sequence. As described in Figure 4.2, the LSTM has three kinds of multiplicative units: a forget gate, an input gate, and an output gate in the memory cells (Li et al., 2018; Huang et al., 2019; Xie et al., 2021; Peixeiro, 2022).

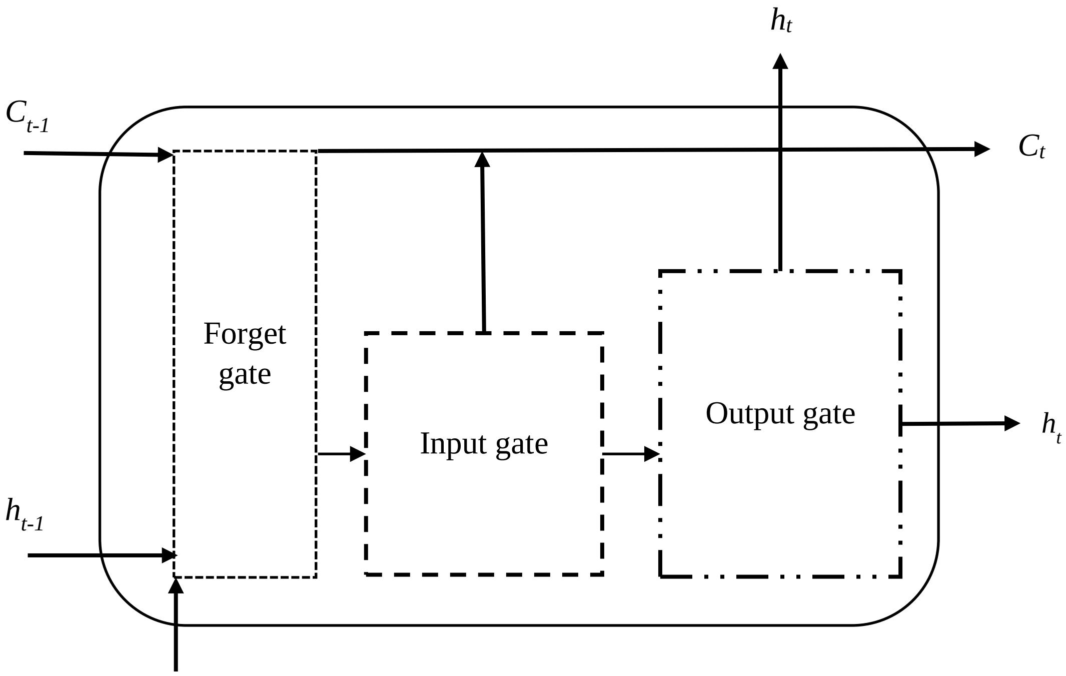


Figure 4.3. The architecture of LSTM

4.1.2.1 The Forget Gate

The LSTM structure starts with the forget gate, which controls how much historical information should be overlooked or how much new information should be kept in the network from both the historical and current values of the sequence. The different inputs entrance through the forget gate can be seen looking at Figure 4.3. Firstly, the past hidden state h_{t-1} and the present value of the sequence *xt* are fed into the forget gate. Then, h_{t-1} and x_t are combined and duplicated. While one copy is sent to the input gate, which will be studied in Section 4.1.2.1, the other copy goes through a sigmoid activation function. The sigmoid function is stated in equation 4.1 and displayed in Figure 4.3.

$$f(x) = \sigma = \frac{1}{1 - e^{-x}} \tag{4.3}$$

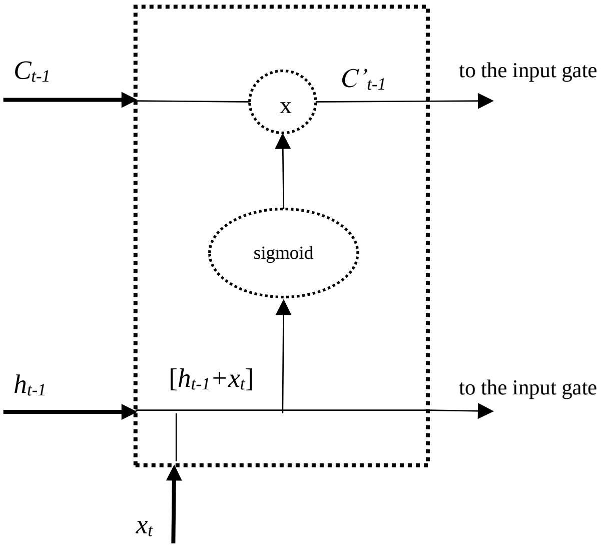


Figure 4.4. The forget gate

The sigmoid function has only output values between 0 and 1. Therefore, which information to save or forget could be defined by forwarding the hidden state and current sequence element across the function. An output value close to 0 means that the information could be overlooked. Contrastingly, output comparable to 1 means that the information must be saved. Then, the output and previous memory cell C_{t-1} are combined by applying pointwise multiplication, which creates an updated memory cell called C'_{t-1}. Presently, an updated memory cell, a copy of the combination of the past hidden state and current element of the sequence, is sent to the input gate (Li et al., 2018; Huang et al., 2019; Shao et al., 2021; Yuan et al., 2021; Peixeiro, 2022).

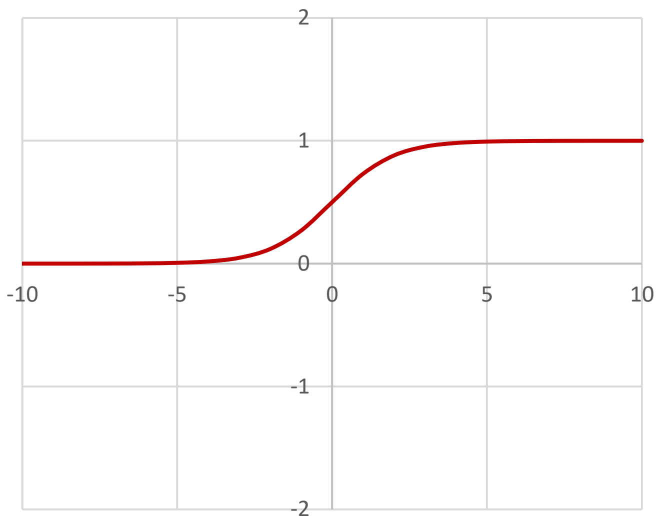


Figure 4.5. The sigmoid function

4.1.2.2 The Input Gate

The information proceeds to the input gate after passing through the forget gate, the stage where the network defines which information is appropriate from the current element of the sequence. The memory cell is updated here, leading to the final cell state. Figure 4.5 visualize the input gate configuration.

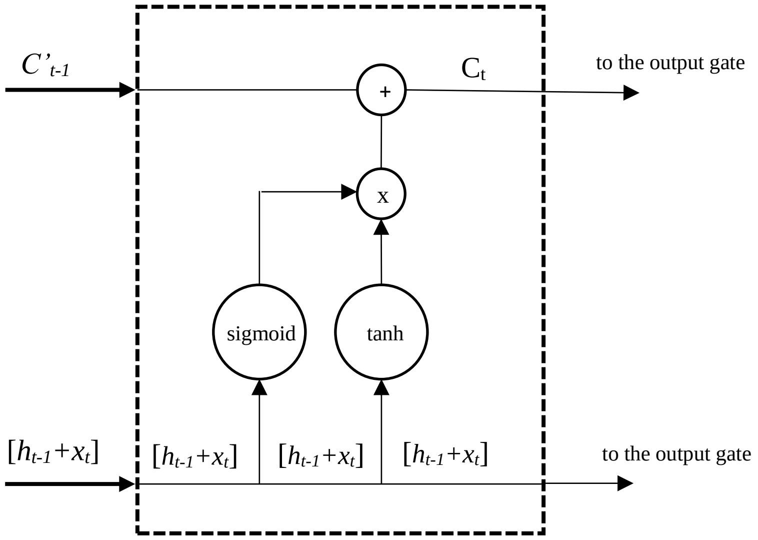


Figure 4.6. The input gate

After duplicating the past hidden state and current element of the sequence again, they are sent through the sigmoid and a hyperbolic tangent (tanh) activation function. The hyperbolic tangent (tanh) activation function is stated in equation 4.2 and visualized in Figure 4.5.

$$f(x) = tanh = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{4.2}$$

As with the forget gate, the sigmoid defines which information to keep or forget, while the tanh function organizes the network to keep it computationally effective. The hyperbolic function is visualized in Figure 4.6. Pointwise multiplication is used to combine the results of both operations. Then, it is utilized to update the memory cell using pointwise addition, resulting in the final memory cell C_t. Subsequently, the final memory cell C_t and the same combination $[h_{t-1}+x_t]$ are sent to the output gate.

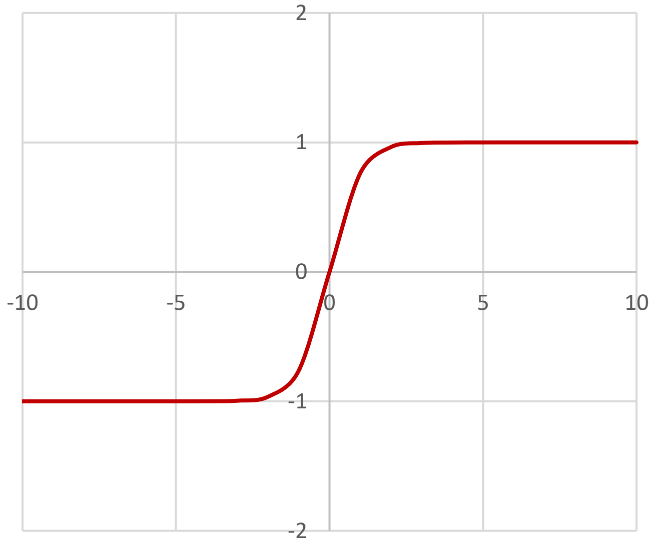


Figure 4.7. The hyperbolic tangent (tanh) function

Figure 4.5 shows that the hyperbolic tangent function can only output values between -1 and 1, which allows for the regulation of the network. That ensures values do not get uncontrollably large and allows training the model to be computationally effective. Therefore, the information from the current element in the sequence is added to the network's long memory in the input gate. Then, the newly updated memory cell is sent to the output gate (Li et al., 2018; Huang et al., 2019; Shao et al., 2021; Yuan et al., 2021; Peixeiro, 2022).

4.1.2.3 The Output Gate

C_t is finally used to process the current element of the sequence, utilizing what it has learned from the past elements. Also, the output gate gives a result to the output layer or determines new information to be delivered to the operation of the following element in a sequence.

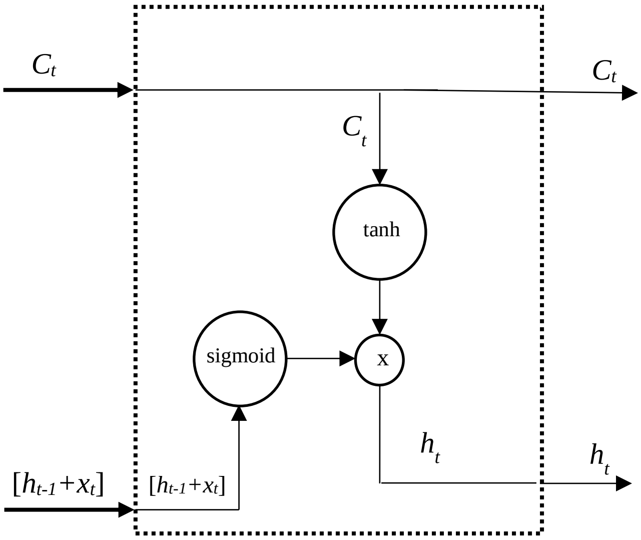


Figure 4.8. The output gate

Figure 4.5 shows that a sequence's past hidden state and current element are sent into the sigmoid function. As stated before, it is known that the output value will be between 0 and 1, and it decides whether the information is saved or not.

Besides, the memory cell passes the tanh function. The results of these processes are combined utilizing pointwise multiplication, producing an updated hidden state h_t. This step is that the historical information kept in the memory of the network is used to process the information of the current sequence element. Then, the present hidden state is sent off the output gate. It will either be directed to the output layer of the network or to the next LSTM neuron, which deals with the following sequence element. The same is valid for the memory cell C_t (Li et al., 2018; Huang et al., 2019; Shao et al., 2021; Yuan et al., 2021; Peixeiro, 2022).

4.2 ERA5 Dataset

The historical weather observations are combined by reanalysis using an atmospheric weather model (Hayes et al., 2021). Reanalysis employs historical data to regulate the original model parameters and then re-predict (Liao et al., 2022). ERA5 is a comprehensive reanalysis dataset produced by European Centre for Medium-range Weather Forecast (ECMWF) for the global climate and weather in recent years. The “comprehensive” data integrates satellite remote sensing, station observations, such as wind profiler, ship, Synop, radar, Metar, aircraft, radio sounding, and numerical model simulation (Gualtieri et al, 2021; Liao et al., 2022). Most recent studies have used wind data reanalyzed due to their global coverage, availability for the long term, and free access to any location. Although several reanalysis datasets are available, ERA5 has a higher spatial and temporal resolution (Nefabas et al., 2021; Ahmad et al., 2022). Available data is currently from 1959 to 5 days behind real-time. ERA5 offers hourly estimations for many atmospheric, ocean-wave, and land-surface parameters.

ERA5 are updated version of ERA-Interim reanalysis, which was terminated in 2019. ERA5 implements several improvements over the former product, including an increase in horizontal and vertical resolution and time step. Additionally, ERA5 brings an uncertainty estimate that was not existing in ERA-Interim. Besides the increased time resolution to 1 h, the most significant feature of using ERA5 reanalysis datasets for wind energy is the availability of a higher number of parameters, especially wind speed at 100 m.

Dataset is rearranged on a regular latitude-longitude grid of 0.25 degrees for the reanalysis and 0.5 degrees for the uncertainty estimation. There are hourly and monthly products on both pressure and single levels. Table 4.1 lists the data description of ERA5 hourly data on pressure levels from 1959 to the present (Hersbach et al.,2018).

Table 4.1. The description of ERA5 hourly data on pressure levels from 1959 to the present (Source: Hersbach et al.,2018)

DATA DESCRIPTION	
Data Type	Gridded
Projection	Regular latitude-longitude grid
Horizontal coverage	Global

(Cont. of the next page)

(Cont. of the next page)

Horizontal resolution	Reanalysis: 0.25° x 0.25°
Vertical coverage	1000 hPa to 1 hPa
Vertical resolution	37 pressure levels
Temporal coverage	1959 to present
Temporal resolution	Hourly
File Format	GRIB

The downloaded ERA5 data is obtained by sub-region extraction, and the four coordinates are 38.25 26.25, 38.25 26.50, 38.00 26.25, and 38.00 26.50. The dataset is at an atmospheric level that corresponds to 100 m in height. The data is a NetCDF file and consists of date-time, latitude, longitude, temperature (K), air pressure (kPa), U-component (m/s), and V-component (m/s) of wind. The data is from 01 January 2019 to 31 December 2020. Table 4.2 lists the names, units, and descriptions of the main variables chosen for wind speed and power forecasting.

Table 4.2. The name, units, and descriptions of the main variables under study (Source: Hersbach et al., 2018)

MAIN VARIABLES		
Name	**Units**	**Description**
Temperature	K	The temperature in the atmosphere has kelvin (K) units. It is available on multiple atmosphere levels.
U-component of wind	m s^{-1}	This parameter is the horizontal speed of air moving towards the east. The air is moving toward the west when it has a negative sign. Combination with the V-component of wind gives the horizontal wind speed and direction.
V-component of wind	m s^{-1}	This parameter is the horizontal speed of air moving towards the north. When it has a negative sign, air moves toward the south. Combination with the U-component of wind gives horizontal wind speed and direction.

4.2 Wind Power Forecasting Methodology

This study proposes a multivariate LSTM network for wind speed and power forecasting. The framework of the wind power forecasting methodology is described in Figure 4.9. As mentioned in Section 4.2, an ERA5 dataset is used because there is no available SCADA data for Kores Kocadag and Germiyan wind farms.

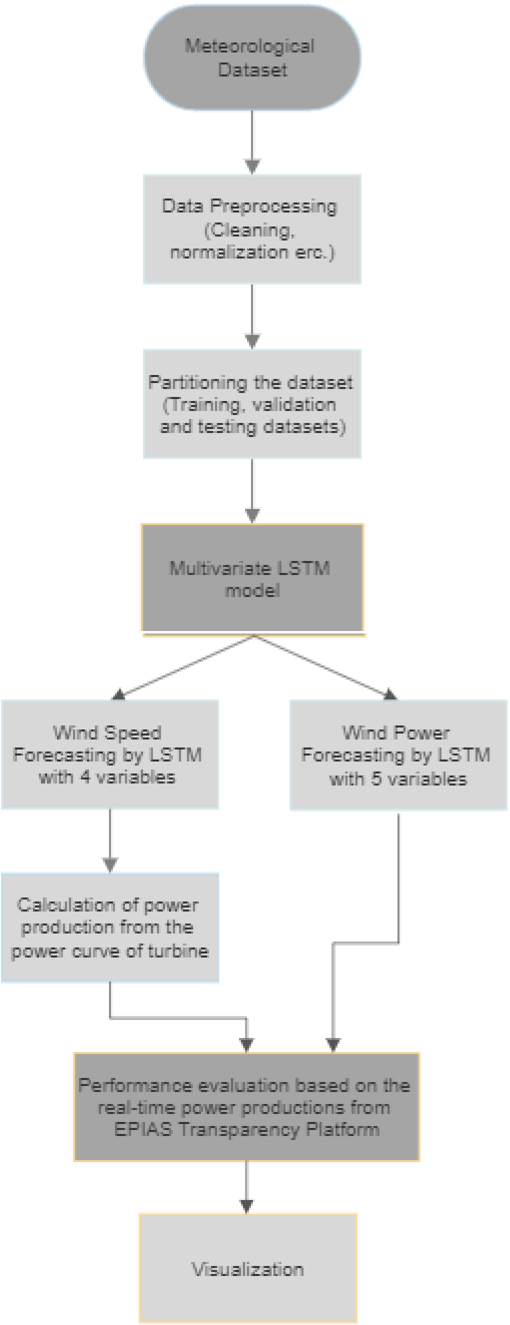

Meteorological Dataset
Data Preprocessing (Cleaning, normalization erc.)
Partitioning the dataset (Training, validation and testing datasets)
Multivariate LSTM model
Wind Speed Forecasting by LSTM with 4 variables
Wind Power Forecasting by LSTM with 5 variables
Calculation of power production from the power curve of turbine
Performance evaluation based on the real-time power productions from EPIAS Transparency Platform
Visualization

Figure 4.9. The framework of the wind power forecasting

CHAPTER 5

RESULTS

5.1 ERA5 Dataset Validation

It is necessary to validate the ERA5 dataset with the IZTECH meteorological mast. Firstly, the dataset is controlled by whether there is a time shift. The process is performed by plotting each parameter of both datasets on the same graph. This situation is handled both annually and monthly.

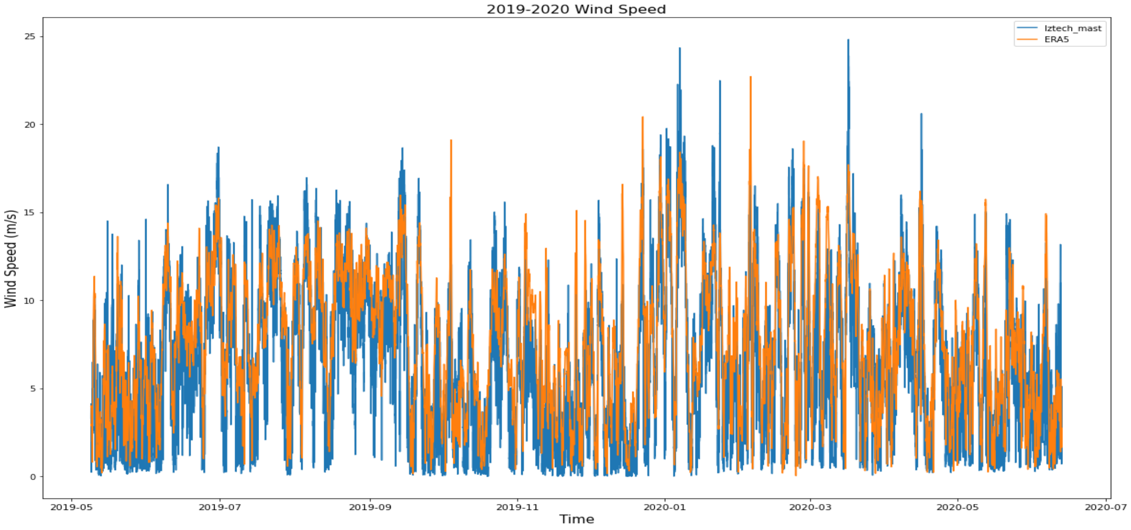


Figure 5.1. The wind speed data of ERA5 vs. IZTECH met. mast

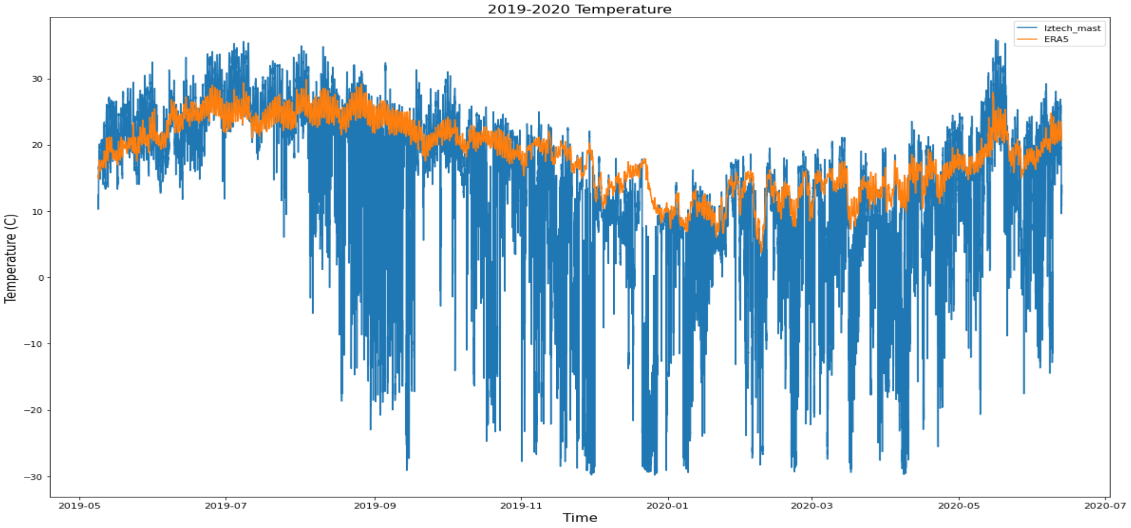


Figure 5.2. The temperature data of ERA5 vs. IZTECH met. mast

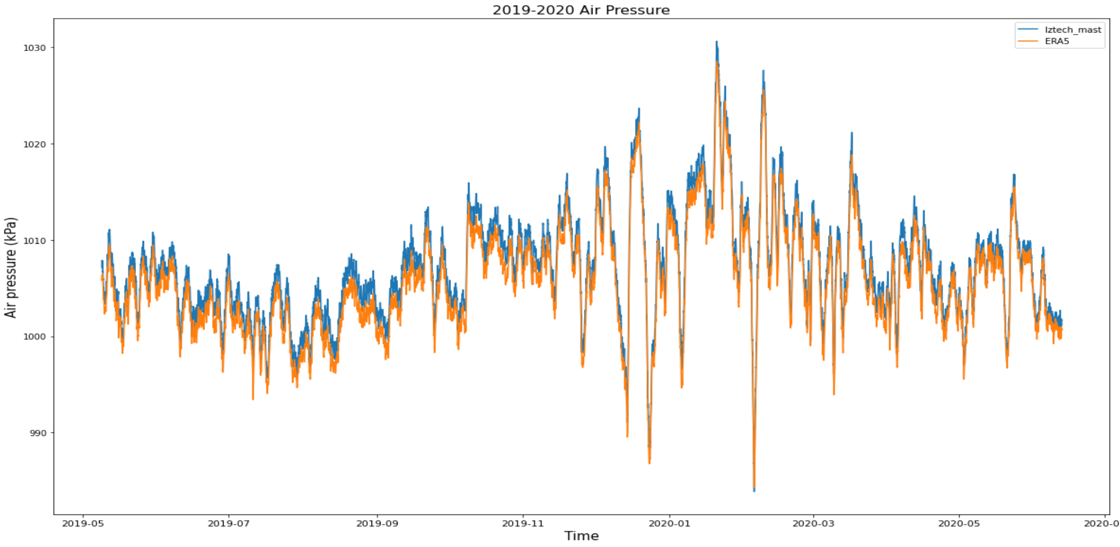


Figure 5.3. The air pressure data of ERA5 vs. IZTECH met. mast

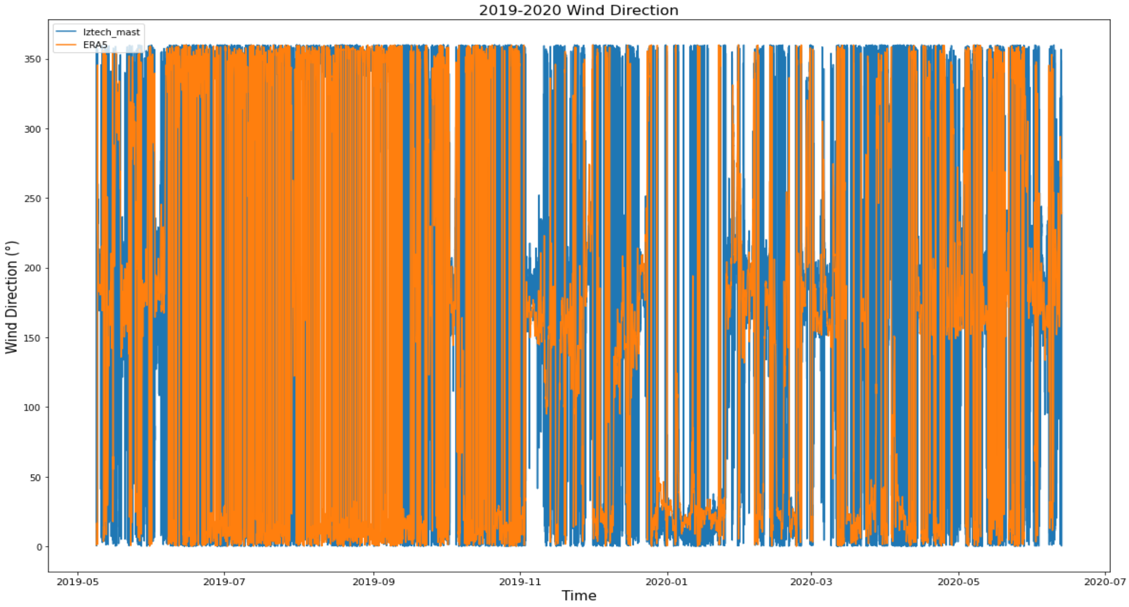


Figure 5.4. The wind direction data of ERA5 vs. IZTECH met. mast

The hourly average wind directions are calculated and split into twelve sectors. The first and second sector filters the dataset. Firstly, the dataset is filtered lower than 0.35 m/s because the anemometer cannot read values lower than 0.35 m/s. Secondly, the dataset is also filtered greater than 2.5 m/s because the cut-in wind speed of a wind turbine is mostly greater than 2.5 m/s. Then, the scatter plot is obtained from the wind speed data of ERA5 and IZTECH meteorological mast, illustrated in Figure 5.5. The coefficient of determination R^2 0.7596 denotes the correlation between the ERA5 and IZTECH meteorological mast.

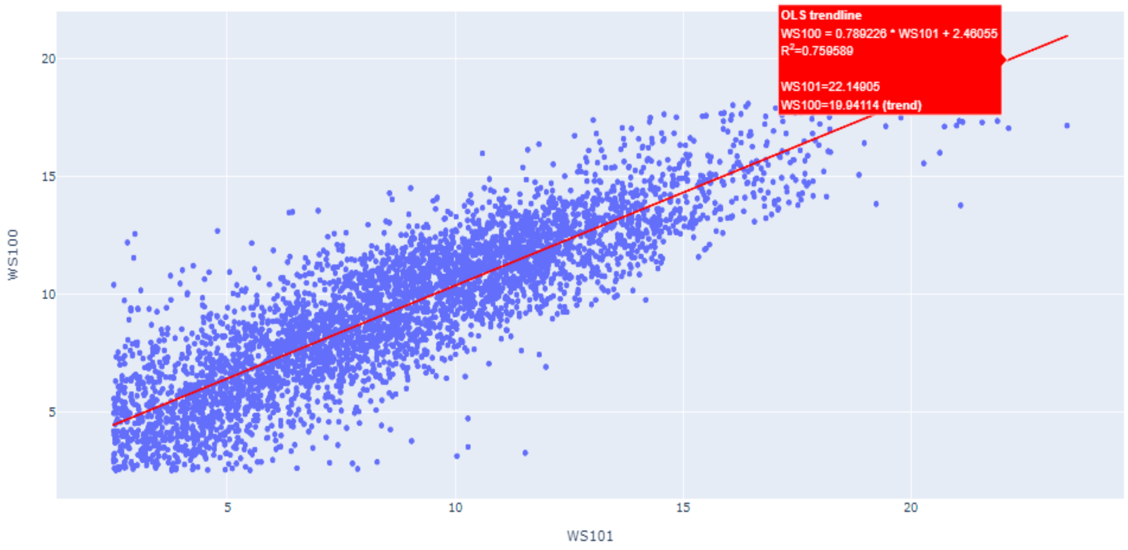


Figure 5.5. The scatter plot of wind speed data of ERA5 vs. IZTECH met. mast

5.2 Wind Power Forecasting Using the Power Curve of Turbines

This section compares and analyzes the wind power forecasting results using the power curve of turbines for three operational wind farms. The R-square, RMSE, MAE, and MAPE values are determined for each operational wind farm to verify the method's performance. To demonstrate the error measures more intuitively, a radar chart for R-square, a horizontal histogram for RMSE, an area chart for MAE, and a vertical histogram for MAPE are illustrated. Two years dataset, an EAR5 dataset mentioned before, is used to predict the wind speed predictions of last month. The dataset is from 1st January 2019, 00:00 AM to 30th December 2020, 23:50. It includes the four meteorological variables: wind speed, direction, temperature, and air pressure. Thus, a multivariate LSTM model can be obtained using four variables. The LSTM network is trained, validated, and tested using the training, validation, and testing data set. After getting the wind speed forecasting results, the wind power generations are calculated using the fitting curve equation of the power curves of the turbines between the cut-in and cut-out wind speed values. The lower than the cut-in wind speed and the higher than the cut-off wind speed are assumed as not producing wind power and accepted as 0. Wind farms' licensed and installed capacities are also considered while calculating wind power production. If the wind power generation forecasts exceed the licensed installed capacity, it is districted with the licensed installed capacity.

The prediction results of the Urla, Kores Kocadag and Germiyan Wind Farms are illustrated in Figure 5.14, Figure 5.15, and Figure 5.16 show, respectively, as the one-month, the last 350, and the previous 100 samples.

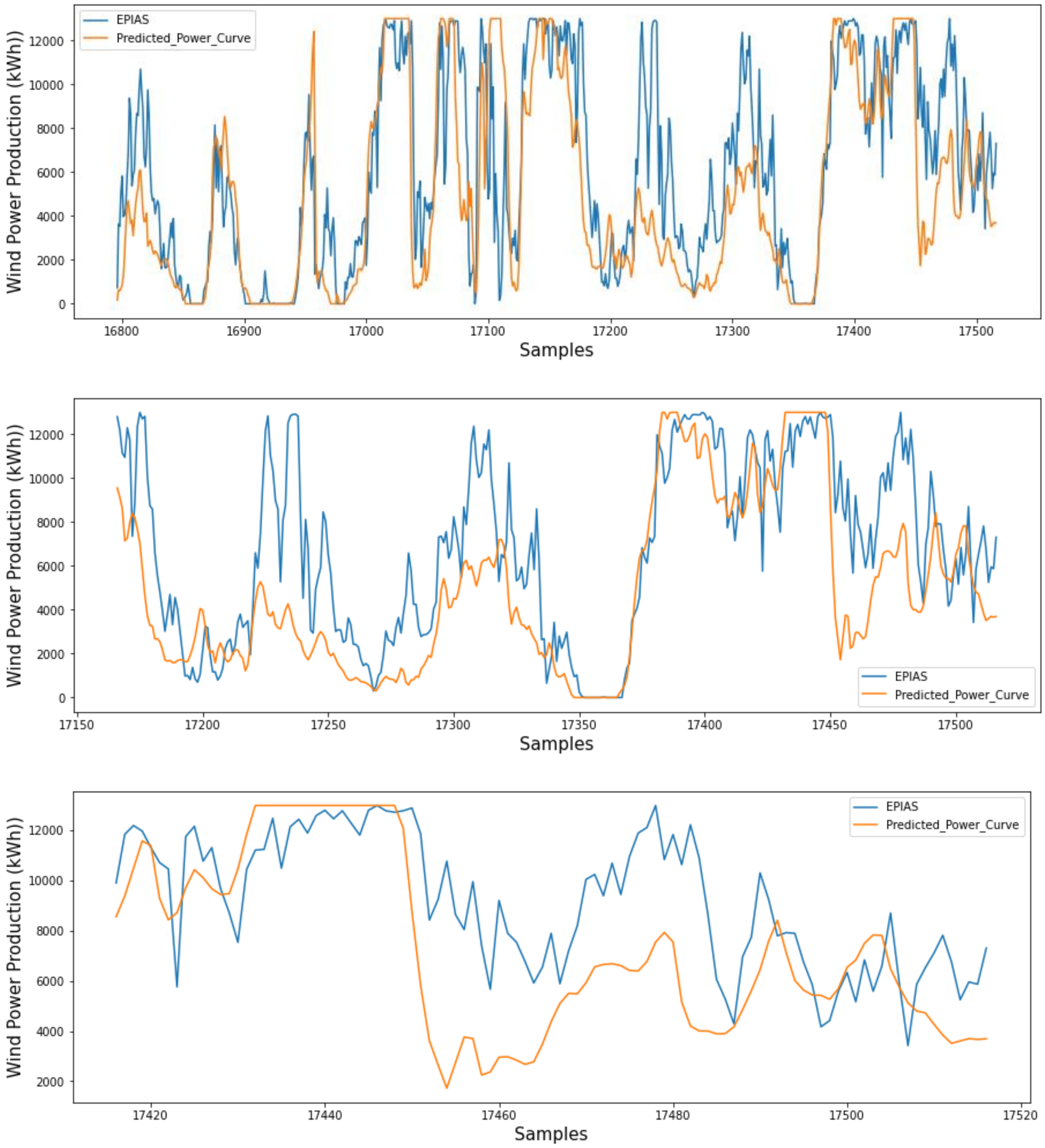


Figure 5.14. The EPIAS versus the forecasts obtained from the power curve of the turbine for Urla Wind Farm (a) the samples of one-month comparison (b) the last 350 samples comparison (c) the last 100 samples comparison

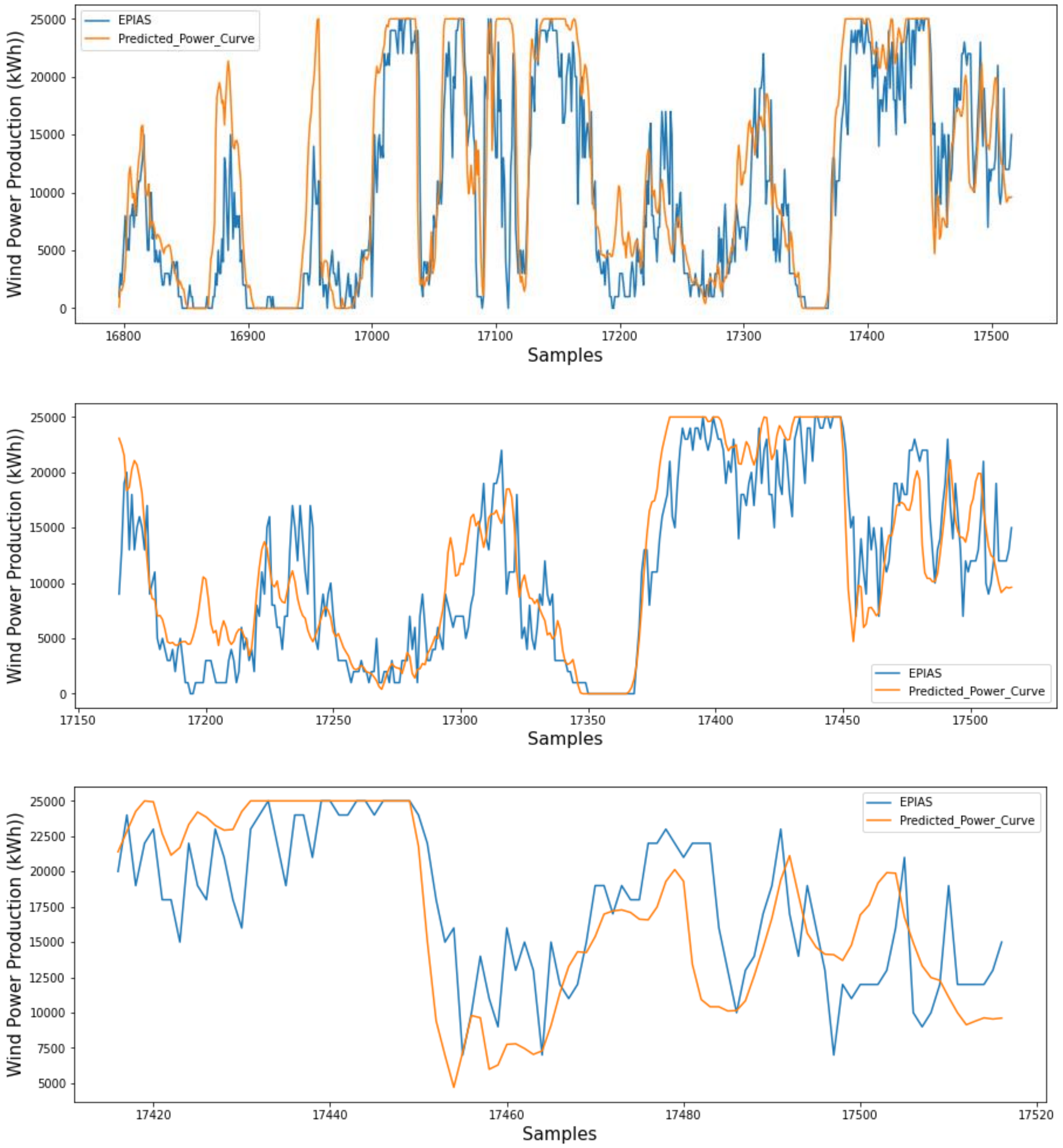


Figure 5.15. The EPIAS versus the forecasts obtained from the power curve of the turbine for Kores Kocadag Wind Farm (a) the samples of one-month comparison (b) the last 350 samples comparison (c) the last 100 samples comparison

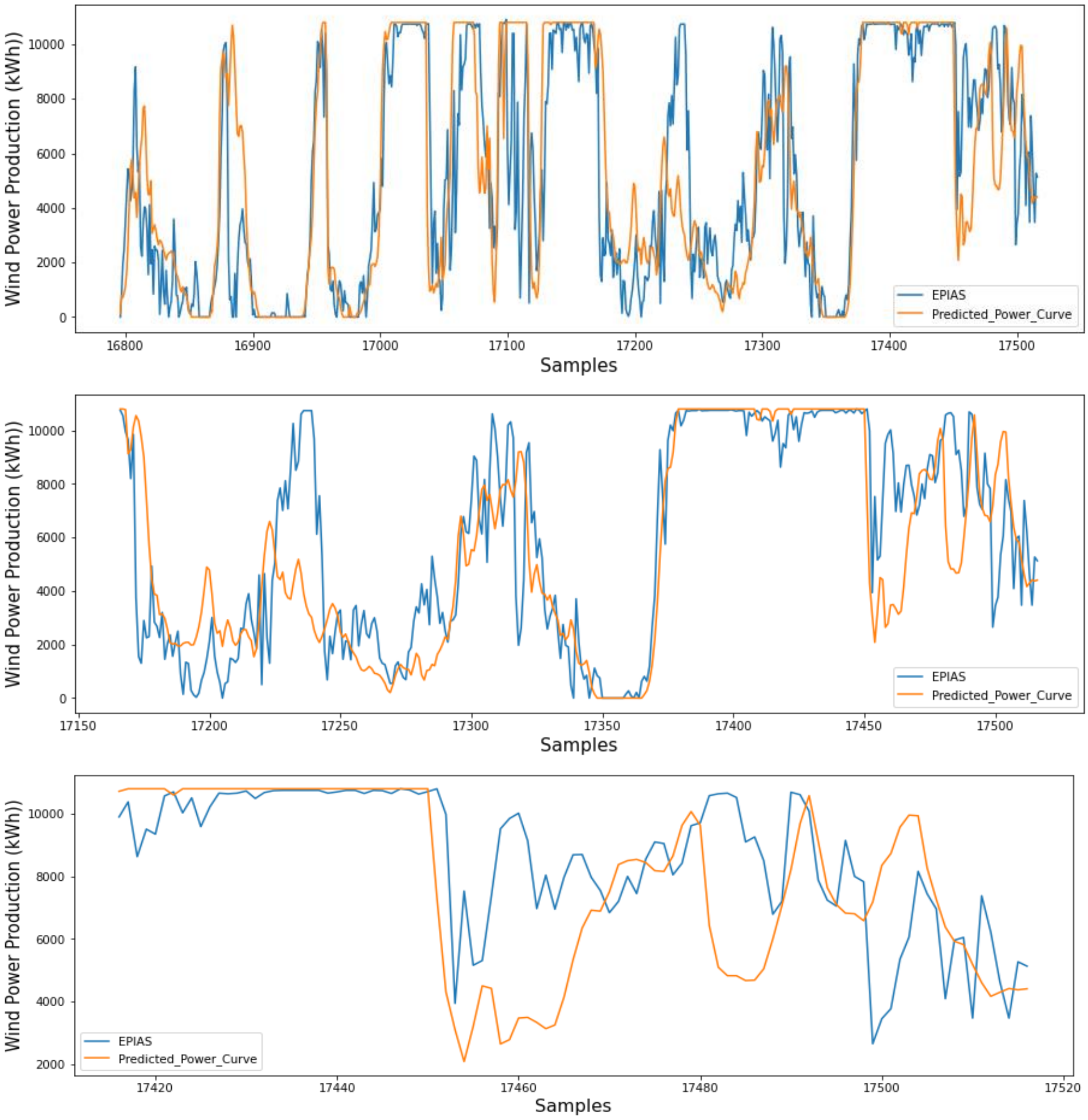


Figure 5.16. The EPIAS versus the forecasts obtained from the power curve of the turbine for Germiyan Wind Farm (a) the samples of one-month comparison (b) the last 350 samples comparison (c) the last 100 samples comparison

Table 5.2 lists the scores of the evaluation criteria. As mentioned in the previous section, MAPE can be the best indicator for comparison. Mainly, Urla Wind Farm exhibits a minor mean absolute percentage error compared with the other wind farms, with a value of 45.98%, which means the accuracy is higher than the other wind farms. Figure 5.17, Figure 5.18, and Figure 5.19 show that the forecast wind power generation of the Urla Wind Farm has more similarity than the other wind farms' actual wind power generation with a significant difference. The correlations denoted by the coefficient of determination R^2 are 0.5979, 0.7389, and 0.6682, respectively, in the Urla, Kores Kocadag, and Germiyan wind farms.

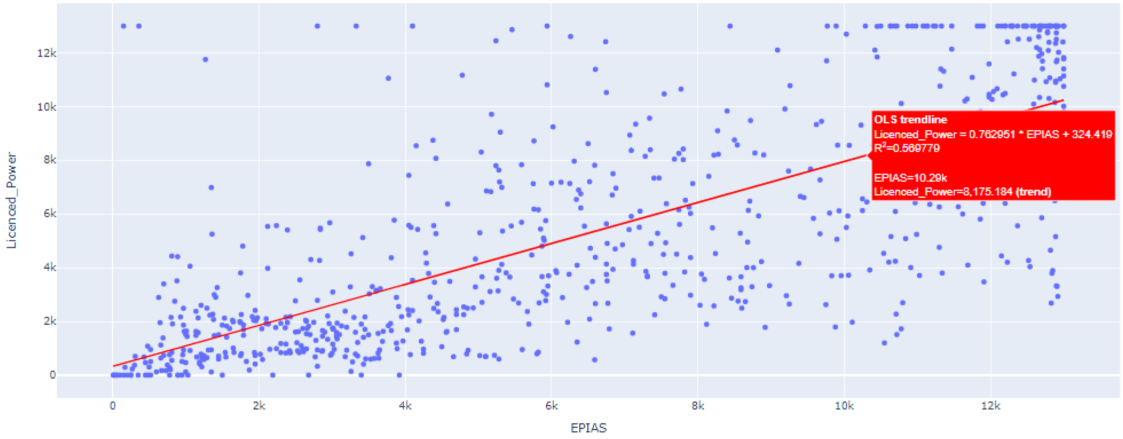


Figure 5.17. The correlation plot between the wind power productions of EPIAS and the forecasts obtained from the power curve of the turbine for Urla Wind Farm (R^2 = 0.687306)

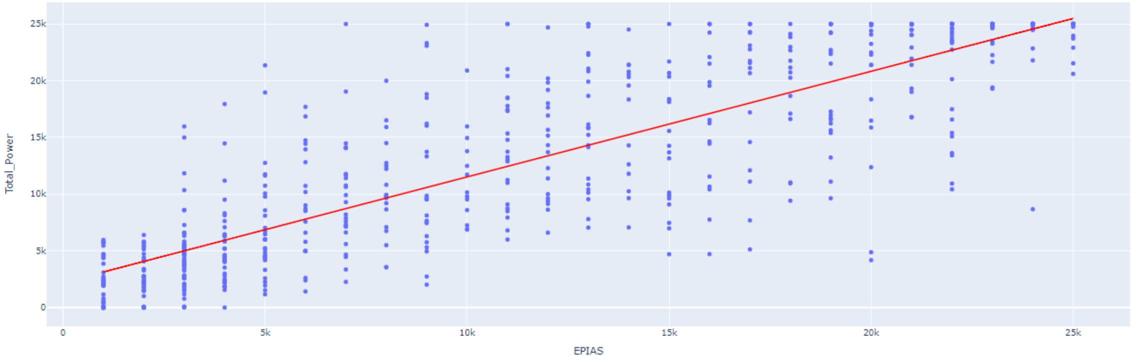


Figure 5.18. The correlation plot between the wind power productions of EPIAS and the forecasts obtained from the power curve of the turbine for Kores Kocadag

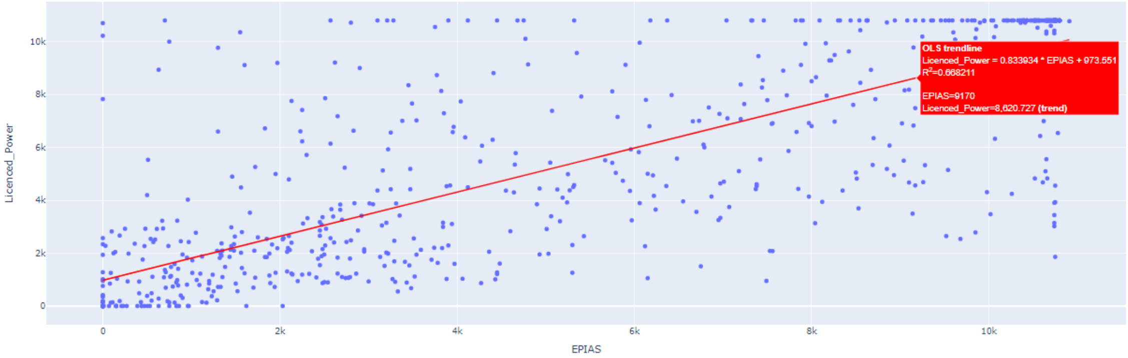


Figure 5.19. The correlation plot between the wind power productions of EPIAS and the forecasts obtained from the power curve of the turbine for Germiyan Wind Farm

Lastly, Urla Wind Farm exhibits 45.98% of the MAPE, followed by Germiyan Wind Farm, with 46.24%. Kores Kocadag Wind Farm occupies the minor rank with 51.12%.

Table 5.2. Statistical measurements of three operational wind farm data using the actual and forecasts obtained from the power curve of turbine

Evaluation Criteria	**Urla**	**Kores Kocadag**	**Germiyan**
R-square	0.5979	0.7389	0.6682
RMSE (kWh)	3115.72	4948.16	2500.74
MAE (kWh)	2278.75	3317.86	1571.1
MAPE (%)	45.98	51.12	46.24

5.3 Wind Power Forecasting Using a Multivariable LSTM Network

This section compares and analyzes the wind power forecasting results using a multivariate LSTM Network for three operational wind farms. To evaluate the proposed model's performance, the R-square, RMSE, MAE, and MAPE are calculated for each operational wind farm. To demonstrate the error measures more intuitively, a radar chart for R-square, a horizontal histogram for RMSE, an area chart for MAE, and a vertical histogram for MAPE are displayed. Two years dataset, an EAR5 dataset mentioned before, is used to predict the wind power predictions of last month. The dataset is from 1st January 2019, 00:00 AM to 30th December 2020, 23:50. It includes the four meteorological variables: wind speed, wind direction, temperature, and air pressure. The wind power generations are also added as a variable for the LSTM model. Thus, a multivariate LSTM model can be obtained using five variables. The LSTM network is trained, validated, and tested using the training, validation, and testing data set.

Here, the wind power productions are recorded by the SCADA system of Urla farm, and data sets from 1st January 2019, 00:00 AM to 30th December 2020, 23:50, are registered every 10 min. Thus, the 10 min power production dataset consists of 105120 observations. Firstly, the wind speed data of 730 days are averaged into the hourly data set because the dataset is also compared with the EPIAS data set. The mentioned process is applied to the EPIAS and SCADA power production values.

Figure 5.6. and Figure 5.7 shows the one month, the last 350, and the previous 100 samples of actual and forecasted wind power of the Urla Wind Farm, respectively, the SCADA and EPIAS data using the LSTM model. The real-time wind power generations are obtained from the EPIAS Transparency Platform. The difference between the actual and forecast values can be understood more intuitively from Figure 5.6a and Figure 5.7a.

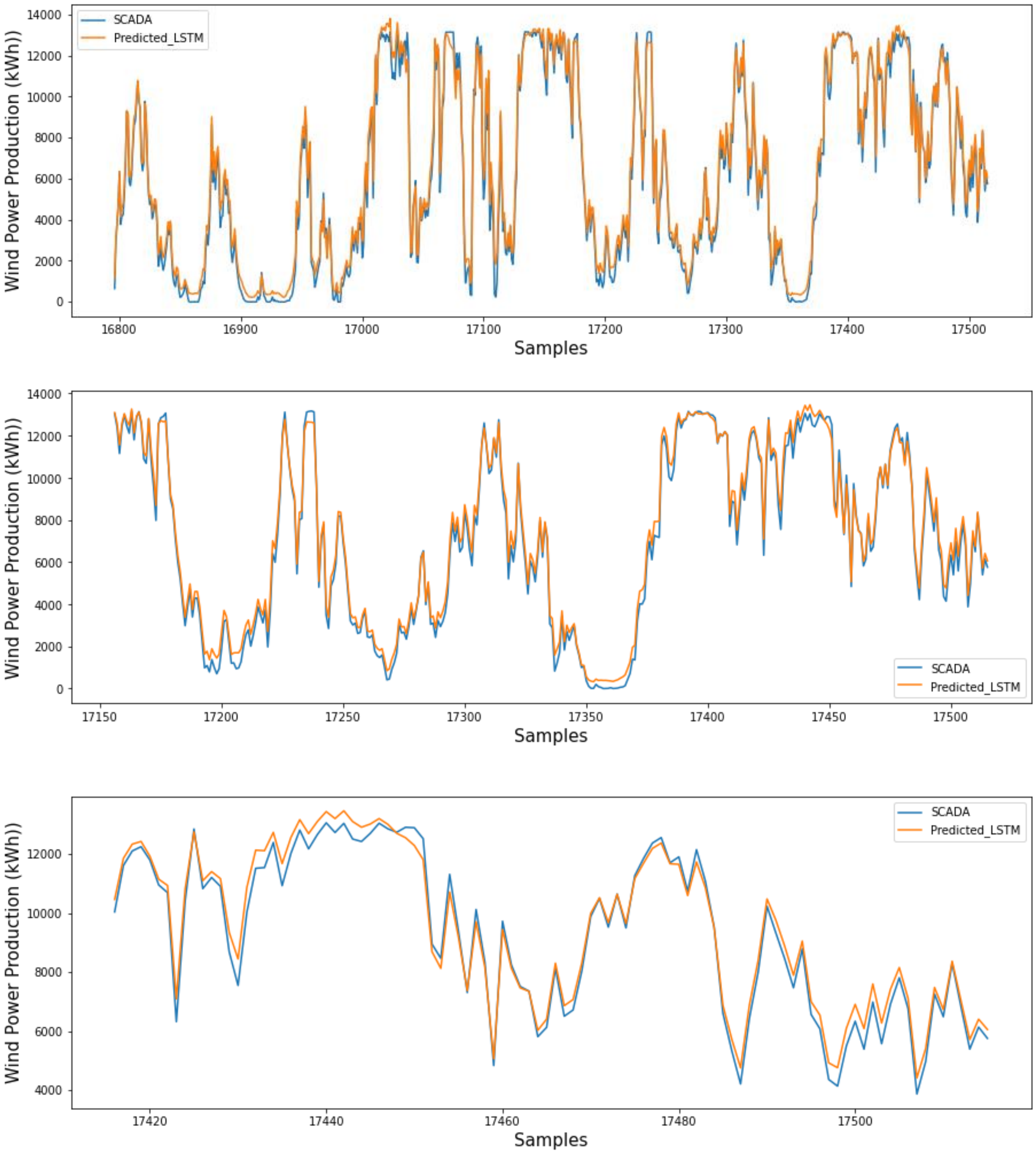


Figure 5.6. The SCADA versus LSTM for Urla Wind Farm (a) the samples of one-month comparison (b) the last 350 samples comparison (c) the last 100 samples comparison

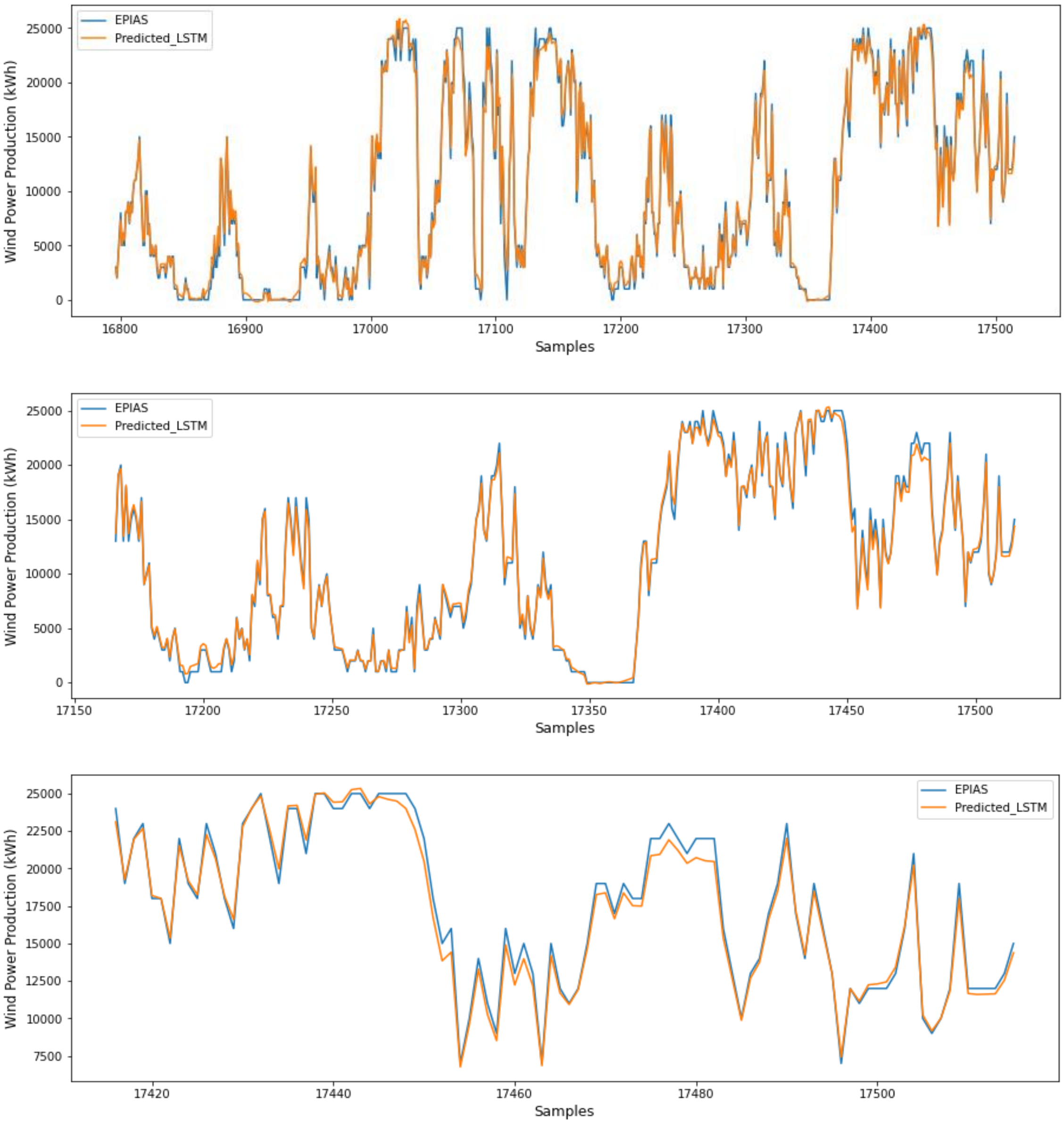


Figure 5.7. The EPIAS versus LSTM for Urla Wind Farm (a) the samples of one-month comparison (b) the last 350 samples comparison (c) the last 100 samples comparison

While Figure 5.8 demonstrates the results of the Kores Kocadag Wind Farm, Figure 5.9 illustrates the results of the Germiyan Wind Farm.

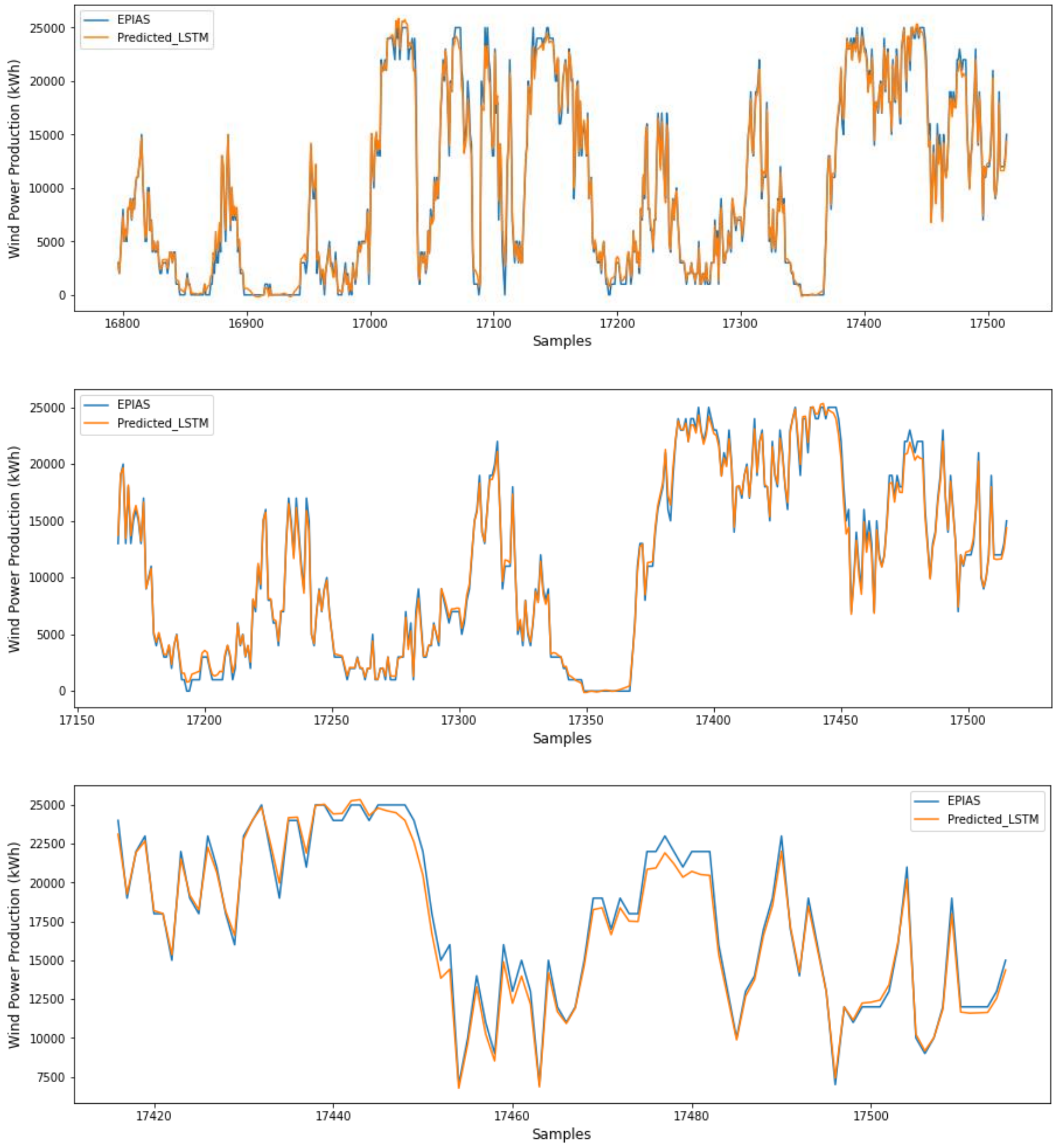
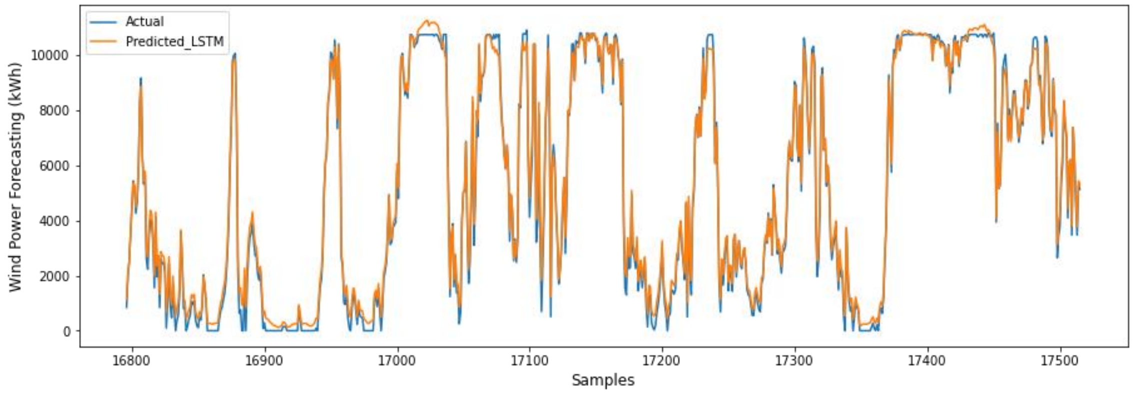


Figure 5.8. The EPIAS versus LSTM for Kores Kocadag Wind Farm (a) the samples of one-month comparison (b) the last 350 samples comparison (c) the last 100 samples comparison



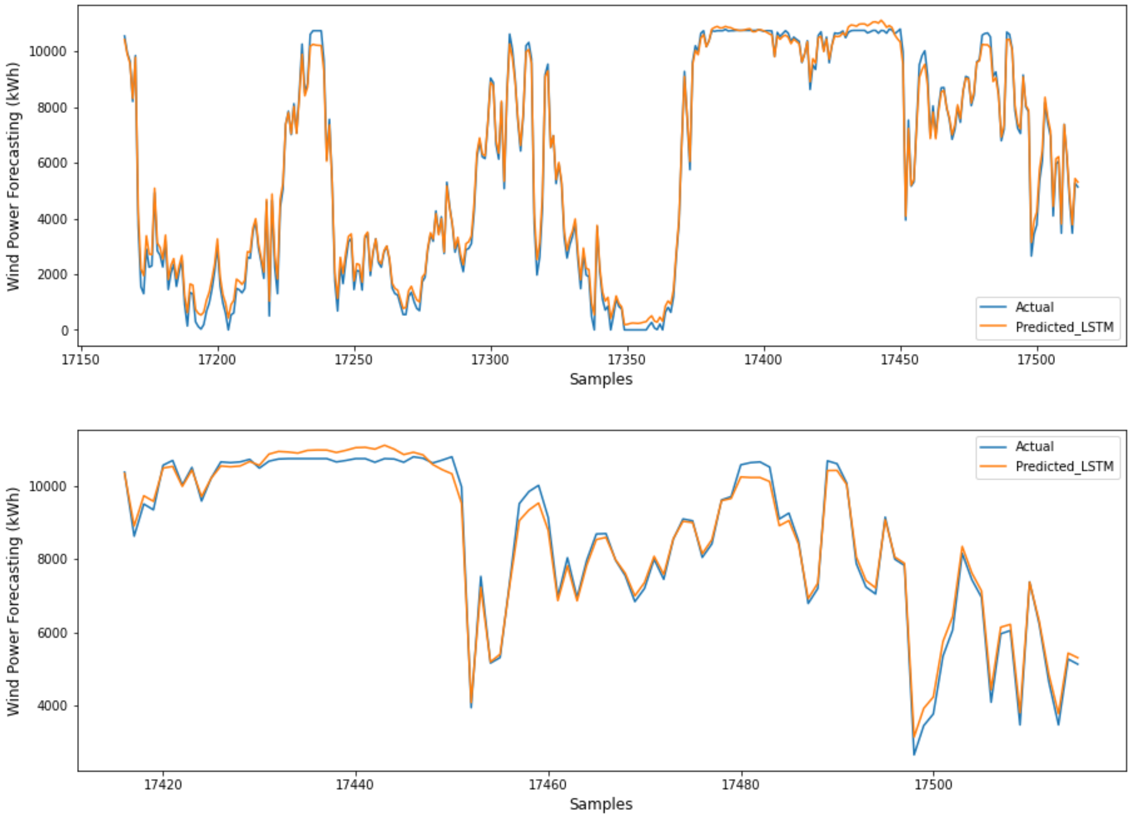


Figure 5.9. The EPIAS versus LSTM for Germiyan Wind Farm (a) the samples of one-month comparison (b) the last 350 samples comparison (c) the last 100 samples comparison

For the Urla Wind Farm, the wind power productions of the SCADA and EPIAS give close results and are very close to the actual wind power productions. That means the wind power production of the EPIAS can be used for the other wind farm because their SCADA data are not included. The statistics and measures also support that, tabulated in Table 5.1, demonstrate the comparison of prediction results. According to Table 5.1, the SCADA's RMSE, MAE, and MAPE are 531.25 kWh, 408.92 kWh, and 29.34%. The error values of EPIAS are 432.21 kWh, 366.46 kWh, and 28.1%, which are 99.04 kWh, 42.46 kWh, and 1.23% lower than the SCADA’s scores. In other words, EPIAS’s wind power forecasting results are more accurate than the SCADAs.

Additionally, the comparisons of actual and predictions of wind power generation utilizing scatter plots are displayed in Figure 5.2 and Figure 5.4, respectively, the SCADA and EPIAS. The scatter plots show how much the wind power predictions are closer to the actual wind power generation. The correlation result can be ranged from 0 to 1. If the correlation value is one, the actual and predicted wind power generation is the same. If the correlation value is 0, the actual and predicted wind power differ. Figure 5.10 and

Figure 5.11 shows that the forecast wind power generation of the EPIAS has more similarity than the SCADA to the actual wind power generation with a minimal difference. The correlations are denoted by the coefficient of determination R^2 are 0.9940 and 0.9945, respectively, the SCADA and EPIAS.

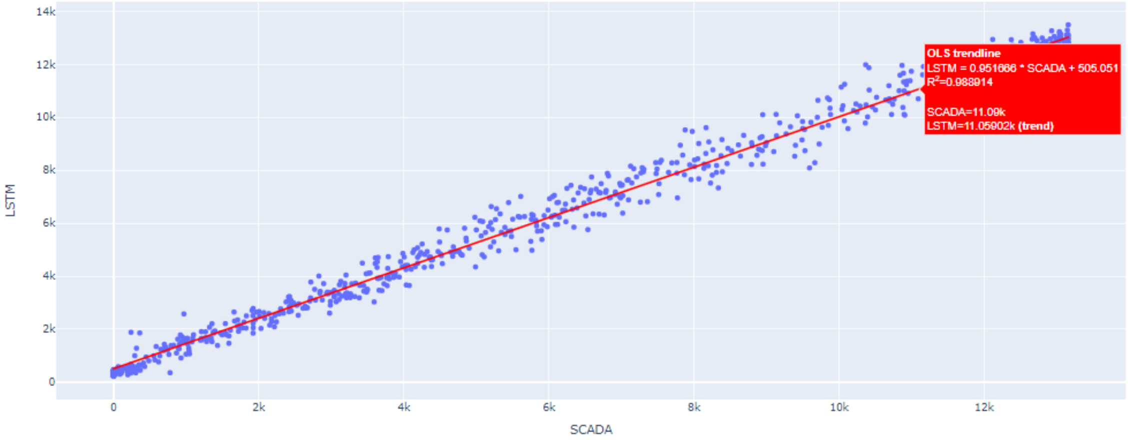


Figure 5.10. The correlation plot between the wind power productions of SCADA and LSTM for Urla Wind Farm (R^2 = 0.994)

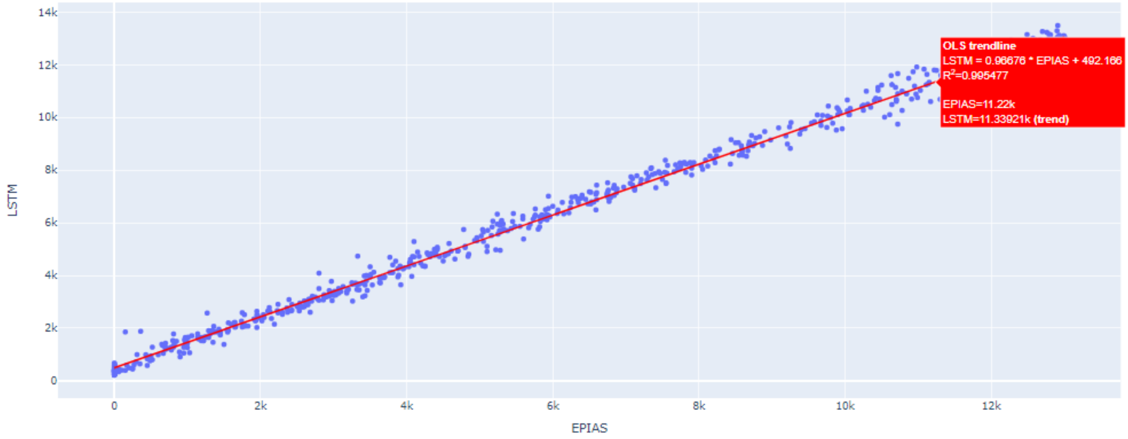


Figure 5.11. The correlation plot between the wind power productions of EPIAS and LSTM for Urla Wind Farm (R^2 = 0.9945)

Figure 5.12 and Figure 5.13, respectively, showcase the correlation results of Kores Kocadag Wind Farm and Germiyan Wind Farm, which are R^2 scores: 0.9947 and 0.997.

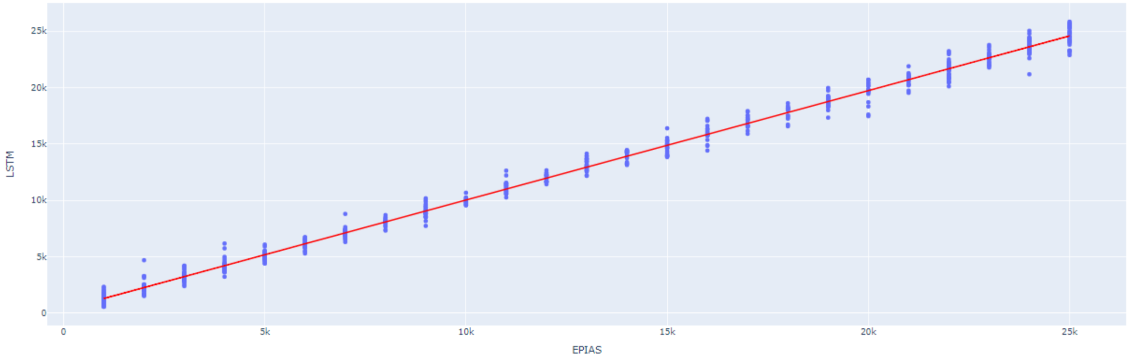


Figure 5.12. The correlation plot between the wind power productions of EPIAS and LSTM for Kores Kocadag Wind Farm

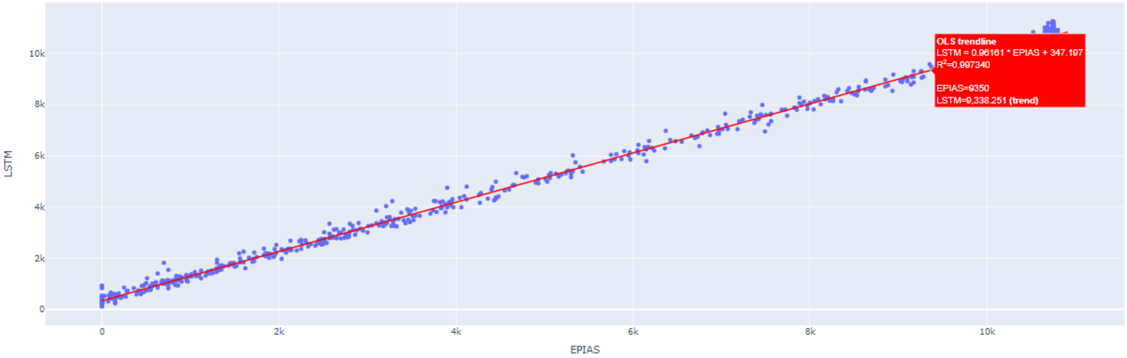


Figure 5.13. The correlation plot between the wind power productions of EPIAS and LSTM for Germiyan Wind Farm

Since the installed capacities of wind farms are significantly different, RMSE and MAE may not show comparable results. Also, R-square values are too close to each other. MAPE could be the best indicator to compare the forecast results of wind farms, which shows the quality of forecasts. Lewis (1982, as cited in Moreno et al., 2013) states that a MAPE of less than 10% is considered highly accurate forecasting; greater than 10% but less than %20 indicates good forecasting, between 20% and 50% indicates reasonable forecasting with low but acceptable accuracy, and greater than 50% is considered as inaccurate and not acceptable.

In this context, MAPE scores are in the range of 8.7%-28.1% obtained by the LSTM model, which are considered accurate results. However, Kores Kocadag Wind Farm gives highly accurate results with a MAPE value of 8.7%. Germiyan Wind Farm has the second-best forecasting results, with a MAPE value of 25.49%. The worst forecasting results belong to Urla Wind Farm, with a MAPE value of 28.1%. Although both Germiyan and Urla wind farms give the forecasting results with lower accuracy, they can be considered reasonable.

Table 5.1. Statistical measurements of three operational wind farm data using the actual and forecasts of the LSTM model

Evaluation Criteria	**Urla (SCADA)**	**Urla (EPIAS)**	**Kores Kocadag**	**Germiyan**
R-square	0.9940	0.9945	0.9947	0.997
RMSE (kWh)	531.25	432.21	615.54	292.96
MAE (kWh)	408.92	366.46	451.24	236.07
MAPE (%)	29.34	28.10	8.7	25.49

5.4 Day-ahead Market (DAM)

Progress in wind power generation forecasting techniques brings millions of dollars in profit to electricity generator companies. Day Ahead Market (DAM) is one of the interdependent platforms of the Turkish Electricity Market. The trading activity should be forecasted a day in advance for the optimum production and best trading strategies on the generator company side and the load delivery on the market operator side. Wind energy production for the next day can be used in investment strategies of the market participants, such as energy consumers and producers (Kalay, 2018).

DAM allows customers and producers to sell out their energy surpluses or shortages for a day later. Under bilateral agreements, market transactions are performed a day in advance. While the generator corporations are obligated to transfer the energy to customers, the customer companies are committed to taking the energy from the producers.

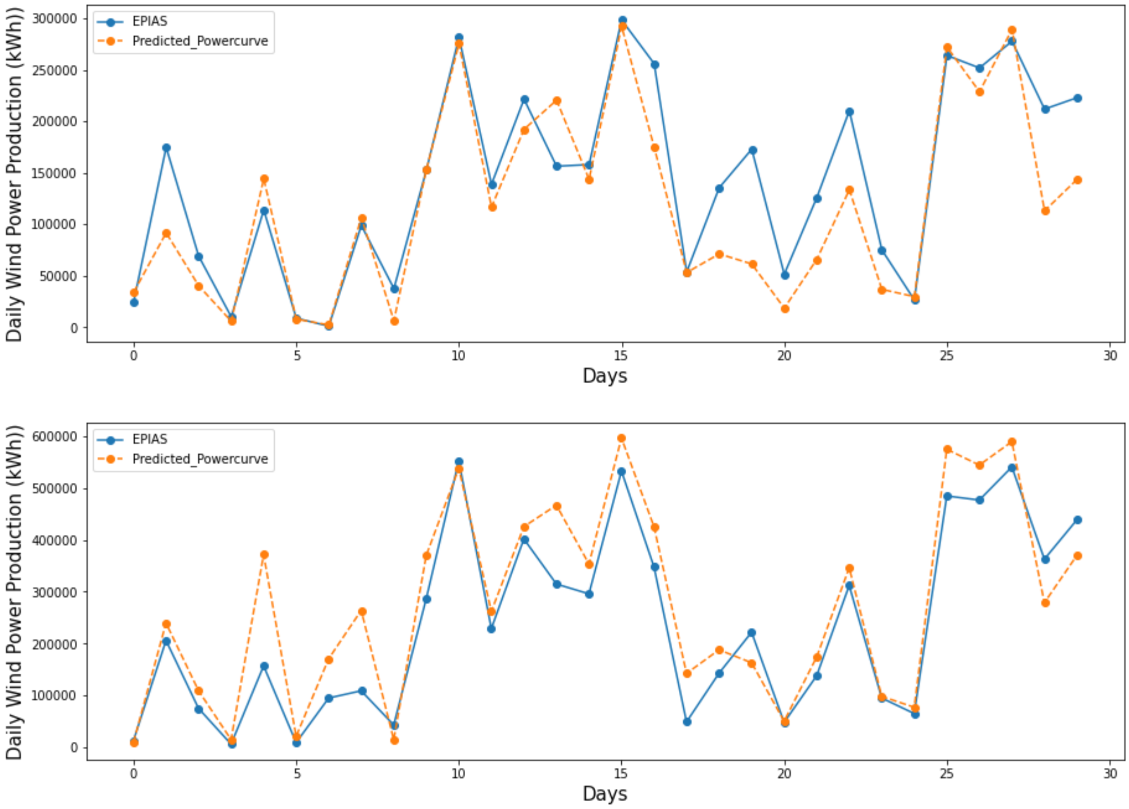
Predicting the total electricity load decreases the imbalance costs and prevents possible foreseeable imbalances for a system operator. Moreover, the total electricity load must be estimated to create a strategy for selling its electricity to the most profitable market for an electricity generator. The next day's wind energy generation is forecasted with two different approaches for these purposes (Kalay, 2018).

Because we do not have information about power failures and cuts, the real-time fluctuations are not precisely calculated. That’s why the forecasting results are negatively affected. If these data are obtained, the real-time fluctuations can be more appropriately followed, and the forecasting accuracy can be improved.

In this section, each wind farm's hourly forecast results are averaged daily to investigate the results on the day ahead market bases. Although there can be hourly deviations, these may not be reflected in the daily forecast. The December 2020 forecasts are plotted, and each wind farm's error metrics are calculated in the following sections.

5.4.1. The Daily Forecasts using the Power Curve of Wind Turbines

According to Figure 5.21, the forecast values of the daily averages of wind energy production obtained using the power curve of wind turbines can be seen in December 2020, which crosses the actual values. Besides the visual inspection of forecasting results, Table 5.4 shows the performance of the forecasting method for each wind farm using the evaluation criteria.



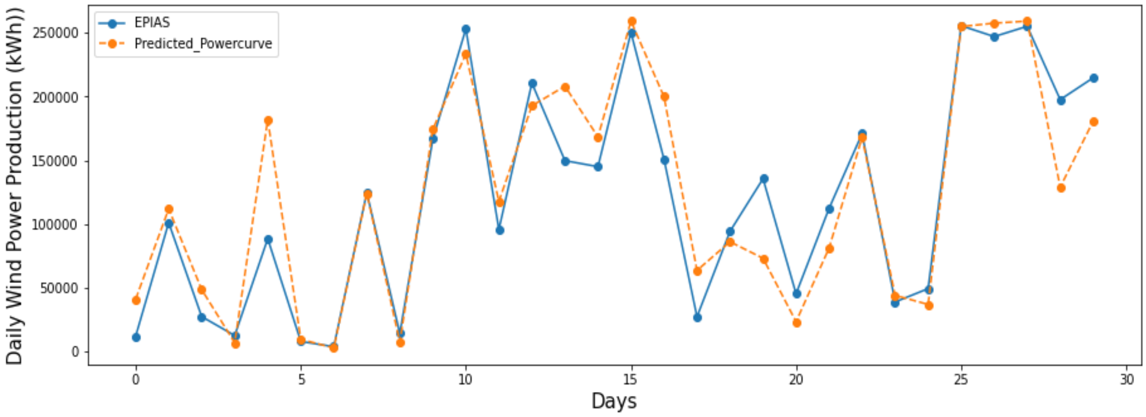


Figure 5.21. The forecasts for December 2020 using the power curve of wind turbines (a) Urla Wind Farm, (b) Kores Kocadag Wind Farm, (c) Germiyan Wind Farm

As mentioned above, MAPE could be the best indicator to compare the forecast results of wind farms, which shows the quality of forecasts. In this method, MAPE scores change between 31.08 % and 44.65% obtained by the power curve of wind turbines, which means they are reasonable forecasts with low accuracies, but they can be considered acceptable. For this method, Urla Wind Farm gives the best forecast results when compared with the MAPE value of the wind farms, which is 31.08%. Unlike the previous approach, a producer cannot have a reference predictor value as accurate as the previous approach. A deviation of reference predictor value varies between 31.08%-44.65% about the wind energy generation of tomorrow. This method may not forecast possible imbalances and system shutdowns at times, which causes necessary actions may not to be appropriately taken a day in advance.

Table 5.4. The evaluation criteria of three operational wind farms for the daily forecasting obtained using the power curve of wind turbines

Evaluation Criteria	**Urla**	**Kores Kocadag**	**Germiyan**
R-square	0.8135	0.8838	0.8633
RMSE (kWh)	47397.76	74918.31	32043.84
MAE (kWh)	34271	57232.04	22513.61
MAPE (%)	31.08	44.65	35.70

5.4.2. The Daily Forecasts using the LSTM Model

Looking at Figure 5.20, the forecast values of the daily averages of wind energy production obtained by the LSTM model can be seen in December 2020, overlapping the actual values. However, the visual inspection cannot be enough to assess the method's performance. That's why Table 5.3 is also used to support the forecasting results, which tabulates the evaluation criteria. Since that means that the LSTM model generates more accurate predictions on daily bases if compared with the hourly forecasting results.

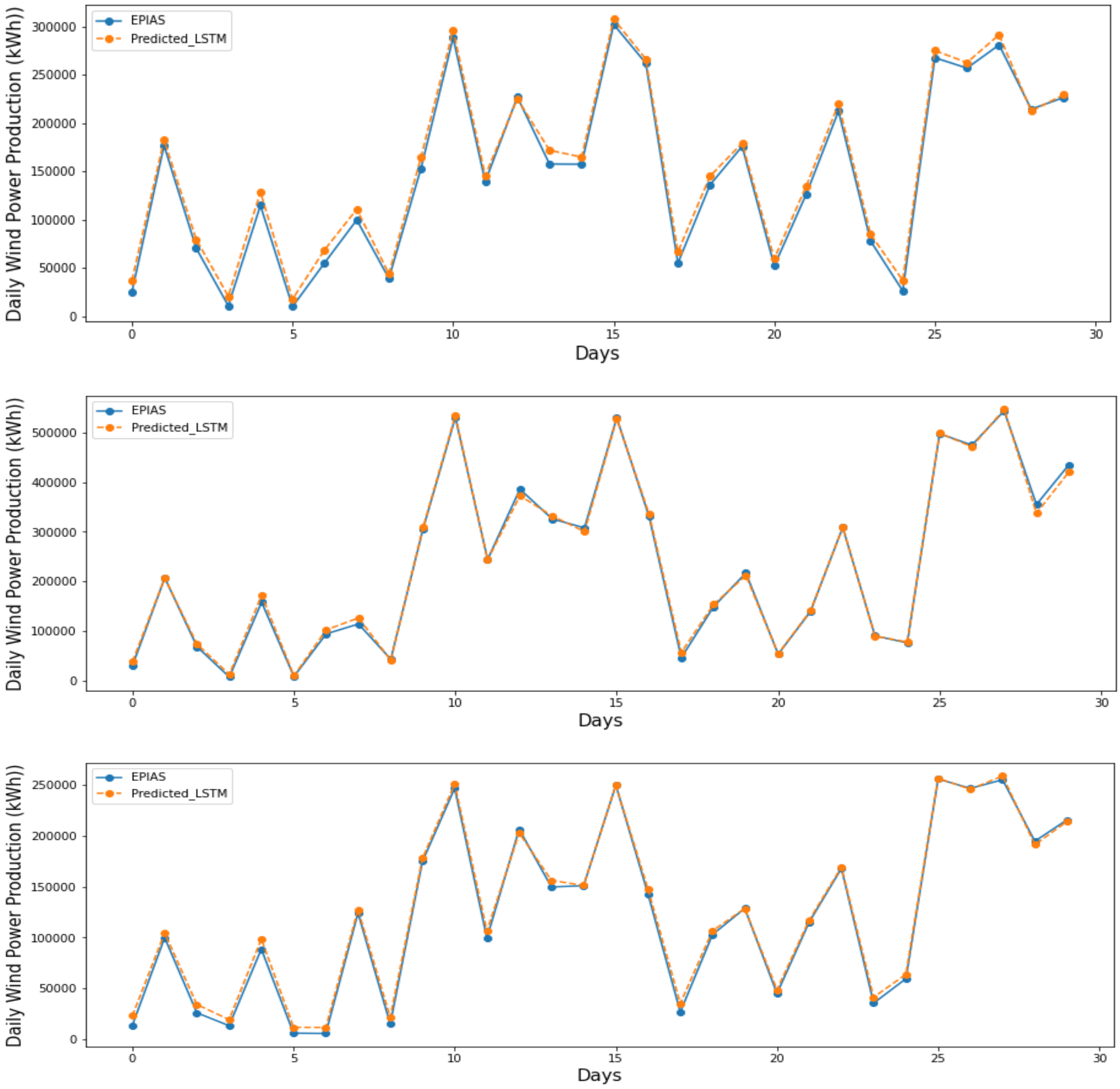


Figure 5.20. The forecasts for December 2020 using LSTM (a) Urla Wind Farm (b) Kores Kocadag Wind Farm (c) Germiyan Wind Farm

Since the installed capacities of wind farms are significantly different, RMSE and MAE may not show comparable results. Also, R-square values are too close to each other.

MAPE could be the best indicator to compare the forecast results of wind farms, which shows the quality of forecasts. Lewis (1982, as cited in Moreno et al., 2013) states that a MAPE of less than 10% is considered highly accurate forecasting; greater than 10% but less than %20 indicates good forecasting, between 20% and 50% indicates reasonable forecasting with low but acceptable accuracy, and greater than 50% is considered as inaccurate and not acceptable. In this context, MAPE scores are in the range of 7.15%-16.77% obtained by the LSTM model, which means they are accurate forecasts. It can be said that Kores Kocadag Wind Farm gives the best forecast results when compared with the MAPE value of the wind farms, which is 7.15%, and highly accurate forecasts. Thus, a producer can have a reference predictor value with deviations in the range of 7.15%-16.77% about the wind energy generation of tomorrow. This method can forecast possible imbalances and system shutdowns, and necessary actions can be taken a day in advance from the system operator's perspective.

Table 5.3. The evaluation criteria of three operational wind farms for the daily forecasting

Evaluation Criteria	**Urla**	**Kores Kocadag**	**Germiyan**
R-square	0.9984	0.9985	0.9991
RMSE (kWh)	8698.74	7228.45	5073.66
MAE (kWh)	8015.59	5423.45	4285.37
MAPE (%)	14.62	7.15	16.77

5.5 Day-ahead Market Analysis by Filtering Data

Because we do not have information about power failures and shutdowns, the forecasting methods do not precisely calculate the real-time fluctuations. That's why possible failures and shutdowns are aimed to be considered by filtering MAPE values higher than %50. In this way, it can be understood that if information about power failures and shutdowns could be obtained, the forecasting performance might be increased.

5.5.1. The Filtered Daily Forecasts using the Power Curve of Turbines

In this approach, five, seven, and three values are filtered, respectively, Urla, Kores Kocadag, and Germiyan wind farms. According to Figure 5.23, the forecast values overlap the actual values more than the non-filtered analysis. Besides the visual inspection of forecasting results, Table 5.6 shows the increase in the performance of the forecasting method for each wind farm using the evaluation criteria.

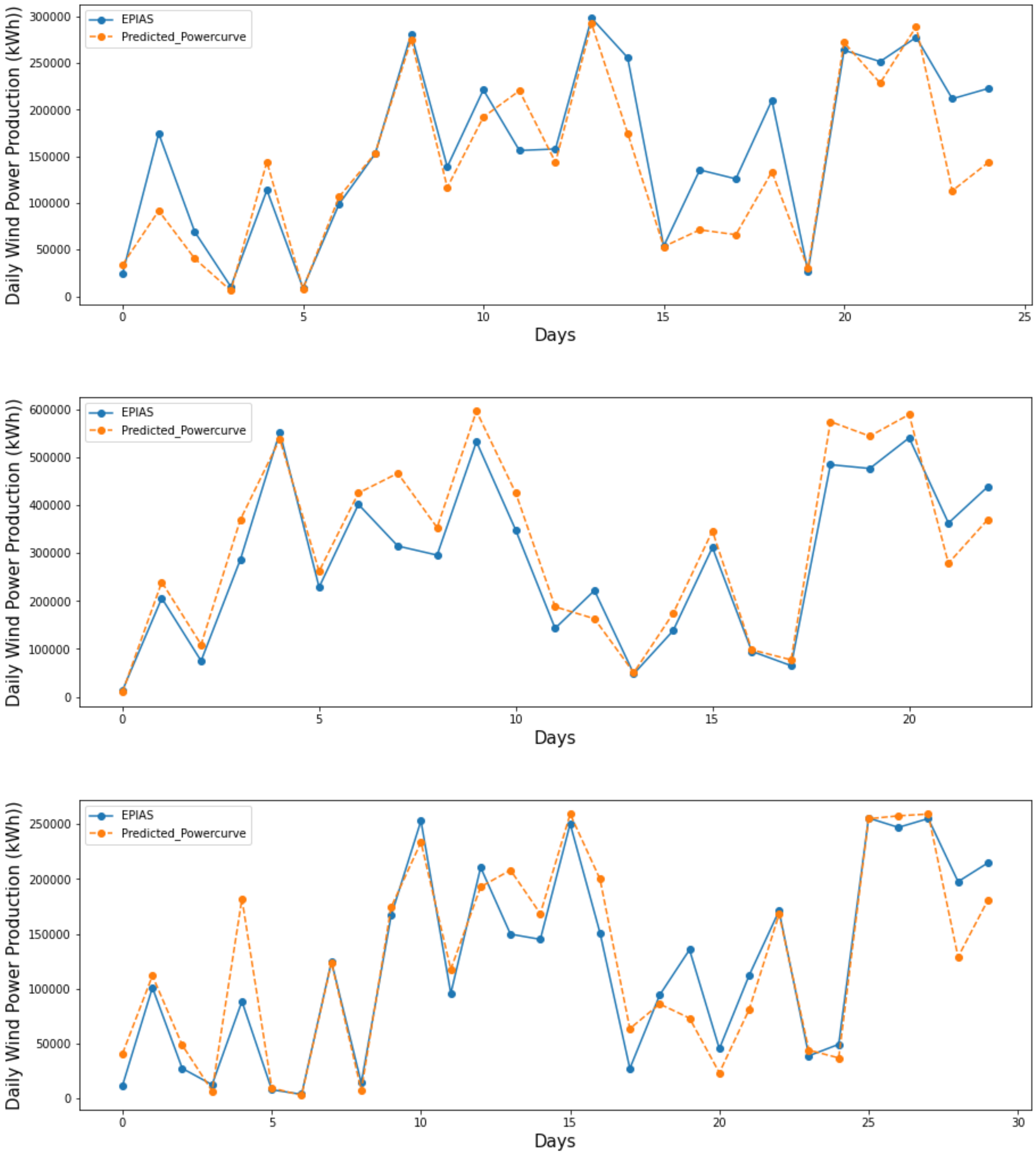


Figure 5.23. The forecasts for December 2020 using the power curve of wind turbines (a) Urla Wind Farm, (b) Kores Kocadag Wind Farm, (c) Germiyan Wind Farm

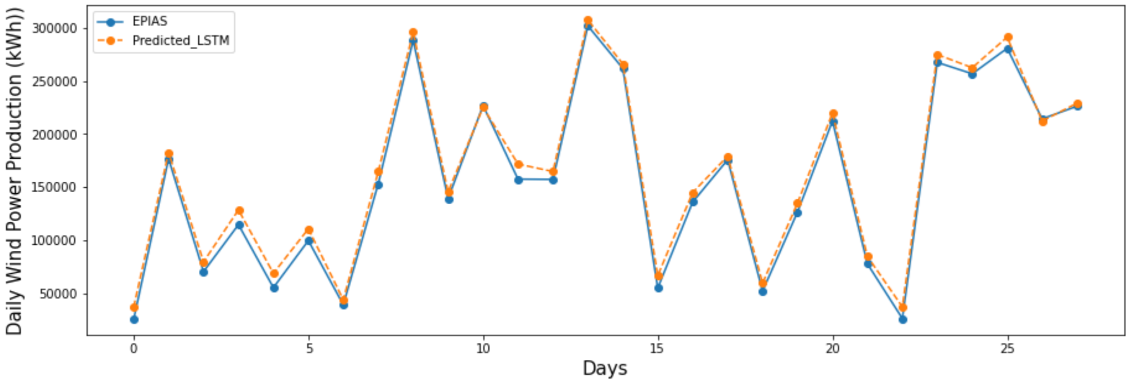
MAPE scores decrease and are obtained between 19.17% and 22.87%, which means they are more accurate forecasts. After filtering data, Germiyan Wind Farm gives the best forecast results when compared with the MAPE value of the wind farms, which is 19.17%. Unlike the non-filtered analysis, a producer can have a more accurate reference predictor value. Now, the deviations of reference predictor value are decreased and vary between 19.17% and 22.87% about the wind energy generation of tomorrow. Now, possible imbalances and system shutdowns can be predicted at times, and necessary actions might be adequately taken a day in advance.

Table 5.6. The evaluation criteria of three operational wind farms for the daily forecasting obtained using the power curve of wind turbines

Evaluation Criteria	Urla	Kores Kocadag	Germiyan
R-square	0.8067	0.9185	0.9031
RMSE (kWh)	45369.88	59958.86	26856.92
MAE (kWh)	32526.39	49072.54	18439.96
MAPE (%)	22.87	19.32	19.17

5.5.2. The Filtered Daily Forecasts using the LSTM Model

In this way, two, one, and three values are filtered, respectively, Urla, Kores Kocadag, and Germiyan wind farms. Looking at Figure 5.22, the forecast values overlap the actual values more than the non-filtered analysis. However, the visual inspection cannot be enough to assess the increase in the method’s performance. That’s why Table 5.5 is also used to support the forecasting results, which tabulates the evaluation criteria.



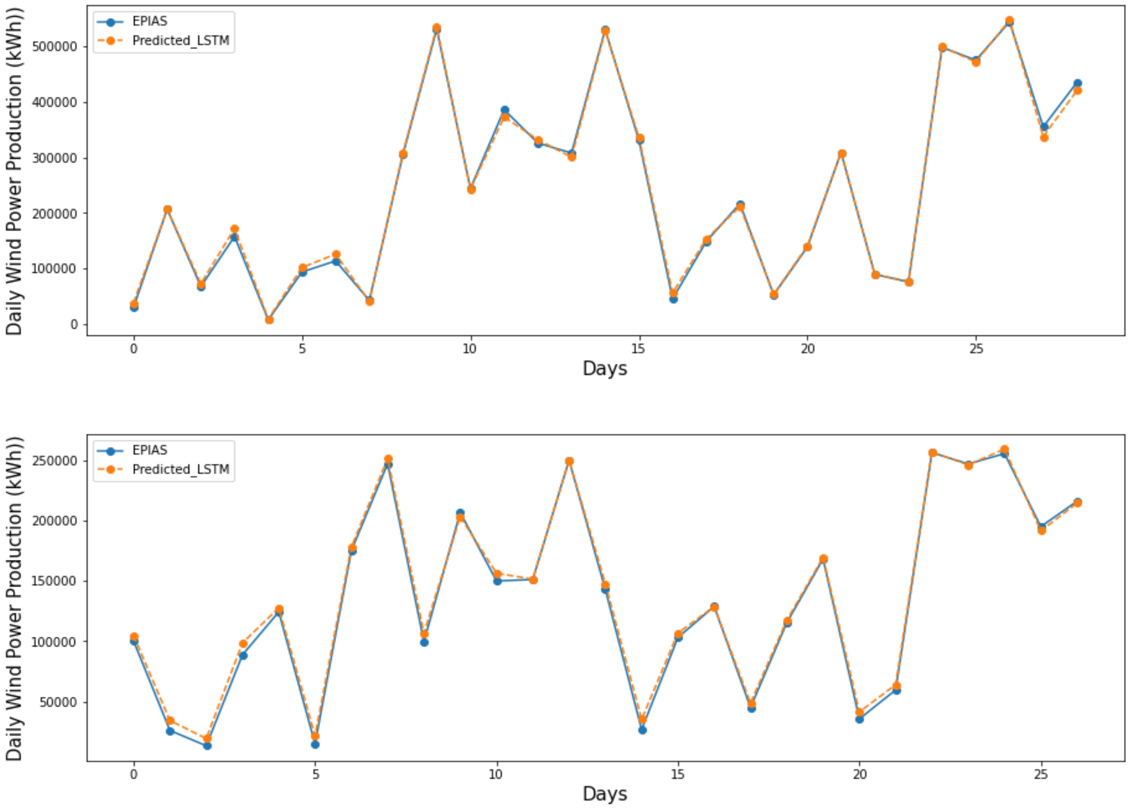


Figure 5.22. The forecasts for December 2020 using LSTM (a) Urla Wind Farm (b) Kores Kocadag Wind Farm (c) Germiyan Wind Farm

In this context, MAPE scores are in the range of 4.92%-9.04% obtained by the LSTM model, which means they are highly accurate forecasts. Kores Kocadag Wind Farm still gives the best forecast results when compared with the MAPE value of the wind farms, which is 4.92%. Thus, a producer can have a highly precise reference predictor value with deviations in the range of 4.92%-9.04% about the wind energy generation of tomorrow.

Table 5.5. The evaluation criteria of three operational wind farms for the daily forecasting

Evaluation Criteria	**Urla**	**Kores Kocadag**	**Germiyan**
R-square	0.9981	0.9984	0.9989
RMSE (kWh)	8393.14	7292.66	4772.55
MAE (kWh)	7429.42	5437.34	3978.79
MAPE (%)	9.04	4.92	8.64

CHAPTER 6

DISCUSSION

The aims of the study are to predict wind power generation and to validate the result against the measurements. At the same time, the study tried to evaluate the suitability of ERA5 reanalysis data for the predictions of hourly wind power generation due to the unavailable SCADA data. The three operational wind farms were selected as the study area to assess the suitability of ERA5 data.

Correlation measures between wind speeds at 101 m from the meteorological mast and 100 m from ERA5 reanalysis data are also obtained. It is found that the cross-validation and correlation plot shows reasonable agreement between the met. mast and ERA5 datasets. That's why it has been decided to use the ERA5 dataset for the wind power predictions in three operational wind farms.

In this paper, the LSTM model is proposed to forecast hourly and day-ahead wind power production. Before it is proposed, the model is compared with Prophet, SARIMA, SARIMAX, and GRU models based on wind speed prediction. The LSTM model outperforms the other models with the lowest errors and highest R^2 score: MSE value of 0.2932, RMSE value of 0.4358, MAE value of 0.3089, and R^2 value of 0.9809. After the comparison, the LSTM model is decided to use for wind power predictions.

Wind power production is predicted based on two different approaches. In the first approach, wind speeds are predicted with the LSTM model using the ERA5 dataset, and wind power productions are determined using the power curves of wind turbines. In the second approach, real-time wind power productions are used as an additional input for the LSTM model to forecast future wind power production.

The results are tabulated in Table 1.1. According to the results, the first approach gives inaccurate results and cannot be acceptable with the highest errors and lowest R^2 value. In the second approach, using the real-time wind power production data as an input improves the forecasting performance by 38.89%, 82.98%, and 44.87%, respectively, for Urla, Kores Kocadag, and Germiyan wind farms.

On the other hand, the hourly forecasts from EPIAS, which is called the final daily production program, are also compared with the actual power production. Its forecasting errors are also tabulated in Table 1.1 and compared with our forecasting results. If the MAPE scores are considered, both approaches 1 and 2 outperform the forecasts from EPIAS. Especially approach 2 improves 43.9%, 84.4%, and 62.26%, respectively, for Urla, Kores Kocadag, and Germiyan wind farms. These results indicate that using the LSTM model with the ERA5 dataset could give better forecasts than wind farms' own forecasts.

Another aim of this study is to give accurate forecasts for Day-ahead Market. While using the daily forecasts of approach 1 improves the forecasting results by 32.41%, 12.66%, and 22.79%, approach 2 improves by 47.97%, 17.82%, and 34.21%, respectively, for Urla, Kores Kocadag, and Germiyan wind farms. According to the results, although there can be hourly deviations, these may not be reflected in the daily forecasts.

Because we do not have information about power failures and shutdowns due to unavailable SCADA data, the forecasting method cannot precisely calculate the real-time fluctuations. That's why possible failures and shutdowns are aimed to be considered by filtering MAPE values higher than %50. In this way, it can be understood that if the SCADA data could be obtained, the forecasting performance might be increased.

Table 6.1. The forecast results of Final Daily Production Program (EPIAS), Approach 1, Approach 2

Wind Farm	**Forecasts**	**RMSE (kWh)**	**MAE (kWh)**	**MAPE (%)**	**R^2 (%)**
Urla	Final Daily Production Program (EPIAS)	1949.02	1439.03	50.09	0.81
	Approach 1	3115.72	2278.75	45.98	0.6
	Approach 2	432.21	366.46	28.1	0.99
Kores Kocadag	Final Daily Production Program (EPIAS)	3624.38	2623.48	55.78	0.82
	Approach 1	4948.16	3317.86	51.12	0.74
	Approach 2	615.54	451.24	8.7	0.99
Germiyan	Final Daily Production Program (EPIAS)	1702.92	1198.19	67.54	0.82
	Approach 1	2500.74	1571.1	46.24	0.67
	Approach 2	292.96	236.07	25.49	0.99

CHAPTER 7

CONCLUSION

The integration of wind energy requires more precise and enhanced predictions. Over the years, researchers have been developing various approaches to improve the accuracy and performance of wind speed and power predictions. The randomness and intermittency of wind characteristics complicate wind energy forecasting for linear methods. Addressing the complications faced by the linear approaches, this study proposes a nonlinear approach, the Long-short Term Memory (LSTM) model, for enhanced wind speed and power forecasting. The ideology behind the LSTM model is to improve forecasting performance by reducing the forecasting process's statistical errors and computation load.

In the first part of the study, a case study is performed using real-time data from the IZTECH meteorological mast. In addition to wind speed and direction, topography and meteorological variables also mainly affect the improvement of forecasting accuracy. Factors affecting wind speed, such as temperature, relative humidity, and barometric pressure, should also be considered. Based on the literature review, five methods are chosen to investigate: Facebook Prophet, SARIMA, SARIMAX, GRU, and LSTM. The statistical performance indicators are used to compare the methods: MSE, RMSE, MAE, and R^2.

According to the daily forecasting results, the LSTM model shows the best performance with the lowest errors: MSE value of 0.2932, RMSE value of 0.4358, and MAE value of 0.3089, although the GRU model has relative values to the LSTM model. Meanwhile, the R^2 value is 0.9809, closer to 1 than the other methods. On the other hand, the SARIMAX model cannot deeply understand the inherently chaotic nature of wind speed time series. The model shows the worst performance with the R^2 value of 0.7498, MSE value of 1.8584, RMSE value of 1.3632, and MAE value of 1.1004. Since wind speed affects wind power generation in the third order, SARIMAX cannot give good forecasting results.

The second part of the study is to forecast the wind power generation using the LSTM model and the wind speed forecasts and power curve of wind turbines in the wind farms. The proposed model is validated using the real-time wind power generation data from the EPIAS Transparency Platform. This part investigates three operational wind farms: Urla Wind Farm, Kores Kocadag Wind Farm, and Germiyan Wind Farm. Due to the unavailable meteorological dataset of Kores Kocadag and Germiyan Wind Farm, an ERA5 dataset of the nearest location is used to predict wind speed and power generation. Before using the ERA5 dataset, it is validated with the real-time dataset of the IZTECH meteorological mast. The correlation between the ERA5 and IZTECH meteorological mast datasets is denoted by the coefficient of determination R^2 of 0.7596.

Firstly, the wind power generation is predicted using a multivariate LSTM network for the three operational wind farms. The R-square, RMSE, MAE, and MAPE values are calculated for each operational wind farm to evaluate the performance of the LSTM model. For the Urla Wind Farm, the wind power productions of the SCADA and EPIAS give close results. They are very close to the actual wind power productions. That means the wind power production of the EPIAS can be used for the other wind farm because their SCADA data are unavailable. The correlations are denoted by the coefficient of determination R^2 are 0.9940 and 0.9945, respectively, the SCADA and EPIAS.

Since the installed capacities of wind farms are significantly different, RMSE and MAE may not show comparable results. Also, R-square values are too close to each other. MAPE could be the best indicator to compare the forecast results of wind farms, which shows the quality of forecasts. In this context, MAPE scores are in the range of 8.7%-28.1% obtained by the LSTM model, which are considered accurate results. However, Kores Kocadag Wind Farm gives highly accurate results with a MAPE value of 8.7%. Germiyan Wind Farm has the second-best forecasting results, with a MAPE value of 25.49%. The worst forecasting results belong to Urla Wind Farm, with a MAPE value of 28.1%. Although both Germiyan and Urla wind farms give the forecasting results with lower accuracy, they can be considered reasonable.

Secondly, wind power generation is forecasted using wind speed forecasts and wind curves of turbines for three operational wind farms. The correlations between the real-time power generations and forecasts are denoted by the coefficient of determination R^2 are 0.5979, 0.7389, and 0.6682, respectively, the Urla, Kores Kocadag, and Germiyan wind farms. Mainly, Urla Wind Farm exhibits a minor mean absolute percentage error compared with the other wind farms, with a value of 45.98%, which means the accuracy is higher than the other wind farms. Lastly, Urla Wind Farm exhibits 45.98% of the MAPE, followed by Germiyan Wind Farm, with 46.24%. Kores Kocadag Wind Farm occupies the major rank with 51.12%.

Progress in wind power generation forecasting techniques brings millions of dollars in profit to electricity generator companies. Day Ahead Market (DAM) is one of the interdependent platforms of the Turkish Electricity Market. Wind energy production for the next day can be used in the investment strategies of the market participants. That’s why the next day's wind energy generation is forecasted with two different approaches.. Although there can be hourly deviations, these may not be reflected in the daily forecast.

The LSTM model generates more accurate predictions on daily bases if compared with the hourly forecasting results. MAPE scores are in the range of 7.15%-16.77% obtained by the LSTM model, which means they are accurate forecasts. Kores Kocadag Wind Farm outperforms the other wind farms with a MAPE value of 7.15%. Thus, a producer can have a reference predictor value with deviations in the range of 7.15%-16.77% about the wind energy generation of tomorrow. This method can forecast possible imbalances and system shutdowns, and necessary actions can be taken a day in advance from the system operator’s perspective.

In the second approach, MAPE scores change between 31.08 % and 44.65%, which means they are reasonable forecasts with low accuracies but can be considered acceptable. Urla Wind Farm outperforms the other wind farms with a MAPE value of 31.08% for this method. Unlike the previous approach, a producer cannot have a reference predictor value as accurate as the previous method. A deviation of reference predictor value varies between 31.08%-44.65% about the wind energy generation of tomorrow. This method may not forecast possible imbalances and system shutdowns at times, which causes necessary actions may not to be adequately taken a day in advance.

In conclusion, the results indicate that using the LSTM model with the ERA5 dataset could give better forecasts than wind farms’ own forecasts. Additionally, because we do not have information about power failures and shutdowns, the forecasting methods cannot precisely calculate the real-time fluctuations. If the SCADA data could be obtained, the forecasting performance might be increased.

REFERENCES

Ahmad, S., Abdullah, M., Kanwal, A., Tahir, Z.R., Saeed, U. B., Manzoor, F., Atif, M., S. Abbas (2022). Offshore Wind Resource Assessment Using Reanalysis Data. *Wind Engineering*.

Asha, J., Rishidas, S., SanthoshKumar, S., P. Reena (2020). Analysis of Temperature Prediction using Random Forest and Facebook Prophet Algorithms. *ICIDCA 2019*, LNDECT 46, 432–439.

Belousov, A., Verzakov, S.A., J. Von Frese (2002). A flexible classification approach with optimal generalization performance; support vector machines. *Chemom Intell Lab Syst* 64, 15–25.

Bessac, J., Constantinescu, E., M. Anitescu (2018). Stochastic Simulation of Predictive Space–time Scenarios of Wind Speed Using Observations and Physical Model Outputs. *The Annals of Applied Statistics* 12(1), 432-458.

Box, G.E., Jenkins, G.M., Reinsel, G.C., G.M. Ljung (2015). Time series analysis: forecasting and control. *John Wiley & Sons*.

Cassola, F., M. Burlando (2012). Wind speed and wind energy forecast through Kalman filtering of Numerical Weather Prediction model output. *Applied Energy* 99, 154-166.

Castelani, F., Astolfi, D., Mana, M., Burlando, M., MeiBner, C., E. Piccioni (2016). Wind power forecasting techniques in complex terrain: ANN vs. ANN-CFD hybrid approach. *J. Phys., Conf. Ser.* 753(8).

Caraka, R.E., Bakar, S., Pardamean, B., A. Budiarto (2018). Hybrid support vector regression in electric load during national holiday season. *International Conference on Innovative and Creative Information Technology: Computational Intelligence and IoT, ICITech 2017*.

Chen, P., Niu, A., Liu, D., Jiang, W., B. Ma (2018). Time series forecasting of temperatures using SARIMA: An example from Nanjing. *IOP Conference Series: Materials Science and Engineering* 394(5), 052024.

Chung, J., Caglar, G., Cho, K.H., B. Yoshua (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *NIPS 2014 Workshop on Deep Learning,* 1–9.

D'Amico, G., Petroni, F., F. Prattico (2013). First and second order semi-Markov chains for wind speed modelling. *Physica* 392, 1194-1201.

Damousis, I.G., Alexiadis, M.C., Theocharis, J.B., P.S. Dokopoulos (2004). A fuzzy model for wind speed prediction and power generation in wind parks using spatial correlation. *IEEE Transactions on Sustainable Energy* 19(2), 352-361.

Di, Z., Ao, J., Duan, Q., Wang, J., Gong, W., Shen, C., Gan, Y., Z. Liu (2019). Improving WRF model turbine-height wind-speed forecasting using a surrogate-based automatic optimization method. *Atmospheric Research* 226, 1-16.

Dong, L., Wang, L., Hao, Y., Hu, G., X. Liao (2011). Prediction of wind power generation based on autoregressive moving average model. *Acta Energiae Solaris Sinica* 32, 617–622.

Dubey, A.K., Kumar, A., Garcia-Diaz, V., Sharma, A.K., K. Kanhaiya (2021). Study and analysis of SARIMA and LSTM in forecasting time series data. Sustainable *Energy Technologies and Assessments* 47, 101474.

Duan, J., Zuo, H., Bai, Y., Duan, J., Chang, M., B. Chen (2020). Short-term wind speed forecasting using recurrent neural networks with error correction. *Energy* 217, 119397.

Dutta, J., S. Roy (2021). IndoorSense: context-based indoor pollutant prediction using SARIMAX model. *Multimedia Tools and Applications* 80, 19989-20018.

Ezzat, A., Jun, M., D. Yu (2019). Spatio-temporal Short-term Wind Forecast: A Calibrated Regime-switching Method. *The Annals of Applied Statistics* 13(3), 1484-1510.

Fathi, M.M., Awadallah, A.G., Abdelbaki, A.M., M. Haggag (2019). A new Budyko framework extension using time series SARIMAX model. *Journal of Hydrology* 570, 827-838.

Fıskın, C.S., A.G. Cerit (2019). Forecasting Domestic Shipping Demand of Cement: Comparison of SARIMAX, ANN and Hybrid SARIMAX-ANN. *(UBMK'19) 4rd International Conference on Computer Science and Engineering*, 68-72.

Filik, Ü.B., T. Filik (2017). Wind Speed Prediction Using Artificial Neural Networks Based on Multiple Local Measurements in Eskisehir. *Energy Procedia* 107, 254-269.

Flores, P., Tapia, A., G. Tapia (2005). Application of a control algorithm for wind speed prediction and active power generation. *Renewable Energy* 30(4), 523–36.

Foley, A.M., Leahy, P.G., Marvuglia, A., E.J. McKeogh (2011). Current methods and Advances in Forecasting of Wind Power Generation. *Renewable Energy* 37, 1-8.

Garcia, J.L.T., Calderon, E.C., Avalos, G.G., Heras, E.R., A.M. Tshikala (2018). Forecast of daily output energy of wind turbine using sARIMA and nonlinear autoregressive models. *Advances in Mechanical Engineering* 11(2), 1-15.

Graves, A., J. Schmidhuber (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw*. 18, 602-610.

G. Gualtieri (2021). Reliability of ERA5 Reanalysis Data for Wind Resource Assessment: A Comparison against Tall Towers. *Energies* 14, 4169.

G.P. Zhang (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* 50, 159-175.

“Germiyan RES.” https://www.egenda.com.tr/en/ (accessed October 19, 2022).

Guoyang, W., Yang, X., W. Shasha (2005). Discussion about Short-term Forecast of Wind Speed on Wind Farm. *Jilin Electric Power* 181(5), 21-24.

H. Geerts (1984). Short range prediction of wind speeds: a system-theoretic approach. *Proceedings of European Wind Energy Conference Hamburg (DE),* 594-599.

Han, S., Y. Liu (2010). The study of wind power combination prediction. *Asia-Pacific Power and Energy Engineering Conference (APPEEC).*

Hayes, L., Stocks, M., A. Blakers (2021). Accurate long-term power generation model for offshore wind farmsin Europe using ERA5 reanalysis. *Energy* 229, 120603.

Hersbach, H., Bell, B., Berrisford, P., Biavati, G., Horányi, A., Sabater, J., Nicolas, J., Peubey, C., Radu, R., Rozum, I., Schepers, D., Simmons, A., Soci, C., Dee, D., J.N. Thépaut (2018). ERA5 hourly data on pressure levels from 1959 to present. *Copernicus Climate Change Service (C3S) Climate Data Store (CDS).*

Hochreiter, S., J. Schmidhuber (1997). Long short-term memory. *Neural Comput* 9(8), 1735–80.

Hossain, M.A., Chakrabortty, R.K., Elsawah, S., Gray, E.M., M.J. Ryan (2021). Predicting Wind Power Generation Using Hybrid Deep Learning with Optimization. *IEEE Transactions on Applied Superconductivity* 31(8), 1-5.

Hosseini, M., Katragadda, S., Wojtkiewicz, J., Gottumukkala, R., Maida, A., T.L. Chambers (2020). Direct Normal Irradiance Forecasting Using Multivariate Gated Recurrent Units. *Energies 2020* 13, 3914.

Hu, S., Xiang, Y., Zhang, H., Xie, S., Li, J., Gu, C., Sun, W., J. Liu (2021). Hybrid Forecasting Method for Wind Power Integrating Spatial Correlation and Corrected Numerical Weather Prediction. *Applied Energy* 293, 116951.

Huang, Y., Liu, H.S., L. Yang (2018). Wind Speed Forecasting Method Using EEMD and the Combination Forecasting Method Based on GPR and LSTM. *Sustainability 2018* 10, 3693.

IEA (2021) Renewables 2021: Analysis and forecast to 2026. OECD Publishing.

J. Dudhia (2014). A history of mesoscale model development. *Asia-Pac. J. Atmos. Sci.* 50, 121-131.

Jensen, U., Pelgrum, E., H. Madsen (1994). The development of a forecasting model for the prediction of wind power production to be used in central dispatch centers. *Proceedings of European Wind Energy Conference,* 353-356.

Ji, L., Fu, C., Ju, Z., Shi, Y., Wu, S., L. Tao (2022). Short-Term Canyon Wind Speed Prediction Based on CNN—GRU Transfer Learning. *Atmosphere 2022* 13, 813.

Jiandong, D., Peng, W., Wentao, M., Shuai, F., H. Zequan (2022). A novel hybrid model based on nonlinear weighted combination for short-term wind power forecasting. *Electrical Power and Energy Systems* 134, 107452.

Kisvari, A., Lin, Z., X. Liu (2021). Wind power forecasting e A data-driven method along with gated recurrent neural network. *Renewable Energy* 163, 1895-1909.

"Kores Kocadag RES." https://www.dostenerji.com/en (accessed October 19, 2022).

L. Landberg (1998). A mathematical look at a physical power prediction model. *Wind Energy* 1, 23-8.

L. Landberg (1999). Short-term prediction of the power production from wind farms. *J Wind Eng Ind Aerodyn* 80, 207-20.

L. Landberg (2001). Short-term prediction of local wind conditions. *J Wind Eng Ind Aerodyn* 89, 235-45.

Lei, M., Shiyan, L., Chuanwen, J., Hongling, L., Z. Yan (2009). A review on the forecasting of wind speed and generated power. *Renewable and Sustainable Energy Reviews* 13, 915-920.

Li, L.L., Liu, Z.F., Tseng, M.L., Jantarakolica, K., M.K. Lim (2021). Using enhanced crow search algorithm optimization-extreme learning machine model to forecast short-term wind power. *Expert Systems with Applications* 184, 115579.

Li, P., Guan, X., Wu, J., X. Zhou (2015). Modeling Dynamic Spatial Correlations of Geographically Distributed Wind Farms and Constructing Ellipsoidal Uncertainty Sets for Optimization-based Generation Scheduling. *IEEE Transactions on Sustainable Energy* 6(4), 1594-1605.

Li, S., Li, W., C. Cook (2017). A fully trainable network with RNN-based pooling. *Neurocomputing* 338, 72-82.

Liao, S., Tian, X., Liu, B., Liu, T., Su, H., B. Zhou (2022). Short-Term Wind Power Prediction Based on LightGBM and Meteorological Reanalysis. *Energies* 15, 6287.

Liu, X., Lin, Z., Z. Feng (2021). Short-term offshore wind speed forecast by seasonal ARIMA – A comparison against GRU and LSTM. *Energy* 227, 120492.

Liu, H., Mi, X.W., Y.F. Li (2018). Wind speed forecasting method based on deep learning strategy using empirical wavelet transform, long short-term memory neural network and Elman neural network. *Energy Conv. Manag.* 156, 498–514.

Liu, H., Yu, C., Wu, H., Duan, Z., G. Yan (2020) A New Hybrid Ensemble Deep Reinforcement Learning Model for Wind Speed Short-term Forecasting. *Energy* 202, 117794.

Liu, X., Zhang, H., Kong, X., K.Y. Lee (2020). Wind Speed Forecasting using Deep Neural Network with Feature Selection. *Neurocomputing* 397, 393-403.

Louka, P., Galanis, G., Siebert, N., Kariniotakis, G., Katsafados, P., G. Kallos (2008). Improvements in Wind Speed Forecasts for Wind Power Prediction Purposes using Kalman Filtering. *Journal of Wind Engineering and Industrial Aerodynamics* 96(12), 2348-2362.

Lu, P., Ye, L., Zhao, Y., Dai, B., Pei, M., Z. Li (2021). Feature extraction of meteorological factors for wind power prediction based on variable weight combined method. *Renewable Energy* 179, 1925-1939.

Manigandan, P., Alam, M.S., Alharthi, M., Khan, U., Alagirisamy, K., Pachiyappan, D., A. Rehman (2021). Forecasting Natural Gas Production and Consumption in United States-Evidence from SARIMA and SARIMAX Models. *Energies* 14, 6021.

Moreno, J.J.M., Pol, A.P., Abad, A.S., B.C. Blasco (2013). Using the R-MAPE index as a resistant measure of forecast accuracy. *Psicothema* 25(4), 500-506.

Nefabas, K.L., Söder, L., Mamo, M., J. Olauson (2021) Modeling of Ethiopian Wind Power Production Using ERA5 Reanalysis Data. *Energies* 14, 2573.

Nie, Y., Liang, N., J. Wang (2021). Ultra-short-term wind-speed bi-forecasting system via artificial intelligence and a double-forecasting scheme. *Applied Energy,* 301:117452.

O. Fathi (2019). Time series forecasting using a hybrid ARIMA and LSTM model. *Velvet Consulting*.

O. Kalay (2018). Electricity Load and Price Forecasting of Turkish Electricity Markets. *Middle East Technical University*.

Oo, Z.Z., S. Phyu (2019). Microclimate Prediction Using Cloud Centric Model Based on IoT Technology for Sustainable Agriculture. *IEEE 4th International Conference on Computer and Communication Systems.*

Osowski, S., K. Garanty (2007). Forecasting of the daily meteorological pollution using wavelets and support vector machine. *Engineering Applications of Artificial Intelligence* 20(6), 745-755.

M. Peixeiro (2022). Time Series Forecasting in Python. *Manning Publications.*

M.G. Zhang (2015), Wind Resource Assessment and Micro-siting. *John Wiley & Sons Singapore Pte. Ltd.*

“Real-time Generation.” https://seffaflik.epias.com.tr/transparency/uretim/gerceklesen-uretim/gercek-zamanli-uretim.xhtml (accessed October 19, 2022).

S. Li (2003). Wind power prediction using recurrent multi-layer perceptron neural networks. *IEEE Power Engineering Society General Meeting.*

Shahid, F., Zameer, A., Mehmood, A., M.A.Z. Raja (2020). A novel wavelets long short-term memory paradigm for wind power prediction. *Applied Energy* 269, 115098.

Sharma, S.K., and S. Ghosh (2016). Short-term wind speed forecasting: application of linear and non-linear time series models. *Int J Green Energy 2016* 13, 1490–1500.

Shao, B., Song, D., Bian, G., Y. Zhao (2021). Wind Speed Forecast Based on the LSTM Neural Network Optimized by the Firework Algorithm. *Advances in Materials Science and Engineering 2021* 6, 1-13.

Siami-Namini, S., Tavakoli, N., A.S. Namin. A comparison of ARIMA and LSTM in forecasting time series. *International Conference on Machine Learning and Applications (ICMLA).*

Skamarock, W., Klemp, J., Dudhia, J., Gill, D., Barker, D., Duda, M., Huang, X., Wang, W., J. Power (2008). A description of the Advanced Research WRF Version 3, NCAR Technical Note. Mesoscale and Microscale Meteorology Division. *National Center for Atmospheric Research.*

Sreelakshmi, K., P.R. Kumar (2008). Performance evaluation of short-term wind speed prediction techniques. *Int J Comput Sci Network Security* 8, 162–9.

Tascikaraoglu, A., M. Uzunoglu (2014). A review of combined approaches for short-term prediction wind speed and power. *Renewable and Sustainable Energy Reviews* 34, 243-254.

Taylor, S.J., B. Letham (2018). Forecasting at scale. *The American Sratistician* 72(1), 37-45.

Thiyagarajan, K., Kodagoda, S., Ulapane, N., M. Prasad (2020). A Temporal Forecasting Driven Approach Using Facebook’s Prophet Method for Anomaly Detection in Sewer Air Temperature Sensor System.

Toharudin, T., Pontoh, R.S., Caraka, R.E., Zahroh, S., Lee, Y., R.C. Cheng (2020). Employing long short-term memory and Facebook prophet model in air temperature forecasting. *Communications in Statistics-Simulation and Computation.*

Tuna, F., F. Bingol (2018). Length scale parametrization and stability analyses with different statistical methods in wind measurements. *Department of Energy Engineering, Izmir Institute of Technology.*

“Urla RES.” https://www.egenda.com.tr/en/ (accessed October 19, 2022).

Vishwas, B., A. Patel (2020). Hands-on Time Series Analysis with Python. From Basics to Bleeding Edge Techniques. *Berkeley, CA: Apress.*

Wang, X., Sideratos, N., Hatziargyriou, L., L.H. Tsoukalas (2004). Wind Speed Forecasting for Power System Operational Planning. *International Conference on Probabilistic Methods Applied to Power Systems.*

Wang, Y., Liu, Y., Li, L., Infield, D., S. Han (2018). Short-term wind power forecasting based on clustering pre-calculated CFD method. *Energies* 11(4), 1-9.

Watson, S., Halliday, J., L. Landberg (1992). Assessing the economic benefits of numerical weather prediction model wind forecasts to electricity generating utilities. *Proceedings of 14th British Wind Energy Association Conference Nottingham (UK),* 291-297.

Wegley, H., Kosorok, M., W Fornica (1984). Subhourly wind forecasting techniques for wind turbine operations. *PNL-4894, Pacific Northwest Laboratory.*

Wendell, L., Wegley, H., M. Verholek (1978). Report from a working group meeting on wind forecasts for WECS. *PNL-2513, Pacific Northwest Laboratory.*

Wu, J., Li, N., Zhao, Y., J. Wang (2022). Usage of correlation analysis and hypothesis test in optimizing the gated recurrent unit network for wind speed forecasting. *Energy* 242, 122960.

Xie, A., Yang, H., Chen, J., Sheng, L., Q. Zhang (2021). A Short-Term Wind Speed Forecasting Model Based on a Multi-Variable Long Short-Term Memory Network. *Atmosphere 2021* 12, 651.

Yuan, C., Tang, Y., Mei, R., Mo, F., H. Wang (2021). A PSO-LSTM Model of Offshore Wind Power Forecast considering the Variation of Wind Speed in Second-Level Time Scale. *Mathematical Problems in Engineering 2021* 1, 1-9.

Z.O. Olaofe (2014). A 5-day wind speed & power forecasts using a layer recurrent neural network (LRNN). *Sustain Energy Technol Assess* 6, 1–24.

Zhou, J., Shi, J., G. Li (2011). Fine-tuning support vector machines for short-term wind speed forecasting. *Energy Conversion and Management* 52, 1990-1998.

Zhu, Q., Chen, J., Shi, D., Zhu, L., Bai, X., X. Duan (2019). Learning Temporal and Spatial Correlations Jointly: A Unified Framework for Wind Speed Prediction. *IEEE Transactions on Sustainable Energy* 11(1), 509-523.

APPENDIX A

PYTHON CODES

Facebook Prophet

```
import pandas as pd
import numpy as np
import xlsxwriter
from prophet import Prophet
from numpy import concatenate
from pandas import read_csv
from pandas import DataFrame
from pandas import concat
from datetime import datetime
from sklearn.metrics import *
from math import sqrt

#prophet univariate

all_df = pd.read_csv('iztech.csv', usecols = ['Name','WS101'])
all_df.columns = ['ds', 'y']

results=[]
for i in range(0, 57601, 2):
    df=all_df[i:i+144]
    m = Prophet(daily_seasonality=True, weekly_seasonality=True, ye
arly_seasonality=True)
    m.fit(df)
    future = m.make_future_dataframe(periods=144,freq='10 min')
    forecast = m.predict(future)
    result = forecast[['ds', 'yhat']]
    result = result.tail(2)
    results.append(result)

#prophet multivariate

train = pd.read_csv('iztech.csv', usecols = ['Name','WS101',
'RH3','T3','P2'])
train.rename(columns={'Name':'ds', 'RH3': 'add1', 'T3': 'add2',
'P2': 'add3', 'WS101': 'y'}, inplace=True)

results_multivariate=[]
for i in range(0, 57601, 2):
    X_train = train.iloc[i:i+144]
    X_val  = train.iloc[i+144:i+146]

    model = Prophet(daily_seasonality=True, weekly_seasonality=True
, yearly_seasonality=True)
    model.add_regressor('add1')
    model.add_regressor('add2')
    model.add_regressor('add3')
    model.fit(X_train)

    forecast_1 = model.predict(X_val.drop(columns="y"))
    result_1 = forecast_1[['ds', 'yhat']]
    results_multivariate.append(result_1)

# Create Pandas dataframes from forecasts
df1 = pd.read_csv('iztech.csv', usecols = ['Name','T3'])
df1.columns = ['ds', 'y']
df1=df1[144:57601]

df1 = pd.DataFrame(df1)
df2 = pd.DataFrame(np.concatenate(results),columns=['ds', 'yhat'])
df3 = pd.DataFrame(np.concatenate(results_multivariate), columns=['
ds', 'yhat'])

expected=df1['y']
prediction_multi=df2['yhat']
prediction_uni=df3['yhat']

MSE_multi = mean_squared_error(expected, prediction_multi)
R2_multi = r2_score(expected, prediction_multi)
RMSE_multi = sqrt(mean_squared_error(expected, prediction_multi))
MAE_multi = mean_absolute_error(expected, prediction_multi)

print(MSE_multi)
print(R2_multi)
print(RMSE_multi)
print(MAE_multi)
```

SARIMA

```
from statsmodels.graphics.tsaplots import plot_pacf
from statsmodels.graphics.tsaplots import plot_acf
from statsmodels.tsa.statespace.sarimax import SARIMAX
from statsmodels.tsa.stattools import adfuller
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
from tqdm import tqdm_notebook
import numpy as np
import pandas as pd
```

```
from itertools import product

from sklearn.metrics import *
from math import sqrt
import warnings
warnings.filterwarnings('ignore')

all_df = pd.read_csv('iztech.csv', usecols = ['Name','WS101'])

forecast=[]
for i in range(0, 57601, 2):
    data = all_df[i:i+144]

    def optimize_SARIMA(endog, parameters_list, d, D, s):

        results = []

        for param in tqdm_notebook(parameters_list):
            try:
                model = SARIMAX(endog, order=(param[0], d, param[1]
), seasonal_order=(param[2], D, param[3], s), simple_differencing=F
alse).fit(disp=False)
            except:
                continue

            aic = model.aic
            results.append([param, aic])

        result_df = pd.DataFrame(results)
        result_df.columns = ['(p,q)x(P,Q)', 'AIC']

        #Sort in ascending order, lower AIC is better
        result_df = result_df.sort_values(by='AIC', ascending=True)
.reset_index(drop=True)

        return result_df

    p = range(0, 3, 1)
    d = 1
    q = range(0, 3, 1)
    P = range(0, 3, 1)
    D = 1
    Q = range(0, 3, 1)
    s = 4

    parameters = product(p, q, P, Q)
    parameters_list = list(parameters)
    result_df = optimize_SARIMA(data['WS101'], parameters_list, 1,
1, 4)
    result_df_min=result_df[result_df.AIC == result_df.AIC.min()]
    p=result_df_min['(p,q)x(P,Q)'][0][0]
    q=result_df_min['(p,q)x(P,Q)'][0][1]
    P=result_df_min['(p,q)x(P,Q)'][0][2]
    Q=result_df_min['(p,q)x(P,Q)'][0][3]

    best_model = SARIMAX(data['WS101'], order=(p,d,q), seasonal_ord
er=(P,D,Q,s), simple_differencing=False)
    res = best_model.fit(disp=False)

    n_forecast = 2
    predict = res.get_prediction(end=best_model.nobs + n_forecast)

    prediction=predict.predicted_mean[-n_forecast:]
    forecast.append(prediction)

    df=all_df[144:57601]
    df1 = pd.DataFrame(np.concatenate(forecast))
    expected=df['WS101']
    prediction=df1

    MSE = mean_squared_error(expected, prediction)
    R2 = r2_score(expected, prediction)
    RMSE = sqrt(mean_squared_error(expected, prediction))
    MAE = mean_absolute_error(expected, prediction)

    print(MSE)
    print(R2)
    print(RMSE)
    print(MAE)
```

SARIMAX

```
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.statespace.sarimax import SARIMAX
from statsmodels.tsa.stattools import adfuller
import statsmodels.api as sm
from sklearn.metrics import mean_squared_error
from tqdm import tqdm_notebook
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from itertools import product
import warnings
from sklearn.metrics import *
from math import sqrt
```

```
warnings.filterwarnings('ignore')
%matplotlib inline

macro_data = pd.read_csv('iztech.csv', usecols = ['Name','T3','RH3'
,'P2','WS101'])

forecast=[]
for i in range(0, 144, 2):
    data = macro_data[i:i+144]
    def optimize_SARIMAX(endog, exog, parameters_list, d, D, s):
        """
            Returns dataframe with parameters, corresponding AIC

            endog - the observed variable
            exog - the exogenous variables
            parameters_list - list with (p, d, P, Q)tuples
            d - integration order
            D - seasonal integration order
            s - length of the season
        """

        results = []

        for param in tqdm_notebook(parameters_list):
            try:
                model = SARIMAX(endog,
                               exog,
                               order=(param[0], d, param[1]),
                               seasonal_order=(param[2], D, param[3
], s),

                               simple_differencing=False).fit(disp=
False)
            except:
                continue

            aic = model.aic
            results.append([param, aic])

        result_df = pd.DataFrame(results)
        result_df.columns = ['(p,q)x(P,Q)', 'AIC']
        result_df = result_df.sort_values(by='AIC', ascending=True)
.reset_index(drop=True)

        return result_df
    p = range(0, 3, 1)
    d = 1
    q = range(0, 3, 1)
    P = range(0, 3, 1)
    D = 0
    Q = range(0, 3, 1)
    s = 4

    parameters = product(p, q, P, Q)
    parameters_list = list(parameters)

    endog = macro_data['WS101'][i:i+142]
    exog = macro_data[['T3','RH3','P2']][i:i+142]

    result_df = optimize_SARIMAX(endog, exog, parameters_list, 1, 0
, 4)
    result_df_min=result_df[result_df.AIC == result_df.AIC.min()]
    p=result_df_min['(p,q)x(P,Q)'][0][0]
    q=result_df_min['(p,q)x(P,Q)'][0][1]
    P=result_df_min['(p,q)x(P,Q)'][0][2]
    Q=result_df_min['(p,q)x(P,Q)'][0][3]

    best_model = SARIMAX(endog,
                     exog,
                     order=(p,d,q),
                     seasonal_order=(P,D,Q,s),
                    simple_differencing=False)
    res = best_model.fit(dis=False)

    n_forecast = 2
    predict = res.get_prediction(end=best_model.nobs + n_forecast,
                            exog = exog.iloc[-3:])
    prediction=predict.predicted_mean[-n_forecast:]
    forecast.append(prediction)

    df=macro_data['WS101']
    expected=df[144:57601]
    prediction=pd.DataFrame(np.concatenate(forecast))

    MSE = mean_squared_error(expected, prediction)
    R2 = r2_score(expected, prediction)
    RMSE = sqrt(mean_squared_error(expected, prediction))
    MAE = mean_absolute_error(expected, prediction)

    print(MSE)
    print(R2)
    print(RMSE)
    print(MAE)
```

GRU

```
import pandas as pd
import numpy as np
```

```
from math import sqrt
from numpy import concatenate
from matplotlib import pyplot
from pandas import read_csv
from pandas import DataFrame
from pandas import concat
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import mean_squared_error
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM, GRU
import tensorflow as tf
from datetime import datetime
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (15,5)
from sklearn.metrics import *
from math import sqrt


dataset = read_csv("iztech.csv",
                   parse_dates={'dt' : ['Name']},
                   infer_datetime_format=True,
                   index_col= 0,
                   na_values=['nan','?'])
dataset.fillna(0, inplace=True)
values = dataset.values

dataset.head(4)

dataset.drop(columns  = [ 'WS101Max', 'WS101Min', 'WS101Count', 'WS76', 'WS76Max', 'WS76Min', 'WS76Std','WS76Count','WD28Max','WD28Min','WD28Std','WD28Count','rho10','rho10Max','rho10Std'], inplace = True)

dataset.drop(columns  = [ 'WS101Std', 'WS30', 'WS30Max', 'WS30Min', 'WS30Std', 'WS30Count', 'WD74','WD74Max','WD74Min','WD74Std','WD74Count','WD28'], inplace = True)

dataset = dataset[[ 'T3','P2','RH3', 'WS101']]

values = dataset.values
# ensure all data is float
values = values.astype('float32')

# normalizing input features
scaler = MinMaxScaler(feature_range=(0, 1))
scaled = scaler.fit_transform(values)
scaled =pd.DataFrame(scaled)
```

```
scaled.head(4)

def create_ts_data(dataset, lookback=1, predicted_col=3):
    temp=dataset.copy()
    temp["id"]= range(1, len(temp)+1)
    temp = temp.iloc[:-lookback, :]
    temp.set_index('id', inplace =True)
    predicted_value=dataset.copy()
    predicted_value = predicted_value.iloc[lookback:,predicted_col]
    predicted_value.columns=["Predcited"]
    predicted_value= pd.DataFrame(predicted_value)

    predicted_value["id"]= range(1, len(predicted_value)+1)
    predicted_value.set_index('id', inplace =True)
    final_df= pd.concat([temp, predicted_value], axis=1)
    #final_df.columns = ['var1(t-1)', 'var2(t-1)', 'var3(t-
1)', 'var4(t-1)', 'var5(t-1)', 'var6(t-1)', 'var7(t-1)', 'var8(t-
1)','var1(t)']
    #final_df.set_index('Date', inplace=True)
    return final_df

#We now create the time series dataset with looking back one time s
tep

reframed_df= create_ts_data(scaled, 1,3)
reframed_df.fillna(0, inplace=True)

reframed_df.columns = ['var1(t-1)', 'var2(t-1)', 'var3(t-
1)', 'var4(t-1)', 'var5(t-1)',]
print(reframed_df.head(4))

# split into train and test sets
values = reframed_df.values
training_sample =int( len(dataset) *0.7)
train = values[:training_sample, :]
test = values[training_sample:, :]
# split into input and outputs
train_X, train_y = train[:, :-1], train[:, -1]
test_X, test_y = test[:, :-1], test[:, -1]

# reshape input to be 3D [samples, time steps, features]
train_X = train_X.reshape((train_X.shape[0], 1, train_X.shape[1]))
test_X = test_X.reshape((test_X.shape[0], 1, test_X.shape[1]))
print(train_X.shape, train_y.shape, test_X.shape, test_y.shape)

model_gru = Sequential()
model_gru.add(GRU(75, return_sequences=True,input_shape=(train_X.sh
ape[1], train_X.shape[2])))
```

```
model_gru.add(GRU(units=30, return_sequences=True))
model_gru.add(GRU(units=30))
model_gru.add(Dense(units=1))
model_gru.compile(loss='mae', optimizer='adam')
model_gru.summary()

# fit network
gru_history = model_gru.fit(train_X, train_y, epochs=10,validation_data=(test_X, test_y), batch_size=64, shuffle=False)

pred_y =  model_gru.predict(test_X)
pred_y_1 =  model_gru.predict(train_X)

#dont run this cell if you are running this cell than add "validation_data=(test_X, test_y)" in model_gru.fit()
pyplot.plot(gru_history.history['loss'], label='GRU train', color='brown')
pyplot.plot(gru_history.history['val_loss'], label='GRU test', color='blue')
pyplot.legend()
pyplot.show()

test_y.reshape(17323,1)

MSE = mean_squared_error(test_y, pred_y)
R2 = r2_score(test_y, pred_y)
RMSE = sqrt(mean_squared_error(test_y, pred_y))
MAE = mean_absolute_error(test_y, pred_y)


print(MSE)
print(R2)
print(RMSE)
print(MAE)

#plotting predicted test value vs actual test value
plt.plot(test_y, label = 'Actual')
plt.plot(pred_y, label = 'Predicted')
plt.legend()
plt.show()

#visualization over full data
tra = np.concatenate([train_X,test_X])
tes = np.concatenate([train_y,test_y])
fp = model_gru.predict(tra)
plt.plot(tes, label = 'Actual')
plt.plot(fp, label = 'Predicted')
plt.legend()
plt.show()
```

```
#inverse normalizing
dataset_1 = dataset[['WS101']]
dataset_1 = dataset_1[40421:57744]
dataset_1.head()

values_1 = dataset_1.values
# ensure all data is float
values_1 = values_1.astype('float32')

# normalizing input features
scaler_1 = MinMaxScaler(feature_range=(0, 1))
scaled_1 = scaler_1.fit_transform(values_1)
scaled_1 =pd.DataFrame(scaled_1)

scaled_1.head(4)

inversed_1 = scaler_1.inverse_transform(scaled_1)
print(inversed_1)
print(values_1)
print(scaled_1)

test_y_inv = scaler_1.inverse_transform([test_y])
test_y_inv=test_y_inv.reshape(17323,1)
pred_y_inv = scaler_1.inverse_transform(pred_y)
pred_y_1_inv = scaler_1.inverse_transform(pred_y_1)
test_X_1=dataset['WS101']
test_X_1=test_X_1[:40420]

MSE = mean_squared_error(test_y_inv, pred_y_inv)
R2 = r2_score(test_y_inv, pred_y_inv)
RMSE = sqrt(mean_squared_error(test_y_inv, pred_y_inv))
MAE = mean_absolute_error(test_y_inv, pred_y_inv)

print(MSE)
print(R2)
print(RMSE)
print(MAE)
```

LSTM

```
import pandas as pd
import numpy as np
from math import sqrt
from numpy import concatenate
from matplotlib import pyplot
from pandas import read_csv
from pandas import DataFrame
from pandas import concat
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import mean_squared_error
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM, GRU
import tensorflow as tf
from datetime import datetime
from sklearn.metrics import *
from math import sqrt
from sklearn.metrics import *
from math import sqrt

dataset = read_csv("iztech.csv",
                   parse_dates={'dt' : ['Name']},
                   infer_datetime_format=True,
                   index_col= 0,
                   na_values=['nan','?'])
dataset.fillna(0, inplace=True)
values = dataset.values
dataset.head(4)

dataset.drop(columns  = [ 'WS101Max', 'WS101Min', 'WS101Count', 'WS
76', 'WS76Max', 'WS76Min', 'WS76Std','WS76Count','WD28Max','WD28Min
','WD28Std','WD28Count','rho10','rho10Max','rho10Std'], inplace = T
rue)
dataset.drop(columns  = [ 'WS101Std', 'WS30', 'WS30Max', 'WS30Min',
 'WS30Std', 'WS30Count', 'WD74','WD74Max','WD74Min','WD74Std','WD74
Count','WD28'], inplace = True)

dataset = dataset[[ 'T3','P2','RH3', 'WS101']]

values = dataset.values
# ensure all data is float
values = values.astype('float32')

# normalizing input features
scaler = MinMaxScaler(feature_range=(0, 1))
scaled = scaler.fit_transform(values)
scaled =pd.DataFrame(scaled)

def create_ts_data(dataset, lookback=1, predicted_col=3):
    temp=dataset.copy()
    temp["id"]= range(1, len(temp)+1)
    temp = temp.iloc[:-lookback, :]
    temp.set_index('id', inplace =True)
    predicted_value=dataset.copy()
    predicted_value = predicted_value.iloc[lookback:,predicted_col]
    predicted_value.columns=["Predcited"]
    predicted_value= pd.DataFrame(predicted_value)

    predicted_value["id"]= range(1, len(predicted_value)+1)
    predicted_value.set_index('id', inplace =True)
    final_df= pd.concat([temp, predicted_value], axis=1)
    #final_df.columns = ['var1(t-1)', 'var2(t-1)', 'var3(t-
1)', 'var4(t-1)', 'var5(t-1)', 'var6(t-1)', 'var7(t-1)', 'var8(t-
1)','var1(t)']
    #final_df.set_index('Date', inplace=True)
    return final_df

#We now create the time series dataset with looking back one time s
tep

reframed_df= create_ts_data(scaled, 1,3)
reframed_df.fillna(0, inplace=True)

reframed_df.columns = ['var1(t-1)', 'var2(t-1)', 'var3(t-
1)', 'var4(t-1)', 'var5(t-1)',]
print(reframed_df.head(4))

# split into train and test sets
values = reframed_df.values
training_sample =int( len(dataset) *0.7)
train = values[:training_sample, :]
test = values[training_sample:, :]
# split into input and outputs
train_X, train_y = train[:, :-1], train[:, -1]
test_X, test_y = test[:, :-1], test[:, -1]

# reshape input to be 3D [samples, time steps, features]
train_X = train_X.reshape((train_X.shape[0], 1, train_X.shape[1]))
test_X = test_X.reshape((test_X.shape[0], 1, test_X.shape[1]))
print(train_X.shape, train_y.shape, test_X.shape, test_y.shape)


model_lstm = Sequential()
model_lstm.add(LSTM(75, return_sequences=True,input_shape=(train_X.
shape[1], train_X.shape[2])))
model_lstm.add(LSTM(units=30, return_sequences=True))
model_lstm.add(LSTM(units=30))
model_lstm.add(Dense(units=1))
model_lstm.compile(loss='mae', optimizer='adam')
model_lstm.summary()

# fit network
```

```
lstm_history = model_lstm.fit(train_X, train_y, epochs=10,validation_data=(test_X, test_y), batch_size=64, shuffle=False)

pred_y =  model_lstm.predict(test_X)
pred_y_1 =  model_lstm.predict(train_X)

#dont run this cell if you are running this cell than add "validation_data=(test_X, test_y)" in model_gru.fit()
pyplot.plot(lstm_history.history['loss'], label='lstm train', color='brown')
pyplot.plot(lstm_history.history['val_loss'], label='lstm test', color='blue')
pyplot.legend()
pyplot.show()

test_y.reshape(17323,1)

MSE = mean_squared_error(test_y, pred_y)
R2 = r2_score(test_y, pred_y)
RMSE = sqrt(mean_squared_error(test_y, pred_y))
MAE = mean_absolute_error(test_y, pred_y)


print(MSE)
print(R2)
print(RMSE)
print(MAE)

#plotting predicted test value vs actual test value
plt.plot(test_y, label = 'Actual')
plt.plot(pred_y, label = 'Predicted')
plt.legend()
plt.show()

#visualization over full data
tra = np.concatenate([train_X,test_X])
tes = np.concatenate([train_y,test_y])
fp = model_lstm.predict(tra)
plt.plot(tes, label = 'Actual')
plt.plot(fp, label = 'Predicted')
plt.legend()
plt.show()

# inverse normalizing
dataset_1 = dataset[['WS101']]
dataset_1 = dataset_1[40421:57744]
dataset_1.head()

values_1 = dataset_1.values
```

```
# ensure all data is float
values_1 = values_1.astype('float32')

# normalizing input features
scaler_1 = MinMaxScaler(feature_range=(0, 1))
scaled_1 = scaler_1.fit_transform(values_1)
scaled_1 =pd.DataFrame(scaled_1)

scaled_1.head(4)

inversed_1 = scaler_1.inverse_transform(scaled_1)
print(inversed_1)
print(values_1)
print(scaled_1)

test_y_inv = scaler_1.inverse_transform([test_y])
test_y_inv=test_y_inv.reshape(17323,1)

pred_y_inv = scaler_1.inverse_transform(pred_y)
pred_y_inv

pred_y_1_inv = scaler_1.inverse_transform(pred_y_1)
pred_y_1_inv

test_X_1=dataset['WS101']
test_X_1=test_X_1[:40420]

MSE = mean_squared_error(test_y_inv, pred_y_inv)
R2 = r2_score(test_y_inv, pred_y_inv)
RMSE = sqrt(mean_squared_error(test_y_inv, pred_y_inv))
MAE = mean_absolute_error(test_y_inv, pred_y_inv)

print(MSE)
print(R2)
print(RMSE)
print(MAE)
```