A. ÖZTEKİN

ATILIM UNIVERSITY 2024

UNIVARIATE TIME SERIES METHODOLOGY FOR WIND ENERGY BASED ON HYBRID DEEP LEARNING MODELS

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

ANASTASYA ÖZTEKİN

A MASTER OF SCIENCE THESIS
IN
THE DEPARTMENT OF INDUSTRIAL ENGINEERING

MAY 2024

UNIVARIATE TIME SERIES METHODOLOGY FOR WIND ENERGY BASED ON HYBRID DEEP LEARNING MODELS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

BY

ANASTASYA ÖZTEKİN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
THE DEPARTMENT OF INDUSTRIAL ENGINEERING

MAY 2024

Approval of the Graduate School of Natural and Applied Sciences, Atilim University.

_____

Prof. Dr. Ender Keskinkılıç
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science in Department of Industrial Engineering, Atılım University.

_____

Prof. Dr. Turan Erman ERKAN
Head of Department

This is to certify that we have read the thesis UNIVARIATE TIME SERIES METHODOLOGY FOR WIND ENERGY BASED ON HYBRID DEEP LEARNING MODELS submitted by ANASTASYA ÖZTEKIN and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Assoc. Prof. Dr. Kamil Demirberk ÜNLÜ
Supervisor

Examining Committee Members:

Assoc. Prof. Dr. Yıldırım Akbal
Applied Data Science Master Program, TEDU _____

Assoc. Prof. Dr. Kamil Demirberk ÜNLÜ
Industrial Eng. Department, Atılım University _____

Prof. Dr. Turan Erman ERKAN
Industrial Eng. Department, Atılım University _____

Date: 29 May 2024

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Anastasya Öztekin

Signature :

ABSTRACT

UNIVARIATE TIME SERIES METHODOLOGY FOR WIND ENERGY BASED ON HYBRID DEEP LEARNING MODELS

ÖZTEKİN, Anastasya
MS., Department of Industrial Engineering
Supervisor : Assoc. Prof. Dr. Kamil Demirberk ÜNLÜ

May 2024, 76 pages

Wind power is presently considered the most favorable renewable energy alternative, gathering global concern. It contributes to the preservation of environmental resources on a global scale. Reliable wind energy forecasting is crucial for efficiently employing renewable energy sources, given their fluctuating patterns. In this study, statistical models based on time series data have been widely used to address this concern.

This study aims to forecast the wind energy generated by the wind farms in Karaburun, Izmir, Turkey by developing univariate hybrid models based on Sequence-to-Sequence and Convolutional Neural Network approaches. The study's forecasting range spans from short-term to long-term periods. Numerous hybrid models were developed and trained using the data to identify the most accurate forecast. Benchmarking results reveal that the hybrid model combining Convolution Neural Network with Sequence-to-Sequence stacked with Long-Short Term Memory cells yields the most accurate forecasts for both short and long terms. Based on actual data, the hybrid model provides significant forecasting results for short-term. In long-range forecasts, even though there is a decrease in the coefficient of variation compared to

short-range forecasts, determination metrics such as mean squared error, mean absolute error, and mean absolute percentage error indicate that the model maintains accuracy in the long-term forecasts.

Keywords: Time Series Methodology, Wind Energy, Seq2Seq, CNN, LSTM, Turkey

ÖZ

RÜZGAR ENERJISI İÇIN HIBRIT DERIN ÖĞRENME MODELLERINE DAYALI TEK DEĞIŞKENLI ZAMAN SERISI METODOLOJISI

ÖZTEKİN, Anastasya
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. Kamil Demirberk ÜNLÜ

Mayıs 2024, 76 sayfa

Rüzgar enerjisi günümüzde en çok tercih edilen yenilenebilir enerji alternatifi olarak kabul edilmekte ve küresel bir ilgi toplamaktadır. Ayrıca, rüzgar enerjisinin etkin kullanımı küresel ölçekte çevresel kaynakların korunmasına büyük ölçüde katkıda bulunmaktadır. Rüzgar enerjisinin değişkenliği göz önünde bulundurulduğunda, yenilenebilir enerji kaynaklarının verimli bir şekilde kullanılması için güvenilir bir tahmin geliştirmek çok önemlidir.

Literatürde rüzgar enerjisinin doğru tahmini için birden fazla model geliştirilmiştir. Bu çalışmada bu sorunu gidermek için zaman serisi verilerine dayalı istatistiksel modeller kullanılmıştır. Bu çalışma, Diziden Diziye ve Evrişimli Sinir Ağı yaklaşımlarına dayalı tek değişkenli hibrit modeller geliştirerek Karaburun, İzmir, Türkiye'deki rüzgar santralleri tarafından üretilen rüzgar enerjisini tahmin etmeyi amaçlamaktadır. Çalışmanın tahmin aralığı kısa vadeden uzun vadeye kadar uzanmaktadır. En doğru tahmini belirlemek adına gerçek veriler kullanılarak çok sayıda hibrit model geliştirilmiştir. Karşılaştırma sonuçları, Evrişimli Sinir Ağı'nı Uzun Kısa Süreli Bellek hücresiyle istiflenmiş Diziden Diziye ile birleştiren hibrit modelin hem kısa hem de

uzun vadede en doğru tahminleri sağladığını ortaya koymaktadır. Geliştirilen hibrit model, kısa vadeli tahminler için önemli bir değişim katsayısı ortaya koyar. Uzun vadeli tahminlerde, kısa vadeli tahminlere göre değişim katsayısında bir azalma olsa da ortalama hataların karesi, ortalama mutlak hata ve ortalama mutlak yüzde hata gibi belirleme metrikleri, modelin uzun vadeli tahminlerde doğruluğunu koruduğunu göstermektedir.

Anahtar Kelimeler: Zaman Serisi Metodolojisi, Rüzgar Enerjisi, Seq2Seq, CNN, LSTM, Turkey

To My Family

ACKNOWLEDGMENTS

I would like to express my sincere gratitude and appreciation to the individuals and organizations who have played a pivotal role in the completion of my thesis.

Firstly, I would like to extend my sincere appreciation to my valuable advisor, Demirberk ÜNLÜ, for all of his assistance during this study, from selecting the thesis topic through conducting the research. I offer sincere thanks for his valuable insights and encouragement during the study. His contribution and guidance play a vital role in this thesis.

I would like to acknowledge the members of my thesis dissertation committee, Assoc. Prof. Dr. Yıldırım Akbal, Assoc. Prof. Dr. Kamil Demirberk Ünlü and Prof. Dr. Turan Erman Erkan, for their valuable support and suggestions, which helped to accomplish this thesis.

Additionally, I offer sincere thanks to the Scientific and Technical Research Council of Turkey (TUBITAK) for graduation scholarship provided by TUBITAK 2210-A and project assistance provided by TUBITAK 1002-A (Grant number: 123F064). During this process, it was a great motivation for me to work on projects for TUBITAK.

Finally, I want to express my heartfelt thanks to my dear family and friends for their support and presence, which have been the main motivation and encouragement for me.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Climate change stands out as one of the most urgent challenges in our contemporary world, causing substantial and detrimental effects on the environment. In this context, human activities play a significant role; in particular, energy use has been found to be a key factor in the most recent shifts in climate patterns [1]. In the course of electricity production using traditional resources, numerous harmful materials are discharged into both water bodies and the atmosphere, and the environment is additionally contaminated by airborne dust particles. Furthermore, traditional energy sources like coal, natural gas, and oil are insufficient to fulfill the demands of the worldwide economy [2]. To address forthcoming environmental transformations, it is crucial to implement various measures, including a shift away from existing energy generation technologies. Conventional methods like coal combustion have harmful consequences for the environment. Therefore, nations worldwide are increasingly adopting eco-friendly energy generation approaches, such as solar and wind power [3].

According to the Energy Information Administration (EIA), the utilization of renewable energy sources has experienced the most rapid growth in recent times [4]. Advanced economies are actively advocating for renewable sources as a means to enhance their energy supply security and manage their greenhouse gas emissions (GHG) [5]. The promotion of renewable energy options contributes to national objectives for sustainability and economic growth, in addition to assisting modernize the energy industry [6]. The adoption and growth of renewable energy sources (RES) within a nation provide assurance of access to local and sustainable resources. Consequently, the country gains a greater pool of resources for energy generation. This, in turn, leads to increased resilience during times of crisis, which may stem from

the exhaustion of conventional energy supplies and additionally from geopolitical and economic disputes. Importantly, it reduces the country's dependence on foreign resource imports [7]. Consequently, just like many other regions around the world, Turkey has experienced a favorable trend marked by the growing contribution of renewable energy sources to its overall energy mix.

Turkey is heavily reliant on the use of fossil fuels, which are classified as primary energy and mostly comprise oil, coal, and natural gas. Primary energy consumption has reached 75% foreign dependence, while electricity consumption has reached 60% [8]. For this reason, there is a need for domestic, inexpensive, continuous, reliable, and sustainable solutions in energy, as well as well-designed new energy policies, strategies, and programs. Turkey, on the other hand, currently uses renewable energy sources in terms of sustainability, economics, and the environment. Renewable energy not only reduces the foreign dependency of countries but also creates new business potential in the relevant country.

Approximately 20% of the market share in the wind power sector is held by the state-owned Electricity Generation Company (EÜAŞ), with numerous private companies also operating in the sector. The record for the highest daily share of wind power, reaching 25%, was achieved in 2022. Turkey currently boasts approximately 300 onshore wind farms, collectively housing roughly 4,000 wind turbines. By 2022, the cumulative installed capacity had surpassed 11 gigawatts (GW), with an impressive capacity factor averaging around 40%. Unlicensed installations are limited to 5 MW for maximum power output. In 2021, investments in wind energy reached one billion euros, resulting in the construction of 1.4 GW of capacity, with an average power rating exceeding 5 GW, surpassing other European countries' onshore capacities. Soma stands as the largest wind farm in Turkey as of 2021, followed by Karaburun. The Energy Ministry's assessment suggests that the onshore wind potential in Turkey stands at around 48 GW, specifically in areas where wind speeds exceed 7.5 m/s at a height of 50 meters. This estimation is based on the assumption of utilizing 5 MW capacity turbines. The north-western regions of Turkey are identified as the windiest, with an average wind speed of approximately 7 m/s at the same altitude, and

consequently, these areas host the majority of wind farms. Additionally, the mountain ranges in the western part of the country are oriented perpendicular to the coast, facilitating the inland flow of wind [9].

Precise and effective forecasting is critical for increasing renewable energy consumption rates, particularly for wind power-generating facilities. Day-ahead wind energy predictions are essential for unit responsibility management and commodities markets. An overestimation of the amount of power generated will result in the provision of extra energy and unnecessary operating costs. Underestimating causes the need to acquire necessary items at a greater cost and causes an unexpected deployment of extra peak capacity. The importance of forecasting motivates research into developing a more precise forecasting system, which will lower operation and maintenance costs significantly and improve power supply and delivery system dependability [10]. Utility businesses reduce risk by using accurate electricity forecasts. The ability to make financially viable judgments about future investments in generation and transmission is made possible by the company's ability to strategically plan with the help of the long-term forecast for electricity production. Forecasting the amount of electricity generated is essential for future planning, which includes taking future production plant location, size, and type into account. Utilities can create electricity near the load by identifying regions with high or growing demand, which eliminates the need for large transmission and distribution facilities.

Numerous predictive studies have been conducted, focusing on forecasting ramping occurrences, projecting uncertainties, and predicting wind velocity and energy across various timeframes. These investigations aim to address these specific objectives and contribute valuable insights to the field.

Over the years, the study of renewable power forecasting has employed a diverse set of methodologies. The multifaceted nature of these methodologies reflects the complexity of accurately forecasting renewable power generation.

Wind energy time-series systems can be categorized as short-term, mid-term and long-term modeling approaches. Short-term forecasting involves the use of various

modeling techniques, such as time series models, convolutional neural networks, and Kalman Filtering (KF) [11]. These methodologies contribute to accurate predictions within shorter time horizons, offering valuable insights for immediate decision-making in the context of wind energy generation.

Traditional statistical techniques, like Autoregressive Integrated Moving Average (ARIMA) models, offer an infrastructure for modeling time series data by considering elements such as trends, seasonality, and irregular components [12]. They work well when dealing with stationary time series data. However, when faced with complex temporal patterns, these methods encounter difficulties in capturing the relationships between continuous independent variables and the intended series, which restricts their effectiveness in forecasting complex time series [13].

In recent years, the availability of more extensive data and increased computing power have elevated the importance of machine learning in the evolution of time-series forecasting models. Models are developed using algorithms in machine learning techniques that may identify patterns and correlations in data. The use of feature engineering is usually needed for these models, which choose or develop appropriate features from the actual data to support forecasting. Support Vector Machines (SVM), gradient boosting machines, decision trees, Random Forests (RF), and linear regression are examples of machine learning models that are frequently used for forecasting [14]. Owing to their ability to match nonlinear patterns, these learning-based approaches excel at identifying hidden feature representations in time series data.

One of the branches of machine learning is deep learning, which incorporates the use of multi-layered artificial neural networks to extract complex patterns and interpretations from data. Deep learning models such as convolutional neural network (CNN), long short-term memory network (LSTM), and recurrent neural network (RNN) have demonstrated notable performance in a variety of forecasting models, particularly when working with time-series data [15]. Because time-series problems span diverse domains, a multitude of neural network design options have emerged. In

time-series forecasting, where time-dependent variables are essential, deep learning models are more suited to handle complicated patterns and vast volumes of data than traditional machine learning models, which may be more appropriate for simple forecasting tasks with clearly defined variables. A notable recent trend in deep learning involves the development of hybrid models designed to address the limitations of traditional approaches.

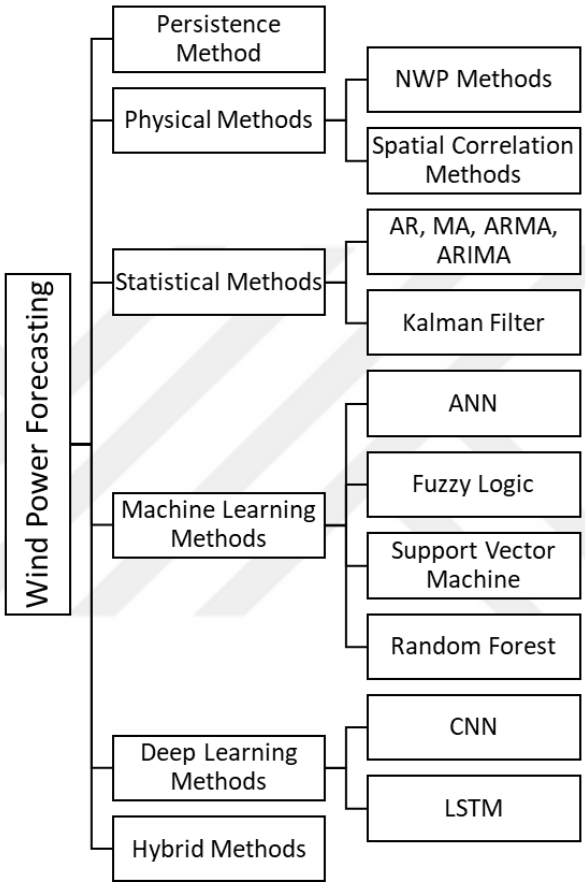


Figure 1.1 Wind Power Forecasting Techniques

The methods developed to forecast wind power are summarized in Figure 1.1. As mentioned, there are multiple approaches for estimating wind energy. Wind power forecasting (WPF) can be generally categorized into six main subsections, such as persistence methods, physical methods, statistical methods, machine learning methods, deep learning methods, and hybrid methods. For the physical methods,

numerical weather prediction (NWP) and spatial correlation methods can be used. Among the statistical methods are Autoregressive (AR), Moving Average (MA), Autoregressive Moving Average (ARMA) and ARIMA, and the KF. Artificial Neural Network (ANN), Fuzzy Logic, SVM, and RF are examples of machine learning methods [16]. The most frequently used deep learning methods are CNN and LSTM. Finally, there are hybrid approaches that tend to yield the most precise forecasts.

Wind energy is a clean and renewable energy source that is increasingly used in Turkey. The Aegean and Marmara regions of Turkey have the greatest amount of wind energy potential; 33.7% of the installed power is in İzmir, 25.5% in Balıkesir, roughly 17.7% in Çanakkale, 13.7% in Manisa, and 9.4% in Istanbul [17]. Based on the latest data received in December 2022 and a report published by Türkiye Elektrik Üretim A.Ş., it was demonstrated that Turkey completed the year 2022 with an installed power of 103,809.3 MW and 11,427 power plants. The installed power of wind power plants increased by 30.6 MW compared to the previous month and reached 11,396.2 MW [18]. However, wind power generation fluctuates naturally due to the variability of weather conditions.

This study aims to provide a new method for medium and long-term wind energy production forecasting. This method is expected to provide more accurate results in wind energy production forecasts than other methods and can therefore be used to optimize activities in the wind energy sector. Additionally, this study may offer a new perspective on research on the use of hybrid deep learning models in wind energy production forecasting. Within the scope of the study, the electricity production of the electricity production facility in Karaburun, Izmir, will be modeled using deep learning methods, in order to forecast the production volume of renewable energy resources efficiently and to contribute to the increase in employment to be provided with the data obtained. In the modeling part, the first hybrid deep learning models will be proposed. In the second phase of the study, the proposed models will be compared with the methods in the literature by prediction metrics. When creating hybrid methods, Sequence-to-Sequence (Seq2Seq) learning mechanisms with different structures will first be taken into consideration. In the third step, the created method will be trained

for medium and long-term forecasting. When the existing literature is examined, hybrid structure methods have generally been used for short-term forecasting [19]. The main objective of this study is to train such models for medium and long-term forecasting. It is thought that this method, with its univariate structure, will enable successful prediction results to be achieved at stations in different regions.

The main question of the research can be summarized as follows: "Is it possible to make medium and long-term wind energy production forecasts with high success using a hybrid deep learning model?" Can this method provide more accurate results compared to other methods?

The data obtained from [20] will be used in the study. The Karaburun Wind Power Plant (WPP), which is situated in the Karaburun region of Izmir, was selected as the study's sample station. With a total installed power of 226.80 MWe, Karaburun WPP is the 79th largest power plant in Turkey and the fourth largest in Izmir [21]. The data set, which is expressed in MWh, represents the hourly wind energy production. It contains a total of 83451 observations.

CHAPTER 2

LITERATURE REVIEW

In this section, the literature on wind power forecasting will be reviewed. Deep learning algorithms, machine learning algorithms, and hybrid approaches can be used to categorize the studies that have been done. These algorithms have proven to be effective in various aspects of wind energy systems. Due to their superior accuracy, studies employing hybrid approaches are highlighted more extensively in this section.

Currently, there is significant importance placed on techniques for forecasting wind power. This is largely due to the rapid global expansion of wind power generation, which serves purposes such as understanding wind behavior, assessing available resources, and predicting power output. It is crucial to develop precise methods for forecasting both wind patterns and power output to minimize inaccuracies resulting from uncertain wind conditions and improvements in wind energy conversion technology. [22, 23]

There are four distinct categories of wind power forecasting: very short-term, short-term, mid-term, and long-term. These categories are based on different time frames: a few seconds to thirty minutes, thirty minutes to six hours, six hours to twenty-four hours, and one to seven days, respectively. These forecasting approaches have specific applications. Short-term forecasting is utilized for forecasting wind speeds and estimating power output from wind turbines, while long-term forecasting plays a crucial role in planning wind turbine maintenance programs. [24]

Recent research has capitalized on the progress made in machine learning technology. This has led to the development and widespread adoption of novel deep learning algorithms, especially for the prediction of time series data. Notably, within this

spectrum of algorithms, LSTM, a type of RNN, has gained prominence. These LSTM networks have proven particularly effective in achieving precise forecasts of wind speeds [25].

A prediction model that is based on LSTM and offers short-term forecasts for wind and power generation utilizing data gathered from the northwestern region of Turkey is proposed by [26]. When comparing the prediction results with the MA and multilayer perceptron (MLP) techniques, the proposed model demonstrates superior performance with a lower coefficient of determination (R^2) value of 0.9366.

ANN has emerged as a highly promising technology with applications spanning various domains, including wind energy assessments. ANN's versatility encompasses tasks like pattern recognition, approximation, and time-series prediction, making it a valuable tool in the field of wind energy. A multitude of distinct ANN models have been documented in the literature, showcasing their effectiveness in forecasting wind speeds across a range of time frames, spanning from just a few seconds to as far as one week ahead [24].

Short-term wind speed prediction in Mexico, utilizing ANN methodology, is applied to hourly time series data collected from the site. The two-layer model with just two input neurons and one output neuron was shown to be the most successful among the tested models for forecasting short-term wind speeds. This model demonstrated superior performance, achieving mean squared error (MSE) and mean absolute error (MAE) values of 0.0016 and 0.0399, respectively [27]. An ANN model to forecast short-term wind speeds across eleven different sites in western Himalaya is employed by [28]. The resulting performance metrics showed a mean absolute percentage error (MAPE) of 4.55% and a correlation coefficient of 98%.

Many hybrid approaches have been developed for short-term wind power forecasting. The hourly wind speed data in Halifax is analyzed using LSTM and ARIMA models to determine the most effective predictive model for the short term. Based on the results, it is concluded that the LSTM model performs better with a root mean square error (RMSE) value of 3.124 compared to the ARIMA model, which has an RMSE

value of 3.423 [29]. Using ANN, specifically the backpropagation neural network (BPNN), radial basis function neural network (RBFNN), and adaptive neuro-fuzzy inference system (ANFIS), short-term wind speed prediction was conducted across both temporal and geographical dimensions. Three wind observation stations in Iran were used for this inquiry. The average wind speed distribution error for the best model from each group was about 2.6%, which is an encouraging level of performance [30].

For the purpose of time series forecasting, ARIMA models developed by Box and Jenkins have been frequently employed. Two combinations of hybrid models, specifically ARIMA-ANN and ARIMA-SVM, were employed for comparison alongside individual ARIMA, ANN, and SVM forecasting models. The study focused on short-term wind speed prediction and utilized a two-year hourly dataset obtained from a wind observation location in Colorado, USA. The outcomes of the simulations indicate that among these, the 1-step ARIMA-ANN hybrid model yielded the most favorable results, with a MAE of 55.354269 [31]. The technique of univariate Box-Jenkins time-series analysis of ARIMA is proposed by [32] to model and forecast upcoming energy production and consumption in Asturias, Spain. The authors adhered to a univariate methodology, and the outcomes demonstrated that the suggested ARIMA model yielded a low approximation error compared to exponential smoothing and regression models with a MAPE of 0,52.

The Genetic Algorithm-LSTM (GLSTM) methodology is introduced and focuses on short-term wind power prediction to optimize LSTM parameters for the European dataset from seven wind farms [33]. The simulation results indicate that the proposed approach outweighs other alternatives, achieving a significantly lower MAE value of 0.07271. Specifically, compared to existing methods, GLSTM has shown improvements ranging from 6% to as much as 30% in the accuracy of wind power predictions. A methodology is introduced that incorporates linear regression models as the baseline alongside advanced techniques such as RNN, LSTM networks, and dynamic neural networks (DNN), specifically the Nonlinear Autoregressive Exogenous (NARX) network [34]. The hybrid approach is utilized for the purpose of conducting short-term wind speed forecasting in the Ecuadorian Andes. According to

the simulation results, the multivariable LSTM network produced superior forecasting performance outcomes than the NARX network by at least 10%.

A novel approach is presented by [35] for short-term forecasting called RMR-HFS-LSTM, which combines a deep learning technique, LSTM, with Relief Mutual Information Recursive feature elimination Hybrid Feature Selection (RMR-HFS). This methodology is applied to predict short-term outcomes. The data used for this study consists of hourly recordings of electricity load and weather conditions from Switzerland, a European country. The proposed RMR-HFS-LSTM model demonstrated superior performance compared to earlier models, particularly when compared to the MLP and RNN, achieving a MAPE result of 7.828. The nonlinear combination forecasting approach, the deep belief networks based on particle swarm optimization (PSO-DBN) model, is introduced by [36] for short-term forecasting based on data taken from a wind farm in Shaanxi, Dingbian, China. The method incorporates Variational Mode Decomposition (VMD) data preprocessing, PSO-DBN, and LSTM deep learning prediction models. The simulation results indicate that the nonlinear combination strategy based on PSO-DBN outperforms the linear combination strategy, achieving a MAE of 35.37.

The Gated Recurrent Unit (GRU), which serves as a streamlined variant of the LSTM architecture, functions as a gating mechanism within RNN [37]. It effectively addresses the challenge of the vanishing gradient problem encountered in models with extended temporal dependencies, similar to the LSTM model.

A hybrid deep learning technique is developed for ultra-short-term wind power generation at the Boco Rock Wind Farm, located in Australia. The hybrid approach involves incorporating GRU, LSTM, and a fully connected neural network, referred to as LSTM-GRU-ANN. When the efficacy of the suggested model is compared to more complex models, it is discovered that the forecasting model exhibits 38.77% greater MAPE accuracy than an ANN [38]. A novel approach is introduced for short-term forecasting in Xinjiang, China, by combining feature-weighted principal component analysis (WPCA) with an improved GRU neural network model, and the

hyperparameters are optimized using a particle swarm optimization (PSO) algorithm by [39]. The proposed WPCA-PSO-GRU model achieves an impressive minimum MAE result of 1.43 compared to a single-model alternative for this dataset. A hybrid deep learning approach with the aim of enhancing the precision of very short-term wind power forecasting at the Bodangora wind farm, Australia is proposed by [40]. This hybrid model encompasses convolutional layers, GRU layers, and a fully connected neural network. In both data sets, it has been found that the hybrid deep learning model performs better than previous forecasting models, numerically increasing the accuracy of wind power forecasts for the Bodangora wind farm by up to 8.13 percent in MAPE. A new hybrid deep learning approach is employed for short-term wind power forecasting in a wind farm located in Jiang County, Shanxi, China [41]. The hybrid model combines VMD, CNN, and GRU algorithms to efficiently determine time-varying features that are hidden. Compared to other deep learning models, the VMD-CNN-GRU model proves to be the most accurate for short-term forecasting, achieving an impressive MAE value of 0.8161.

A novel hybrid approach, known as Bayesian model averaging and ensemble learning (BMA-EL), is introduced by [42] for short-term wind power forecasting by using data from a wind farm situated in China. Through a comparison with simulation results and existing literature approaches, the proposed BMA-EL method achieves a MAPE result of 6.0140, demonstrating accurate and reliable forecasting of wind power outputs. Another hybrid model for short-term WPF applies real historical wind power data gathered from the Pennsylvania-New Jersey-Maryland Interconnection (PJM) to an ARFTCAN (Adaptive Recurrent Fuzzy Time-Series CANFIS) model. The suggested model significantly outperforms traditional SSA-based deep learning models in a low power output scenario, reducing MAPE by more than 13% [43]. A hybrid model that combines an AR model with Gaussian process regression (GPR) is presented for the purpose of stochastic short-term wind speed forecasting [44]. The study employs data from three wind farms located in China. The persistence model, ANN, and SVM were employed by the authors as the foundational techniques for forecasting wind speed at a certain site. The outcomes indicate that the proposed approach outperformed the ANN model with a RMSE value of 1.05, highlighting its enhanced accuracy.

A novel method is introduced that integrates the RNN model with Multivariate Time Series (MTS) information to enhance the accuracy of short-term load forecasting by using data from a wind farm located in Poland. With a MAPE value for forecasting short-term loads of 0.66, the results demonstrate that the proposed model outperforms previous forecasting methods [45]. The performance of hybrid SVR models integrated with a wavelet-based decomposition method is assessed for short-term wind power forecasting. The evaluation is conducted using data from two wind farm locations, one in the USA and the other in India. Among the various regression models studied, the hybrid approach using Time Series SVR (TSVR) demonstrates superior short-term forecasting capabilities, yielding a RMSE value of 0.0100 [46]. The Fuzzy C-means (FCM) Clustering Algorithm is employed for day-ahead wind power forecasting, analyzing historical data from two distinct wind farms in the north-eastern region of China. The results show that the power curve based on this approach outperforms other available alternatives for day-ahead wind power prediction, yielding an RMSE value of 4.12 [47]. A new hybrid modeling approach designed for short-term wind speed forecasting is introduced by using the Kalman filter approach. The method involves the fusion of a non-linear vector support regression-based phase-space model with an unscented Kalman filter-based dynamic state prediction. The study makes use of information acquired from three different Massachusetts-based sites. The simulation results demonstrated that the 1-step AR-Kalman technique produced better outcomes in terms of RMSE values, in particular obtaining an RMSE of 0.2849 [48]. Another hybrid model was created by integrating Recurrent Kalman Filter (RKF), Neuro-Wavelet (WNN), and Adaptive Neuro-Fuzzy Inference System (ANFIS) in [49]. The outcomes of the study demonstrate that while all the suggested hybrid models exhibit strong performance, the sequence of ANFIS followed by RKF and then WNN (ANFIS + RKF + WNN) emerges as the most proficient among them with a 2.0343 MAPE result.

Various models for successful short-term wind speed forecasting using correlation analysis (CA) and improved complete ensemble empirical mode decomposition with additive noise (ICEEMDAN), the Harris Hawks optimization algorithm (HHO), and Seq2Seq based spatial and temporal attention (STAt-S2S) are established by [50]. The

study region examined is a site located in Gwadar, Pakistan. In comparison to existing independent and hybrid forecasting models, the CA-ICEEMDAN-HHO-STAt-S2S and CA-ICEEMDAN-STAt-S2S models show improved prediction results. The CA-ICEEMDAN-STAt-S2S data shows RMSE, MAE, and MAPE values of 0.639 m/s, 0.474 m/s, and 15.710 m/s with the Nash–Sutcliffe efficiency (NSE) of 0.922. The research was conducted at the Sotavento wind farm, located in Spain and introduced a wind power output forecasting model using a combination of a deep learning network and a multivariate time series clustering algorithm [51]. The comparison of predicted outcomes with alternative models highlights the superior performance and proficient capability of Seq2Seq models in extracting deep features. This underscores the potential of intricate deep learning techniques for enhancing wind power forecasting.

The hybrid model, which merges dynamic and fuzzy time series methodologies for mid-term forecasting, is evaluated using real load data from the Seoul metropolitan area [52]. Compared to the Koyck and ARIMA models, the proposed hybrid model yields superior forecasts. It achieves a MAPE of less than 3% for the total load forecast. A short- to mid-term univariate model is used by [53] to employ Seq2Seq learning method to forecast the amount of power generated by wind farms in Manisa, Turkey. For hidden states and cell states, the study used LSTM layers with 150 dimensions, and for the Seq2Seq model, it used LSTM and GRU layers with 150 internal dimensions. The LSTM-based Seq2Seq model with 100 lags beat competing models, reaching an exceptionally low MSE value of 0.0216 for forecasting the average electricity generation, according to the performance evaluation. A method is suggested for forecasting wind power for a wind turbine in Poland within short and medium timeframes by [54]. This approach utilizes a data-centric machine learning technique, specifically focusing on three distinct implementations of gradient boosting (GB) regressors. The model was trained using three different GB implementations to predict the output power of wind turbines. These implementations include CatBoost, LightBoost, and XGBoost. Out of the various models, the CatBoost model stands out as it exhibits the most impressive performance. This is evident from its lowest RMSE of 76.18 and MAE value of 54.87.

As wind patterns are inherently random, it is essential to depict the wind resource by utilizing long-term averages that outline the potential power at the intended location [55]. A novel CGRU (Convolutional Gated Recurrent Unit) multistep wind speed forecasting model is presented for long-term forecasting using historical wind speed data from a wind farm in Shandong Province, China. The model is based on secondary decomposition (SD) and the multi-label-specific XGBoost feature selection method. The results show that the 1-step prediction curves of the SD-MLXGBoost-GA-CGRU model have the closest fit to the actual wind speed curve, achieving a MAE value of 0.2423 [56]. In order to forecast the long-term (10-year) wind resource, linear measure correlate predict (MCP) algorithms were applied at the same locations and surrounding meteorological stations. MCP methods are being used at 37 sites around the United Kingdom. The long-term forecasted mean wind speed and power density using the linear regression MCP method over a 12-month period had average percentage errors of 3.0% and 7.6%, respectively [57]. Machine learning models for time series analysis were formulated to predict the long-term power output of the Adama II wind farm in Ethiopia [58]. The research involved the evaluation of six different supervised learning algorithms. The outcomes of the study reveal that a hybrid model combining Prophet and XGBoost demonstrates strong fitting performance and offers precise insights into power generation, achieving a MAPE value of 6.9%.

A set of online learning algorithms designed for training local RNN is proposed by [59]. These algorithms encompass the global recurrent prediction error (RPE) and three local variations. The purpose of these algorithms is to facilitate long-term power data forecasting, utilizing data obtained from Eastern Crete, Greece. The experimental findings indicate that the recurrent forecasting models deliver superior multi-step predictions in comparison to the persistent method, atmospheric models, and time-series models. Cascade-forward neural networks, support vector machines, and random forest algorithms were developed to forecast long-term wind energy production on Palestinian farms. Among these approaches, the cascade-forward neural network algorithm outperformed the others, achieving a RMSE value of 41.1659 kWh, indicating its superior predictive accuracy [60]. A forecasting approach that merges a quaternion convolutional neural network with a bi-directional long-short-term

memory recurrent network was developed by [61]. The designed forecasting model is built using data gathered from Lesvos and Samothraki Greek islands in the North Aegean Sea, and the forecasting scope extends to one day ahead, representing a long-term prediction. Compared to the bi-directional LSTM (BLSTM) model, the suggested model's accuracy increased significantly in these case studies by 13% and 20%, respectively. A Multiplicative Compositional Policies (MCP) approach, built on the modeling of the underlying probability distribution using the Box-Cox transformation, was applied for long-term forecasting [62]. This was carried out using data collected from 22 pairs of sites in the UK. The Box-Cox approach demonstrated better performance compared to regression methods across all measurement periods, achieving a MAE value of 0.11.

Table 2.1 Literature Review Table

Reference	**Model**	**Data Source**	**Term**	**Performance Metric**
[26]	LSTM	Turkey	Short-term	R^2: 0.9366 MAE: 0.022 MSE: 0.0012
[27]	ANN	Mexico	Short-term	MSE: 0.0016 MAE: 0.0399
[28]	ANN	Himalaya	Short-term	MAPE: 4.55%
[29]	ARIMA LSTM	Halifax	Short- term	RMSE MAE ARIMA: 3.423 2.772 LSTM: 3.124 2.457
[31]	ARIMA-ANN	USA	Short-term	MAE: 55.35

Table 2.1 (continued)

[32]	Box-Jenkins Time Series - ARIMA	Austria	Short-term	MAPE: 0.52
[33]	Hybrid approach GLSTM	Europe	Short- term	MSE: 0.00924 MAE: 0.07271 RMSE: 0.09615
[34]	LSTM vs NARX	Ecuadorian Andes	Short-term	MSE R^2 LSTM: 0.09 0.99 NARX: 0.33 0.98
[35]	RMR-HFS-LSTM	Switzerland	Short-term	MAPE: 7.828
[38]	LSTM-GRU-NN	Australia	Very short-term	MAPE: 38.77%
[41]	VMD-CNN-GRU	China	Short-term	RMSE: 1.5651 MAE: 0.8161 MAPE: 11.62%
[42]	Hybrid model based on BMA-EL	China	Short- term	MAPE: 6.0140
[43]	Hybrid model based on SSA-ARFTCAN	USA	Short- term	RMSE: 0.0053
[45]	MTS based RNN	Poland	Short-term	MAPE: 0.66
[46]	Time Series SVR	USA	Short-term	RMSE: 0.0100
[48]	AR-Kalman	USA	Short-term	RMSE: 0.2849

Table 2.1 (continued)

[50]	CA-ICEEMDAN–HHO–STAt-S2S	Pakistan	Short- term	RMSE: 0.639 MAE: 0.474 MAPE: 15.710
[52]	Hybrid dynamic and fuzzy time	South Korea	Mid-term	MAPE: 2.6%
[53]	GRU LSTM	Turkey	Short to midterm	MSE MAE LSTM: 0.0216 0.1115 GRU: 0.0217 0.1126
[54]	CatBoost	Poland	Short and medium term	RMSE: 76.18 MAE: 54.87
[56]	SD-MLXGboost-GA-CGRU	China	Long- term	RMSE: 0.3192 MAE: 0.2423
[57]	Linear MCP algorithms	UK	Long- term	MAE: 0.15
[58]	Prophet-XGBoost	Ethiopia	Long-term	MAPE: 6.9%
[62]	Box- Cox (MCP)	UK	Long-term	MAE: 0.11

Table 2.1 presents a summary of wind speed forecasting, categorizing articles based on the forecasting term. Notably, a preference for short-term forecasting methods emerges among researchers, driven by the challenges posed by unpredictable factors like nature, economy, seasonality, or insufficient historical data. This preference for short-term approaches is further underscored due to the demand for increased reliability in long-term forecasting. Various deep learning, machine learning, and

hybrid approaches have been developed to enhance the accuracy and dependability of forecasts. The findings indicate that hybrid methods exhibit superior performance in terms of error metrics. These hybrid models have demonstrated improved performance compared to pure statistical or machine learning models in various applications. Hybrid methods integrate well-established quantitative time-series models with deep learning techniques, employing deep neural networks to generate model parameters at each time step. It's essential to highlight that certain articles feature multiple test sets and forecast results for numerous scenarios. Therefore, Table 2.1 only highlights the performance metrics of the best-case scenarios.

CHAPTER 3

METHODOLOGY

In this chapter of the thesis, the methodology used for wind power forecasting is covered in detail. The methodology employs time series methodology, machine learning, and deep learning approaches, particularly GRU, LSTM, and Seq2Seq models, for wind power forecasting. The activation functions used in models and performance metrics are explained. Data collection and model training are outlined. Hybrid methods are proposed during the chapter since they improve forecast accuracy by reducing the biases inherent in individual models.

3.1. Time Series

One significant area of forecasting is time series forecasting, which builds a model describing the underlying relationship by gathering and analyzing historical observations of the same variable. Time series forecasting is a comprehensive quantitative method that involves gathering and examining past data in order to create a suitable model. Characteristic evaluation, modeling, forecasting are the three main processes involved in the theoretical analysis of time series [34]. Whereas forecasting determines the system's short-term evolution, modeling determines the system's long-term behavioral characteristics. Characteristic evaluation includes identifying basic features such as the degrees of freedom or the randomness measurement [63]. Using this analysis, predictive models with the fewest errors possible can be developed. Time series modeling is especially advantageous when there is a lack of adequate explanatory models that connect the forecasting variable to other explanatory variables or when insufficient information is known about the underlying process for producing data.

The Box-Jenkins Model, a mathematical framework, is utilized to forecast data ranges based on inputs from time series data. It's versatile in analyzing various types of time series data for predictive purposes.

There are some assumptions of error term for time series. The error term is assumed to have equal variance (homoscedastic), there is no connection between its variability and the dependent variable for time series. If there's a relationship between the dependent variable and the variance of the error term, the data is deemed heteroskedastic.

Time series can be classified as stationary and non-stationary, where non-stationary data are unstable and, due to their complexity, cannot be modeled to generate an accurate forecast [64]. In stationary data, the long-term mean and long-term variance are constant and independent of time, while the mean and variance of nonstationary data are variable and lack long-term stability. In order to obtain consistent and reliable results with accurate modeling, non-stationary data needs to be converted to stationary data. Univariate and multivariate time series are another type of time series classification. Only one variable is assessed in a univariate series over a predetermined length of time. Multiple variables that change over time are taken into consideration in multivariate time series. For the general approach to modeling time series, data should be examined based on their unique characteristics. If the data is non-stationary, stationary residuals should be obtained by removing seasonal and trend factors. A few sample statistics should be employed to determine a suitable model to fit the residuals. Eventually, to forecast the actual series, the residuals have to be predicted and the stationary transformation should reverse.

3.1.1. Autoregressive Integrated Moving Average

ARIMA, is among the most fundamental and preferred time series models. The latencies of the stationarized series are referred to as AR factors in the forecasting equation, whereas the delays of the forecast errors are termed MA. Additionally, a time series that requires differencing to achieve stationary is described as an "integrated" version of a stationary series. The ARIMA model's widespread use can be attributed

to both its statistical characteristics and the widely recognized Box-Jenkins methodology [65] employed in constructing the model. Considering their outstanding performance and efficacy, the various types of ARIMA models are widely employed when modeling the forecast of linear and stationary time series [66]. ARIMA models are particularly flexible since they can support a wide range of time series. However, their primary weakness is the model's assumed linear form. The ARIMA model lacks the ability to detect any nonlinear features since the time series values are considered to have a linear correlation pattern. The ARIMA model can be shown as

$$Y_t = \varphi_0 + \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + \cdots + \varphi_p Y_{t-p} + \varepsilon_1 - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \quad (1)$$

Where Y_t is the actual value, φ_i and θ_j are the coefficients and ε_t is the random error. "p" is an integer that indicates autoregressivity and denotes the quantity of lag variables that will be employed as predictors whereas, "q" refers to the moving average and the number of forecast errors that are lag-related in the ARIMA model.

Using linear models to solve complicated real-life problems occasionally fails to yield successful results. Therefore, artificial neural networks which have the ability to model nonlinearly in a flexible way have been proposed as a time series forecasting alternative in recent times.

3.2. Artificial Neural Networks

Neural networks, also known as ANN, are a type of machine learning and are essential for deep learning techniques. Drawing inspiration from the human brain, both in name and structure, they replicate the communication pattern of biological neurons. ANN has found extensive application in modeling time series across diverse fields [67]. Given the intricacy of time series observations and encounters, it is suggested that ANN be used, as it has demonstrated great performance in modeling complex nonlinear relationships without requiring prior knowledge about the nature of the relationships [68]. For example, ANN was able to effectively model and forecast chaotic time series by controlling nonlinearity, demonstrating forecasting skills that

are superior to those of conventional techniques [69]. Typically, forecasting the future state of a noncovariate time series relies on awareness of current measurements and potentially recent historical data.

ANN consists of numerous nodes functioning in parallel and communicating through interconnected synapses [70]. ANN can effectively represent complex nonlinear relationships without the need for pre-established assumptions about the nature of these relationships. ANN is constructed with node layers, and neurons are organized into layers, with each layer's neurons transmitting information to the subsequent layer while also receiving input from that same layer. Networks of this type do not permit connections to neurons in the same or preceding layers. The intermediary layers between the input and output layers are termed hidden layers, and the final set of neurons forms the output layer. The input layer is composed of special input neurons, through which only externally applied input is conveyed via their outputs. The basic illustration of the ANN structure can be seen in the Figure 3.1.

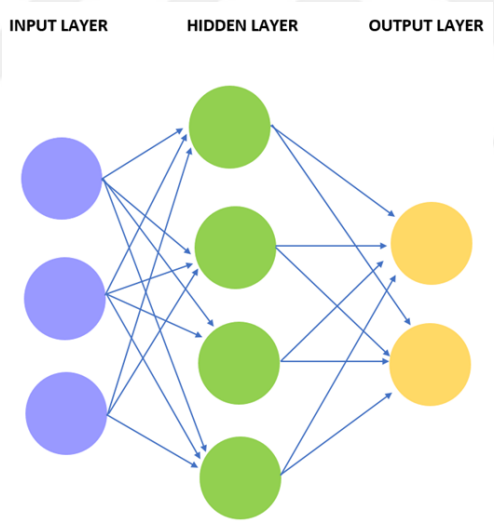


Figure 3.1 ANN Structure

There are weights and thresholds that associate each node in the network. Whenever a node's output exceeds the specified threshold activation starts by leading data transmission to the next layer. However, data cannot proceed to the network's next layer if the output is less than the specified threshold.

Each node is considered a separate linear regression model consisting of input, output, weights, and a bias. The formula, therefore, can be written as;

$$Y_t = \sum_{i=1}^{n} x_i \omega_i + bias \tag{2}$$

$$Y_t = \begin{cases} 1 \; if \sum_{i=1}^{n} x_i \omega_i + bias \geq 0 & (3) \\ 0 \; if \sum_{i=1}^{n} x_i \omega_i + bias < 0 & (4) \end{cases}$$

After the identification of the input layer, weights are assigned. Larger weights have considerably more effects on the output than smaller ones, assisting in determining the relative importance of each variable. Subsequently, every input is multiplied by its corresponding weight and added together. An activation function is then applied to the output, thereby determining the output. To activate a node, its output should surpass a predetermined threshold, forwarding data to the network's subsequent layer [71]. Consequently, the output of one node becomes the input of the subsequent node. During this neural network architecture, data is transferred from one layer to the next; therefore, it is referred to as a feed-forward network.

3.2.1. Backpropagation

The majority of deep neural networks operate in a feedforward manner, moving unidirectionally from input to output. Nevertheless, backpropagation, which requires moving from output to input in a reverse direction, can be employed for training the model. The most popular and effective learning algorithm among the several neural network topologies that have been proposed is back-propagation. It serves as a crucial mathematical instrument for enhancing prediction accuracy in data mining and machine learning. In essence, backpropagation is an algorithm employed to swiftly compute derivatives in a neural network, capturing the alterations in output resulting from tuning and adjustments [72]. ANN and deep neural network utilize backpropagation as a learning algorithm to calculate a gradient descent, a form of

optimization algorithm that directs the user towards the maximum or minimum of a function [73]. The gradient descent method aids the system in minimizing the difference between expected and actual system outputs in the context of machine learning. By modifying the weight values for different inputs to reduce the disparity between outputs, the algorithm fine-tunes the system. It is also referred to as the error between the two. More precisely, a gradient descent algorithm employs a step-by-step procedure to give information about the parameters of a network that must be changed in order to lessen the difference between the intended and actual outputs. This procedure is directed by a cost function, which is an assessment metric. This error is quantified mathematically by the cost function. Finding the necessary parameter adjustments to lower the cost function and raise overall accuracy is the aim of the algorithm [74].

One advantage of backpropagation is its ability to hold patterns that are larger than their fundamental vector dimension. It has a multilayered, feedforward architecture and uses supervised learning. In this method, the network is trained to associate specific responses with each input it receives. The weights in the network are determined by comparing the actual response with the expected response, aiming to minimize the sum-squared error (SSE), which is calculated by

$$SSE = \sum_{i=0}^{n} (y_i - \hat{f}_i)^2 \tag{5}$$

Where y_i represents the actual value and $\hat{f}_i$ represents the forecasted value.

3.2.2. Activation Functions

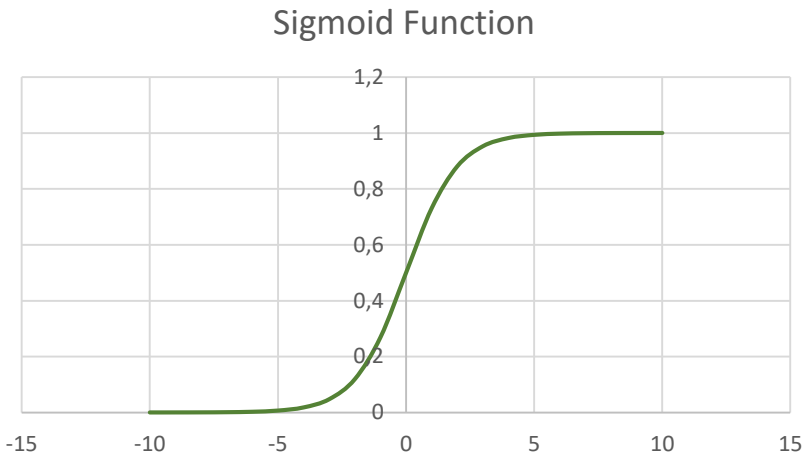
Activation functions serve a vital role in deep neural network architectures, determining whether information should be passed on to the next neuron [75]. The evolution of activation functions in deep learning algorithms considers aspects like facilitating a healthy learning process, preventing overfitting, enhancing accuracy and performance, and reducing computational costs. However, without careful selection of activation functions, deep learning models may not achieve the desired success and

might end up behaving akin to linear regression, limiting their learning capacity. Hence, non-linear activation functions are favored in deep neural network architectures to introduce the necessary complexity. These non-linear activation functions can be categorized into fixed and trainable types [76]. Deep learning algorithms employ the backpropagation algorithm to update parameters, culminating in the return of derivative values for the functions. Consequently, the activation functions integrated into deep learning architectures must possess the capability to receive derivatives continuously. The adoption of specific activation functions has garnered user attention, contributing to the growing interest in deep learning architectures.

In the literature, the training of deep neural networks began incorporating activation functions, leading to successful outcomes [77]. Activation functions within deep learning architectures are anticipated to possess key features, including being derivative, non-linear, and capable of reaching the global optimum without getting trapped in local optima. Non-linear fixed parameter activation functions play an active role in deep learning architectures without requiring external parameters. The fixed parameter activation functions highlighted in the literature include sigmoid, hyperbolic tangent, Rectified Linear Unit (ReLU), and the Gaussian Error Linear Unit (GELU) [78].

Sigmoid Activation Function:

In deep learning architectures, the preference for the backpropagation algorithm is evident as it facilitates the learning process between neurons. The sigmoid activation algorithm updates parameters by computing derivatives within the architectures. Consequently, activation functions must be differentiable to enable this updating process. Linear activation functions, yielding a constant value during derivative calculation, impede the learning process in deep architectures when assisted by the backpropagation algorithm [79]. To address this issue, a proposed solution involves the use of a non-linear sigmoid activation function, as shown in the following figure.



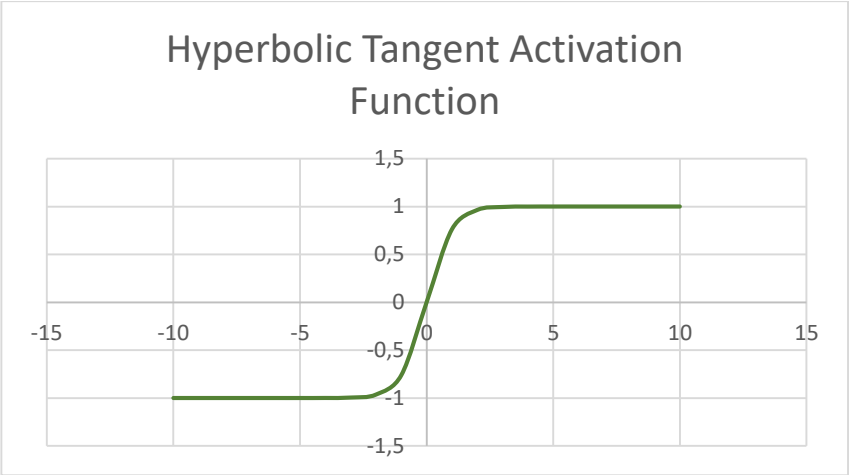
$$f(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

Figure 3.2 Sigmoid Activation Function Graphic

The function, which involves an input value represented by x and the mathematical constant e, transforms any input to a value ranging from 0 to 1. Its benefit for tasks involving values included in this range, such as logistic regression and binary classification. Notably, the function's domain spans from negative to positive infinity, while its range is limited to (0,1). A notable feature of the sigmoid function is its distinctive "S" shape. With rising input values, the output initially rises slowly, followed by a rapid acceleration towards 1, and eventually stays at the level of 1.

Hyperbolic Tangent Activation Function:

The hyperbolic tangent activation function, a non-linear activation function, is capable of performing the derivative operation. Structurally, it resembles the sigmoid activation function. The graph of the function at its formula can be seen in the figure below.



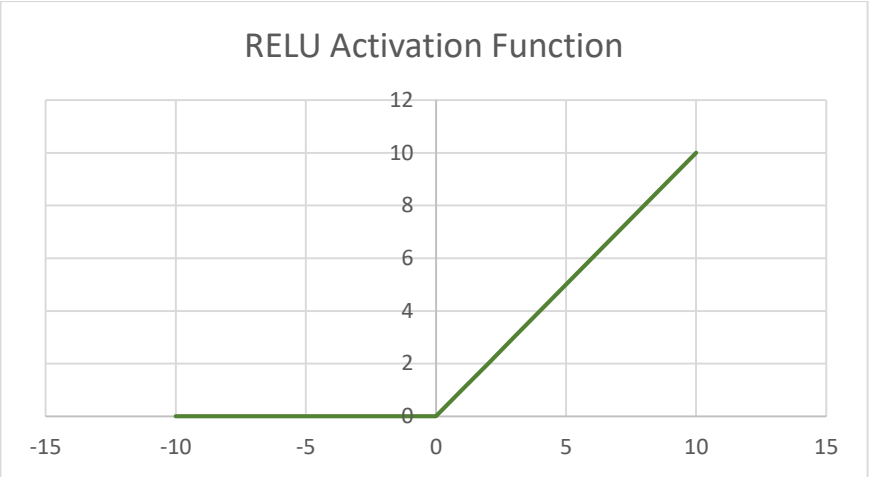
$$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{7}$$

Figure 3.3 Hyperbolic Tangent Activation Function Graphic

Examining the figure, the hyperbolic tangent activation function exhibits a sigmoidal curve and an S-shaped curve. It can be computed basically with a tangent function. Unlike the sigmoid activation function, which generates output between [0,1], the hyperbolic tangent activation function produces values between [-1,1]. The learning process is observed to extend to negative values due to the wider value range of the hyperbolic tangent activation function. However, it encounters the vanishing gradient problem as the derivative values approach zero after reaching values in the range of [-1, 1] [79].

Rectified Linear Unit (ReLU) Activation Function:

The vanishing gradient problem, prevalent in sigmoid and hyperbolic tangent activation functions due to their inability to be differentiated beyond a certain threshold, led to the development of the ReLU activation function as a solution. ReLU is a non-linear activation function capable of performing the derivative operation. The graph of the function is shown in Figure 3.4.



$$f(x) = \max(0, x) \tag{8}$$

Figure 3.4 ReLU Activation Function Graphic

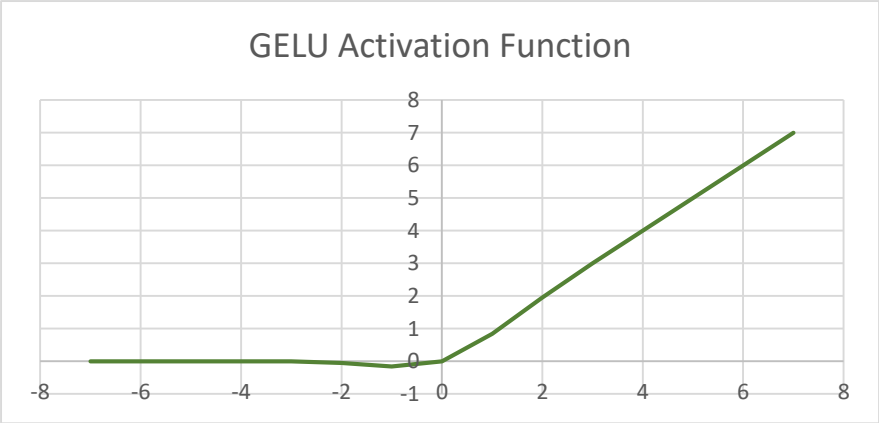
where x indicates the input value and the output value of the ReLU activation function is denoted by $f(x)$. The function yields the input x if x is equal to or greater than 0; otherwise, it outputs 0. Mathematically, it is expressed as selecting the larger value between 0 and the input x [80].

ReLU has gained widespread popularity for its effectiveness in overcoming the vanishing gradient problem. However, it does introduce a challenge known as the "dead neurons" problem in its negative region, where negative values are set to zero, impeding the calculation of derivatives and slowing down the learning process [81]. Despite this drawback, the ReLU activation function has a number of important benefits, one of which is that it requires less computing power than other functions, which makes it a good choice for multi-layered designs.

The Gaussian Error Linear Unit (GELU) Activation Function:

The rectifier function can be approximated smoothly and differently with the GELU activation function. GELU activation has become an increasingly popular option for many deep learning applications quite quickly since its desirable properties, which

include smoothness, differentiability, and approximation of the popular ReLU function. The adaptability and effectiveness of the GELU activation function have been demonstrated by its successful integration into a number of neural network architectures [82]. The GELU activation function that uses $\Phi(x)$, which represents the cumulative distribution function of the standard normal distribution, frequently denoted as $N(0,1)$ can be defined as



$$f(x) = x\Phi(x) \tag{9}$$

Figure 3.5 GELU Activation Function Graphic

Additionally, the approximate function can be stated as follows:

$$f(x) = \frac{1}{2}x\left(1 + \tanh\left(\sqrt{\frac{2}{\pi}}(x + 0{,}044715^3)\right)\right) \tag{10}$$

This formulation gives an approximation to the GELU function through the utilization of the hyperbolic tangent (tanh) function, which is an employed method for incorporating the GELU function within computational frameworks.

The objective of developing the GELU activation function was to provide a smooth and differentiable substitute for the popular ReLU activation function while

maintaining the latter's inherent advantages. The ReLU function introduces nonlinearities into the network, which can lead to challenges throughout gradient-based optimization, including dead neurons or unpredictable training characteristics. To address these problems, the GELU activation function is designed as an approximation to the ReLU function, maintaining differentiability at all points while retaining the necessary nonlinear characteristics for deep learning architectures. The Gaussian cumulative distribution function (GDF), which is distinguished by its fundamental smoothness and differentiability properties, serves as the model for the GELU function [83].

3.3. Recurrent Neural Network

RNN is a category of artificial neural network designed for processing sequential or time series data. RNN is commonly applied to problems involving order or temporal relationships. Similar to feedforward and CNN, RNN receives information from data collected during training but stand out due to their "memory" feature. This aspect of memory involves integrating data from previous inputs to impact both the current input and output. RNN outputs are impacted by previous parts in the sequence, unlike traditional deep neural networks, which presume input-output independence. RNN provides recurrent connections so that data can be transmitted from one step to the next continuously. However, it's worth noting that unidirectional recurrent neural networks are limited in predicting future events that could influence the output of a given sequence [84]. The RNN structure is demonstrated in the figure.

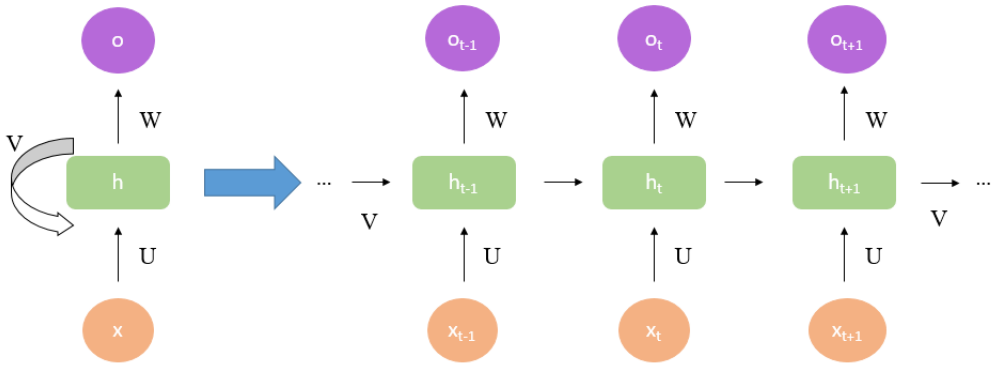


Figure 3.6 RNN Architecture

Another unique feature of recurrent networks is their parameter sharing across each layer. RNN employs a single weight parameter for all nodes in a layer, in contrast to feedforward networks, which provide different weights to each node. To support the network's reinforcement learning, these common weights are nevertheless capable of modification through the gradient descent and backpropagation processes.

RNN employs the backpropagation through time (BPTT) algorithm for gradient computation, specifically designed for sequential data. While BPTT shares similarities with traditional backpropagation in terms of error calculation and parameter adjustment from output to input layers for training, it differs in that it sums errors at each time step. This is unlike feedforward networks, which lack shared parameters across layers and thus do not require error summation [85].

RNN faces two common challenges during operation: exploding gradients and vanishing gradients. These issues are distinguished by the magnitude of the gradient, which indicates the slope of the loss function along the error curve. Vanishing gradients occur when the gradient is excessively small, leading to weight parameter changes that are not sufficient to promote significant learning, hence stopping the training process. Secondly, very large gradients cause exploding gradients, which create an unstable model with too many weights. The number of hidden layers in the neural network should be reduced to eliminate the complexity caused by these problems [86].

Feedforward networks take a single input and produce a corresponding output. Although recurrent neural networks are often depicted similarly, they don't adhere to this limitation. Unlike feedforward networks, RNN can handle inputs and outputs of varying lengths. Various types of RNN are applied in diverse scenarios.

There are four frequently used variations of RNN which are one-to-one configuration, one-to-many, many-to-one and many-to-many, as shown in the figure.

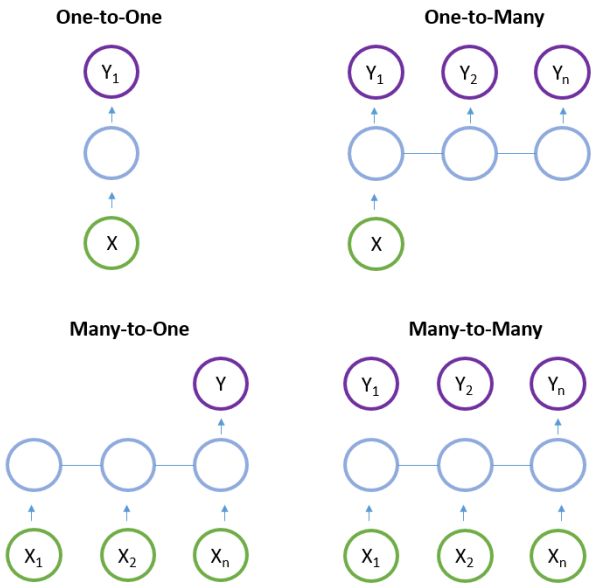


Figure 3.7 RNN Configurations

The most basic form of RNN, the One-to-One configuration, processes input to produce a singular output. With fixed input and output dimensions, it functions akin to a conventional neural network. One-to-Many is a category of RNN that yields multiple outputs when provided with a solitary input. Operating with a consistent input size, it generates a sequence of data outputs. Many-to-One is applied when a solitary output is needed from numerous input units or a sequence thereof. It processes a series of inputs to produce a stable output. Many-to-Many is employed to produce output data based on a sequence of input units. This RNN type is subdivided into the following two subtypes: In equal unit size, both the input and output units share identical quantities. This is frequently applied to tasks such as name-entity recognition. In contrast, unequal unit size configuration involves varying numbers of units for inputs and outputs. It is commonly used in applications like machine translation [87].

3.3.1. Long-Short Term Memory

An LSTM is a subtype of RNN designed to grasp long-term relationships within sequential data. In contrast to traditional RNN, LSTM-based machine learning approaches aim to address and mitigate issues related to long-term dependencies in the data. The training of an LSTM involves backpropagation through time, effectively

overcoming the vanishing gradient problem. Including nonlinear, data-dependent processes in the RNN cell was a key finding in the LSTM design [88]. This makes it possible to train these mechanisms by ensuring that, during training via gradient descent, the gradient of the objective function with respect to the state signal does not vanish. This gradient is a number that is directly related to changes in parameters. Different from traditional artificial neural networks, LSTM networks consist of interconnected memory blocks arranged in layers that are sequential [89]. The fundamental components of an LSTM include an input layer, an LSTM layer, a fully connected layer, and a regression output layer. The Figure 3.8 illustrates the structure of an LSTM cell. The configuration of the network operates by initially processing input through the input layers, followed by the LSTM layer. Sequential or time-series data is introduced into the network architecture through a sequence of input layers.

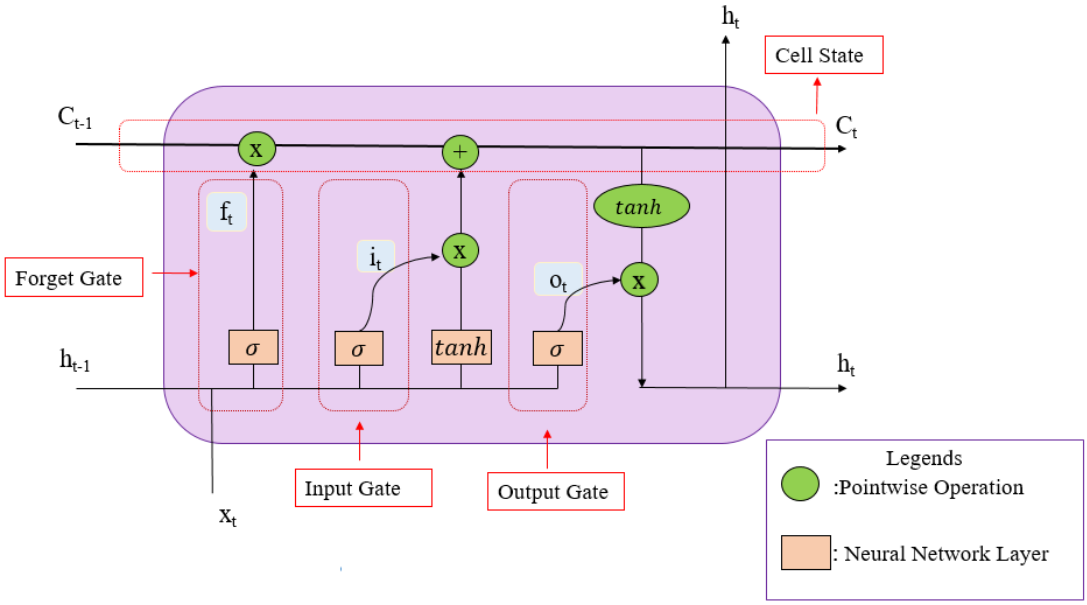


Figure 3.8 LSTM Architecture

The LSTM layer is designed to understand extended relationships among time steps in sequential data [90]. Consequently, there are three gates, which are the input gate, forget gate, and output gate, demonstrated by i_t, f_t and o_t respectively. For these gates formulations are given as

$$i_t = \sigma(V_i h_{t-1} + U_i x_t + b_i), \tag{11}$$

$$f_t = \sigma\big(V_f h_{t-1} + U_f x_t + b_f\big), \tag{12}$$

$$o_t = \sigma(V_o h_{t-1} + U_o x_t + b_o). \tag{13}$$

The rest of the updating equations are

$$d_t = \tanh(V_d h_{t-1} + U_d x_t + b_d), \tag{14}$$

$$c_t = f_t * c_{t-1} + i_t * d_t, \tag{15}$$

$$h_t = o_t * \tanh(c_t), \tag{16}$$

In equations, * demonstrates component- wise multiplication and $\sigma(\alpha) = \frac{1}{1+e^{-a}}$ which is a sigmoid function. The input gate determines which information is to be incorporated into the cell, while the forget gate determines which selected information should be discarded. The output gate is responsible for choosing the information to be passed as input to the cell for the next step.

The initial forget gate gathers information at time *t* based on the input x_t and the previous hidden layer h_{t-1}. If the value within the forget gate is near 1, the previous memory cell c_{t-1} will be preserved; otherwise, it will be deleted. In the latter case, the new information obtained is merged with the old data to form the input gate, ultimately generating a new memory cell c_t [53].

3.3.2. GRU

GRU is a type of RNN architecture and has a structure similar to LSTM. GRU is intended to simulate sequential data, similar to LSTM, by enabling certain information

to be deliberately retained or forgotten over time. GRU and LSTM differ primarily in how they manage the state of the memory cell [91]. The input gate, output gate, and forget gate are the three gates that are used in LSTM to update the memory cell state, which is kept distinct from the hidden state. However, a candidate activation vector, which is updated via the reset and update gates, takes the place of the memory cell state in GRU [92].

GRU mitigates the vanishing gradient problem that is encountered by vanilla recurrent neural networks, which involves the diminishing values used for updating network weights. Therefore, GRU is a well-liked substitute for LSTM in the modeling of sequential data, particularly when limited computational resources or a more straightforward architecture are required. The computation time of a GRU model is significantly faster than that of an LSTM model since the GRU only has two types of control gates [93].

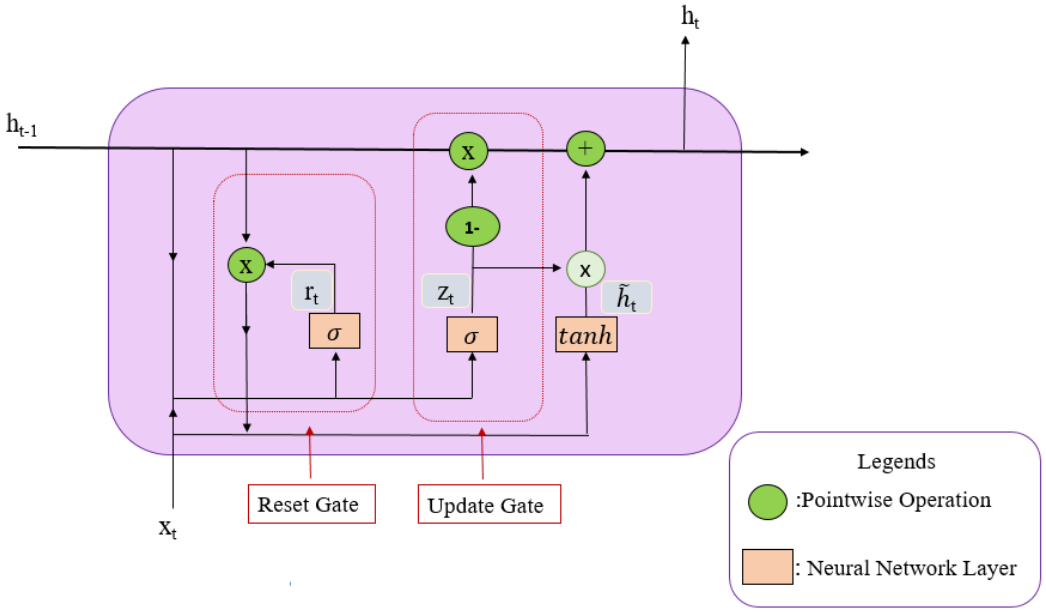


Figure 3.9 GRU Architecture

In addition to the LSTM structure, GRU introduces two additional gates: the update gate and the reset gate. Sequential data, including time series, is transmitted to the GRU by the input layer. The recurrent computation takes place in the hidden layer.

According to the previous hidden state and the current input, the hidden state is updated at each time step. The network's memory of the prior inputs is represented by a vector of numbers called the hidden state. The reset gate plays a crucial role in deciding the extent to which the prior hidden state should be forgotten. The amount to which the previous hidden state is reset at the current time step is determined by this vector. The amount of the candidate activation vector that is incorporated into the new hidden state is decided by the update gate. Similarly, to the reset gate, it generates a vector of numbers between 0 and 1 that controls the extent to which the candidate activation vector is integrated into the next hidden state, given the inputs of the prior hidden state and the present input [94]. A candidate activation vector is calculated by the GRU at each time step by integrating data from the input and preceding hidden state. The hidden state is then updated for the subsequent time step using this candidate vector.

The equations for the new gates are;

$$h_t = (1 - z_t)h_{t-1} + z_t h_t, \tag{17}$$

$$\widetilde{h_t} = \tanh(U_t r_t h_{t-1} * x_t), \tag{18}$$

$$r_t = \sigma(U_r h_{t-1} * x_t), \tag{19}$$

$$z_t = \sigma(U_z h_{t-1} * x_t). \tag{20}$$

Two gates, the reset gate and the update gate, are applied to calculate the candidate activation vector. The amount of information from earlier hidden states that is carried over into the current state is managed by the update gate, as mentioned. Greater information from earlier states is retained when the update gate's value is higher. A similar mechanism governs the amount of information from earlier states that is disregarded; the lower the reset gate's value, the more information is disregarded. As

a result, reset gates are often activated frequently to capture short-term dependencies, whereas update gates are activated in response to long-term dependencies. The network's output is generated by the output layer using the last hidden state as input. The implementation of the reset and update gates in GRU enhances its performance. These gates contribute to time efficiency and improve the model's performance [95].

3.4. CNN

CNN operates as a type of feedforward neural network that uses convolutional structures to obtain features from data. Its purpose is to identify features within datasets and classify complex data sets with high dimensions. Drawing inspiration from visual perception, CNN architecture acts like biological neurons, with artificial neurons corresponding to their biological counterparts [96]. By using this analogy, CNN kernels are multiple receptors that can detect different features, and activation functions are the biological mechanism that transmits neural electric signals to the next neurons when they exceed a threshold. Compared to other extraction techniques, CNN has the advantage of not requiring the elimination of features manually. Loss functions and optimizers serve as tools devised to guide the CNN system in learning our desired outcomes [97]. CNN uses local connections that reduce parameters and simplify integration by connecting neurons with only a small subset of neurons from the previous layer instead of all of them. To reduce parameters, by the weight sharing, the connection with the same weight can be grouped. CNNs down sample images by taking advantage of the local correlation principle, which lowers the amount of data while maintaining important information. Furthermore, this procedure helps to reduce the number of parameters by removing features that are not important [98].

Pooling or downsampling is introduced to mitigate redundancy since convolution feature maps containing numerous features can lead to overfitting issues. In the pooling operation, the information contained in the receptive field is subjected to an aggregation function by the kernel, which fills the output array [100].

The structure of CNN with feature extraction and classification is given in the following figure.

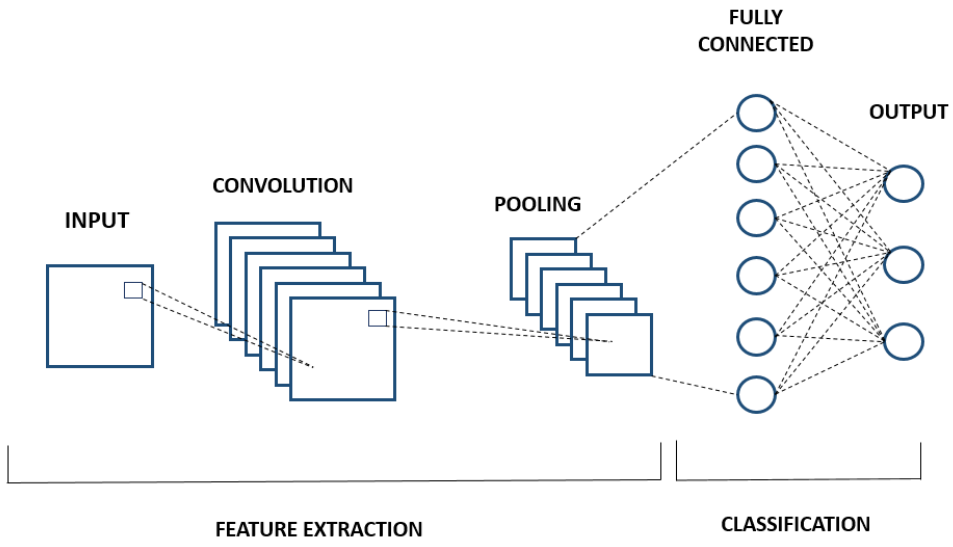


Figure 3.10 CNN Architecture

In order to extract features and create feature maps, convolution is a key component. Data loss at the borders may occur when defining a convolution kernel of a particular size. In order to expand the input with zero values and thus indirectly adjust the size, padding is implemented. The distance, or number of pixels, that the kernel travels across the input matrix is indicated by its stride. A larger stride yields a smaller output. Furthermore, the stride is employed to control the convolutional density. In the case of filters that differ from the input image, zero-padding is commonly used to create an output that is equal to or larger in size by setting elements outside the input matrix to zero. Optimizing these parameters prior to training ensures that the CNN performs at its best [99].

3.5.SEQ2SEQ

Seq2Seq models have gained increasing popularity in the field of deep learning, particularly for handling sequential or temporal data. Seq2Seq is a type of RNN architecture, and its initial applications were predominantly in the field of natural language processing, including activities such as machine translation and text production. These are particular types of models that simply receive a sequence as input and produce a sequence as output. However, the underlying architecture of these models can be diverse, encompassing recurrent structures to convolutional designs and

even incorporating transformers, specifically encoder- decoder transformers or hybrid combinations of these components. Therefore, it is a general framework or methodology for processing temporal data [53]. The encoder-decoder architecture is the architecture that is most frequently employed for developing Seq2Seq models. Seq2Seq models comprise an encoder responsible for condensing the input sequence into a hidden context vector and a decoder that comprehends this context vector to generate an appropriate output sequence in response, as shown in the figure.

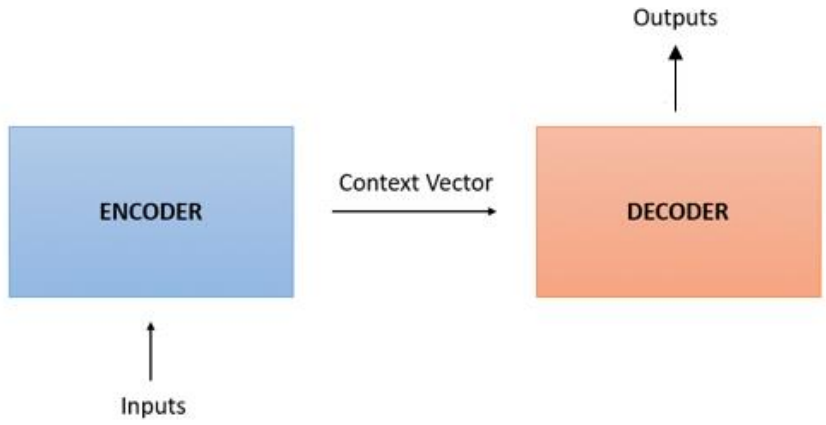


Figure 3.11 Encoder Decoder Architecture

Both the encoder and the decoder are frequently implemented as LSTM models, while they are occasionally implemented as GRU models. The encoder's function is to receive the input sequence and summarize the data into internal state vectors, also referred to as hidden state and cell state vectors, used in LSTM architecture [101]. Only the internal states are retained, while the encoder's outputs are discarded. This context vector is intended to gather data from each input element to help the decoder generate accurate forecasts.

The basic structure of the Seq2Seq architecture with the LSTM layer is shown in the figure.

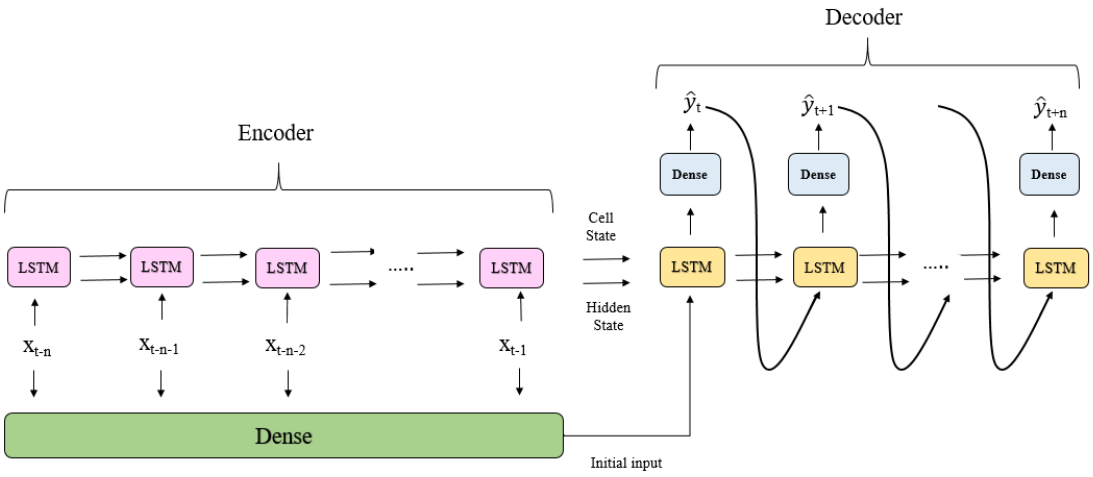


Figure 3.12 Seq2Seq Architecture with LSTM Cells

The LSTM, or GRU, executes each sequence one step at a time while processing the data sequentially. The LSTM examines the input sequence in 't' time steps if its length is 't'. x_t denotes the input sequence at each step in time *t*. The hidden state ('*h*') and the cell state ('*c*') are the two states that the LSTM retains at each time step. The internal condition of the LSTM at the first step was created through the integration of these states. $\hat{y}_t$ is the output sequence at step *t* [102].

The beginning states of the decoder, an LSTM, are configured to correspond with the encoder LSTM's final states. The context vector is fed into the decoder network's first cell as input from the final encoder cell. With these initial states, the decoder initiates the output sequence generation while also incorporating these outputs for subsequent predictions [103].

The initial predicted value $\hat{y}_t$which was achieved by reducing the dimension of the LSTM cell output through a dense layer, along with the cell state and hidden state, are directly inputted into the subsequent LSTM cell, resulting in $\hat{y}_{t+1}$. The weights of the entire model are learned through back-propagation. Dense layer takes the decoder outputs $\hat{y}_{t,}\hat{y}_{t+1},\hat{y}_{t+2},\ldots,\hat{y}_{t+n}$ and inputs them into a dense layer with only one output and no activation function. The information contained in the encoder only transmits

via the hidden states in the GRU Seq2Seq model, in which GRU cells take on the role of LSTM cells [53].

3.6. Proposed Models

During the study, two hybrid models of Seq2Seq and CNN was employed, and since LSTM and GRU achieved superior performance in the literature, they were employed as cell states within these hybrid models.

Similar to the basic Seq2Seq architecture, these hybrid models have an encoder and decoder architecture. The encoder receives the input sequence as a hidden context vector, and the decoder uses this vector to generate an output [104]. For the CNN-Seq2Seq LSTM model, in the encoder part, a convolution layer is added between the dense layers and LSTM cells. Accordingly, for the CNN-Seq2Seq GRU model, a convolution layer is added between dense and GRU cells. The illustrations of the proposed hybrid Seq2Seq and CNN models are shown in the figures.

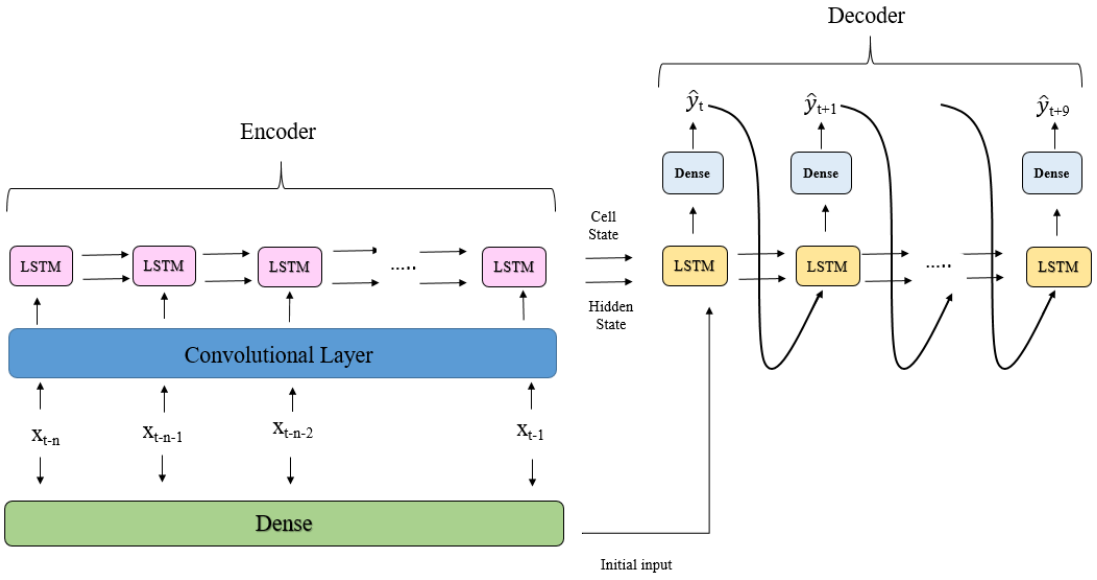


Figure 3.13 Seq2Seq CNN Architecture with LSTM Cells

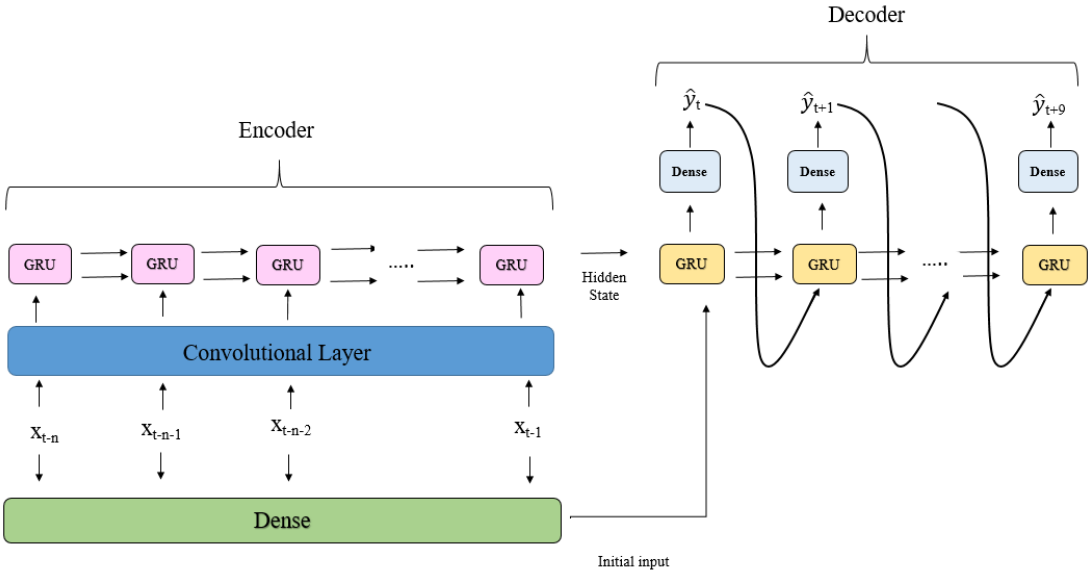


Figure 3.14 Seq2Seq CNN Architecture with GRU Cells

CNN is employed for extracting coupling characteristics, whereas LSTM cells focus on capturing time-series features. In addition, LSTM reduces data and determines the encoder layer's hidden state. Tasks involving load forecasting are performed using LSTM in accordance with the weight matrix and input data.

For the GRU hybrid model, instead of LSTM cells, GRU cells transmit input sequence from encoder part to decoder part only using hidden states. GRU cells may be preferred over LSTM since it is a simplified version of LSTM with fewer gates.

Batch normalization is integrated between the encoder and decoder stages of both Seq2Seq-CNN architectures. The incorporation of batch normalization is crucial since it has been demonstrated to improve precision, encourage generalization, and accelerate the training process. Batch normalization is highly favored in deep learning due to its effectiveness, particularly in more intricate models like Seq2Seq, surpassing the simplicity of basic models [105].

3.7. Performance Metrics

Performance or determination metrics generally examine the accuracy and error size of expected data compared to actual observations, evaluating the degree and precision of variances between anticipated values and the real data [53]. In order to assess the effectiveness of the proposed hybrid models, four distinct performance metrics are utilized. These include R^2, MSE, MAE and MAPE.

For linear regression models, R^2 is a measure of goodness-of-fit that demonstrates a percentage of the variance in the dependent variable that can be interpreted by the independent variables. This statistic, which is usually shown on a scale from 0% to 100%, measures the effectiveness of the correlation between the dependent variable and the model. When a model's forecasted value fits the observed data, it is expected to have an R^2 value close to 1. This indicated that the independent variables in the model effectively interpret a significant amount of the variability in the dependent variable.

It is computed using the formula:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}, \tag{21}$$

Where $SS_{res} = \sum_{i=1}^{n}(y_i - \widehat{f_i})^2 = \sum_{i=1}^{n} {e_i}^2$ shows the residual of the sum of squares and $SS_{tot} = \sum_{i=1}^{n}(y_i - \bar{y})^2$ demonstrates the total sum of squares. Given that $\bar{y}$ is the mean of the observed data, y_i is the i^{th} observation and $\hat{f}_i$ is the forecasted value of the i^{th} observation.

MSE computes the average of the squared differences between the forecasted values and the actual observations. A lower MSE suggests better performance, whereas a higher MSE suggests poorer model accuracy.

MSE is calculated by

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}_i\right)^2 = \frac{1}{n}\sum_{i=1}^{n} {e_i}^2, \tag{22}$$

In the context of the MSE calculation, $\hat{f}_i$ represents the forecasted value of the i^{th} observation, and y_i represents the i^{th} observation or actual value.

One statistic used to assess the average magnitude of the differences between forecasted values and actual observations is MAE. Similar to MSE, a lower MAE suggests a better-performing model, and a higher MAE shows a less accurate model. There's an inverse relationship between prediction accuracy and the MAE metric. MAE can be calculated by

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{f}_i\right| = \frac{1}{n}\sum_{i=1}^{n}|e_i|\,. \tag{23}$$

MAPE is another determination metric that indicates the average difference between the forecasted and actual values in percentages, regardless of whether the deviation was positive or negative. A forecast's accuracy is demonstrated by its reduced MAPE; the forecast's outcome and its actual result correspond precisely when the MAPE is 0%. Conversely, a higher MAPE signifies a lower level of accuracy in the prediction. MAPE can be calculated by

$$MAPE = 100\frac{1}{n}\sum_{i=1}^{n}\frac{\left|y_i - \hat{f}_i\right|}{y_i} \tag{24}$$

3.8. Data

The dataset comprises hourly electricity generation data from a wind farm situated in Karaburun, İzmir, Turkey, spanning from June 2013 to January 2023 and consisting of 83452 recorded observations obtained by [20]. Karaburun is a district of the Izmir province of Turkey.

Karaburun is located 106 kilometers away from the city center of Izmir. It spans a surface area of 421 square kilometers. To its south lies the Urla district, while it is bordered by the Aegean Sea to the west, north, and east [9].

Karaburun WPP ranks as Turkey's 79th largest power plant and the 4th largest in Izmir, boasting an installed capacity of 226.80 megawatts (MWe). It stands as Turkey's 2nd largest wind power plant, utilizing 83 Enercon wind turbines in its RES. With an average electricity production of 477,512,455 kilowatt-hours, the Karaburun Wind Power Plant has the capacity to fulfill the electrical energy requirements of 131,474 individuals in their daily activities, encompassing housing, industry, metro transportation, government offices, and environmental lighting. When considering solely residential electricity consumption, the plant's output can cater to the energy needs of 159,864 households [18].

Table 3.2 Karaburun Wind Power Plant Annual Electricity Production

Year	Electricity Production (kWh)	Proportion relative to Izmir's consumption	Proportion relative to Turkey's consumption
2013	196.768.270	%1,2	%0,08
2014	306.793.540	%1,8	%0,12

Table 3.2 (continued)

2015	320.313.500	%1,8	%0,12
2016	384.763.440	%2,1	%0,14
2017	345.962.000	%1,8	%0,12
2018	337.419.000	%1,6	%0,11
2019	443.780.000	%2,2	%0,15
2020	768.267.000	%3,7	%0,25
2021	840.629.800	%4,0	%0,26
2022	830.428.000	%4,0	%0,25

The table above demonstrates the annual electricity production in kWh of Karaburun WPP from 2013 to 2022. It also presents the production as a proportion relative to the consumption of Izmir and Turkey. Over the years, electricity production has accounted for 4% of Izmir's consumption and 0.25% of Turkey's consumption.

3.9. Data Standardization

Standardizing the input sequences is essential since there are various units of variables and a wide range of numerical ranges that are important for forecasting. Using data sets with cleaned data may lead to some errors in predicted values or decrease the efficiency of the process. Therefore, standardization ensures the data's accuracy,

efficiency, and dependability. Additionally, data normalization should be applied to allow concurrent comparison and weighing of data across sequences. Data normalization is the process of scaling and mapping data to the [0,1] interval. All variables are normalized and then transformed into values without dimensions. After normalization, all “0” variables can remain, which makes it suitable for loading data that contains "0" values. Normalization can enhance the efficacy of network training by preventing gradient explosions during the training phase [105].

Normalization can be computed by the formula,

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{25}$$

Where X' is the normalized data, X is the actual data, X_{max} and X_{min} demonstrates the maximum and minimum value in the sequence.

3.10. Forecasting

This section examines the analysis and forecasting of electricity demand generated by wind speed. In order to evaluate the performance of the proposed models, different models were tested and the results were compared. These models are CNN, LSTM, GRU, CNN-GRU, CNN-LSTM, CNN-Seq2Seq GRU, and CNN - Seq2Seq LSTM. All the models presented are trained using Tensorflow. TensorFlow is an open-source machine learning architecture that simplifies the development of machine learning models, notably for deep learning tasks including neural networks. Initially, pre-processing was done on the data, which involved interpolating missing values. Subsequently, the original data were normalized in order to prepare the suggested models for training, validation, and testing. In order to minimize the mean squared error, the training was performed using the Adam optimizer and the stochastic gradient descent backpropagation method [99]. The Keras library and the Adam optimizer are employed with a batch size of 48 and a learning rate set to 0.00001. To mitigate overfitting, a dropout layer with a dropout rate of 0.1 is inserted before each dense

layer, in addition to the implementation of an early stopping mechanism. The dataset is divided into lags and actual values with sliding windows. This divided data is then separated into two parts, which are the training set and the test set. The model is trained on 80% of the data, and 20% is retained for testing. The train set is used to train the model via backpropagation, whereas the test set evaluates the model's performance on previously unknown data. In the convolutional and output layers, the ReLU activation function was employed with a filter size of 256, a kernel size of 8, and a stride of 2. Different cell sizes were trained, and LSTM cells were implemented with both hidden states and cell states set to a dimension of 196 in the first phase of the empirical analysis. Furthermore, LSTM and GRU cells with an internal size of 196 are used in the Seq2Seq model. A number of statistical measures, including R^2, MAE, MSE, and MAPE, were employed to evaluate the suggested model's accuracy.

A variety of models with different lag lengths were evaluated in order to identify the optimal lag duration. Only lag data are used in order to estimate observations for time steps *t,t+1,t+2,…,t+9* hours. The objective is to compute $\hat{y}_{t,}\hat{y}_{t+1}, \hat{y}_{t+2}, \ldots, \hat{y}_{t+9}$ in a sequence without explicitly providing the method with observed values. Therefore, lagged values were used as inputs to estimate electricity production for the subsequent 10 hours without relying on current observations.

CHAPTER 4

ANALYZES

The hybrid models were evaluated across ten different lag lengths, ranging from 24 to 240, with their effectiveness and reliability assessed over 10-hour intervals by comparing some performance metrics as R^2, MAE, MSE and MAPE. In the table below, the results of all lag lengths are given for the Seq2Seq-CNN model with LSTM cells.

Table 4.1 Results of Seq2Seq-CNN LSTM Model

LAG LENGHT	FORECASTED VALUES	R^2	MAE	MSE	MAPE
24	Y_t	0.9421	0.0505	0.0060	23.5129
	Y_{t+1}	0.8602	0.0793	0.0145	38.7022
	Y_{t+2}	0.7829	0.1038	0.0226	55.3835
	Y_{t+3}	0.7131	0.1220	0.0298	71.1661
	Y_{t+4}	0.6490	0.1377	0.0365	85.0548
	Y_{t+5}	0.5911	0.1503	0.0426	97.6369
	Y_{t+6}	0.5390	0.1612	0.0480	109.4668
	Y_{t+7}	0.4938	0.1708	0.0527	117.5955
	Y_{t+8}	0.4506	0.1788	0.0572	125.7503
	Y_{t+9}	0.4115	0.1861	0.0613	135.2372
48	Y_t	0.9426	0.0498	0.0060	22.9626
	Y_{t+1}	0.8605	0.0823	0.0145	41.0798
	Y_{t+2}	0.7825	0.1061	0.0227	56.6167
	Y_{t+3}	0.7147	0.1228	0.0297	67.5110
	Y_{t+4}	0.6559	0.1369	0.0359	82.0796
	Y_{t+5}	0.6024	0.1487	0.0414	95.7768
	Y_{t+6}	0.5526	0.1597	0.0466	108.5671
	Y_{t+7}	0.5068	0.1701	0.0514	116.9263
	Y_{t+8}	0.4650	0.1798	0.0558	122.7472
	Y_{t+9}	0.4283	0.1879	0.0596	133.4675

Table 4.1 (continued)

72	Y_t	0.9404	0.0534	0.0062	29.8172
	Y_{t+1}	0.8620	0.0826	0.0144	46.1444
	Y_{t+2}	0.7867	0.1053	0.0222	65.1088
	Y_{t+3}	0.7203	0.1237	0.0291	78.7280
	Y_{t+4}	0.6598	0.1380	0.0355	90.0950
	Y_{t+5}	0.6069	0.1508	0.0410	102.4376
	Y_{t+6}	0.5577	0.1619	0.0461	112.8566
	Y_{t+7}	0.5132	0.1723	0.0507	122.1518
	Y_{t+8}	0.4709	0.1815	0.0551	129.2062
	Y_{t+9}	0.4357	0.1893	0.0588	141.9678
96	Y_t	0.9357	0.0558	0.0062	24.6381
	Y_{t+1}	0.8560	0.0848	0.0144	41.0631
	Y_{t+2}	0.7830	0.1065	0.0222	59.7932
	Y_{t+3}	0.7146	0.1237	0.0291	71.6522
	Y_{t+4}	0.6473	0.1409	0.0355	80.6154
	Y_{t+5}	0.5992	0.1516	0.0410	92.9286
	Y_{t+6}	0.5519	0.1626	0.0461	104.2321
	Y_{t+7}	0.5093	0.1724	0.0507	116.3311
	Y_{t+8}	0.4657	0.1816	0.0551	122.5276
	Y_{t+9}	0.4268	0.1902	0.0588	130.9123
120	Y_t	0.9395	0.0536	0.0063	29.8764
	Y_{t+1}	0.8606	0.0809	0.0145	42.7920
	Y_{t+2}	0.7849	0.1011	0.0224	56.0702
	Y_{t+3}	0.7215	0.1191	0.0290	67.2411
	Y_{t+4}	0.6608	0.1349	0.0353	78.7597
	Y_{t+5}	0.6102	0.1458	0.0406	93.9020
	Y_{t+6}	0.5633	0.1575	0.0455	109.0814
	Y_{t+7}	0.5184	0.1677	0.0502	119.7913
	Y_{t+8}	0.4768	0.1784	0.0545	134.6155
	Y_{t+9}	0.4392	0.1874	0.0584	144.0346
144	Y_t	0.9396	0.0524	0.0063	24.0613
	Y_{t+1}	0.8604	0.0818	0.0145	40.2487
	Y_{t+2}	0.7884	0.1021	0.0220	54.8574
	Y_{t+3}	0.7235	0.1193	0.0288	70.4657
	Y_{t+4}	0.6628	0.1333	0.0351	84.4820
	Y_{t+5}	0.6085	0.1458	0.0408	97.8359
	Y_{t+6}	0.5648	0.1570	0.0453	109.6830
	Y_{t+7}	0.5203	0.1671	0.0500	121.0397
	Y_{t+8}	0.4806	0.1769	0.0541	130.4769
	Y_{t+9}	0.4452	0.1849	0.0578	136.8836

Table 4.1 (continued)

168	Y_t	0.9386	0.0532	0.0064	27.7677
	Y_{t+1}	0.8587	0.0832	0.0147	47.1897
	Y_{t+2}	0.7878	0.1045	0.0221	61.5830
	Y_{t+3}	0.7219	0.1205	0.0290	74.6339
	Y_{t+4}	0.6600	0.1355	0.0354	91.1877
	Y_{t+5}	0.6109	0.1480	0.0405	100.9242
	Y_{t+6}	0.5676	0.1588	0.0451	112.0836
	Y_{t+7}	0.5241	0.1682	0.0496	122.6280
	Y_{t+8}	0.4844	0.1765	0.0537	131.2944
	Y_{t+9}	0.4473	0.1846	0.0576	141.1595
192	Y_t	0.9396	0.0526	0.0063	26.2387
	Y_{t+1}	0.8607	0.0827	0.0145	40.8873
	Y_{t+2}	0.7889	0.1032	0.0220	52.9448
	Y_{t+3}	0.7258	0.1181	0.0286	66.7107
	Y_{t+4}	0.6678	0.1337	0.0346	80.2469
	Y_{t+5}	0.6162	0.1458	0.0400	91.6787
	Y_{t+6}	0.5700	0.1567	0.0448	101.7883
	Y_{t+7}	0.5275	0.1662	0.0492	110.4318
	Y_{t+8}	0.4820	0.1768	0.0540	116.4623
	Y_{t+9}	0.4473	0.1847	0.0576	125.7115
216	Y_t	0.9352	0.0568	0.0068	24.017
	Y_{t+1}	0.8582	0.0831	0.0148	42.225
	Y_{t+2}	0.7841	0.1053	0.0225	55.895
	Y_{t+3}	0.7196	0.1221	0.0292	70.466
	Y_{t+4}	0.6610	0.1363	0.0353	77.507
	Y_{t+5}	0.6143	0.1468	0.0402	91.253
	Y_{t+6}	0.5703	0.1573	0.0448	103.70
	Y_{t+7}	0.5278	0.1675	0.0492	114.89
	Y_{t+8}	0.4876	0.1766	0.0534	123.67
	Y_{t+9}	0.4547	0.1839	0.0568	138.387
240	Y_t	0.9394	0.0529	0.0063	26.2215
	Y_{t+1}	0.8593	0.0829	0.0146	45.2755
	Y_{t+2}	0.7882	0.1033	0.0220	57.4312
	Y_{t+3}	0.7229	0.1213	0.0288	72.6087
	Y_{t+4}	0.6608	0.1364	0.0353	82.3078
	Y_{t+5}	0.6104	0.1482	0.0405	96.1723
	Y_{t+6}	0.5622	0.1598	0.0456	108.3588
	Y_{t+7}	0.5178	0.1690	0.0502	118.1017
	Y_{t+8}	0.4754	0.1789	0.0546	129.1235
	Y_{t+9}	0.4395	0.1877	0.0584	136.5407

Based on the table, the optimal performance for short-term forecasting within the first hour occurs when the lag length is set to 48. The highest achieved R^2 value is 0.94, while the lowest is 0.42 for this lag length. Despite the decrease in the R^2 value for the next few hours, both MAE and MSE results indicate that the model's performance remains within an acceptable range. Consequently, the model demonstrates impressive results with an MAE of 0.04 for the first hour and 0.18 for the final hour, which can be regarded as a good result compared to other models mentioned in the literature. Nevertheless, in the context of long-term forecasting for the 10th hour, the model yields superior results when the lag length is set to 216, achieving an R^2 value of 0.45. Furthermore, the obtained MAE, MSE, and MAPE results are notably low, suggesting that the model can effectively be utilized for long-term forecasting with a lag length of 216.

The graphs indicating the forecasted values and observed values are demonstrated in figures for the 1st, 4th,7th and 10th hours.

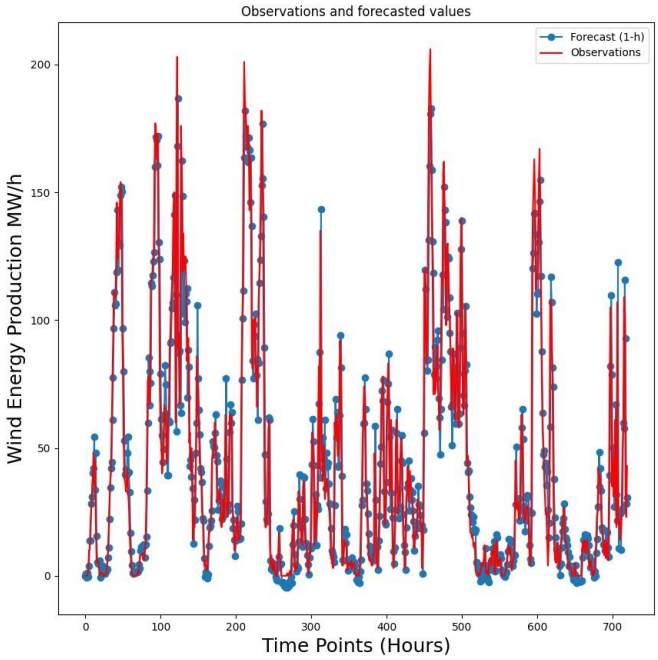


Figure 4.1 Forecasted Values and Observations of 1st hour for LSTM cells

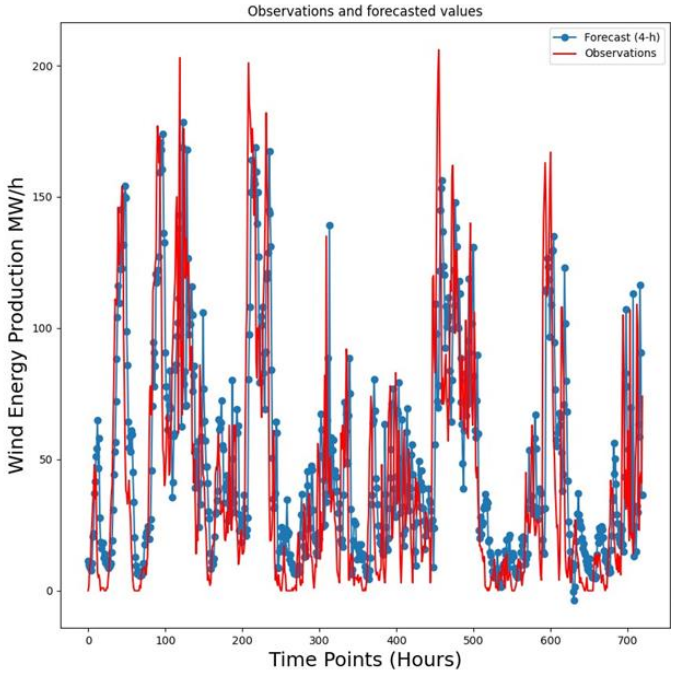


Figure 4.2 Forecasted Values and Observations of 4th hour for LSTM cells

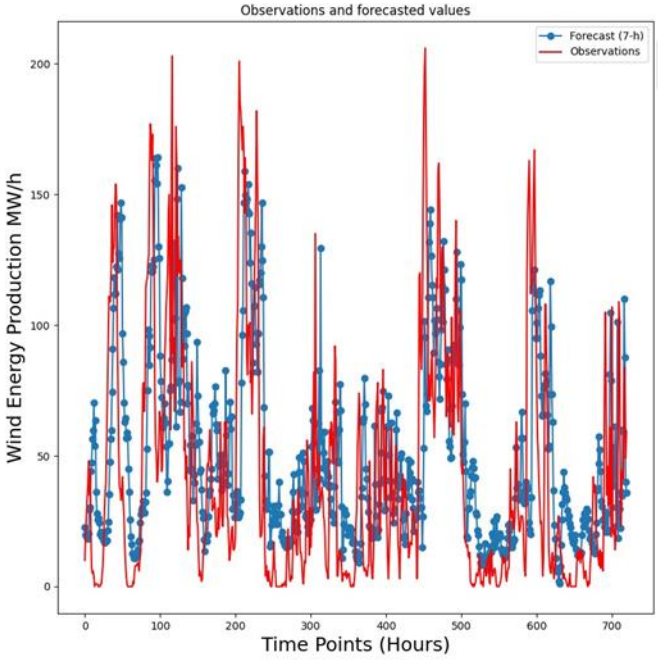


Figure 4.3 Forecasted Values and Observations of 7th hour for LSTM cells

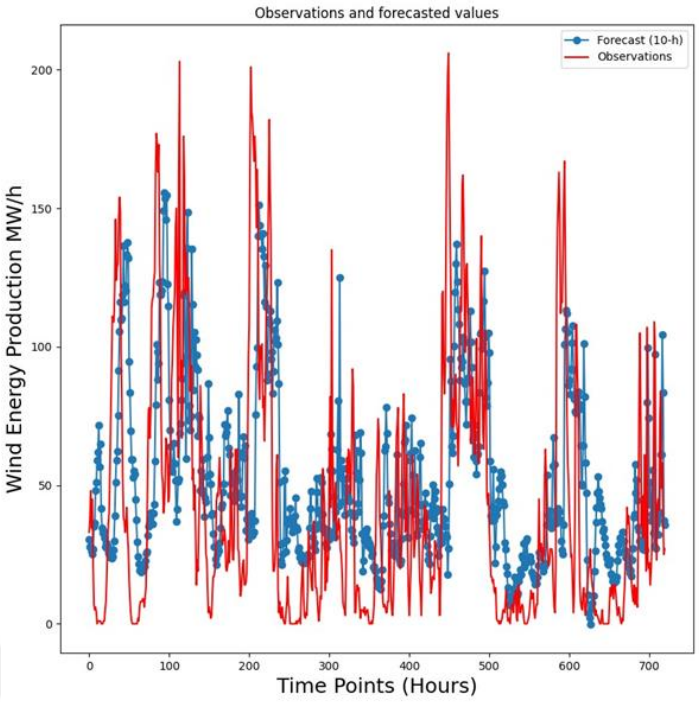


Figure 4.4 Forecasted Values and Observations of 10th hour for LSTM cells

The graphs illustrate hourly outcomes based on the optimal lag length and parameters, which are 216 for the LSTM cells. The x-axis represents time points in hours, while the y-axis indicates wind energy production measured in MW/h. The red lines represent observed data, while the blue points indicate the forecasted values. The aim of the graph is to compare observed values with forecasted values, to assess the accuracy and performance of the forecasting model. When the blue points align closely with the red lines, it indicates that the forecasted values closely match the actual data, suggesting that the model provides accurate solutions. This close alignment can be observed in the initial graph, which represents the first hour of forecasting. Furthermore, the table analysis confirms this observation, revealing an R^2 value of 0.93 for the first hour of forecasting. When examining the graphs for other time intervals, it becomes evident that the accuracy of the model decreases gradually as the observed and forecasted values deviate from a perfect fit. Although, the model does not show perfect fits for the last hours, it demonstrates sufficiently satisfactory results in long term forecasting.

Additionally, GRU cells were implemented into the hybrid Seq2Seq-CNN model. The table below indicates the forecasted result of ten consecutive hours with ten different lag lengths.

Table 4.2 Results of Seq2Seq-CNN GRU Model

LAG LENGHT	FORECASTED VALUES	R^2	MAE	MSE	MAPE
24	Y_t	0.9409	0.0510	0.0061	21.5303
	Y_{t+1}	0.8576	0.0811	0.0148	37.8802
	Y_{t+2}	0.7758	0.1043	0.0233	53.8530
	Y_{t+3}	0.7017	0.1231	0.0310	67.9595
	Y_{t+4}	0.6344	0.1394	0.0381	80.5461
	Y_{t+5}	0.5951	0.1526	0.0442	92.0608
	Y_{t+6}	0.5411	0.1644	0.0499	102.2006
	Y_{t+7}	0.4808	0.1745	0.0551	109.9538
	Y_{t+8}	0.4359	0.1839	0.0598	117.6838
	Y_{t+9}	0.3944	0.1922	0.0641	124.7830
48	Y_t	0.9406	0.0518	0.0062	22.679
	Y_{t+1}	0.8537	0.0846	0.0153	42.199
	Y_{t+2}	0.7718	0.1081	0.0238	58.600
	Y_{t+3}	0.6975	0.1261	0.0315	71.344
	Y_{t+4}	0.6302	0.1417	0.0386	84.976
	Y_{t+5}	0.5889	0.1548	0.0449	94.674
	Y_{t+6}	0.5347	0.1670	0.0506	103.741
	Y_{t+7}	0.4932	0.1772	0.0559	111.586
	Y_{t+8}	0.4372	0.1864	0.0607	118.195
	Y_{t+9}	0.3865	0.1947	0.0650	124.709
72	Y_t	0.9387	0.0534	0.0064	26.8411
	Y_{t+1}	0.8530	0.0832	0.0153	44.1738
	Y_{t+2}	0.7700	0.1061	0.0240	60.5370
	Y_{t+3}	0.6924	0.1250	0.0321	73.9844
	Y_{t+4}	0.6348	0.1414	0.0391	86.0431
	Y_{t+5}	0.5835	0.1544	0.0455	96.4111
	Y_{t+6}	0.5268	0.1666	0.0514	104.4145
	Y_{t+7}	0.4656	0.1774	0.0567	112.7798
	Y_{t+8}	0.4207	0.1868	0.0614	120.2895
	Y_{t+9}	0.3995	0.1950	0.0657	125.9636

Table 4.2 (continued)

96	Y_t	0.9377	0.0548	0.0065	25.2562
	Y_{t+1}	0.8526	0.0851	0.0154	41.6705
	Y_{t+2}	0.7698	0.1085	0.0240	58.2835
	Y_{t+3}	0.6917	0.1278	0.0321	71.5968
	Y_{t+4}	0.6247	0.1435	0.0391	84.8406
	Y_{t+5}	0.5639	0.1563	0.0454	95.3220
	Y_{t+6}	0.5189	0.1680	0.0512	105.0133
	Y_{t+7}	0.4652	0.1785	0.0568	113.1897
	Y_{t+8}	0.4202	0.1875	0.0615	121.4522
	Y_{t+9}	0.3907	0.1956	0.0656	128.2192
120	Y_t	0.9396	0.0527	0.0063	24.733
	Y_{t+1}	0.8512	0.0861	0.0155	43.732
	Y_{t+2}	0.7692	0.1080	0.0241	58.108
	Y_{t+3}	0.7117	0.1278	0.0321	73.983
	Y_{t+4}	0.6737	0.1435	0.0392	86.072
	Y_{t+5}	0.6123	0.1564	0.0456	94.959
	Y_{t+6}	0.5667	0.1678	0.0514	103.95
	Y_{t+7}	0.5052	0.1779	0.0568	112.60
	Y_{t+8}	0.4400	0.1883	0.0615	119.512
	Y_{t+9}	0.3996	0.1958	0.0657	126.989
144	Y_t	0.9388	0.0525	0.0064	25.0925
	Y_{t+1}	0.8525	0.0851	0.0154	48.3416
	Y_{t+2}	0.7996	0.1082	0.0240	63.4497
	Y_{t+3}	0.7126	0.1266	0.0320	75.9627
	Y_{t+4}	0.6647	0.1432	0.0391	89.7827
	Y_{t+5}	0.6048	0.1563	0.0453	100.6140
	Y_{t+6}	0.5497	0.1677	0.0511	110.0718
		0.4996	0.1778	0.0563	117.7987
	Y_{t+7}	0.4447	0.1870	0.0610	125.3573
	Y_{t+8}	0.3964	0.1949	0.0650	132.2962
	Y_{t+9}				
168	Y_t	0.9390	0.0528	0.0064	24.3158
	Y_{t+1}	0.8516	0.0849	0.0155	42.4260
	Y_{t+2}	0.7689	0.1075	0.0241	56.2441
	Y_{t+3}	0.7311	0.1279	0.0322	73.5060
	Y_{t+4}	0.6943	0.1434	0.0392	84.8665
	Y_{t+5}	0.6441	0.1564	0.0454	95.8690
	Y_{t+6}	0.5890	0.1680	0.0512	105.1963
	Y_{t+7}	0.5274	0.1785	0.0565	113.5537
	Y_{t+8}	0.4622	0.1875	0.0613	119.9522
	Y_{t+9}	0.4021	0.1956	0.0654	126.2146

Table 4.2 (continued)

192	Y_t	0.9382	0.0543	0.0064	29.4804
	Y_{t+1}	0.8517	0.0852	0.0155	45.6349
	Y_{t+2}	0.7716	0.1075	0.0238	60.0041
	Y_{t+3}	0.7032	0.1277	0.0320	72.5612
	Y_{t+4}	0.6546	0.1439	0.0391	84.5084
	Y_{t+5}	0.5989	0.1566	0.0453	95.9538
	Y_{t+6}	0.5389	0.1682	0.0512	104.8605
	Y_{t+7}	0.4987	0.1787	0.0564	113.3148
	Y_{t+8}	0.4541	0.1877	0.0611	121.2718
	Y_{t+9}	0.4030	0.1955	0.0654	127.8713
216	Y_t	0.9411	0.0509	0.0061	22.491
	Y_{t+1}	0.8553	0.0828	0.0151	37.968
	Y_{t+2}	0.7947	0.1061	0.0235	55.563
	Y_{t+3}	0.7299	0.1257	0.0313	70.532
	Y_{t+4}	0.6732	0.1419	0.0382	85.656
	Y_{t+5}	0.6245	0.1549	0.0443	97.501
	Y_{t+6}	0.5509	0.1663	0.0499	107.29
	Y_{t+7}	0.4501	0.1765	0.0551	114.61
	Y_{t+8}	0.4461	0.1857	0.0598	121.88
	Y_{t+9}	0.4020	0.1936	0.0640	128.653
240	Y_t	0.9396	0.0531	0.0063	21.6182
	Y_{t+1}	0.8580	0.0831	0.0148	41.2530
	Y_{t+2}	0.7811	0.1050	0.0228	56.7752
	Y_{t+3}	0.7097	0.1243	0.0303	73.0032
	Y_{t+4}	0.6482	0.1391	0.0367	84.7700
	Y_{t+5}	0.5931	0.1520	0.0424	97.1951
	Y_{t+6}	0.5446	0.1629	0.0475	107.7716
	Y_{t+7}	0.4982	0.1726	0.0523	116.8472
	Y_{t+8}	0.4546	0.1815	0.0568	124.6525
	Y_{t+9}	0.4121	0.1900	0.0613	131.7567

The same model was trained using GRU cells instead of LSTM cells. As demonstrated in the table, the optimal performance for short-term forecasting within the first hour was achieved with a lag length of 24, yielding an R^2 value of 0.94, which is similar to the performance of the LSTM cell. This metric then declines to 0.39 for a forecast horizon of ten hours. The MAE results for this lag length also indicate favorable outcomes, starting at 0.05 and increasing to 0.1922 by the 10th hour. It's important to highlight that MAE values approaching zero indicate a high level of accuracy and

satisfactory performance by the model. When examining long-term forecasting, it's evident that the model achieves superior performance when the lag length is set to 240. Comparing the R^2 and MAE values for this lag length at the 10th hour, it's apparent that this length yields more favorable outcomes compared to other lengths. The highest R^2 value obtained is 0.41, accompanied by an MAE value of 0.1900. Consequently, the model demonstrates its suitability for long-term forecasting tasks when the lag length is set to 240.

Accordingly, the graphs illustrating the forecasted values and observed values for the hybrid model with GRU cells are presented in the figures for the 1st, 4th,7th and 10th hours.

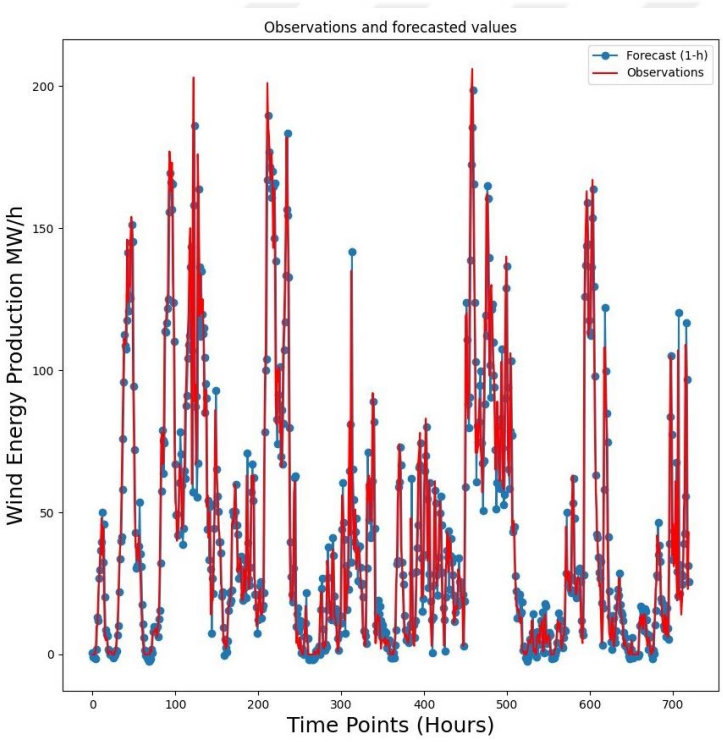


Figure 4.5 Forecasted Values and Observations of 1st hour for GRU cells

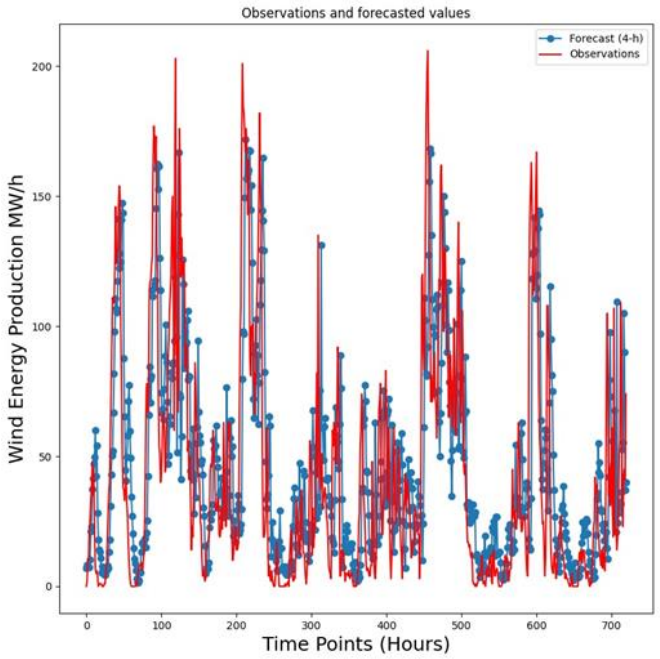


Figure 4.6 Forecasted Values and Observations of 4th hour for GRU cells

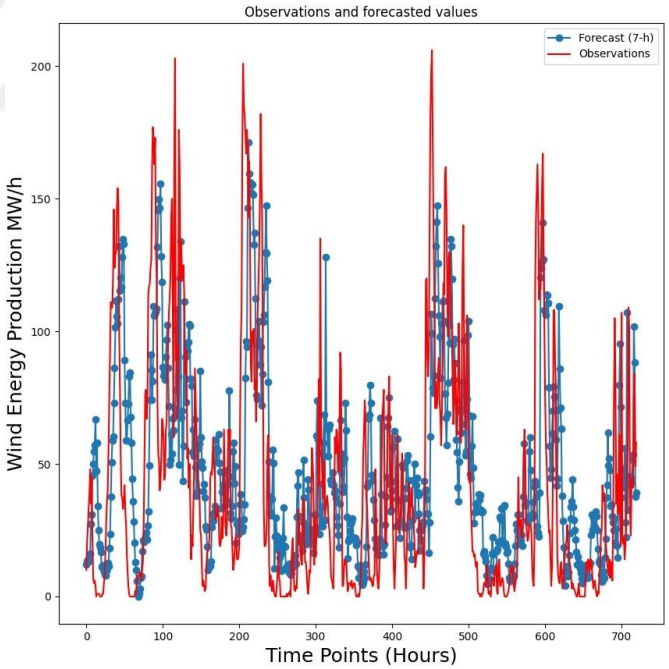


Figure 4.7 Forecasted Values and Observations of 7th hour for GRU cells

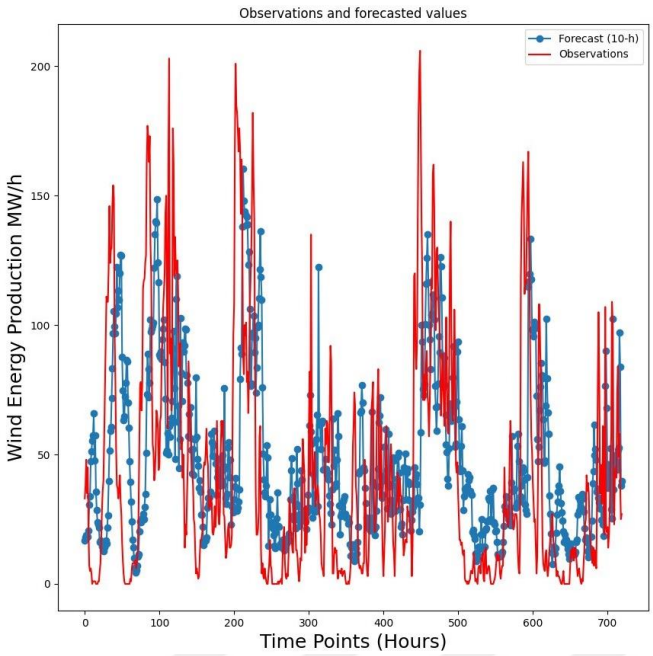


Figure 4.8 Forecasted Values and Observations of 10th hour for GRU cells

Hourly results based on the optimal lag length of 240 and parameters for the GRU cells are represented in the graphs. The initial graph, representing the prediction for the 1st hour, displays a strong correlation between observed and forecasted values. This suggests that the hybrid Seq2Seq-CNN GRU model performs exceptionally well in short-term forecasting, as evidenced by the R^2 and MAE results presented previously in the table. Upon examination, it's observed that the correlation between observed and forecasted values diminishes at the 4th, 7th, and 10th hours. Despite this decline, the model still exhibits acceptable performance in mid and long-term forecasting scenarios.

Two different hybrid models, Seq2Seq-CNN LSTM and Seq2Seq-CNN GRU, were evaluated over 10-hour periods with various lag lengths. Forecasted values according to the performance metrics and correlation graphs of the observed value and predicted value were analyzed. While both models yield comparable outcomes, the hybrid model employing LSTM cells outperforms the model with GRU cells, considering all ten lag

lengths. This superiority is evident in performance metrics including R^2, MAE, MSE, and MAPE, observed in both short-term and long-term evaluation.

In order to demonstrate the accuracy of the proposed models, data was forecasted using various models in the literature such as CNN, LSTM, GRU, CNN-GRU Hybrid, CNN-LSTM Hybrid and the outcomes were compared. Table 4.3 demonstrates the forecasting results of each model.

Table 4.3 Forecasting Results of the Models

Method	Resolution	R^2	MAE	MSE	MAPE
CNN + Seq2Seq (GRU)	1-h	0.9411	0.0508	0.0061	22.4905
	5-h	0.6732	0.1419	0.0382	85.6555
	10-h	0.4020	0.1936	0.0639	128.6531
CNN GRU	1-h	0.9396	0.0526	0.0062	23.2136
	5-h	0.6470	0.1408	0.0367	86.6890
	10-h	0.4151	0.1917	0.0609	137.8977
CNN LSTM	1-h	0.9380	0.0535	0.0064	26.9151
	5-h	0.6388	0.1428	0.0376	87.0787
	10-h	0.3952	0.1965	0.0630	140.2151
LSTM	1-h	0.9337	0.0540	0.0069	24.9199
	5-h	0.6180	0.1420	0.0398	79.9261
	10-h	0.3722	0.1940	0.0654	125.3772
GRU	1-h	0.9393	0.0524	0.0063	22.1651
	5-h	0.6195	0.1404	0.0396	78.6668
	10-h	0.3665	0.1957	0.066	124.6236
CNN	1-h	0.9376	0.0542	0.0065	29.2529
	5-h	0.6572	0.1382	0.0357	90.7340
	10-h	0.4217	0.1907	0.0602	137.7135
CNN + Seq2Seq (LSTM)	1-h	0.9351	0.0567	0.0067	24.0179
	5-h	0.6610	0.1362	0.0353	77.5070
	10-h	0.4546	0.1839	0.0568	138.3877

The evaluation in the given table involved analyzing forecasts for 1st, 5th and 10th hours for all models using performance metrics like R^2, MAE, MSE, and MAPE. This ten hours' interval can be considered short, mid and long term forecast solution. Lag lengths starting from 24 to 240 were trained and results were examined. The accuracy of the results did not exhibit a direct or inverse correlation with increasing lag length. Consequently, the values presented in the table were chosen by selecting random lag lengths, indicating no systematic relationship between lag length and accuracy.

Likewise, various dropout values were experimented with on the dataset, revealing no noticeable relationship with the outcomes.

Notably, the initial hour for each model showed significantly better results compared to the 5^{th} and 10^{th} hours. Despite a gradual decrease in performance leading up to the 10^{th} hour, the models exhibited commendable performance compared to existing forecasting models in the literature. Regarding comparison between models, while most models showed similar performance in the 1^{st} hour, the Seq2Seq-CNN GRU model was the most accurate, at 94%. It indicates that this hybrid model can be effectively used for short-term forecasts. Furthermore, the table reveals that the Seq2Seq-CNN LSTM model achieves 45% accuracy at the 10th hour, representing the highest performance among all other models. This suggests that, for long-term forecasts, the hybrid Seq2Seq CNN model with LSTM layers demonstrates significant efficiency and effectiveness.

CHAPTER 5

CONCLUSION

The demand for energy has been increasing recently around the world. Nuclear energy and fossil fuels have been the main sources of supply for this need until now. But since they have an adverse effect on the environment and are limited in supply, people started looking for alternative natural energy sources. Renewable energy, derived from natural sources like sunlight, wind, water, and geothermal heat, ensures sustainability without polluting the environment. These sources provide numerous environmental and economic advantages compared to conventional energy sources. Renewable energy is sourced from resources that are essentially inexhaustible and emit fewer pollutants. This fundamental distinction from fossil fuels has led many nations, including Turkey, to advocate for its widespread adoption and utilization. Wind energy is considered the cleanest and least damaging natural energy source. Wind energy does not face transportation challenges and does not demand high-end technology for utilization. The technology for converting wind energy into mechanical and electrical energy is more cost-effective compared to other energy conversion systems.

Wind power currently accounts for approximately 10% of Turkey's electricity generation, primarily concentrated in the western regions, including the Aegean and Marmara areas. It is steadily increasing its contribution to the overall renewable energy portfolio of the nation. As of 2023, Turkey has a total installed capacity of 11 GW from wind turbines.

Accurate wind power estimations are essential for power generation planning and power grid management. Therefore, suitable and accurate forecasting models are required for the planning and operation of electrical power systems. Research into

developing a more accurate forecasting system is driven by the significance of forecasting, which would greatly reduce operating and maintenance costs and increase the reliability of the power supply and distribution system. A wide range of approaches have been used in the research on renewable power forecasting. Forecasting wind energy and velocity over a range of timescales, estimating uncertainty, and predicting ramping events have all been the subject of several forecasting studies. Mainly, time series and statistical techniques such as ARIMA models were used for the forecasting.

The aim of this study is to model and forecast the hourly electricity production of one of the biggest wind farms in Turkey, which is located in Karaburun, İzmir. Throughout this study, short-term, mid-term, and long-term forecasts are essential. To enhance forecasting accuracy beyond traditional models, machine learning, deep learning, and hybrid models have been devised and evaluated using the available data. Specifically, the Seq2Seq-CNN hybrid model has been developed, integrating LSTM cells and GRU cells independently to improve forecasting performance. Seq2Seq models have demonstrated effectiveness across various sequential data tasks, particularly those characterized by complex relationships and variable lengths between input and output. In the context of time series forecasting, Seq2Seq models were enhanced by integrating CNN to capitalize on historical data for predicting future data points. These hybrid models utilize LSTM and GRU cells, which have shown superior performance in existing literature.

The performance of these hybrid models was assessed by comparing them with CNN, LSTM, GRU, CNN-GRU, and CNN-LSTM models using various performance metrics to investigate the error margin of the forecasts, which are R^2, MAE, MSE, and MAPE. Each model was optimized using the most suitable parameters, with 10 lag lengths ranging between 24 and 240. For each lag length, the models were executed, and a graph illustrating 10 hours of observed and forecasted values was generated, starting from the time y_t to y_{t+9}. Upon examining the results, it was noted that the models yielded comparable outcomes. It was observed that while short-term predictions were highly accurate, prediction accuracy diminished over the long term. The objective of this study was to ensure a reasonable level of accuracy in predicting

wind energy over the long term. Despite the decline in results, the analysis of performance metrics revealed that the 10th-hour prediction rates remained within an acceptable range and outperformed examples found in the literature.

While the Seq2Seq-CNN GRU model achieves the highest R^2 value of 0.94 for the 1st hour, the Seq2Seq-CNN LSTM model surpasses in long-term forecasting with an R^2 value of 0.45 for the 10th hour. It can be concluded that the Seq2Seq-CNN GRU model is well-suited for short-term forecasting, while the Seq2Seq-CNN LSTM model is preferable for long-term forecasting. Examining the graphs of two hybrid models revealed an almost perfect correlation between the actual data and the forecasted data, indicating that the forecasted values closely aligned with the observed values. Therefore, for this study and the data obtained from İzmir Karaburun, the algorithms were found to be reliable and efficient for forecasting wind power.

REFERENCES

[1] J. Bogner, R. Pipatti, S. Hashimoto, "Mitigation of global greenhouse gas emissions from waste: conclusions and strategies from the Intergovernmental Panel on Climate Change (IPCC) Fourth Assessment Report. Working Group III (Mitigation)". *Waste Management & Researh,* 26(1), pp. 11-32, Feb. 2008.

[2] O. Rosales-Calderon, V. Arantes, "A review on commercial scale high value products that can be produced alongside cellulosic ethanol". *Biotechnol Biofuels*, vol. 12(240), pp. 1-58, Oct. 1979.

[3] R.I. Lotz, "The impact of renewable energy consumption to economic growth: A panel data application". *Energy Economics*, vol. 53, pp. 58-63, Jan. 2016.

[4] "World Energy Outlook 2009, IEA, Paris", Internet: https://www.iea.org/reports/world-energy-outlook-2009, Licence: CC BY 4.0 [Sep. 10, 2023].

[5] B. Moselle, "Why Support Renewables?", presented at EPRG Spring Research Seminar, University of Cambridge, UK, 2011.

[6] K. Kaygusuz, "Energy for sustainable development: key issues and challenges". *Energy Sources B Econ. Plan. Policy*, vol. 2(1), pp. 73–83, Apr. 2007.

[7] K. Adamczak, "Energia odnawialna a bezpiecze ´nstwo Polaków (Renewable energy and the security of Poles)". *Securitologia*, vol. 2(24), pp. 51–61, Feb. 2016.

[8] N. A. Erdemir, "Energy Dependence of Turkey: The Role of Renewable Energy Sources". *Başkent University Journal of Commercial Sciences*, vol. (6)1, pp. 1 – 14. Mar. 2022.

[9] "Wind power in Turkey", Internet: https://en.wikipedia.org/wiki/Wind _power_in_Turkey#cite_note-42, Sep. 17, 2023 [Oct. 14, 2023].

[10] D.R. Drew, D.J. Cannon, J.F. Barlow, P.J. Coker, T.H.A. Frame, "The importance of forecasting regional wind power ramping: A case study for the UK". *Renewable Energy*, vol. 114(B), pp. 1201-1208, Dec. 2017.

[11] G. Marulanda, A. Bello, J. Cifuentes, J. Reneses, "Wind power long-term scenario generation considering spatial-temporal dependencies in coupled electricity markets". *Energies*, vol. 13(13), pp. 1-19, Jul. 2020.

[12] D. Zhang, J. Yuan, J. Zhu, Q. Ji, X. Zhang, H. Liu, "Fault Diagnosis Strategy For Wind Turbine Generator Based On The Gaussian Process Metamodel". *Mathematical Problems in Engineering,* vol. 2020, pp. 1–10, Jan. 2020.

[13] Y. Akbal, K.D. Ünlü, "A deep learning approach to model daily particular matter of Ankara: key features and forecasting". *International journal of Environmental Science and Technology*, vol. 19(1), pp. 5911–5927, Oct. 2021.

[14] A. Krizhevsky, I. Sutskever, GE. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks". *Advances in Neural Information Processing Systems*, vol. 25(2), pp. 1097–1105, Jan. 2012.

[15] J. Tong, L. Xie, W. Yang, K. Zhang, J. Zhao, "Enhancing time series forecasting: A hierarchical transformer with probabilistic decomposition representation". *Information Sciences*, vol. 647, 119410, Nov. 2023.

[16] X. Zhao, S. Wang, T. Li, "Review of Evaluation Criteria and Main Methods of Wind Power Forecasting". *Energy Procedia*, vol. 12, pp. 761-769, Dec. 2011.

[17] "Tracking Clean Energy Progress, 2023, IEA, Paris", Internet: https://www.iea.org/reports/tracking-clean-energy-progress-2023, License: CC BY 4.0, [Sep. 12, 2023].

[18] "TEIAS, The website of Turkish electricity transmission corporations", Internet: https://www.teias.gov.tr/en-US, Sep. 10, 2023 [Sep. 13, 2023].

[19] Y. Tang, Z. Song, Y. Zhu, et al., "A survey on machine learning models for financial time series forecasting". *Neurocomputing*, vol. 512(3), pp. 363–380, Sep. 2022.

[20] "Karaburun Wind Power Plant (WPP)", Internet: www.enerjiatlasi. com/ ruzgar/ karaburun-ruzgar-santrali.html, Aug. 10, 2023 [Sep. 13, 2023].

[21] "Karaburun WPP (Turkey)", Internet: www.thewindpower.net/ windfarm _en_ 6095_karaburun-res.php, Jul. 25, 2023 [Sep. 13, 2023].

[22] J. Jung, R.P. Broadwater, "Current status and future advances for wind speed and power forecasting". *Renewable and Sustainable Energy Reviews*, vol. 31, pp. 762-777, Mar. 2014.

[23] A.P. Marugán, F.P.G. Márquez, J.M.P. Perez, D.R. Hernández, "A survey of artificial neural network in wind energy systems". *Applied Energy*, vol. 228, pp. 1822-1836, Oct. 2018.

[24] G. López, P. Arboleya, “Short-term wind speed forecasting over complex terrain using linear regression models and multivariable LSTM and NARX networks in the Andes Mountains, Ecuador”. *Renewable Energy*, vol. 183(2), pp. 351-368, Jan. 2022.

[25] F. Castellani, et al., “Investigation of terrain and wake effects on the performance of wind farms in complex terrain using numerical and experimental data“. *Wind Energy*, vol. 20(7), pp. 1099-1824, Feb. 2017.

[26] I. Delgado, M. Fahim, “Wind Turbine Data Analysis and LSTM-Based Prediction in SCADA System”. *Energies*, vol. 14(1), pp. 125-137, Jan. 2021.

[27] E. Cadenas, W. Rivera, “Short Term Wind Speed Forecasting In La Venta, Oaxaca, México, Using Artificial Neural Networks”. *Renewable Energy*, vol. 34(1), pp. 274-278, Jan. 2009.

[28] Y. Wang, D.L. Wu, C.X. Guo, Q.H. Wu, W.Z. Qian, J. Yang, "Short-term wind speed prediction using support vector regression” in IEEE PES General Meeting, Minneapolis, MN, USA, 2010, pp. 1-6.

[29] M. Elsaraiti, A. Merabet, “A Comparative Analysis of the ARIMA and LSTM Predictive Models and Their Effectiveness for Predicting Wind Speed”. *Energies*, vol. 14(20), pp. 6782-6798, Oct. 2021.

[30] Y. Noorollahi, M. A. Jokar, A. Kalhor, “Using artificial neural networks for temporal and spatial wind speed forecasting in Iran”. *Energy Conversion and Management*, vol. 115, pp. 17-25, May 2016.

[31] J. Shi, J. Guo, S. Zheng, “Evaluation of hybrid forecasting approaches for wind speed and power generation time series”. *Renewable and Sustainable Energy Reviews*, vol. 16(5), pp. 3471-3480, Jun. 2012.

[32] S.G. Chavez, J.X. Bernat, H.L. Coalla, “Forecasting of energy production and consumption in Asturias (northern Spain)”. *Energy*, vol. 24(3), pp. 183-198, Mar. 1999.

[33] F. Shahid, A. Zameer, M. Muneeb, “A novel genetic LSTM model for wind power forecast”. *Energy*, vol. 223(1), 120069, May 2021.

[34] X. Zhang, R. Wang, T. Zhang, Y. Liu, Y. Zha, “Short-Term Load Forecasting Using a Novel Deep Learning Framework”. *Energies*, vol. 11(6), pp. 1554-1569, Jun. 2018.

[35] S. Subbiah, J. Chinnappan, “Deep learning based short term load forecasting with hybrid feature selection”. *Electric Power Systems Research*, vol. 210, 108065, Sep. 2022.

[36] J. Duan, P. Wang, W. Ma, S. Fang, Z. Hou, “A Novel Hybrid Model Based On Nonlinear Weighted Combination For Short-Term Wind Power Forecasting”. *International Journal of Electrical Power & Energy Systems,* vol. 134(11), 107452, Jan. 2022.

[37] Y. Bengio, “Learning phrase representations using rnn encoderedecoder for statistical machine translation”. in proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2024, pp. 1724-1734.

[38] M.A. Hossain, R.K. Chakrabortty, S. Elsawah and M.J. Ryan, "Hybrid Deep Learning Model for Ultra-Short-Term Wind Power Forecasting". in 2020 IEEE International Conference on Applied Superconductivity and Electromagnetic Devices (ASEMD), Tianjin, China, 2020, pp. 1-2.

[39] Y. Xiao, C. Zou, H. Chi, R. Fang, “Boosted GRU Model For Short-Term Forecasting Of Wind Power With Feature-Weighted Principal Component Analysis”. *Energy,* vol. 267, 126503, Mar. 2023.

[40] Md. A. Hossain, R. Chakrabortty, S. Elsawah, M. Ryan, “Very Short-term Forecasting of Wind Power Generation using Hybrid Deep Learning Model”. *Journal of Cleaner Production,* vol. 296, 126564, Feb. 2021.

[41] Z. Zhao, S. Yun, L. Jia, J. Guo, Y. Meng, N. He, X. Li, J. Shi, L. Yang, “Hybrid VMD-CNN-GRU-Based Model For Short-Term Forecasting Of Wind Power Considering Spatio-Temporal Features”. *Engineering Applications of Artificial Intelligence,* vol. 121, 105982, May 2023.

[42] G. Wang, R. Jia, J. Liu, H. Zhang, “A Hybrid Wind Power Forecasting Approach Based On Bayesian Model Averaging And Ensemble Learning”. *Renewable Energy,* vol. 145, pp. 2426-2434, Jan. 2020.

[43] Z. Shao, J. Han, W. Zhao, K. Zhou, S. Yang, “Hybrid Model For Short-Term Wind Power Forecasting Based On Singular Spectrum Analysis And A Temporal Convolutional Attention Network With An Adaptive Receptive Field”. *Energy Conversion and Management,* vol. 269, 116138, Oct. 2022.

[44] C. Zhang, H. Wei, X. Zhao, T. Liu, K. Zhang, “A Gaussian process regression based hybrid approach for short-term wind speed prediction”. *Energy Conversion and Management,* vol. 126, pp. 1084-1092, Oct. 2016.

[45] B. Zhang, JL. Wu, PC. Chang, “A multiple time series-based recurrent neural network for short-term load forecasting”. *Soft Computing,* vol. 22(5), pp. 4099–4112, Jun. 2018.

[46] H. Dhiman, D. Deb, J. Guerrero, “Hybrid machine intelligent SVR variants for wind forecasting and ramp events”. *Renewable and Sustainable Energy Reviews*, vol. 108, pp. 369-379, Jul. 2019.

[47] M. Yang, C. Shi, H. Liu, “Day-ahead wind power forecasting based on the clustering of equivalent power curves”. *Energy*, vol. 218(3), 119515, Mar. 2021.

[48] K. Chen, J. Yu, “Short-term wind speed prediction using an unscented Kalman filter based state-space support vector regression approach”. *Applied Energy*, vol. 113, pp. 690-705, Jan. 2014.

[49] H.H. Aly, “A Hybrid Optimized Model of Adaptive Neuro-Fuzzy Inference System, Recurrent Kalman Filter and Neuro-Wavelet for Wind Power Forecasting Driven by DFIG”. *Energy*, vol. 239, 122367, Jan. 2022.

[50] H. Bashir, M. Sibtain, Ö. Hanay, M. I. Azam, Q. Ain, S. Saleem, “Decomposition and Harris hawks optimized multivariate wind speed forecasting utilizing sequence2sequence-based spatiotemporal attention”. *Energy*, vol. 278, 127933, Sep. 2023.

[51] Y. Zhang, Y. Li, G. Zhang, “Short-term wind power forecasting approach based on Seq2Seq model using NWP data”. *Energy*, vol. 213, 118371, Dec. 2020.

[52] W. Lee, J. Hong, “A hybrid dynamic and fuzzy time series model for mid-term power load forecasting”. *International Journal of Electrical Power & Energy Systems*, vol. 64, pp. 1057-1062, Jun. 2015.

[53] Y. Akbal, K. D. Ünlü, “A univariate time series methodology based on sequence-to-sequence learning for short to midterm wind power production”. *Renewable Energy*, vol. 200, pp. 832-844, Nov. 2022.

[54] R. A. Sobolewski, M. Tchakorom, R. Couturier, “Gradient boosting-based approach for short- and medium-term wind turbine output power prediction”. *Renewable Energy*, vol. 203, pp. 142-160, Feb. 2023.

[55] B.H. Bailey, S.L. McDonald, D.W. Bernadett, M.J. Markus, K.V. Elsholz, *Wind resource assessment handbook: Fundamentals for conducting a successful monitoring program*. National Renewable Energy Lab, United States, 1997, pp. 1-80.

[56] Z. Jiang, J. Che, M. He, F. Yuan, “A CGRU multi-step wind speed forecasting model based on multi-label specific XGBoost feature selection and secondary decomposition”. *Renewable Energy*, vol. 203, pp. 802-827, Feb. 2023.

[57] S.M. Weekes, A.S. Tomlin, S.B. Vosper, A.K. Skea, M.L. Gallani, J.J. Standen, “Long-term wind resource assessment for small and medium-scale turbines using

operational forecast data and measure–correlate–predict". *Renewable Energy*, vol. 81, pp. 760-769, Sep. 2015.

[58] S. T. Ayele, M. B. Ageze, M. A. Zeleke, T. A. Miliket, "Adama II wind farm long-term power generation forecasting based on machine learning models". *Scientific African*, vol. 21, e01831, Sep. 2023.

[59] T.G. Barbounis, J.B. Theocharis, "Locally recurrent neural networks for long-term wind speed and power prediction". *Neurocomputing*, vol. 69(4-6), pp. 466-496, Jan. 2006.

[60] T. Khatib, R. Deria, A. Isead, "Assessment of Three Learning Machines for Long-Term Prediction of Wind Energy in Palestine". *Mathematical Problems in Engineering*, vol. 2020, pp. 1-11, Oct. 2020.

[61] M. Neshat, M. M. Nezhad, S. Mirjalili, G. Piras, D. A. Garcia, "Quaternion convolutional long short-term memory neural model with an adaptive decomposition method for wind speed forecasting: North aegean islands case studies". *Energy Conversion and Management*, vol. 259, 115590, May 2022.

[62] S.M. Weekes, A.S. Tomlin, "Comparison between the bivariate Weibull probability approach and linear regression for assessment of the long-term wind energy resource using MCP". *Renewable Energy*, vol. 68, pp. 529-539, Aug. 2014.

[63] B. Lim, S. Zohren, (2021, Feb.), "Time-series forecasting with deep learning: a survey". Philosophical Transactions A, vol. 379, 20200209, Feb. 2021.

[64] J. Kaur, K.S. Parmar, S. Singh, "Autoregressive models in environmental forecasting time series: a theoretical and application review". *Environmental Science and Pollution Research*, vol. 30(8), pp. 1-25, Jan. 2003.

[65] G.E.P. Box, G.M. Jenkins, *Time Series Analysis, Forecasting and Control*, 3rd ed. California: Holden-Day, 1970, pp. 165-172.

[66] G.P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model". *Neurocomputing*, vol. 50(17), pp. 159-175, Jan. 2003.

[67] G. Zhang, B.E. Patuwo, Y.H. Michael, "Forecasting with artificial neural networks: The state of art". *International Journal of Forecasting*, vol. 14(1), pp. 35-62, Mar. 1998.

[68] G.P. Zhang, M. Qi, "Neural network forecasting for seasonal and trend time series". *European Journal of Operational Research*, vol. 160(2), pp. 501-514, Feb. 2005.

[69] N. Smaoui, “An artificial neural network noise reduction method for chaotic attractors”. *International Journal of Computer Mathematics*, vol. 73(4), pp. 417–431, Jan. 2000.

[70] J.J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities”. *Proceedings of the National Academy of Sciences of the United States of America*, vol. 79(8), pp. 2554–2558, Apr. 1982.

[71] S. Buhamra, N. Smaoui, M. Gabr, “The Box–Jenkins analysis and neural networks: prediction and time series modelling”. *Applied Mathematical Modelling*, vol. 27(10), pp. 805-815, Oct. 2023.

[72] F.S. Wong, “Time series forecasting using backpropagation neural networks”. *Neurocomputing*, vol. 2(4), pp. 147-159, Jul. 1991.

[73] P. Antwi et al., “Estimation of Biogas and Methane Yields in an UASB Treating Potato Starch, Processing Wastewater With Backpropagation Artificial Neural Network”. *Journal Bioresour Technology*, vol. 228, pp. 106–115, Mar. 2017.

[74] J. Tarigan, Nadia, R. Diedan, and Y. Suryana, “Plate Recognition Using Backpropagation Neural Network and Genetic Algorithm”. *Procedia Computer Science*, vol. 116, pp. 365–372, Jan. 2017.

[75] A. Apicella, F. Donnarumma, F. Isgrò, R. Prevete, “A survey on modern trainable activation functions”. *Neural Networks*, vol. 138(11), pp. 14–32, Feb. 2021.

[76] S. Scardapane, S. Van Vaerenbergh, S. Totaro, A. Uncini, “Kafnets: Kernel-based non-parametric activation functions for neural networks”. *Neural Networks*, vol. 110, pp. 19–32, Jul. 2017.

[77] K. I. Funahashi, “On the approximate realization of continuous mappings by neural networks”. *Neural Networks*, vol. 2(3), pp. 183–192, May 1989.

[78] S. Kiliçarslan, K. Adem, M. Çelik, “An overview of the activation functions used in deep learning algorithms”. *Journal of New Results in Science*, vol. 10(3), pp. 75-88, Dec. 2021.

[79] Y. Bengio, P. Simard, P. Frasconi, “Learning long-term dependencies with gradient descent is difficult”. *IEEE Transactions on Neural Networks*, vol. 5(2), pp. 157–166, Mar. 1994.

[80] G. Lin, W. Shen, “Research on convolutional neural network based on improved Relu piecewise activation function”. *Procedia Computer Science*, vol. 131, pp. 977–984, May 2018.

[81] N. Jinsakul, C.F. Tsai, C.E. Tsai, P. Wu, “Enhancement of deep learning in image classification performance using exception with the swish activation function for colorectal polyp preliminary screening”. *Mathematics*, vol. 7(2), 1170, Dec. 2019.

[82] D. Hendrycks, K. Gimpel, “Gaussian error linear units (GELUs)”. *Computer Science*, vol. 1, Jun. 2016.

[83] M. Lee, “Mathematical Analysis and Performance Evaluation of the GELU Activation Function in Deep Learning”. *Journal of Mathematics*, vol. 2023, pp. 1-13, Aug. 2023.

[84] A. Tokgöz, G. Ünal, "A RNN based time series approach for forecasting Turkish electricity load". in 26th Signal Processing and Communications Applications Conference (SIU), Izmir, Turkey, 2018, pp. 1-4.

[85] A. Sherstinsky, “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network”. *Physica D: Nonlinear Phenomena*, vol. 404, 132306, Mar. 2020.

[86] G. Keren, B. Schuller, "Convolutional RNN: An enhanced model for extracting features from sequential data". in International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 2016, pp. 3412-3419.

[87] A.F. Ganai, F. Khursheed, "Predicting next Word using RNN and LSTM cells: Stastical Language Modeling". in Fifth International Conference on Image Information Processing (ICIIP), Shimla, India, 2019, pp. 469-474.

[88] V. Prema, S. Sarkar, K.U. Rao, A. Umesh, “LSTM based deep learning model for accurate wind speed prediction”. *ICTACT Journal On Data Science And Machine Learning*, vol. 1(1), pp. 6–11, Dec. 2019.

[89] W. Liu, Y. Bai, X. Yue, R. Wang, Q. Song, “A wind speed forecasting model based on rime optimization based VMD and multi-headed self-attention-LSTM”. *Energy*, vol. 294, May 2024.

[90] A.G. Salman, Y. Heryadi, E. Abdurahman, W. Suparta, “Weather Forecasting Using Merged Long Short-Term Memory Model (LSTM) and Autoregressive Integrated Moving Average (ARIMA) Model”. *Journal of Computer Science*, vol. 14(7), pp. 930-938, Jul. 2018.

[91] L. Wu, C. Kong, X. Hao, W. Chen, “A Short-Term Load Forecasting Method Based on GRU-CNN Hybrid Neural Network Model”. *Mathematical Problems in Engineering*, vol. 2020, pp. 1-10, Mar. 2020.

[92] X. Liu, Z. Lin, Z. Feng, “Short-term offshore wind speed forecast by seasonal ARIMA - A comparison against GRU and LSTM”. *Energy*, vol. 227, 120492, Mar. 2021.

[93] X. Gao, X. Li, B. Zhao, W. Ji, X. Jing, Y. He, “Short-term electricity load forecasting model based on EMD-GRU with feature selection”. *Energies*, vol. 12(6), 1140, Mar. 2019.

[94] C. Li, G. Tang, X. Xue, A. Saeed, X. Hu, “Short-term wind speed interval prediction based on ensemble GRU model”. *IEEE Transactions on Sustainable Energy*, vol. 11(3), pp. 1370- 1380, Jul. 2020.

[95] L. Chuang, L. Guojie, W. Keyou, H. Bei, “A multi-energy load forecasting method based on parallel architecture CNN-GRU and transfer learning for data deficient integrated energy systems”. *Energy*, vol. 259, 124967, Nov. 2022.

[96] I. Livieris, E. Pintelas, P. Pintelas, “A CNN-LSTM model for gold price time series forecasting”. *Neural Computing and Applications*, vol. 32(5), pp. 17351–17360, Dec. 2020.

[97] I. Koprinska, D. Wu, Z. Wang, "Convolutional Neural Networks for Energy Time Series Forecasting". in International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 2018, pp. 1-8.

[98] Q.V. Le, "A Tutorial on Deep Learning Part 2: Autoencoders, Convolutional Neural Networks and Recurrent Neural Networks". *Google Brain*, vol. 20, pp. 1-20, Oct. 2015.

[99] A. Wan, Q. Chang, K.A. Bukhaiti, J. He, (2023, Jun.), “Short-term power load forecasting for combined heat and power using CNN-LSTM enhanced by attention mechanism”. *Energy*, vol. 282, 128274, Jun. 2023.

[100] G. Zhang, X. Bai, Y. Wang, “Short-time multi-energy load forecasting method based on CNN-Seq2Seq model with attention mechanism”. *Machine Learning with Applications*, vol. 5(2), 100064, Jun. 2021.

[101] S. Xu, Y. Wang, X. Xu, G. Shi, Y. Zheng, H. Huang, C. Hong, “A multi-step wind power group forecasting seq2seq architecture with spatial–temporal feature fusion and numerical weather prediction correction”. *Energy*, vol. 291, 130352, Mar. 2024.

[102] Z. Masood, R. Gantassi, A. Ardiansyah, Y. Choi, “A Multi-Step Time-Series Clustering-Based Seq2Seq LSTM Learning for a Single Household Electricity Load Forecasting”. *Energies*, vol. 15(7), 2623, Apr. 2022.

[103] G. Gong, X. An, N. Mahato, S. Sun, S. Chen, Y. Wen, “Research on Short-Term Load Prediction Based on Seq2seq Model”. *Energies*, vol. 12(16), 3199, Aug. 2019.

[104] “ACAI '19: Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence “, 2019, pp. 49–55.

[105] K. Setiawan, G. Elwirehardja, B. Pardamean, “Comparison of deep learning sequence-to-sequence models in predicting indoor temperature and humidity in solar dryer dome”. *Communications in Mathematical Biology and Neuroscience*, vol. 2022(98), pp. 1-26., Oct. 2022.