Final Year Project Report

Deep Learning-Based Wind Speed Prediction for Optimized Wind Turbine Operation

Sofiya Alimukhambetova, Ayana Kochkarova

A thesis submitted in part fulfilment of the degree of

BSc in Robotics and Mechatronics

Supervisor: Berdakh Abibullaev, Ton Duc Do

Department of Robotics and Mechatronics
Nazarbayev University

May 2, 2024

Table of Contents

Abstract

Wind energy has been a promising source of clean energy that does not negatively affect our environment. Because of the fluctuations in wind speed, it is crucial to predict its values for wind turbines to have the maximum effective power output. This project aims to develop a way for short-term wind speed prediction based on deep learning technologies, such as CNN, LSTM, RNN, and GRU models, alone and in combination. Through iterative experimentation and evaluation, we develop ten final models and assess their performance based on Mean Squared Error (MSE), score, and computational efficiency. Our findings reveal that the GRU model achieves the highest performance with a MSE of 0.00238 m/s and R^2 score of 0.8796. Additionaly, the similarly structured LSTM model demonstrates superior computational efficiency along with high R^2 value, outperforming GRU model. By examining the performance of multiple deep learning architectures, the project seeks to identify the most suitable approach for wind speed prediction, thereby facilitating more efficient and sustainable utilization of wind resources for power generation.

Chapter 1: **Introduction**

Due to the depleting of fossil fuels such as coal, oil, gas and their harmful impact on the environment, researchers have been searching and implementing more eco-friendly technologies for clean energy production. Wind has been a promising resource of energy because of its availability almost everywhere, zero greenhouse gas emissions, and low operational costs after wind turbine installation. However, to effectively generate wind energy, it is crucial to maintain proper prediction of wind speed. Given the unpredictable nature of wind, accurate wind speed forecasting is essential for optimizing the efficiency and reliability of wind energy generation [1]. Traditional statistical models have long been employed for wind speed prediction. These models, including autoregressive models, moving average models, and physical models, have demonstrated effectiveness in capturing temporal patterns and making predictions based on historical data. In recent years, advancements in deep learning techniques, including Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, Gated Recurrent Units (GRUs), and Recurrent Neural Networks (RNNs), have shown promise in addressing the challenges of wind speed prediction.

CNNs, originally designed for image recognition tasks, have been adapted for time-series forecasting by exploiting their ability to automatically extract relevant features from sequential data. LSTM networks, renowned for their capability to capture long-range dependencies and handle sequential data, offer a powerful framework for modeling the temporal dynamics of wind speed fluctuations. Gated Recurrent Unit (GRU) networks, similar to LSTM networks, are well-suited for modeling sequential data and have been widely used in time-series forecasting tasks. GRUs offer a simplified architecture compared to LSTMs, with fewer parameters and computational overhead, making them more computationally efficient. Recurrent Neural Networks (RNNs), the foundational architecture for both LSTM and GRU networks, are inherently designed to handle sequential data and have been extensively applied in time-series forecasting tasks.

Moreover, CNNs are ofthen paired with RNN-like models (RNN, GRU, LSTM) to extract spatial and temporal features from sequential data, resulting in improved performance. These hybrid models, whether CNN-LSTM, CNN-GRU, or CNN-RNN, offer a comprehensive framework for capturing both short-term patterns and long-range dependencies in time-series data.

By integrating these deep learning architectures into wind speed forecasting systems, this research tends to enhance the accuracy and reliability of wind energy generation. Through the application of CNNs, LSTMs, GRUs, RNNs and its combinations, this project aims to contribute to the advancement of renewable energy technologies, fostering a more sustainable and environmentally friendly future.

Chapter 2: **Related Work**

Effective forecasting of wind speed is an essential tool for optimizing wind turbine operations. With the burgeoning significance of renewable energy sources, accurate prediction methodologies have garnered increasing attention. There is a great variety of approaches for effective wind speed and wind power prediction. Each methodology presents distinct advantages and challenges, necessitating a thorough exploration of their applicability within the context of wind energy optimization. The process of prediction usually utilizes traditional or deep-learning based techniques.

2.1 Traditional models

The majority of traditional approaches for wind speed and power prediction involves using physical models and traditional statistical tools. Physical based tools like numerical weather prediction (NWP) is useful to predict wind speed with available meteorological and physical data (e.g., temperature and air pressure). Physical approaches are usually used in exact locations where the wind turbine is installed [2]. The flowchart of how physical models work can be seen in Figure 2.1.

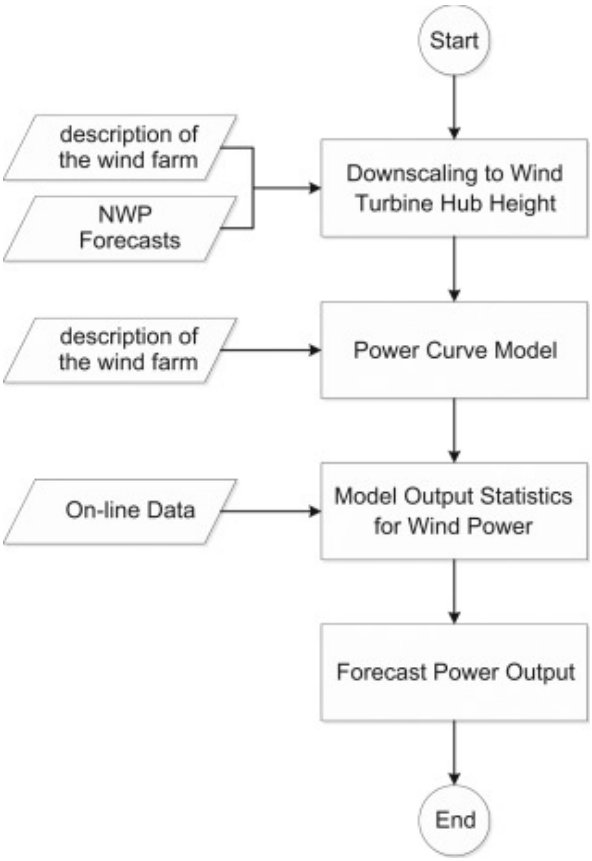

Figure 2.1: The physical approach to forecasting wind speed and power. Adapted from [2]

For example, in early studies [3] the downscailing of NWP by using mesoscale and Computational Fluid Dynamics (CFD) models it was possible to significantly reduce the error and processing time in wind speed prediction.

Another type of traditional tools used in wind speed prediction is the utilisation of time series statistical models, such as autoregressive integrated moving average model (ARIMA) and its components (autoregressive model (AR), moving average model (MA), autoregressive moving average model (ARMA)). The structure of the simple statistical model used for wind speed prediction can be viewed in Figure 2.2.These models adjust their parameters by analyzing historical data to assess whether the discrepancy between the model's output and the actual observations follows a random walk pattern. Such statistical techniques are used for forecasting wind speeds and providing the stability and linearity of output [4].

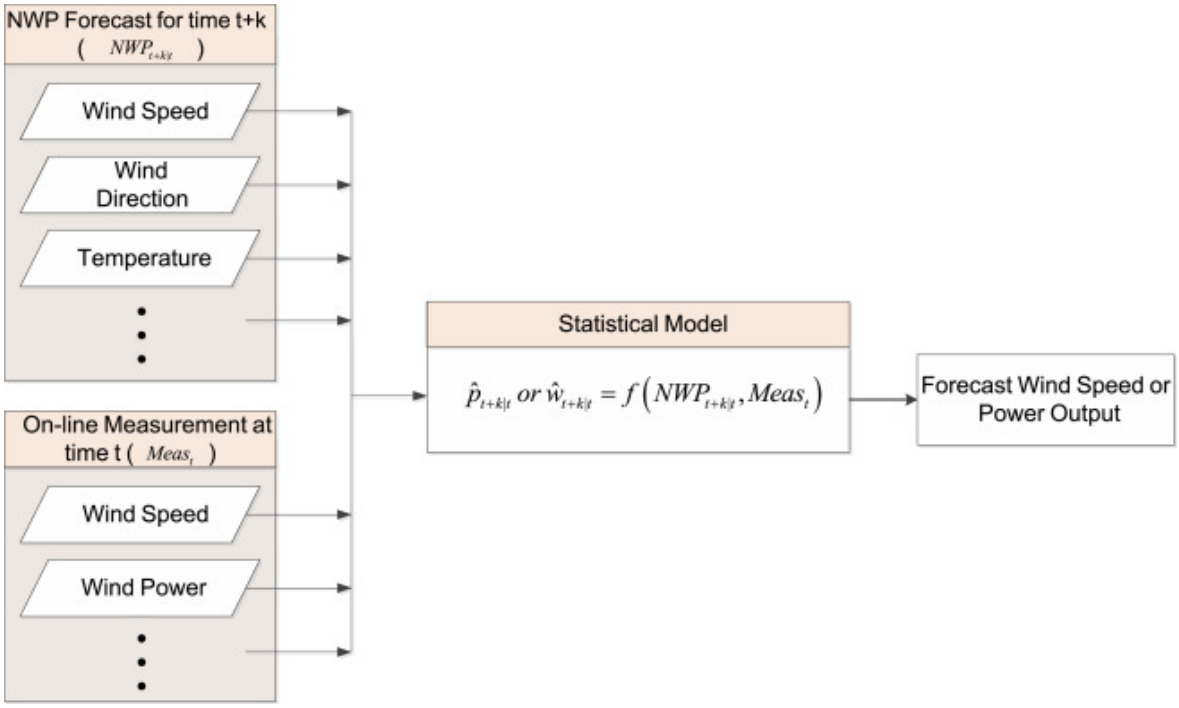

Figure 2.2: The statistical approach to forecasting wind speed and power. Adapted from [2]

For example, in [5], authors used ARIMA to model 10-min average wind speed time-series data obtained from several wind turbine plants. The results of this study showed a high level of effectiveness of the modified ARIMA model in accurately representing the wind speed characteristics and indicating that a model developed at one location can be effectively applied to another location within the same geographical area.

Overall, physical and statistical models have been fundamental tools in wind speed prediction, offering distinct advantages. Physical models provide insights into the underlying physics of wind flow and can capture large-scale atmospheric patterns effectively. Additionally, they offer the potential for long-term forecasting and can simulate the impact of environmental changes on wind patterns. Statistical models, on the other hand, rely on historical data and mathematical algorithms to identify patterns and relationships between meteorological parameters. They are often computationally efficient and can provide relatively simple predictions.

However, despite their strengths, physical and statistical models come with several limitations. Physical models require extensive computational resources and detailed input data, making them challenging to implement in real-time applications. Additionally, the simplifications and assumptions used in physical models may lead to inaccuracies, particularly in

complex terrain. Statistical models, while more accessible and computationally efficient, may struggle to capture nonlinear relationships and complex interactions between meteorological variables. They are also limited by the availability and quality of historical data, which can affect their predictive accuracy, especially in regions with sparse or unreliable data.

While physical and statistical models have played a crucial role in wind speed prediction, their limitations have spurred the exploration of alternative approaches, such as deep learning-based models.

2.2 Deep learning-based models

Deep learning techniques, including CNNs, LSTMs, RNNs, and GRUs, offer promising avenues for overcoming the challenges faced by traditional methods. The variety of studies demonstrates the usage of only selected deep-learning models, its combinations, and other techniques to achieve the most effective results. The generalized structure of layers in CNN, LSTM, RNN, GRU models designed for wind speed and power prediction is illustrated in Figure 2.3.

The most important tool in deep-learning, artificial neural network (ANN) architecture, comprises three main layers: the input layer, one or more hidden layers, and the output layer. Within each layer, numerous artificial neurons are interconnected, following a connectionist paradigm where neurons are linked to those in the preceding layer. Through a training and learning phase, this architecture can effectively capture intricate nonlinear relationships between input and output layers. Unlike the physical and statistical methods discussed earlier, ANNs do not rely on explicit mathematical expressions. Additionally, ANNs possess self-learning, self-organizing, and self-adapting capabilities, allowing them to autonomously adjust and refine their internal representations based on the data they encounter.

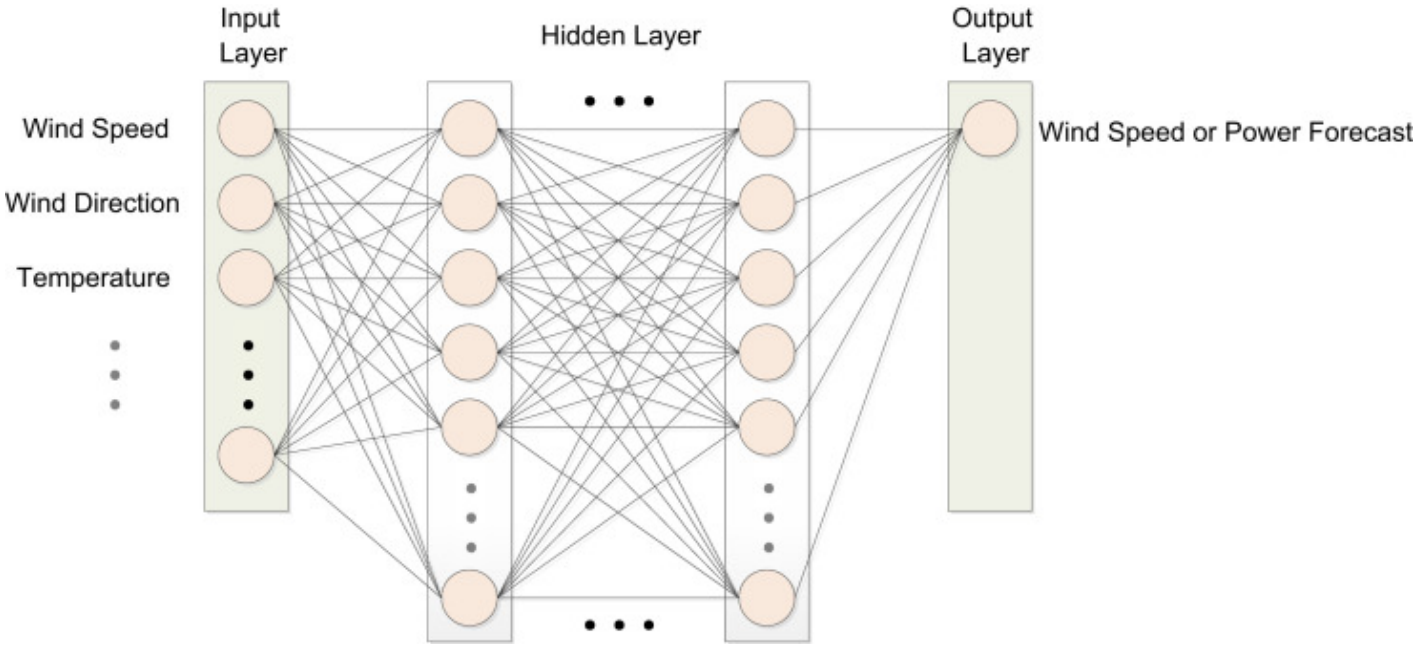

Figure 2.3: ANN approach for wind speed and power forecast. Adapted from [2]

For example, the combination of CNNs, multiple stacked bidirectional long short-term memory (Bi-LSTM) networks, and the attention mechanism [6]. In this study, CNN is used for

extracting multidimensional wind features via convolution and pooling operations, Bi-LSTM for managing the fusion of sequential features, and an attention mechanism for generating masks to extract crucial features and predict future wind power points. Evaluation with a real-world wind power dataset demonstrates the method's effectiveness compared to traditional models. The same architecture is used in [7], where the effectiveness of the proposed model is additionally verified with the historical data, suggesting that the hybrid combination can help fit the peak power output more accurately than the basic LSTM, GRU, and the auto-encoder networks. Another hybrid research [8] with CNN and gated recurrent neural network (GRU) introduces a transfer learning-based time series prediction method for short-term canyon wind speed forecasting during hydropower station construction, addressing data scarcity in early stages. This approach leverages historical wind speed and temperature data for accurate predictions, demonstrating nearly a 20% improvement in MAE and RMSE. The overall approach for CNNs based techniques for wind speed prediction can be viewed in Figure 2.4.

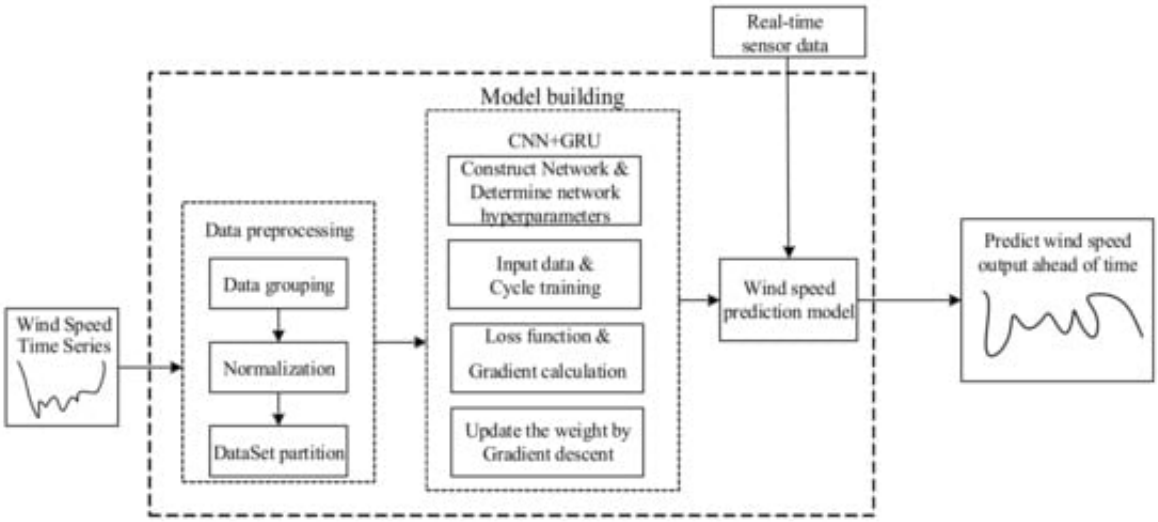

Figure 2.4: The framework of the wind speed prediction model. Adapted from [8]

Another study [9] with the usage of LSTM introduces a forecasting method to improve wind speed prediction for wind power integration into the grid. Analyzing 15-minute wind speed data from four wind farms, the results demonstrate the algorithm's superiority over the ARIMA tool in terms of root mean square error, mean absolute error, and mean absolute percentage error. The findings highlight the potential of this deep learning approach for intelligent grid programming and its support in shaping wind power industry policies. The improved version [10] with the additional usage of Principal Component Analysis (PCA) utilized for data dimension reduction, addressing the challenges posed by the randomness and volatility of wind energy, reveals the effectiveness of the proposed PCA-LSTM method in improving wind power prediction accuracy.

The combination of LSTM and gated recurrent unit (GRU) models [11] also reveals the efficiency in wind speed prediction. The research emphasizes the effectiveness of machine learning models in predicting wind power values and suggests their application in unknown geographical areas for wind power plant installations based on reasonable location models. Also it highlights that GRU significantly outperforms a simple LSTM across all metrics and forecasting horizons, showcasing the adaptability and versatility of these models for precise wind speed and wind power forecasting, thereby promoting sustainable and cost-effective wind energy. Further research supports that GRU models effectively handle the temporal dynamics critical for managing the intermittent nature of wind [12]. It demonstrates that GRUs are not only comparable but often superior to other deep learning models in predicting complex wind speed patterns, thereby contributing to more accurate and reliable energy

generation forecasts. This underscores the practical benefits of integrating GRU models into wind energy systems to enhance operational planning and increase efficiency. Notably, a study [13] provides concrete statistical evidence illustrated in Figure 2.5, showing that the GRU model achieves lower prediction errors with a MAPE of 9.517233% and RMSE of 0.363026 for wind farm A, and a MAPE of 16.50382% and RMSE of 0.413986 for wind farm B. This indicates the model's higher accuracy even in environments with fluctuating wind speeds, where the GRU model consistently outperformed ARMA and LSTM models in predictive accuracy.

Unit	**Prediction Error**	**Predictive Model**		
		ARMA	***LSTM***	***GRU***
unit A	MAPE (%)	11.1813	10.87622	9.517233
	RMSE	0.39527	0.383883	0.363026
unit B	MAPE (%)	19.1851	17.46326	16.50382
	RMSE	0.428445	0.428255	0.413986

Figure 2.5: Error results of three prediction models. Adapted from [13]

Another approach of using deep learning in this field is to apply Recurrent Neural Network models (RNN). RNNs have shown significant promise in improving the accuracy of wind speed forecasting, a critical factor in the efficient operation of wind turbines. The ability of RNNs to capture and utilize the sequential nature of wind speed data effectively allows for more precise predictions, which in turn can significantly enhance the operational efficiency of wind power generation [14]. Also RNN models, with their nonlinear processing capabilities, provide more accurate forecasts compared to traditional linear models like ARIMA [15]. The prediction accuracy of these models are illustrated in the Figure 2.6.

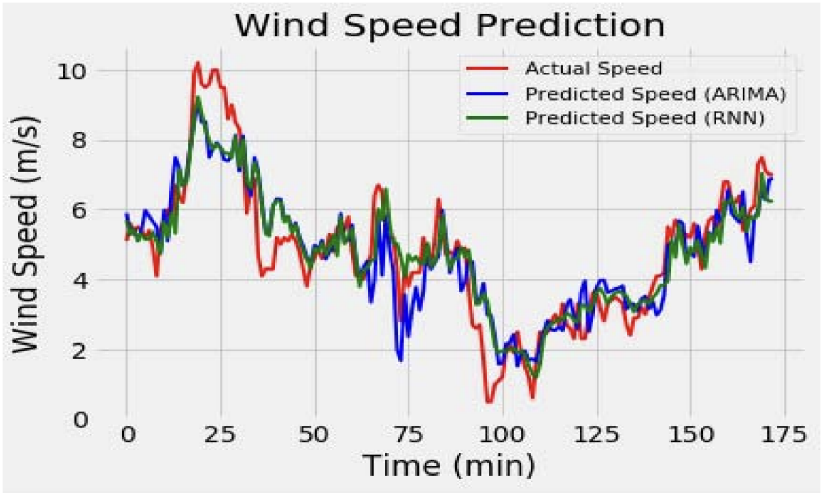

Figure 2.6: Prediction plots of ARIMA and RNN. Adapted from [15]

The Figure 2.7 shows that RNN provides an 18.18% improvement in Mean Absolute Error (MAE) and a reduction in Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) by 20 to 25% and 11 to 14% respectively, compared to ARIMA models.

Parameters	ARIMA		RNN	
	Dataset A	Dataset B	Dataset A	Dataset B
MAE	1.10	0.77	0.90	0.63
MSE	1.63	0.92	1.29	0.69
RMSE	1.27	0.96	1.13	0.83
Time (sec.)	1.41	2.51	36.33	833.79

Figure 2.7: Error result of ARIMA and RNN. Adapted from [15]

The integration of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) units into wind speed and power forecasting represents a significant advancement in renewable energy management. The CNN-LSTM framework demonstrates a reduction of MAE by up to 32.7%, marking a substantial improvement over traditional forecasting methods [16]. The enhancement of predictive capability through such a combination is corroborated by another related study [17], which also employs an unsupervised algorithm for data preprocessing. The Figure 2.8 demonstrates such prediction framework. This approach effectively reduces redundancy and accelerates convergence, further confirming the efficacy of the combined model.

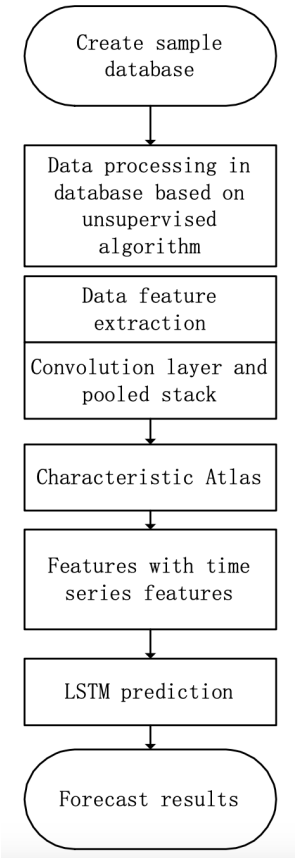

Figure 2.8: CNN-LSTM prediction framework. Adapted from [17]

The combination of CNN and GRU allows the model to effectively handle the complex nature of wind speed dynamics which are influenced by various atmospheric conditions over time [18]. This model employs convolutional layers to extract spatial features from sequential input data, and GRUs to capture temporal dependencies within the time series data of wind speed. The Figure 2.9 illustrates the superior performance of CNN-GRU model. Therefore, it can be concluded that predictive capability of this model shows a significant improvement over other traditional and deep learning models.

	Forecasting methods	MAPE, %	MAE, m/s	RMSE, m/s	NRMSE
day-ahead	CNN-GRU	5.459	0.155	0.261	0.029
	CNN-LSTM	6.633	0.187	0.296	0.033
	2D-CNN	8.683	0.235	0.336	0.038
	GRU	9.099	0.229	0.341	0.039
	LSTM	9.670	0.260	0.361	0.041
	SVM	13.303	0.378	0.510	0.058
	KNN	20.206	0.700	0.960	0.110
real-time	CNN-GRU	6.121	0.18	0.284	0.028
	CNN-LSTM	9.765	0.341	0.543	0.054
	2D-CNN	10.025	0.367	0.557	0.055
	GRU	9.912	0.357	0.554	0.055
	LSTM	10.161	0.378	0.574	0.057
	SVM	14.989	0.590	0.819	0.081
	KNN	21.955	0.648	0.893	0.089

Figure 2.9: Performance of wind speed prediction models. Adapted from [18]

Another combination of CNNs for spatial feature extraction with RNNs for capturing temporal dynamics highlights the effectiveness in improving the accuracy and reliability of wind power forecasting models [19]. Such architecture effectively integrates spatial and temporal information by using CNNs to capture spatial details across multiple numerical weather prediction models and geographical locations, and RNNs to assimilate temporal sequences from historical power data of wind farms. The proposed approach is summarized in Figure 2.10.

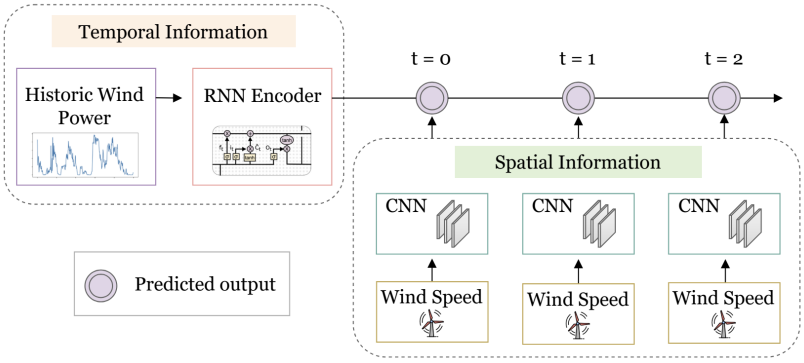

Figure 2.10: Performance of wind speed prediction models. Adapted from [19]

However, it is noted that when the CNN-RNN model is trained individually on data from each wind farm, it does not maintain the same level of high performance, suggesting that the model benefits from a richer, more varied dataset provided by global training. This underscores the model's enhanced ability to generalize across diverse meteorological and geographical conditions, leveraging the broad dataset to better capture the complexities of wind dynamics and power output predictions.

Overall, deep learning techniques have revolutionized wind speed prediction by offering more accurate, flexible, and adaptable forecasting solutions. These techniques have the potential to significantly enhance the efficiency and performance of wind energy systems, ultimately contributing to the advancement of renewable energy technologies and sustainability efforts.

Chapter 3: **Dataset Specifications**

3.1 Dataset Overview

This study applies actual data that covers the recorded wind speed and power output from wind farm located in the Central of Vietnam, Binh Thuan province, the southernmost coastal province in the South Central Coast region. The time period of data collections ranges from February 1, 2022, at 0:00 to December 31, 2022, at 23:30, encompassing a total of 16,032 wind speed and 16,032 power output observations for the year 2022. Each observation encapsulates the wind speed and power generation data of a wind turbine every 30 minutes. The time series of all wind speed data points can be seen in Figure 3.1.

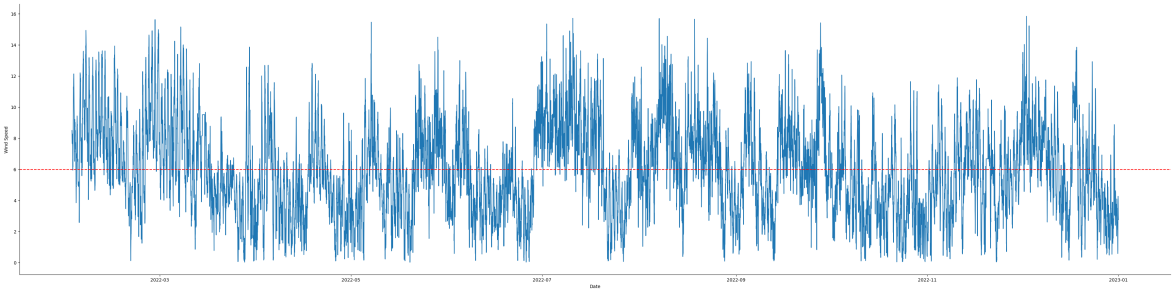

Figure 3.1: The time series of all data points

As depicted in 3.1, the wind speed clearly shows its unpredictable and variable nature without any particular seasonal trends. The average wind speed calculated stands at 5.98 m/s, indicated by the red line on the graph. Wind speed values ranges from minimum at 0.02 m/s to a peak of 15.83 m/s. The standard deviation value was calculated as 2.94 m/s. These parameters provide essential insights for analyzing the characteristics of the wind speed series.

Frequency histogram of wind speed depicted in Figure 3.2 reveals a distribution that closely resembles a normal distribution. This observation suggests that there are no significant outliers in the dataset, as the majority of wind speed values fall within expected ranges without extreme deviations. The histogram provides a visual representation of the probability of different wind speeds, affirming the consistency and reliability of the data collected.

To observe the relationship between collected wind speed and power output values, the graph presented in Figure 3.3 was constructed, along with highlighting the critical thresholds of cut-in and cut-out speeds. As can be seen in the beginning of the graph, cut-in speed, which represents the minimum wind speed required for the wind turbine to begin generating power efficiently, starts at approximately 3 m/s. In practical terms, this threshold indicates the point at which the turbine's rotor blades start rotating and the generator begins producing electricity. Conversely, the cut-out speed, which signifies the maximum wind speed at which the turbine limits it electricity production rate to prevent damage from excessively high winds, peaks at approximately 12-14 m/s. Therefore, there might be some sort of damping mechanism, which kicks in at the cut-out speed, slowing down the spinning blades.

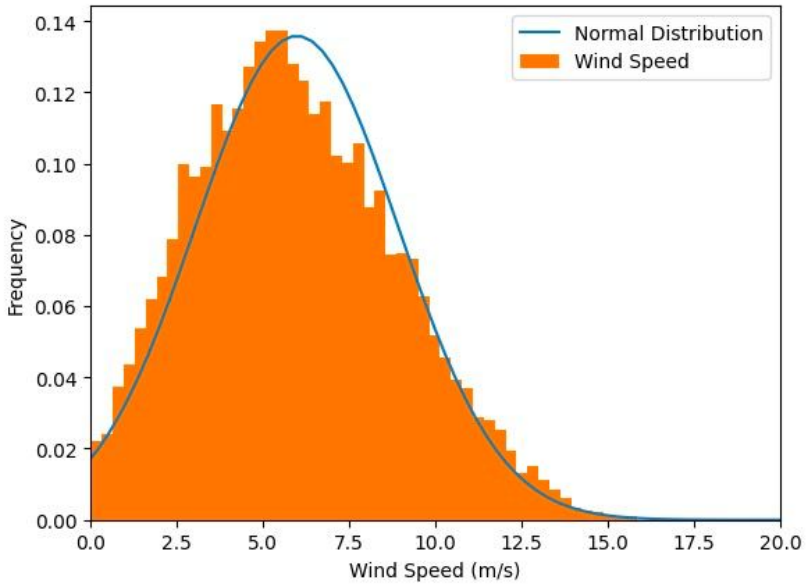

Figure 3.2: The frequency histogram of wind speed

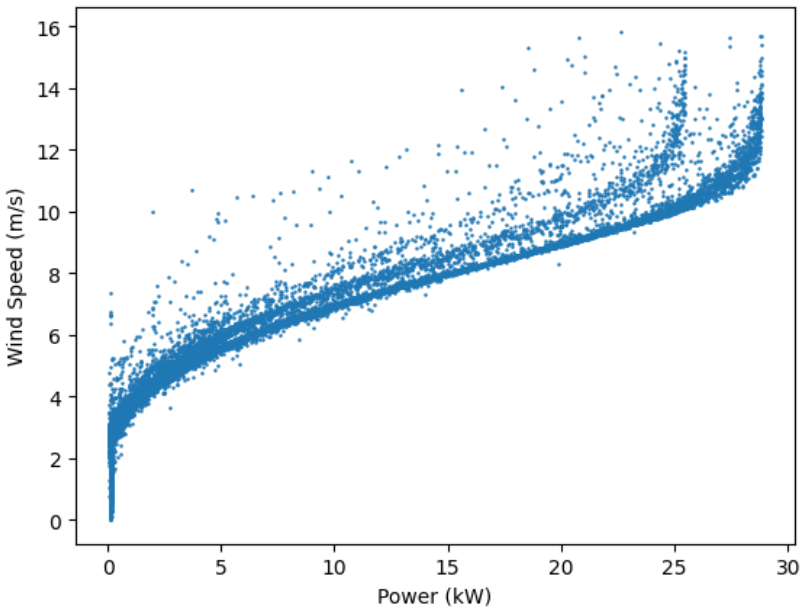

Figure 3.3: Power vs. Wind speed curve with the raw data set

3.2 Data Manipulations

3.2.1 Data Augmentation

Despite the substantial dataset, the variability in meteorological conditions necessitates the use of data augmentation to adequately model and forecast wind speeds under diverse environmental conditions. Data augmentation addresses the limitations posed by the finite size of the dataset and helps in modeling rare but critical events such as sudden gusts or lulls in wind speed, which are pivotal for optimizing turbine performance.

Among the various techniques for data augmentation, jittering, which introduces small, random fluctuations into the dataset, proves particularly effective for time series data like wind

speed measurements. This technique simulates the natural variability in wind speed due to environmental factors, thereby enhancing the model's ability to generalize across different conditions. Noise jittering is instrumental in enriching the feature space in forecasting models, which is crucial for handling the inherent variabilities in wind speed predictions [20].Therefore, significantly enhances the predictive accuracy of models for wind speed forecasting [21].

Implementing the jittering technique increased dataset sixfold from the original 16,032 observations to 96,192 observations. This extensive augmentation not only diversifies the training data but also prevents the model from overfitting, thereby improving the robustness and generalization capability of the model across varied meteorological conditions. The expanded dataset now provides a comprehensive basis for understanding fine-grained diurnal and seasonal patterns in wind behavior and turbine performance, ensuring that the model is well-equipped to predict under a broad range of meteorological conditions.

However, it is worth to mention, that this data augmentation technique was applied to the dataset only for final training of the selected deep-learning models architectures. The process of the best architectures selection, explained in the following chapter, involved training of a variety of CNN, RNN, LSTM, and GRU models with different parameters on raw dataset to minimize the training time due to the high number of the models.

3.2.2 Data preprocessing

Mean and standard deviation normalization is used to improve the performance and stability of deep-learning models by standardizing the scale and distribution of the input data. Often referred to as Z-score normalization, it aims to transform the data distribution to have a mean of 0 and a standard deviation of 1. The normalization of the dataset is done by using the following formula:

$$X_{normalized} = \frac{X_i - X_{mean}}{X_{stdev}}$$

X_i is the original data, $X_{normalized}$ is the normalized data, X_{mean} is the mean of the dataset, and X_{stdev} is the standard deviation of the dataset.

Before model training, the data should be organized in sequences to be transferred into first layer and then be analyzed for temporal dependencies and patterns. Therefore, in our project the look back period of 48 data points was implemented for all models. Since our data points separated by 30 minutes time windows, look back period of 48 points indicates prediction based on the previous 24 hours (1 day). This ensures that the model can learn from past observations and leverage this information to make accurate predictions about future wind speeds.

Chapter 4: **Methodology**

4.1 Best architecture selection

During this important step, our focus lies on meticulously exploring various architectural configurations for CNN, RNN, LSTM, and GRU models. For each model type, a predefined grid of hyperparameters is specified, encompassing ranges for parameters such as the number of convolutional layers, filters, kernel sizes (for CNN), as well as number of features, stacked layers, and hidden sizes (for RNN, LSTM, and GRU). To generate different combinations of architecture parameters, the grid search method iterates over chosen combinations within the specified parameter ranges. Each unique parameter configuration is then used to train a corresponding model on the dataset. During the model architectures selection step, the data split was 85% for training the model, and 15% for testing/validation. Looking at each model:

4.1.1 CNN model architectures

The variety of CNN model architectures was implemented by changing the number of convolutional layers, number of filters, and kernel size. Convolutional layers apply convolution operations to input data, allowing the network to learn spatial features. For our variations, 1 to 4 layers were used for each model. Within each convolutional layer, there are filters, small learnable matrices, that specialize in detecting specific patterns or features. In our setup, we use most common 16, 32, 64, 128 number of filters. Regarding the kernel size, it refers to the dimensions of the filters applied during the convolution operation. The kernel sizes of 3, 5, and 7 were used during the selection process. After each convolutional layer, Batch normalization layer was added to normalize the activations of each layer within mini-batches during training. Moreover, at the end of every CNN model, the Adaptive Average pooling layer was added to reduce the spatial dimensions of the input feature map from previous layers, followed by flattening this map into a one-dimensional tensor, and finally applying a linear transformation to produce the output prediction.

Additionally, Xavier initialization, also known as Glorot initialization, was added for initializing the weights of our CNN models to ensure stable and efficient training. This helpful technique sets th initial weights so that the activations and gradients during training are neither too large nor too small.

Overall, the number of all variations of CNN models achieved 4303 differents models. Each of them was trained for 5 epochs. After the whole process of training, one model was chosed based on the best performance.

4.1.2 RNN, LSTM, GRU model architectures

Same as CNN, other models were created with different combinations of layers and features. Since RNN, LSTM, and GRU all have similar nature, the variation of parameters and selection process involved all three types of models at the same time. For each model architecture we

used RNN, LSTM, or GRU layers as basic layers for feature extraction. The number of layers varies from 1 to 4. However, only 1- and 2-layered architectures were chosen for training due to very long processing time of deeper models. Within this layers we specify whether we want to use only one of features (wind speed and power) or both. Moreover, we also chose the number of hidden state vector in each recurrent unit, i.e. hidden size. This parameter determines the capacity and memory of the model to capture temporal dependencies in sequential data. It provides a trade-off between model complexity, computational resources, and the ability to learn from past observations effectively. In our setup, the hidden sizes of 128, 64, 32 neurons were used in different iterations. The final layer in each model was implemented as linear layer that performs the final transformation to produce the output prediction based on the extracted features.

Overall, 20 different RNN, LSTM, and GRU models were trained to find the best ones in terms of performance. Each of them was trained for 20 epochs. Among them, 6 best performed models (2 per each type, differ in number of features) were chosen for future training and evaluation.

4.2 Hyper parameters and Evaluation metrics

As an activation function, the Rectified Linear Unit (ReLU) was used for every model. This function is defined as:

$$f(x) = max(0, x)$$

The ReLU function is instrumental in introducing non-linearity across the model's layers, thereby enabling the intricate learning of complex patterns present within the input data [22]

In this project, a learning rate (lr) of 0.001 has been consistently implemented across all models utilized. The choice of this learning rate is crucial in optimizing the training process, as it determines the magnitude of parameter updates during the backpropagation phase. According to the previous research, a learning rate of 0.001 is commonly selected, allowing for sufficient exploration of the parameter space and ensuring stable convergence during training.

To encapsulate the model's learning process, it is compiled using the Adam optimizer, a decision motivated by Adam's adaptability in managing sparse gradients and its proficiency in navigating the stochastic nature of wind speed data [23].

As an evaluation metric, we eployed the Mean Squared Error (MSE) loss function, which expressed as

$$MSE(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

which is particularly suited for regression tasks, offering a straightforward quantification of the model's prediction accuracy by measuring the average squared difference between the estimated values and the actual value [24].

Another metric used to evaluate the performance of the models was a R-squared (R^2) measure

to determine how well the models' predictions align with the actual values of the target variable. This metric can be expressed as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4.1}$$

Another metric on which the performance evaluation was based included time duration of training. For each model, average time per epoch (sec) was recorded along with the time to best epoch, when model converges.

The use of ReLU for non-linear activations and a linear activation for the output layers underpins the model's capacity to accurately forecast wind speeds. The employment of the Adam optimizer and MSE loss function serves to refine the model's weights during the training phase, optimizing its predictive performance [25].

4.3 Experimental Setup

All models were developed in Jupyter Notebook with Python version 3.12.2 and PyTorch 2.2.1. The models were trained and evaluated on Apple M2 Pro SoC with 12 core CPU and 19 core GPU, 16 GB of unified memory, shared between GPU and CPU. The computations were accelerated using PyTorch MPS backend, optimized for Apple Metal framework, an alternative to Nvidia CUDA.

4.4 Training process of selected models

As during the model architectures selection step, the data split remains 85% for training the model, while the other 15% is reserved for testing, thereby balancing the need for a substantial training dataset with the necessity of validating the model's predictive accuracy on unseen data.

All 7 chosen models, along with their combinations, which are CNN-RNN, CNN-LSTM, and CNN-GRU were trained for maximum of 200 epochs, using early stopping technique.

Early stopping is a regularization technique commonly used to prevent overfitting and improve model generalization. During the training, it monitors the performance of the model and stops training when the performance starts to degrade.

Chapter 5: **Results and Discussion**

5.1 Best architecture selection

As mentioned previously, a variety of model architectures were trained to decide on which model architecture to choose and train further.

n	L_1	L_2	L_3	Val Loss	Best R^2	Train Loss	Train R^2
2	(128, 3)	(128, 3)		0.0003	0.2997	0.0000	0.4055
2	(64, 3)	(128, 3)		0.0003	0.2990	0.0000	0.3750
2	(128, 3)	(64, 5)		0.0003	0.2945	0.0000	0.4062
3	(16, 3)	(128, 3)	(128, 3)	0.0003	0.2905	0.0000	0.3974
3	(16, 3)	(64, 3)	(64, 3)	0.0003	0.2885	0.0000	0.3380
3	(16, 5)	(128, 7)	(16, 3)	00.0009	-1.6322	0.0001	-0.1226
4	(16, 3)	(64, 5)	(32, 7)	0.0010	-1.7804	0.0001	0.2397
3	(32, 3)	(128, 7)	(16, 3)	0.0012	-2.2415	0.0001	0.1097
4	(16, 3)	(16, 7)	(128, 7)	0.0013	-2.5550	0.0001	0.3017
4	(16, 3)	(128, 5)	(128, 7)	0.0014	-2.8031	0.0001	0.2201

Table 5.1: CNN models training

After training CNN models for 5 epochs, all their variations demonstrated low performance. The best CNN model achieved R^2 score of 0.2997 on validation dataset. The table demonstrates the tendency for higher performance to be achieved by shallower models with 2 layers, a higher number of CNN filters, and lower kernel sizes of 3 or 5.

type	n_features	hidden_size	num_layers	min_val_loss	max_val_r2	min_train_loss	max_train_r2
str	i64	i64	i64	f64	f64	f64	f64
"RNN"	1	128	2	0.000012	0.839904	0.000002	0.850844
"GRU"	2	128	1	0.000017	0.76228	0.000003	0.77981
"LSTM"	2	128	2	0.000023	0.689486	0.000005	0.693415

Figure 5.1: Best RNN-like model in each class

Further, the proposed RNN networks trained on the same data produced much higher results compared to CNN models. The best R^2 scores of RNN, GRU, LSTM model are 0.8399, 0.7623 and 0.6895 respectively. Higher performance of RNN models is explained by the nature of data, which consists of historical recordings of the wind speed.

These intermediate results provided an insight on which models tend to perform better on the given data.

5.2 Selected models and its combinations

After selecting the best architectures and modelling their combinations, such as 2-layered CNN followed by RNN, the following results were achieved.

As expected from the architecture selection stage, RNN models demonstrated the best performance after training them for 200 epochs on the augmented data. Furthermore, RNN models converged significantly faster than CNN models at around 15th epochs. However, all RNN models began overfitting after 100th epoch, as it can be seen from the figure 5.2 and 5.3.

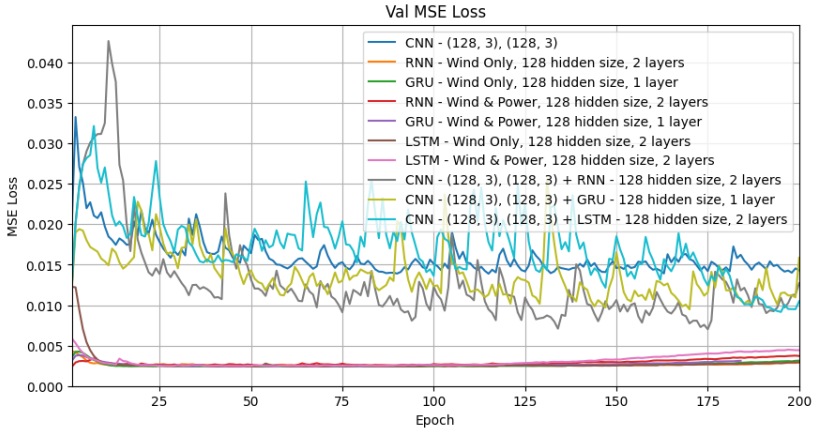

Figure 5.2: Validation Loss for each selected model

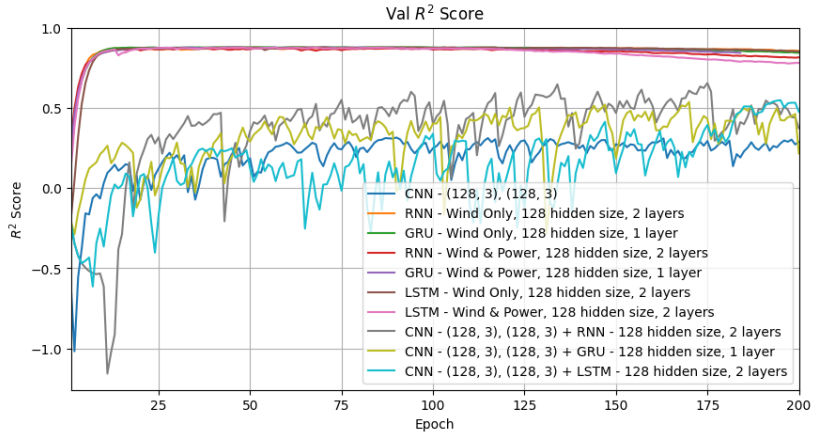

Figure 5.3: R^2 score for each selected model

CNN models and their combinations with RNN models demonstrated unstable learning process with metrics jittering and inability to converge after 200 epochs. The resulted training history represented in figures 5.2 and 5.3 is smoothed with exponential decay method with $\alpha = 0.3$. The model with CNN only layers performed the worst scoring only 0.3438 R^2 score as shown in 5.4. In comparison, the same model followed by 2 LSTM layers achieved R^2 score of 0.5852, significantly improving the performance.

Figure 5.4 illustrates the efficiency of RNN, GRU, and LSTM models in different configurations with similar results, where the best GRU model achieves 0.8797 and the worst RNN models achieves 0.8742 R^2 scores, demonstrating approximately identical results.

Taking the best single layer GRU model with 128 features in hidden layer, the training history is shown in figure 5.5. The graph of MSE loss during training shows a "scissors" effect, as validation line intersects with training at epoch 100 and starts growing, indicating the model overfitting. The model achieved the lowest loss of 0.002388 at epoch 67.

Taking into consideration limited computational resources, each model training is profiled with time taken to iterate over one epoch. Thus, despite the fact that RNN models required

Model	Best Iteration	Train Loss	Val Loss	Train r2_score	Val r2_score	Average Time per Epoch (sec)	Time To Best Epoch (sec)
str	i64	f64	f64	f64	f64	f64	f64
"GRU - Wind Only, 128 hidden size, 1 layer"	67	0.002368	0.002388	0.931753	0.879661	2.717454	179.618753
"LSTM - Wind Only, 128 hidden size, 2 layers"	74	0.002571	0.002403	0.926422	0.879258	0.854111	63.493945
"GRU - Wind & Power, 128 hidden size, 1 layer"	60	0.002366	0.002417	0.931469	0.878248	2.823957	170.712898
"LSTM - Wind & Power, 128 hidden size, 2 layers"	42	0.002424	0.002416	0.930403	0.877408	0.842852	35.203049
"RNN - Wind Only, 128 hidden size, 2 layers"	35	0.00248	0.002457	0.927658	0.87494	2.256417	77.669561
"RNN - Wind & Power, 128 hidden size, 2 layers"	31	0.002496	0.002472	0.927776	0.874265	2.339716	68.722813
"CNN - (128, 3), (128, 3) + RNN - 128 hidden size, 2 layers"	134	0.005327	0.005766	0.84624	0.715629	6.0294	800.067294
"CNN - (128, 3), (128, 3) + GRU - 128 hidden size, 1 layer"	152	0.006393	0.008093	0.81598	0.602465	7.306537	1106.213503
"CNN - (128, 3), (128, 3) + LSTM - 128 hidden size, 2 layers"	191	0.008033	0.008486	0.770058	0.585158	3.67469	668.915994
"CNN - (128, 3), (128, 3)"	132	0.012335	0.01324	0.643802	0.343825	0.536691	70.378912

Figure 5.4: Metrics for selected models after 200 epochs

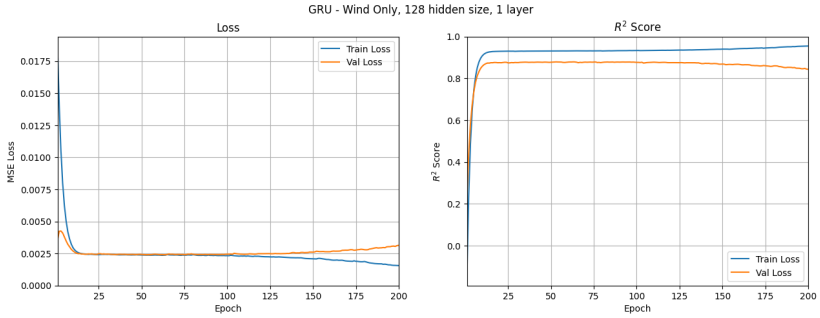

Figure 5.5: GRU Model Training History

the least number of epochs to achieve optimal results, their actual training time is not the best. For example, 2 layered LSTM model achieved the best result at epoch 74, but required 63 seconds, while RNN models required 68 to 77 seconds.

The most complex combinational models required the most time to train with a maximum of 1106 seconds for 152 epochs.

Training histories for other models are provided in the appendix.

Chapter 6: **Challenges and Solutions**

In the progression of the project, several challenges were encountered, particularly during the model training phase.

The first challenge is that of proper data preparation in training the LSTM model, moreover taking into consideration that the project was pioneering in the application of LSTM networks for wind speed prediction. However, prior to working with LSTM models, it is necessary to prepare the dataset in a format that captures temporal dependencies, which is a challenging task without prior experience in this area. To mitigate this, best practices in forecasting time-series with LSTM models was reviewed. This further led to applying the windowing technique in which the dataset was framed in such a way that the structure was in sequences of 7 days to be able to predict the wind speed of the next entry. This kind of method takes advantage of and captures the temporal pattern features perfectly for accurate predictions. The data was also normalized by scaling the input features to help the model converge better during the training. This was further complimented with the extensive experimentations on the window sizes and strides carried to configure the best parameters that would give the best forecasting results.

Another significant challenge was that after a good number of training epochs, the model could start predicting some NaN (Not a Number) values. To address this issue the learning rate was adjusted to be dynamically reduced if the validation loss stopped improving. This helped fine-tune the learning process to avoid being overly aggressive in the updates, which might tend to destabilize the model.

Chapter 7: **Future Work**

While this study represents a significant step towards improving wind speed prediction for enhanced wind energy generation, several avenues for future research remain to be explored.

Firstly, further investigation into different deep-learning techniques and its usage could provide valuable improvements for wind speed prediction problems. One of the promising techniques involve usage of Transformer models, that use attention-based mechanism for features and patterns extraction.

Additionally, incorporating additional meteorological variables such as temperature, humidity, and atmospheric pressure into the modeling framework may lead to more robust and accurate wind speed forecasts, considering the complex interplay between these factors and wind behavior.

Another way to improve the quality of wind speed prediction can be obtained through the usage of longer lookback steps, reaching 3, 5, 7 days and more. By analyzing more distant previous values, it might be possible to find more interesting patterns and spatial features.

Overall, these future research directions hold the potential to further advance the state-of-the-art in wind speed prediction and contribute to the broader goal of sustainable energy generation from wind resources

Chapter 8: **Conclusion**

Wind energy stands as a promising solution for meeting the world's growing energy needs while mitigating the environmental impact of traditional fossil fuel sources. However, the variable nature of wind presents a challenge for using its full potential, requiring accurate prediction of wind speed to optimize the performance of wind turbines. This project focuses on improving wind speed prediction for enhanced wind energy generation by leveraging deep learning technologies, including CNN, LSTM, RNN, and GRU networks and its combinations. After the iterating different models architecture variations, 10 final models were evaluated based on MSE, R^2, and time duration. Although the difference in results of all RNN-like models is minimal, GRU model with only wind speed as feature presents the highest performance, reaching validation MSE = 0.00238 m/s and $R^2 = 0.8796$. In terms of time duration, similar structured LSTM model reaches only 0.879 s per epoch, compared to the average time per epoch of 2.717 s for GRU. These findings underscore the potential of deep learning approaches in optimizing wind energy generation and contribute to the ongoing efforts in providing sustainable eco-friendly future.

Bibliography

[1] J. F. Manwell, J. G. McGowan, and A. L. Rogers, *Wind energy explained: theory, design and application*. John Wiley & Sons, 2010.

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

[3] S. Al-deen, A. Yamaguchi, and T. Ishihara, "A physical approach to wind speed prediction for wind energy forecasting," *Journal of Wind Engineering*, vol. 108, pp. 349–352, 01 2006.

[4] J. Hu, J. Heng, J. Wen, and W. Zhao, "Deterministic and probabilistic wind speed forecasting with de-noising-reconstruction strategy and quantile regression based algorithm," *Renewable Energy*, vol. 162, pp. 1208–1226, 2020.

[5] K. Yunus, T. Thiringer, and P. Chen, "Arima-based frequency-decomposed modeling of wind speed time series," *IEEE Transactions on Power Systems*, vol. 31, no. 4, pp. 2546–2556, 2016.

[6] Z. Ma and G. Mei, "A hybrid attention-based deep learning approach for wind power prediction," *Applied Energy*, vol. 323, p. 119608, 2022.

[7] J. Wong, R. Wang, and S. Bu, "Comparison analysis of deep learning forecasting models with hybrid structure for short-term wind power prediction," in *12th IET International Conference on Advances in Power System Control, Operation and Management (APSCOM 2022)*, vol. 2022, pp. 65–70, 2022.

[8] L. Ji, C. Fu, Z. Ju, Y. Shi, S. Wu, and L. Tao, "Short-term canyon wind speed prediction based on cnnmdash;gru transfer learning," *Atmosphere*, vol. 13, no. 5, 2022.

[9] D. Tian and D. Zhao, "Wind speed prediction based on long short-term memory network and policy suggestions for wind power industry development," in *2021 6th International Conference on Communication, Image and Signal Processing (CCISP)*, pp. 284–288, 2021.

[10] W. Lang, X. Ma, Y. Luo, B. Zhou, and D. Yang, "Wind power prediction based on principal component analysis and long short-term memory networks," in *2019 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, pp. 3063–3068, 2019.

[11] M. D. Chaka, A. G. Semie, Y. S. Mekonnen, C. A. Geffe, H. Kebede, Y. Mersha, F. A. Anose, and N. E. Benti, "Improving wind speed forecasting at adama wind farm ii in ethiopia through deep learning algorithms," *Case Studies in Chemical and Environmental Engineering*, vol. 9, p. 100594, 2024.

[12] M. Liu, P. Qiu, and K. Wei, "Research on wind speed prediction of wind power system based on gru deep learning," in *2019 IEEE 3rd Conference on Energy Internet and Energy System Integration*, IEEE, 2019.

[13] Z. Xu, W. Yixian, C. Yunlong, C. Xueting, *et al.*, "Short-term wind speed prediction based on gru," in *2019 IEEE Sustainable Power and Energy Conference (iSPEC)*, pp. 2309–2313, IEEE, 2019.

[14] Q. Cao, B. Ewing, and M. Thompson, "Forecasting wind speed with recurrent neural networks," *European Journal of Operational Research*, vol. 221, no. 1, pp. 148–154, 2012.

[15] K. Sandhu and A. Nair, "A comparative study of arima and rnn for short term wind speed forecasting," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–6, IEEE, 2019.

[16] Y. Chen, Y. Wang, Z. Dong, J. Su, Z. Han, D. Zhou, Y. Zhao, and Y. Bao, "2-d regional short-term wind speed forecast based on cnn-lstm deep learning model," *Energy Conversion and Management*, vol. 244, p. 114451, 2021.

[17] H. Shi, M. Guan, and M. Ding, "Wind power prediction of cnn-lstm network model based on unsupervised algorithm processing," in *International Conference on Mechanical Engineering, Measurement Control, and Instrumentation*, vol. 11930, pp. 986–991, SPIE, 2021.

[18] M. Afrasiabi, M. Mohammadi, M. Rastegar, and A. Kargarian, "Probabilistic deep neural network price forecasting based on residential load and wind speed predictions," *IET Renewable Power Generation*, vol. 13, no. 11, pp. 1840–1848, 2019.

[19] S. Kazmi, B. Gorgulu, M. Cevik, and M. G. Baydogan, "A concurrent cnn-rnn approach for multi-step wind power forecasting," *arXiv preprint arXiv:2301.00819*, 2023.

[20] S. Demir, K. Mincev, K. Kok, and N. G. Paterakis, "Data augmentation for time series regression: Applying transformations, autoencoders and adversarial networks to electricity price forecasting," *Applied Energy*, vol. 304, 2021.

[21] A. Flores, H. Tito-Chura, and V. Yana-Mamani, "An ensemble gru approach for wind speed forecasting with data augmentation," 2021.

[22] R. Wang, B. Chen, B. Liu, H. Lin, and W. Chen, "The application research of deep learning based method for short-term wind speed forecasting," in *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp. 2378–2381, 2019.

[23] Y. Hu and L. Chen, "A nonlinear hybrid wind speed forecasting model using lstm network, hysteretic elm and differential evolution algorithm," *Energy Conversion and Management*, 2018.

[24] A. P. Sari, D. A. Prasetya, T. Yasuno, A. N. Sihananto, M. M. A. Haromainy, and W. S. Saputra, "Forecasting model of wind speed and direction by convolutional neural network - deep convolutional long short term memory," in *2022 IEEE 8th Information Technology International Seminar (ITIS)*, pp. 200–205, 2022.

[25] R. Yu, J. Gao, M. Yu, W. Lu, T. Xu, M. Zhao, J. Zhang, R. Zhang, and Z. Zhang, "Lstm-efg for wind power forecasting based on sequential correlation features," *Future Gener. Comput. Syst.*, vol. 93, pp. 33–42, 2019.

Chapter 9: **Appendix**

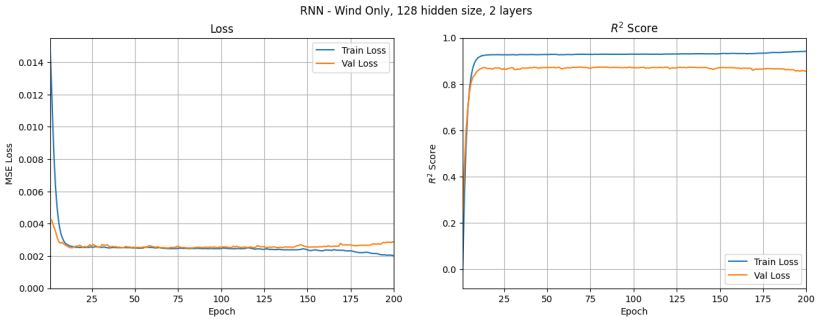

Figure 9.1: RNN - Wind Only, 128 hidden size, 2 layers

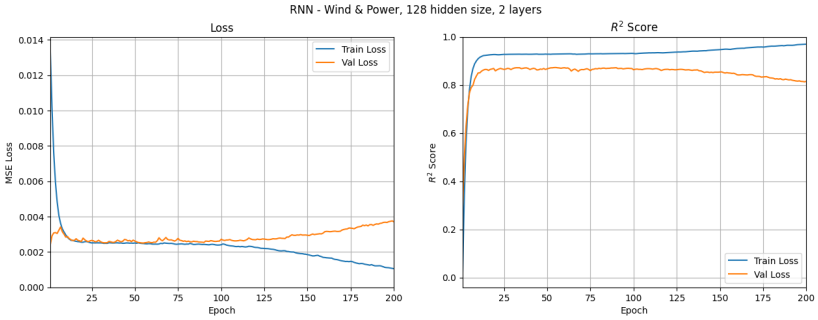

Figure 9.2: RNN - Wind & Power, 128 hidden size, 2 layers

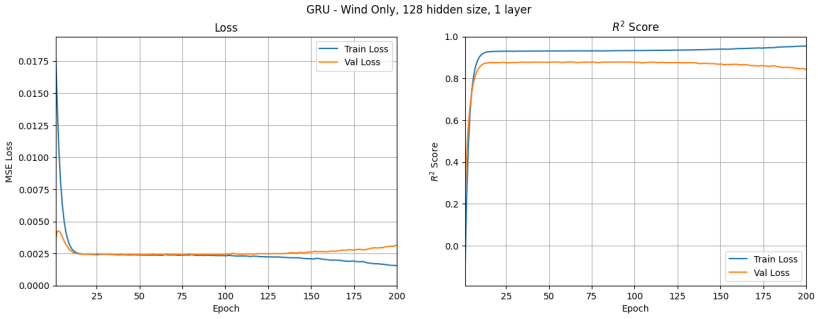

Figure 9.3: GRU - Wind Only, 128 hidden size, 1 layer

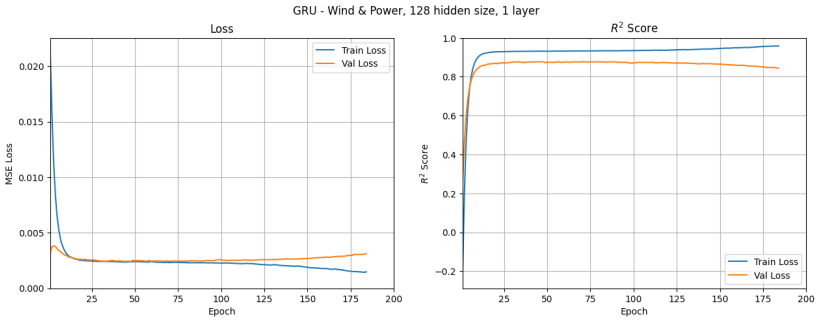

Figure 9.4: GRU - Wind & Power, 128 hidden size, 1 layer

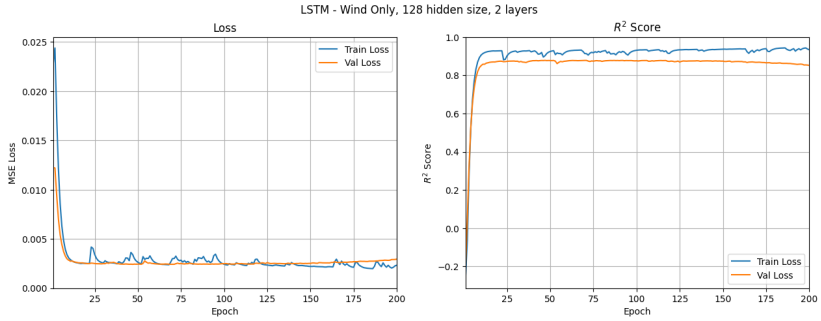

Figure 9.5: LSTM - Wind Only, 128 hidden size, 2 layers

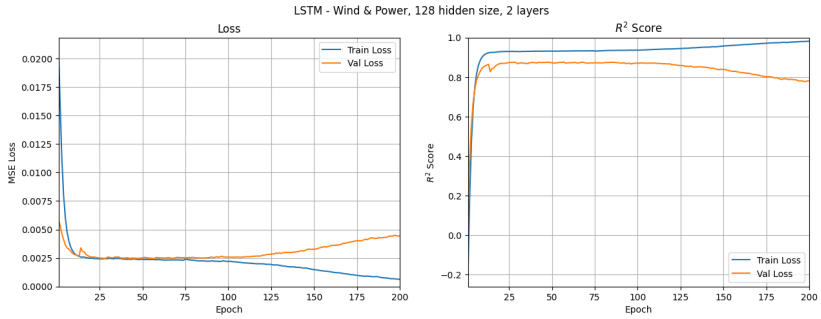

Figure 9.6: LSTM - Wind & Power, 128 hidden size, 2 layers

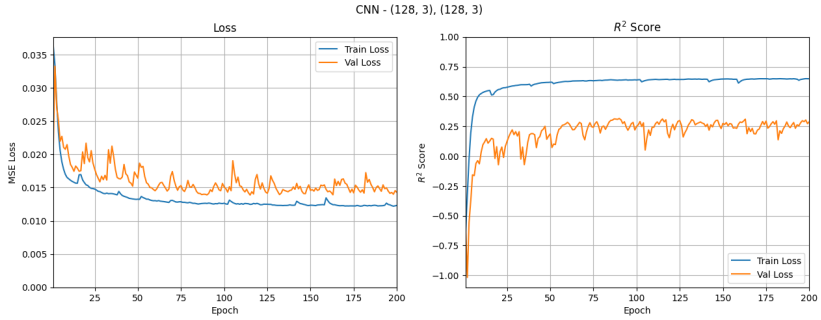

Figure 9.7: CNN - (128, 3), (128, 3)

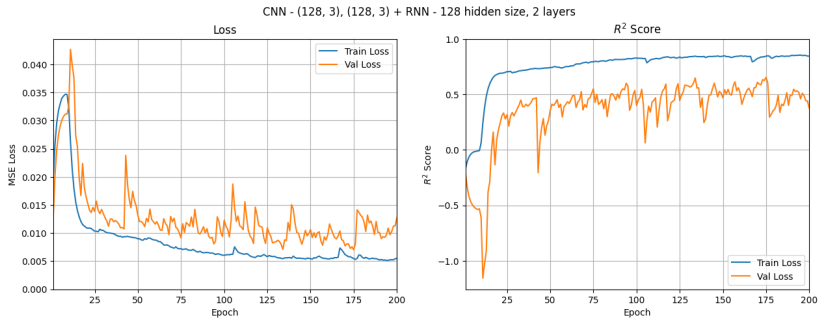

Figure 9.8: CNN - (128, 3), (128, 3) + RNN - 128 hidden size, 2 layers

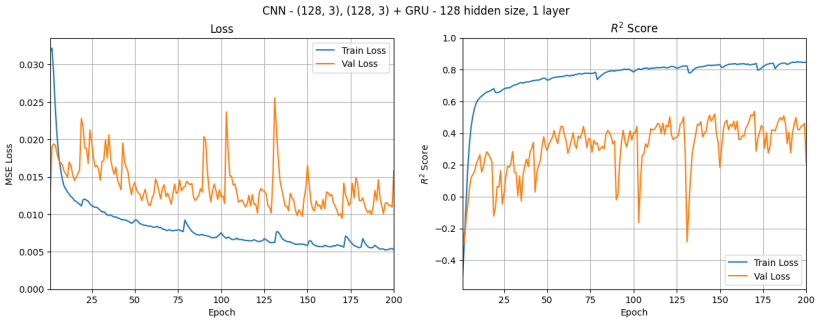

Figure 9.9: CNN - (128, 3), (128, 3) + GRU - 128 hidden size, 1 layer

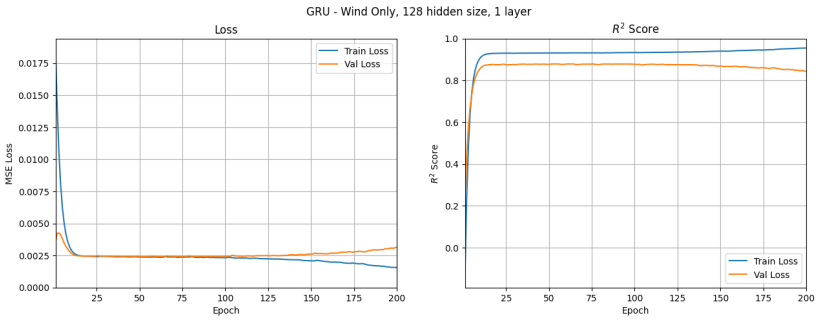

Figure 9.10: CNN - (128, 3), (128, 3) + LSTM - 128 hidden size, 2 layers