Wind Power Forecasting using Machine Learning

Abstract

During the course on power systems, I was introduced to the power grid and the challenges faced by the grid operators due to unpredictable and intermittent nature of renewable energy generation. Wind power strikes the best balance between cost and output. Accurately forecasting wind power will greatly simplify the task for companies, traders and on the longer scale, governments as well. This forecasting has been traditionally carried out using statistical models and recently, using machine learning. Due to the high non-linearity and the strong dependence on weather, this problem has been frequently tackled with a huge ensemble of predictors and lot of simulated data. With the very recent open-source release of the SDWPF dataset [12], we can now experiment with practical real-time data and explore the efficacy of machine learning models. Along with the dataset expansion, it introduces two new features: ERA5 [4] reanalysis data and the elevation data of the turbines. This project aims to explore the dataset, and build a robust model for forecasting wind power, in the light of the new features.

Background

Wind power forecasting is essentially a time-series forecasting problem.

Features

On my survey of various systems and markets, I found a variety of features which affect the predictions. The most influential features are those are of weather, such as wind speed, temperature, humidity, etc. These weather features are usually predicted using large numerical weather models which take 4-6 hours on a supercomputer to give an output. By that time, the conditions usually have changed significantly. Hence, alternative approximate algorithms like the Rapid Refresh are used to give a workable enough prediction. Thus, we have many different forecasts and models which are today used in practice depending upon the requirement.

Depending on the granularity of the data available, other features like Technology of the Turbine, Geographical Location of teh Wind Farm are also used. Physically, Betz's law limits the maximum efficiency of a wind turbine, given by,

$$\eta = \frac{16}{27}\left(1 - \left(\frac{v}{v_{\text{rated}}}\right)^3\right) \tag{1}$$

where v is the wind speed and v_{rated} is the rated wind speed of the turbine.

Finance plays a dominant role in determining the output of the wind farm. In decentralized markets, the price of electricity is determined for every node in the grid. As mentioned in [3], it is called the Locational Marginal Price. Various market participants, decide generation/consumption based on this value metric. The grid's physical constraints (reactive power limits and line capacity) and other events (downtime, outages, and operation limits) also need to be taken into account. The balancing authority, thus provides limits on the actual generation to the plant owners. This results in frequent curtailments of wind power (data in the SCADA Supervisory Control and Data Acquisition system) which isn't visible in the publicly available outputs. With the rise of energy storage facilities and "off-shore" wind power generation, the problem further becomes complicated.

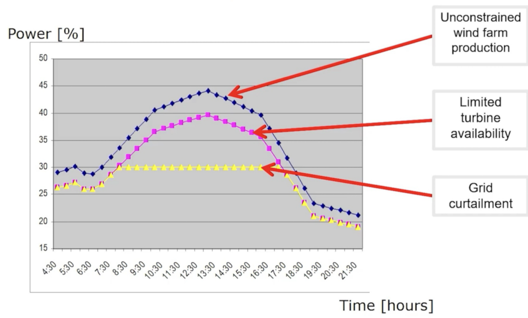


Figure 1: Curtailment of Wind Power [9]

Use-cases

Different users (transmission Operators, wind power owners and electrical utilities) have different cost functions for optimised operation. Even within organisations, it might not even be obvious for them. For eg:

- Trading wind power
- Reserve requirements
- Unit Commitment and Economic Dispatch
- Operation of Combined Wind-Hydro
- Wind Associated with storage
- Design of Optimal Trading Strategies
- Electricity Market Design

Further the forecast horizons and observation least count needs to be adjusted accordingly. The type of forecast (deterministic vs probabilistic) also is chosen based on need. Probabilistic Forecasts model the quantiles of predictions providing some measure of uncertainty. These algorithms are however very expensive to train and are usually used in the balancing markets. Some forecasts also need Ramp Predictions, which are very important for grid stability. Ramps usually indicate sudden changes in wind speeds which needs to be controlled for sufficient amount of inertia in the system. Overall, trading-based optimizations usually results in a spread between value and accuracy of forecasts. Every user needs a different forecast and hence the problem becomes very complex.

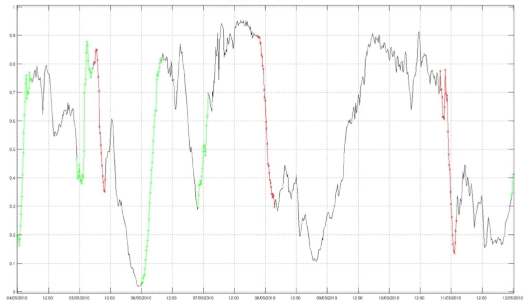


Figure 2: Ramp Rates in Wind Power (Eg: 40% in 2 hrs) [9]

Key Elements for Forecast Solution Selection

The IEA's [2] renewable energy guidelines mentions that Selection/Update of forecasting solutions in which **Quality**, **Reliability** and **Price** are in perfect harmony is usually a complex task and should be desired.

Other Challenges

Wind forecasting is usually carried out in variety of different steps, each model trying to predict some aspect of the time series. One example is as follows:

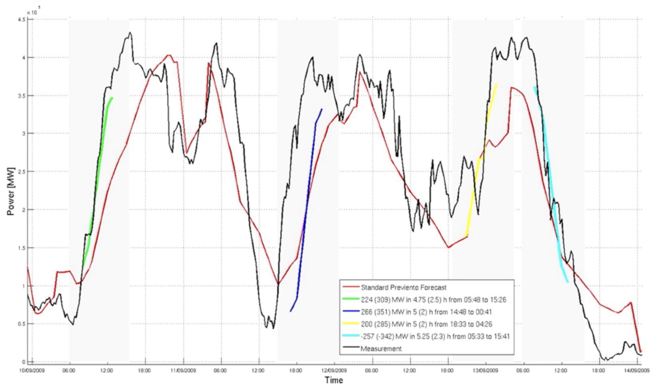


Figure 3: Popular Multi-Step Forecasting [9]

Statistical modelling of the wind power output curve of especially large farms in presence of environmental noise and sensor reading errors becomes inaccurate. Hence models such as ARIMA usually fail to capture the non-stationarities of the time series. Further the wind speed error fom the NWP models is usually uniform over the range of wind speed but that results in huge errors in the power output prediction specially in the most common wind speed ranges.

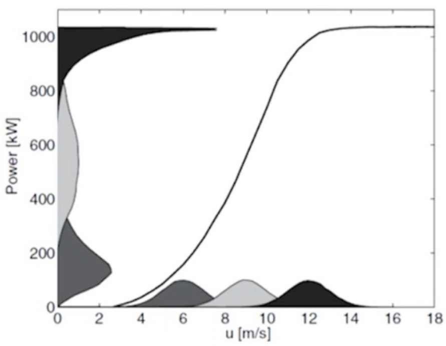


Figure 4: Effects of Non-Linearities

Also, 2 major types of errors - phase and level errors, need to be handled via suitable metrics penalising the optimisation function accordingly. Traditional models usually suffer from the phase error while Machine learning Models usually remain robust to these issues. Following image shows phase and level errors.

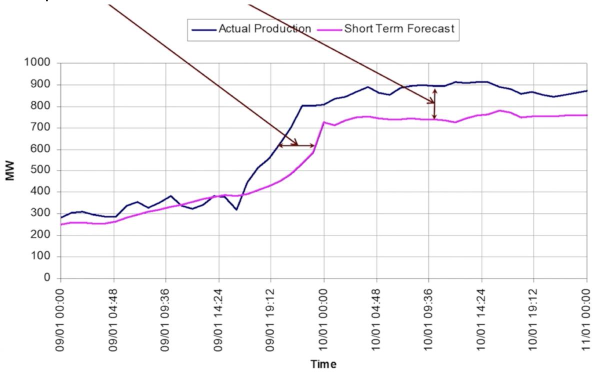


Figure 5: Phase and Level Errors

Project Details

Platform

I have trained and tested the code on an *OSX-ARM* Architecture in a Python 3.11 *miniforge*-based environment. For better GPU access, I tried switching to Kaggle sometimes but due to time limit exhaustion and glitches due to package version, all the hyperparameters are chosen exclusively for my machine with 16 GB RAM.

Code

This Github Repository contains my code for all the phases. The README.md has instructions to run and replicate the results.

Dataset

As mentioned earlier, the recently released (July 2024) SD-WPF dataset [12] was used for all the experiments conducted in the project. The dataset contains data for 134 turbines at a frequency of 25 min, for a duration of around 2 years, for a windfarm (owned by Longyuan Power Group corp. Ltd.) in China, whose location is kept anonymous. All the turbines are given to be of the same make (SL1500/82 turbine type, produced by Sinovel Wind Group Co., Ltd) and the corresponding features are given in Table 1. Apart from the features mentinoed in the table, the data also has the geolocations of each turbines with the elevations for each turbine. The ERA5 reanalysis data provides insights into global farm-level weather data which can be used for sensor calibration, forecast improvements, and other applications.

This dataset was specifically chosen out of the many available open-source datasets as it spans the "real" values (not simulated) of almost 4 out of the 5 categories (SCADA data is missing) mentioned by Effenberger and Ludwig [1].

Evaluation Metrics and Forecast Horizon

Evaluation Metrics play a significant role in determining the "quality" of the output forecast. As an example, for the ramp forecast below, different user choices correspond to differently optimized outputs, having different types of errors. The most common metric used for global evaluation is the mean absolute error (MAE) as it usually penalizes all kinds of errors equally. However, as we can see in figure 6, verifying a ramp with MAE/RMSE brings us to a dilemma, either create a good MAE Score or serve the user needs, not both.

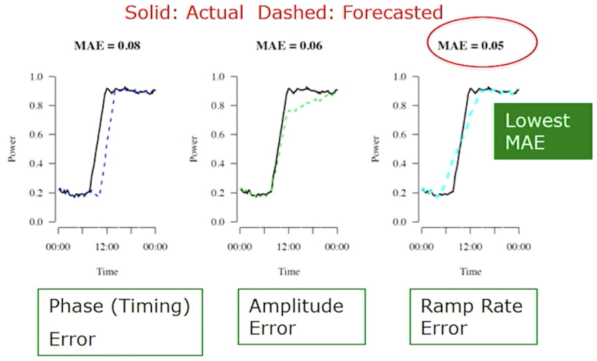


Figure 6: Impact of Metrics [9]

Overall, for this project, it was determined that MAE will be good for maintaining generalisability to other applications as I was mostly interested in modelling the trend and variation of the time series, rather than specific features like ramps.

The forecast horizon also is a key factor in model choice, as my experiments have revealed. For example, the LSTM model performed better for the mid-term forecasts (hourly, 1 week or so), while an ensemble of statistical and tree-based models were good at the ultra-short term periods (intra-day, minute-wise forecasts). However, both are miserable at extrapolating trends for the longer durations (years) which are usually looked at by investors, where very simple models are good at de-trending (as they take into account growth via compounding).

For the purpose of this project, I have chosen the forecast horizon to be 1 day with a frequency of 1 hour. This strikes the balance between the available dataset size and my local computational resources.

Note: For probabilistic forecasts, the popular metrics are different: quantile loss, CRPS, Skill Score, Hit-Miss Rate and Variogram Score are some of the popular ones. Due to lack of computational resources, I have chosen to stick wih deterministic forecasts in this project.

Project Organisation

The project was divided into 3 phases:

1. Data Analysis, Baseline and Initial Approach
2. Exploiting ERA5: Turbine-Level Forecast Refinement
3. Spatio-Temporal Wind Farm Output Modelling

Phase 1

In this phase, I analysed the dataset following the visualisation strategies mentioned in the Coursera Specialisation [6], to develop intuition and simultaneously derive insights for predictions, and built a baseline model using a LSTM-CNN architecture. To start with the forecasting, I restricted myself to a single turbine in this phase. Following are the key highlights of the analysis.

- The turbine locations (with the elevations) are as follows. Clearly, the geometry of the cliff points to the fact that the wind speed as seen by each turbine should be different. This is a key feature to be exploited in Phase 3.

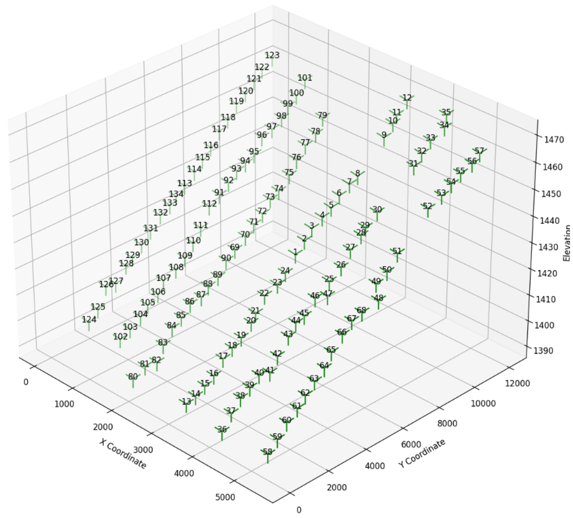


Figure 7: Turbine Locations

- Different parts of the time series were visualised as shown in the table 19. The correlation matrix clearly highlights the high dependence of wind speed on the output:

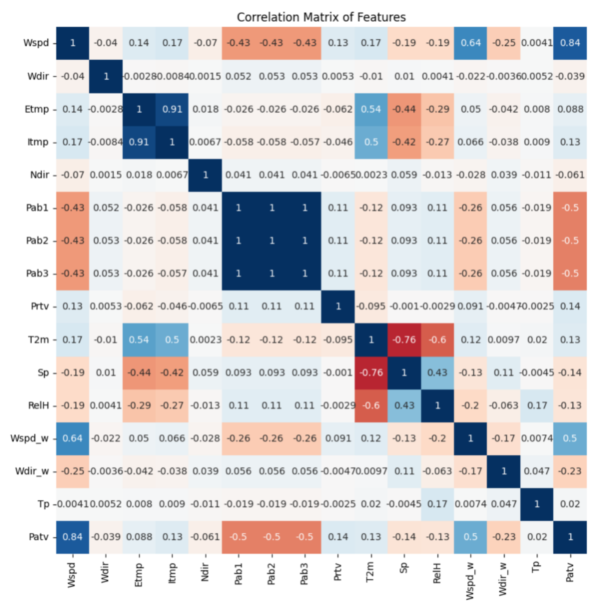


Figure 8: Correlation Matrix

- **Missing Values**: Some values are missed by the SCADA system. Filling the missing values with with next values was found to be better than interpolation. The interpolation method is linear interpolation and the effect is not as good as the other method. [8] In terms of correlation, the correlation between wind speed and wind power is the highest, which is about 0.814801. However, wind speed does not change linearly, so linear interpolation is not suitable to fill the missing value.

- **Wind Speed Detection**: Wind Speed measured by the anemometer roughly follows the Betz Law as seen in figure 9. The noise in the middle part indicates the correction due to curtailment. Further a Poisson-like behaviour is noticed in the distribution of the wind speeds as indicated in Figure 10 which is exploited in Phase 2.

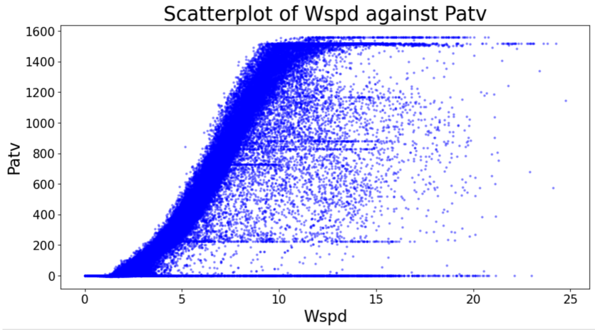


Figure 9: Betz Law Pattern Observation

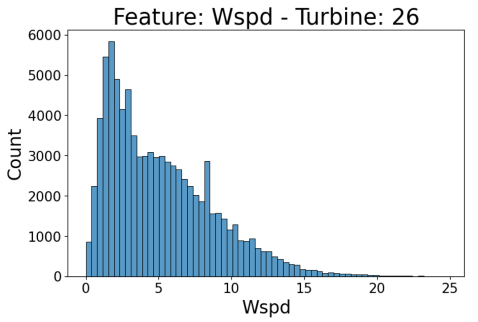


Figure 10: Wind Speed Distribution

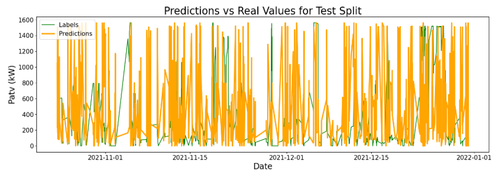
- **Change of Angle Units**: There are 3 features ('Wdir', 'Ndir', 'Pab') which are encoded in degrees. This is problematic because the model has no way of knowing that angles with very different values (such as 0° and 360°) are actually very similar (the same in this case) to each other. Thus transform them into 'sine'/'cosine' representations.

- **Abnormal Value Correction**: Both 'Etmp' and 'Itmp' have really negative values. In fact, these minimum values are very close to the absolute zero (-273.15 °C) which is most certainly an error. Use linear interpolation for the same. Active power has negative values which doesn't make sense in the context of the problem at hand. The paper [12] also addresses this issue by mentioning that all negative values should be treated as zero.

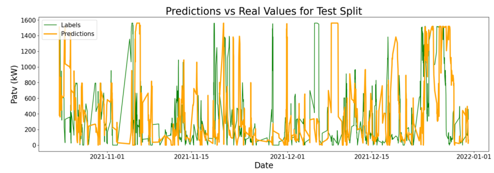
- **Adding Time of Day Feature (TOD)**: To aid the model in learning temporal aspects easily, an angle-based time of day feature was added for each datapoint.

- **Creation of a Baseline**: 3 Baseline Models were created. A poor performance of these naive strategies is observed.

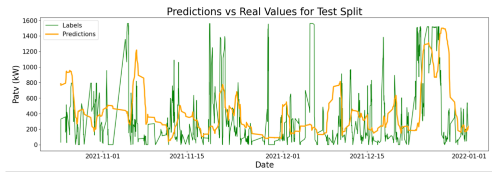
  1. Uniformly Random Forecast: This model predicts the wind power output as a random value between the minimum and maximum values of the training set. It resulted in a MAE of 468.37 KW.



  2. Replicate Yesterday's Behaviour: This model predicts the wind power output as the same value as the previous hour. It resulted in a MAE of 412.99 KW (an improvement of 11.82% over the random baseline).



  3. Moving Average Forecast: A statistical moving average method was used, which gave an MAE of 369.20 KW (an improvement of 21.17% over the random baseline).



- **Initial Approach**: A LSTM-CNN model with the following architecture was used for predictions:

```
model = tf.keras.Sequential(
  [
    tf.keras.Input(shape=(days_in_past * 24,
         num_features)),
    tf.keras.layers.Masking(mask_value=-1.0)
        ,
    tf.keras.layers.Conv1D(256, activation="
        relu", kernel_size=(CONV_WIDTH)),
    tf.keras.layers.Bidirectional(
    tf.keras.layers.LSTM(32,
        return_sequences=True)
    ),
    tf.keras.layers.Bidirectional(
    tf.keras.layers.LSTM(32,
        return_sequences=False)
    ),
    tf.keras.layers.Dense(
    OUT_STEPS * 1, kernel_initializer=tf.
        initializers.zeros()
    ),
```

```
    tf.keras.layers.Reshape([OUT_STEPS, 1]),
  ]
)
```

Referring to the past 1 day of data, predictions were made for the subsequent day in the following scenarios:

1. Model uses only Target Variable History - MAE: 420.91 KW (10.13% improvement over the random baseline)
2. Using all Feature Variable's History - MAE: 434.48 KW (7.24% improvement over the random baseline)
3. Using "Perfect" Wind Speed Forecasts (ERA5) for Predictions - MAE: 232.51 KW (50.36% improvement over the random baseline)
4. Using "Noisy" Wind Speed Forecasts (ERA5) for Realistic Predictions - MAE: 264.80 KW (43.47% improvement over the random baseline). The outputs are plotted in figure 11.

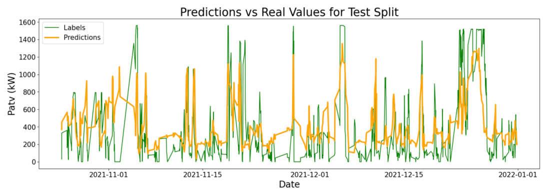


Figure 11: Initial Approach Outputs for Noisy Forecasts

Note: The noise was chosen such that it varies randomly but such that the magnitude increases linearly with time as is generally observed from some NWP forecasts available online.

- **Some Additional Observations and Conclusions**: As the initial approach highlights, the LSTM-CNN model is clearly able to capture the trend and provides 45% improvement over the randomly predicting baseline. Using forecasts has clear advantages over the blind predictions into the future. The MAE still remains high. One of the major challenges I faced training this model was avoiding overfitting while capturing the variance in the data. After applying dropout, regularisation and learning rate scheduling the above results were obtained. However, it was seen that LSTMs cannot learn trends easily with smaller context sizes and generally overfit in presence of noise. Here, due to the limited size of the dataset, other behaviours like Seasonality, etc. couldn't be tried out. Further, turbine locations were unused as we focused on single turbine.

Phase 2

In this phase, I aim to refine the forecasts by exploiting the ERA5 climate re-analysis data. The baseline model was retained and predictions were made only for a single turbine. The *key highlights, results and observations* in this phase are as follows:

- Only wind speed (ERA5 + additive noise) forecasts are considered for simplicity. Since the forecast data was not available, random noise was introduced with special characteristics (like random windows of noise correction) in the same.

  Note: The forecasts were made for ERA5 wind speed data and not the individual turbine level data, as is observed in most cases.

- Using ERA5 not only enhances forecasts, but also reduces reliance on the sensor data of a particular turbine. The correlation matrix in figure 8 shows that the wind speed from ERA5 is correlated in different amounts with different wind speeds measured by the turbines. Further using farm-level weather features like temperature, humidity can help smoothen out intrinsic noise in the system.

- The aim of the method was concentrated on forecast refinement to ensure model output's consistency during generalisation to large-scale forecasts in phase 3.

- The devised architecture consists of an initial **FAISS**-based "closest" datapoint searching. Here *cosine-similarity* was used to find the historically closest point where similar weather and forecasts were observed. Not using this search led to the tree model unable to replicate the full distribution of the wind speed spectrum. However, it also equally prioritized weather over the available forecasts, which were themselves noisy.

- To correct that a domain transformation (encoder-decoder type structure) was first carried out for the forecasted weather time series (next 29 hours) using **Variable Mode Decomposition (VMD)** for number of coefficients $K = 10$. This classical technique however lead to large dimensions of the coefficients.

- Akin to the method in [13], **PCA** was then used to reduce the dimensionality to 7 features per coefficient. Overall this led to a dimensionality reduction 119 for each datapoint, making it more manageable.

- Next to restore the output, an **XgBoost model** was trained on the PCA features + weather features to predict the wind speed.

- As observed in Part 1, wind-speed was found to majorly obey a poisson-like distribution, hence to optimise this model, the Bayesian Optimisation Criterion was set to **count:poisson**.

  Note: Normalising with respect to PoissonRegressor did not turn out to be useful for tree-based models, as they are usually agnostic to many such transformations.

- The XgBoost model required different hyperparameters based on the kind of noise that the NWP model was introducing and also based on how the other features behaved. Thus, for efficient parameter selection, finetuning was carried out using **Bayesian Optimisation** for the 10th horizon (pivot point chosen looking at constancy the **KL divergence** 12).

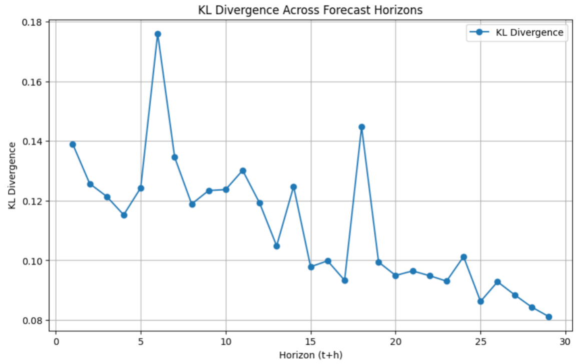


Figure 12: KL divergence between original and corrected forecasts

- Finally, to ensure the outputs were correct, the output distributions are plotted as follows:

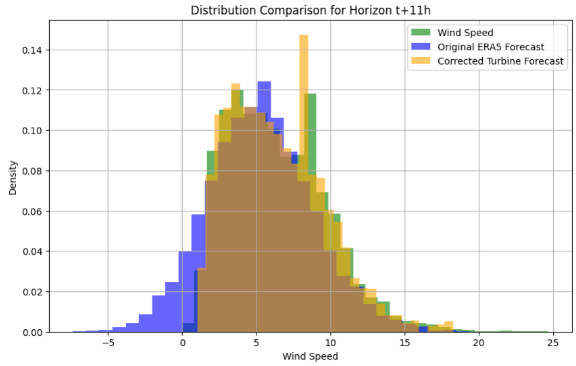


Figure 13: Forecast Distribution

- The robustness to different kind of noise was also seen in the below outputs, where behaviour is greatly stabilized:

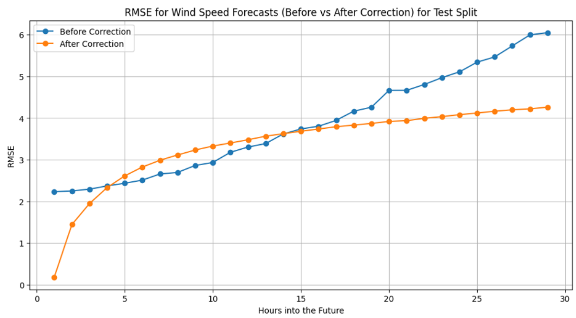


Figure 14: Modified Linearly Increasing Noise Behavior

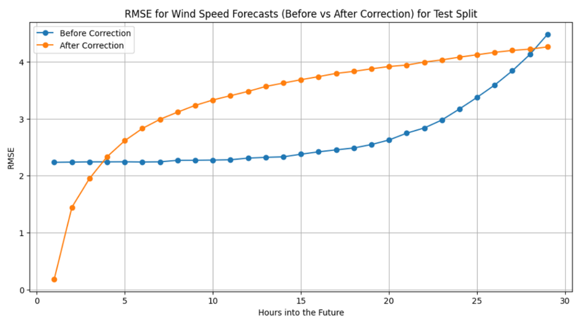


Figure 15: Modfiied Exponential Noise Behavior

- With the updated noise forecasts, we see that the ultra-short term forecasts are very good. A distinct non-linear response was observed, which the CNN-LSTM model found very difficult to learn. Instead it was devised to employ position based encoding and introduce a transformer model for making predictions using multi-headed attention:

```
# Input
inputs = tf.keras.Input(shape=(days_in_past
    * 24, num_features))

# Positional Encoding
positional_encoding = tf.keras.layers.
    Embedding(
    input_dim=days_in_past * 24, output_dim=
        D_MODEL
)(tf.range(days_in_past * 24))

x = tf.keras.layers.Dense(D_MODEL)(inputs)
    # Linear projection to embedding space
x = x + positional_encoding

# Transformer Encoder Blocks
for _ in range(NUM_LAYERS):
    attention_output = tf.keras.layers.
        MultiHeadAttention(
        num_heads=NUM_HEADS, key_dim=D_MODEL
            , dropout=DROPOUT_RATE
    )(x, x)
    attention_output = tf.keras.layers.
        Dropout(DROPOUT_RATE)(
        attention_output)
    x = tf.keras.layers.LayerNormalization(
        epsilon=1e-6)(x + attention_output)

    # Feedforward Network
    ffn_output = tf.keras.layers.Dense(
        D_MODEL * 4,
        activation="relu",
        kernel_regularizer=tf.keras.
            regularizers.l2(L2_Regulation),
    )(x)
    ffn_output = tf.keras.layers.Dense(
        D_MODEL, kernel_regularizer=tf.keras
            .regularizers.l2(L2_Regulation)
    )(ffn_output)
    ffn_output = tf.keras.layers.Dropout(
        DROPOUT_RATE)(ffn_output)
    x = tf.keras.layers.LayerNormalization(
        epsilon=1e-6)(x + ffn_output)

# Output Head
x = tf.keras.layers.GlobalAveragePooling1D()
    (x)
x = tf.keras.layers.Dense(
    OUT_STEPS * 1, kernel_regularizer=tf.
        keras.regularizers.l2(L2_Regulation)
)(x)
outputs = tf.keras.layers.Reshape([OUT_STEPS
    , 1])(x)
```

- The outputs are plotted in figure 16.

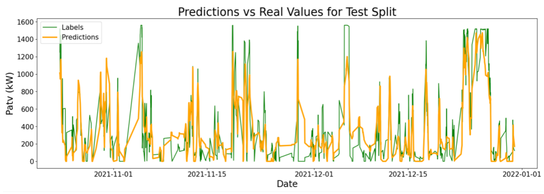


Figure 16: Transformer Model Outputs

Overall, the final output RMSE was observed to be 146.57 KW (a 68.7% improvement over the baseline).

- Clearly, in spite of many architectural and pipeline improvements, the prediction accuracy improvement isn't very high as is seen with many real-world time series, which are usually very noisy, non-linear and intermittent. The model is still not able to capture all the non-linearities in the data. This is a common problem with many time series models and is usually tackled by using more data or more complex models. This is an important limitation of the current approach.

Phase 3

In this phase, I aim to model the spatio-temporal wind farm output, extending from a single to multiple turbines, and making use of the geolocations of the turbines. The *key highlights, results and observations* in this phase are:

- Due to the computational restrictions, the dataset, consisting of the data from 134 turbines, was stripped down to 18 handpicked turbines. The turbines were chosen so that they represent different areas of the hill.

- To begin with a baseline, a simple **CNN-GRU** model was created which would directly predict the aggregated time series using the averaged out features of the concerned turbines.

- Surprisingly, the model performed much better than the one that was used earlier for the single turbine, indicating that some aggregation smoothens out the minor internal variations present within the data.

- An output MAE of 1371.10 KW was observed. Note that these predictions are independent of the wind speed forecasts. However to be able to train the model without exceeding time and memory constraints, I was forced to use smaller horizon (10 time steps). To compensate for this, I converted the problem to short-term forecasts with 25 min intervals.

- The output is plotted in figure 17.

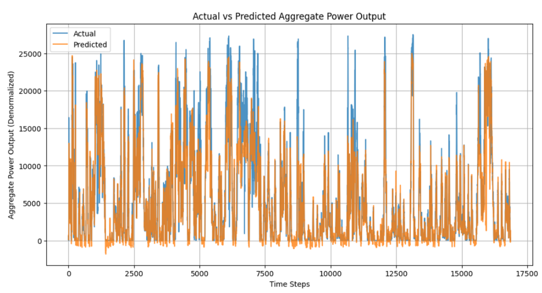


Figure 17: Spatio-Temporal Model Outputs

- To add on the location information, the most efficient way currently known is a message-passing graph neural network. Spatial Graph Structure connects turbines based on normalized distances, effectively factoring spatial relationships and elevation differences into the model.

- The devised architecture consists of GCN Layers that encode spatial information via edge attributes. The GRU Layer captures sequential temporal patterns of node-specific features. Finally a fully Connected Layer maps the learned hidden representations to the forecast power output (Patv).

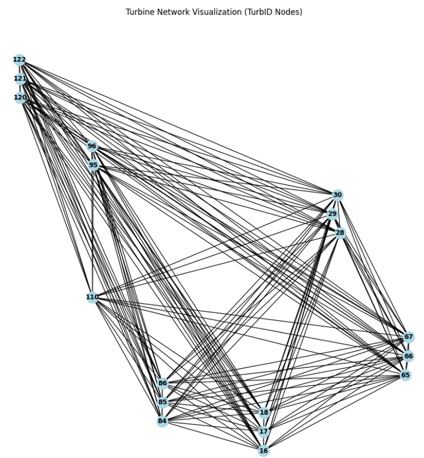


Figure 18: Graph Neural Network Model

- Due to large size of the model and unavailability of a good GPU, only 1000 datapoints per turbine had to be retained for training. This combined with the 7 sized forecast horizon, led to strong overfitting in the model. To make the result more statistically significant, more work will be required.

- On my survey, some works like [7] [5] have taken this one step higher, but with reduced number of features in their available dataset. They reduced the oversimplification of real-world turbine interactions by using dynamic graph networks which use attention-based message passing algorithms. Given how easily my model overfits the data, it appears to be a promising direction to explore with new available features.

Future Work

There are many areas where the project can be extended. Some of the key areas include:

- Financial and real forecasts Impacts have not been analysed due to unavailability of data

- Forecasting Ramps and other aspects of time-series

- False Data Injection [11] Attacks and reducing their impacts on the model.

- Developing Probabilistic Forecasts based on Quartiles. Pinball (Quantile) loss for median oriented MAE to reduce smoothening.
- Using BOA for creating more pivot points rather than 10 used as the hyperparameter generalisations may not work.
- Neural ODE [10] based approach for wind power forecasts has not been explored. When integrated with our graph model, we could have a physics-informed hybrid model which can leverage Chebyshev polynomial-based graph convolution to dynamically tackle the spatial and temporal forecasting problem in phase 3.

Column Name	**Specification Note**
TurbID	Wind turbine ID
Tmstamp	Created time of the record (Time zone UTC + 08:00)
Wspd (m/s)	The wind speed at the top of the turbine (Recorded by mechanical anemometer)
Wdir(°)	Relative wind direction, which is the angle between the wind direction and the turbine nacelle direction. Wind direction and nacelle direction are in degrees from true north
Etmp	Temperature of the surrounding environment (Measured on the outer surface of the nacelle)
Itmp	Temperature inside the turbine nacelle
Ndir (°)	Nacelle direction, the yaw angle of the nacelle (In degrees from true north)
Pab1 (°)	Pitch angle of blade 1 (The angle between the chord line and the rotation plane of the blade)
Pab2 (°)	Pitch angle of blade 2 (Same as above)
Pab3 (°)	Pitch angle of blade 3 (Same as above)
Prtv (kW)	Reactive power
T2m	Temperature at 2 m above the surface (ERA5)
Sp (Pa)	Surface pressure from ERA5
RelH	Relative humidity (Derived based on 2 m dew point temperature and 2 m temperature using Python package metpy)
Wspd_w (m/s)	Wind speed from ERA5 (At a height of 10 m)
Wdir_w (°)	Wind direction from ERA5 (At a height of 10 m)
Tp (m)	Total precipitation from ERA5
Patv (kW)	Active power, the wind power produced by a wind turbine

Table 1: Time series features of the raw SDWPF dataset with specifications

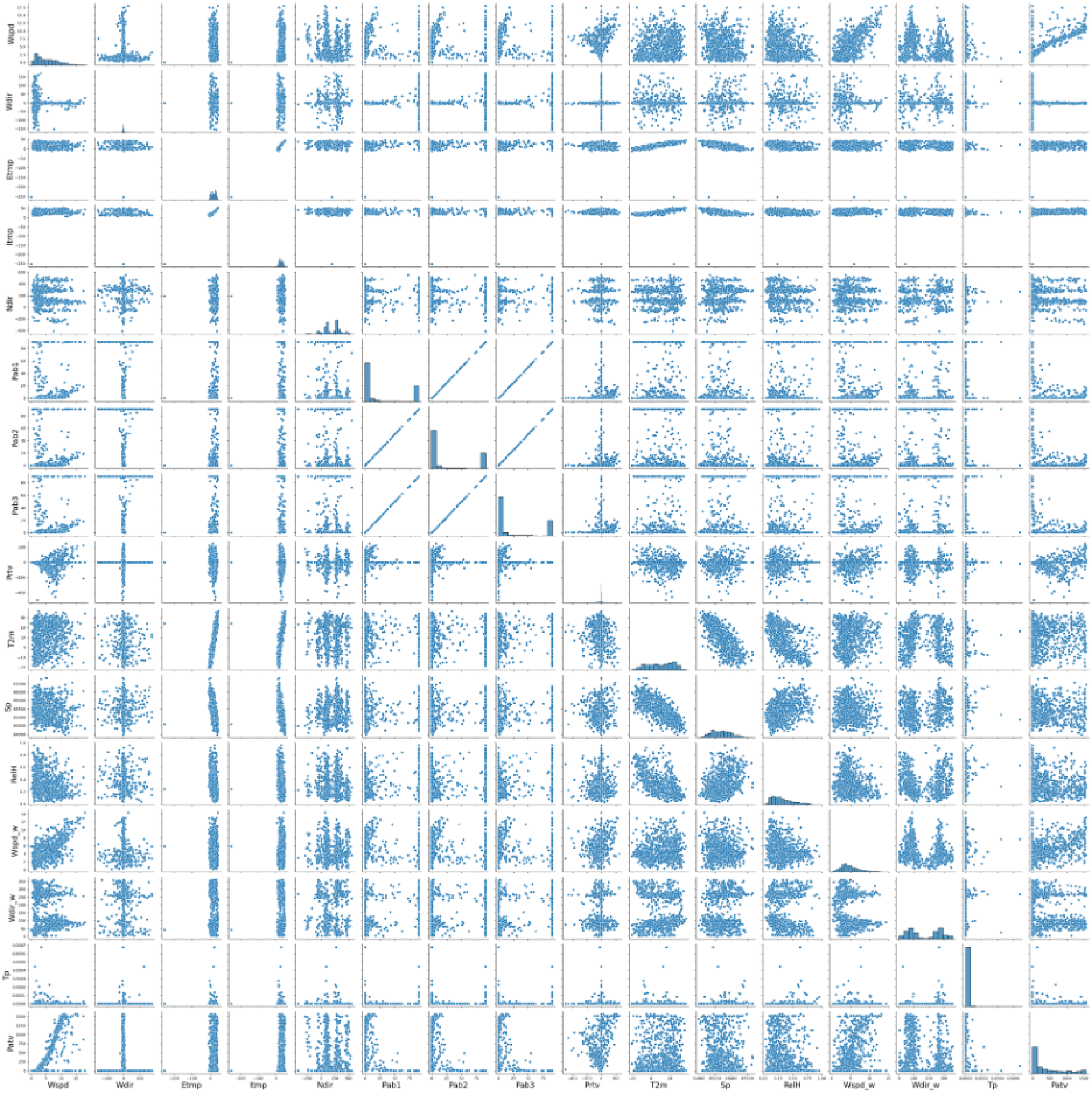


Figure 19: Feature Visualisation

References

[1] Nina Effenberger and Nicole Ludwig. A collection and categorization of open-source wind and wind power datasets, 2022.

[2] Mohrlen C Zack W. Giebel G. *IEA Wind Recommended Practice for the Implementation of Renewable Energy Forecasting Solutions*. Elsevier, 2022.

[3] Zhaoqiang Ge, Minhui Ge, Libing Yang, and Dexing Wang. Locational marginal price of east china electric power market. *International Journal of Power Systems Engineering*, 3(2):214–217, 2008.

[4] Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, Adrian Simmons, Cornel Soci, Saleh Abdalla, Xavier Abellan, Gianpaolo Balsamo, Peter Bechtold, Gionata Biavati, Jean Bidlot, Massimo Bonavita, Giovanna De Chiara, Per Dahlgren, Dick Dee, Michail Diamantakis, Rossana Dragani, Johannes Flemming, Richard Forbes, Manuel Fuentes, Alan Geer, Leo Haimberger, Sean Healy, Robin J. Hogan, Elías Hólm, Marta Janisková, Sarah Keeley, Patrick Laloyaux, Philippe Lopez, Cristina Lupu, Gabor Radnoti, Patricia de Rosnay, Iryna Rozum, Freja Vamborg, Sebastien Villaume, and Jean-Noël Thépaut. The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020.

[5] Jiawei Jiang, Chengkai Han, and Jingyuan Wang. Buaa bigscity - spatial-temporal graph neural network for wind power forecasting in baidu kdd cup 2022, 2023.

[6] Robert Monarch. Ai for good specialization. `https://www.coursera.org/specializations/ai-for-good`, 2024. Online course.

[7] Shurong Peng, Lijuan Guo, Haoyu Huang, Xiaoxu Liu, and Jiayi Peng. Forecastnet wind power prediction based on spatio-temporal distribution. *Applied Sciences*, 14(2), 2024.

[8] Yusu Mao Ruizhi Zhang, Zeyu Long and Nengjun Zhu. Two strategies to reduce the negative effects of abnormal and missing values for wind power forecasting, 2022.

[9] Denmark Technical University. Wind energy. `https://www.coursera.org/learn/wind-energy`, 2024. Online course.

[10] Rui Ye, Xutao Li, Yunming Ye, and Baoquan Zhang. Dynamicnet - a time-variant ode network for multi-step wind speed prediction. *Neural Networks*, 152:118–139, 2022.

[11] Yao Zhang, Fan Lin, and Ke Wang. Robustness of short-term wind power forecasting against false data injection attacks. *Energies*, 13(15), 2020.

[12] Jingbo Zhou, Xinjiang Lu, Yixiong Xiao, Jian Tang, Jiantao Su, Yu Li, Ji Liu, Junfu Lyu, Yanjun Ma, and Dejing Dou. Sdwpf: A dataset for spatial dynamic wind power forecasting over a large turbine array. *Scientific Data*, 11(1):649, 2024.

[13] S. Zhou, C. Y. Gao, Z. Duan, X. Xi, and Y. Li. A robust error correction method for numerical weather prediction wind speed based on bayesian optimization, variational mode decomposition, principal component analysis, and random forest: Vmd-pca-rf (version 1.0.0). *Geoscientific Model Development*, 16(21):6247–6266, 2023.