ReNew Power Hiring Hackathon

Objective : *To predict the rotor bearing temperature of wind turbines*

Shubhankar Mishra

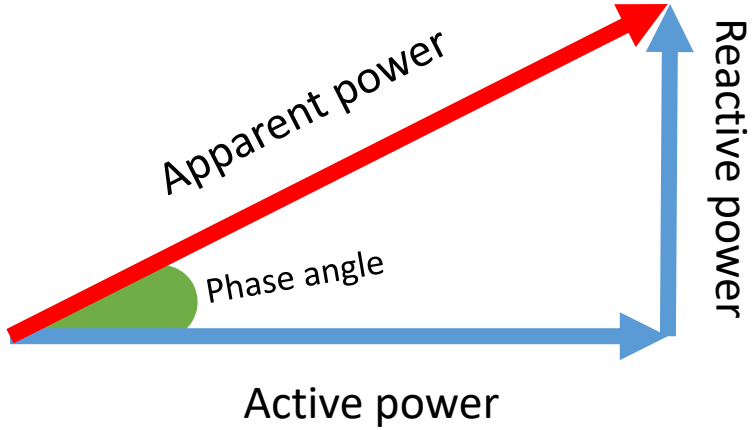
13 Sep, 2022

Overview

- Import training dataset
- Create new relevant features with available data
- Anomaly detection and removal using PCA transformation
- Clustering based on climate/weather features to segregate data into two clusters (most likely hot and cold seasons)
- Data transformation:
  - Skewness correction
  - Concatenation of different turbines data into single df
  - Remove outlier based on visualization wrt. target column
  - OHE for turbine_id
- Finalize features based on impact on the model and also remove high correlated columns
- Split data into train-test with stratification on turbine_ids (train-test should have proportioned data of each turbine)
- Construct Tensorflow sequential neural network
- Define learning rate schedule, optimizer and loss function
- Model training
- Plot losses and metrics wrt. epochs
- Save/load model
- Submission:
  - Repeat same transformations on the test df
  - Prediction from the loaded model
  - Check the predicted data
  - Save the submission file

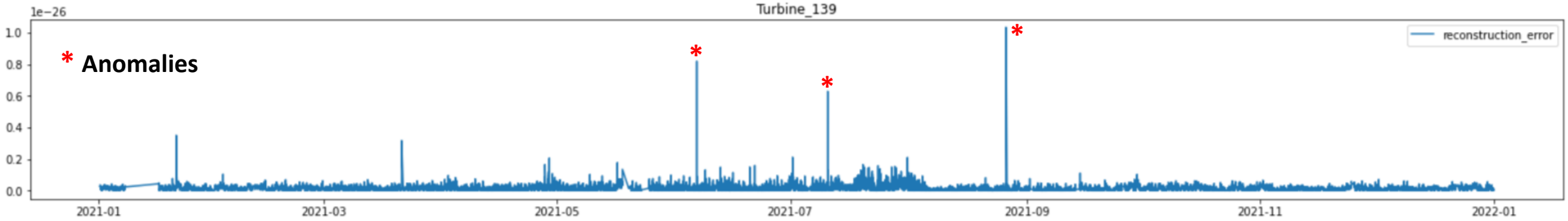
Create new features relevant to wind energy system

- Apparent power – hypotenuses to active and reactive power
- Power difference between current active power and 10 min average power
  - Sudden surge in power can be identified from this feature
- Power from wind speed:
  - Based on expression, $P = \frac{1}{3} \times \rho \times v^3$
  - Power $\propto (wind\ velocity)^3$
  - Power $\propto$ air density $\propto \frac{1}{ambient\ temperatutre}$
- Convert wind direction (degree) into cosine [-1, 1]
- Frictional factor $\propto (generator\ speed)^2$
- Difference between raw and convertor calculated power
- Temperature difference between inside and outside nacelle temperature
- $Phase\ angle = \tan^{-1}\left(\frac{reactive\ power}{active\ power}\right)$



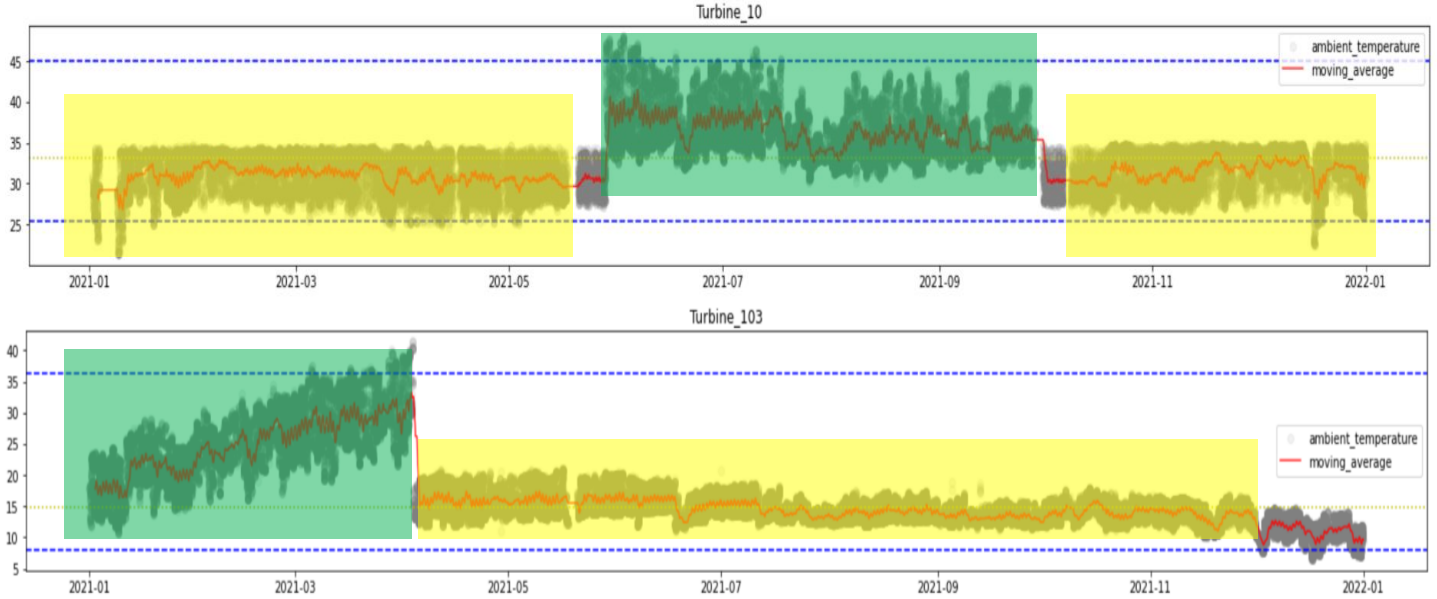
Anomaly detection and removal

- Divide df based on ‘turbine_id’:
  - Use pandas Groupby function
  - Sort data based on ‘timestamp’ (only for visualization)
  - Save dependent and independent variables separately
- PCA transformation of independent variables into same number of features of original data
- Use this transformed features to reconstruct the original data
- Anomaly will have relatively large difference between the original data and the reconstructed data
- Use z-score to eliminate data with extreme reconstruction error



Visualizing features trend of different turbines

- Plot different features separately for each turbine based on sorted timestamp variable
- Use average rolling for better inference on the data
- Some features show different trends in different parts of year
- This could be seasonal effects, such as:
  - power increases with demand surge in summer or good wind conditions
  - ambient and nacelle temperatures increases in hot conditions
- 'ambient_temperature', 'wind_direction_raw', 'wind_speed_turbulence', 'wind_speed_raw'

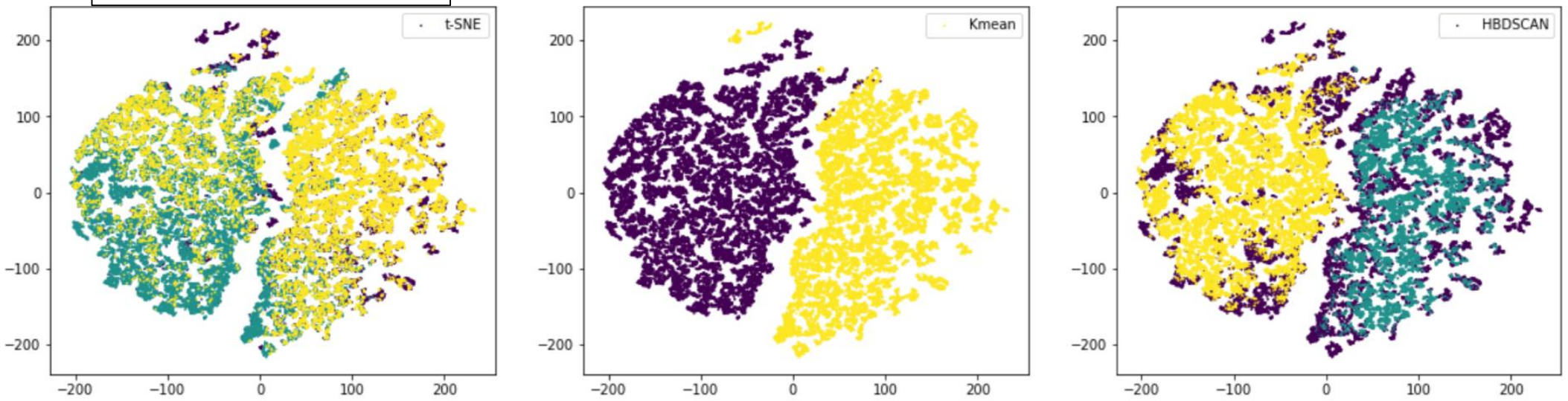


Different trends in feature for same turbine
Different trends in same feature for different turbine

Clustering to label the data

- For each turbine, data can be clustered into two labels
- This label can be used as a new feature for modelling
- Scale the data before clustering to avoid feature dominance
- k-mean and hdbscan clustering based on seasonal features from previous visualization
- The formed clusters labels can be compared with seasons/months extracted from timestamp variable (only for inference)

Seasons based on timestamp

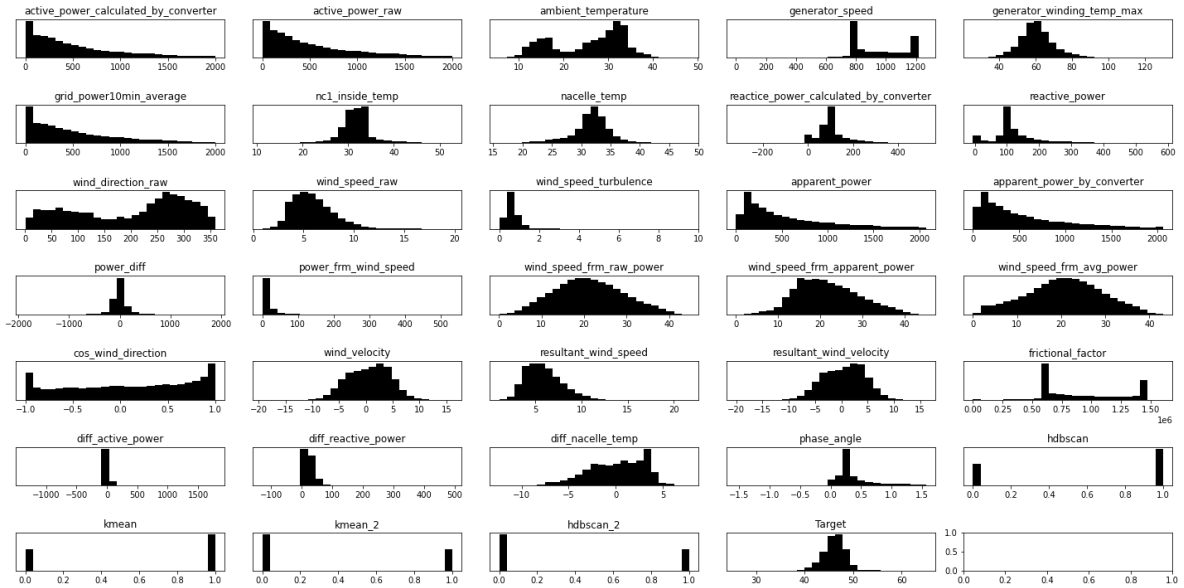


Clusters are able to segregate the data similar to seasons/months from timestamp

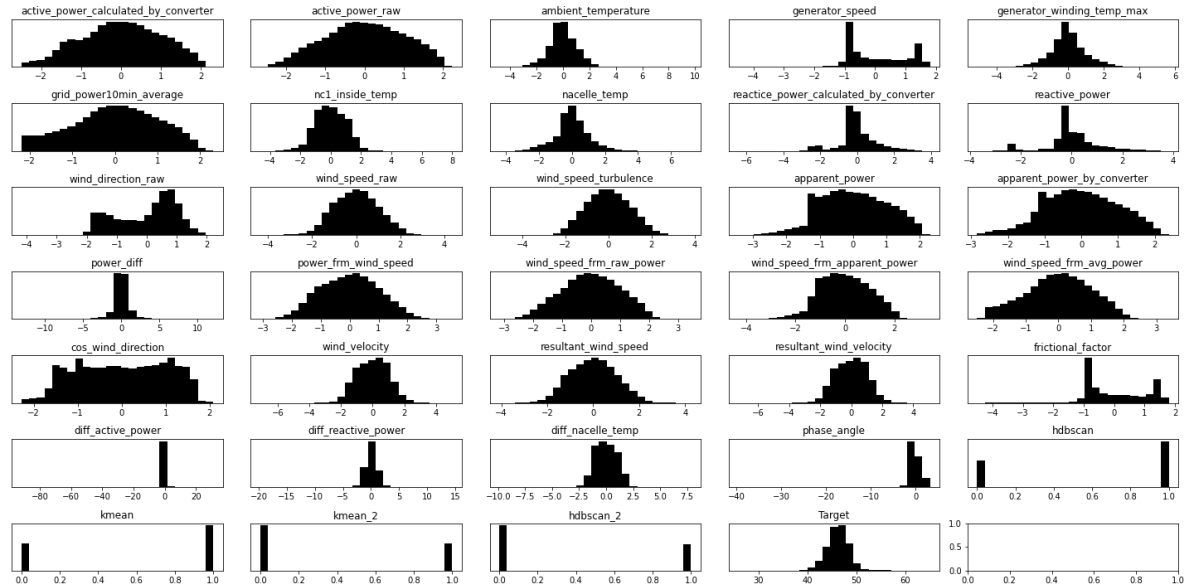
Skewness correction

- Many features are highly skewed which can affect distance based models and neural networks
- Skewness can be corrected with PowerTransformer from sklearn
- yeo johnson transformation will be used as some features contains negative values
- It saves the lambda value hence same can be used for the skewness correction of test data

Before

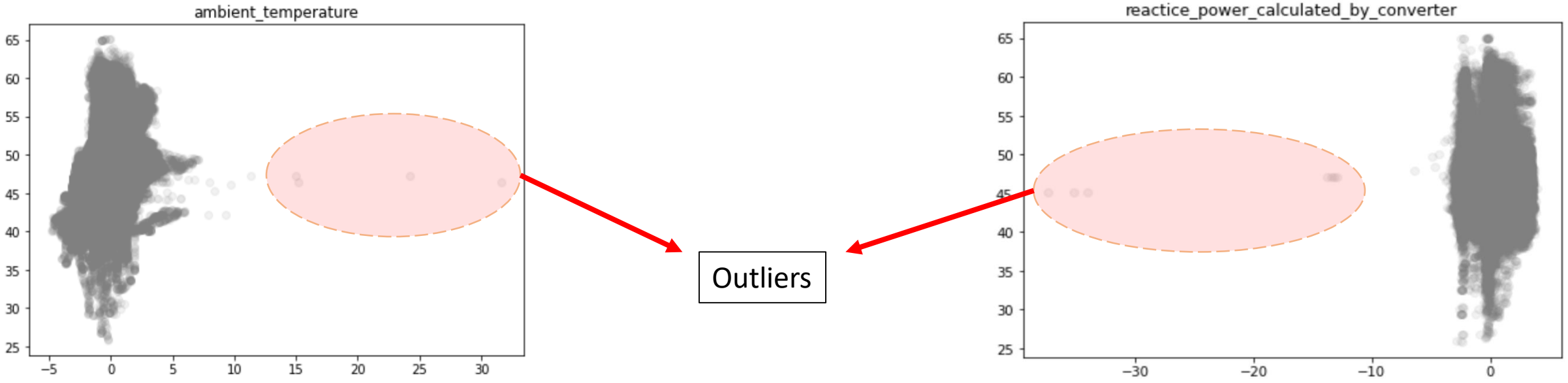


After correction



Outliers removal

- Outliers can be detected and removed after skewness correction
- Scatter plot between Target column and normalized feature columns can be used to distinguish outliers
- After detection these can be removed for performance enhancement

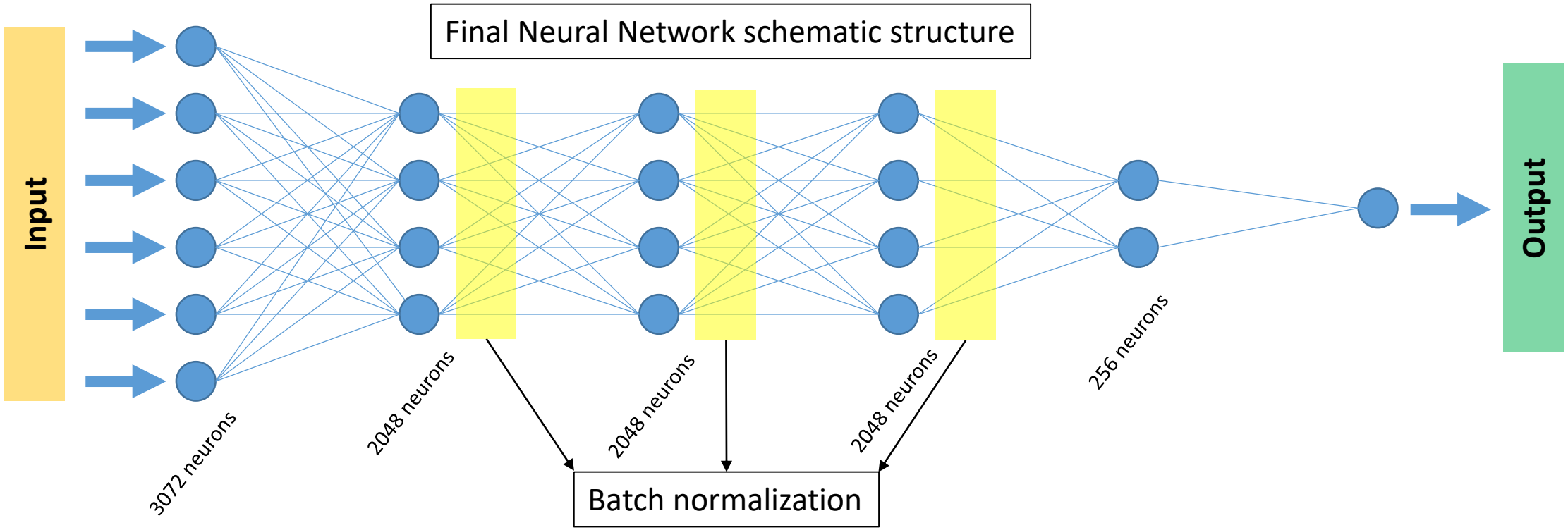


Other transformations & EDA

- One Hot Encode turbine_id column
- Create profile report to check dependencies, cardinality of the features
- Check Correlation between columns
- Feature selection:
  - Many columns are highly correlated (> 0.9). Drop columns with multi-collinearity
  - Drop columns which doesn't impact the model by hit & trail
- Stratified kfold split of data into train-test
- Ensure proportional split of data based on turbine groups for better model training and evaluation

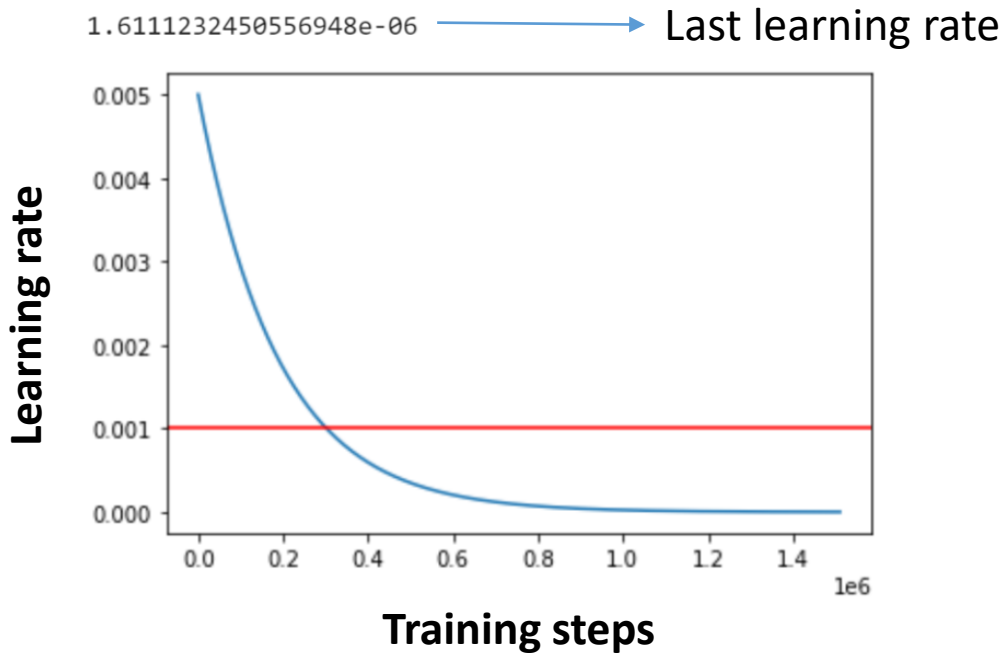
Modelling

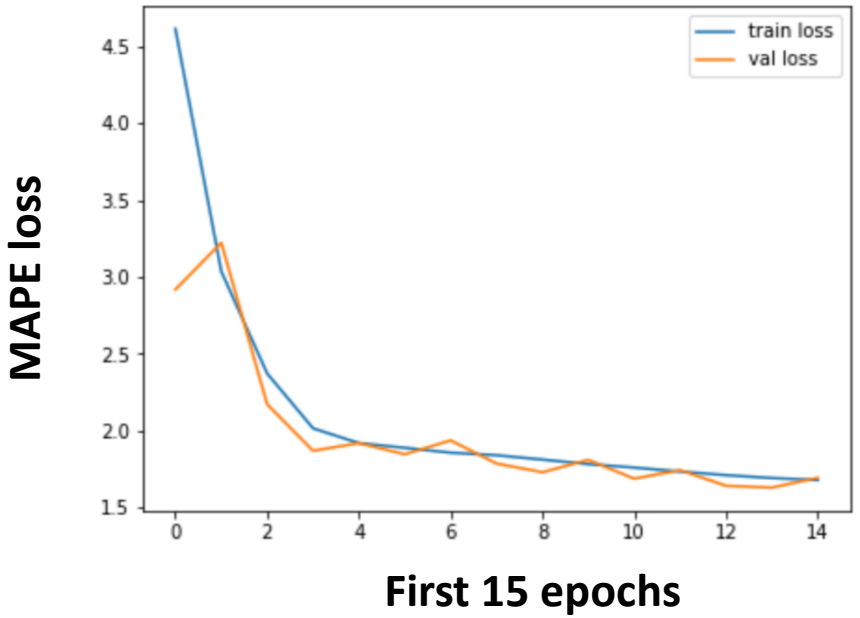
- Different models such as KNN, LinearRegression, RandomForest, XGBoost, Catboost were trained on train split of the data with default parameters
- Simplistic vanilla neural network was also trained with fixed learning rate and SGD optimizer
- Models evaluated based on MAPE and R2 score on test split of the data
- Neural network outperformed other models and was choosen for optimization



Modelling

- Final sequential neural network was constructed with TensorFlow-Keras
- To continue the learning process, decaying learning rate schedule was fed instead of constant learning rate
- Different optimizers such as RMSProp, Adam, SGD were tried; Adam outperformed others
- Error functions such as MSE, MAE, MAPE and MSLE were used; Optimization on MAPE gave best score for the competition
- Regularization techniques such as dropout and batch-normalization were employed to avoid overfitting





Results

- Model trained on complete data for 1000 epochs with 600 batch size
  - Training loss ~ **0.00925 MAPE**
  - Run time ~ **7 hrs**
  - MachineHack public score: **0.01042**
- Trained model and weights saved as json and h5 file resepectively
- Loaded saved model for prediction on test dataset
- Submission file:
  - Repeat same transformations on the test df
  - Prediction from the loaded model
  - Check the predicted data
  - Save the submission file

Inferences / Improvement prospects:

- Clustering of data (most likely seasonal) had significant impact on the model improvement
- Features such as ‘ambient_temperature’ and ‘generator_winding_temp_max’ are found to be extremely important for Target prediction (dropping these affect the model performance)
- Highly correlated features such as ‘active_power_raw’, ‘active_power_calculated_by_convertor’, ‘generator_speed’ etc., provide similar information during training (keeping one and dropping others doesn’t impact the model significantly)
- New features can be derived from high correlated features to increase their importance
- Increasing perceptron/neurons in the first layer improves the optimization
  - Increasing the neural network capability of extracting low level features in the front layers
- Local anomaly/outlier removal can be tried
- Training neural network takes approx. 7 hrs. Tuning of epoch, batch_size and learning rate can decrease computational time

Conclusion

How can ReNew predict failure of a component by looking at target temperature and prevent such failures before they happen?

- If the model is highly accurate then the residual between predicted value and actual value of the new data can be indicative of wind turbine's abnormal behavior
- A threshold (based on training error μ and σ) can be set to consider if the residual value is significant enough
- An indicator which counts the instances of residual crossing this threshold in a fixed time period can be made
- If for a particular turbine, indicator value is relatively high then it can examined and called to maintenance

THANK YOU