Power Consumption AI Prediction

Software Requirement Specification Document

Authors of the SRS

Students' names:

Dania Altman 208853804

Lior Alvas 316121532

Noam Borger 316550482

Ben Yaakobi 313133910

Anzor Torikashvili 321178204

Georgiy Kravchenko 342483179

Supervisors' names:

Mark Israel

Mayya Lihovodov

Purpose of The Research

Commercial electricity users in Ukraine are required to state how much electricity they are going to use in each following month and buy it accordingly. However, if they don't use all of it they practically lose money. On the other hand, the fee for excessive usage is twice(and sometimes three times) than the regular fee by the law. According to recent statistics, AWDP increased to UAH 585 per MWh in 2015.

As a result, these users are seeking the most accurate estimation(or prediction) of the following month's usage.

Hence, this study's aim is to predict, with at least 95% accuracy, the electricity usage of commercial entities.

To achieve this kind of accuracy, this research will include Machine Learning algorithms that would take into account various factors that could affect the power consumption, such as: weather conditions, factory machines' power usage, etc.

Comparison among methods

Research Name	Research Purpose	Pre-processing	Factors	Methods	Accuracy	Notes
Power Consumption Prediction and Anomaly Detection Based on Transform and K-Means.	In this paper, we combine the widely used deep learning model Transformer with the clustering approach K-means to estimate power consumption over time and detect anomalie.	To make the data more stable, we apply the Min-Max Normalization procedure. This will make the model's training easier and its convergence faster. We designed the model supervision task to estimate the following hour's electric energy usage based on multivariate data collected every 23 h, and we implemented it by using a 23-hour sliding window We utilized the strategy of randomly inserting abnormal points in the test data to better compare and assess the model's anomaly detection capabilities because this experimental data set does not mark aberrant time points. In the 200 days (4,800 h) of the test set, we randomly selected a value every day and double it, and assume it is an outlier, so there are 200 outliers in the 4,800 data in the test set.	power theft	K-Means clustering LSTM Transformer model	Accuracy K-means:0.96 LSTM:0.97 our method: 0.97 precision: K-means: 0.82 LSTM: 0.74 our method: 0.80 Recall K-means: 0.28 LSTM: 0.60 our method: 0.66 F1 K-means: 0.42 LSTM: 0.66 our method: 0.72 RMSE(prediction) K-means:- LSTM: 0.91 our method: 0.74	We tested the proposed model, and the results show that our method outperforms the most commonly used LSTM time series model. To make the data more stable, we apply the Min-Max Normalization procedure. This will make the model's training easier and its convergence faster. We designed the model supervision task to estimate the following hour's electric energy usage based on multivariate data collected every 23 h, and we implemented it by using a 23-hour sliding window. K means algorithm will help estimate power consumption over time and detect anomalies.

Research Name	Research Purpose	Pre-processing	Factors	Methods	Accuracy	Notes
Energy consumption prediction by using machine learning for smart building: Case study in Malaysia.	The research aims to address the problem of reducing energy wastage by developing a predictive model for energy consumption in Microsoft Azure cloud-based machine learning platform.	The cleaning method was evaluated based on Raw Bias (RB), Coverage Rate (CR) and R-Squared criteria.	The input data used was weather predictions and hourly electricity loads from two days earlier. They used one year's power consumption.	k-NN	The RMSE value was comparably smaller than the other method at 14.93431 and 0.5439403 MAPE 0.942855477 and for another tenant was 0.40 the training time was 37.916s	It produced a promising result with low raw bias and coverage rate of more than 90% of the actual value.
				SVM	the RMSE value was 4.7506789 and 3.5898263 and the MAPE 0.241318507, for different tenant 0.666018364 the training time was 18 h 38 m 55.324s	
				ANN	the RMSE value was 8.874015 and 4.540988 and the MAPE 1.108522675, for different tenant 1.841600488 the training time was 4 h 39 m 30.035s	

Research Name	Research Purpose	Pre-processing	Factors	Methods	Accuracy	Notes
Short-Term Electricity Consumption Forecasting Based on the EMD-Fbprophet-LSTM Method	This paper adopts a time-series prediction model based on the EMD-Fbprophet-LSTM method to make short-term power consumption predictions for an enterprise's daily power consumption data.	For missing data and anomalies, the average of the valid data before and after two weeks is used to fill in the missing and anomalous data. The preprocessing of the data does not remove data points such as holidays which cause significant changes in electricity consumption.	Season, Weather, Holidays, Daily, weekly, and seasonal cyclicality	ARIMA	MAPE - 2.21% RMSE - 0.27 MAE - 0.19	
				Fbprophet	MAPE - 2.33% RMSE - 0.20 MAE - 0.20	
				EMD-Fbprophet	MAPE - 1.71% RMSE - 0.15 MAE - 0.15	
				LSTM	MAPE - 0.49% RMSE - 0.05 MAE - 0.04	
				EMD-Fbprophet-LSTM	MAPE - 0.28% RMSE - 0.03 MAE - 0.02	

Research Name	Research Purpose	Pre-processing	Factors	Methods	Accuracy	Notes
Forecasting electricity consumption: A comparison of regression analysis, neural networks and least squares support vector machines	In this study, the LS-SVM is implemented for the prediction of electricity energy consumption of Turkey	The interpolation method was used to finc	Installed capacity (IC) gross electricity generation (GEG) population (P) and total subscribership (TS)	LS-SVM	MAPE=1.004 MaxError=4.4 MSE=2.06 RMSE=1.435 SSE=26.782 R-squared=99.98	The LS-SVM model has resulted in absolute training and testing errors of 0.876% and 1.004% respectively, which is more successful than the other two models. Also, the success of the ANN model in the training and testing processes is close to that of the LS-SVM model.However, ANN's lack of convergence to the actual value in some of the test data reduces the success rate in comparison to LS-SVM.This is also clearly seen in the sensitivity and specificity analysis. The test results are not within acceptable limits in a traditional regression analysis and have a higher error rate. Especially, for the year 2010, which was used for validation purposes, the LSSVM model achieved more successful results than the MLR and ANN models by 1.70% and 0.88% respectively.
				ANN	MAPE=1.19 MaxError=5.92 MSE=3.3 RMSE=1.82 SSE=42.85 R-squared=99.98	
				MLR	MAPE=3.34 MaxError=8.25 MSE=6.45 RMSE=2.54 SSE=83.79 R-squared=99.76	

Research Name	Research Purpose	Pre-processing	Factors	Methods	Accuracy	Notes
Predictions of electricity consumption in a campus building using occupant rates and weather elements with sensitivity analysis: Artificial neural network vs. linear regression			Hourly data Dividing the data into two groups – working and non-working days. Temperature humidity ratio wind speed normal solar radiation cloud type	Linear regression	Normalized mean bias error (NMBE) Working 17.61 Non-working 13.80 Coefficient of variation of the root mean square error (CVRMSE) Working 11.54 Non-working 5.04	They used impact factor value (IV) to find the impact of each input node parameter based on statistical approaches after the module training process. The results showed that the ANN model was more accurate and stable than the linear regression method
			irradiance building occupancy rates electricity consumption of a campus building	Artificial neural network (ANN)	Normalized mean bias error (NMBE) Working 10.97 Non-working 13.90 Coefficient of variation of the root mean square error (CVRMSE)	

Research Name	Research Purpose	Pre-processing	Factors	Methods	Accuracy	Notes
An Energy-Fraud Detection-System Capable of Distinguishing Frauds from Other Energy Flow Anomalies in an Urban Environment.	This paper describes a novel algorithm of energy fraud detection, utilizing en-ergy and energy consumption specialized knowledge poured into AI front-end.		$EActual$ —Energy that is actually consumed by the customer $Evisible$—Energy that is registered in meters in the fraud-case $Ehidden$— Energy that is not recorded by meters but is consumed by the	Proposed Ridge + HDS	Accuracy-data Separation of Data Mismatches Anomaly-yes Separation of Preventive Maintenance Anomaly—yes Separation of Cyber-Attack Anomaly—yes Reported Super Consumption Identification and Separation-yes	
				Proposed KNN + HDS	Accuracy-0.84 Precision-0.885 f1-Score- 0.835 Separation of Data Mismatches Anomaly-yes Separation of Preventive Maintenance Anomaly—yes Separation of Cyber-Attack Anomaly—yes Reported Super Consumption Identification and Separation-yes	
				Proposed RF + HDS	Accuracy-0.89 Precision-0.915 f1-Score- 0.89 Separation of Data Mismatches Anomaly-yes Separation of Preventive Maintenance Anomaly—yes Separation of Cyber-Attack Anomaly—yes Reported Super Consumption Identification and Separation-yes	
				RUSBoost	Accuracy-0.869 Precision-0.85 f1-Score- 0.871 Separation of Data Mismatches Anomaly-no Separation of Preventive Maintenance Anomaly—no Separation of Cyber-Attack Anomaly—no Reported Super Consumption Identification and Separation-no	

Methods

SVM

SVM is a machine learning algorithm that can be used for both classification or regression challenges.
The SVM kernel is a function that takes low dimensional input space and transforms it to a higher dimensional space i.e. it converts non separable problem to separable problem. It is mostly useful in non-linear separation problems.In one research they used SVM and was used with Radial Basis Function (RBF) as its kernel function.This methodology is usually known as a maximum margin classifier and is utilized to tackle problems regarding classification and regression for a large dataset.

Pros

- Very accurate.
- It can work with a lot of dimationsios and it uses a small group of data to create the vectors.

Cons

- Long time of data processing
- It is hard for SVM to process with overlap of classes.

LSTM

LSTM (Long short-term memory) is an artificial recurrent neural network (RNN) architecture. LSTM can connect previous information to the present task, meaning it remembers information for a long period of time.

Just like RNN, LSTM has a form of a chain of repeating modules of neural network but the repeating module has a different structure. LSTM uses internal mechanisms called gates that can regulate the flow of information. These gates can learn which data is important to keep or throw away.

The cell state and its various gates are the core concept of LSTM. The cell state, in theory, can carry relevant information that can be used in later time steps. As the cell state goes on its journey, the gates are adding or removing information. The gates learn what information is relevant to keep or forget during training.

The gates contain sigmoid activation. Sigmoid takes the values and changes them to be between 0 and 1. Using this function the neuron forgets data (multiply by 0) or keeps the data (multiply by 1). This is the forget gate.

The Input gate is the gate that updates the cell state. Using sigmoid function, the neuron will decide which value will be updated by transforming the previous hidden state and current input to values between 0 and 1.

Now there is enough information to calculate the cell state. the cell state gets pointwise multiplied by the forget vector. Then the neuron takes the output from the input gate and does pointwise addition which updates the cell state to new values that the neural network finds relevant.

The output gate decides what the next hidden state should be. Passing the previous hidden state and the current input into sigmoid function. then passing the newly modified cell state to the tanh function. Then multiplying the tanh output with the sigmoid output to decide what information the hidden state should carry. The output is the hidden state. The new cell state and the new hidden state are carried over to the next time step.

Pros

- Able to model long-term sequence dependencies.
- Can solve the short term memory problem better than regular RNN.

Cons

- Increase the computing complexity (compared to RNN).
- The memory required is higher (compared to RNN) due to the presence of several memory cells.

Transformers

Transformers is a deep learning model that adopts the mechanism of self-attention, differentially weighting the significance of each part of the input data. Like recurrent neural networks (RNNs), transformers are designed to handle sequential input data, such as natural language, for tasks such as translation and text summarization.It is used primarily in the field of natural language processing (NLP) and in computer vision (CV). Transformers solve problems by using Convolutional Neural Networks together with attention models. Attention increases the speed of how fast the model can translate from one sequence to another.

The transformers model consists of six encoders and six decoders. Each encoder consists of two layers: self-attention and a feed Forward Neural Network.

The encoder's inputs first flow through a self-attention layer. It helps the encoder look at the other words in the input sentence as it encodes a specific word. Each word is embedded into a vector. Firstly it creates three vectors from each of the encoder's input(for each word we create a query, key and value vector. These vectors are created by multiplying the embedding by three matrices that we trained during the training process. These vectors are abstractions that are useful for calculating and thinking about attention.). Secondly we calculate a score. The score determines how much focus to place on other parts of the input sentence as we encode a word at a certain position, the score is calculated by taking the dot product of the query vector with the key vector of the respective word we're scoring. After the scoring, we divide it by 8 and then pass the result through a softmax operation (softmax normalizes the scores so they're all positive and add up to 1). The softmax score determines how much each word will be expressed at this position (higher softwax score means that the word must be in this position). After softmaxing scores we take each value vector and multiply them by their softmax score. The intuition here is to keep intact the values of the words we want to focus on and cast out irrelevant words. Finally we sum up the weighted value vectors. This produces the output of the self attention layer at this position (this concludes the self-attention calculation). The resulting vector is one we can send to the feed-Forward Neural Network.

Pros

- Training on larger datasets than was once possible with LSTM.
- Unlike RNNs, transformers do not necessarily process the data in order. Rather, the attention mechanism provides context for any position in the input. This allows for more parallelization computing than RNNs which reduces training times. (example: if the input data is a natural language sentence, the transformer does not need to process the beginning of the sentence before the end. Rather, it identifies the context that confers meaning to each word in the sentence).

Cons

- Limited Access to Higher Level Representations.

Decision Tree

Decision Tree is one of the easiest and popular classification algorithms to understand and interpret.
Classification is a two-step process, learning step and prediction step, in machine learning. In the learning step, the model is developed based on given training data. In the prediction step, the model is used to predict the response for given data.
In our project we will need a low value of Entropy.

The goal of using a Decision Tree is to create a training model that can be used to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data).

Data types-Classification and continuous variables.

Pros

- Can be used for solving regression and classification problems.

Cons

- A small change in the data can cause a large change in the structure of the decision tree causing instability.

ANN

Artificial Neural Networks (ANN) are multi-layer fully-connected neural nets. They consist of an input layer, multiple hidden layers, and an output layer. Every layer is a dense layer meaning each node in one layer is connected to each node in the next layer.

The forward pass: A given node takes the inputs, multiplies them by the weights, sums the results, adds the bias and passes the sum through a non-linear activation function. The result from the activation function is the output of the node. This is the process of prediction. Before the prediction there is a need to train the model.

To train the model, first the weights are randomly initialized. Then for every training example, perform a forward pass using the current weights. The result from the last node is the final result. Using a loss function, the model will measure the error. Then backwards pass will propagate the error to every individual node using backpropagation. The backpropagation will help to update the weights to get a better result.

The main benefit of having the deeper model is being able to do more non-linear transformations of the input and drawing a more complex decision boundary.

Pros

- The ability to learn and model nonlinear, complex relationships helps model the real-life relationships between input and output.
- Fault tolerance means the corruption of one or more cells of the ANN will not stop the generation of output.
- The ability to produce output with incomplete knowledge with the loss of performance being based on how important the missing information is.
- No restrictions are placed on the input variables, such as how they should be distributed.
- The ability to learn hidden relationships in the data without commanding any fixed relationship means an ANN can better model highly volatile data and non-constant variance.

Cons

- There is a risk of the gradient vanishing or explodes as it propagates backward which leads to a vanishing and exploding gradient.
- ANN cannot capture sequential information in the input data which is required for dealing with sequence data.
- The requirement of processors with parallel processing abilities makes neural networks hardware-dependent.
- The network works with numerical information, therefore all problems must be translated into numerical values before they can be presented to the ANN.
- Lack of explanation behind the solution generates a lack of trust in the network.

EMD-FBprofit-(LSTM)

Principle of the EMD algorithm:

Empirical mode decomposition (EMD) is an adaptive time-series decomposition method that smooths the nonlinear time series into feature components with different feature scales, namely, the intrinsic mode function (IMF) component, and the residual component $r(t)$. The decomposed resulting IMF represents the fluctuating components of different time scales implicit in the raw electricity consumption data, and its values and waveforms can well reflect the characteristics of the raw electricity consumption data. The residual term, on the other hand, represents the trend of the forecast series. The basic steps are as follows:

1. Find the maximum and minimum points of the original signal $x(t)$, fit the upper and lower envelope lines $e_{max}(t)$ and $e_{min}(t)$ of $x(t)$, and calculate the mean values of $e_{max}(t)$ and $e_{min}(t)$, $m(t)$:
   $$mt(t) = \frac{e_{max}(t)+e_{min}(t)}{2}$$

2. Calculate the difference between the raw data and the mean of the envelope lines, $d(t)$:
   $d(t) = x(t) - m_1(t)$

3. Determine the nature of $d(t)$; if it is satisfied that the numbers of zero crosses and extreme points in $d(t)$ are the same and the local maxima and minimal values, respectively, form envelope lines that are zero-mean, then $d(t)$ is an IMF component. Otherwise, the above steps need to be repeated for screening.

4. Define residuals $r(t)$:
   $r(t) = x(t) - d(t)$

5. Repeat filtering to the point where no new IMF can be filtered. Then the original signal $x(t)$ can be expressed as the decomposed IMF and the residuals of the sum.

   $x(t) = \sum IMF + r(t)$

Principle of the Fbprophet Algorithm:

The Fbprophet model is a time-series prediction tool proposed by Facebook in 2017, which carries out time-series prediction through time-series-based decomposition and machine learning fitting. The Fbprophet prediction process consists of four parts: model building, evaluation of predictions, problem solving, and manual checking, which work in a sequential loop until a suitable predictive model is obtained. The Fbprophet algorithm decomposes the time series y_t into three parts, as shown in the following equation:

$$y_t = g_t + s_t + h_t + \varepsilon_t$$

Where g_t is the trend term, which represents the trend of the time series on a nonperiodic basis; s_t is the seasonal term (period term); h_t is the holiday term; and $\varepsilon(t)$ is the error.

By fitting these three items through the submodel, it can be reasonably applied to a variety of time-series predictions with regularity and can identify and adjust the anomalies in the data, effectively dealing with the jump points and periodicity of the time series, while the Fbprophet algorithm also takes into account the seasonal and holiday effects that are of concern to the prediction model and is a convenient and efficient time-series prediction tool.

LS-SVM

LS-SVM is an alteration of the standard SVM and was improved by Suykens. The LS-SVM uses the least squares loss function to construct the optimization problem based on equality constraints. The least squares loss function entails only the solution of a linear equation series instead of the long and computationally difficult quadratic programing in the e-insensitive loss function of the original support vector machines. LS-SVM is generally used for the optimal control, classification and regression problems.

Comparisons of SVM and LS-SVM

- Least square SVM is the standard approach to approximate the solution of overdetermined systems when compared to normal SVM
- The hard margin SVM based on the Euclidean distance measure, may be comparable to LS-SVM for high dimensional small sample size data
- For an inseparable data LS-SVM is preferable to normal SVM

Pros

- Fast convergence in comparison to the standard SVM.
- High sensitivity in comparison to the standard SVM.
- Simple computation in comparison to the standard SVM .

Cons

- Difficult to select appropriate parameters for LS-SVM because there are no theoretical methods for doing so, which is a critical process in the accuracy of the regression. Optimization techniques should be utilized in order to manage this process properly.

Random forest

Random forests is an ensemble learning method used for classification and regression that operates in the way of constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned.Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

Pros:

- Easy to interpret
- Handles both categorical and continuous data well.
- Works well on a large dataset.
- Not sensitive to outliers.
- Non-parametric in nature.

Cons:

- These are prone to overfitting.
- It can be quite large, thus making pruning necessary.
- It can’t guarantee optimal trees.

DS Description

A - Active power: The power which is actually consumed or utilized in an AC Circuit is called True power or Active power or Real power. It is measured in kilowatt (kW) or MW. It is the actual outcomes of the electrical system which runs the electric circuits or load.

Positive sign: The component shows inductive behavior. When the active power flows from the “source” through the metering point and into the “load” we say the Active Power (Watts) are being DELIVERED. Therefore when the Active Power is being supplied by the “SOURCE” into the load it will be referred to as Delivered Power (Watts) and has a positive sign.

Minus: The component shows capacitive behavior. When the Active Power (Watts) flows from the “LOAD” through the metering point and into the “SOURCE” we say the Active Power (Watts) are being RECEIVED. Therefore when the Active Power is being supplied by the “LOAD” into the source it will be referred to as Received Power (Watts) and has a negative sign.

R - Reactive power: The power which flows back and forth that means it moves in both the directions in the circuit or reacts upon itself, is called Reactive Power. The reactive power is measured in kilo volt-ampere reactive (kVAR) or MVAR.

Positive sign: The component shows capacitive behavior. When the Reactive Power (Vars) flow from the “SOURCE” through the metering point and into the “LOAD” we say the Reactive Power (Vars) are being DELIVERED. Therefore when the Reactive Power is being supplied by the “SOURCE” into the load it will be referred to as Delivered Reactive Power (Vars) and have a positive sign.

Negative sign: The component shows inductive behavior. When the Reactive Power (Vars) flow from the “LOAD” through the metering point and into the “SOURCE” we say the Reactive Power (Vars) are being RECEIVED. Therefore when the Reactive Power is being supplied by the “LOAD” into the source it will be referred to as Received Reactive Power (Vars) and have a negative sign.

dts: The dates can affect the power consumption through weather and holidays.

real_val: The purpose variable shows the amount of electricity consumed.

weather parameters: As a part of the prediction there is a need to use some factors to predict the power usage.
We use temperature, humidity, wind direction, cloud type - because these factors affect the power consumption as the studies have shown.

EDA

As can be seen from the diagram 1.1, most of the values of "real val" are between 1 and 0 and rather close to 0. This means that they are very small values.

When we examined and compared the different graphs of each year, we realized that there is no difference between them. Moreover, we not only compared the years, but also looked to see if there was a difference between the generators, and they were similar as well. Therefore, we did not treat any of them differently.

According to the diagrams: 1.3, 1.4, 1.5, 1.6, comparing the different years and the different generators in terms of months, we realized that there is a correlation between the time of the month and the energy consumption. At the end of each month, the consumption is higher than at the beginning of the month. Therefore, it is necessary to take this into account in the process.

Also, we know from the diagram number 2 of the "Density-Weather Parameters" that if the density is similar, we can assume that the variable does not contribute much. Therefore, we only used the different variables.

<u>Pivot diagrams:</u>

Pivot diagrams allow us to see what features we need to use to predict power consumption as accurately as possible. As shown in diagrams 3.1, 3.2, 3.3, and 3.4, wind speed, humidity, time of day, and temperature affect electricity consumption, and we need to use them to predict electricity consumption.

<u>features and average power consumption:</u>

Moreover, in the graphs 4.1, 4.2, 4.3, 4.4, and 4.5 we examine each feature and in that feature we searched the values that were the same in order to replace the similar one due to the fact that we don't want that in one feature to have similar values for the different column, for example: time that can have a big influence on the final result.
Eventually in 4.1 we examine each hour of the day and compare it to all the other ones and as it seems, 3,2,4,5,22,23,24 are similar to 1, so we replaced them with 1 to give them the exact same significance. We did the same for 15 with 10 and 14 with 11.

In addition, we examined the average power consumption for each direction as seen in graph 4.2 and we saw that 1 and 4 held the same significance so we replaced 4 with 1.

As seen in graph 4.3 before we had textual weather categories like: “Light snow”, “overcast”, etc. In order to use all of the different categories in one column in the final data set, we switch them with representative numbers. While examining the result we noticed 9 & 12 hold the same significance as 6 so we replaced them with 6.

We did the same for 8 & 34 with 1; 18, 24 & 28 with 4; 20 & 25 with 2; 19 & 29 with 15 and 13 with 3.

For the last two graphs 4.4 and 4.5 there wasn't any similarity in the graph to that fact there wasn't a need to make any changes in the last features: wind direction and fiders.

Feature Selection:

Finally, after examining each feature and making changes that were made in order to fit the features to the model in the best way, the correlation of each feature with the features with 'real val' (electricity consumption) was determined. The result showed that the features that had the highest correlation were: 'idfiders', 'ind', 'dr', 'weather_categories', 'wind_dir', 'wind_speed', 'temp'.

Thus, the use of this feature in the model was directly related to the success of the power consumption prediction model.

Moreover, the future 'humidity' has a high correlation with electricity consumption - 'real val', but it also has a correlation above the value of 0.4 with temperature - 'temp', as a result of that fact we can omit it.

Algorithm selection:

In order to achieve the most accurate result, we tried several algorithms. Each yielded different results as seen in the following results table:

Algorithm	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error	R2 Score Error
Linear Regression	0.2797111153652	0.3358540598436	0.5795291708306	0.0726181624881
Decision Tree	0.0721798853649	0.0921125193761	0.3035004437824	0.7456529853571
Lasso Regression	0.2707442647546	0.3438501679392	0.5863873872613	0.0505387943777
Random Forest Regressor	0.0622560891443	0.0591386011296	0.2431842945785	0.8367026898258

As seen in the table, Random Forest Regressor algorithm yields the best results and therefore we went with this choice.

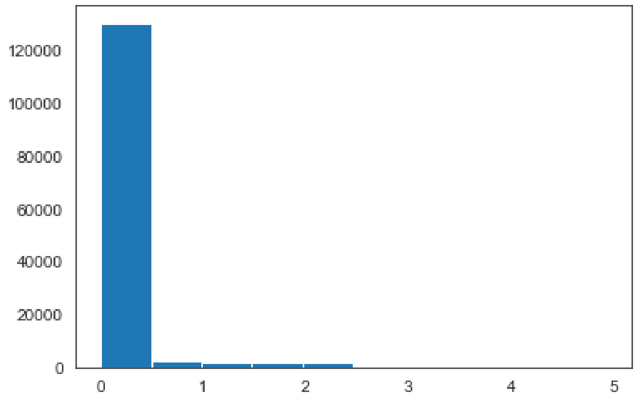
Random Forest result:

As seen in 5.1, 5.2 the result of reando forest was the best and the most accurate

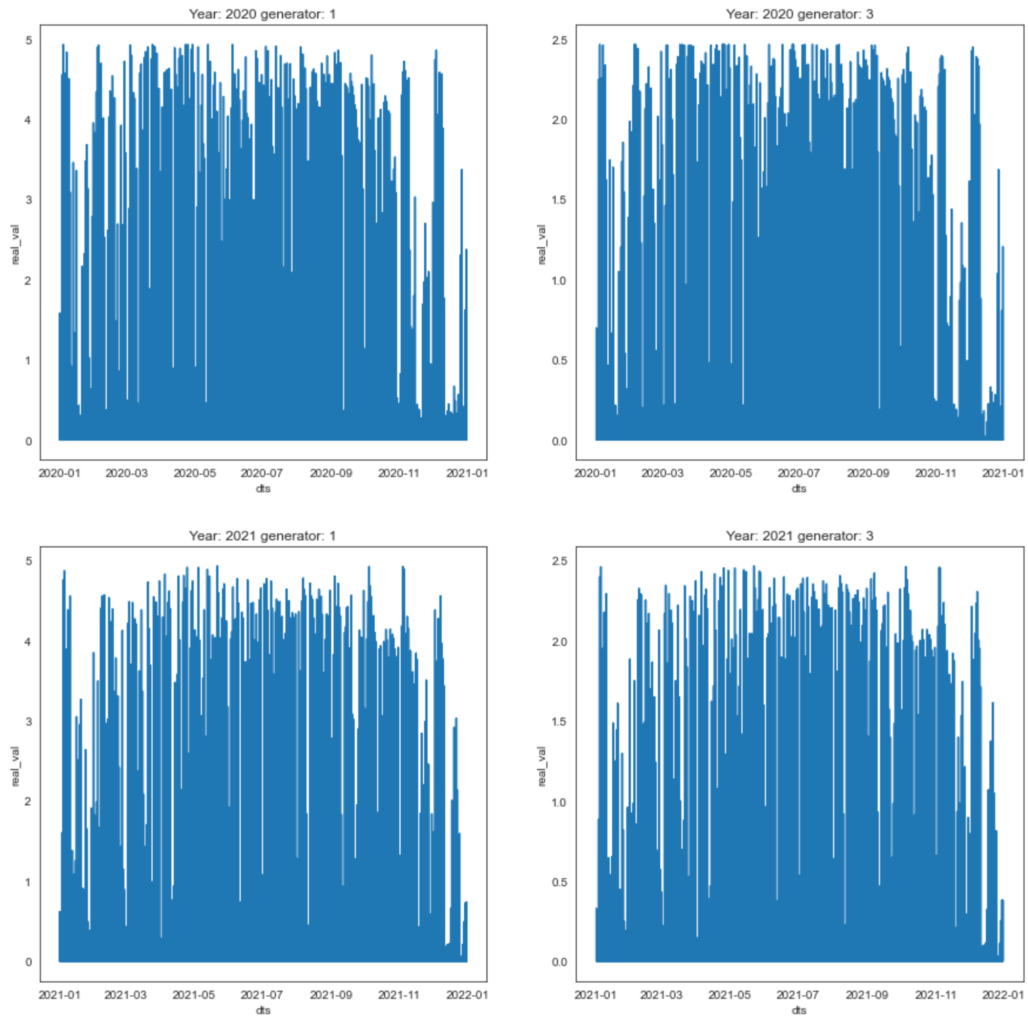
Mean Absolute Error	Mean Squared Error	Root Mean Squared Error	R2 Score Error
0.02593754371818532	0.01206558854464862	0.10984347292692735	0.9664821479987263

Moreover we used cross validation with kFold=10 to make sure the results are true. As seen in the table on 5.3 the results of r2 score are close to each other therefore the model that we chose was accurate.

Graph

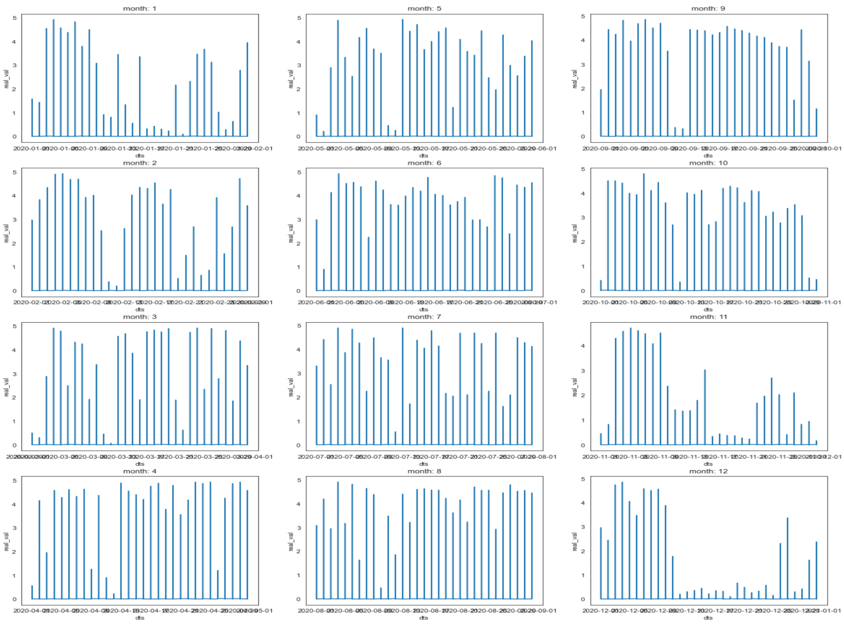


1.1 The distribution of ‘real_val’

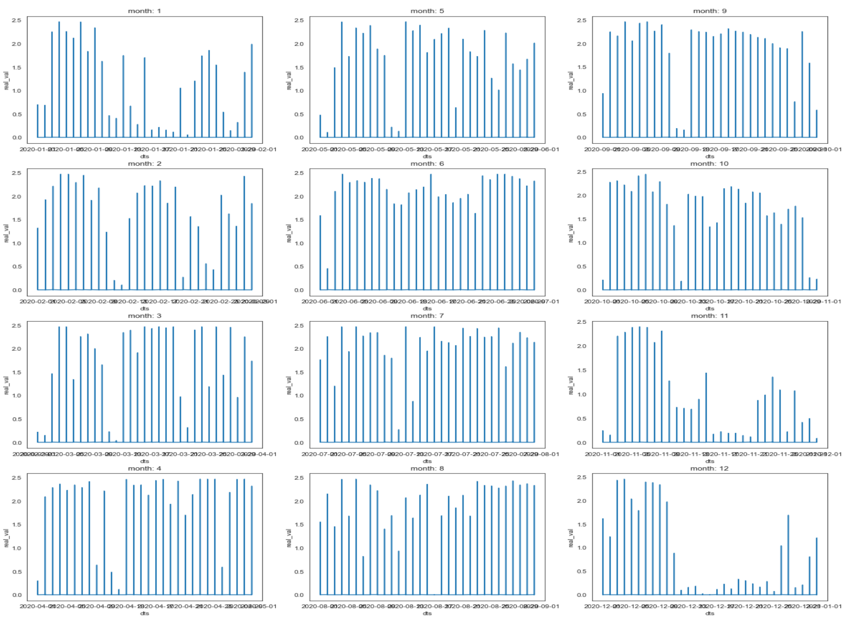


1.2 ‘real_val’ over the year

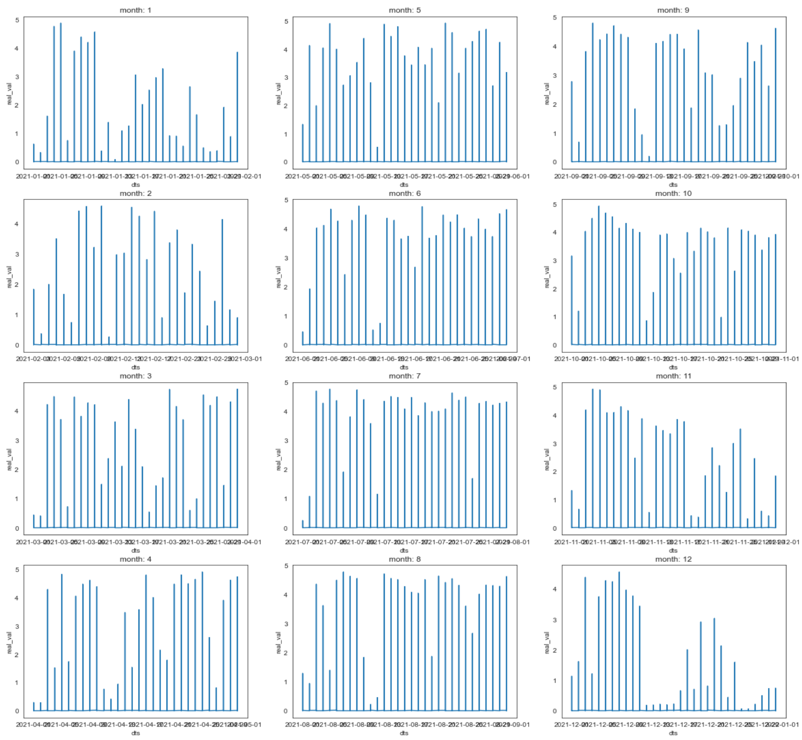
The ‘real_val’ over the months:



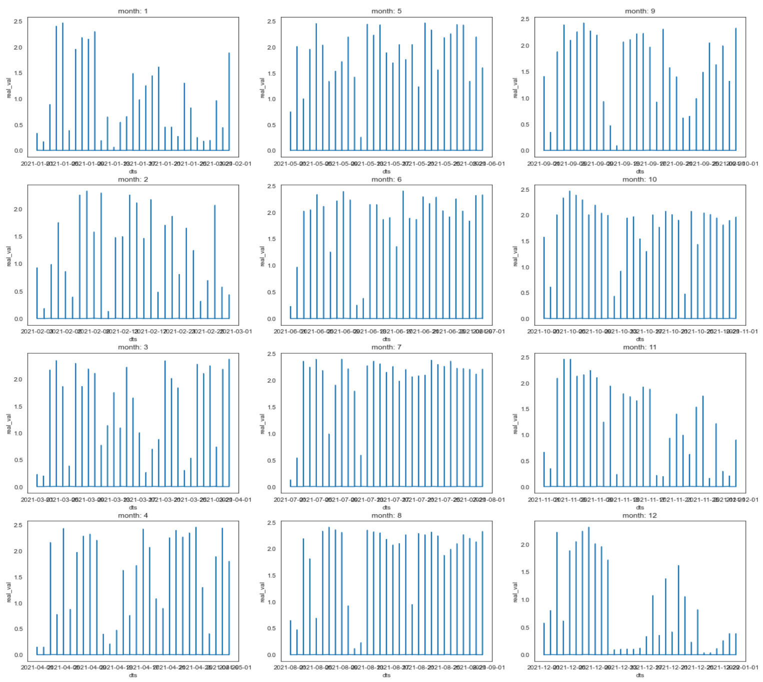
1.3 Year: 2020 ,generator: 1



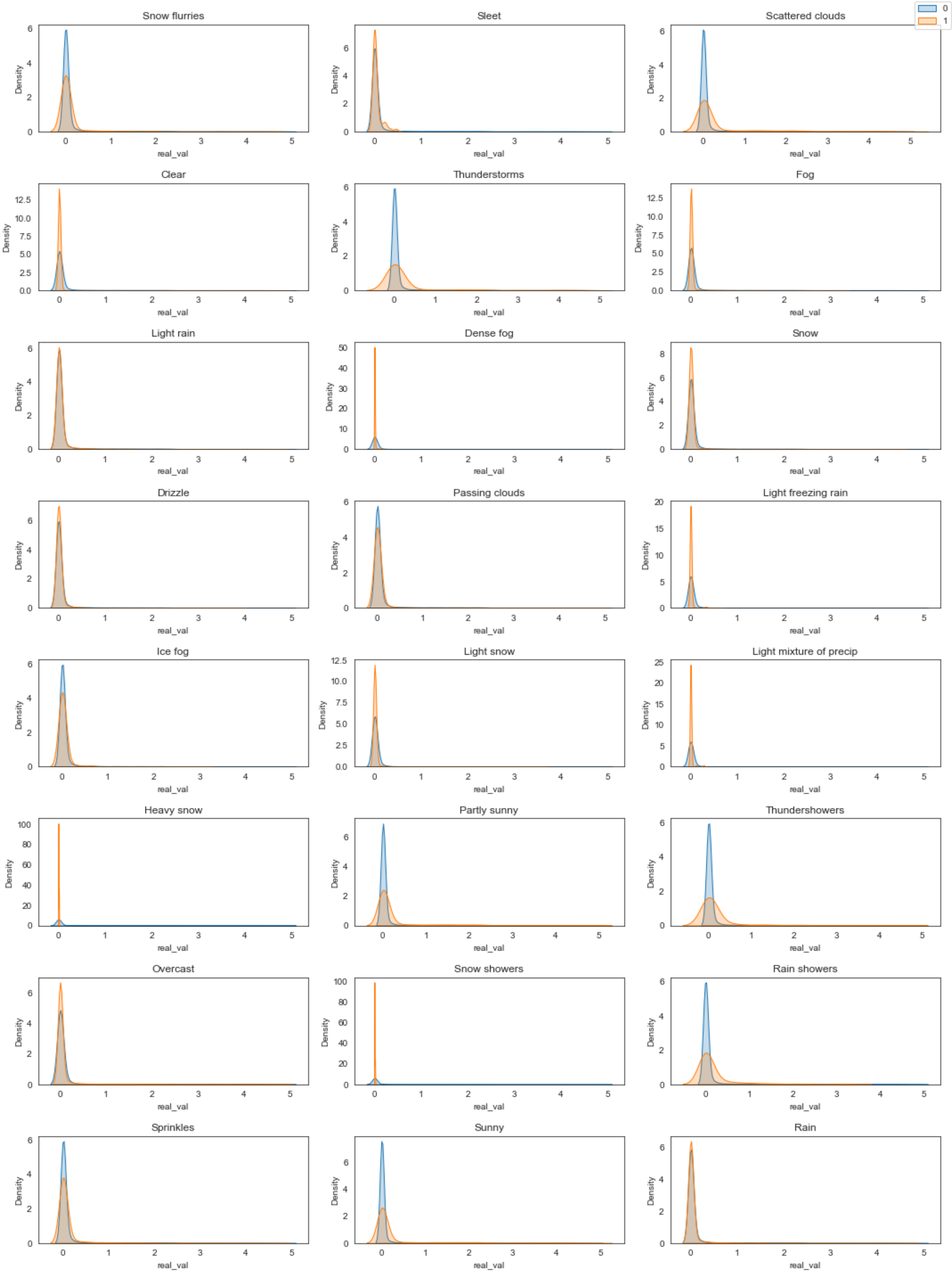
1.4 Year: 2020 ,generator: 3



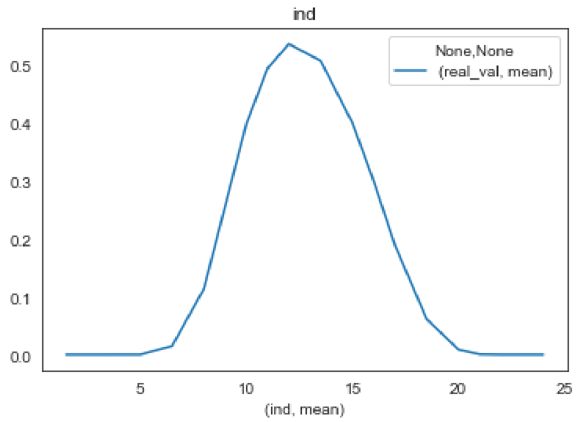
1.5 Year: 2021 ,generator: 1



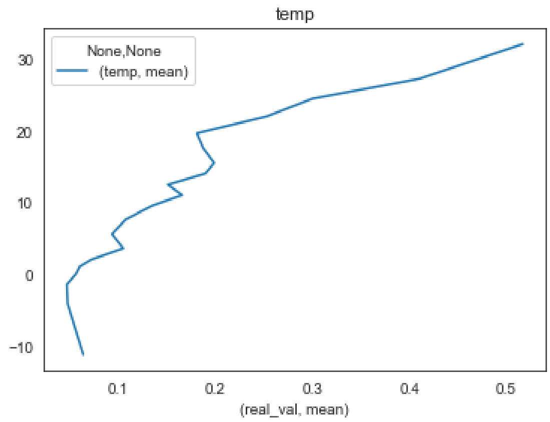
1.6 Year: 2021 ,generator: 3



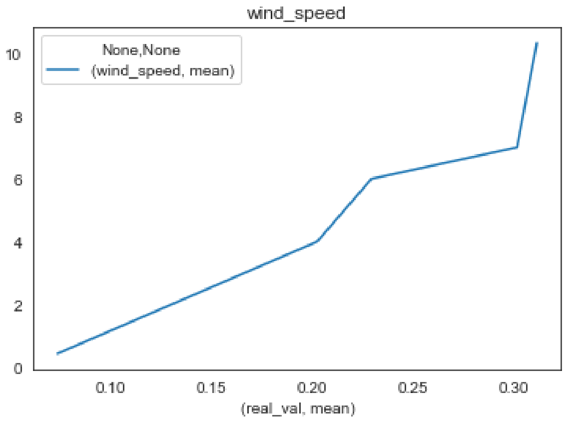
2 Density weather parameters



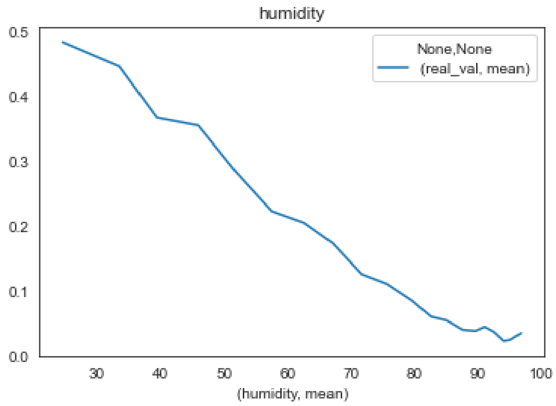
3.1 ‘ind’ (hour in the day) using pivot



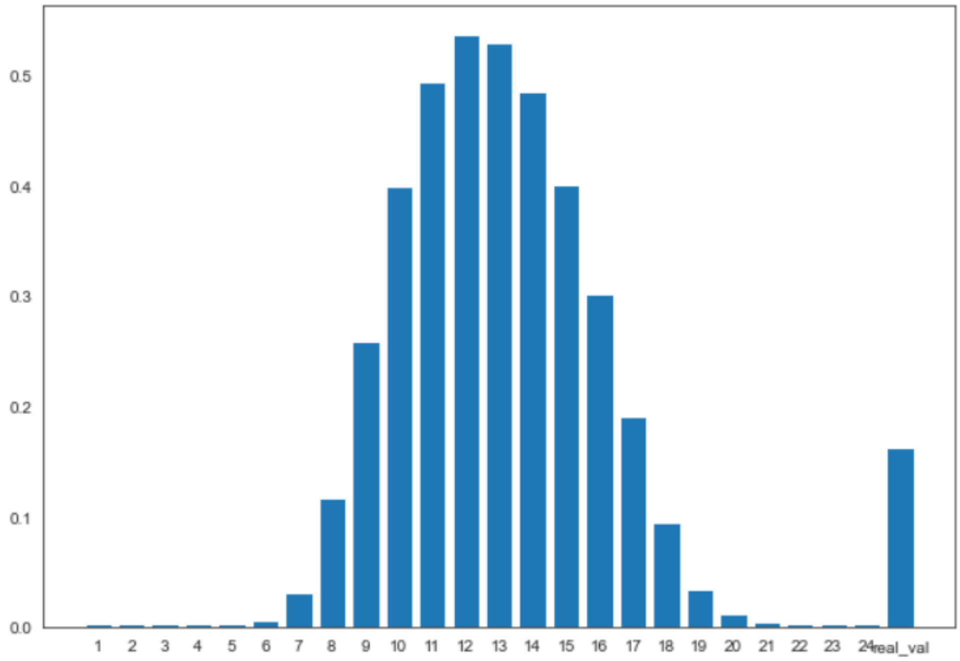
3.2 temperature using pivot



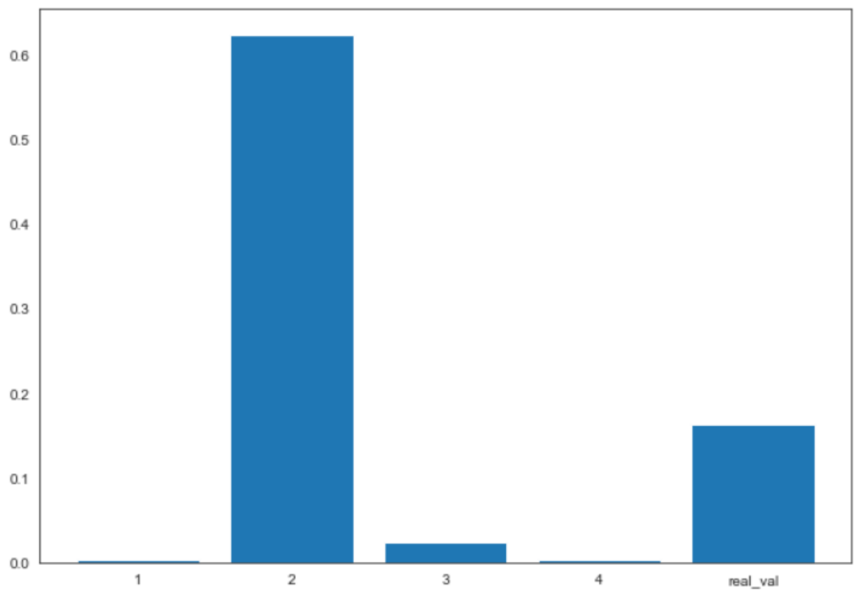
3.3 wind speed using pivot



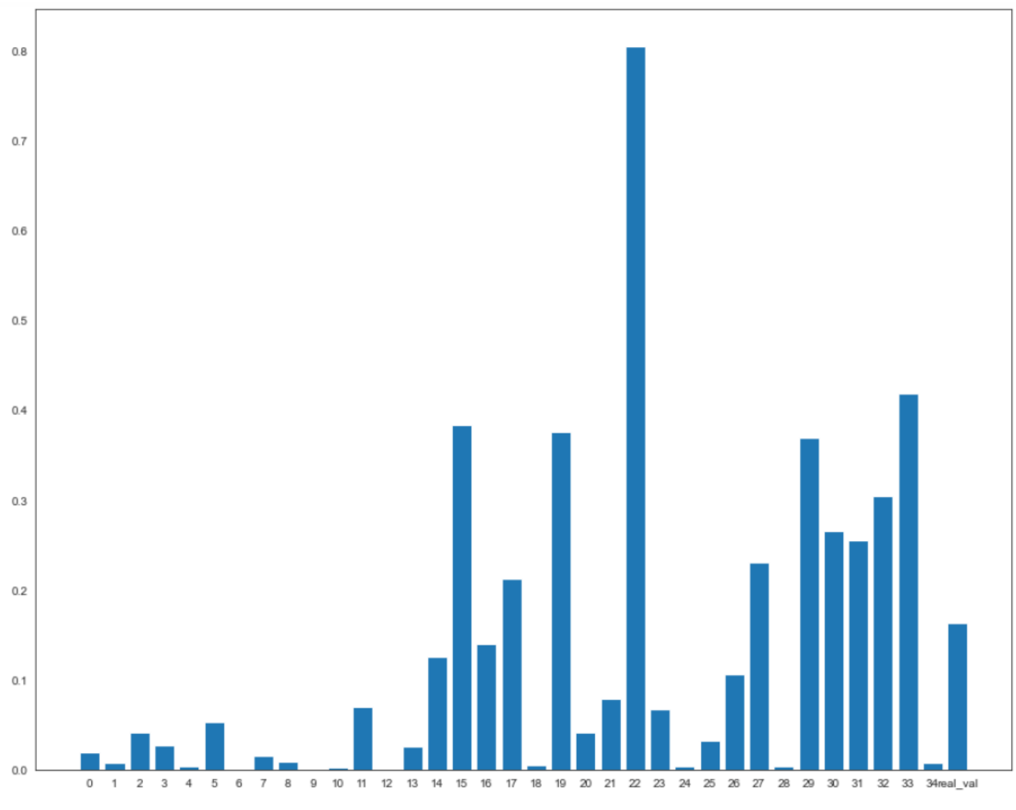
3.4 humidity using pivot



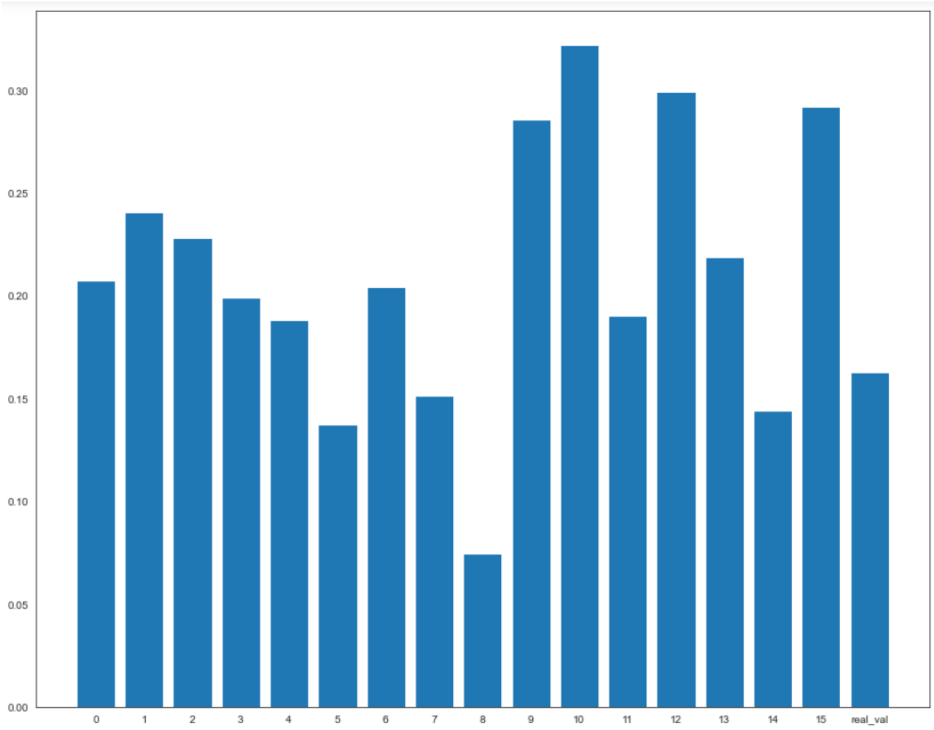
4.1 bar plot ‘ind’ (hour in the day) and average ‘real_ val’



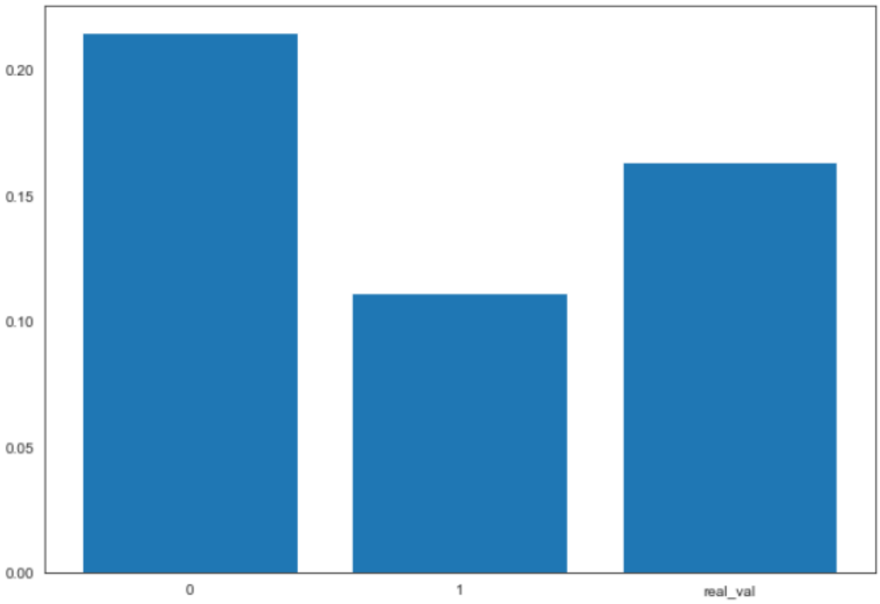
4.2 Box plot of the direction and its average power consumption



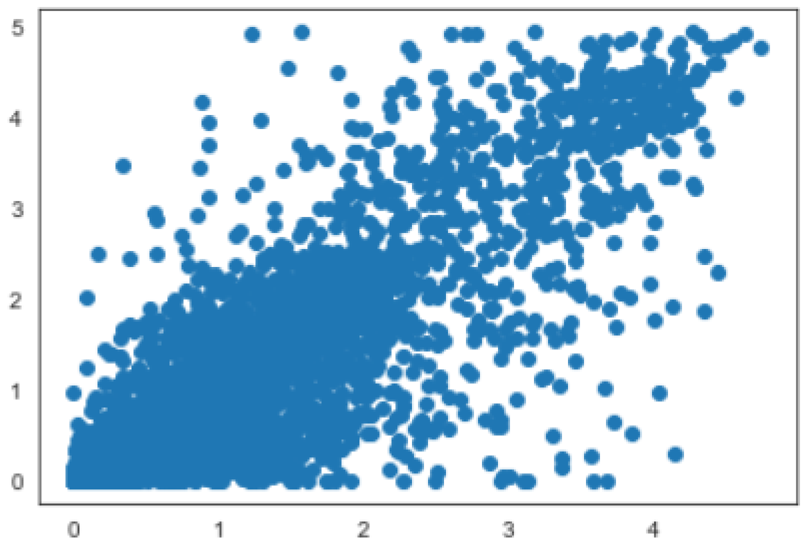
4.3 Box plot of each weather category and its average power consumption



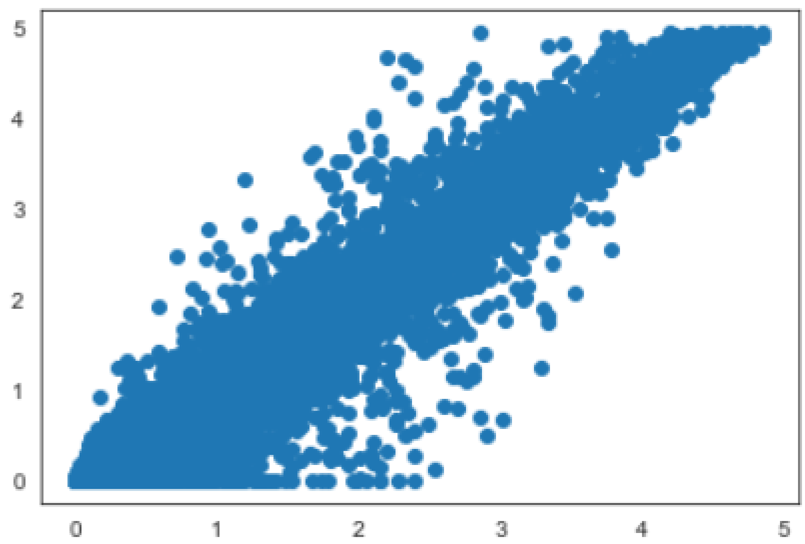
4.4 Bar plot real val and wind direction



4.5 Bar plot for each of the fiders and their average power consumption



5.1 Random forest test result



5.2 Random forest train result

	r2_score	MAE	MSE	RMSE
0	0.841274	0.061672	0.057483	0.239757
1	0.840083	0.061713	0.057914	0.240654
2	0.839448	0.061800	0.058145	0.241132
3	0.839565	0.061734	0.058102	0.241044
4	0.838819	0.061673	0.058372	0.241603
5	0.840926	0.061537	0.057609	0.240019
6	0.841347	0.061450	0.057457	0.239701
7	0.839990	0.061612	0.057948	0.240724
8	0.839853	0.061725	0.057998	0.240827
9	0.840154	0.061814	0.057889	0.240600

5.3 Cross validation for k = 10

References

Guorong Zhu, Sha Peng, Yongchang Lao,Qichao Su, and Qiujie Sun, "Short-Term Electricity Consumption Forecasting Based on the EMD-Fbprophet-LSTM Method", State Grid Zhejiang Economic Research Institute and The North China Electric Power University, 13-04-2021, https://www.hindawi.com/journals/mpe/2021/6613604

Power Consumption Predicting and Anomaly Detection Based on Transformer and K-Meanshttps://www.frontiersin.org/articles/10.3389/fenrg.2021.779587/full#e1

LSTM

Illustrated Guide to LSTM's and GRU's: A step by step explanation | by Michael Phi | Towards Data Science

Recurrent Neural Networks | Towards Data Science

ANN

Applied Deep Learning - Part 1: Artificial Neural Networks | by Arden Dertat | Towards Data Science

ANN vs CNN vs RNN | Types of Neural Networks (analyticsvidhya.com)

source for the drs

Microsoft Word - power Flow2.doc (landisgyr.com)