Determining the phase angle of power in electricity substations

David Teulings - s4576837

April 2023

1 Introduction

Alliander is an energy grid operator that seeks to play a role in the energy transition towards sustainable energy use. Energy grids are the central and peripheral infrastructure through which we power our homes. Even though the energy grid makes use of alternating current (periodically direction reversing current) we can think of energy as traffic that is being transported from the generators to the consumers.

Classically energy grids were relatively straight forward, having one (or just a few) centralised locations that provide energy, which makes most of the energy transport resemble one-way streets. However the rise of windmills, solar panels, and other sustainable technologies allows consumers to generate their own energy and give back the surplus to the grid. In essence creating a two-way street. Additionally, depending on weather- or energy consumption patterns this can completely flip the behaviour of a single consumer within a matter of hours. From needing energy to being an energy provider and vice verse. This gives rise to harder to predict decentralised energy generation.

Energy providers work on tight margins, since grid wide energy surplus cannot be stored, yet a grid wide shortage causes power blackouts which disrupt the consumers. Therefore grid managers, like Alliander, need to be able to predict energy needs accurately, however the decentralised generation adds complexity, which makes it harder to empirically measure the behaviours of the grid. This causes problems like allocating too much energy to a consumer, effectively wasting it, or providing an incomplete image about where Alliander still has capacity to expand the grid. Currently, for many locations in the grid, Alliander only uses absolute current measurements to observe the energy flow in the grid. This basically acts as a 'car counter' if we stick to the road analogy. Absolute current only measures the amount of cars that pass by, but does not measure their direction. With decentralised power generation on the rise these measurements in substations (dutch: *middenspanningsruimten*, or *MSR*) become less and less reliable. With decreased knowledge about power flow in these MSRs inefficiency will increase. Especially in this time of energy crisis solving this problem is in the interests of Alliander as a grid management company, but also in the interests of society as a whole. Solving preventable energy losses with a low cost machine learning method would be ideal.

1.1 Physics behind the Energy grid

Electrical grids are quite complex things. To start from the basics electric power (P) is measured in Watts, which is Joule per second. Watts over a certain amount of time is electrical Work (W), which in essence is electrical energy converted in to mechanical energy to make components move. Electric power is comprised of several elements, namely electric current (I) in Ampere (A) and voltage (U) in volt (V). The formula for electrical power (in direct current) is: $P = U \cdot I$. Imagine current to be the flow of power moving through the net, while voltage is the pressure urging the power to move.

As visible in Figure 1 if the voltage and current are in phase we transfer the maximal active power, and by extension maximal average active power. Average active power is the power which is used for Work. The angle between the voltage and current is called the phase angle.

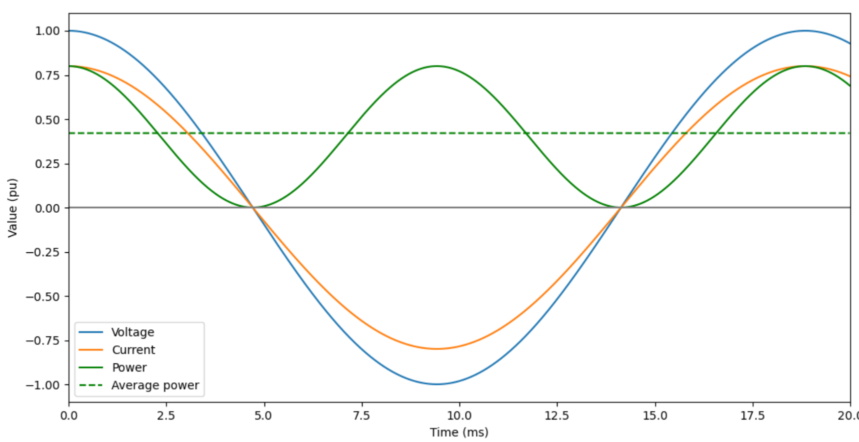


Figure 1: Example of voltage (blue), current (orange), and power (green) being in phase. Image taken from [4]

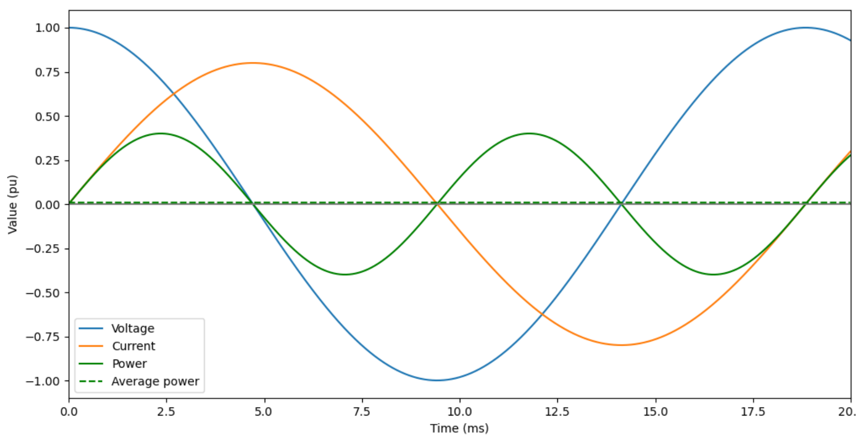


Figure 2: Example of voltage (blue), current (orange), and power (green) being in phase. Image taken from [4]

We can see that if the voltage and the current go out of phase, see Figure 2, the active power goes down. With in the worst case the phase being such that the active power is zero. The ways in which this phase angle can deviate is by power demand being higher than the generators can provide, or by the generators providing more power than demanded.

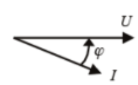
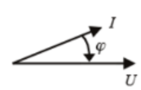
Power	**Generator-convention**	**Current and load angle**	**Diagram**
P positive	Active power generation		
P negative	Active power absorption		
Q positive	Reactive power generation (overpowered Generator)	Lagging current. Angle ϕ positive	
Q negative	Reactive power absorption (underpowered generator)	Leading current. Angle ϕ negative	

Figure 3: Table describing the meaning of different signs for P and Q within the grid. Table taken from [4]

The fact that the voltage and current are out of phase does not mean that there is less power overall. The power is still in the grid, just the transfer of this power has become less efficient. We can therefore differentiate between the apparent power (S), which is the total power in the system, and the active power (P), which is the power being actively transferred to consumers. As a counterpart to the active power we have the reactive power (Q), which is the remaining power that is not transferred, but is still in the system. In Figure 4 we can see the visual relation between S, P, and Q. Here we can calculate the active power as $P = U \cdot I \cdot cos(\phi)$, where ϕ is the angle separating the voltage and current phases. We see that if the phase angle is minimal ($\phi = 0$), then we can say that $S = |P|$. All the power is then actively used. Similarly $Q = U \cdot I \cdot sin(\phi)$. And finally we can say that $S = \sqrt{P^2 + Q^2}$. The ratio between active power and apparent power is called the power factor, which is dependent on the phase angle *phi*: $\frac{P}{S} = cos(\phi)$. This phase angle in essence is what we are interested in learning if we want to know the direction of power flow.

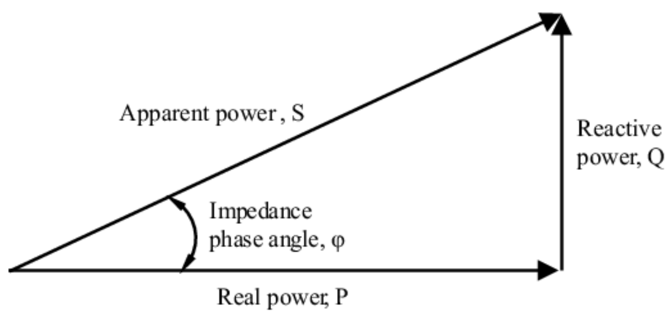


Figure 4: Relation between S, P, and Q. With the apparent power (S), active power (P), and reactive power (Q). Image taken from [3]

2 Main topics of research and experimentation

The scale of solving the problem of power flow inference in substations (MSR) is quite large. There are many different aspects to explore, such as the effects of weather patterns, geographical relations between MSRs, customer load patterns, and so on. We discussed to strip this problem down to a pure data based problem, since no prior research on this has been done so far by Alliander. Including even one of the previously mentioned aspects would increase the scope of this project beyond the time limit of the internship unfortunately. Also it is good to have an idea how machine learning models perform on the base substation data available before adding other features. Therefore we focus mainly on the raw substation data. The set we worked with contained 10 substations for which we had P and Q measurements. This data was presented as time series data, where each time step represents five minutes of measurements for a specific field, in a specific installation, in a specific substation. The station, installation, and field names are present as column features. Each field has four concrete measurements presented as column features: Absolute current (I) in Ampere, voltage (U) in Megavolt, active power (P) in Megawatt, and reactive power (P) in Megavolts-Ampere-Reactive. Further features included flags for potential measurement mistakes, which we ignored for this first basic exploration. For a small example of the data used refer to Figure 5.

	TA_B1_NAME	TA_B3_NAME	DATUM_TIJD	M_VALUE_P	M_VALUE_Q	M_VALUE_I	BEDRIJFSSPANNING
0	HrvH	V106	2021-05-14 16:50:00	0.00	0.00	0.00	0.0210
1	Nk	2.08	2021-10-04 16:55:00	0.30	-0.11	18.14	0.0105
2	HrvH	V102	2021-01-13 09:00:00	0.44	0.02	11.81	0.0210

Figure 5: Sample of the data. Where TA_B1_NAME and TA_B3_NAME refer to the Substation and field names respectively. DATUM_TIJD refers to the datetime. M_VALUE_P, M_VALUE_Q, M_VALUE_I, and BEDRIJFSSPANNING refer to P, Q, I and U respectively. With the units for P being in MWA, Q being in MVAr, I in A, and U in MV.

In short the predictive features that we have would therefore be I, U, the station, the field, and the datetime. The targets would be P and Q at a certain datetime.

2.1 Bayesian Machine Learning

The original concept for the internship was looking for some method which is able to combine real world knowledge with live power measurement data to determine the power flow. The starting point for this would be a Bayesian method, since generally Bayesian machine learning is able to incorporate prior knowledge. The goal would be to look for a method where we could insert the physical knowledge how the grid should behave in theory and apply it to a model. This was quite a daunting task and quickly the problem became much more about data representation than about Bayesian machine learning, however I will still introduce the concepts of Bayesian machine learning and Gaussian Processes.

Bayesian machine learning very simply put is the combination of machine learning with Bayesian statistics. It allows for the incorporation of prior knowledge and uncertainty in to the model, resulting in more robust and accurate predictions. Using Bayes' theorem the model will compute the posterior distribution using given data and prior knowledge (in the form of a prior distribution). The resulting distribution adds uncertainty to the result, allowing for more informed choices and flexible use. Within the field of Bayesian machine learning several different models exist, however for this internship I picked Gaussian Processes (GP).

Gaussian Processes are powerful models which are able to solve regression and probabilistic classification problems. We specifically are interested in GP regression. The main advantage we thought would be useful was its non-parametric nature. Meaning that, unlike most other machine learning models, the model is not limited to optimising the parameters of a single function, but will "calculate the probability distribution of parameters over "ll admissible functions that fit the data" [7]. This allows for the model to be powerful and flexible, but at the cost of computational power, because the optimisation process is performed over a covariance matrix of the parameters. A benefit here is that the GP models are able to perform well on little amounts of data.

With GPs we wanted to incorporate physical constraints of the grid in to a machine learning model, primarily focusing on Kirchhoff's current law. This law states that the sum of all currents going in and out of any point within an electrical grid should sum to 0. Because the voltages at fields within the same substation are generally constant we can extend this to say that the sum of all powers should be equal to 0. The problem that we ran in to while wanting to incorporate this as prior knowledge was the fact that we had to use outputs to pass a certain check (sum constraint), which we wanted to incorporate as a prior. This type of output feedback was not able to be cleanly implemented within the time span of this internship, if at all. One recent paper by Pilar and colleagues [5] touches upon the idea of having such a sum constraint for Gaussian processes and seems promising. The core idea is achieved by conditioning the prior distribution on the constraint fulfilment. However we were not able to implement a working prototype with these sum constraints.

2.2 Data exploration

Initial data exploration consisted of understanding the relation and the functions of the data. As mentioned earlier the data represents a time series of measurements of fields, in substations. We are ignoring the layer of installations, since all the substations in this dataset only had one installation each.

In Figure 6 we see and example station placed in Texel. Here we see the basic layout of a station with notable fields being V134 and V146 being incoming, V149 through V151 being reserved for testing, and the rest being outgoing. From the data it is not directly clear which field does what, so initial exploration assumed all fields to have equal behaviour.

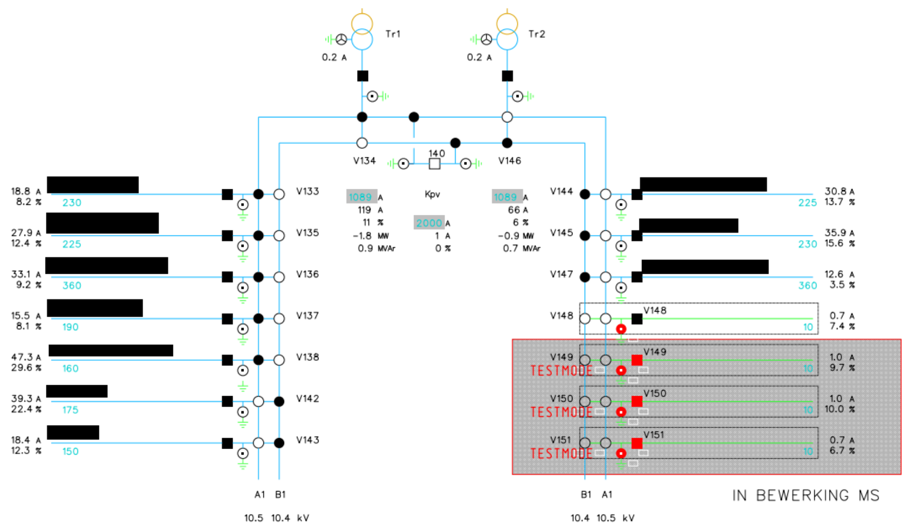


Figure 6: Layout of a single station in Texel. Note that V134 and V145 are incoming fields, which come from a higher voltage part of the grid and flow downstream to the other fields.

The initial idea for this internship focused in trying to see if we could learn some of the underlying physics governing the grid. Therefore we decided to first look at creating a general model and for a general model we needed to generalise the data. However rarely are stations ever the same. Therefore finding a way to shape a dataset containing multiple stations in such a way that you could represent the temporal relation between time steps as well as the spatial relation between fields of the same station was challenging. However before we got to that we first needed to decide what to do with holes in the data.

2.3 Missing values and imputation

The data contained a few NaNs at random moments. The Nans were could contain one or some combination of the features P, Q and I. Voltage was available at all time stamp. In the data available around 11.4% of the I measurements were NaN. P and Q contained 5.0% and 5.6% NaN entries. We decided to apply first calculate I where possible. This meant in each time step where P, Q and U were known I was computed according to the following equation:

$$I = \sqrt{\frac{P^2 + Q^2}{U \cdot \sqrt{3}}} \tag{1}$$

This equation was derived from equating the apparent power formulae containing P, Q, I, and U. $\sqrt{3}$ is used because of the three-phase AC system [4]. The I calculated here represents perfect power transference, which is rarely the case, but it would provide a reasonable estimate. Further missing values were imputed using the dataset wide averages.

2.4 Temporal or spatial data

During the internship quite some time was spent on thinking of way to combine the two aspects of temporal and spatial information. Because the substations vary greatly in shape it would be difficult to create a model that trains and tests easily on all stations. The smallest substation had 2 fields, whereas the largest had 25 fields. However most substations had between 12 and 18 fields. In essence we required a three dimensional data structure which had set dimensions for time and measurements, but allowed for variable length lists of fields. In the end I found perhaps one data structure which allowed such a set-up, but this data structure would never be interpreted correctly by any machine learning model known to me. And creating or modifying such a model would fall out of the scope of the internship.

In the end no proper representation of both aspects could be found, so the decision was made to run two parallel experimental pipelines. The first pipeline would serve as a baseline and would purely look at the temporal aspect of the data, while the second timeline would focus purely on the spatial relation between fields.

2.4.1 Temporal data

For the temporal data no spatial relations would be incorporated. This would be easy to implement since the data as was was already a times series. Therefore we just needed to add several features to represent the temporal relations. The most important feature here would be lagging the data. Such that every time stamp t would have the measurements at t, but also at t-1, t-2, until a certain point t-n. The lagged feature would be the current I, since this would be the primary indicator for P and Q. Voltage generally remains the same across the field. For the tests we ran a lag of 12 time steps, which equals one hour of data. For the very first time step in the data we don't have any information so we need to fill the lag with generated data of our choosing. However within this internship any lagged time step outside of the bounds of our data was filled with a 0.

To create the lagged data we first change the datetime format in to interpretable integer values, where the time of the first measurement at 2021-01-01 00:00:00 equals 0 and the time of the final measurement at 2021-12-31 23:55:00 equals 105119. Then we put all time steps in to chronological order.

2.4.2 Spatial data

For the spatial data the core idea would be a spatial representation of this time series data. In essence this means reshaping the data such that we still have time steps and datetimes as features, but that each time step would correlate with the P, Q, I, and U measurements of an entire station. What we did to create this dataset was loop through each time step and gather the measurements per station in to a list. This list would would then form the features. The features would be presented as I_0, I_1, etc with the index indicating the field.

Similar to the temporal data we first transcribe all datetimes to integers, then we order them chronologically. The fields and substations were ordered in order of occurrence, meaning there is no significance between a field being first or second and so on. The resulting dataset contained a lot more features per time step. Notably it contained $M \cdot F$ features, where M is the amount of measurements, or features, per field and F is the amount of fields. With just two measurements per field and, for example, 25 fields this is still manageable, but the amount of features will scale very quickly with added measurements or fields. Because of steep increase in features we split the data in to a feature and a target set, where the feature set contained all current and voltage measurements. And where the target set contained all P and Q measurements.

To address the difference in fields per station the dataset was designed to accommodate the largest substation. This results in sparse data, where some of the higher field numbers for certain stations were empty. These sparse areas were filled with 0 so that a model could interpret them as unused fields. For a representation of this data refer to Figure 7.

	DATUM_TIJD	STATION	I_0	U_0	I_1	U_1	I_2	U_2	I_3	U_3	...	I_20	U_20	I_21	U_21	I_22	U_22
0	0.0	Tex	0.000000	0.0105	76.790000	0.0105	44.380000	0.0105	52.980000	0.0105	...	0.0	0.0000	0.00	0.0000	0.00	0.0000
1	0.0	Hby	2.080332	0.0105	25.878577	0.0105	2.743656	0.0105	9.843535	0.0105	...	0.0	0.0000	0.00	0.0000	0.00	0.0000
2	0.0	Dtn	65.649983	0.0105	45.709995	0.0105	6.767125	0.0105	0.000000	0.0000	...	0.0	0.0000	0.00	0.0000	0.00	0.0000

Figure 7: Sample of the spatial data features. Where DATUM_TIJD is the integer representation of the datetime, STATION the substation label, and the features I_0 and U_0 through I_24 and U_24.

2.5 Metrics

This is a regression problem, since we are trying to predict the active power of certain fields depending on a certain moment in time, their current and voltage, and the current and voltage measurements of same-substation fields. There are several ways to measure the performance of regression models, but for this basic overview we decided upon the mean squared error (MSE), since it is a simple, yet intuitive measure for regression performance.

Furthermore Alliander realises that without any prior research in this field any model resulting from this internship would probably not predict these powers even close to perfect. Ideally the model would predict power flow within 10% of the true flow, but some people of the Advanced Analytics department also advised that simply being able to predict the sign of the power could already be helpful. Because they would be able to combine the predicted sign with other systems. Therefore we also added an extra metric of sign prediction accuracy. This metric can be used to see the problem as a binary classification task, where given them time, current and voltage the model has to predict whether the sign of the power will be positive or negative.

2.6 Experimental variables

Right now we have two parallel pipelines, temporal and spatial, as well as two metrics to judge the performance. However because this data was not quite explored in this method yet and a few more questions arose we decided to add more variables to experiment upon. The first variable, the one that we originally set out to test, whether we can usefully apply Gaussian Process regression to predict the power. Further questions that arose were whether there is a difference in prediction performance if the models are trained on individual stations or on combined data from multiple stations spanning different weeks of the year. Finally we also test if the knowledge about fields influences predictions.

2.6.1 Machine learning models

The original question in this internship revolved around the application of Gaussian Process (GP) regression, however due to the explorative nature of the problem and the data we settled on a very simple GP model. The GP used was straight out of the box from the SciKit-learn python package. We used a standard GaussianProcessRegressor with the default kernel. The default kernel was a ConstantKernel, with fixed value bounds, multiplied with a RadialBasisFunction (RBF) with fixed length scale bounds. For alpha we set it to 10 and did not change it during testing.

As a baseline model we decided to use XGBoostRegressor from the xgboost package[6] in python. This model was chosen because it was already widely used in Alliander's Advanced Analytics department as well as it being widely used in general due to it being able to handle large data sets and reaching good performances. For XGBoostRegression we used a standard squared error loss function.

2.6.2 Training on individual stations or weeks

Some other questions that arose were the impact of training on combined data or on individual stations, as well as training on different weeks of the year.

Despite the initial approach to this problem being a one-model-fits-all approach it seemed useful to at least test whether the performance of individual substation models would not do better, or even if the generalisable underlying physics could be learned from one substation alone. Therefore one of the added experimental variable would be training and testing on different stations. If this model was selected the model would perform a cross validation across the entire set of available substations. Training one one, testing on all the others with the selected machine learning model. The amount of training data used for station cross validation was just a few days, since GP regression was not able to handle more training data than that.

Finally the other variable we wanted to consider was whether training on a certain period of time would generalise well to other time periods. Therefore we added a variable that allowed for training and testing on different weeks of the year. This was compiled in to a cross validation where the chosen model would train on a period of time and test on all other periods of time in the year. The period we selected was two weeks. Therefore the model would cross validate across the year in train-test periods of two weeks.

2.6.3 Field knowledge as a feature

This variable was added towards the end of the internship. Jacco Heres put together a dataset labelling each field in the stations of the available data. These labels would correspond with a field being an incomming field, meaning coming from a higher voltage grid, or an outgoing field. Further labels were *defect*, meaning the measurement was unreliable, or *Nvt* meaning none of the previous labels was applicable. Due to the amount of features of the spatial data and the short amount of time left in the internship we only added these features to the temporal dataset, where they could easily be added as a single column.

Jacco's recommendation was to leave out all fields with an Nvt or defect label, however due to time constraints this was not implemented. The experimental variable managing this feature allowed you to add all labels and left the Nvt and defect fields as they were or not add them at all.

2.7 Testing

Testing was performed in two parallel pipelines. One for spatial data and one for temporal data. Both pipelines would perform a cross validation recording the performance based on sign accuracy and MSE. The pipelines allowed for the experimental variables to be tested as mentioned earlier. To summarise these variables as follows: Cross validation would be performed on spatial or temporal data. First you would select which machine learning model to use, either XGBoost regression or GP regression. Then you would choose to train on individual stations or on two week periods of time. Finally if the temporal data allowed for an additional choice of adding or not adding the field knowledge as a feature. Adding up all of these choices a total of 12 permutations were possible, resulting in 24 cross validation matrices (one for sign accuracy and one for MSE). Tests were performed on a device containing an Intel Core i7-7500U-processor with Intel HD Graphics 620 and 16 Gb of RAM. Individual cross validations runs took between roughly 10 minutes and 6 hours. Where the longest experiment to run was on spatial data trained on two-week cross validation with the Gaussian Process regressor.

3 Results

With 12 separate experiments we got 12 accuracy and MSE cross validation matrices, for which we can do several comparisons. Half of the results will be a square matrix of size 10, for 10 station cross validation, while the other half will be a square matrix of size 26, for two week cross validation. Two extra columns are added on the beginning and ending of each result matrix to make visual inspection easier. The first column simply contains the strings "Tested on n", where n is the cross validation fold. The final column contains the percentage of the value 0 occurring in the full feature set (All Xs for training and testing). Removing the first and last column we are left with square matrices containing the cross validation results.

To comprehend the large amount of raw result data we have we first try to aggregate some results. Therefore we calculate the average performances in each cross validation, averaging across testing per trained model, removing instances of self training (e.g. trained on station 1, tested on station 1). This results in a list of performances across all folds per experiment.

With these lists of performances per experiments we can perform some basic comparisons. As a tool for comparisons we use a two-sided t-test to make several comparisons. First of all we can compare the difference in performance between models indiscriminate of data sets. Secondly we can check for differences in performances within each data set, so within temporal or spatial data. Thirdly we can compare the difference in performances between both data sets, so between spatial and temporal. Finally we can see the difference in performance between adding the field knowledge as a feature and not adding it, which we can only compare within the temporal data.

However because of the sheer size of comparisons I will first display a visual inspection of a two specific sets of model predictions. For the sign accuracy comparisons across all experiments I will refer you to Appendix A .This will give an idea of the statistical tests and comparisons performed. For the same comparisons of the MSE I would refer you to the code. They have been moved to the appendices because are not easily presentable within this short report. The key results that should be noted is that from the metrics the average sign accuracy for GP regression was 0.82 and the average MSE 13.63. This means that 82% of the signs were correctly predicted in the binary classification of positive or negative. For XGboost regression the average accuracy was 0.74 and the average MSE 26.57. Interpreting the mean squared error as follows: an average error of about $\sqrt{13.63} = 3.69$ MWA and $\sqrt{26.57} = 5.15$ MWA. For context fields roughly range between 1 and 16 MWA in active power measurements.

3.1 Visual inspection of performance

In Figures 8 and 9 we show the predictions of both XGboost and Gaussian Process regression for a randomly selected field. The results display the true active power as the green line, the prediction of the GP as the orange line, and the prediction of XGboost as the blue line. To unify these visual inspections with the metrics the sign accuracy and MSE Figure 8 were 0.82 and 6.48 for the GP and 0.63 and 12.17 for XGboost.

Take note that visual inspection of other random stations and field often resulted in very similar predictions. Gaussian Process regression seems to most often predict 0, whereas the XGboostRegressor sometimes predicts nonzero values, but will often predict values around 0. Because of the size of the data and a lack of time it was not possible to find any discernible patterns for model prediction. There were a few fields where the models seemed to predict the power reasonable, but these were rare and we did not manage to find this visually similar prediction again.

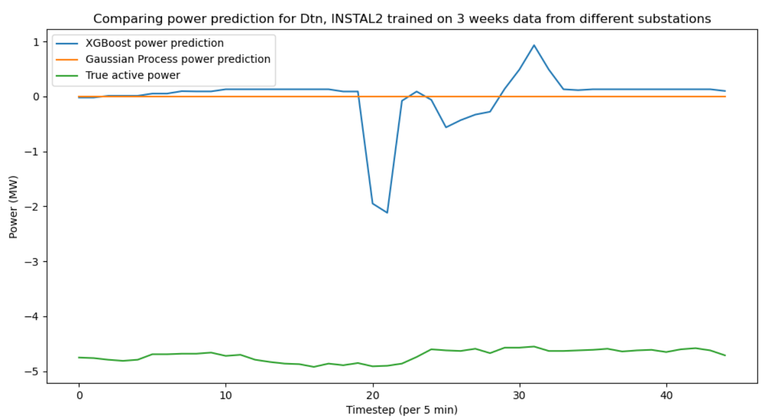


Figure 8: Visualised prediction for substation Dtn, field INSTAL2 for both GP and XGboost. The models were trained on 3 weeks of data containing all substations.

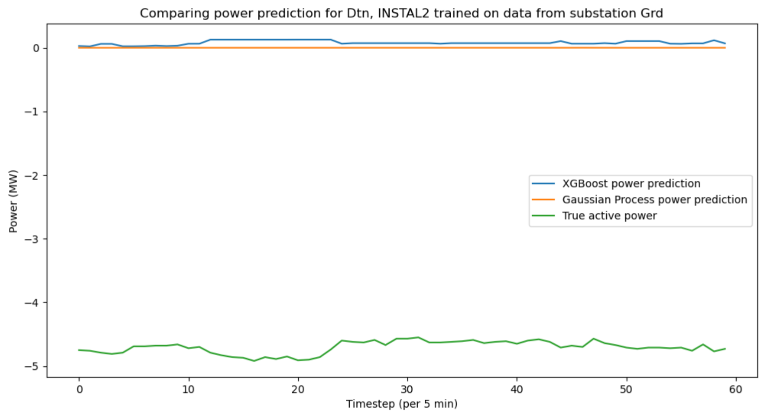


Figure 9: Visualised prediction for substation Dtn, field INSTAL2 for both GP and XGboost. The models were trained on substation Grd.

4 Conclusion & Discussion

The primary conclusion that we can draw is that GP regression does seem to outperform XGboost regression slightly. However this conclusion should be taken with a big pinch of salt. Looking purely at the metrics of specific experiments the results are good. A sign accuracy of 82% and an MSE of 13 could be acceptable. However it is undeniable that when we look past these two metrics the models are lacking. Gaussian Process regression often predicts zero to minimise squared error and XGboost regression does something similar. Therefore we were not able to achieve desirable results from this initial iteration of pipeline creation and model testing. Gaussian Processes are powerful models in theory, however trying to train them to be generalist models applying to all There lots of facets of improvement for this experimental pipeline, let alone this problem.

First of all one aspect to consider is the very simple versions of machine learning models used. Simple RBF kernel for GP and a squared error loss might not be powerful enough for the complexity of this problem. Equally likely, just having the absolute current and voltage (both purely positive values) to predict two features with both negative and positive possible values is probably not enough. A second question that arose at this point was the amount of data required for each machine learning model to be properly trained and not overfitted. Currently both models receive the same amount of data for training, however Gaussian Processes are known to be able to work well on smaller data sets due to their probabilistic predictions [2]. Because of memory limitations on the device the experiments ran on we had to downscale the amount of training data for GP regression when cross validating on stations. This meant that the same down scaling in training data was done for XGboost. This might have affected its performance.

Regarding the GP and XGboost specific implementations. At the time of writing no research has been performed on implementing specific physical constraints as priors. However a paper by Pilar and colleagues, 2022 [5] touches upon sum constraints. So further research starting from that paper could yield some fruit. For XGboost looking in to a custom loss function that works on a station wide level could prove beneficial. Such a function could be as simple as adding regular squared error loss to the minimisation of the sum of P predictions.

Dealing with the differently shaped stations was quite a challenge and during this internship and we did not manage to find a good solution. However towards the end we were informed of the existence of Evolving-Graph Gaussian Processes (e-GGP). This is a graph based approach towards Gaussian Processes is able to handle dynamic graph structures over time. The paper on Blanco-Mulero and colleagues [1] describes e-GGPs. It appears to be a promising model for future reasearch capable of combining the spatial and temporal data.

Final mention should go out to the data used and the experimental pipelines that were written during this internship. At the start of the internship we extensively discussed the underlying physics principles governing the energy grid, however on closer inspection it was hard to observe these principles in action. The primary principle of Kirchhoff's current law, where the sum of all powers in the same substation/installation should approach 0, was often violated. It is said that the measurements of active and reactive power would not be as accurate as for example the current and voltage measurements. The expected error here was around 10%, however often the errors got approached 30%. This discrepancy between theory and practice was noted by me, but I accepted it as part of the data. It is hard to argue that this can be fully representative. A deeper look in reality checking the data and measurements would be advisable, however it could be that these reality checks have already been performed by error flagging measurements at certain time steps and fields. Some of these flags were available in the data, but were not used in this internship. Regarding the experimental pipelines created there is still a large possibility of bugs. Up until a week before presenting the results of the internship bugs were still being found and fixed. We are fairly sure that currently there is a bug in the experimental pipeline when choosing between creating a data split for cross validation between stations and periods of two weeks. Furthermore in the code written to create the spatial dataset it seems that the order of fields does not always match the field order in the temporal data. This is something that caused some confusion when trying to reconstruct predictions. These are the two main bugs that have been left unsolved at the time of writing.

5 References

[1] David Blanco-Mulero, Markus Heinonen, and Ville Kyrki. *Evolving-graph gaussian processes.* Nov. 2021. URL: https://arxiv.org/abs/2106.15127.

[2] Christopher J. Burges. "Dimension reduction: A guided tour". In: *Foundations and Trends® in Machine Learning* 2.4 (2009), pp. 275–364. DOI: 10.1561/2200000002.

[3] Shahriar Rahman Fahim et al. "A Novel Fractional Order Power Factor Measurement". In: *SN Applied Sciences* (Dec. 2019). DOI: 10.1007/s42452-019-1652-4.

[4] Phase to Phase. *Phase to Phase - Netten voor distributie van elektriciteit.* Alliander, 2020. URL: https://www.phasetophase.nl/boek/index.html.

[5] Philipp Pilar et al. *Incorporating sum constraints into multitask gaussian processes.* Feb. 2023. URL: https://arxiv.org/abs/2202.01793.

[6] *python package - xgboost 1.7.5 documentation.* URL: https://xgboost.readthedocs.io/en/stable/python/python_intro.html.

[7] Hilarie Sit. *Quick start to gaussian process regression.* Oct. 2021. URL: https://towardsdatascience.com/quick-start-to-gaussian-process-regression-36d838810319#:~:text=Gaussian%20process%20regression%20(GPR)%20is,uncertainty%20measurements%20on%20the%20predictions..

6 Appendix A

6.1 Model comparison

All of these comparisons were performed across both spatial and temporal data and compared GP against XGboost.
Significant difference:

- spatial two weeks GP **outperformed** spatial two weeks XGboost
- temporal two weeks GP **outperformed** temporal two weeks XGboost
- spatial station GP **outperformed** spatial station XGboost

Insignificant difference (described as *equalled* for lack of a better term):

- temporal two weeks field knowledge GP **equalled** temporal two weeks field knowledge XGboost
- temporal station field knowledge GP **equalled** temporal station field knowledge XGboost
- temporal station GP **equalled** temporal station XGboost

6.2 Field knowledge addition comparison

All of these comparisons were performed within the temporal data and compared field knowledge against no field knowledge.
Significant difference:

- two weeks GP **outperformed** two weeks field knowledge GP
- two weeks field knowledge XGboost **outperformed** two weeks XGboost

Insignificant difference:

- station field knowledge GP **equalled** station GP
- station field knowledge XGboost **equalled** station XGboost

6.3 Within temporal data comparison

All of these comparisons were performed within the temporal data and compared two weeks training against station training.
Significant difference:

- two weeks field knowledge GP **outperformed** station field knowledge GP
- two weeks GP **outperformed** station GP
- station XGboost **outperformed** two weeks XGboost

Insignificant difference:

- two weeks field knowledge XGboost **equalled** station field knowledge XGboost

6.4 Within spatial data comparison

All of these comparisons were performed within the spatial data and compared two weeks training against station training.
Significant difference:

- station GP **outperformed** two weeks GP
- station XGboost **outperformed** two weeks XGboost

No insignificant differences.

6.5 Temporal-spatial data comparison

All of these comparisons were performed across both spatial and temporal data.
Significant difference:

- spatial two weeks GP **outperformed** temporal two weeks GP
- spatial station GP **outperformed** temporal station GP
- spatial station XGboost **outperformed** temporal station XGboost

Insignificant difference:

- spatial two weeks XGboost **equalled** temporal two weeks XGboost