University of Liège
School of Engineering and Computer Science

Machine learning applied to evaluation of power grid reliability

Master's thesis completed in order to obtain the degree of Master of Science in Computer Science by Catteeuw Tristan

Academic year 2021-2022

Acknowledgement

This research has been supported by the EPOC 2030-2050 project (Belgian energy transition funds).

I would like to give special thanks to Mr. Wehenkel and Mr. Karangelos for their valuable input throughout this project.

Contents

Chapter 1

Problem Statement and necessary background

1.1 Research context

Achieving a sustainable energy transition is one of the main challenges in today's world. Climate change has encouraged an increase in the usage of renewable energy. In 2021, Belgium's energy mix contained 22.7% renewable energy [1]. More recently, the geopolitical context with the war in Ukraine has encouraged a reduction in the usage of gas as a combustible.

In this context, it is important to be able to plan ahead the consequences of long-term decisions, such as building new power generation plants. Understanding the reliability of the power grid that would result from long term decisions is a difficult task, but one of vital importance. Current research uses models of planned future grid on which different scenarios are simulated. These simulations give an indication about the studied power grid configuration and its reliability.

This work contributes to the EPOC project [2]. The EPOC project combines the expertise of 14 Belgian academic partners to improve the current state-of-the art energy models, providing a consistent calculation for the long-term energy future in Belgium.

The remainder of this chapter serves as a formal problem statement. First, the nature of forecasts and scenarios, as well as how they are created is detailed. Afterwards, the metrics used to evaluate the scenarios are motivated and explained. Finally, the motivations behind the use of machine learning in this task are introduced.

1.2 Forecast and scenario generation

Forecasts consist of a day-ahead prediction for the power grid, while a scenario is the next-day realisation. The forecasts consist of wind and solar power generation prediction as well as load demand prediction in each node. In other terms, the forecast estimates how much power will be generated by wind and solar production at each node as well as the power usage at each node. A node can be seen as an actor of the power grid, like a generation plant or an apartment building. The nodes are connected together by lines which bring power from the generation node to the consumption node. The scenarios are given as a deviation from the forecast.

Forecasts are obtained by utilizing historical data. A year is taken as a reference, 2015 in this case, where information about wind production, solar production and load is available. To obtain forecasts for a future period, the values from that year are rescaled to reflect the estimated change in magnitude between both dates for each energy generator. This is performed by the Belderbos Belgian model [3]. The model has the particularity that, after the rescaling, a high percentage of the power generation capacities come from renewable energies. Another particularity is that nuclear generators are replaced by gas-fired power plants.

Possible realisations of next-day scenarios are generated based on the forecasts using normalizing flows introduced in [4]. Normalizing flows are generative models allowing to express probability distributions in a simple manner. The scenario generation tool used in this research has been introduced in [5].Scenarios represent the situations that effectively could happen the next day. Performing evaluation on a wide array of generated scenarios is crucial to get a precise evaluation of the overall reliability of the grid.

1.3 Evaluation metrics and grid modeling

1.3.1 Objective function

To evaluate the scenarios, the primary metric is the following cost function :

$$\min_{u,e,w,p_g,f,\theta} \left\{ \sum_{n\in\mathcal{N}} \left[u_{n,0} + \sum_{c\in\mathcal{C}\setminus 0} \frac{u_{n,c}}{|\mathcal{C}|-1} \right] + \frac{0.1}{|\mathcal{C}|} * \sum_{s\in\mathcal{S}} \left[(e^+_{s,0} + e^-_{s,0}) + \sum_{c\in\mathcal{C}\setminus 0} \frac{(e^+_{s,c} + e^-_{s,c}}{|\mathcal{C}|-1} \right] \right\} \quad (1.1)$$

Preserving the balance of energy production and demand is crucial. Sometimes, preserving this balance requires to take unwanted actions that cause issues on the grid, but is the only way to prevent more catastrophic failures. The equation considers two types of unwanted actions :

- $u_{n,}$ represents load shedding at node n. When total load is greater than available production, load shedding involves dropping energy supply to a certain number of consumers, until the system's production and load are balanced again. In this situation, some consumers would get cut from the grid and not get their desired supply. This is why this solution is only used in last resort.
- $e_{s,}$ represents the storage redispatch of the storage device s. This is another solution to deal with imbalances in the system that doesn't involve as many issues as load shedding. It consists in utilizing stored energy to satisfy part of the load. As this is less problematic, it is scaled down in equation 1.1 compared to load shedding. However, as storage capabilities are limited, it is sometimes not possible to restore the balance using this method. This is when load shedding becomes unavoidable.

Equation 1.1 accounts for the situations in the preventive case and in the post-contingencies $c \in \mathcal{C} \setminus 0$. The situations can be described as the working state of a power grid. The preventive one is where all lines are operating as usual. Post-contingencies evaluate $N-1$

situations of the power grid, in which one line is down and thus cannot be used to transport power. The evaluation of contingencies is scaled down by the number of contingencies -1, to prevent them from having a disproportionate impact on the overall cost. The preventive situation is usually referred to as contingency 0.

Additionally to the load shedding for all nodes n and the storage redispatch for all storages s, the equation is also minimized with regards to $\hat{p}_g$ - the hydro/thermal power generation edispatch, w - the renewable power generation curtailment and f - the branch flow. They do not appear in the equation 1.1 as they are not classified as unwanted actions, but they indirectly influence its value. Overall, the result of the equation is expressed in energy volumes as a weighted sum of all the load power demand that needs to be shed, and the power redispatched from the storages.

1.3.2 Grid modelling and constraints

The power produced needs to move in the grid to be able to satisfy load demand at a certain destination node n that requires it. The unability of the grid to move enough power can be a reason for a high value of equation 1.1.

Consider the power flow f_l for each line l. For all nodes n and considering the contingency 0 :

$$\sum_{l\in\mathcal{L}_n} f_{l,0} = \sum_{g\in\mathcal{G}_n} [\hat{p}_g + ((p^+_{g,0} + p^-_{g,0})] + (r_n - w_{n,0}) - (d_n - u_{n,0}) - \sum_{s\in\mathcal{S}_n} [\hat{e}_s - (e^+_{s,0} - e^-_{s,0})] \tag{1.2}$$

Where $\hat{p}_g$ the hydro/thermal power generation dispatch of generator g, $\hat{e}_s$ the energy storage dispatch of storage s, r_n the renewable power generation, d_n the load demand, $p^{+/-}_{g,0} \geq 0$ the hydro/thermal power generation redispatch, $e^{+/-}_{s,0} \geq 0$ the energy storage redispatch, $w_{n,0} \geq 0$ the renewable power generation curtailment, $u_{n,0} \geq 0$ the load demand shedding and $f_{l,0}$ the branch power flow.

This equation can be adapted for all nodes n and contingencies $c \in \mathcal{C} \setminus 0$:

$$\begin{aligned}
&\sum_{l\,\in\mathcal{L}_n} f_{l,c} && (1.3)\\
&= \sum_{g\in\mathcal{G}_n} [\hat{p}_g + ((p^+_{g,c} + p^-_{g,c})] + (r_n - w_{n,c}) - (d_n - u_{n,0} - u_{n,c}) - \sum_{s\in\mathcal{S}_n} [\hat{e}_s - (e^+_{s,0} - e^-_{s,0}) - (e^+_{s,c} - e^+_{s,c})]
\end{aligned}$$

Where all the constraints of equation 1.2 still hold for all contingencies. Load shedding for a post-contingency c is represented as $u_{n,0} + u_{n,c}$ as a contingency can only add more load shedding compared to the preventive case.

Equation 1.3 is subject to constraints :

$$f_{l,c} = (\frac{a_{l,c}}{X_l}) * \sum_{n\in\mathcal{N}} \lambda_{l,n} * \theta_{n,c} \tag{1.4}$$

$$-\bar{f}_{l,c} < f_{l,c} < \bar{f}_{l,c} \tag{1.5}$$

Where $\lambda_{l,n}$ the branch incidence, $a_{l,c}$ the branch availability (0 if $l = c$ and 1 otherwise), X_l the reactance and $\bar{f}_l$ the grid capacity are all parameters / constraints. $\theta_{n,c}$ is a variable representing the voltage angle.

The meaning of equations 1.3, 1.4 and 1.5 is straightforward. Equation 1.3 indicates that the power flow is the sum of all the power that passes through the line. Equation 1.4 constraints this value by forcing it to 0 if the branch is unavailable, and otherwise limiting its value to what is physically achievable given the reactance of the line, the voltage angles and the branch incidences. Equation 1.5 further constraints the value by limiting the value to the absolute line capacity.

Further constraints are applied on the hydro/thermal generators. For all generators $g \in \mathcal{G}$ and all contingencies $c \in \mathcal{C}$:

$$v_g * P_g \leq \hat{p}_g + (p^+_{g,c} + p^-_{g,c}) \leq v_g * \bar{P}_g \tag{1.6}$$

Where v_g generator on/off status indicator, $\hat{p}_g$ generator day-ahead dispatch, $P_g/\bar{P}_g$ generator min./max. power output bounds and r_n realized RES power generation. The equation constraints the generators to operate within their generation bounds.

Furthermore, the curtailment cannot exceed the generation. For all renewable (RES) generators at nodes $n \in \mathcal{N}$ and contingencies $c \in \mathcal{C}$:

$$0 \leq w_{n,c} \leq R_n \tag{1.7}$$

The load shedding cannot be negative or exceed the total system load. More formally, for all nodes $n \in \mathcal{N}$ and all contingencies $c \in \mathcal{C} \setminus 0$:

$$0 \leq u_{n,0} + u_{n,c} \leq d_n \tag{1.8}$$

$$0 \leq u_{n,c} \tag{1.9}$$

Concerning the storage devices, for all storages $s \in \mathcal{S}$ and all contingencies $c \in \mathcal{C} \setminus \prime$,

When in charging mode, $\hat{e}_s \geq 0$:

$$0 \leq e^+_{s,0} + e^+_{s,c} \leq \hat{e}_s \tag{1.10}$$

$$0 \leq e^+_{s,c} \tag{1.11}$$

$$0 \leq e^-_{s,0} + e^-_{s,c} \leq 0 \tag{1.12}$$

$$0 \leq e^-_{s,c} \leq 0 \tag{1.13}$$

When in discharging mode - else:

$$0 \leq e^+_{s,0} + e^+_{s,c} \leq 0 \tag{1.14}$$

$$0 \leq e^+_{s,c} \leq 0 \tag{1.15}$$

$$0 \leq e^-_{s,0} + e^-_{s,c} \leq |\hat{e}_s| \tag{1.16}$$

$$0 \leq e^-_{s,c} \tag{1.17}$$

1.3.3 Security margin

The margin is a second evaluation metric that is calculated in the event when there is no or very temporary load shedding / storage redispatch. More contretely, this value is calculated to give more insights into scenarios where the value of 1.1 is below 0.0005. It represents the additional load that can be added to the system while keeping the value of equation 1.1 below the threshold. As such, it can be seen as a distance to risk metric.

When considering margin, for all nodes $n \in \mathcal{N}$ and contingencies $c \in \mathcal{C}$:

$$\sum_{l \in \mathcal{L}_n} f_{l,c} = \sum_{g \in G_n} [\hat{p}_g + ((p^+_{g,c} + p^-_{g,c})] + (r_n - w_{n,0}) - (d_n + m_n) - \sum_{s \in S_n} \hat{e}_s \tag{1.18}$$

Notice the similarity between equation 1.18 and equations 1.2 and 1.3. In equations 1.2 and 1.3, load shedding is deducted from the system load, as it represents load demand that is no longer satisfied to preserve. The margin is the opposite, as it adds fictive load to the system while maintaining balance.

Where the goal is to maximize the margin:

$$\max_{m,w,p_g,f,\theta} \sum_{n \in \mathcal{N}} m_n \tag{1.19}$$

Subject to equations 1.4 to 1.7

1.4 Goals of machine learning

Machine learning has been applied to the topic of energy systems reliability in a variety of ways. In fact, 366 articles has been posted on this subject between January 2000 and October 2019 [6]. For this work, three main objectives where identified.

First, machine learning can be a way to speed up the evaluation process. To compute the result of equations 1.1 and 1.19, scenarios need to go through a simulator, which is a time consuming task as the min operation of 1.1 and the max of equation 1.19 are of high complexity. Furthermore, equation 1.1 requires the evaluation of each of the 46 possible post contingencies. A machine learning model's prediction can be computed in a lower complexity. As such, the first goal of machine learning is to study how well it can predict the values, and if it is possible to obtain a model with a satisfying performance on predictions.

Secondly, the use of machine learning can help understand the data and how the input variables of the scenarios relate to the results of equations 1.1 and 1.19. Some models allow the computation of a feature importance metric which allows to understand how a model achieves its predictions. By studying the error of such models, it is also possible to see if any unexpected result is detected. Given the results obtained, researchers will be able to study them and possibly find shortcomings in the simulation or evaluation process.

Finally, the goal of the evaluations is to get a sense of the reliability of the grid. The interest is not necessarily in the exact values of equations 1.1 and 1.19, but rather in a classification of secure and insecure situations. Sometimes, a human expert could trivially perform this classification task, for example because issues are immediately apparent. The simulations however calculate the precise value of both equations, which might not provide valuable additional information. Indeed, in a very reliable situation, equation 1.19 would be high and 1.1 low, while in a very unreliable situation the result of equation 1.1. Instead, it might be desirable to run the full simulation only on situations that are difficult to classify for an expert, and use estimations of the values otherwise. To perform this, a machine learning model can be fill the role of an expert to classify the situation into categories and based on the confidence of the classification, the decision to either run the full simulation or to perform an estimation can be taken.

Chapter 2

Data analysis

In this chapter, a study of the available data is performed. As a base for this chapter and the following ones, 1000 scenarios for the same day are considered, which contain in total 24 000 hourly snapshots. This chapter aims to understand the distribution of input variables (wind, solar and load) and values of equations 1.1 and 1.19 across the dataset of scenarios. This will later help to make sense of the results that are obtained by the machine learning models that are presented in chapters 4 and 5.

Unless mentioned otherwise, all power values are expressed in per-unit with a base of 100 MVA (1 per-unit = 100MW) .

2.1 Unit commitment & economic dispatch (UC&ED)

The day-ahead forecast gives expected load, solar and wind values for the next day. Similar forecasts are used in the real world to take decisions about the power grid, such as which thermal generator to activate. This process is also simulated and creates a UC&ED. Given the forecast, a UC&ED consists of hourly decisions about storages and generators.

For hydro/thermal generators, the activation of each generator is decided on a day-ahead basis. Generators need some time to start up as they require heat. This is why it is necessary to take decisions in advance. Each generator has a minimum (Pmin) and a maximum (Pmax) possible power generation. A system-wide Pmin and Pmax can be computed. From here, the generator production for the next day can only vary between the Pmin and Pmax at all times.

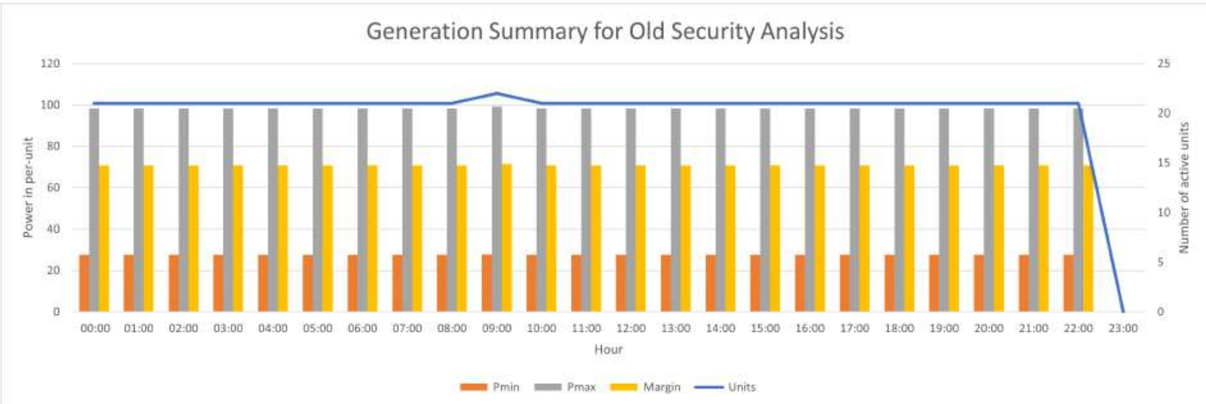
A storage device can either be seen as a generator or a as load by the system. When power is stored, it is seen as load as the power needs to be provided by the grid. When power is used from the storage, it acts like a generator by providing more power to the system. The maximum power that can be stored or used from the storage varies from hour to hour based on the energy already in the device stored and the planned use for the rest of the day. As such, the max load and generation that the storage can provide are decided for each hour based on the day-ahead.

This whole process is also referred to as unit commitment : what unit of production and what use of storage will be available at what hour for the next day.

During the course of this project, two different UC&ED were analysed. Technically, one of the UC&ED is an improvement upon the other. However, it remains interesting to study both security analyses, as comparing results between both will allow to identify important variables in each case, and study what kind of performance can be achieve in both situations. It is also an opportunity to analyse the improvements provided by the update. For the remainder of this work, the upgraded UC&ED will be referred to as "updated" or "new", while the other will be referred to as "original" or "old".

Figures 2.1 and 2.2 show the results of the UC&ED. Figures 2.1 represents the hydro/thermal generators unit commitment in both the old and updated UC&ED. The Pmin and Pmax of the combined hydro/thermal generators is also shown. The margin is the difference between Pmax and Pmin (not to be confused with eqn. 1.19). Units refers to the number of committed (active) hydro/thermal generators for each hour.

The old generation summary presents very little variation between hours except for the the last hour where all values are set to 0. The only other hour which presents a difference is 9:00, which has 22 active generators instead of 21, which is a marginal difference. For the new UC&ED, the variation is more consequent with more units are active during the daylight hours.

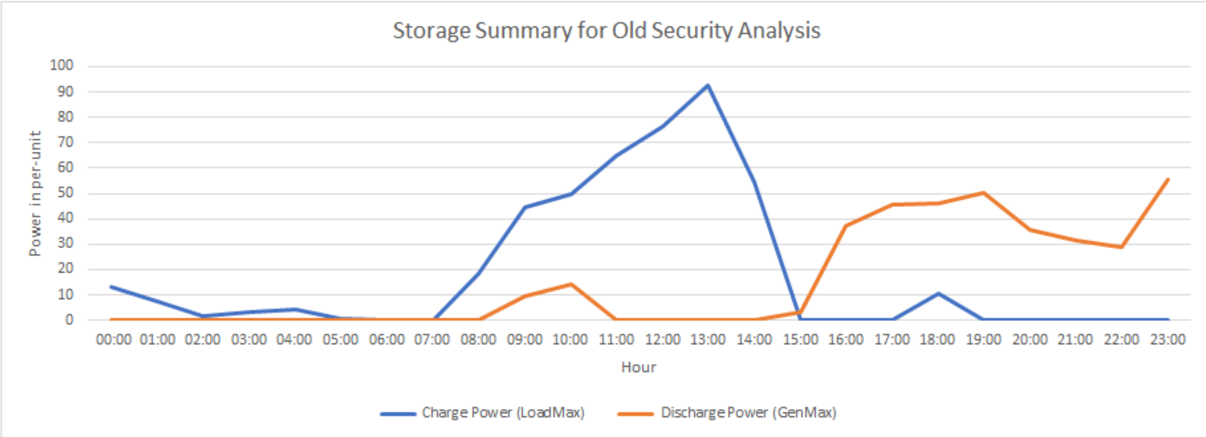


(a) Old UC&ED in per-unit (100 MW)

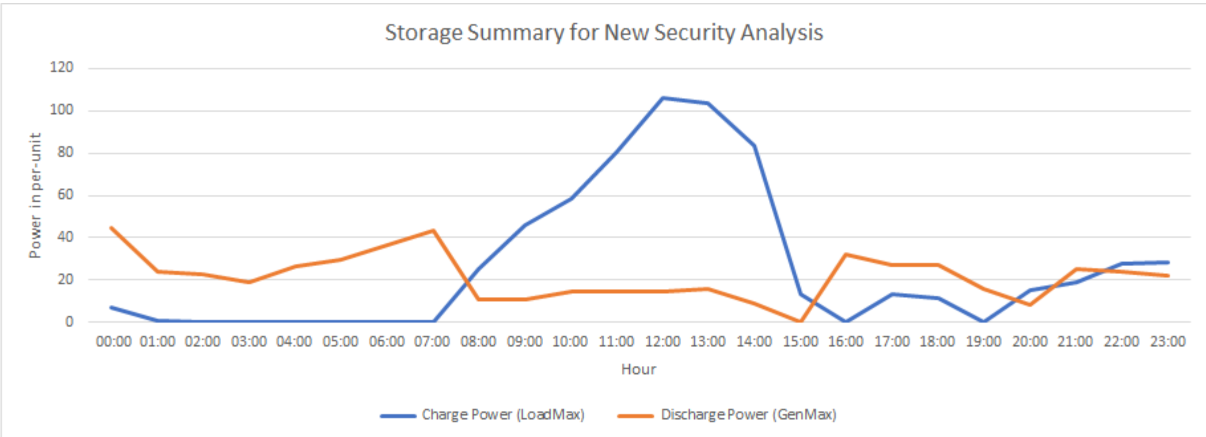
(b) New UC&ED in per-unit (100 MW)

Figure 2.1. Comparison of generation summary for the old and new UC&ED

Figure 2.2 shows the difference in storage for both security analyses. They have a similar evolution throughout the day, with a peak of charge during the daylight hours. This might be to account for the higher production that was seen in figure 2.1. However, they also present some differences. First, the updated UC&ED has more opportunities to use the storage as a generator. Only one hour during the day cannot use storage as generator against eleven for the old one. Furthermore, the old UC&ED seems to charge the storage at the beginning and middle of the day and empty it at the end of the day. This gives no opportunity for using the stored energy in the beginning of the day, as the storage is empty. Overall, the old UC&ED makes the use of the storage more restrictive.



(a) Old UC&ED in per-unit (100 MW)



(b) New UC&ED in per-unit (100 MW)

Figure 2.2. Comparison of storage summary for the old and new UC&ED

2.2 Forecasts

Figure 2.3 shows the forecast for the hourly total load demand as well as solar and wind production. The values presented are the sum across all nodes for load, solar and wind values respectively. It is interesting to note that the highest requirement for additional power happens in the morning, as this is when the biggest discrepancy happens between load demand an renewable generation. Between 11:00 and 14:00, the solar generation is enough to provide energy to the whole grid load demand forecast, assuming that the transport of the power can be insured.

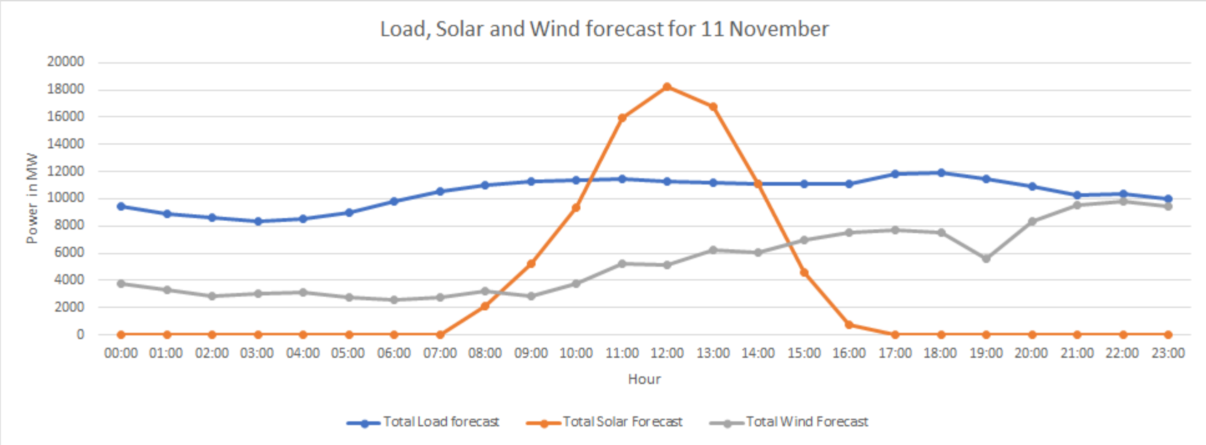


Figure 2.3. Hourly Forecasts for the 11 November for wind, sun and load values

2.3 Evaluating Scenarios

To better understand the scenarios, the figure 2.4 shows the distribution of objective values (equation 1.1) obtained for each scenario and classified by hour. The objective value is the main indicator to evaluate the problems that occur in a scenario is at a given hour. The first UC&ED suffers from a severe spike in objective value at hour 23:00, and has very small values otherwise. This might be due to the fact that no hydro/thermal generator is committed at this time, as was shown in figure 2.1a. This severe spike presents issues as the prediction at 23:00 is significantly different from all other predictions, which might make it hard to make relevant analyses for the other hours.

Comparatively, the updated version has consistently higher objective values at multiple hours of the day. However, as can be seen by the 90th percentile point, most of the scenarios before 15:00 do not present any significant issues, and high objective values are outliers. In the evening, the number of scenarios with high objective values increases significantly.

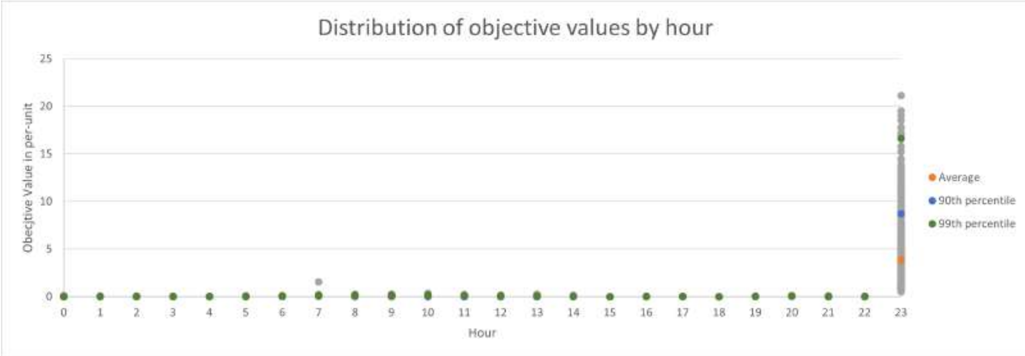
The same analysis is made for the margin on figure 2.5. The margin distributions are similar between both security analyses, with peaks during daylight hours. The new UC&ED never has a margin value above 0 after 15:00. This is because the value of equation 1.1 is never lower than the threshold of 0.0005 after this time.

2.4 Training Datasets

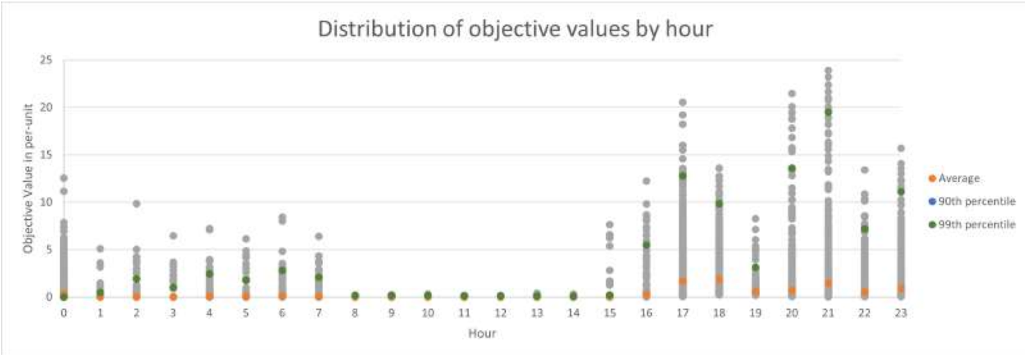
The first objective is to train machine learning models with high performance to predict the values of 1.1 and 1.19. The basic data used during the training of the raw data from the scenarios, that is : the load demand and solar and wind generation values. The solar and load values are given for each node, as they are present everywhere in the grid while the wind values only appear in some nodes. This leads to 46 inputs for the load values, 46 for the solar values, and 5 for the wind production, which produces a total of 97 features.

Additional datasets were constructed by adding more features to the datasets.

The datasets were built using the following inputs :

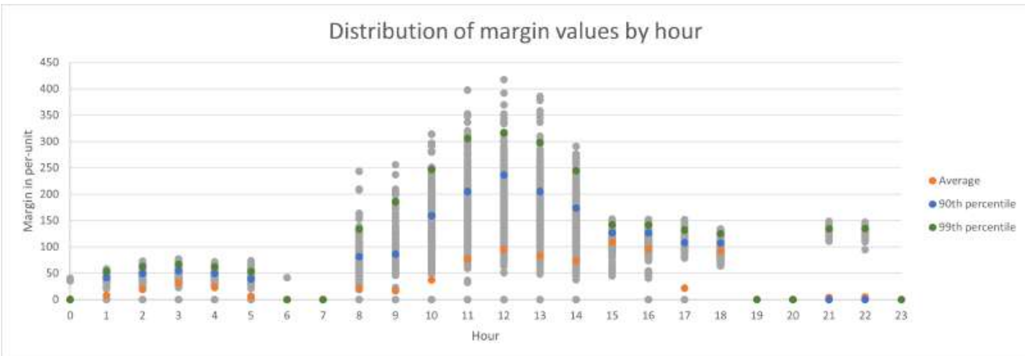


(a) Old UC&ED

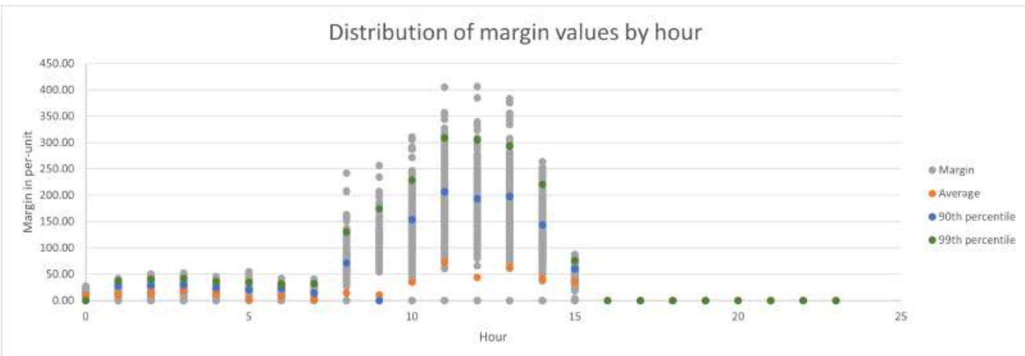


(b) New UC&ED

Figure 2.4. Comparison of distribution of objective values (eqn. 1.1) across scenarios per hour



(a) Old UC&ED



(b) New UC&ED

Figure 2.5. Comparison of distribution of margin values (eqn. 1.19) across scenarios per hour

- Basic : Only contains the 46 load, 46 solar and 5 wind inputs.
- Time: The basic set with an additional field that gives the time of the day
- Features : The basic set enhanced with five hand-crafted features : the sum of loads, the sum of solar power generation, the sum of wind power generation, the difference between load and renewable production and the percentage of load that can be satisfied using renewable energy.
- Time / Features : A combination of the Time and the Features variables
- Excluding 23 : Same as the basic set, but where the values at time 23:00 were removed. This only exists for datasets applied on the old UC&ED and was created to remove the significant outlier detected at figure 2.4a.

Furthermore, two types of data were considered for the load, sun and win values :

- Absolute : the absolute values are the true physical values of the load and the renewable generation on the grid. They are expressed in per-unit (100MW)
- Deviations: the deviations to the forecast. It is the difference between the day-ahead expectation and the observed values for the load and renewable production. They are also expressed in per-unit.

The reasoning behind this is that, while it is expected that the absolute data contains more information, as it depicts the real situation of the grid, deviation data might be more suitable in some circumstances. Indeed, as the unit commitment is based on the forecast, if the deviation is close to 0 it is likely that there will be less issues due to the unit commitment process.

2.5 Classification

The task was also reformulated as a binary classification task. The classes are zero and one. Zero corresponds to a "secure" situation where equation 1.1 is ≤ 0.0005, while one is when it is equal or above to that value.

The training set for the old UC&ED had a distribution of 58.5 percent ones and 41.5 percent zeros. In the test set, the distribution is exactly the same, at 58.5 percent and 41.5 percent respectively. For the new UC&ED, the distribution in the training set is 60.6 percent ones and 39.4 percent zeros, while the test set has a distribution of 59.7 percent ones and 40.3 percent zeros.

For the old UC&ED, the results of the dataset without the last hour were not included in the report to avoid redundancy. Indeed, these models performed very similarly to the ones trained one the basic dataset in all situations.

Chapter 3

Machine learning algorithms

Using the datasets described in section 2.5, four machine learning algorithms were chosen to learn a good prediction.

All regression models are evaluated using the R2 score and mean squared error metrics while the classification models are evaluated using accuracy. The datasets are split into a training and a test set with a respective size of 80/20.

3.1 Trees

The first type of models used are tree ensembles. Individual decision trees are trained, and the final outputs of the ensemble model is obtained by combining the different trees. For the regression problems, the final prediction is the average prediction across all trees for, and the majority class for classification problems.

A major advantage of decision trees is their interpretability, which allows to analyze what variables were used by the model to reach the prediction. Unfortunately, ensemble models are slightly less interpretable as their prediction can depend on hundreds of different decision trees. However, it is still possible to calculate feature importance, which is the main interest in this case.

The two models that were tested are Random forests and Extra trees.

3.1.1 Random Forest

Random forest [7] is an ensemble method that builds a serie of trees. It uses bootstrap sampling to associate each tree to a unique learning sample. Each tree is then built by using random feature subset selection. Instead of choosing the best split among all features, like a classic decision tree, the split is chosen from a subset of k features. The value of k needs to be set as a hyperparameter. A smaller value of k leads to a bigger variance decrease at the cost of an increase in bias.

The hyperparameters used in the experiments were as follow : 500 trees were built and k was set to the number of features. This means that the random forest is equivalent to tree bagging. The leaves were split as soon as they contained two samples. Squared error was used as a metric to evaluate the split for the regression problems, and Gini impurity for the classification problems.

3.1.2 Extra Trees

Extra Trees [8] is a method similar to random forest as it also select a random subset of features to perform the split on. However, instead of choosing the split point that decreases the most the impurity, the split point is chosen randomly for each feature, and the best split among all features is chosen. This increases the randomness compared to random forests, which usually leads to a slight increase in bias but a decrease in variance.

The hyperparameters used are the same as the Random Forests.

3.2 Linear models

3.2.1 Linear / Logistic regression

Linear regression is a simple weighted linear model that minimizes the sum of squares between the observed and the predicted values while logistic regression computes a linear classification threshold. The output is expressed as a weighted sum of the input variables.

As the name implies, linear regression is limited to produce linear models. Due to the limited expressive capabilities of such a model, state of the art performance is not expected, but it should be interesting to compare how this simple model compares to more complicated ones, like ensemble of trees and neural networks.

3.2.2 Multilayer Perceptron

The multilayer perceptron is an architecture composed of layers where each layer is comprised of nodes. Each node in a layer is the result of a weighted sum of nodes in the previous layer. We talk of fully connected feedforward networks when all nodes in a layer are connected to all nodes in the next layer. Training a neural network consists of performing gradient descent to learn good weight values between nodes of consecutive layers.

Usually, activation functions are used to break the linearity and allow the MLP to express nonlinear dependencies.

Compared to trees, hyperparameters had a bigger impact on performance. As such, to find an optimal set of hyperparameters, 5-fold cross validation was performed. The training set was split into train and validation. The test set is never used to decide the hyperparameters. The hyperparameters with the best average performance across all validation folds were chosen. The criterias used were mean squared error for regression problems and cross entropy loss for classification.

The best performance for the regression problems were obtained with a model trained during 500 epochs with a learning rate of 0.0001 and 128 as hidden layer size. The number of hidden layers was fixed at two for all experiments. A Relu activation layer was used after every layer but the output layer. Intuitively, it would make sense to use a ReLU activation function of the output layer, as all the predicted values should be positive, however this lead to most models suffering from the dying Relu problem where the gradient would be zero for most predictions which leads the model to be unable to learn.

For the classification tasks, a simpler model offered the best performances. The model was trained during 100 epochs with a learning rate of 0.001 and 64 as hidden layer size. As previously, two hidden layers were used for all experiments and a ReLU activation function in all layers expect the last one. As the output of the model should be a binary classification, the sigmoid activation function was applied on the output to restrict the output to values between 0 and 1.

For this architecture, all inputs were normalized to be zero centered with a standard deviations of one to respect the assumptions behind weight initialization that the data is zero centered. To perform normalization, the mean and standard deviation of the training set were calculated, and both the train and test sets were normalized using these values.

Chapter 4

Results

4.1 Basic Results

The Tables 4.1 and 4.2 present the first results for the old and new UC&ED respectively.

Regarding table 4.1, the predictions made with only the basic set of variables are unsatisfying with every model tested. This can partially be explained by the considerable difference between the values obtained at 23:00, and the values obtained during the rest of the day. This leads the time of the day to be a critical variable. The models cannot learn a distribution that explains satisfyingly the true probability distribution without it, as shown by the low R2 scores. The best R2 score is 0.42, and is associated with a mean squared error of 0.59. This is obtained by the neural network model which suggests that a combination of features is needed to make more accurate predictions, which also explains why the tree models have a low score. The error is relatively high regardless, especially considering that the variance across the dataset is only around 1.098. Removing the last hour from the dataset increases R2 score drastically for all models, which confirms the idea that most of the error comes from that hour.

Table 4.2 shows greater performance with the basic set of variables. The best models reach a R2 score of 0.90 and a mean squared error of 0.21. This improved performance is explained by the fact that there is no big outlier when it comes to distribution of objective value per hour, as shown in figure 2.2b. Furthermore, the high R2 score seems to suggest that the distribution of objective across hours can mostly be explained by the values of the forecasts.

The time metric implicitly contains information about the unit commitment at each hour. As such, it encompasses a wide range of information about the grid situation. When adding the time to the models of table 4.1, the mean square error decreases significantly with the old UC&ED. The mean squared error drops to 0.0045 for the best model, a reduction by a factor of 100 compared to the baseline. The R2 score also increases to over 0.99 for the random forest and Extra Trees. The new UC&ED doesn't show as impressive gains, confirming the idea that most information is contained in other variables. However, when adding the time in 4.2, the performance of the trees now surpasses the MLP. This is due to there now being a single feature that contains a lot of information, which favours the performance of trees.

Overall, for table 4.1, the best model is obtained when using only the basic features and the time of the day as input, and using ExtraTrees with deviation values. For table 4.2, the best performance is also obtained by ExtraTrees with time and features, and using absolute values.

Another observation of both tables is that adding the hand crafted features does not have a significant impact on performance. They might marginally improve or worsen the obtained models, but overall the results are too close to one another to draw any conclusions.

Finally, the models trained on the deviation data perform very poorly in some situations and very well in others compared to the absolute set. More specifically, performance is very poor when time is not used as an input. It seems like the model can partially infer the time of the day with the absolute values as input, but not as easily with the deviations. Indeed, absolute load, solar and wind values can be used to determine the time. As we saw in figure 2.3, a high amount of wind is more likely in the evening, the load is lower during the morning, and sun values only appear during the daylight hours with a peak at noon. Note however that deviations also might give some information. Indeed, the deviation of the sun will always be zero outside of daylight hours, as both the predictions and the realization will always be zero.

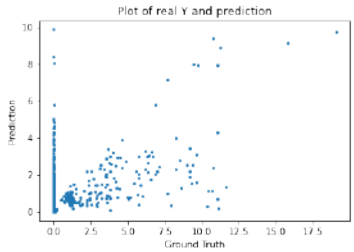
Figures 4.1, 4.2 and 4.3 display some values for prediction against ground truth for the old UC&ED, and 4.4, 4.5 and 4.6 for the new one. These figures show that all models with a R2 score below 0.85 show significant weaknesses in some areas of the plot, and are therefore not satisfying. The most frequent error that appears in the figure is the model prediction a positive value when the ground truth is zero. This is especially apparent in figures like 4.1a, 4.1b, 4.2a, 4.2b and 4.3e

4.2 Feature importance analysis

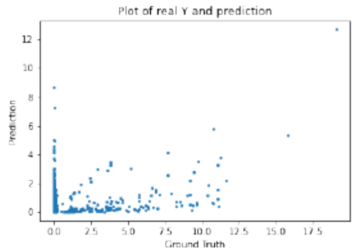
To study which variables are relevant to the problem, it is interesting to compute feature importance for the models with a good prediction. The traditional mean impurity decrease was used to compute them, which checks the ability of a variable to consistently decrease the impurity of the leaves in the tree. An issue is that mean impurity decrease tends to overestimate the importance of numerical features, and underestimate the importance of categorical features. The only categorical feature is the time and as such it might get underestimated. Also note that feature importance only estimates how important a variable is for the trained model. It does not give direct indication on the real information given by the variable.

To limit the length of this section, not all models were analysed for both security analyses. The models chosen were selected for their relevance. Furthermore, only the top 10 variables were plotted for each case. For more detailed plots, please see Appendix A.

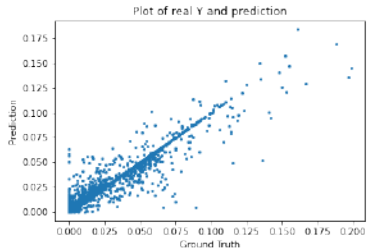
For the old UC&ED, the basic, basic excluding last hour, time and time/features datasets are shown. For each, the model performing best among all tree models was chosen. Figure 4.7a shows that without any additional information, the main criteria used was the wind variables. Solar variables on the opposite are almost never used. Once the last hour is removed, figure 4.7b shows that some solar values are used. Figures 4.7c and 4.7d show that the best models overall use the time as the single most important variable. Solar and load values are almost never used, while wind-related values provide some information.



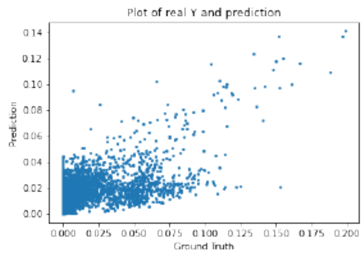
(a) Basic Absolute



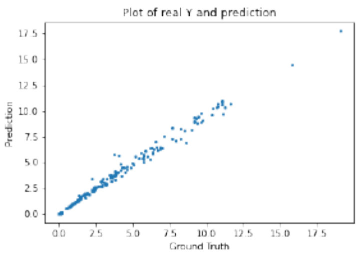
(b) Basic Deviation



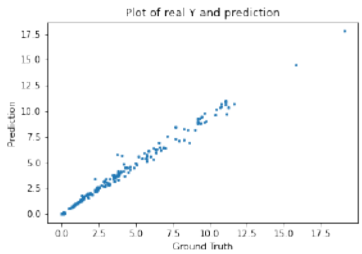
(c) No 23 Absolute



(d) No 23 Deviation

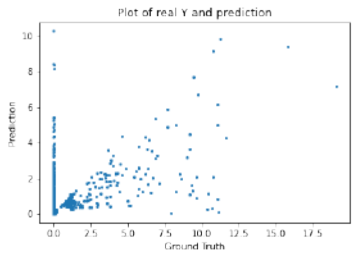


(e) Time Absolute

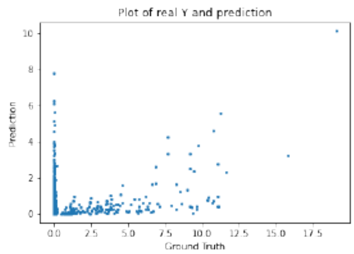


(f) Time deviation

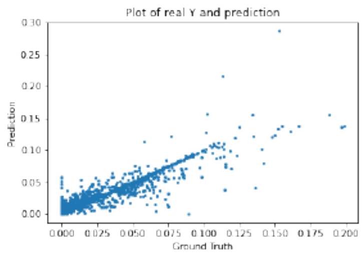
Figure 4.1. Prediction and ground truth of eqn. 1.1 : Random Forest, old Security Analysis



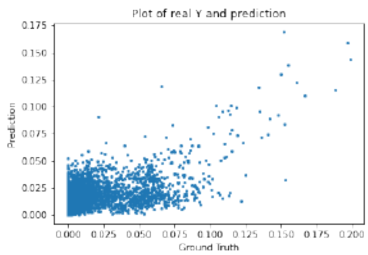
(a) Basic Absolute



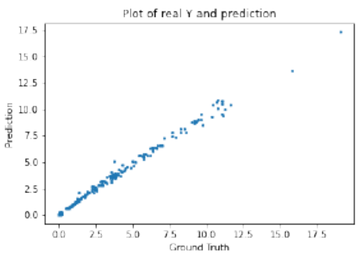
(b) Basic Deviation



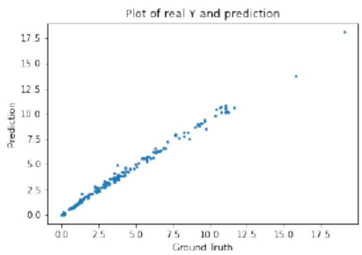
(c) No 23 Absolute



(d) No 23 Deviation

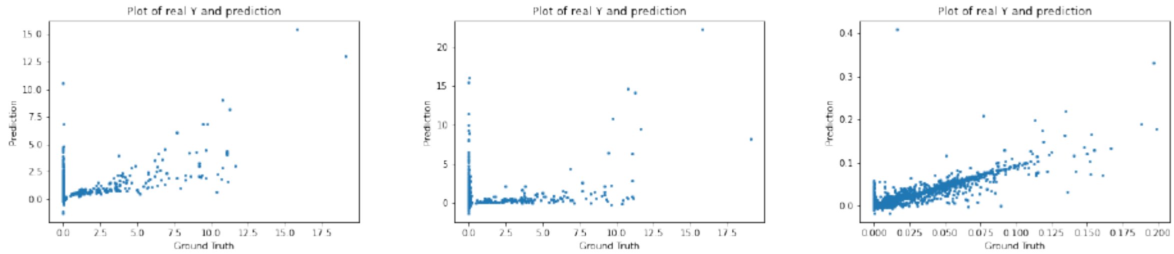


(e) Time Absolute

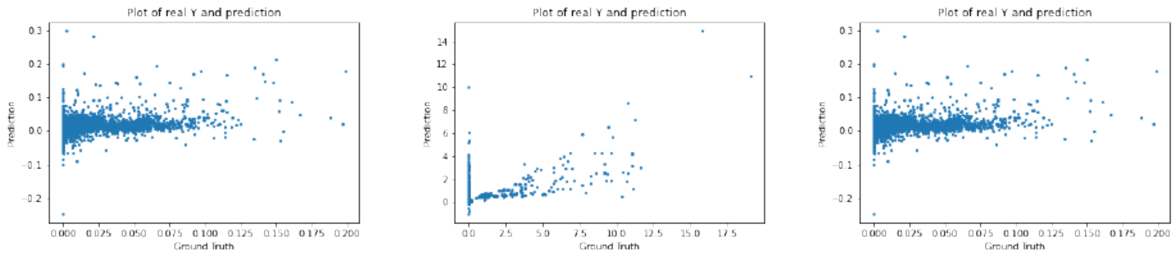


(f) Time deviation

Figure 4.2. Prediction and ground truth of eqn. 1.1 : Extra Trees, old Security Analysis

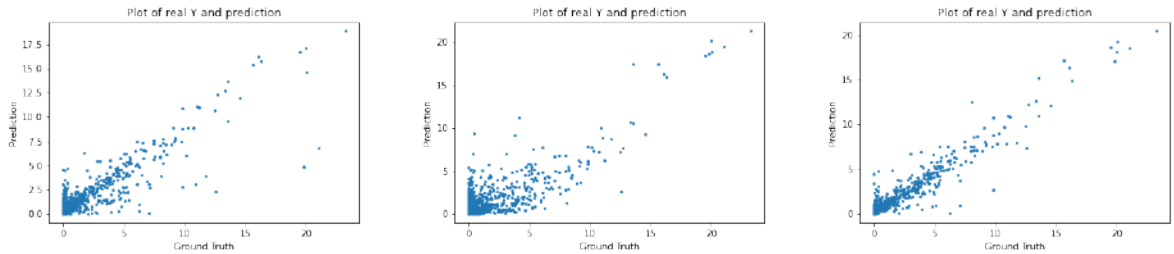


(a) Basic Absolute (b) Basic Deviation (c) No 23 Absolute

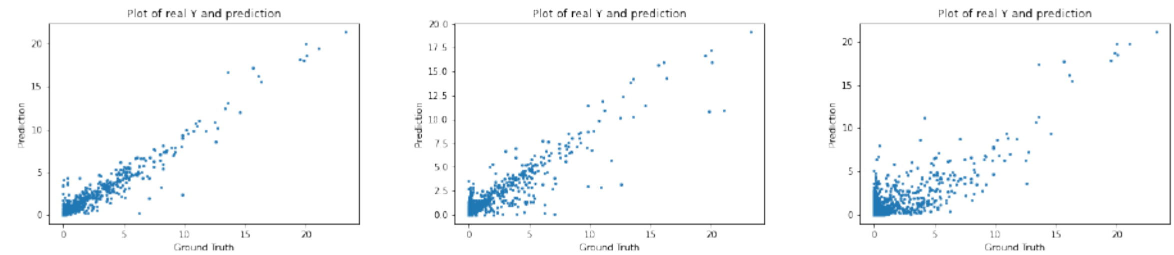


(d) No 23 Deviation (e) Time Absolute (f) Time deviation

Figure 4.3. Prediction and ground truth of eqn. 1.1 : Multi Layer Perceptron, old Security Analysis

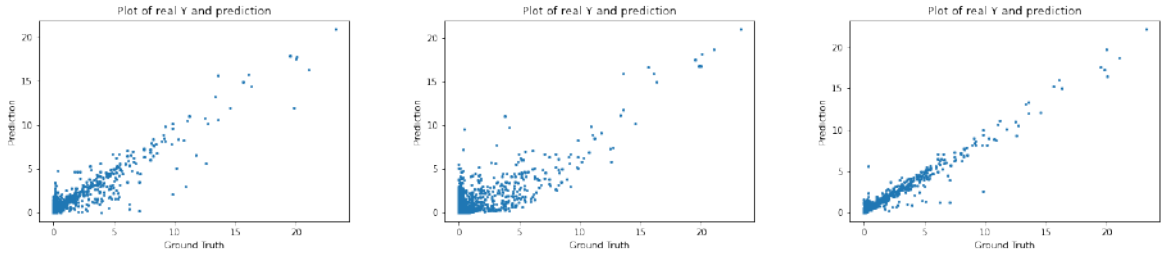


(a) Basic Absolute (b) Basic Deviation (c) Time Absolute

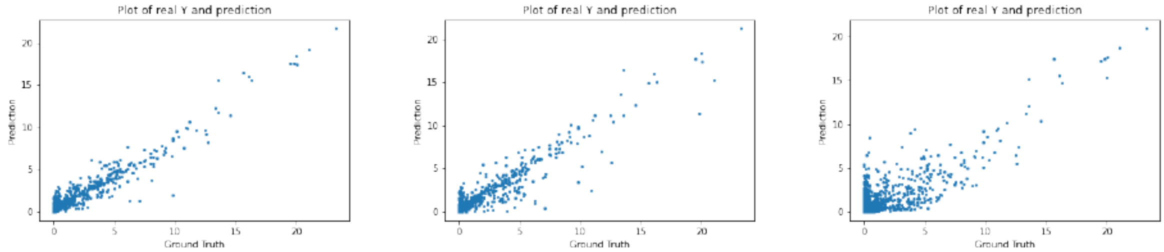


(d) Time Deviation (e) Features Absolute (f) Features deviation

Figure 4.4. Prediction and ground truth of eqn. 1.1 : Random Forest, new Security Analysis

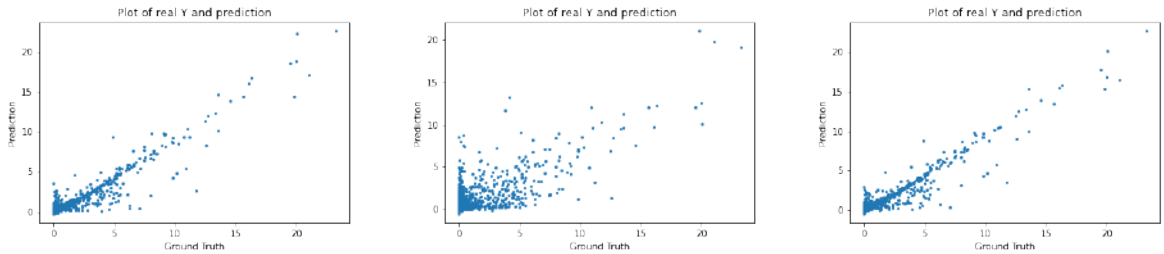


(a) Basic Absolute (b) Basic Deviation (c) Hour Absolute

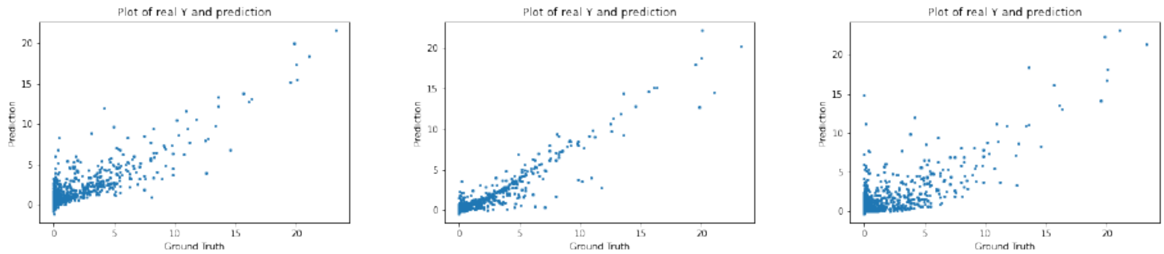


(d) Time Deviation (e) Features Absolute (f) Features deviation

Figure 4.5. Prediction and ground truth of eqn. 1.1 : Extra Trees, new Security Analysis



(a) Basic Absolute (b) Basic Deviation (c) Time Absolute

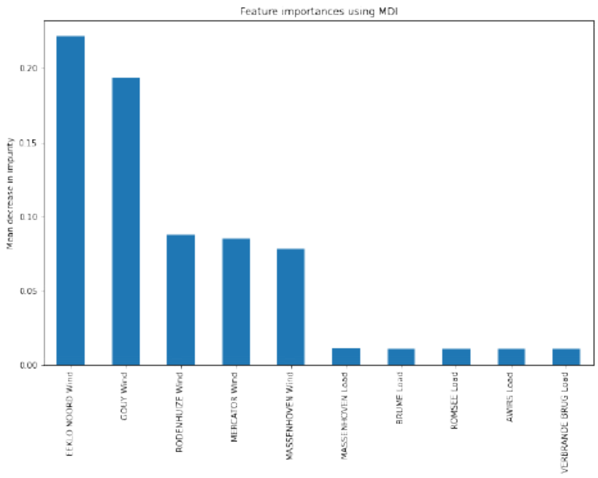


(d) Time Deviation (e) Features Absolute (f) Features deviation

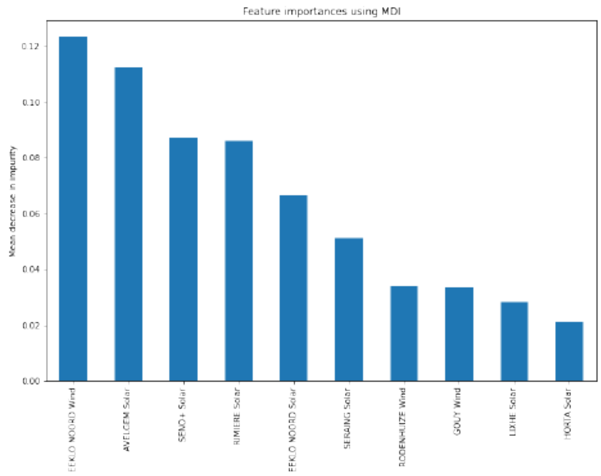
Figure 4.6. Prediction and ground truth of eqn. 1.1 : Multi Layer Perceptron, New Security Analysis

Figure 4.7d also shows that while the additional features don't improve the results of the learned model, they are still used by the model as splitting criteria. As such, "Difference" is the third most important feature and "Sum Wind" the fourth.

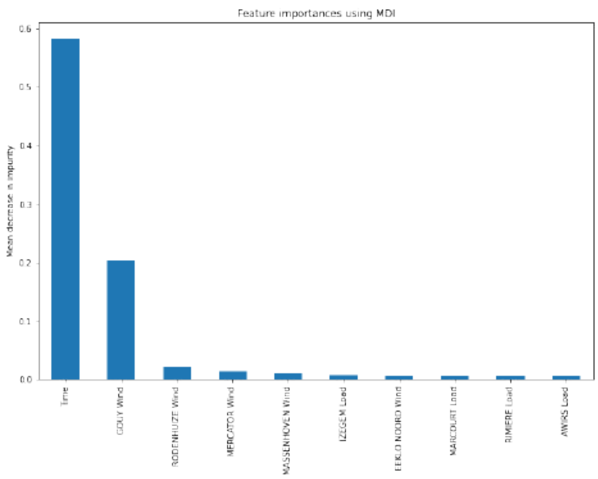
For new UC&ED, the same process is followed, which results in all models being Extra Trees with absolute data as they always perform best among the trees. Figure 4.8a shows the feature importance for the basic set. Again, the wind variables are dominating the load and solar ones. Figure 4.8b shows that by adding the time variable, it becomes the second most important feature and GOUY Wind. GOUY Wind might be relevant because it has a high power generation capacity. The other wind variables are used slightly less when the time is added. Figures 4.8c and 4.8d confirm that the additional features are used by all models where they are added.



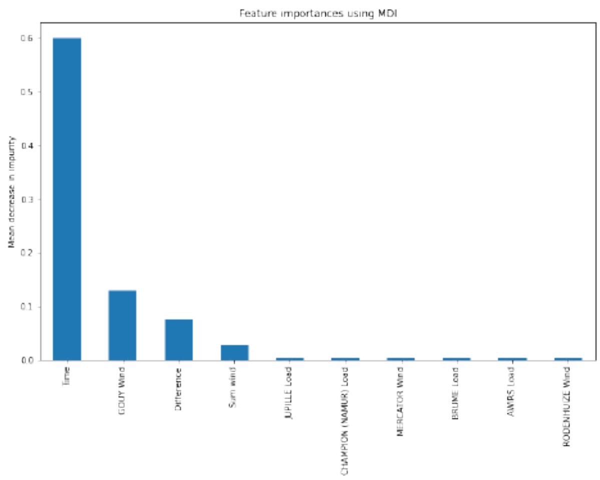
(a) RandomForest Basic Absolute



(b) RandomForest No 23 Absolute



(c) ExtraTrees Time Deviation

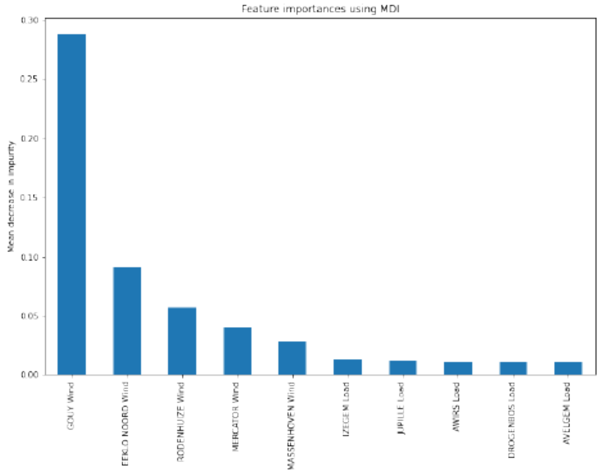


(d) ExtraTrees Time and Features Deviation

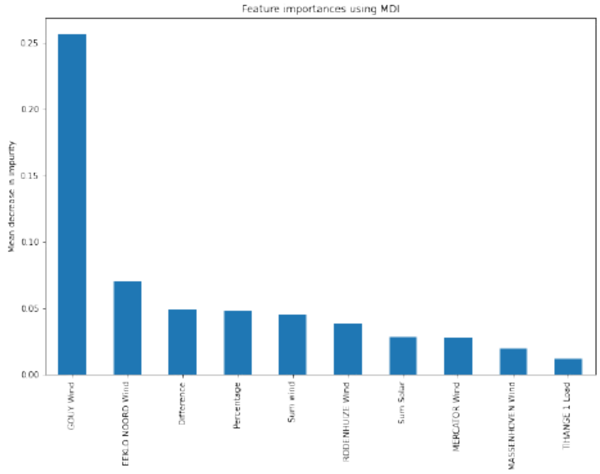
Figure 4.7. Feature importance old UC&ED (eqn. 1.1)

4.3 Classification Results

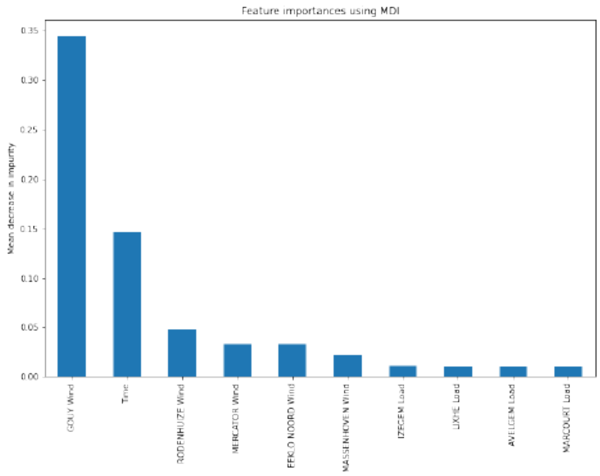
The main results of the binary classification task can be seen of tables 4.3 and 4.4. While the basic variables alone do not contain enough information to make detailed predictions about the value of equation 1.1, especially for the old UC&ED, they do contain enough



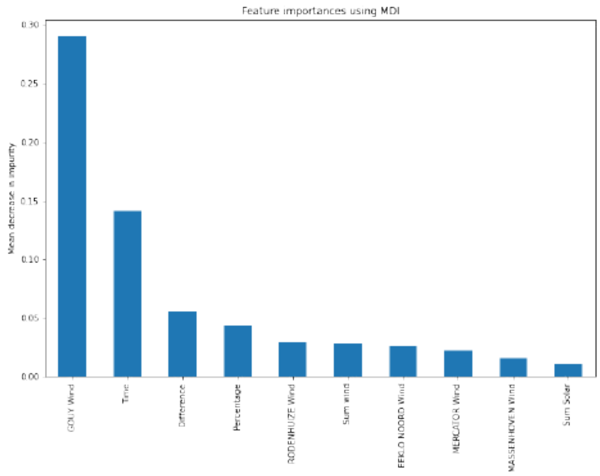
(a) ExtraTrees Basic Absolute



(b) ExtraTrees Additional Features Absolute



(c) ExtraTrees Time Absolute



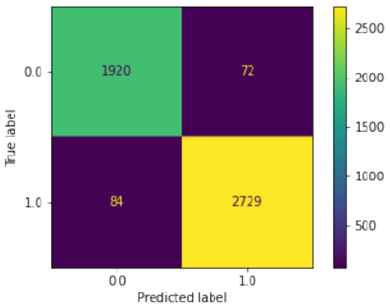
(d) ExtraTrees Time and Features Absolute

Figure 4.8. Feature importance new UC&ED (eqn. 1.1)

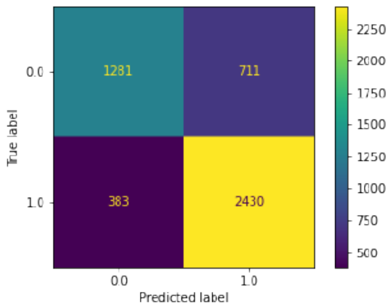
information to make a good classification prediction. Adding additional features and time only improves the accuracy by 1.2 percent in the old UC&ED and around 3.6 percent in the new one.

Random forest models obtain the best performances here, followed by the perceptron, which outperforms the RandomForest slightly for the basic set in table 4.4. The ExtraTrees has similar performance to the perceptron. Neural networks could be the best algorithm given more in depth hyperparameter optimization.

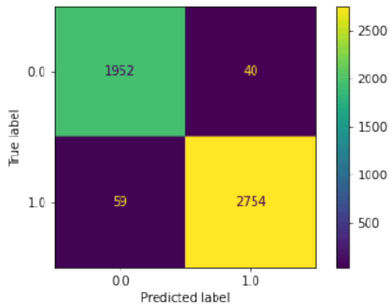
Confusion matrices are analysed to obtain more information about the predictions. Figure 4.9 displays the confusion matrices for the old UC&ED. Missclassifications are evenly distributed for the best models, such as 4.9a. However, poor models such as 4.9b, 4.9f and 4.9g tend to wrongfully predict more ones when the ground true is zero. This implies that the model believes more that safe situation are problematic than the opposite. A similar situation is seen for the new UC&ED on figure 4.10, but overall performance is worse.



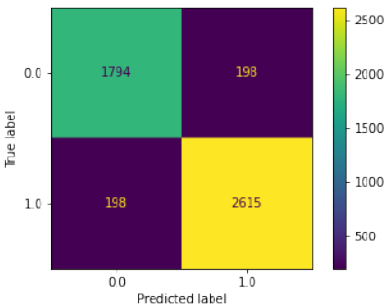
(a) Forest Basic Absolute



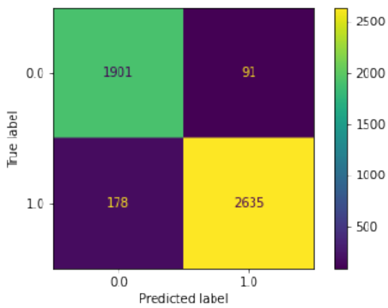
(b) Forest Basic Deviations



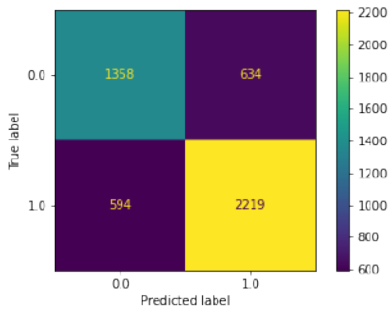
(c) Forest Time Features Absolute



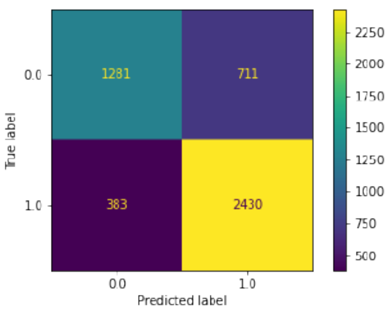
(d) Forest Time Features Deviations



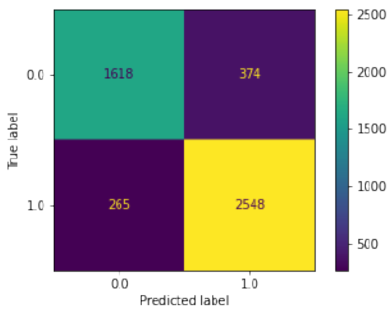
(e) Extra Basic Absolute



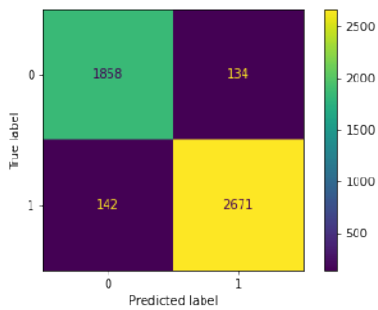
(f) Extra Basic Deviations



(g) Extra Time Features Absolute

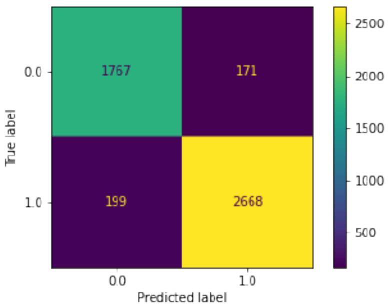


(h) Extra Time Features Deviations

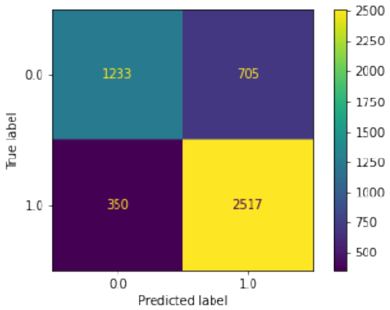


(i) MLP basic absolute

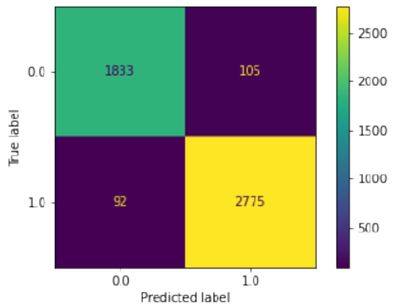
Figure 4.9. Confusion matrix old UC&ED (0 = "secure", 1 = "insecure")



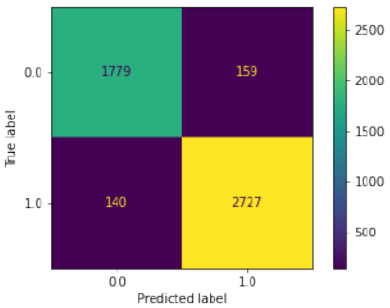
(a) Forest Basic Absolute



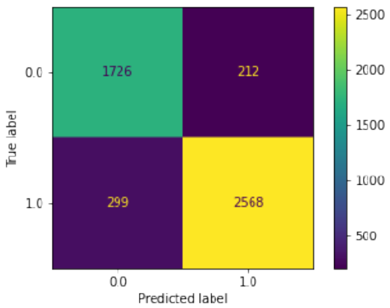
(b) Forest Basic Deviations



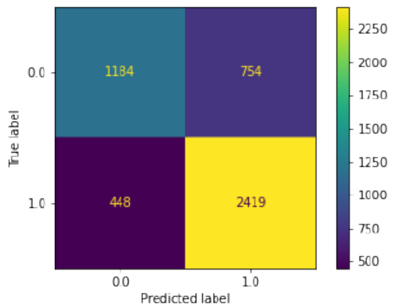
(c) Forest Time Features Absolute



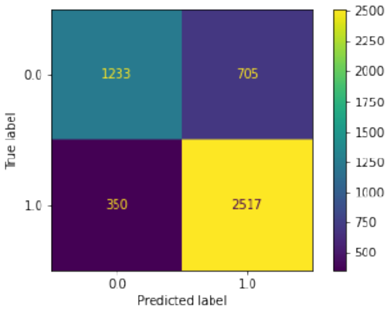
(d) Forest Time Features Deviations



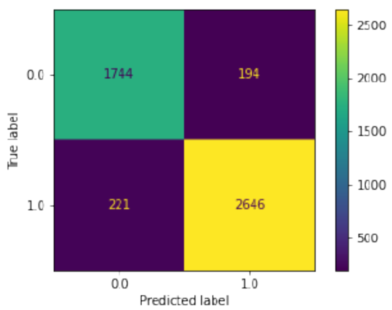
(e) Extra Basic Absolute



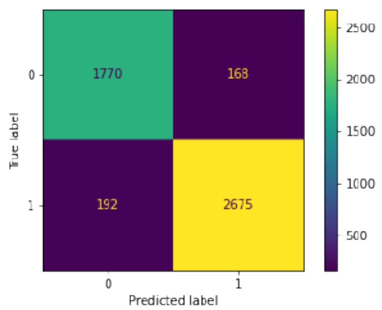
(f) Extra Basic Deviations



(g) Extra Time Features Absolute



(h) Extra Time Features Deviations



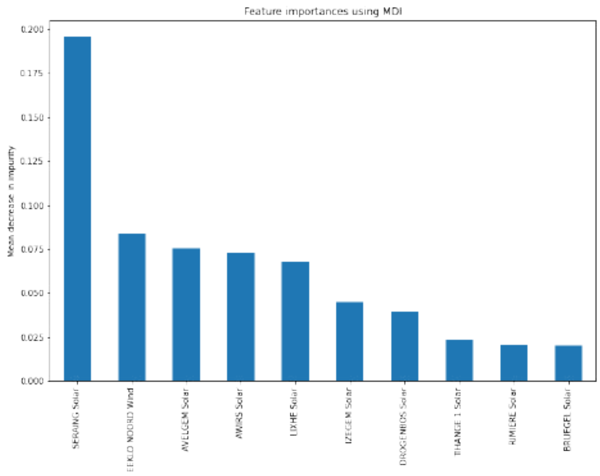
(i) MLP basic absolute

Figure 4.10. Confusion matrix new UC&ED (0 = "secure", 1 = "insecure")

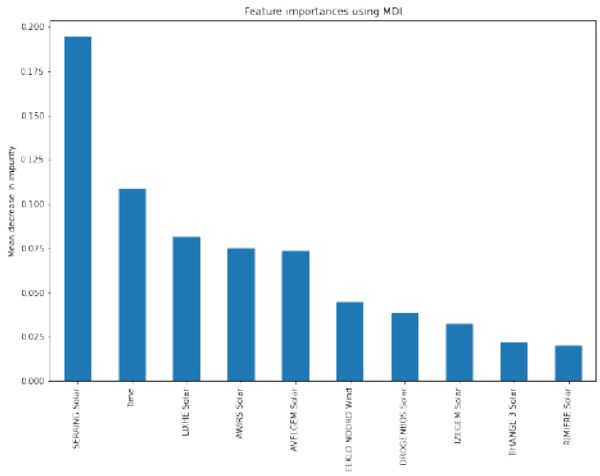
4.4 Margin Results

A similar process has been followed to obtain prediction results for the margin. The results of predicting the margin for both the old and new UC&ED can be found in tables 4.5 and 4.6 respectively. The results show that Random Forests obtain the best prediction for this task. Compared to the previous results, no model manages to get excellent performance, especially for the new UC&ED where the best model only achieve an R2 score of 0.8. This suggests that the margin depends on other values that are neither in the training data, not can be inferred from the time variable.

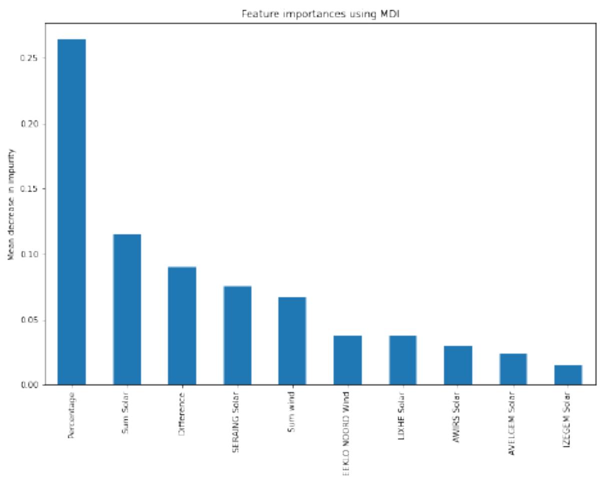
Another observation is that the additional features offer improvements over the models without them, which was also not the case for the previous experiments. By looking at the feature importance (fig 4.11 and 4.12) of those models, it can be seen that the variables used are different from the previous experiments. Overall, the solar values have more impact, as well as the additional features "Percentage", "Difference" and "Sum Solar". Time still plays an important role, especially in the old UC&ED



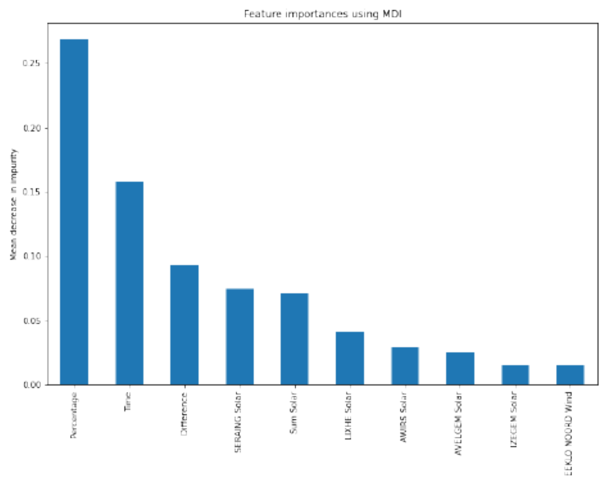
(a) RandomForest Basic Absolute



(b) RandomForest Time Absolute

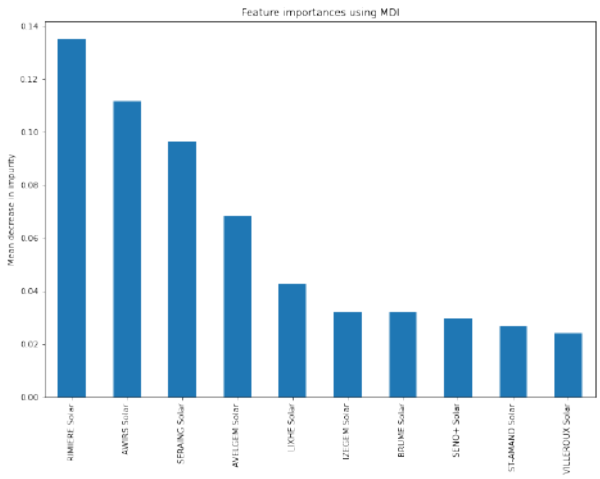


(c) RandomForest Features Absolute

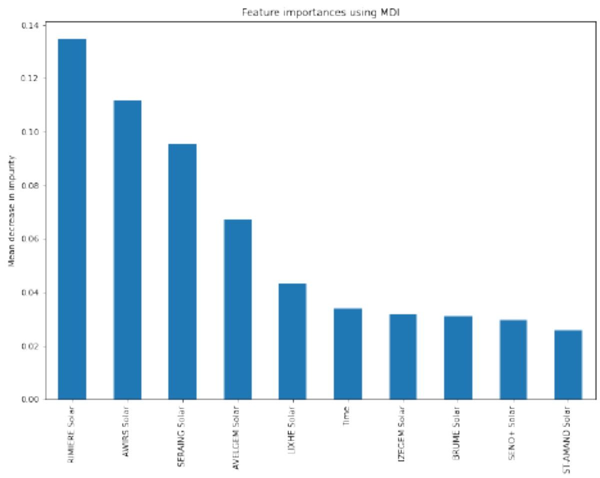


(d) RandomForest Time and Features Absolute

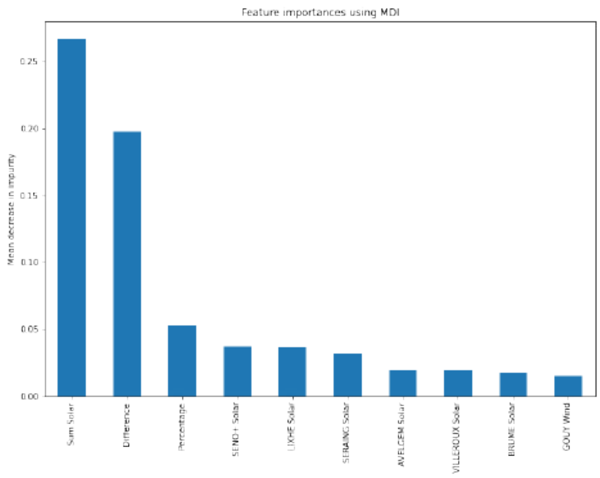
Figure 4.11. Feature importance old UC&ED margin (eqn. 1.19)



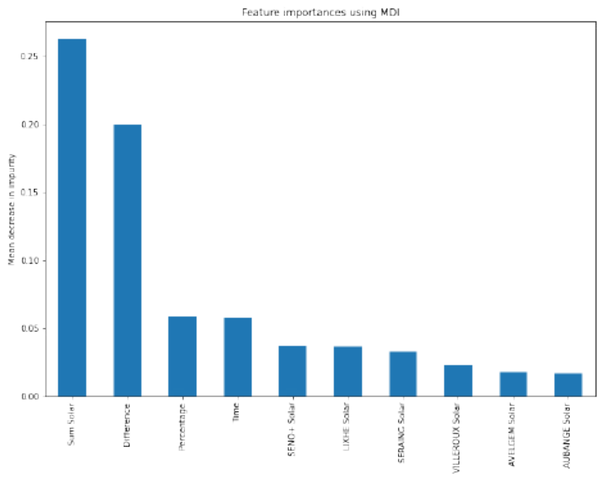
(a) RandomForest Basic Absolute



(b) RandomForest Time Absolute



(c) RandomForest Features Absolute



(d) RandomForest Time and Features Absolute

Figure 4.12. Feature importance new UC&ED margin (eqn. 1.19)

Model	Data	Values	Mean Squared Error	R2 Score
Random Forest	Basic	Absolute	0.711737203	0.300772518
		Deviation	0.881036108	0.134449264
	Basic without 23	Absolute	**0.000041707**	**0.922175166**
		Deviation	0.000343244	0.359512635
	Features	Absolute	0.695835234	0.316394988
		Deviation	0.913344684	0.102708553
	Time	Absolute	0.006516392	0.993598142
		Deviation	0.006463815	0.993649795
	Time / Features	Absolute	0.011915081	0.988294343
		Deviation	0.012053326	0.988158527
ExtraTrees	Basic	Absolute	0.71962192	0.293026385
		Deviation	0.940042879	0.076479615
	Basic without 23	Absolute	0.000049458	0.907711717
		Deviation	0.000373469	0.303113862
	Features	Absolute	0.721120322	0.291554319
		Deviation	0.955858626	0.06094185
	Time	Absolute	0.006393081	0.993719286
		Deviation	**0.004485623**	**0.995593218**
	Time / Features	Absolute	0.007821783	0.992315695
		Deviation	**0.006839528**	**0.993280685**
Linear Regression	Basic	Absolute	0.956126050	0.060679125
		Deviation Values	0.990504561	0.026904862
	Basic without 23	Absolute	0.000433002	0.192026803
		Deviation	0.000482738	0.099220158
	Features	Absolute	0.952170747	0.06456491
		Deviation Values	0.990526349	0.026883459
	Time	Absolute	0.865657554	0.149557519
		Deviation Values	0.924199472	0.092044553
	Time / Features	Absolute	0.864756878	0.150442364
		Deviation Values	0.924192470	0.092051431
Multi Layer Perceptron	Basic	Absolute	**0.5879882**	**0.422346492**
		Deviation Values	1.1236988	-0.103948332
	Basic without 23	Absolute	0.000103044	0.807722808
		Deviation	0.000713551	-0.331469635
	Features	Absolute	**0.5736901**	**0.436393226**
		Deviation Values	1.0701481	-0.051338836
	Time	Absolute	0.5962338	0.414245851
		Deviation Values	0.10961433	0.892312286
	Time / Features	Absolute	0.5873553	0.42296827
		Deviation Values	0.7038876	0.308484187

Table 4.1. Evaluation of prediction for eqn. 1.1, old UC&ED

Model	Data	Values	Mean Squared Error	R2 Score
Random Forest	Basic	Absolute	0.379733811	0.8167131
		Deviation	0.581032674	0.719551764
	Features	Absolute	0.276349765	0.866613691
		Deviation	0.571131357	0.724330853
	Time	Absolute	0.144980706	0.930021864
		Deviation	0.129463471	0.9375116
	Time / Features	Absolute	0.13114375	0.936700576
		Deviation	0.131535028	0.936511717
ExtraTrees	Basic	Absolute	0.234034669	0.887038005
		Deviation	0.5868895	0.716724837
	Features	Absolute	**0.215895248**	**0.895793397**
		Deviation	0.576514586	0.721732519
	Time	Absolute	**0.089706922**	**0.956700975**
		Deviation	0.143872871	0.930556585
	Time / Features	Absolute	**0.079740149**	**0.961511658**
		Deviation	0.142102105	0.931411284
Linear Regression	Basic	Absolute	1.766060799	0.1475718
		Deviation	1.742061735	0.159155478
	Features	Absolute	1.764935974	0.148114722
		Deviation	1.742808771	0.158794905
	Time	Absolute	1.71072047	0.174283031
		Deviation	1.632839843	0.211873833
	Time / Features	Absolute	1.710137572	0.17456438
		Deviation	1.633427139	0.211590362
Multi Layer Perceptron	Basic	Absolute	**0.21204334**	**0.897652595**
		Deviation	0.7531615	0.636470007
	Features	Absolute	0.24318098	0.882623331
		Deviation	0.6935733	0.665231567
	Time	Absolute	0.19361906	0.906545479
		Deviation	0.4166859	0.798877343
	Time / Features	Absolute	0.24176078	0.883308823
		Deviation	0.6461919	0.688101206

Table 4.2. Evaluation of prediction for eqn. 1.1, new UC&ED

Model	Data	Values	Accuracy
Random Forest	Basic	Absolute	**0.967533819**
		Deviation	0.772320499
	Features	Absolute	**0.970447451**
		Deviation	0.778355879
	Time	Absolute	**0.977939646**
		Deviation	0.904474506
	Time / Features	Absolute	**0.979396462**
		Deviation	0.917585848
ExtraTrees	Basic	Absolute	0.944016649
		Deviation	0.744432882
	Features	Absolute	0.946722164
		Deviation	0.748595213
	Time	Absolute	0.957752341
		Deviation	0.859313215
	Time / Features	Absolute	0.959833507
		Deviation	0.867013528
Logistic Regression	Basic	Absolute	0.769614984
		Deviation	0.664516129
	Features	Absolute	0.767533819
		Deviation	0.645577523
	Time	Absolute	0.769406868
		Deviation	0.704682622
	Time / Features	Absolute	0.767741935
		Deviation	0.705515088
Multi Layer Perceptron	Basic	Absolute	0.944224766
		Deviation	0.767117586
	Features	Absolute	0.937148803
		Deviation	0.779396462
	Time	Absolute	0.940270552
		Deviation	0.88491155
	Time / Features	Absolute	0.93673257
		Deviation	0.867637877

Table 4.3. Accuracy for the old UC&ED

Model	Data	Values	Accuracy
Random Forest	Basic	Absolute	0.922996878
		Deviation	0.780437045
	Features	Absolute	**0.931529657**
		Deviation	0.77710718
	Time	Absolute	**0.95359001**
		Deviation	0.933610822
	Time / Features	Absolute	**0.959001041**
		Deviation	0.937773153
ExtraTrees	Basic	Absolute	0.893652445
		Deviation	0.749843913
	Features	Absolute	0.912174818
		Deviation	0.752341311
	Time	Absolute	0.915712799
		Deviation	0.90364204
	Time / Features	Absolute	0.934027055
		Deviation	0.913631634
Logistic Regression	Basic	Absolute	0.709469303
		Deviation	0.664932362
	Features	Absolute	0.710718002
		Deviation	0.651404787
	Time	Absolute	0.704682622
		Deviation	0.769406868
	Time / Features	Absolute	0.710093652
		Deviation	0.768366285
Multi Layer Perceptron	Basic	Absolute	**0.9250780437**
		Deviation	0.779812695
	Features	Absolute	0.921540062
		Deviation	0.763787721
	Time	Absolute	0.922164412
		Deviation	0.926118626
	Time / Features	Absolute	0.922372529
		Deviation	0.912382934

Table 4.4. Accuracy for the new UC&ED

Model	Data	Values	Mean Squared Error	R2 Score
Random Forest	Basic	Absolute	**454.2787979**	**0.870779585**
		Deviation	1589.114006	0.547973685
	Features	Absolute	**442.0468269**	**0.874258991**
		Deviation	1547.114119	0.559920627
	Time	Absolute	**346.3173653**	**0.901489407**
		Deviation	444.6982529	0.873504788
	Time / Features	Absolute	**311.4572803**	**0.911405421**
		Deviation	376.9464414	0.892776912
ExtraTrees	Basic	Absolute	501.4042585	0.857374664
		Deviation	1670.23334	0.524899145
	Features	Absolute	465.5728546	0.867566971
		Deviation	1618.845249	0.53951658
	Time	Absolute	382.9259138	0.89107604
		Deviation	475.340169	0.864788641
	Time / Features	Absolute	334.2785563	0.904913868
		Deviation	401.6438177	0.885751699
Linear Regression	Basic	Absolute	2004.297826	0.429873906
		Deviation	2378.713224	0.323370778
	Features	Absolute	1992.289078	0.433289816
		Deviation	2384.317658	0.321776587
	Time	Absolute	1967.103849	0.440453799
		Deviation	2334.518107	0.335942158
	Time / Features	Absolute	1964.037336	0.441326074
		Deviation	2340.451337	0.33425444
Multi Layer Perceptron	Basic	Absolute	791.6782	0.774805729
		Deviation	2209.7493	0.371432887
	Features	Absolute	853.3027	0.757276546
		Deviation	2068.0288	0.411745532
	Time	Absolute	784.93945	0.776722594
		Deviation	1445.2084	0.588907882
	Time / Features	Absolute	810.3605	0.769491533
		Deviation	1472.5276	0.581136903

Table 4.5. Evaluation of prediction for eqn. 1.19, old UC&ED

Model	Data	Values	Mean Squared Error	R2 Score
Random Forest	Basic	Absolute	**586.175182**	**0.706156689**
		Deviation	771.345808	0.613332647
	Features	Absolute	**535.2820947**	**0.731668846**
		Deviation	746.7285477	0.625673015
	Time	Absolute	**479.2200189**	**0.759772161**
		Deviation	548.0422622	0.725272312
	Time / Features	Absolute	**382.3788546**	**0.808317595**
		Deviation	464.7858931	0.767007833
ExtraTrees	Basic	Absolute	598.5920412	0.699932251
		Deviation	801.6988748	0.598116981
	Features	Absolute	568.4132113	0.715060574
		Deviation	777.316516	0.610339595
	Time	Absolute	504.7101396	0.746994238
		Deviation	552.3087257	0.723133579
	Time / Features	Absolute	411.398436	0.793770391
		Deviation	493.045907	0.752841393
Linear Regression	Basic	Absolute	1018.395776	0.489489157
		Deviation	1200.406465	0.398249157
	Features	Absolute	1014.41408	0.491485138
		Deviation	1207.37017	0.394758327
	Time	Absolute	1010.568891	0.49341269
		Deviation	1195.212309	0.400852931
	Time / Features	Absolute	1009.888051	0.493753988
		Deviation	1202.13274	0.397383794
Multi Layer Perceptron	Basic	Absolute	678.7301	0.659759897
		Deviation	1297.2612	0.349696889
	Features	Absolute	652.9677	0.672674311
		Deviation	1000.7623	0.498328622
	Time	Absolute	677.8122	0.660220017
		Deviation	1054.253	0.471514239
	Time / Features	Absolute	656.0337	0.671137337
		Deviation	1034.7493	0.481291375

Table 4.6. Evaluation of prediction for eqn. 1.19, new UC&ED

Chapter 5

Additional Experiments

A few other experimentation's have been conducted to attempt both to improve the results obtained at chapter 4, and to propose a more concrete solution to use machine learning as a speedup for the simulation process. Both experiments conducted and results obtained are presented in this chapter. Results won't be presented as in-depth as previously as a lot of information would be redundant.

5.1 Improvements

5.1.1 Improvement Idea

A first idea for improvement is to add additional input variables to the model so that it can achieve better performance. Adding more data is usually valuable, as the models have more parameters on which to base their decision.

At this point, the most straightforward way to increase the number of variables is to combine the absolute and deviation inputs presented in Chapter 2 in a single dataset. This dataset will have 194 inputs, 97 absolute values and 97 deviation values. As was seen in the results of chapter 4 (tables 4.1 to 4.6), both absolute and deviation values can present good performances depending on what additional variable is associated to it (time for example) and what the target for training is.

The first experiment is to test the performance of these models on the prediction of equations 1.1 and 1.19, using the same 1000 scenarios as previously. At the same time, a second experiment is performed with an additional 1000 scenarios, effectively doubling the size of both the training and the test sets. This will allow to visualize the possible improvements that could be obtained from simply adding more data. A high difference would indicate that the best way to improve the model is simply to add more data, while a low or no difference indicates that the model reached close to it's peak performance. Note that, while in chapter 4 all results were evaluated on the same test set, this new dataset with doubled size has a test set that differs from the ones considered previously. Not only is it twice as large, but it is also comprised of different scenarios. Differences in performance should thus be taken with caution, as a small increase or decrease in performance could be attributed to this difference in test set.

Only the new UC&ED will be analyzed in this section, as this is the one that is currently in use for the simulations.

5.1.2 Improvement Result

The first results are presented the table 5.1 and 5.2 where table 5.1 shows the performance for predicting the objective value of equation 1.1 and 5.2 the margin value of equation 1.19. The tables do not include results for datasets containing the time of the day or additional features (see section 2.4). Including the time didn't show any improvements, which indicates that all the information contained can be inferred from the other values, even by the decision trees. Adding extra features has been explored but not fully studied. According to first results, they do not help in predicting the values of equation 1.1 but help in predicting the value of equation 1.19.

The results of table 5.1 show improvements over the ones obtained from table 4.2, which indicates that combining both absolute and deviation features is the best approach. Results similar to the previous results including the time (see table 4.2) are obtained for random forest and extra trees, which confirms the idea that time information can be retrieved from the given set of values. Multi layer perceptron improves the previous best model by 3 percent, and improves by 9.4 percent upon the previous best MLP model.

Results on margin are more mixed, as no model managed to reach the 0.80 score obtained in table 4.6. While thorough research has not been performed on this subject, it appears that adding back the features improves the performance by about 2 percent, which would give a similar result to the one obtained in table 4.6. Nevertheless, performance on margin is still somewhat unsatisfying.

The feature importances and the scatter plots showing the differences between ground truth and prediction are presented in the appendix B.

Model	Mean Squared Error (1000 scenarios)	R2 Score (1000 scenarios)	Mean Squared Error (2000 scenarios)	R2 Score (2000 scenarios)
Random Forest	0.137558253	0.933604475	0.140404107	0.926508378
Extra Trees	0.09717301	0.953097303	0.083552595	0.956266125
Multi Layer Perceptron	0.016907455	0.991839244	0.01651777	0.991354116

Table 5.1. Mean Squared Error and R2 score for objective value prediction

Model	Mean Squared Error (1000 scenarios)	R2 Score (1000 scenarios)	Mean Squared Error (2000 scenarios)	R2 Score (2000 scenarios)
Random Forest	453.9974189	0.772415979	614.3179838	0.704259924
Extra Trees	493.1018207	0.752813364	630.8655803	0.696293712
Multi Layer Perceptron	484.67596	0.757037162	569.1941	0.725983105

Table 5.2. Mean Squared Error and R2 score for margin prediction

5.2 Advanced classification

Instead of simply predicting 0 or 1 based on the value of equation 1.1 like in section 4.3, a third class has been added in this experiment. The goal of this third class is the include the value of the margin equation 1.19 in the classification. The three classes are 0 when margin is ≥ 0 (which implies that equation 1.1 ≤ 0.0005), 1 when $margin = 0$ and equation 1.1 ≤ 0.0005, and 2 otherwise. To perform the task of learning this distribution, the Random Forest and Extra Trees models are the same as the ones used for classification in chapter 4. The perceptron however is changed as a sigmoid activation function is not fitted for three class prediction. Instead of outputting a single number between 0 and 1, the model now outputs three numbers. Thanks to a softmax [9] activation layer, the output of the MLP can be converted to the form of a probability distribution across each class. From there, accuracy can be estimated by taking the argmax of the output as the predicted class. To train the MLP, cross entropy loss is used. The hyperparameters used are the same were chosen by evaluating different possibilities on a validation set. It is the same hyperparameters as previous classifications (See section 3.3.2), but 500 epochs. The training loss has been balanced to account for class imbalances by adjusting weights on the cross-entropy loss.

The results of the classification can be seen in table 5.3. Extra trees, random forest and MLP perform similarly to what is given in table 4.4, however the task is more challenging as there are three classes to predict from instead of 2. Contrarily to the regression problems, adding more data seems beneficial for this classification task, as all models improve by .5 to .8 percent accuracy. In the case of the MLP, cross entropy loss decreases from 0.58 to 0.50, which is a notable improvement.

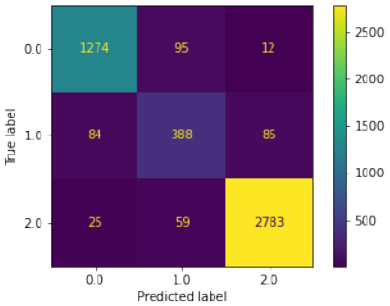
Figure 5.1 gives more information about the errors made by each model. Focus is made on 5.1f as it is the model with the best performance, and all confusion matrices follow a similar pattern. Classification for classes 0 and 2 are relatively strong with an accuracy of 95.7 and 97.8 percent respectively. Classification for class 1 is the most error prone, with an accuracy of just 71.9 percent. This error is obtained despite the rescaling performed in the loss of the MLP to account for class imbalances. Lack of data might still be one explanation for this situation. However, the class 1 is the class with the strictest requirements (eqn. 1.19 $= 0$ and eqn. 1.1 < 0.0005), which also influences the error rate of the prediction.

Model	Accuracy (1000)	Accuracy (2000)
Random Forest	0.925078044	0.930452889
Extra Trees	0.903850156	0.910046851
Multi Layer Perceptron	0.934027055	0.935137949

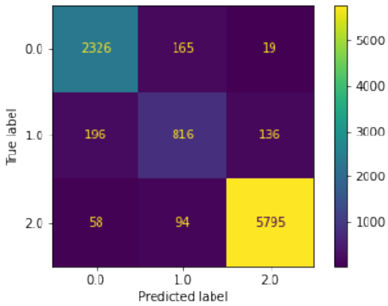
Table 5.3. Accuracy for 3-class prediction

Given these results, it seems unlikely that the classification prediction can be used as is to replace the simulation, as 6.7 percent of misclassification is still high.

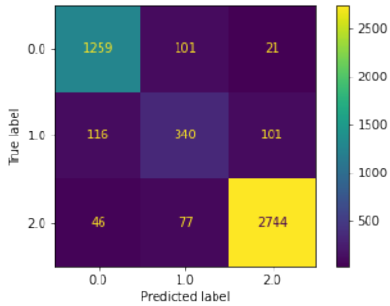
Another possibility is to only look at the classifier for the class 2 (objective value > 0.0005). In the entire training set, there are 4231 times where the prediction of the model for the class 2 is over 34 (before applying the softmax activation). Among these 4231 predictions,



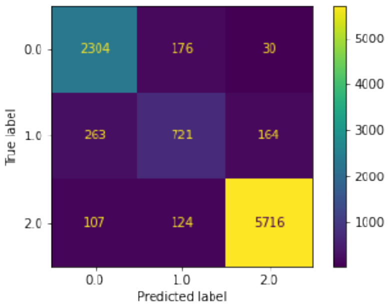
(a) Random Forest (1000)



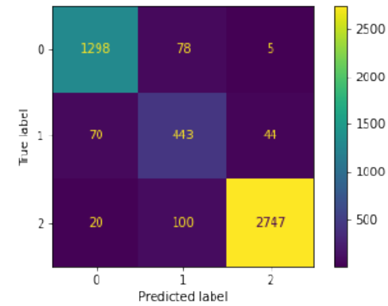
(b) Random Forest (2000)



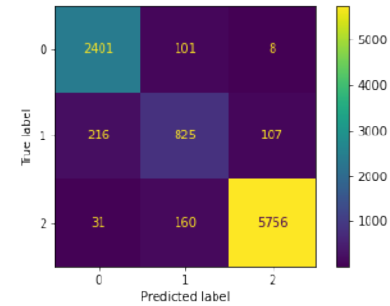
(c) Extra Tree (1000)



(d) Extra Tree (2000)



(e) Multi Layer Perceptron (1000)



(f) Multi Layer Perceptron (2000)

Figure 5.1. Confusion matrix

the accuracy exceeds 0.999, suggesting less than one error for every 1000 entries. On these predictions, the best model shown at section 5.1 can then be applied to estimate the equation value. The other should still be evaluated through the simulator, as both this classification and the models shown in table 5.2 have difficulties predicting the margin value. Assuming that the time to get the predictions from the machine learning model is negligible compared to running the simulation, this would still result in a speedup of about 44 percent (the proportion of predictions that fit the criteria).

To obtain a better speedup, it is necessary to perform more analysis on the margin (eqn. 1.19), and why it is so difficult to predict.

Conclusion

This paper aimed to understand and analyze machine learning results for tasks predicting the reliability of a power grid. Chapter 2 showed interesting results by analysing the forecasts as well as the unit commitment, objective value (eqn. 1.1) and margin value (eqn. 1.19) for both old and new UC&ED. It was shown that the old UC&ED presents issues at the last hour of the day which were corrected in the new one. Furthermore, it displayed the distributions of equations 1.1 and 1.19 per hour of the day, which gives valuable insight into the day-ahead situation. Finally, it also highlighted multiple particularities of the Belderbos model, like the high renewable energy generation.

Afterwards, a large number of experiments were conducted. Chapter 4 studied the performance of four machine learning algorithms on different predictive tasks. Encouraging results were obtained for the prediction of equation 1.1 and more mixed results for margin equation 1.19. Feature importance analysis showed that the old UC&ED requires the time of the day to make decent predictions, which is influenced by the extreme outliers observed at the last hour of the day. Otherwise, it showed that wind variables are usually more important that load or solar variables to make predictions about 1.1, while equations 1.19 showed stronger correlation with solar values.

Chapter 5 then performed additional experiments that improved the predictions across all tasks. Chapter 5 also suggested a way to utilize a classification task to speedup the simulation process. However, this suggestion should be taken with a grain of salt, as no extensive testing has been done to confirm that it works.

In the future, it would be valuable to experiment more with hyperparameter tuning, especially for the multilayer perceptron. Due to the time it takes to perform cross-validation, only a few values were tried. Given more computational resources, performing more tuning should help to improve performance. Another study to perform would be to look more in detail at the margin values and understand why no model is able to predict it satisfyingly.

Bibliography

[1] "Belgium's 2021 electrivity mix: record number of exports due to slight increase in production of renewable energy and a stable nuclear fleet". In: *Belgium's 2021 electricity mix* (Jan. 2022). URL: https://www.elia.be/en/news/press-releases/2022/01/20220107_belgium-2021-electricity-mix (page 1).

[2] "EPOCH Belgium". https://www.epocbelgium.be/en. Accessed: 2022-09-23 (page 1).

[3] Andreas Belderbos et al. "Facilitating renewables and power-to-gas via integrated electrical power-gas system scheduling". In: *Applied Energy* 275 (2020), p. 115082. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2020.115082. URL: https://www.sciencedirect.com/science/article/pii/S0306261920305948 (page 2).

[4] George Papamakarios et al. "Normalizing Flows for Probabilistic Modeling and Inference". In: (2019). DOI: 10.48550/ARXIV.1912.02762. URL: https://arxiv.org/abs/1912.02762 (page 2).

[5] Jonathan Dumas et al. "A deep generative model for probabilistic energy forecasting in power systems: normalizing flows". In: *Applied Energy* 305 (2022), p. 117871. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2021.117871. URL: https://www.sciencedirect.com/science/article/pii/S0306261921011909 (page 2).

[6] Laurine Duchesne, Efthymios Karangelos, and Louis Wehenkel. "Recent Developments in Machine Learning for Energy Systems Reliability Management". In: *Proceedings of the IEEE* 108.9 (2020), pp. 1656–1676. DOI: 10.1109/JPROC.2020.2988715 (page 5).

[7] Leo Breiman. "Random Forests". In: *Machine Learning* 45.1 (Oct. 2001), pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324 (page 13).

[8] Pierre Geurts, Damien Ernst, and Louis Wehenkel. "Extremely randomized trees". In: *Machine Learning* 63.1 (Apr. 2006), pp. 3–42. ISSN: 1573-0565. DOI: 10.1007/s10994-006-6226-1. URL: https://doi.org/10.1007/s10994-006-6226-1 (page 14).

[9] John S. Bridle. "Training Stochastic Model Recognition Algorithms as Networks Can Lead to Maximum Mutual Information Estimation of Parameters". In: *Proceedings of the 2nd International Conference on Neural Information Processing Systems*. NIPS'89. Cambridge, MA, USA: MIT Press, 1989, pp. 211–217 (page 35).

Appendix A

Detailed Feature importance

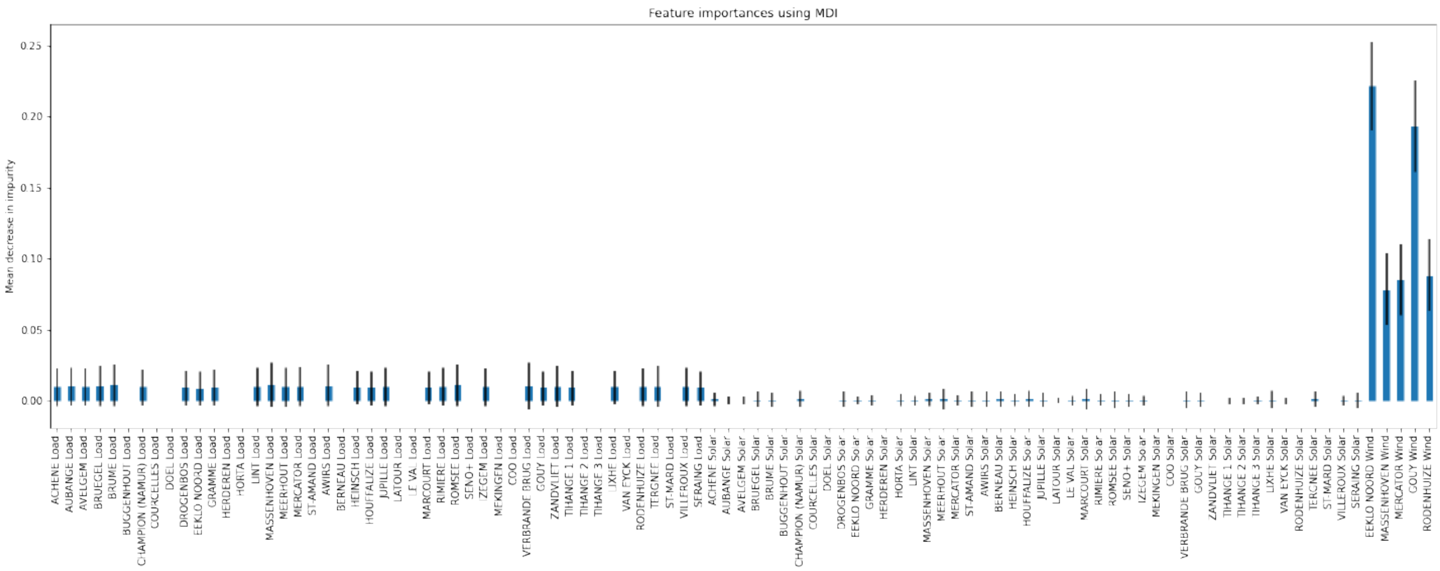


Figure A.1. Feature Importance basic random forest old UC&ED

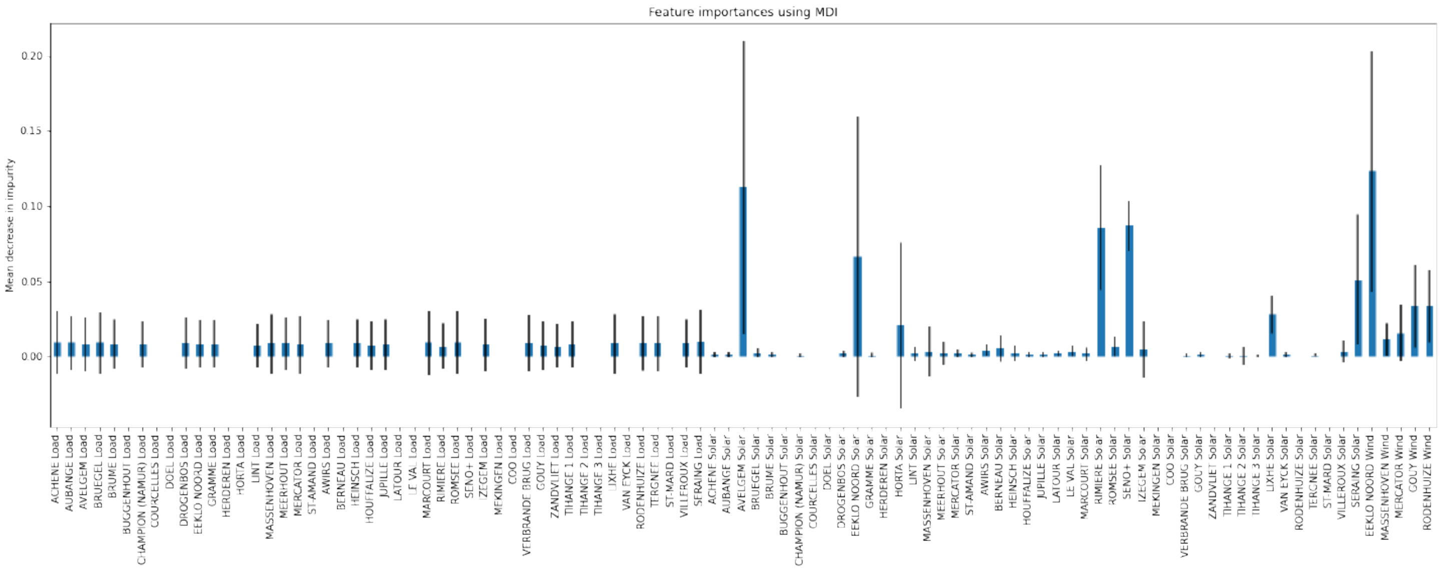


Figure A.2. Feature Importance no 23 random forest old UC&ED

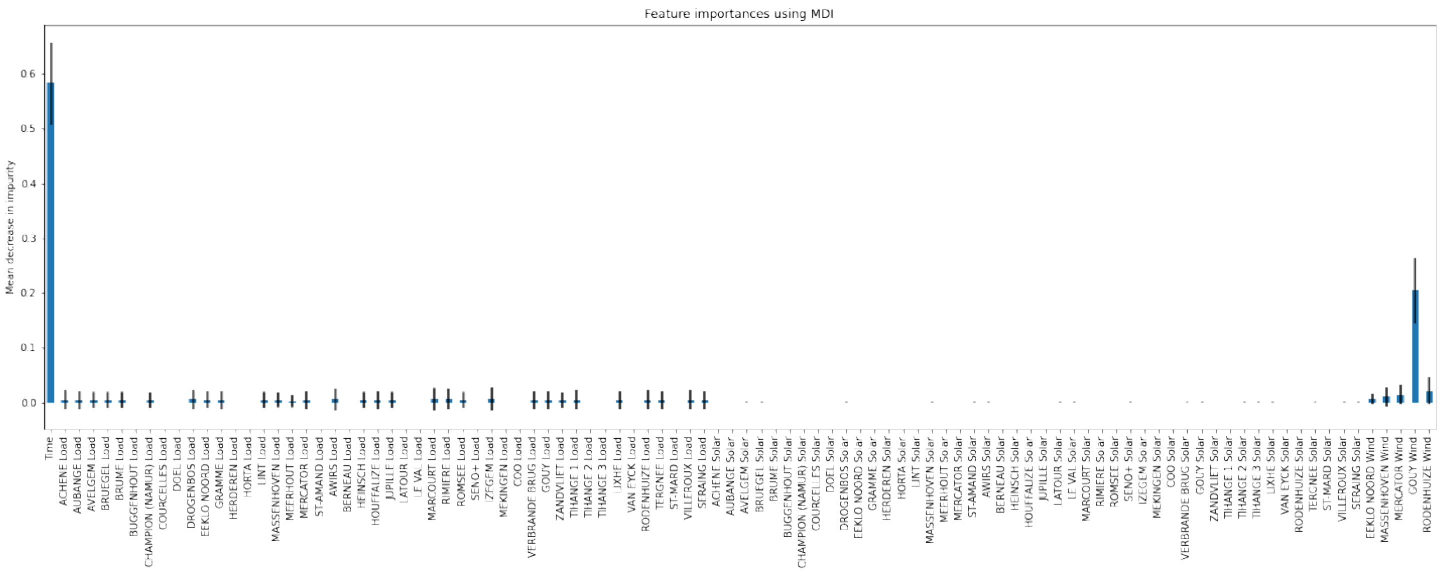


Figure A.3. Feature Importance Hour ExtraTrees old UC&ED

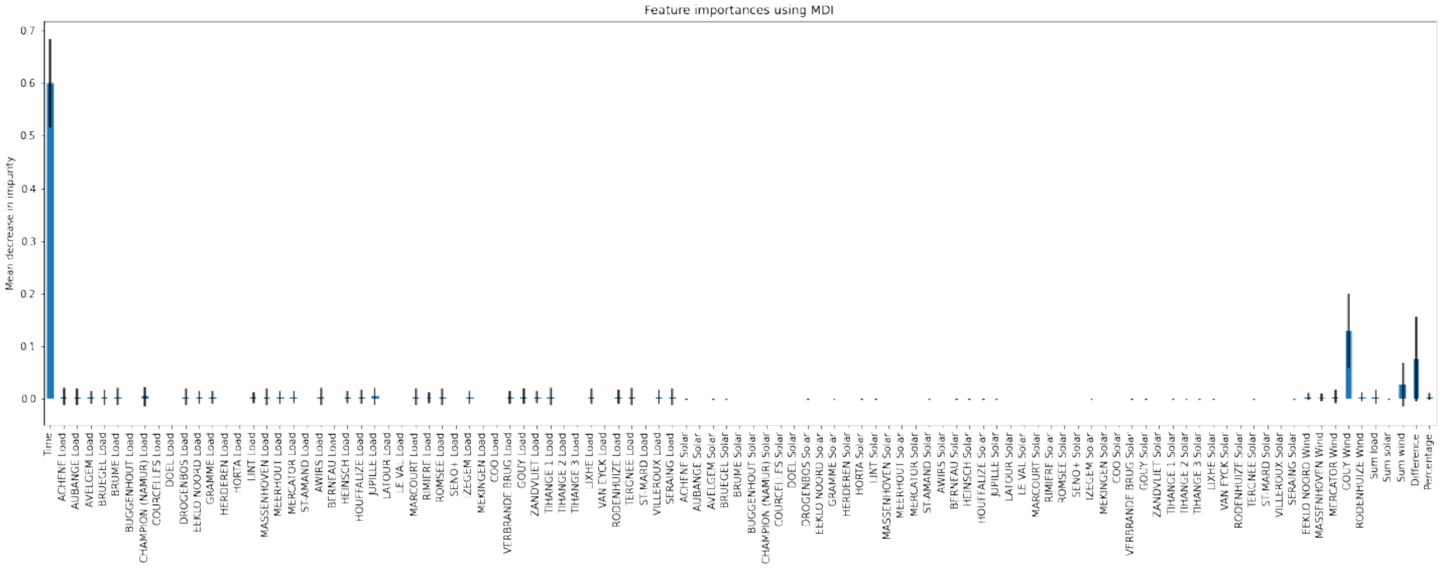


Figure A.4. Feature Importance Hour/Features ExtraTrees old UC&ED

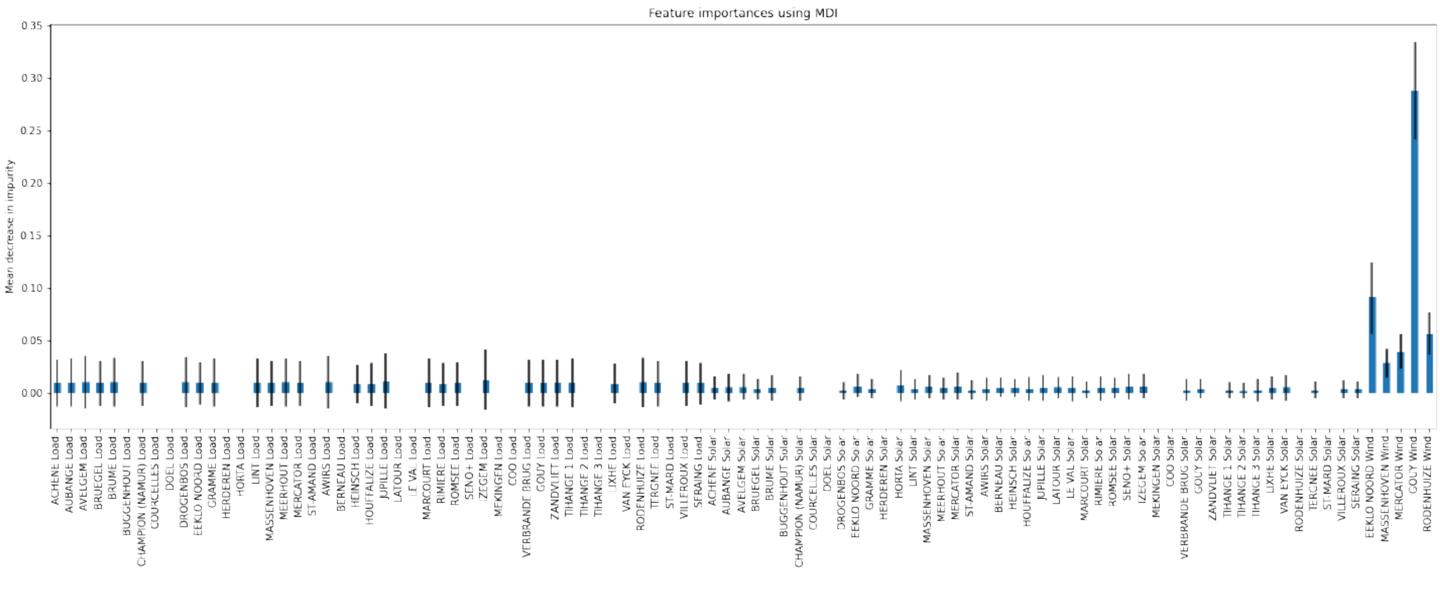


Figure A.5. Feature Importance Basic ExtraTrees new UC&ED

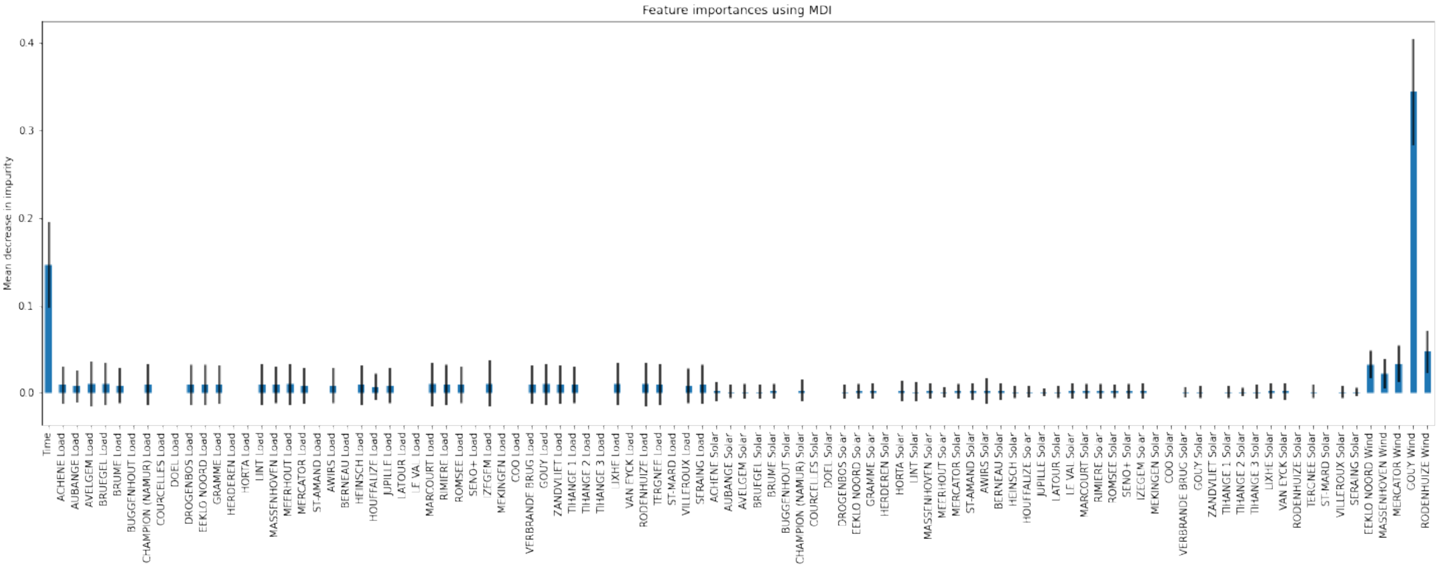


Figure A.6. Feature Importance Hour ExtraTrees new UC&ED

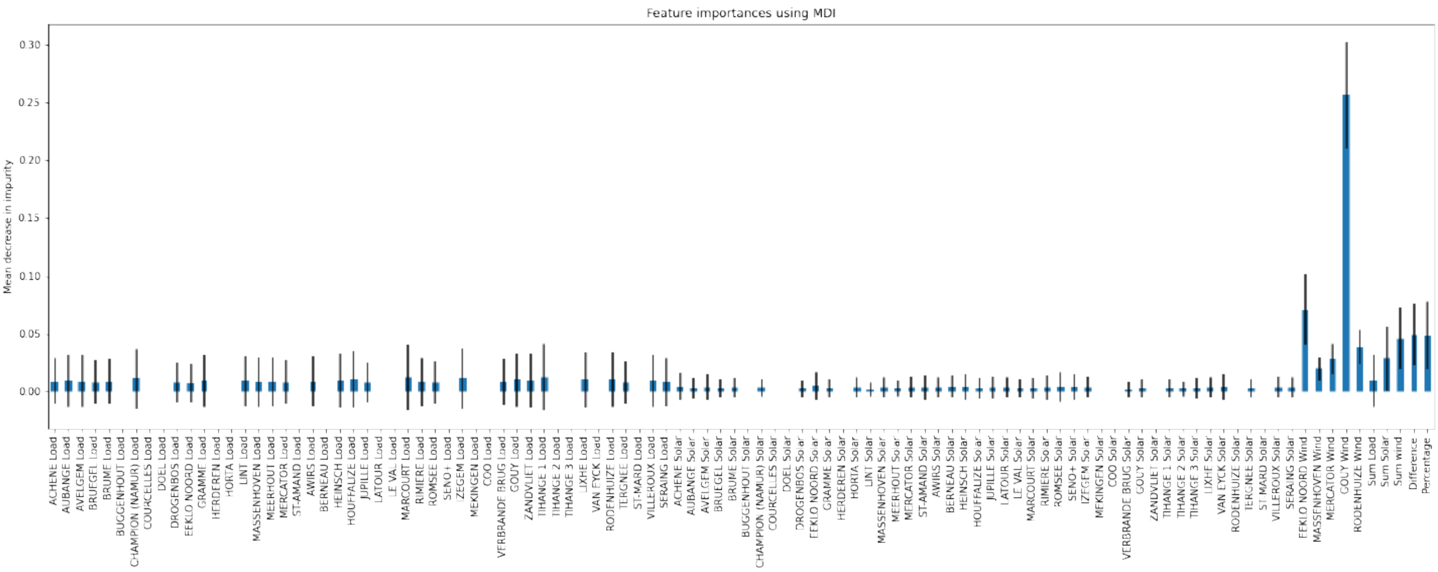
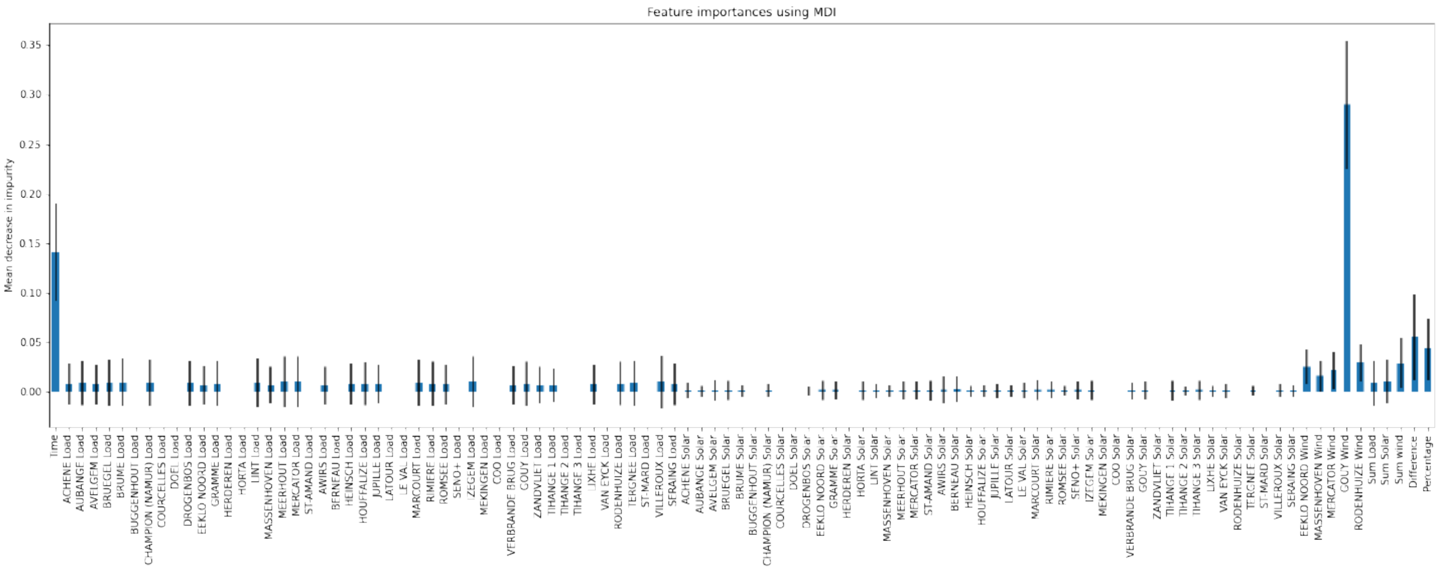


Figure A.7. Feature Importance Addditional Features ExtraTrees new UC&ED

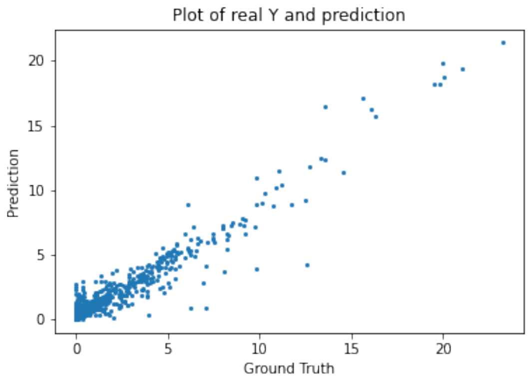


(a) Hour / Features Extra Trees

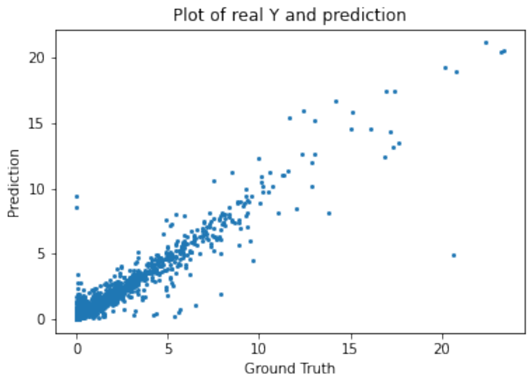
Figure A.8. Feature Importance Hour/Features ExtraTrees new UC&ED

Appendix B

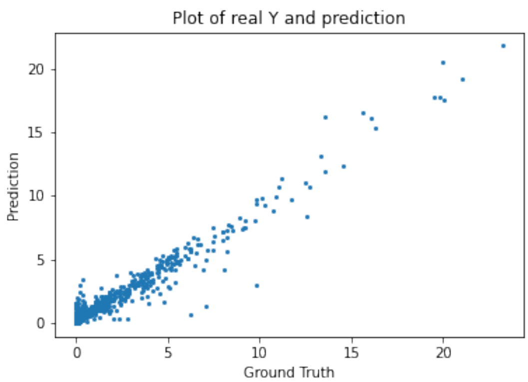
Additional experiments (Ground Truth and Feature Importance)



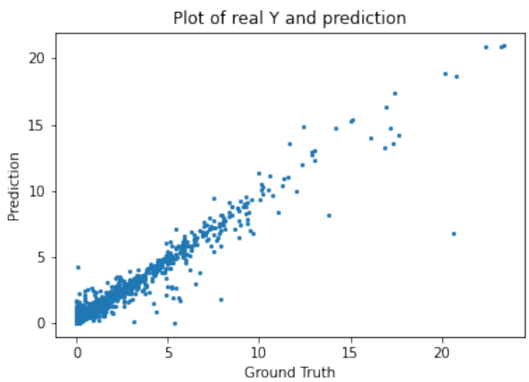
(a) Random Forest (1000 scenarios)



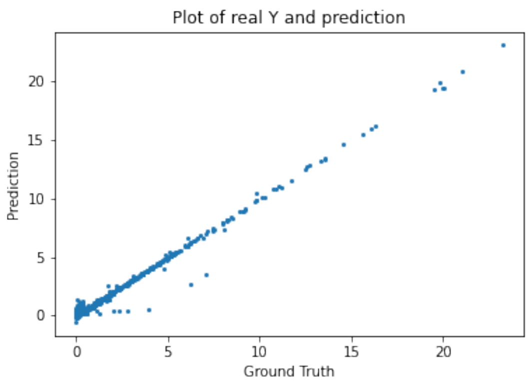
(b) Random Forest (2000 scenarios)



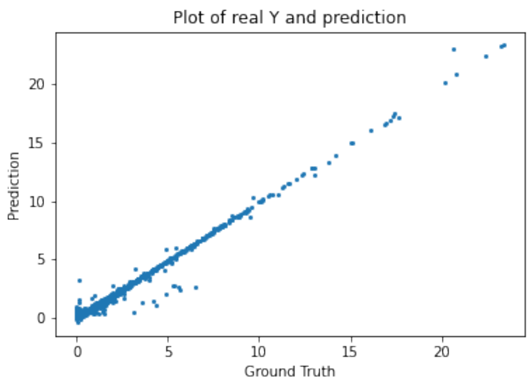
(c) Extra Trees (1000 scenarios)



(d) Extra Trees (2000 scenarios)

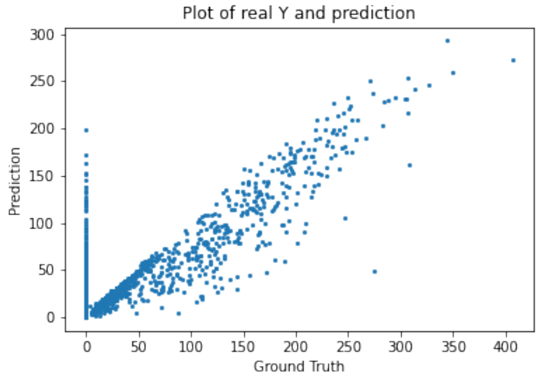


(e) MLP (1000 scenarios)

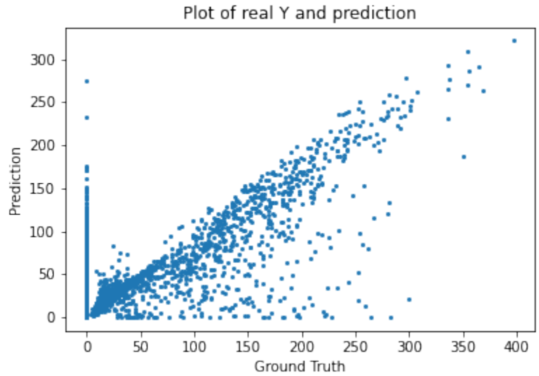


(f) MLP (2000 scenarios)

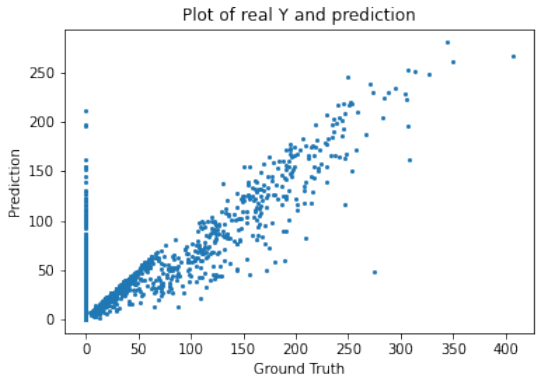
Figure B.1. Ground Truth vs Prediction for objective value (eqn. 1.1)



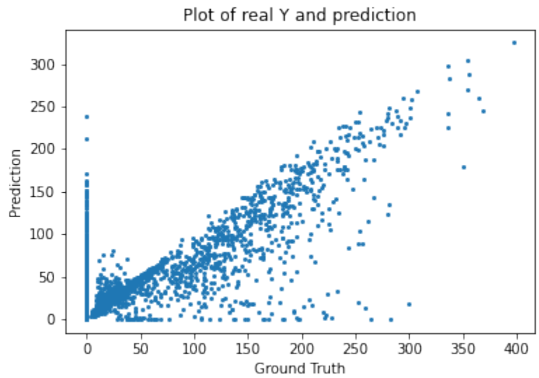
(a) Random Forest (1000 scenarios)



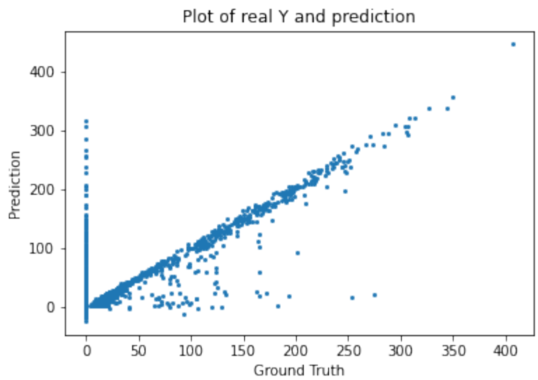
(b) Random Forest (2000 scenarios)



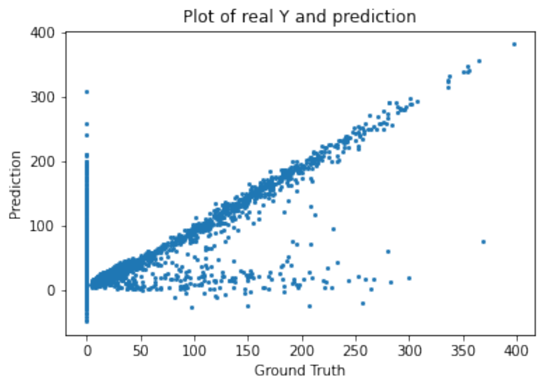
(c) Extra Trees (1000 scenarios)



(d) Extra Trees (2000 scenarios)

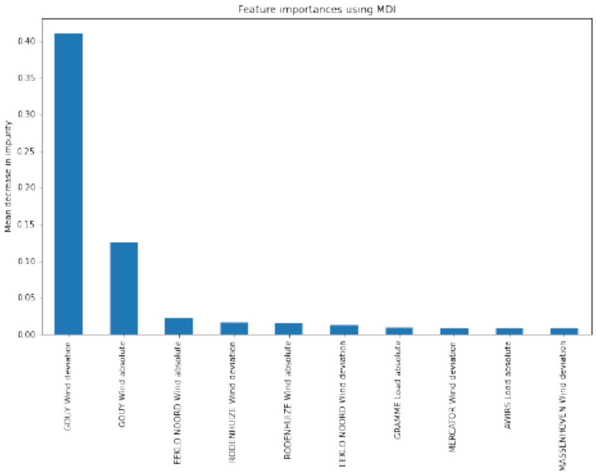


(e) MLP (1000 scenarios)

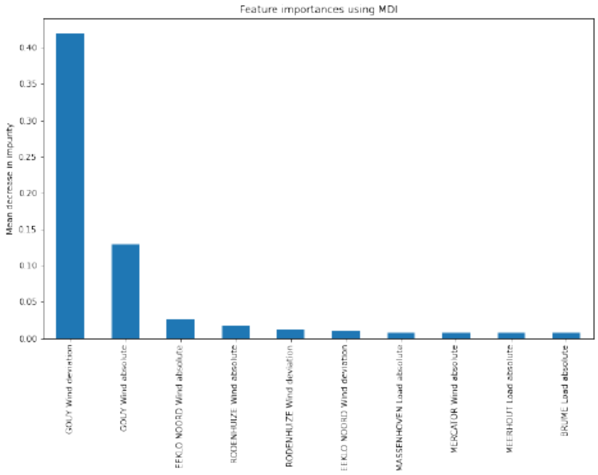


(f) MLP (2000 scenarios)

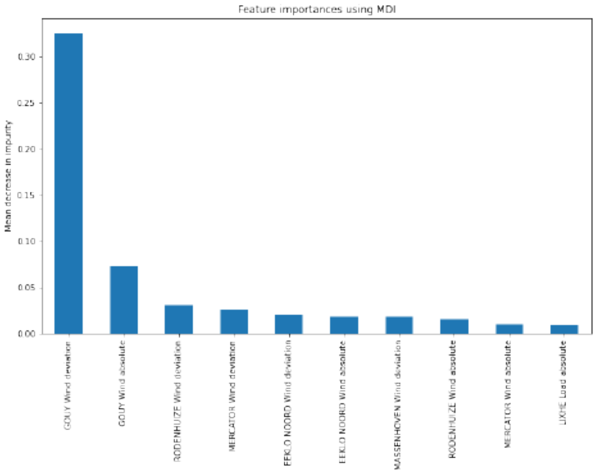
Figure B.2. Ground Truth vs Prediction for margin value (eqn. 1.19)



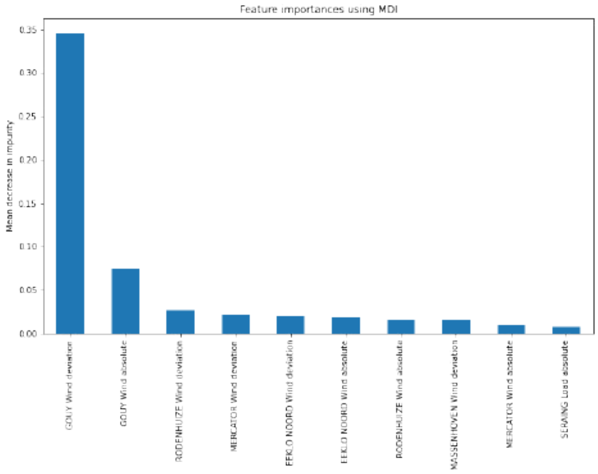
(a) Random Forest (1000 scenarios)



(b) Random Forest (2000 scenarios)

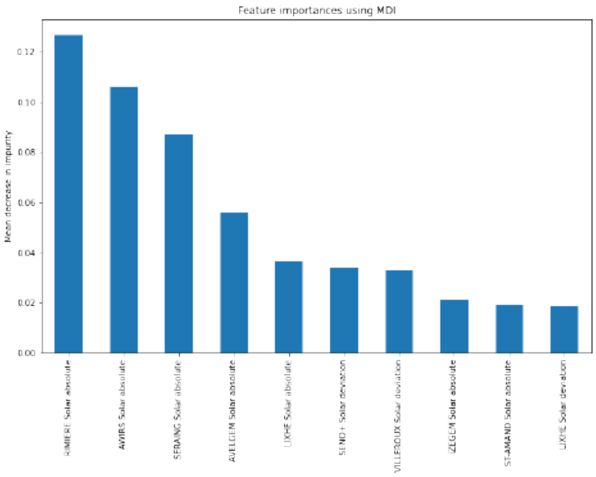


(c) Extra Trees (1000 scenarios)

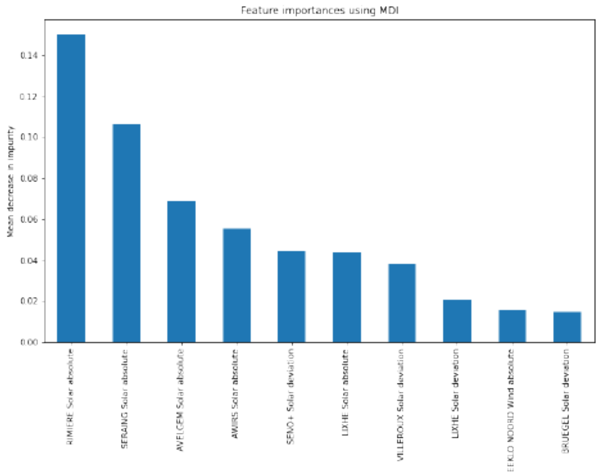


(d) Extra Trees (2000 scenarios)

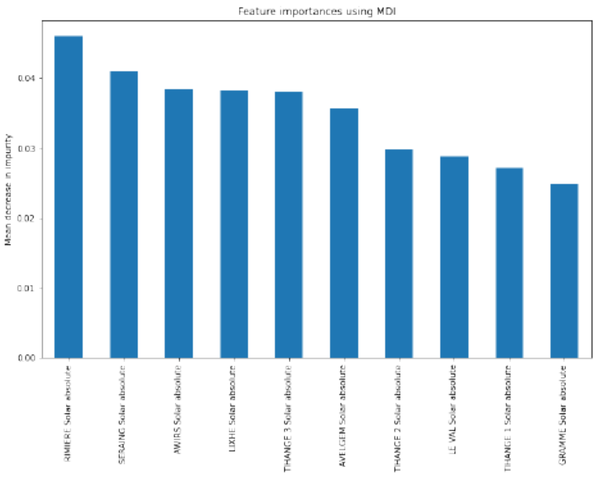
Figure B.3. Feature importance for objective value (eqn. 1.1)



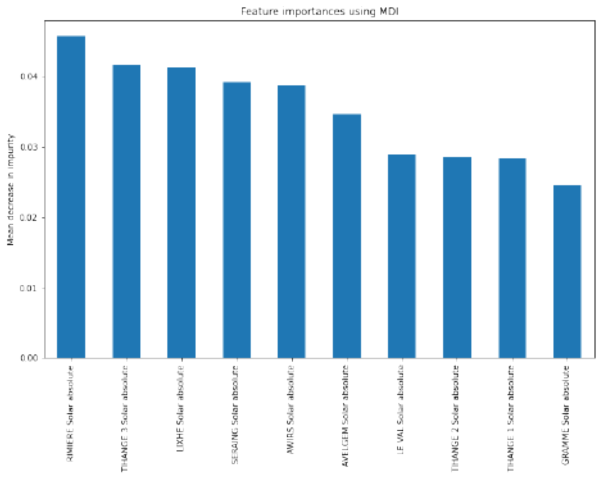
(a) Random Forest (1000 scenarios)



(b) Random Forest (2000 scenarios)

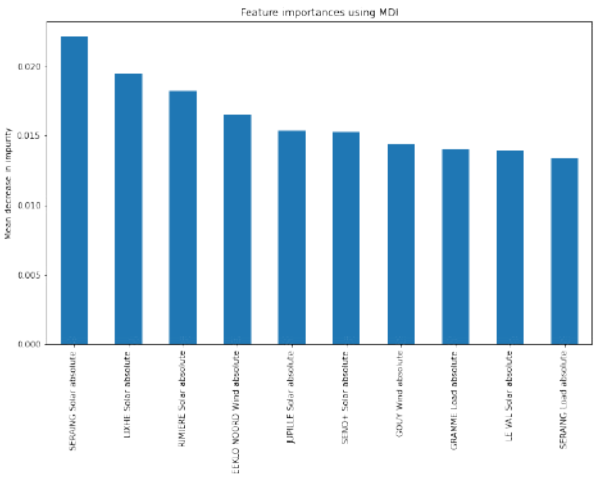


(c) Extra Trees (1000 scenarios)

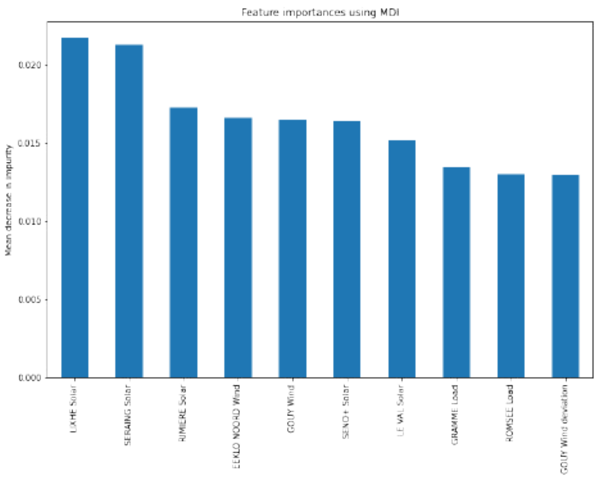


(d) Extra Trees (2000 scenarios)

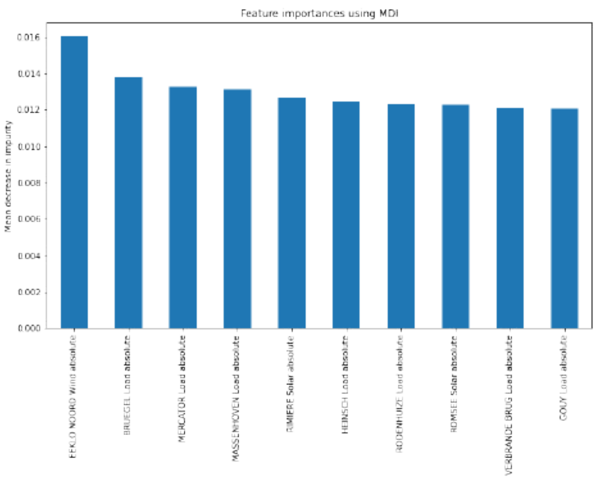
Figure B.4. Feature importance for margin value (eqn. 1.19)



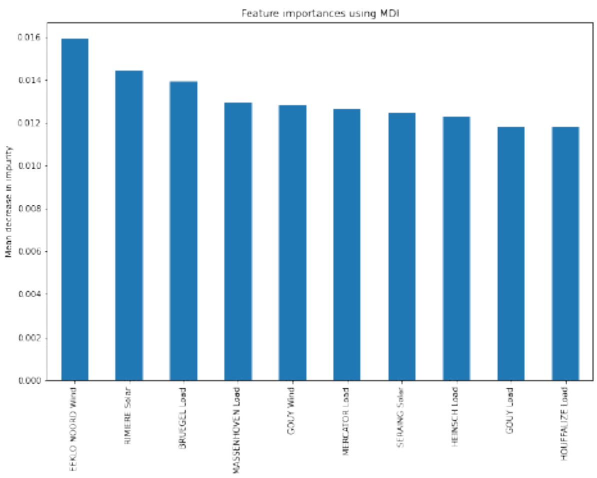
(a) Random Forest (1000 scenarios)



(b) Random Forest (2000 scenarios)



(c) Extra Trees (1000 scenarios)



(d) Extra Trees (2000 scenarios)

Figure B.5. Feature importance for advanced classification