edgeFLEX

D3.2

Report on VPP Optimisation, V1.

The research leading to these results has received funding from the European Union's Horizon 2020 Research and Innovation Programme, under Grant Agreement no 883710.

Project Name	edgeFLEX
Contractual Delivery Date:	31.03.2021
Actual Delivery Date:	29.03.2021
Author(s):	Vadim Omelcenko, Kévin Férin
Workpackage:	WP3 – Optimization of a VPP consisting of variable and dispatchable RES
Security:	P = Public
Nature:	R
Version:	V 1.0
Total number of pages:	**49**

Abstract
In this study, we explore the optimization of virtual power plants (VPP), consisting of a portfolio of biogas power plants, a battery and a set of intermittent sources such as wind and solar. We operate under price and weather uncertainty and in order to handle it, we employ methods of machine learning. For price modelling, we take into account the latest trends in the field and the most up-to-date events affecting the day-ahead and intra-day prices. We demonstrate the performance of the price models by both statistical methods and improvements in the profits of the virtual power plant. Optimization methods will take price and weather forecasts as input and conduct computer solving parallelization, decomposition, and splitting methods in order to handle sufficiently large numbers of biogas power plants and intermittent sources in a VPP. Finally, we demonstrate the positive social impact of such VPPs and the proposed strategies.

Keyword list
Virtual Power Plant, Gradual Increase, Warm Start, Proximal Jacobian ADMM, Multichannel Singular Spectrum Analysis

Executive Summary

In this report, we consider balancing the electricity generated by a wind park with the electricity generated by a portfolio of controllable assets: biogas-fired power plants and batteries. Prior to being included in the VPP, the bio-gas power plants may participate on all existing electricity markets: bilateral, intraday, day ahead, ancillary service, balancing, etc. By becoming part of the VPP, the biogas powered plant is supporting the optimization of incomes received by the conjunction of all participating members: wind power parks and batteries as well. As a result, the aggregated financial outcome is more favourable than what would be the sum of the independent outcomes. The achievement of this task is challenging in terms of optimization and price forecasting methods, and the authors focused on both.

The optimization is of mixed integer type and the authors developed complex decomposition methods to solve the issues.

Several methods based on machine learning have been tested for the wind output forecasts. The goal was to maximizing the resulting cumulative revenue under operational and physical constraints, and given the predictions available for the intermittent energies outputs and for the market. We also utilize commercial forecasts and our forecasting methods turned out to be competitive with these forecasts in terms of the resulting cumulative revenue. We also pay special attention to the robustness of the VPP to handle large amounts of assets within it; at measure that the number of assets in the VPP increases, this becomes more important, hence for the large scale deployment of the optimization algorithm, the speed and safety of convergence that is demonstrated in this document matters most. Besides, we utilize Robust Model Predictive Control in order to take many scenarios into account during decision making.

Authors

Partner	Name	e-mail
ALPQ		
	Vadim Omelčenko	vadim.omelcenko@alpiq.com
	Valery Manokhin	valery.manokhin.2015@live.rhul.ac.uk
	Alexandre Juncker	alexandre.juncker@alpiq.com
	Kévin Férin	kevin.ferin@alpiq.com

Table of Contents

1. Introduction

The optimization of virtual power plants (VPP) operation is of crucial importance for it enables because allows a more efficient incorporation of distributed energy resources (DER) into the grid, and thereby contribute to the achievement of goals associated with ecology. The electricity generation powered by solar and wind energy sources it is characterised by a high degree of uncertainty due to weather dependency, and require constant balancing from other controllable electricity generators. From the ecological point of view, it is desirable that the balancing is carried out by controllable generators powered by non-fossil fuels.

In edgeFLEX project, the balancing of the wind parks has been provided by means of pools of biogas (renewable fuel) power plants, and pools of batteries. The calculations have been accelerated by means of mathematical methods of decomposition and splitting, leveraging special structure of problems, parallelization, and the brute force power of optimization solvers. For the optimization of the VPP's operation, we use state of the art commercial wind forecasts in order to nominate the amounts of energy that our wind turbines are able to produce, and the goal of the flexible assets is to handle the aggregate imbalances in both directions: biogas power plants can balance only deficiencies while batteries can also balance surpluses.

The goal of the biogas power plants and batteries is to maximize their revenues from selling electricity on the day-ahead and intra-day markets while providing balancing to the pool of the DERs, as is set-up as a problem in §2. One important research component within the optimization of VPPs operation is the method known as Alternating Direction Method of Multipliers (ADMM), which is a technique based on replacing the solution of a large-scale optimization problem with an iterative procedure involving solutions of a large number of small subproblems. [6], [17]. A special facet of this method known as Proximal Jacobian ADMM (as will be explained in § 7.3) is proven to (1/k)-converge for linear programming problems and to be amenable to parallelization [8]. This algorithm will also be used in our research. In this study, we apply this method for optimizing large pools of power plants where we deal with integer variables.

Before taking up the main topic of this paper, we mention in the following several relevant research in the field of managing Virtual Power Plants.

In [22], the authors explore the management of virtual power plants and the incorporation of wind and solar parks into them. They propose a model involving separate operation of assets but a joint scheduling of them. Their model is based on Shapley value theory for profit distribution, and it is applied on case studies in China.

In [6], the authors provide ADMM-based dispatch techniques for virtual power plants which are also applied case studies developed in China. However, the authors do not take integer values into account.

In [17], the authors propose the algorithm of the implementation of ADMM within the blockchain while conducting the aggregation by means of a smart contract which can be used for any ADMM-based algorithm. The main point is that the aggregation relates to elementary operations with matrices and vectors, i.e. the operations that can be implemented within ADMM in a matter of milliseconds.

In [26] and [15], the authors provide the management of VPPs where gas-fired power plants balance a wind park. In the wind forecast methods they use spatial correlations. In [15], the authors propose a technique called Robust Model Predictive Control which we also employ in the management of the VPP.

According to [7] it is crucial to incorporate biogas power plants into the grid. To the best of our knowledge, research in the field of optimizing VPPs containing biogas power plants is limited, and in edgeFLEX project, we are going to fill this gap. In the framework of this project we will concentrate on balancing of wind parks, but it can easily be shown that the logic of the presented algorithms can be translated to broader classes of DERs [9].

1.1 Scope of this document

The present document is the report regarding the task 4 of work package 3: “Optimisation of a VPP consisting of variable and dispatchable RES”.

The aim of this task is to analyse the possibilities of balancing the intermittent power production based on variable RES (wind and solar), using dispatchable RES. The assets that we analyse are as follows: a battery, two biogas power plants and a wind park. I.e. the biogas power plants and the battery are aimed at balancing the wind park. The operational process is the following: the wind park operator indicates the electricity generation schedule that is foreseen to be produced, then the schedule becomes mandatory for all VPP members. The flexible (controllable) assets will make sure that the values indicated in the schedule will be realized.

The biogas power plants in our portfolio have neither classical timing constraints nor operation costs. Its only constraint is that the total number of switches of turbines per year should not exceed a predefined number. Apart from exploring this VPP (Table 2 and Figure 2), we also consider hypothetical pools of biogas power plants with hundreds of assets and with turbines having classical timing constraints in order to broaden the scope and the applicability of the proposed optimization methods.

Special attention is devoted to day-ahead energy price forecasts and the quality of the forecasts is measured by the improvements in the cumulative revenue.

Improvements in the forecasting performance of prices and in optimization methods are important to enable a better re-balancing of variable RES, and thereby promoting a further increase of RES in the total energy mix.

1.2 How to read this document

This report goes through optimization methods aimed at maximizing the total revenue and we focus on two basic features: precision in the presence of uncertainty and the robustness of optimization methods to the size of problems, i.e. the number of assets in the VPP. Table 5 and Figure 1 summarize the results related to the ability of the VPP to deal with new flexible assets in the pool. Figure 2 and Table 6 summarize the cumulative revenues yielded for different forecast methods (commercial, mSSA, N-BEATS, DeepAR) and control methods (MPC vs RMPC). We also provide the details of the features of the problem in the section Nuances of Optimization. Section 2 provides the description of the assets and the optimization problem. In Section 3, we provide the structural form of the optimization problem, which is amenable to asset-wise decomposition and splitting. In Section 3, we also describe our method called Gradual Increase. Section 4 is devoted to Model Predictive Control techniques. Section 5 describes our day-ahead price forecasts. Section 6 mentions the specific features of the optimization. Section 7 presents the results: Subsection 7.1 studies the robustness of the system to the increasing number of assets; Subsection 7.2 studies the precision yielded by different price forecasts; Subsection 7.3 mentions the positive social impact of the proposed methods. And then we move to the conclusion.

2. Problem Description

This chapter includes the description of the assets that will be subject for optimization, and the mathematical model of the optimization problem.

2.1 Description of the assets

edgeFLEX have a portfolio of two biogas power plants and a battery and their goal is to balance the production of wind parks located in 8 regions of Germany: the power plants and the battery are also located in Germany.

The process develops as follows: the operators of the wind assets forecast the amount of electricity produced by their assets. The electricity production based on wind is uncertain because it is weather dependent. Therefore, balancing the electricity generated by such assets is of utmost importance. The goal of the pool of biogas power plants and batteries is to handle the aggregate imbalance. The names of neither biogas power plants nor batteries nor wind parks can be disclosed for confidentiality reasons. We have detailed data for two biogas power plants and one battery.

The turbines within the biogas power plants have no classical timing constraints except for the condition that the maximum number of switches per year is limited to 1500. When exploring the speed of our algorithms, we impose timing constraints because they are typical for the turbines. We replicate the assets within the pool in order to explore how the increasing number of assets affects the speed of the calculations. When replicating assets, we predefine the number of assets in the pool and randomly assign the storage capacities, gas inflows, maximum and minimum productions of the turbines and minimum on and off times for the turbines within biogas power plants (timing constraints). In replicating, the wind park increases proportionally to the increase in flexible assets. This enables us to apply the proposed algorithms for a broader scope of asset types.

The field of operation is the German market EEX (day-ahead and intra-day) and we use the price forecasts as objective value coefficients. We model the imbalance by means of autoregressive models. In our biogas power plants the processes of fertilization and electricity production are separated. We are guaranteed to obtain fixed amounts F_k (see Table 1) of biogas (measured in MWh) every 15 minutes and we concentrate only on electricity production. In this study, we do not consider biogas power plants with heating rods, but we see this as a subject for further research.

2.2 The Notation

The formulation of the optimization models requires a specific notation which is provided in Table 1 and Table 2.

Table 1 Notation: constants

Symbol	Explanation
$\#A$	the number of assets
$\#T(k)$	the number of turbines within asset k
E_k^{max}	the maximum storage of asset k (MWh)
F_k	the flow of biogas power plant k every 15 minutes (MWh)
η_k^d	the efficiency of the discharge of battery k
η_k^c	the efficiency of the charge of battery k
$Pmin_{ki}$	the minimum power of Turbine i of Asset k (MW)

$Pmax_{ki}$	the maximum power of Turbine i of Asset k (MW)
$Frc_t^{(DA)}$	the day-ahead price at time t
$Frc_t^{(ID)}$	the intra-day price at time t
C_{ki}^w	the cost of switching-on of Turbine i of Asset k
C_{ki}^v	the cost of switching-off of Turbine i of Asset k at time t
$UT^{(k,i)}$	the minimum on time of Turbine i of Asset k at
$DT^{(k,i)}$	the minimum off time of Turbine i of Asset k at
$\forall t, k, i$	the shorthand for: for all $t \in \{1, \dots, T\}$ and all $k \in \{1, \dots, \#A\}$ and all $i \in \{1, \dots, \#T(k)\}$

Table 2 Notation: Variables

Symbol	Explanation
$p_{t,k,i}$	the total power produced at time t by Turbine i of Asset k
$p_{t,k,i}^{DA}$	the day-ahead power produced at time t by Turbine i of Asset k
$p_{t,k,i}^{ID}$	the intra-day power produced at time t by Turbine i of Asset k
$u_{t,k,i}$	the state of Turbine i of Asset k at time t
$v_{t,k,i}$	the switch-on decision of Turbine i of Asset k at time t
$w_{t,k,i}$	the switch-off decision of Turbine i of Asset k at time t
$soc_{t,k}^{Bg}$	the storage level of Asset (Power Plant) k at time t (MWh)
$soc_{t,k}^{Bt}$	the storage level of Asset (Battery) k at time t (MWh)
$p_{t,k,1}^{d}$	the discharge of battery k at time t
$p_{t,k,1}^{c}$	the charge of battery k at time t
x_k	the set of all variables p, u, v, w, soc, p reduced to asset k
y_t	the set of all variables p, u, v, w, soc, p reduced to time t
z^*	the optimal value of variables z. Any variable superscripted with a star denotes the value optimal for the objective function

2.3 The Objective

The objective is to maximize the revenues of pools of biogas power plants and batteries while balancing the wind park and taking technical constraints into account. The technical constraints of the turbines are their maximum and minimum production per period and the timing constraints.

We define the objective to be maximized as follows:

$$\sum_{t=1}^{T}\sum_{k=1}^{\#A}\sum_{i=1}^{\#T(k)}(Frc_t^{(DA)}p_{t,k,i}^{DA} + Frc_t^{(ID)}p_{t,k,i}^{ID} - C_{ki}^{v}v_{t,k,i} - C_{ki}^{w}w_{t,k,i}), \quad (1)$$

which implies that we optimize our revenue from the sale of energy on the market penalizing every switch-on and -off with costs C_{ki}^{v} and C_{ki}^{w}, where k denotes the index of an asset and i is the index of a turbine. In the case of a battery, these costs are zero.

The constant T denotes the prediction horizon that we use for the optimization and $Frc_t^{(DA)}$, $Frc_t^{(ID)}$ are the forecasts of day-ahead and intra-day prices at time t, respectively.

2.4 The Constraints

In this subsection, we describe all of the constraints associated with biogas power plants.

2.4.1 Binary Constraints

We employ 3bin formulation [16], and all u, v, and w variables (Table 2) are binary, i.e.

$$u_{\{t,k,i\}}, v_{\{t,k,i\}}, w_{\{t,k,i\}} \in \{0,1\} \quad \forall\, i,k,t. \tag{2}$$

2.4.2 Constraints for the biogas power plants

In this subsection, we write out all the constraints associated with biogas power plants.

2.4.2.1 Power Constraints

For every turbine we have two basic values: $Pmin_{ki} > 0$ and $Pmax_{ki} > Pmin_{ki}$ which implies that it can either do nothing or produce within the interval $[Pmin_{ki}, Pmiax_{ki}]$ i.e.

$$p_{t,k,i} \in \{0\} \cup [Pmin_{ki}, Pmax_{ki}] \quad \forall\, t,k,i \tag{3}$$

These constraints can be written using the $u_{t,k,i}$ variables which equal 1 if at time t the i-th turbine of Asset k is on and it is 0 otherwise:

$$p_{t,k,i} - Pmin_{ki}\, u_{t,k,i} \geq 0, \qquad Pmax_{ki}\, u_{t,k,i} - p_{t,k,i} \geq 0 \quad \forall\, t,k,i. \tag{4}$$

The power produced can be decomposed to two components: the power used for the Day-Ahead market and the power used for the Intra-Day market, i.e.

$$p_{t,k,i} = p_{t,k,i}^{ID} + p_{t,k,i}^{DA} \quad \forall t,k,i \tag{5}$$

2.4.2.2 Storage constraints

Every biogas power plant k has its own storage and a constant flow of gas F_k (expressed in MWh) into it. The new storage level is equal to the old level added by F_k and subtracted by the aggregate energy produced by all turbines within Asset k during one period:

$$soc_{t,k}^{Bg} = soc_{t-1,k}^{Bg} + F_k - \sum_{i=1}^{\#T(k)} p_{t,k,i} \cdot \Delta\, t \quad \forall\, t,k, \qquad \Delta\,\mathrm{t} = 15 \text{ minutes} \tag{6}$$

and we have the box constraints:

$$soc_{t,k}^{Bg} \in [0, E_k^{max}] \quad \forall t,k. \tag{7}$$

2.4.2.3 Timing constraints

The on and off decisions for the turbines are conducted via binary switch-on (v) and binary switch-off (w) variables as follows [16]:

$$u_{t,k,i} - u_{t-1,k,i} = v_{t,k,i} - w_{t,k,i} \quad \forall\, t,k,i. \tag{8}$$

If at time t a turbine i of asset k is on or off than it has to be on or off for at least $UT^{(k,i)}$ and $DT^{(k,i)}$ periods, respectively, which is expressed as follows:

$$\sum_{j=t-UT^{(k,i)}+1}^{t} v_{j,k,i} \leq u_{\{t,k,i\}} \text{ and } \sum_{t-DT^{(k,i)}+1}^{t} w_{j,k,i} \leq 1 - u_{t,k,i} \quad \forall\, t,k,i. \tag{9}$$

2.4.2.4 Timing constraints tuning

We have “light” requirements on our turbines from the biogas power plants that we explore. This term of “light” requirement identifies complementary constraints, in this case the maximum number of switch-ons per year. These conditions can be handled by imposing penalties for switch-ons and switch-offs. On the other hand there are, generally, turbines with strict timing constraints therefore in our exploration of the speed of the decomposition algorithms, we intentionally impose timing constraints in order to broaden the scope of applications of our algorithms. In any case, if it is possible to handle the timing constraints by penalizations C^v and C^w, we should do this because the constraints (9) account for most of inequality constraints of the problems and their removal can lead to significant reductions of the computation time. Even if it is not possible to handle the timing constraints with the penalization, we can observe that it is not likely that we would switch off when the price is high or switch on when the price is low except for technical requirements. So we formulate if-then conditions dependent of the average price of the prices in the objective value and on the average price in the preceding history in order to decide which constraints from (9) to remove. If these constraints are not tight, then they will not affect the objective value [4]. We always check whether the resulting solution is feasible for the whole problem and otherwise we solve the whole problem. We denote this method as (≤)-prune (See Table 5). In our further research, we will consider the methods of machine learning to replace these if-then conditions.

2.4.3 The constraints for batteries

In this subsection, we write out all the constraints associated with batteries.

2.4.3.1 Power constraints

For every battery we have the following power constraints:

$$\begin{aligned} Pmax_{k,1}\, u_{t,k,1} - p^d_{t,k,1} &\geq 0 \quad \forall\, t,k, \\ Pmax_{k,1}\,(1 - u_{t,k,1}) - p^c_{t,k,1} &\geq 0 \quad \forall\, t,k, \\ p^c_{t,k,1} - Pmin_{k,1}\, w_{t,k,1} &\geq 0 \quad \forall\, t,k, \\ p^d_{t,k,1} - Pmin_{k,1}\, v_{t,k,1} &\geq 0 \quad \forall\, t,k. \end{aligned} \tag{10}$$

Constraints (10) ensure that we cannot charge and discharge at the same time.

The update of the storage level and the state of charge is conducted as follows:

$$soc_{t+\Delta\, t,k}^{Bt} = soc_{t,k}^{Bt} + \Delta t \cdot \eta_k^c \cdot p_{t,k,1}^c - \frac{p_{\{t,k,1\}}^d \Delta\, t}{\eta_k^d} \quad \forall\, t, k, \tag{11}$$

where Δt is always 15 minutes.

2.4.3.2 Tightening constraints

$$v_{t,k,1} \leq u_{t,k,1} \text{ and } w_{t,k,1} \leq 1 - u_{t,k,1} \quad \forall\ t, k, \tag{12}$$

and for all t we have box constraints:

$$soc_{t,k}^{Bt} \in [0, E_k^{max}] \ \forall\, t, k. \tag{13}$$

2.4.3.3 Switches of the batteries

The on and off decisions for the batteries are conducted via binary switch-on (v) and binary switch-off (w) variables as follows:

$$u_{t,k,1} - u_{t-1,k,1} = v_{t,k,1} - w_{t,k,1} \quad \forall\, t, k. \tag{14}$$

2.4.3.4 Components of the objective function

We define the variables $p_{t,k,1}^{ID}$ as follows:

$$p_{t,k,1}^{ID} = p_{t,k,1}^d - p_{t,k,1}^c \quad \forall\, t, k, \tag{15}$$

which is included in the objective (1).

2.4.4 Coupling constraints

The coupling constraint is a task that all flexible assets have to fulfill. This is given by the imbalance IB_t $\forall\, t$ between the realized and forecasted production of the intermittent sources:

$$\sum_{k=1}^{\#A} \sum_{i=1}^{\#T(k)} p_{t,k,i}^{ID} = IB_t\,, \qquad \forall\, t. \tag{16}$$

Let us note that the right-hand side of (16) is the input that we have to estimate. This boils down to estimating the imbalance which is a challenging task. In this study we approach the imbalance by means of two methods:

- Taking the historical imbalance: perfect foresight. We will use it as a benchmark. This will be the upper bound to the problem.
- Fitting the imbalance with ARMA processes and using simulations.

2.4.5 Optimization problem formulation

Summarizing the aforementioned constraints and objectives, we can formulate the optimization problem with the forecast parameters: Frc^{DA}, Frc^{ID}, and $Disbalance$ as follows:

$$\text{maximize} \sum_{t=1}^{T} \sum_{k=1}^{\#A} \sum_{i=1}^{\#T(k)} (Frc_t^{(DA)} p_{t,k,i}^{DA} + Frc_t^{(ID)} p_{t,k,i}^{ID} - C_{ki}^v v_{t,k,i} - C_{ki}^w w_{t,k,i})$$

$$s.t. \qquad (2) - (16).$$

which we denote as follows: $VPP(Frc^{DA}, Frc^{ID}, Disbalance)$. For the problem $VPP(Frc^{DA}, Frc^{ID}, Disbalance)$, we define the additional notation:

- $VPP_x(Frc^{DA}, Frc^{ID}, IB)$ is the optimal solutions of the problem, i.e.: $VPP_x(Frc^{DA}, Frc^{ID}, IB) = (p^{DA*}, p^{ID*}, p^*, p^{c*}, p^{d*}, u^*, v^*, w^*, soc^*)$
- $VPP_r(Frc^{DA}, Frc^{ID}, IB)$ is the realized value of the problem, i.e. $VPP_r(Frc^{DA}, Frc^{ID}, IB) =$

$$= \sum_{t=1}^{T} \sum_{k=1}^{\#A} \sum_{i=1}^{\#T(k)} (Spt_{t,k,i}^{DA} p_{t,k,i}^{DA*} + Spt_t^{(ID)} p_{t,k,i}^{ID*} - C_{ki}^{v} v_{t,k,i}^{*} - C_{ki}^{w} w_{t,k,i}^{*})$$

- $VPP_{rv}(Frc^{DA}, Frc^{ID}, IB)$ is the revenue yielded from $VPP_x(Frc^{DA}, Frc^{ID}, IB)$ i.e.

$$VPP_{rv}(Frc^{DA}, Frc^{ID}, IB) = \sum_{t=1}^{T} \sum_{k=1}^{\#A} \sum_{i=1}^{\#T(k)} (Frc_t^{(DA)} p_{t,k,i}^{DA} + Frc_t^{(ID)} p_{t,k,i}^{ID})$$

is the realized revenue when the prices and the imbalance become known. When we replace the forecast of the imbalance IB with its realizationIB^R, then we check if the feasibility is preserved. If so we directly apply the sum $VPP_{rv}(Spt^{DA}, Spt^{ID}, IB)$ as the realized value. Otherwise we buy the missing energy on the market.

- $VPP_{rv}^{(h)}(Spt^{DA}, Spt^{ID}, IB^R)$is the realized revenue when the prices and the imbalance become known within the execution horizon h, i.e.

$$VPP_{rv}^{(h)}(Spt^{DA}, Spt^{ID}, IB^R) = \sum_{t=1}^{h} \sum_{k=1}^{\#A} \sum_{i=1}^{\#T(k)} (Spt_t^{(DA)} p_{t,k,i}^{DA} + Spt_t^{(ID)} p_{t,k,i}^{ID})$$

This value will be used in *Model Predictive Control*.

- $VPP_{rv}^{*}(Spt^{DA}, Spt^{ID}, IB^R)$ is the realized revenue from the solution of the perfect foresight problem $VPP(Spt^{DA}, Spt^{ID}, IB^R)$, where IB^R is the realized imbalance. The settings of Problem $VPP_{rv}^{*}(Spt^{DA}, Spt^{ID}, IB^R)$ correspond to the state of perfect foresight, i.e. when all the uncertain values are known up front. This value is used as the benchmark.
- The goal is to elaborate such a strategy that the ratio

$$\sigma = \frac{VPV_{rv}(Spt^{DA}, Spt^{ID}, IB^R)}{VPP_{rv}^{*}(Spt^{DA}, Spt^{ID}, IB^R)}$$

is as close to 1 as possible which can be increased by better optimizations and better forecasts.

We also introduce the optimization problem $VPP(Frc^{DA}, Frc^{ID}, IB, State)$ which differs from $VPP(Frc^{DA}, Frc^{ID}, IB)$by specifying starting conditions of turbines and storage levels constrained in the variable $State$.

3. A Structural Asset-Based Problem Formulation

The problem $VPP(Frc^{DA}, Frc^{ID}, IB)$ has a linear objective and all linear constraints, therefore it can easily be shown that it can be rewritten in the following form:

$$\text{maximize} \sum_{k=1}^{\#A} p_k^T \cdot x_k \tag{17}$$

$$\text{s.t} \quad \sum_{k=1}^{\#A} A_k^T \cdot x_k = a, \tag{18}$$

$$B_k \cdot x_k \leq b_k \tag{19}$$

$$x_k(I) \in \{0, 1\} \tag{20}$$

$$\forall\, k \in \{1,2,3, \dots, \#A\} \tag{21}$$

where x_k represents all variables associated with Asset k, in other words x_k constrains all variables from Table 2, whose Asset's index is k. The vector $p_k$consists of all objective value coefficients from Expression (1) associated with Asset k and Expression (17) is equivalent to Expression (1). In an analogous manner Equation (18) is equivalent to Equation (16) and Expression (19) is equivalent to Expressions (3)-(15). And the Expression (20) is equivalent to (2).

3.1 Description of gradual increase

The main idea behind the method of Gradual Increase lies in the proper use of the warm start. When we have to handle a task a from (18), we can check whether it is possible to implement it with a smaller number of assets (or turbines within the assets). If it is possible, then the resulting solution can be used as a start in either a larger or the entire pool. We try to start with a minimum sub-pool capable of implementing the task and add assets to the pool with its consequent optimization, until the entire pool is achieved. When implementing this algorithm, we make sure that the branch & bound trees will not be destroyed. We achieve this by the parallel run of the problem containing all assets which is interrupted whenever a new feasible solution is found: we feed the problem with a new resume point and resume optimization. Some problems are so complex that it may happen that even the best solver will not be able to find a feasible solution to it. However, Gradual Increase enables us to find a feasible solution for the entire pool from the solution of a subproblem. Formally, the method is as follows: let us assume that $SubSet(0) \subset \{1,2, \dots, \#A\}$, where $SubSet(0)$ is a first sub-pool of the whole pool. Then the first problem, can be formulated as follows:

$$\text{maximize} \sum_{k \in SubSet(0)} p_k^T \cdot x_k \tag{22}$$

$$\text{s.t.} \quad \sum_{k \in SubSet(0)} A_k^T \cdot x_k = b, \tag{23}$$

$$B_k \cdot x_k \leq b_k, \tag{24}$$

$$x_k(I) \in \{0, 1\}, \tag{25}$$

$$\forall\, k \in SubSet(0). \tag{26}$$

As a warm start, for the first problem, we can choose the states of turbines (u) from the solution of the problems in the previous period. Since $\#SubSet(0) < \#A$, the Problem (22)-(26) contains fewer variables and constraints than the initial problem and should be solved faster except for specific cases, e.g. when the pool is so small that it is overloaded. Having solved this problem, we get a sequence of vectors z_k, $k \in SubSet(0)$. Then, when we get a larger sub-pool $SubSet(1)$ such that:

$$SubSet(0) \subset SubSet(1) \text{ and } SubSet(0) \neq SubSet(1).$$

Hence, we will be able to formulate the following problem for $j = 0$:

$$\begin{aligned} \text{maximize} \quad & \sum_{k \in SubSet(j+1)} p_k^T \cdot x_k \\ \text{s.t.} \quad & \sum_{k \in SubSet(j+1)} A_k^T \cdot x_k = a, \\ & B_k \cdot x_k \leq b_k, \quad x_k(I) \in \{0,1\}, \\ & \forall k \in SubSet(j+1), \\ & x_m.start = z_m \quad \forall m \in SubSet(j), \\ & x_m.start = g_m \quad \forall m \notin SubSet(j), \end{aligned} \tag{27}$$

where g_m is the optimal solution (argmin) of the following problem:

$$\begin{aligned} & \text{maximize } p_m^T \cdot x_m \\ & \text{s.t.} \quad A_m^T \cdot x_m = \mathbf{0}, \\ & x_m(I) \in \{0,1\}, \\ & B_m \cdot x_m \leq b_m, \end{aligned} \tag{28}$$

Thus, the solution γ_k, $k \in \{1,2,\dots,\#A\}$ defined as follows:

$$\gamma_k = \begin{pmatrix} z_k, & k \in SubSet(j+1) \\ g_k, & k \notin SubSet(j+1) \end{pmatrix}$$

is a feasible solution of Problem (22)-(26), i.e. suitable for the *warm start*, because:

$$\sum_{k=1}^{\#A} A_k^T \cdot \gamma_k = \sum_{k \in SubSet(j+1)} A_k^T \cdot \gamma_k + \sum_{k \notin SubSet(j+1)} A_k^T \cdot \gamma_k =$$

$$\sum_{k \in SubSet(j+1)} A_k^T \cdot z_k + \sum_{k \notin SubSet(j+1)} A_k^T \cdot z_k = a.$$

The rest (related to the feasibility of γ) is evident.

These *warm starts* have enabled us to solve a large number of problems much faster than when we would just rely on the power of an open-source or commercial *solver*. When we optimize the same pool then we can take the experience from previously solved problems in order to accelerate the solution of new problems as follows:

1) Exploration of which sub-pools would lead to a faster solution on previously solved problems and applying these sub-pools for new problems.

2) Exploration of the warm start: we can check if the binary variables from the previous problem can be used in the warm start: if it is possible then the first feasible solution will be obtained by means of LP otherwise the *solver* will start from scratch.
3) Exploration of what *solver* parameters from previously solved problems would lead to faster solutions of these problems and application of these parameters on the new problems.

We will compare the results obtained by the Proximal Jacobian ADMM and Gradual Increase and we will combine both of the methods.

4. Operation of the VPP based on Historical Data

We consider the VPP to be composed of predefined assets, and we apply historical prices and production together with their corresponding historical forecasts. The wind production forecast is the task of the wind park: based on the prognosis the operators of the park inform the market operator that they are able to produce a specific schedule. The imbalance between the produced and forecasted wind energy is the task of the flexible assets: the surplus of the will be consumed by the batteries while the deficiency will be covered either by the biogas power plants or by discharging the batteries. This implementation will be conducted in the Model Predictive Control fashion where we solve the problem every 15 minutes when new information about prices and the weather arrives.

4.1 Model predictive control (MPC)

Model Predictive Control is a feedback control technique that naturally incorporates optimization [15], [3], and [14]. In this study we consider certainty equivalent MPC and robust MPC proposed in [15]. In certainty equivalent MPC, we replace random quantities with predictions, and solve the associated optimization problem to produce the schedule over the selected planning horizon. After optimization, we execute the first power schedule, i.e. the one associated with the time of optimization. For the next step, we repeat this process incorporating the updated information about price and imbalance forecasts. Following the notation from Table 4, we define the MPC algorithm as follows:

Take the initial state $State_0$ of the system (storage levels and states of the turbines) as the first input.

for $t = 1$ **to** N **do**:

- ***Forecast.*** Make price and imbalance forecasts that will be used as inputs in the optimization:

$$\boldsymbol{Frc_t} = [Frc_t, Frc_{t+1}, \ldots, Frc_{t+H-1}],$$

$$\Delta\, prod_t^W = [\Delta\, prod_t^W, \Delta\, prod_{t+1}^W, \ldots, \Delta\, prod_{t+H-1}^W].$$

- ***Optimize.*** Solve the dynamic optimization problem:

$$VPP(Frc_t^{DA}, Frc_t^{ID}, \Delta\, prod_t^W, State_{t-1}), \tag{29}$$

where $\boldsymbol{Frc_t}$ determines the objective value and $prod_t^W$ determines the right-hand side of the power production constraints. Solving this optimization problem yields the decision vector $\boldsymbol{y_t} = [y_t, y_{t+1}, \ldots, y_{t+H-1}]$.

- ***Execute.*** We execute only y_t from the whole vector $\boldsymbol{y_t}$ because this decision relates to the most up-to-date time step and discard the rest of the components of $\boldsymbol{y_t}$.
- Determine the value $Revenue_t$ which is the revenue associated with the execution of y_t which equals:

$$Revenue_t = VPP_{rv}^h(Spt_t^{DA}, Spt_t^{ID}, \nabla\, prod_t^W)$$

and the next state is: $State_t = f(State_{t-1}, y_t, \nabla\, prod_t^W)$, where f is a linear function which determines the updates of the storage levels and states of the turbines/batteries according to (3)-(11).

end for

Thus, the ultimate goal is the maximization of the sum:

$$TotRev = \sum_{t=1}^{N} Revenue_t \rightarrow \max,$$

i.e. the solution of the problems $VPP(\cdot,\cdot,\cdot)$ is an intermediate goal aimed at maximizing the value $TotRev$. And in this study we assess the model in terms of the value of $TotRev$ and the total speed-up. All the methods only differ by the approach to problems $VPP(\cdot,\cdot,\cdot)$ and we are going to apply the following:

- **Generic Approach**: we rely on the power of the *solver* without any decomposition or splitting.
- **Gradual Increase**: we perform asset-wise decomposition as it is shown in the description of the method and in a similar way we apply this approach on the turbines within each biogas power plants.
- **Proximal Jacobian ADMM**: we start with LP relaxation in order to ensure that the coupling constraint is satisfied and then we apply MILP methods to sub-pools (most on single power plants) in order to extract an integral solution.
- **Partial Integrality**: at time t, we relax all integral constraints of all variables $y_{t+1}, y_{t+2}, \dots, y_{t+H-1}$. This leads to a significant reduction of binary variables while providing a feasible action y_t . (Variable y_t is defined in Table 2.)
- **Management of Timing Constraints by Penalization**: in many situations, we can enforce the timing constraints by imposing high penalties on switch-ons and switch-offs. This enables us to get rid of most of the inequality constraints and thereby to significantly accelerate the calculations.
- **Parameter tuning with MPC**: within MPC, we can run parameter tuning of the *solver* after we have solved a problem. It can be conducted parallel to the solution of the new problems. After solving 20 problems, we change the tuning parameters which provides further speed-up. In Python's Gurobi environment, this can be conducted by means of the operation **model.tune( )**.
- **Hybrid of GI and Proximal Jacobian ADMM**: instead of adding single power plants in the pool we add them by blocks and in order to solve block subproblems faster, we apply Gradual Increase.

4.2 Robust Model predictive control (RMPC)

The difference between MPC and RMPC lies in a different approach to the second step of the algorithm (*Optimize*). In (29), there is a single forecast of prices and imbalances. In RMPC however [15], we use a predefined number of scenarios M, i.e.

$$Scenario_m = (Frc_t^{DA,m}, Frc_t^{ID,m}, \Delta\, prod_t^{W,m} \quad \text{for } m = 1,2,\dots,M,$$

and use each scenario m in order to solve

$$\begin{aligned} &\text{maximize} \sum_{m=1}^{M} VPP(Frc_t^{DA,m}, Frc_t^{ID,m}, \quad \Delta\, prod_t^{W,m},\ State_{t-1}) \\ &\text{s.t.} \\ &p_{\tau,k,i}^{ID,m^*} - p_{\tau,k,i}^{ID,m^{**}} = 0 \quad \forall\, k,i \text{ and } \forall\, m^*, m^{**} \le M \text{ and } \forall\, \tau \le h, \\ &p_{\tau,k,i}^{DA,m^*} - p_{\tau,k,i}^{DA,m^{**}} = 0 \quad \forall\, k,i \text{ and } \forall\, m^*, m^{**} \le M \text{ and } \forall\, \tau \le h, \end{aligned} \tag{30}$$

where the sum in (30) means that we add up all the objectives from the problems $VPP(Frc_t^{DA,m}, Frc_t^{ID,m},\ \Delta\, prod_t^{W,m},\ State_{t-1})$ for each m and maximize their sum; as for the constraints, they all are added into the set of the constraints of the resulting problem. The equality constraints in (30) ensure that the power produced within each scenario $Scenario_m$ will be the same for all the scenarios up to time h. This also ensures that the execution in the the MPC phase, i.e.

$$Revenue_t = VPP_{rv}^h(Spt_t^{DA}, Spt_t^{ID}, \nabla\, prod_t^W),$$

will be the same for all the scenarios. The rest of the steps of the MPC algorithm remain unchanged. It can easily be shown that the same power values for all scenarios also imply the same states of the turbines and the same storage levels for all the scenarios when $t < h$.

5. Wind power output forecast

5.1 Introduction

The present chapter describes the solution proposed in edgeFLEX for developing a novel spatio temporal wind forecasting method (WP 3, Task 3.5 in the GA).

The goal of this task is to develop an innovative wind forecasting model with improved performances compared to traditional site-specific models. in time spans covering 1-6 hours before delivery. A better forecasting performance, in this timespan, is important for enabling a better re-balancing of intermittent electricity generation, and thereby promoting a further increase of sun and wind energies in the overall mix.

This chapter will go through the details about the data sources, what are they and what transformation have been made in order to create a dataset. Secondly, the next part will focus on Machine Learning model selection and a brief mathematical explanation of how it works. Then, we will dive into the results and the comparison to another top of the basket model. And finally, the next steps and way of improvements will be discussed just before a conclusion.

5.2 Data sources

In order to create models that may be able to forecasts wind power we need to have some wind power data and we need to identify what can have an influence on wind power production in the near future.

5.2.1 Production data

Alpiq handles a portfolio of wind power farms across Germany and for this project we have considered using 9 different assets that are well dispatched in Germany. We decided to select only 9 assets as we needed to have a long enough history of production to use. Some are offshore and others are well in lands. The graph below shows roughly where the different assets are located, and the size of the dot represents the maximum power output. This data comes with a granularity of 15 minutes which is what we need. Unfortunately, the data is available only at the asset level and not at the turbine level. There are only few missing values.

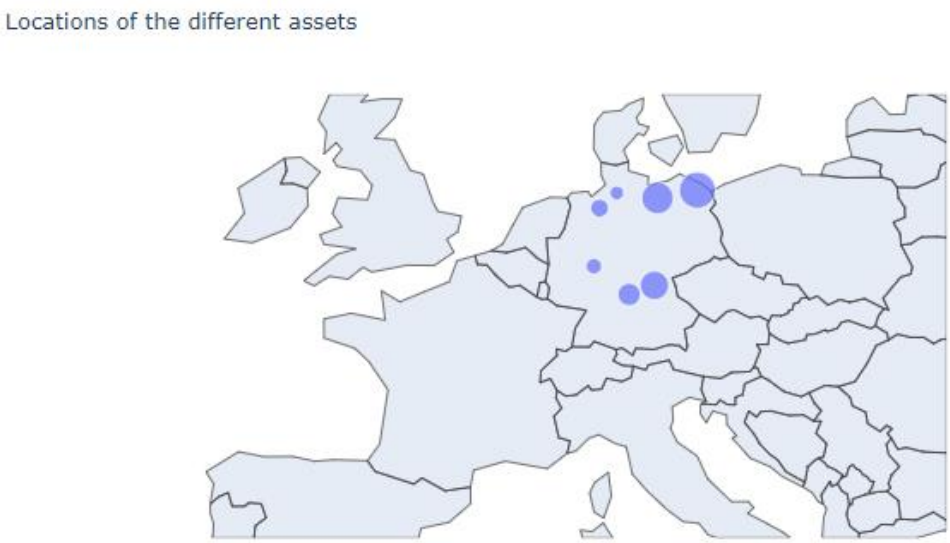

Figure 1 Locations of the studied assets

5.2.2 Weather data

The wind power output from wind farms is directly correlated to the current wind speed. For this reason, the weather data have been added to the production data.. The weather data have been divided into two categories: observation data and weather forecast data.. The observation data includes the current observed value of some weather features. The aim of this data is to provide the most up-to-date information that is 100% correct. The weather forecast data is used to support the forecasts of the weather features values. The more consistent and detailed information we provide to the model, the better are the results.

5.2.2.1 Weather observations data

These data are collected from the Deutscher Wetterdienst (DWD) which is the German meteorological service. It monitors weather and meteorological conditions all over Germany with a lot of weather stations. Those data are made available freely through the DWD portal. Unfortunately, only 419 stations measure the wind speed and direction and thus the distance between the closest weather station and a wind farm is sometimes more than 50km. As one will see in the latter part of this report, the distance is of great importance and it has been decided not to focus on the assets 2, 5 and 7 because the distance to the closest station was too big. So from now on, it will be refer to the assets 1,3,4,6,8,9 as A,B,C,D,E,F.

It has been decided to collect not all but a selection of the meteorological features that were available:

- Wind speed, wind direction and wind gust [Speed is the average, gust is the maximal value]
- Temperature at 5 meters and at 200 meters
- Pressure

For every asset, the closest station that measures all the weather features listed above has been selected. The data is collected with a granularity of 10 minutes and is then resampled to match the 15 minutes granularity of the production data. Then those 6 meteorological features are concatenated to the productions data. This approach is considered reasonably acceptable in the lack of better alternative given the flat geography of Northern Germany.

5.2.2.2 Weather forecasts data

In order to help the model forecasting better, it has been decided to provide it weather forecasts that are freely available and reliable. After several research, the European Centre for Medium-Range Weather Forecasts (ECMWF) was identified as the only one reliable, ease of access and free sources of gridded weather forecast. The International Grand Global Ensemble (TIGGE) consists of ensemble forecast data from 10 global Numerical Weather Prediction (NWP) centers. These data are available through the ECMWF portal and it was decided to use them. Most of the NWP center are run only twice a day, only 3 of them are run 4 times a day. For simplicity and to make the data more understandable only one NWP center data was taken: NCEP. The data is composed of 4 weather features which are:

- Wind speed and wind direction
- Temperature at 2 meters
- Pressure

This data is gridded which means that it is available worldwide on a grid that with a certain latitude and longitude difference between every point. An option that allows to reduce the distance between every point is available and thus a grid of 0.15/0.15 was selected. That means a point every 0.15 latitude and longitude. It was not needed to retrieve the entire grid as the assets are only located in Germany, so a grid just a bit bigger than Germany was selected.

For every asset it has been decided to select only the points of the grid that were within a given radius. This radius has been set to 20km for issues of computation time during the training phase of the machine learning model. This add approximatively 200 columns to the dataset (4 x nb of points within radius).

The figure below illustrates the grid (blue points), the assets (red points) and the radius of selection (green circle). On this figure, for the sake of visibility the radius was set to 50 km.

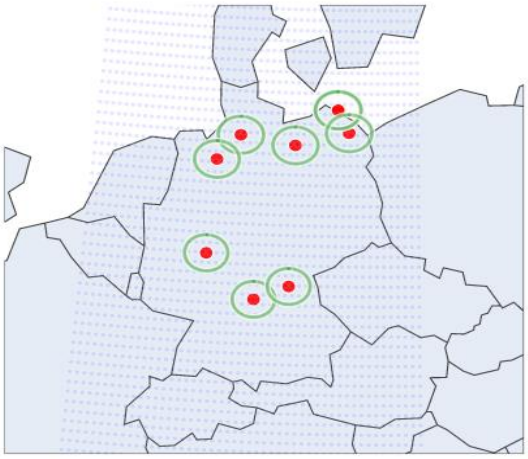

Figure 2 Weather forecasts grid and selection of points around assets

Another important point is about the granularity. In ideal conditions, it is desired to have weather forecasts of 24 hours ahead with a granularity of 15 minutes and a model producing such data every 15 minutes. However here the NWP models are run only 4 times a day (0h, 6h, 12h, 18h) and with a granularity of 6 hours. So, we had to interpolate the data that we have with the NWP models in order to obtain what we want but a significant point was not to use data that has been generated in the future.

To do so, the forecasts were taken at time 0h, this represents 5 points for forecast at value +0, +6, +12, +18, +24. An interpolation was generated (linear or quadratic) to have values +0, +0.25, …, + 30. Then a sliding window of 24h was used that was slided every 15 minutes to have the values in a format that we want. The graph below illustrates this with greater details.

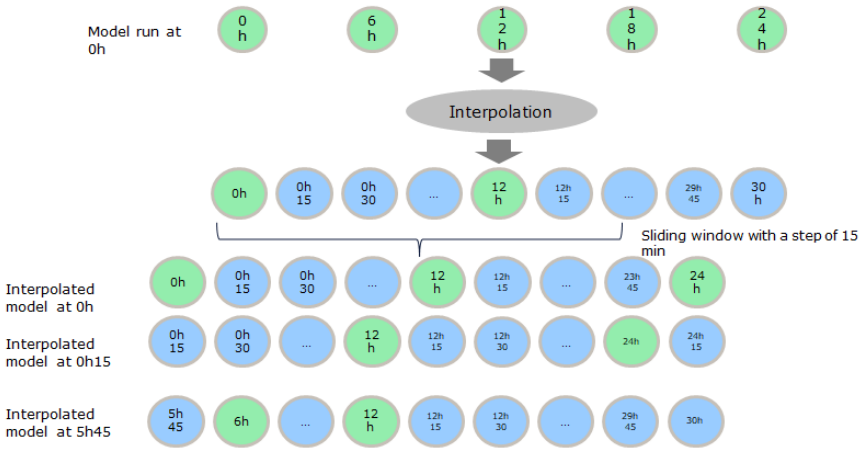

Figure 3 Detailed description of the interpolation method to reduce the weather forecast granularity

5.2.3 Seasonality data

Wind is often stronger during the winter with the different storms, so it was decided to look deeper into the production data across seasons. In order to see a general trend, the monthly average production data of every asset has been printed on the y axis and the time on the x axis. The figure below shows the result.

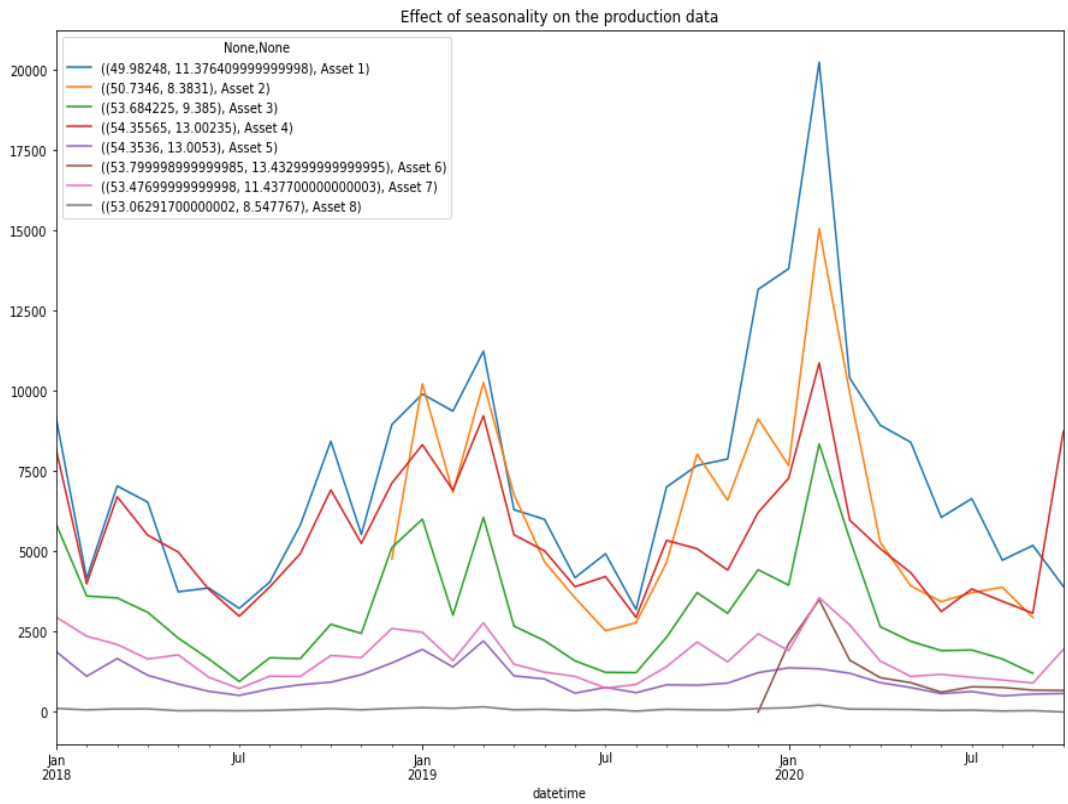

Figure 4 Illustration of the effect of seasonality on the production

One can easily see that around January each year the production is at its highest. Hence, it has been decided to introduce columns that can capture the month of the year each row is in. Those columns are called dummies because there are 12 columns, one for each month, and for each row there is a 1 in the month corresponding to the date and 0 in the others.

5.3 Machine Learning Model

5.3.1 Description of the problem

The problem here is a typical forecast problem using machine learning. Machine learning is a tool that requires a fixed (i.e. stable) environment and fixed conditions to work at its best, especially when working with time series. Due to computation limits, it is impossible to give the entire history available as input. So, it is important to define a fixed size context that will be used before every prediction and a fixed size label that will be the ground truth for every prediction.

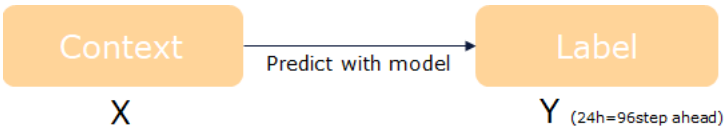

Figure 5 High level architecture of a machine learning problem

5.3.2 Auto-regression versus Sequence-to-Sequence approach

The data is represented as a time series i.e. a continuum of rows, one every 15min for more than a year. To be able to train a machine learning model we must transform this time series format into a digestible format for a machine learning model. Such a format has been briefly described into the previous section and we will now dig deeper into the mechanism of transformation of the data.

To begin with, it was needed to establish what will be the context also known as X and what will be the label also known as Y. In an ideal world, X would be all data before the moment of prediction but in order to optimize the computation time, it has been decided to take only a window of 24h past data as X.

What is desired is to forecast the next 6 hours of wind power output. To do so, there are two main methods which will be described below. The method selected will indicate how the data should be transformed to train a model.

5.3.2.1 Auto-regressive method

The idea is to take the input X as mentioned above and predict only the next value of wind power, we then adjust the window of input and take the predicted value as input to predict another value. And this process is repeated until the model has predicted all the data that we want as you can see on the *Figure 6 Autoregressive method to predict multi-step ahead* . There are some positive and negatives to this method:

- Positive: Easy to predict output that have different length, model easier to train because the label has only a size of 1
- Negative: Errors accumulation, so small error on the model leads to huge error for long output

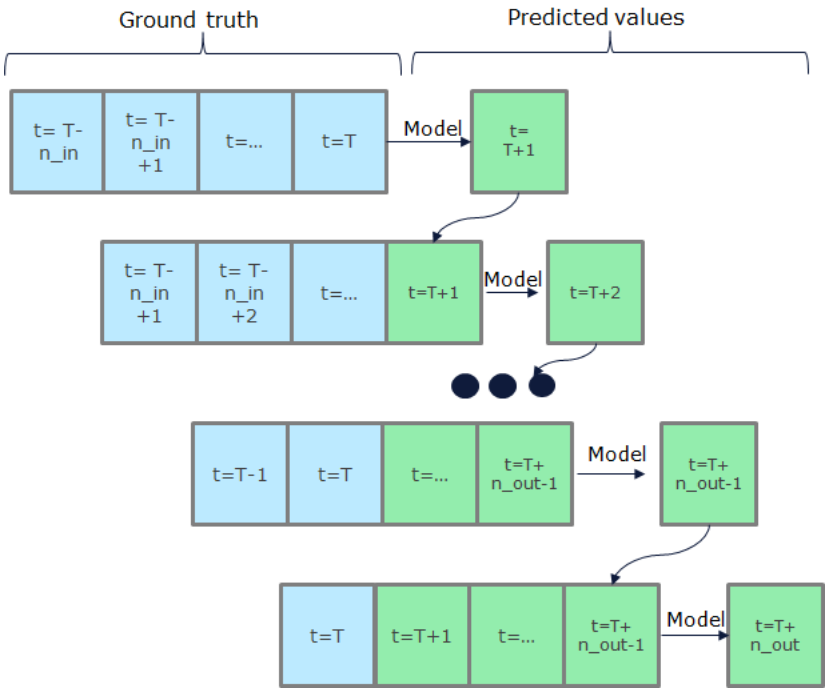

Figure 6 Autoregressive method to predict multi-step ahead forecast

The negative part of this method represents a too big counterpart that is why this is not the method that was decided to go further with.

5.3.2.2 Seq2Seq

The second approach is in theory much simpler to understand. Seq2Seq stands for Sequence to Sequence and it means that the model takes a sequence as input and output a sequence. It has firstly been introduced in 2014 with the paper by (Sutskever, Vinyals and Le).

This means that a window can be given of for instance 24 hours of data to the model and it will compute the entire 24h ahead wind power. Those kinds of models are more difficult to handle than the previous ones as the error is computed on the entire output sequence. To be able to achieve good results, the data must contain enough information to be able to encapsulate the general trend of the data.

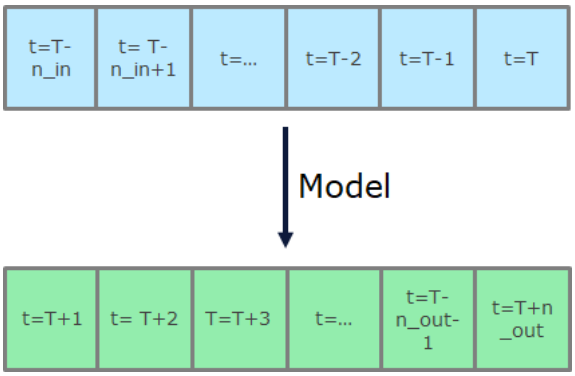

Figure 7 Seq2Seq method for multi-step ahead forecast

This second type of model is the one selected in this case, and the next section will develop the actual detail of the model that has been used along with a bit of theory to understand better how it works.

5.3.3 Model selected – LSTM encoder decoder

5.3.3.1 Encoder Decoder

The Encoder Decoder is the way Sequence to Sequence model works. In fact, the input sequence goes into the encoder that convert the whole sequence into a context vector. Then this context vector is given as input to the decoder which generates a sequence as output. Each block of the encoder and of the decoder are Recurrent Neural Network (RNN) cells and in the next section it

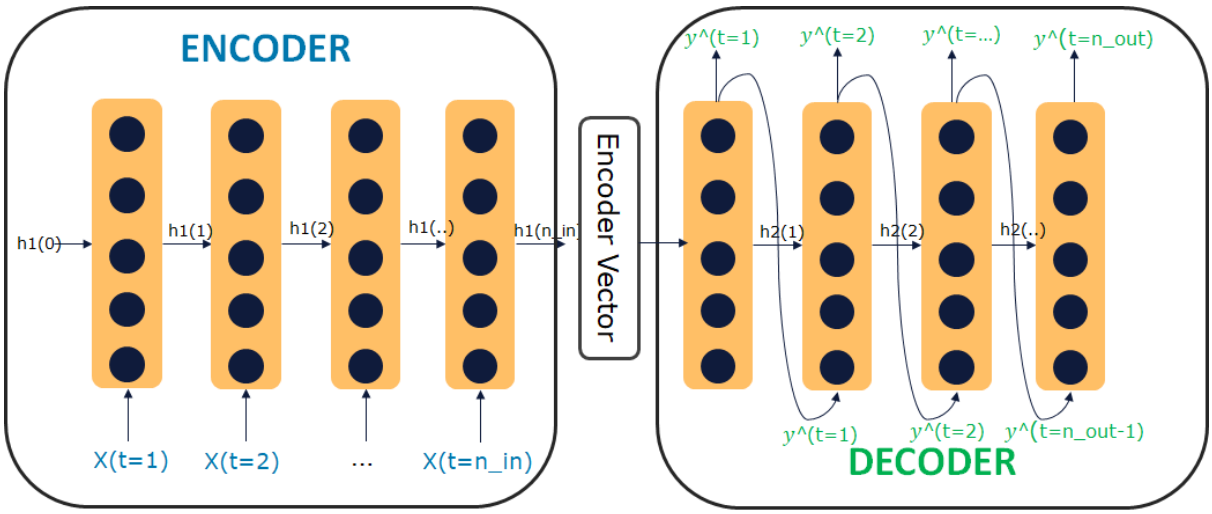

Figure 8 Encoder Decoder architecture

will be discussed further about which RNN cells it is decided to go with and why. The figure below depicts the general idea of the encoder decoder.

5.3.3.1.1 Encoder

The Encoder transforms an input sequence into a fixed-shape Encoder Vector (it will be named C for context). It is composed of RNN cells and the ideas lies under the propagation of the hidden vector h. The hidden states h_i are computed using the formula:

$$\boldsymbol{h1_t} = \boldsymbol{f}(\boldsymbol{h1_{t-1}}, \boldsymbol{X_t}) \tag{1}$$

The matrix W is the matrix of weight. And the Encoder Vector is then often computed as follow:

$$c = q\left(h_1, \dots, h_{n_{in}}\right) \tag{2}$$

Usually the function q is just that the last vector h computed is taken. All the outputs from the encoder are discarded as they are useless for the task desired.

5.3.3.1.2 Decoder

The decoder takes as initial vector for its hidden vectors the Encoder Vector that contains the information of the input sequence. Then it generates one by one the output sequence by taking its hidden vector and the output from the previous step:

$$\begin{gathered} \boldsymbol{h2_0} = \boldsymbol{c} \\ \boldsymbol{h2_t} = \boldsymbol{g}(\boldsymbol{\hat{y}_{t-1}}, \boldsymbol{h2_{t-1}}) \\ \boldsymbol{\hat{y}_t} = \boldsymbol{softmax}\,(\boldsymbol{W}\ \boldsymbol{h2_t}) \end{gathered} \tag{3}$$

5.3.3.2 LSTM Cell

Before introducing the LSTM cell, it needs to be understood why the basic RNN cells are not suitable for the task. First, the function f and g mentioned earlier are called activation function and for the RNN cell they usually are *tanh* function:

$$\mathrm{h_t} = \tanh(\mathrm{W^T} * \mathrm{concat}(\mathrm{X_t}, \mathrm{h_{t-1}}) + \mathrm{b}) \tag{4}$$

RNN cells are good as they allow to transfer information from the past in order to make the output better but there are two main problems:

- **Long term dependency**: the information is not carried long time as the network treat all new information the same and thus values from a long time ago will have less impact than the closest one.
- **Vanishing gradient**: Because when you train such a network you want to take the derivative of the gradient in order to backpropagate the error made and thus update the weights. As the derivative of the *tanh* function is in [0,1] then the gradient quickly come very close to zero.

To avoid these problems, **(Hochreiter and Schmidhuber)** have introduced Long Short-Term Memory (LSTM) cells have been introduced. In a few words, instead of just using a $\boldsymbol{tanh}$ function as activation to compute the hidden vectors LSTM cells introduce some notions that will allow to keep in memory important values and to forget some thanks to three gates: Input gate, forget gate and output gate. In big words, the input gate will tell what new information is going to be stored into the cell state, then the forget gate will indicate the information to throw away and finally the output gate provide the activation to the final output of the cell.

$$\begin{aligned} \mathrm{i_t} &= \sigma(\mathrm{w_i}[\mathrm{h_{t-1}}, \mathrm{x_t}] + \mathrm{b_i}) \\ \mathrm{f_t} &= \sigma(\mathrm{w_f}[\mathrm{h_{t-1}}, \mathrm{x_t}] + \mathrm{b_f}) \\ \mathrm{o_t} &= \sigma(\mathrm{w_o}[\mathrm{h_{t-1}}, \mathrm{x_t}] + \mathrm{b_o}) \end{aligned} \tag{5}$$

Then the combination of those three information follows the equations:

$$\begin{aligned} \tilde{c}_t &= \tanh(w_c[h_{t-1}, x_t] + b_c) \\ c_t &= f_t * c_{t-1} + i_t * \tilde{c}_t \\ h_t &= o_t * \tanh(c_t) \end{aligned} \tag{6}$$

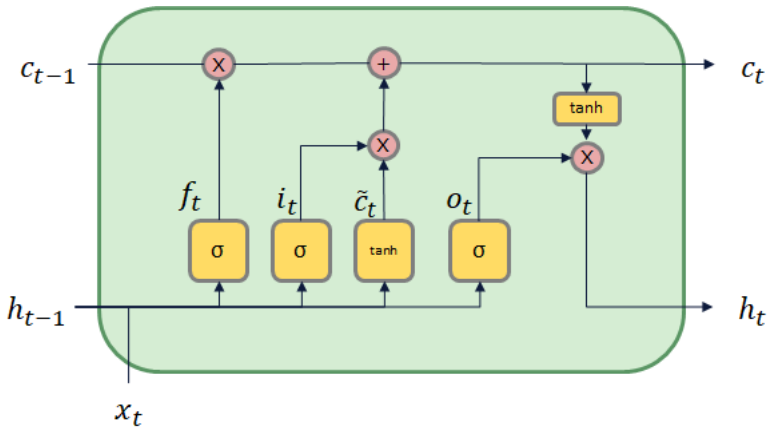

Figure 9 Visual representation of a LSTM cell

By using this cell into the Encoder Decoder model that have been described earlier it will be possible to take a quite long sequence as input and then output another sequence possibly of different size. The LSTM will make sure that only the important part of the input sequence is stored into the hidden vector and thus the decoder will be able to generate a sequence with a better understanding of the input sequence.

5.3.4 Technology used

Different technologies have been used for different tasks alongside the project. The different part and technologies are listed below:

- Preparation of the data: python (pandas, numpy)
- Encoder decoder model: Tensorflow (Keras)
- Work environment: Jupyter notebook

5.4 Results

As stated in the proposal of the task 3.5 of the edgeFLEX project, the results should be compared to site-specific models. After several tries, it was not possible to access such models, so no baseline was available to compare the results with. It has been decided to look deeper at what could be a benchmark in wind power forecasts and found out that Energy Meteo is a company that provides wind power forecasts for specific sites. The Energy Meteo forecasts are widely used by German energy traders because they are one of the best available in the market. It was

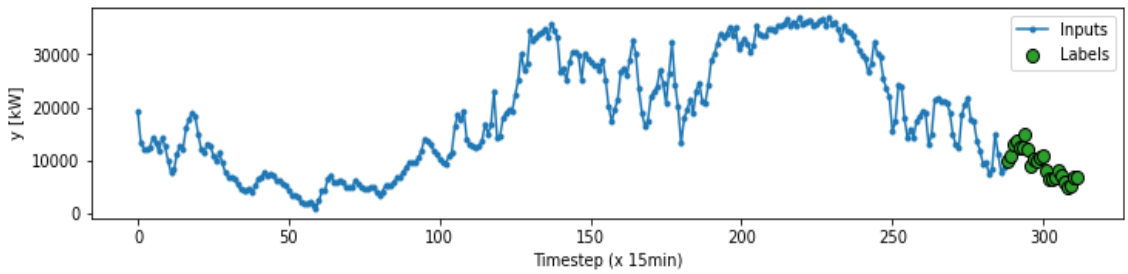

Figure 10 Visualisation of the input production data and the ground truth for the ass asset A

decided to use those forecasts as benchmark to evaluate the models with the market. The objective as stated in the introduction is to forecast with a horizon of 1 to 6 hours. To cover, it was decided to go with a forecast horizon of 6 hours with a granularity of 15 minutes. To avoid printing all the figure for every asset it will only shown for the asset A. All the curves would have looked the same for every asset.

The input length is a parameter of the model as one could take only few hours of input to as big as one month. For the sake of simplicity, it was decided to go with 3 days as the increase of the input length slows the training phase and does not bring significant improvements. On the figure below, one can see a part of the input data and the ground truth that are predicted. The blue part is only the past production data which represents only one feature out of approximatively 200.

The first results can be seen below for the asset A again. This is not the same curve as they are generated by taking randomly an input in the test set and the figure above was from the train set. The orange points correspond to the forecast of the model and the red points to the forecast of the energy meteo model.

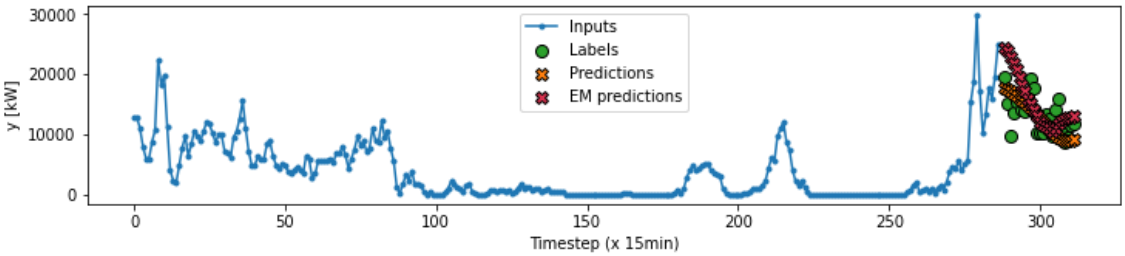

Figure 11 Visual comparison of the model and Energy Meteo forecast for asset A

Even if this representation allows us to visually see the difference between ground truth, our forecasts and Energy Meteo forecasts, it can only be seen one forecast at a time, so it does not allow to compare well. Hence, it was decided to print the mean difference between EM forecasts and ground truth for every time step, along with the mean difference between the forecasts and ground truth for each time step.

To interpret well this graph, one should know that the graph is about the asset A that have a

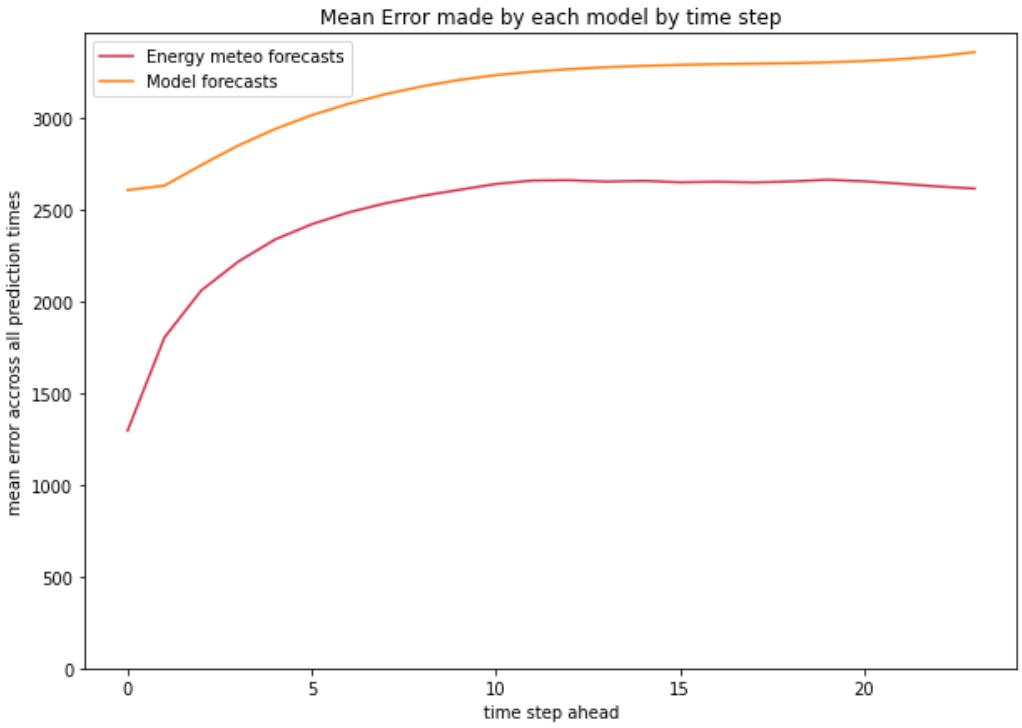

Figure 12 - Mean error made by each model for every time step in the 6 hours horizon for asset A

maximum output capacity of 38625 kW. The mean error across all timestep is about 3146.6 kW for the model and 2476.7 kW for Energy Meteo models.

This graph for the different assets has always similar shape so it would be redundant to print all of them. So below is a table which summarize the results for the different assets.

Table 3 Mean error of the model and EM model on all assets

Assets	Max capacity (kW)	Model mean error (kW)	EM model mean error (kW)	Relative difference
Asset A	38625	3146.6	2476.7	0.0173
Asset B	14900	1228.6	976.6	0.0169
Asset C	29500	1803.9	1517.6	0.0097
Asset D	9900	353.9	288.0	0.0066
Asset E	500	36.82	28.70	0.0162
Asset F	1200	90.77	72.36	0.0153

As the table above show, our model is less efficient than the Energy Meteo models for all the assets. However, the delta error between the two models is small and can be explained by several reasons which is being explored in the later part of this report.

5.5 Sensitivity analysis

5.5.1 Effect of input sequence length

The size of the input sequence is a matter of great importance as it is one of the biggest parameters. Below will be described the different outputs of our model that are obtained when taking different length for the input sequence. This analysis will be done again for the asset A to illustrate this but the results are similar for every asset. All the other parameters will be kept constant. The idea is to see how important the input length is.

Table 4 Effect of the input sequence length on error and training time

Size of input sequence	Error by the model (kW)	Time per training epoch
6 hours	3223.46	24s
1 day	3224.92	27s
3 days	3146.6	33s
7 days	3249.09	45s
14 days	3227.84	59s
1 month	3325.00	88s

There is one easy conclusion to make here: The input length is not important and input length smaller than 7 days tends to be better. This can be explained by several assumptions such as the wind is spontaneous so giving as input the measurement of the wind in the closest weather station 10 days ago gives no information to the model.

In other words, physically the weather is a highly Brownian thing, so any data relative to more than a week are just noise given to the model.

It was decided to go with 3 days as it was the one that gave the best results and with a training time per epoch not too significant.

5.5.2 Impact of the size of training data

It is a well-known fact that a neural network requires a lot of data to achieve great results. However, in this case the amount of clean data that could be taken was limited due to several factors. The main factor is the ECMWF forecasts. A part of the historical forecasts that was being used cannot be accessed due to damaged tape within ECMWF data center. Therefore, it could only be retrieved data from April 2019 onwards. This small analysis tries to extrapolate a curve that shows how the increase of training data would improve the performance of our model.

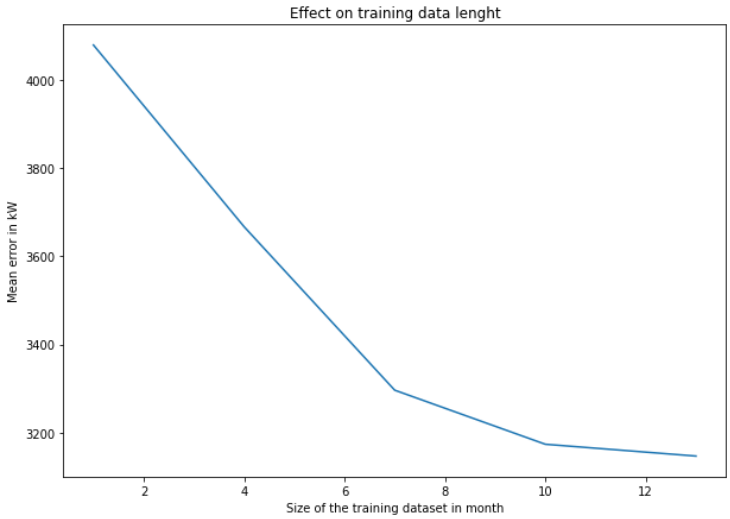

Figure 13 Impact of training size on error

One can see above a curve which shows how the error evolves by increasing the number of consecutive months in the training data.

The results must be interpreted with caution as the training of the model is not perfectly stable due the limited amount of data that was possible to gather. During the training process, batch of data are fed into the model to compute the error and then backpropagate the error to update the weights of the networks. However, the batch are selected randomly so the results are not constant and depend on the batches. Usually, the huge amount of data makes the network more stable but here since the dataset is limited, it is not our case. To be more accurate, a lot more data would be needed and plot the same graph for typically 5 years. The conclusion would be that more data would for sure help the model understands better and that would lead to decreasing the error.

5.5.3 How the distance between weather stations and assets impact the output?

To begin this analysis, one can find in the following table the distance between the closest weather station and each asset. Those station are the one selected to be used as input for each asset.

Table 5 Distances between assets and the closest dwd weather station

Assets	Distance to the closest dwd weather station (km)
Asset A	23.83
Asset B	21.23
Asset C	18.70
Asset D	16.83
Asset E	19.14
Asset F	18.96

The second point that is important to emphasize here is power curves. Scientifically, when a wind turbine is built, the constructor has a power curves which links wind speed to power output for this specific turbine. This curve should look similar for every turbine as it illustrates the equation below:

$$\boldsymbol{Power} = \frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{\rho A C_p v^3} \tag{7}$$

With ρ being the air density, A the swept area of the turbine, v the wind speed and $\boldsymbol{C_p}$ is the maximum power coefficient. This equation tells us that the power should be a cubed function of the wind. This figure below shows how a curve should look like:

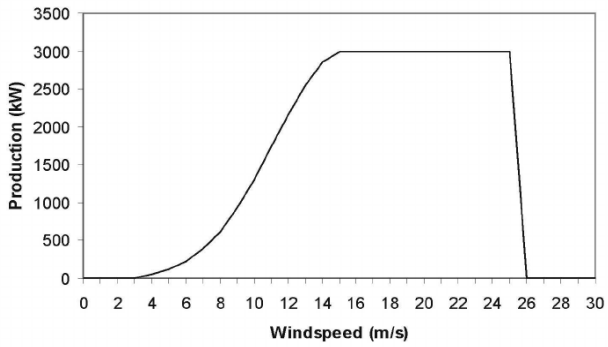

Figure 14 Theoretical power curve shape

Now this curve is plotted for our case study with different weather stations that are more and more far from the asset. Because the theoretical curve is about the production of one turbine only and not the entire asset, it was decided to study the asset E which has only one turbine. Below is a figure with different plots for different dwd weather stations. In the title of each plot one can find the distance between the weather station and the asset.

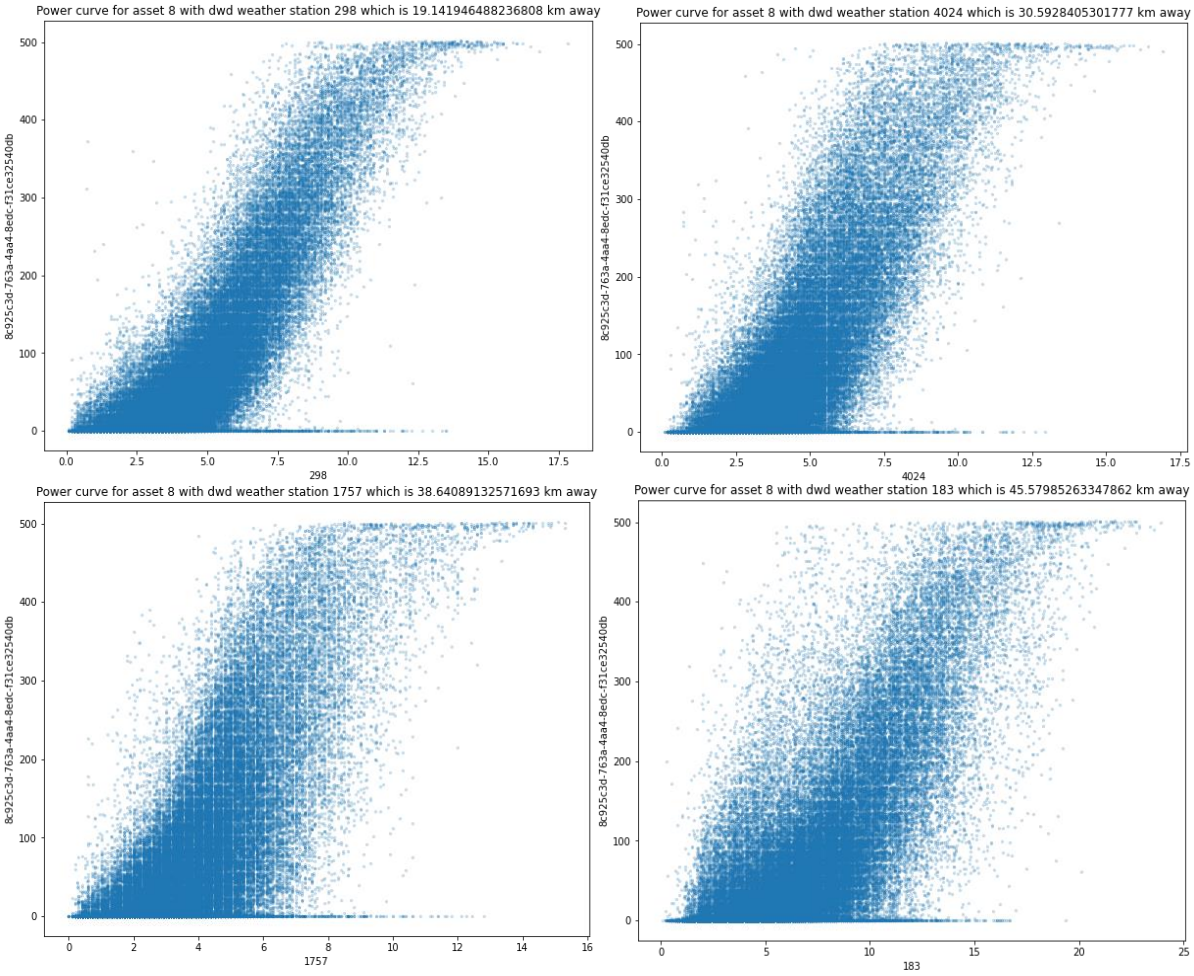

Figure 15 Power curve for the asset E with different weather stations

One can clearly see a difference in the shape of the power curve when the distance between the asset and the dwd weather station increases. This can explain by the fact that the wind is really localized and that few kilometres can have very distinct wind speed.

The impact of the distance can be identified by looking at the distance of the closest weather stations and the relative difference in the error between EM and our model. This relative error can be computed as follow:

$$\boldsymbol{Relative\ error} = \frac{\boldsymbol{Error_{model}} - \boldsymbol{Error_{EM}}}{\boldsymbol{MaxCapacity}} \quad (7)$$

Table 6 Correlation between relative error and distance to closest weather station

Asset	Distance to closest dwd weather station (km)	Relative error
Asset D	16.83	0.0066
Asset C	18.70	0.0097
Asset F	18.96	0.0153
Asset E	19.14	0.0162
Asset B	21.23	0.0169
Asset A	23.83	0.0173

There is one easy correlation to see here: the further the dwd weather station the bigger the relative error. That means that our model performs better when the distance with the asset is smaller. Of course, this must be taken with caution also as the error of our model is not stable due to the lack of consistent data.

As a side observation, these curves also expose the drama of wind power: its intermittence. These graphs clearly show how seldom the power produced equals the installed capacity, and the observer can concretely see that most of the time, the turbines produce nothing or close to nothing.

5.5.4 Different models for different timeframes

When observing the commercial predictions available, the continuity from present conditions towards the future is effective, which is not always the case when using one single neural network trained model. Therefore, it is likely that hybrid approaches are used to provide short term and mid-term predictions.

In effect, short-term and mid-term predictions are likely to behave differently. Hence using different trained neural networks for each one should make sense, with some logic to reconcile the eventual gaps between the two at their frontier, or even giving the first as input for the second.

5.6 Conclusion

5.6.1 Framework

To sum up this report, the goal was to use machine learning to forecast wind power production and compare this forecast to traditional site-specific models. In terms of site-specific models, Energy Meteo forecasts were used, which are the best in the market at the time of writing this report.

The sequence-to-sequence model was used, which are one of the best approaches in terms of multi-step ahead forecasts.

5.6.2 Limitations - data quality and availability

There is a principle which says that getting and preparing the data is 80% of the project. And this project did not make an exception to the rule. Among the several issues that were encountered during this project, most were related to the data gathering.

In order to get the data that make sense for this project different sources had to be explored across different origins, including internal data available, and the internet. To get weather forecast, ECMWF was the best choice as it is well-known and commonly used in every project concerning power forecast and wind power forecast. However, even there, they have an issue with their storage servers which make impossible to retrieve most of the much-needed gridded weather forecasts.

So, it ended up that as of early 2021, the best range of data possibly available corresponds to one year for the training part, 2 months for the validation and 2 months for the test data. After different approaches to asset owners it was not possible to access actual weather measurements on-site, so measurement from nearby weather stations had to be taken.

Note: it will not be possible to Alpiq to obtain better data for this study; and the achievements highlighted here-after constitute the best outcome that was possible to achieve with the data at hand.

5.6.3 Achievements

Even if they are strictly speaking not as good as Energy Meteo, the results that were achieved are, for all assets, almost as performing as the top-of-the-market forecasts provided by Energy Meteo. Besides, these performances are quite encouraging considering the several limitations that had to be dealt with.

Hence, with longer and closer data at hand, the results show that it is possible to achieve results equal or better than those of Energy Meteo. So, a sensitivity analysis was conducted with the aim of illustrating that better results can be achieved with data of better quality. This analysis focuses on the size of the data, the size of the input length to feed into the model, and on the effect of the distance between asset and weather stations.

The following conclusions arise:

- For the data that was available, the size of the input sequence, ranging from 6 hours to a month, did not have a significant importance to predict 6 hours of wind production.
- By increasing the size of the input data, the error of the model would decrease but not drastically.
- The distance between actual weather measurements and the asset is of crucial importance. The power output of a turbine is highly correlated to the wind speed which is very local. This effect was observed by plotting different power curves for the same asset but only changing the distance between the asset and the weather station. The shape of the curve got scattered as the distance increased. It was also possible to compute the relative error with the Energy Meteo forecasts and it has been identified that the bigger the distance, the bigger the relative error.

All those conclusions are encouraging for the moment and circumstances when, one day, higher quality data would be available to train a similar model.

6. Price Forecasts

Since the deregulation of the energy markets in the 1980s, electricity price forecasting (EPF) has become an increasingly important element of the decision-making of energy companies, whilst attracting more attention from researchers in academia.

Electricity price forecasting is a challenging and complicated task, both due to the multitude of factors affecting electricity price and the characteristics of electricity that make it is a rather unique element of the commodities asset class. Electricity cannot be stored at large scale, making it difficult to balance supply with demand - a prerequisite for the efficiency and stability of the energy systems. This in turn results in volatility and complex price dynamics [12]. A multitude of external variables such as seasonal patterns, weather, other fuel prices increase uncertainty about demand and supply, further complicating the task of forecasting electricity prices. Balancing energy systems has become even more of a serious challenge in recent years due to the growth in volatile “renewable” energy supply. Maintaining accurate forecasts is thus an essential element of an energy company's operations, with forecast accuracy improvements translating into significant element of financial and operating performance.

The significant body of electricity price forecasting research has, until recently, been primarily focused on point prediction. However modern developments such as intermittent energy, smart grids and increased competition have brought on better understanding of the importance of probabilistic electricity price forecasting due to increased volatility and complexity of the future supply [18].

In his review of state of the art methods at the time, Weron [25] speculated that probabilistic forecasting would be one of the key directions for the development during the next decade (2014-2024). This prediction has been largely confirmed with probabilistic price forecasting becoming both more developed and popular, as evidenced by participation in the Global Energy Forecasting Competition (GEFCom2014) [11]. Probabilistic price forecasting has significant benefits by allowing energy companies to evaluate uncertainty in a more comprehensive and rigorous manner, resulting in improved strategic and tactical planning, as well as the improved effectiveness of the submitted bids. [18]. At the same time, the current state of forecasting still retains significant focus on point predictions, requiring both the development of new probabilistic forecasting methods and their improved use by the industry.

Over the years, both statistical and machine learning methods have been employed to forecast electricity prices, until recently the results have been rather inconclusive in terms of both classes of methods providing superior performance across wide range of datasets. The statistical methods are designed to capture the characteristics of electricity prices [24] by modelling the dependence of future electricity prices with the past history and external variables. A typical parsimonious model would only use historical prices and often utilize an 'auto-regression' model structure. The main advantage of statistical models is that they are well-understood, are easy to interpret and are tailored to account for electricity price patterns. The disadvantages include utilization of linear dependencies that are often unable to fully capture complicated and rather unique patterns of electricity patterns, resulting in sub-optimal forecast performance.

Deep learning has seen great success in areas such as computer vision (CV) and natural language processing (NLP). Whilst deep learning research for time-series forecasting has been lagging compared to CV and NLP, the recent forecasting competitions such as M4 [13] and M5 have demonstrated clear advantages of considering machine and deep learning methods for time-series forecasting problems in addition to classical statistical methods. All the winners of the M5 forecasting competition have utilized machine / deep learning methods, with additional research indicating that some machine learning methods provide better forecasts in terms of both accuracy and bias [21]. Cross-learning across time-series in particular provides clear advantages of deep learning methods compared to statistical ones, although other advantages such as the ability to process complex non-linear structures and external variables, combined with expressive power and the ability to exhibit temporal dynamic behaviour [10] provide clear advantages for univariate time-series forecasting.

Following developments in machine learning research for time series forecasting, several state of the art machine and deep learning approaches were considered for the modelling of electricity prices:

1) Multichannel Singular Spectrum Analysis (mSSA) - a popular and widely used time series forecasting method. As demonstrated in [1], mSSA was found to outperform deep neural network architectures such as LSTM and DeepAR in the presence of missing data and noise level.
2) DeepAR a methodology (developed by Amazon Research) for producing accurate probabilistic forecasts, based on training an auto-regressive recurrent network model [20]. GluonTS [2] implementation of DeepAR is used for the experiments.
3) N-BEATS - a deep neural architecture based on backward and forward residual links and a very deep stack of fully-connected layers [19]. The architecture demonstrated good performance in M4 forecasting competitions and more recently [19] has been used for electricity load forecasting.

The above models are applied to generate and benchmark 168-hour ahead forecasts at three time points: 1) 31-Dec-2020 2) 31-Jan-2021 3) 29-Feb-2021. Our approach is to demonstrate benefits from using machine and deep learning forecasting technologies, rather than comprehensive evaluation of different classes of methods or add to the debate on the benefits of machine learning vs. statistical algorithms. Forecasts were generated for 168 hours (7 days) ahead and benchmark forecasts against 168 hour lagged naive forecast that is able to capture daily and hourly dynamics of electricity prices (thus is competitive benchmark in the short term, especially as our forecasts are of parsimonious nature and only use historic price information).

Mean squared error (MSE) has been utilized to measure performance of considered forecasting frameworks to obtain the following results in Table 4. Our results demonstrate significant benefits (as measured by forecasting value added - FVA) from applying powerful deep learning frameworks such as DeepAR and N-Beats, as well data-driven models such as mSSA that are able order to capture dynamic and feature-rich behaviour of electricity prices. The applied frameworks are able to demonstrate, even without any hyper-parameter optimization, that powerful open-source frameworks such as mSSA, DeepAR and N-Beats are able to generate (see Table 6) very competitive forecasting inputs in comparison with expensive commercial forecast feeds.

Table 7 Mean Squared Error

Forecast	Jan-20	Feb-20	Mar-20
Naïve	493.62	553.76	372.92
mSSA	372.92	402.01	1016.17
DeepAR	272.93	220.53	254.01
N-Beats	275.22	201.1	106.24

7. Nuances of Optimization

We optimize against prices which implies that turbines should produce when the price is high and be off when it is low. As for the battery, it should discharge when the price is high and it should charge the battery when it is either low or negative.

7.1 The choice of sub-pools

At time t, we take the optimal solution from time $t-1$ and calculate the following arrays:

$$P_k = \sum_{i=1}^{\#T(k)} \sum_{\tau=1}^{T} p_{\tau,k,i}^{ID}, \qquad k \in \{1,2,\dots,\#A\},$$

$$P_{ki} = \sum_{\tau=1}^{T} p_{\tau,k,i}^{ID}, \qquad k \in \{1,2,\dots,\#A\}, i \in \{1,2,\dots,\#T(k)\}.$$

So, when choosing which power plant to add to the sub-pool, we prefer such power plants k for which P_k is larger. When we chose a power plant k as part of the pool, we decide which turbines to add first according to the values P_{ki}, i.e. we prefer the higher values of P_{ki}. Since we run optimization every 15 minutes, in many cases there is not a very significant difference between the problems at time t and $t-1$ and this is one of ways to exploit it.

7.2 Hardware and solvers

We use m5d.4xlarge EC2 instance within Amazon Web Services i.e. 16 vCPUs and 64 GiB RAM. In optimization, we employ two solvers: Cbc and Gurobi. In order to optimize our set of assets, we can get by with CBC, but in order to deal with portfolios of replicated biogas power plants, we will need to resort to the commercial solver Gurobi.

7.3 Implementation of Proximal Jacobian ADMM

The proximal Jacobian ADMM (Alternating Direction Method of Multipliers) was developed in [8] and [23] and its convergence for linear programming problems was proven in the same literature. In this study apart from Gradual Increase, we also use Proximal Jacobian ADMM, where we first relax all integrality constraints and let the algorithm iterate until the acceptable violation of coupling constraints is achieved: the preservation of the rest of the constraints is provided by the subproblems. After finishing the linear programming part we get the action vectors x_k^{LP}, for each $k \in \{1,2,\dots,\#A\}$. Let a_k^{LP} denote $A_k \cdot x_k^{LP}$, i.e. $a_k^{LP} = A_k \cdot x_k^{LP}$. Then for each asset k, we recover the mixed-integer action vector by solving the following subproblems.

$$\begin{aligned} \text{maximize} \quad & p_k^T \cdot x_k \\ s.t. \quad & A_k \cdot x_k = a_k^{LP}, \\ & B_k \cdot x_k \le b_k, \\ & x_k(I) \in \{0,1\}. \end{aligned} \tag{31}$$

Sometimes these problems are infeasible, but we handle this as follows: if K is a subset of $\{1,2,\dots,\#A\}$ such that for each $k \in K$, problem (31) is infeasible, then we can solve:

$$\text{maximize} \quad \sum_{k \in K} p_k^T \cdot x_k$$

$$s.t. \quad \sum_{k \in K} A_k \cdot x_k = \sum_{k \in K} a_k^{LP},$$

$$B_k \cdot x_k \leq b_k,$$

$$x_k(I) \in \{0,1\},$$

$$\forall k \in K.$$

In all our problems $\#K \leq 4$ and such pools are solved in matter of seconds. However this procedure provides sub-optimal solutions, but this technique is amenable to parallelization [23] and can handle extra large pools of assets. It can also be easily shown that this algorithm can be combined with *Gradual Increase*.

7.4 Parallelization

Parallelization is crucial in optimizing these kinds of virtual power plants. And when applying any aforementioned methods, we propose using two independent machines where the first machine runs the problem from scratch uninterrupted and the second machine starts from scratch, but gets interrupted when decomposition, splitting or pruning methods find a new feasible solution and then, with a new start, they resume the calculations, having preserved all the branch & bound trees. When any method gets the confirmation that there is the optimal solution, then all other cores terminate. The same happens when the time limit is expired. In this case of all feasible solutions found by all methods, we choose the one which yields the largest value of the objective function. The usage of an independent and uninterrupted core is proposed in order to make sure, that decomposition methods will not lead to longer computation times. This can happen in situations when a solver was capable of finding the optimal solution almost immediately and most of the time was spent on the confirmation that the solution is optimal. In such special cases GI would only decelerate the total computation time but it will not happen if it is coupled with such a core. We also propose utilizing an interrupted machine in order to preserve branch & bound trees: each feasible solution can enrich the search space.

Table 8 Notation for Model Predictive Control

The Symbol	Explanation
H	the prediction horizon
h	the execution horizon (realized schedule)
$\nabla prod_t^W$	the realized imbalance
$\boldsymbol{\nabla prod_t^W}$	the forecasted imbalance for time t $\boldsymbol{\nabla prod_t^W} = [\Delta prod_t^W, \Delta prod_{t+1}^W, \ldots, \Delta prod_{t+H-1}^W]$
$\Delta prod_t^W$	the forecasted imbalance for time t
$\boldsymbol{\Delta prod_t^W}$	the forecasted imbalance for times from t to $t + H - 1$
$\boldsymbol{State_t}$	the state of the system at time t: storage levels, and the states of turbines.

8. Results

In this section, we present the results of our calculations related to the speed (Subsection 7.1) and the optimality in a stochastic environment (Subsection 7.2). In both cases, we consider the following period: 2020/1/1 - 2020/9/26. In the case of the speed, our prediction horizon (H) is two days ahead. In the case of optimality, our prediction horizon (H) is 3 days ahead. In both cases the execution horizon (h) is 24 hours. In the case of the speed, we use only state of the art commercial price forecast and assume perfect knowledge of the imbalance. We emphasize that the size of imbalance increases proportionally to the number of assets; otherwise the optimization problem would be trivial. In the case of optimality, we consider different cases, described in Subsection 7.2.

8.1 Speed

When we explore the speed, our benchmark is the time needed to solve the MILP problem by the Gurobi *solver*. We call this approach the *Brute Force* and the proposed decomposition and splitting algorithms must enable us to find the solution faster. Table 5 summarizes the average calculation times provided by different methods. The headers of each column refer to the method applied and in the caption of the table there is the explanation of the abbreviations in the headers. The methods in the headers with italic font are the techniques which yield sub-optimal solutions but the resulting value of $TotRev$ yielded around 99.96% of the maximum. Such methods are: partial integrality, partial integrality coupled with gradual increase, proximal Jacobian ADMM, and proximal Jacobian ADMM coupled with Gradual Increase. The methods such as Brute Force, Gradual Increase, and (≤)-prune lead to the optimal solution, provided the *solver* has enough time. In Figure 1 and Table 5, we can observe how Gradual Increase (GI) outperforms Brute Force (BF) in terms of time. When we solve problems by Brute Force, we only have the time limit of 15 minutes, i.e. the computations finish if either the optimal solution is found (the relative difference between the upper and the lower bound is below 0.01%) or the time limit has expired. In columns BF, GI, *PI, GI-PI*, and (≤)-prune we try to get the optimality message within 15 minutes. As for methods based on ADMM, this approach is irrelevant and the stopping criteria for these algorithms are described in Subsection 6.3. So, in Figure 1, we can see that GI improves over BF when we increase the number of assets, but when we get closer to 200 then the difference is smaller. This is because we finish either when the optimal solution is found or when the time limit is exceeded. Thus, the more assets we have in the portfolio, the more cases we face when the computation time would expire. As stated above, GI helps us find the optimal solution, but it has nothing to do with the confirmation of the optimal solution (calculation of upper bounds). In order to accelerate this confirmation, we can either resort to the method (≤)-prune or to *Partial Integrality*. In this case, we do not get to the expiration of the time limit and the fastest method turned out to be the combination of *Gradual Increase* and *Partial Integrality* which yields average calculation time 4.17 times smaller than that of *Brute Force*. The method *Proximal Jacobian ADMM* is relevant only for large pools of assets, therefore we started calculations from 52 assets in the pool. And we can see that *Brute Force* outperforms this method in terms of speed for 52, 102, and 150. But when we have 200 assets, then *Proximal Jacobian ADMM* outperforms *Brute Force*. When we combine *Proximal Jacobian ADMM* with *Gradual Increase*, then we get further acceleration (column ***GI-PJ-ADMM***) for 200 assets – approximately 1.3 times faster than *Brute Force*. Another positive side of *Proximal Jacobian ADMM* is that it imposes lighter memory requirements on the machine than other proposed methods [5]. In our further research, we are going to combine all these methods and to conduct pruning non-tight constraints by means of the methods of machine learning.

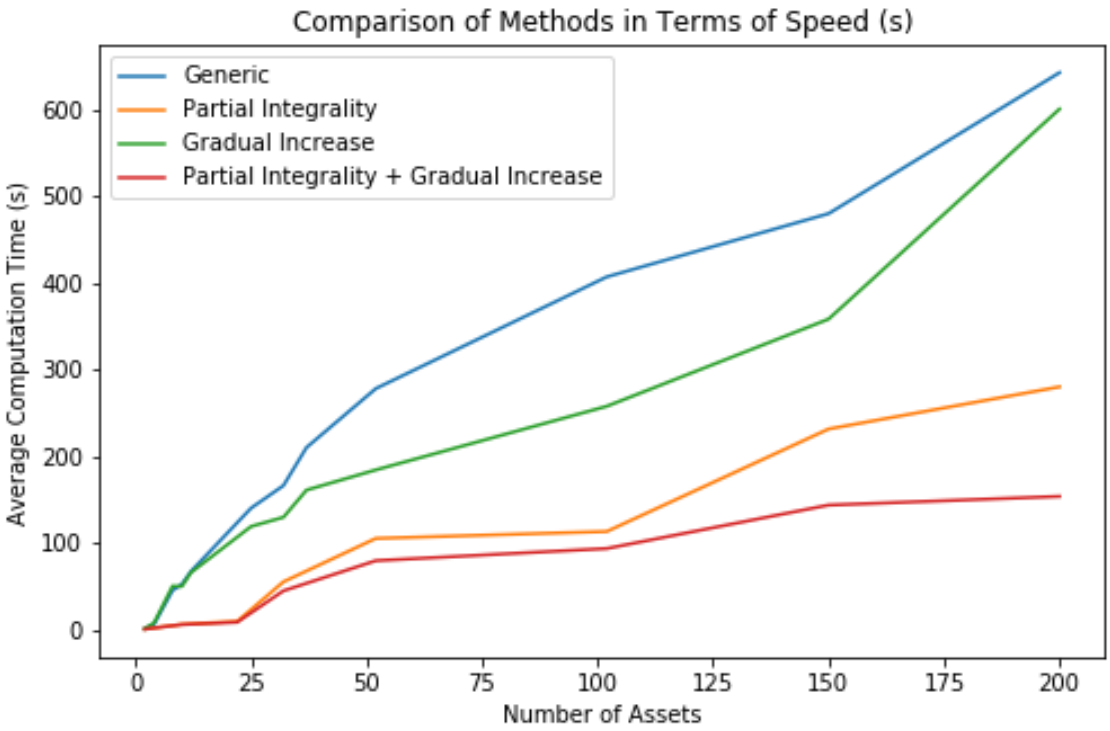

Figure 16 Comparison of Average Computation Times (s)

8.2 Optimality

When we explore the optimality, we attempt to get as close as possible to the total revenue that we would achieve in the case of perfect foresight, when both prices and imbalances are known in advance. So, the cumulative revenues under perfect foresight is our benchmark and we explore the percentage of it provided by different methods. We explore this by means of variable σ (Defined in Subsection 2.4.5), which is apparently unit for the case of perfect foresight. Figure 2 summarizes the cumulative revenues. All these revenues are divided by total revenue yielded by the perfect forecast, i.e. the variable $TotRev$ associated with perfect foresight defined in Subsection 4.1. So the last point of every line in Figure 2, i.e. the one, situated furthest to the right, is the value of σ corresponding to the forecasts and MPC methodology. Table 6 summarizes the σ-values. We can observe that when we use commercial state of the art forecasts and assume perfect foresight of productions, then RMPC yields slightly higher revenues. In the case of uncertain productions, we utilize only RMPC because we prefer simulations of the realizations of productions rather than trying to find a point forecast: the forecast of the imbalance is a challenging task and the exploration of it exceeds the scope of this report. But mimicking this imbalance by ARMA processes yields σ equal to 84.19% if we use commercial forecasts and σ equal to 81.29%, if we use mSSA, which can be achieved by utilizing open-source software.

Table 9 Average speed of calculations (in seconds) for methods: Brute Force (BF), Gradual Increase (GI), Partial Integrality (PI), Combination of Gradual Increase and Partial Integrality (GI-PI), Pruning of non-tight constraints ((≤)$-prune), Proximal Jacobian ADMM (PJ-ADMM), Combination of Gradual Increase and PJ-ADMM (GI-PJ-ADMM).

#Assets	BF	GI	*PI*	*GI-PI*	(≤)-prune	*PJ-ADMM*	*GI-PJ-ADMM*
4	6	8	-	-	4	-	-
8	45	50	-	-	33	-	-
10	52	49	6	5	27	-	-
25	140	119	9	8	87	-	-
32	166	129	55	44	95	-	-
52	278	184	105	79	192	390	368
102	407	257	113	93	233	450	411
150	480	358	231	143	279	497	479
200	643	601	280	154	381	532	495

Table 10 The σ-value for MPC/RMPC and different forecast methods (σ is the percentage of maximum possible revenue).

Price Input	Imbalance Input	Control Method	σ (in %)
Perfect Foresight	Perfect Foresight	MPC	100.00
Commercial	Perfect Foresight	RMPC	94.40
Commercial	Perfect Foresight	MPC	93.98
mSSA	Perfect Foresight	MPC	88.75
Commercial	ARMA simulations	RMPC	84.19
mSSA	ARMA simulations	RMPC	81.29
DeepAR	ARMA simulations	RMPC	77.34
N-BEATS	ARMA simulations	RMPC	72.53

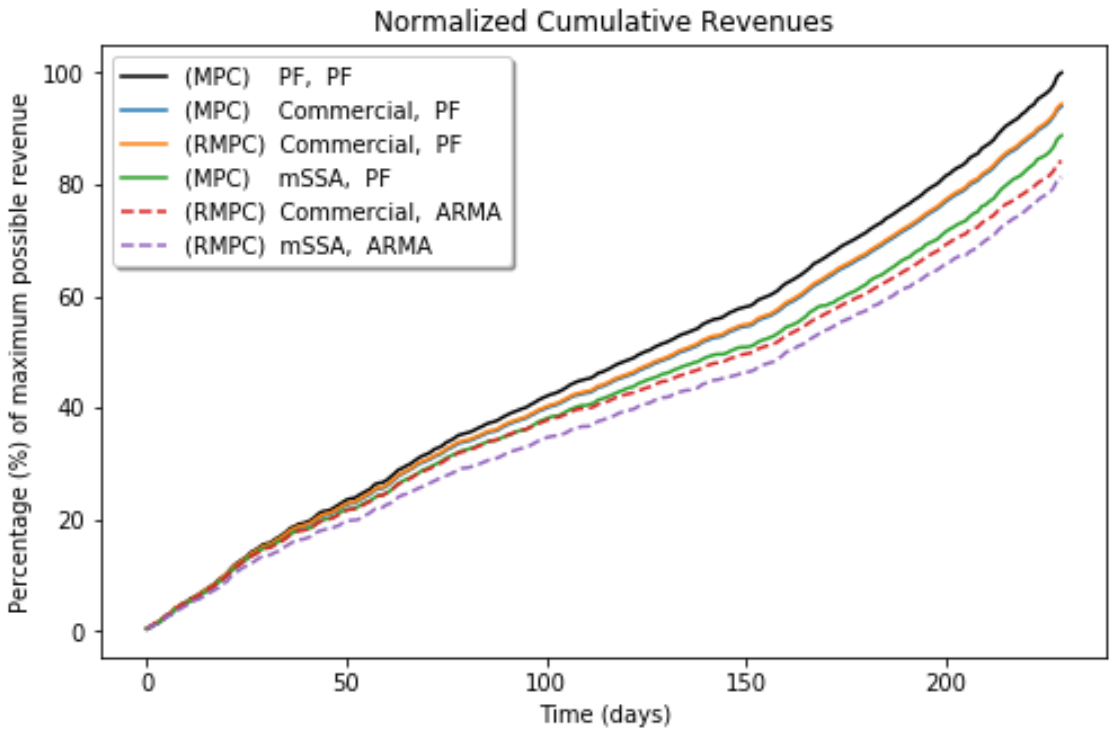

Figure 17 Comparison of Cumulative Revenues (normalized)

9. Conclusion

One of the most important objective of a VPP operation is to optimize the revenues resulted from its assets operation. In this report is presented an algorithm for optimization of a VPP operation, with the significant side effect of increasing the volume of electricity generated on RES. The complex optimization algorithm developed in edgeFLEX requires input data of a high quality. Therefore, the authors of the present report have developed innovative algorithms for the forecast of electricity prices and volume of wind-based electricity generation. Full implementation of the solutions presented in this report, in the operation of a VPP is expected to increase the revenues of the RES-based electricity generators members of the respective VPP.

We have explored the computation times of GI and Proximal Jacobian ADMM dependent on the number of biogas power plants. The main point of these calculations is that we are able to handle up to hundreds of power plants within a pool if we properly use decomposition and splitting methods (or their combination). We also learned that the combination of Gradual Increase with Partial Integrality yields further improvements in computation time. Similar results are achieved when we combined Gradual Increase with Proximal Jacobian ADMM. We have also conducted experiments with a pool of two biogas power plants and a single battery in order to compare different forecasts (Commercial state of the art, mSSA, DeepAR, N-BEATS) and optimization methods (MPC and Robust MPC). The usage of Robust MPC yielded the revenue 0.447 % higher than that of MPC, therefore in our future research we will consider combining Robust MPC with decomposition methods in order to achieve analogous results for larger pools. When we optimize the VPP by means of RMPC under price and imbalance uncertainty, then the utilization of commercial price forecasts yields 84.19% of the maximum possible revenue whilst the usage of our mSSA forecasts yields 81.29% of it. However, this forecast can be achieved by using of open-source software. The violet line in Figure 6 (i.e. the line '(RMPC) mSSA, ARMA' was calculated by entirely open-source software: the optimization was conducted via CBC solver. All the optimization calculations related to Table 6 and Figure 2 were conducted with CBC solver. The ability to run virtual power plants by means of open-source software also facilitates balancing and the incorporation of newly distributed energy resources into the grid. However, there is still the need for commercial solvers when we deal with large systems consisting of hundreds of assets.

10. List of Tables

11. List of Figures

12. References

[1] [Dataset] Agarwal, A., Alomar, A., and Shah, D. (2020). On multichannel singular spectrum analysis

[2] Alexandrov, A., Benidis, K., Bohlke-Schneider, M., Flunkert, V., Gasthaus, J., Januschowski, T., et al. (2020). Gluonts: Probabilistic and neural time series modeling in python. Journal of Machine Learning, Research 21, 1–6

[3] Bemporad, A. (2006). Model predictive control design: New trends and tools. In Proceedings of the 45th, IEEE Conference on Decision and Control (IEEE)

[4] Bertsimas, D. and Stellato, B. (2020). The voice of optimization doi:arXiv:1812.09991

[5] Boyd, S., Parikh, N., Chu, E., and Peleato, B. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning 3, 1–122. doi:10.1561/22000000165

[6] Chen, G. and Li, J. (2018). A fully distributed ADMM-based dispatch approach for virtual power plant problems. Applied Mathematical Modelling 58, 300–312. doi:https://doi.org/10.1016/j.apm.2017.06.010.

[7] DCbrain (2020). Integrating green gases. how to reach new levels of performance with hybrid ai

[8] Deng, W., Lai, M.-J., Peng, Z., and Yin, W. (2017). Parallel multi-block ADMM with o(1 / k) convergence. Journal of Scientific Computing 71, 712–736

[9] Graabak, I. and Korpas, M. (2016). Balancing of variable wind and solar production in continental europe with nordic hydropower – a review of simulation studies. Energy Procedia 87, 91–99. doi: https://doi.org/10.1016/j.egypro.2015.12.362. 5th International Workshop on Hydro Scheduling in Competitive Electricity Markets

[10] Hewamalage, H., Bergmeir, C., and Bandara, K. (2021). Recurrent neural networks for time series forecasting: Current status and future directions. International Journal of Forecasting 37, 388–427

[11] Hong, T., Pinson, P., Fan, S., Zareipour, H., Troccoli, A., and Hyndman, R. J. (2016). Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond. International Journal of Forecasting 32, 896–913. doi:10.1016/j.ijforecast.2016.02.001

[12] Kaminsky, V. (2013). Energy Markets (Risk books)

[13] Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2020). The m4 competition: 100, 000 time series and 61 forecasting methods. International Journal of Forecasting 36, 54–74. doi:10.1016/j.ijforecast. 2019.04.014

[14] Mattingley, J., Wang, Y., and Boyd, S. (2011). Receding horizon control. IEEE Control Systems Magazine 31, 52–65.

[15] Moehle, N., Busseti, E., Boyd, S., and Wytock, M. (2019). Dynamic energy management. Springer, Large Scale Optimization in Supply Chains and Smart Manufacturing , 69–126

[16] Morales-Espana, G., Gentile, G., and Ramos, A. (2015). Tight MIP formulations of the power-based unit commitment problem. OR Spectrum 37, 929–950. doi:https://doi.org/10.1007/s00291-015-0400-4

[17] Munsing, E., Mather, J., and Moura, S. (2017). Distributed optimization and statistical learning via the alternating direction method of multipliers. IEEE Conference on Control Technology and Applications (CCTA) , 2164–2171doi:10.1109/CCTA.2017.8062773

[18] Nowotarski, J. and Weron, R. (2018). Recent advances in electricity price forecasting: A review of probabilistic forecasting. Renewable and Sustainable Energy Reviews 81, 1548–1568. doi:10.1016/j.rser.2017.05.234

[19] Oreshkin, B. N., Dudek, G., and Pelka, P. (2020). N-beats neural network for mid-term electricity load forecasting. ArXiv abs/2009.11961

[20] [Dataset] Salinas, D., Flunkert, V., and Gasthaus, J. (2019). DeepAR: Probabilistic forecasting with autoregressive recurrent networks

[21] Spiliotis, E., Makridakis, S., Semenoglou, A.-A., and Assimakopoulos, V. (2020). Comparison of statistical and machine learning methods for daily SKU demand forecasting. Operational Research , 1–25

[22] Tan, Z., Li, H., Ju, L., and Tan, Q. (2018). Joint scheduling optimization of virtual power plants and equitable profit distribution using shapely value theory. Mathematical Problems in Engineering 2018

[23] Wei, E. and Ozdaglar, A. (2013). On the o(1/k) convergence of asynchronous distributed alternating direction method of multipliers. In EEE Global Conference on Signal and Information Processing (IEEE)

[24] Weron, R. (2006). Modeling and Forecasting Electricity Loads and Prices (John Wiley & Sons Ltd). doi:10.1002/9781118673362

[25] Weron, R. (2014). Electricity price forecasting: A review of the state-of-the-art with a look into the future. International Journal of Forecasting 30, 1030–1081. doi:10.1016/j.ijforecast.2014.08.008

13. List of Abbreviations

ADMM	Alternating Direction Method of Multipliers
ARMA	AutoRegressive Moving Average
BF	Brute Force
DER	Distributed Energy Resources
DWD	Deutsche Wetter Dienst
ECMWF	European Centre for Medium-Range Weather Forecasts
EM	Energy Meteo
GI	Gradual Increase (a method of warm starts)
LSTM	Long Short-Term Memory
ML	Machine Learning
MPC	Model Predictive Control
mSSA	Mulchannel Singular Spectrum Analysis
NWP	Numerical Weather Prediction
PI	Partial Integrality
PJ	Proximal Jacobian
RMPC	Robust Model Predictive Control
RNN	Recurrent Neural Network
TIGGE	The International Grand Global Ensemble
VPP	Virtual Power Plant