# Data Storage Cost Optimization Based on Electricity Price Forecasting with Machine Learning in a Multi-Geographical Cloud Environment


Jakob Wirén, jakwi732@student.liu.se

Supervisor: Ioannis Avgouleas, ioannis.avgouleas@liu.se
Examiner: Vangelis Angelakis, vangelis.angelakis@liu.se

August 16, 2018

# Abstract

As increased demand of cloud computing leads to increased electricity costs for cloud providers, there is an incentive to investigate in new methods to lower electricity costs in data centers. Electricity price markets suffer from sudden price spikes as well as irregularities between different geographical electricity markets.

This thesis investigates in whether it is possible to leverage these volatilities and irregularities between different electricity price markets, to offload or move storage in order to reduce electricity price costs for data storage.

By forecasting four different electricity price markets it was possible to predict sudden price spikes and leverage these forecasts in a simple optimization model to offload storage of data in data centers and successfully reduce electricity costs for data storage.

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

ACF - Autocorrelation Function
AEMO - Australian Energy Operator Market
AI - Artificial Intelligence
AIC - Akaike Information Criterion
API - Application Programming Interface
ARMA - Autoregressive Moving Average
BIC - Bayesian Information Criterion
CAPEX - Capital Expenditures
CDN - Content Delivery Network
CET - Central European Time
CPI - Consumer Price Index
DC - Data Center
IESO - Independent Electricity System Operator
IT - Information Technology
EEA - Energy Exchange Austria
EWMA - Exponential Weighted Moving Average
IaaS - Infrastructure-as-a-Service
MAE - Mean Absolute Error
MSE - Mean Squared Error
PaaS - Platform-as-a-Service
PACF - Partial Autocorrelation Function
RBF - Radial Basis Function
SaaS - Software-as-a-Service
SLA - Service Level Agreement
SVM - Support Vector Machine
SVC - Support Vector Classification
SVR - Support Vector Regression
VM - Virtual Machine
QoS - Quality of Service

# Chapter 1

# Introduction

*This chapter is intended to give an introduction of the thesis and an overview of the main components of the report as well as the main questions the thesis is intended to answer.*

## 1.1 Motivation

Cloud computing has risen in popularity by offering users centralized computing and storage resources, which lower large investment costs of servers in order for companies and organizations to pay and scale as they grow. As data traffic grows with an exponential pace, the demand for data centers (DCs) grow as well. DCs consume large amounts of electricity; in fact DCs constitute to 2% of the total electricity consumption worldwide with an estimated growth rate of 12% per year [19][12]. The power distribution can be broken down into four categories: cooling, IT, power and lighting and others which constitute for 39%, 45%, 13% and 3% respectively [24]. This translates into US$30 billion in 2008 for enterprise DCs [20].

Even if cloud computing enables consolidation and virtualization of hardware to use resources more efficiently, utilization levels are still relatively low. According to Ericsson, non-virtualized servers operate on average at a utlization level of 6 to 15% whereas virtualized servers can increase that number to 30% [27].

Typically DC providers have several DCs across different geographical locations to ensure reliability by using replication as well as coming closer to customers to meet latency demands. This implies that, different DCs in different geographical locations are subject to different electricity price markets. These electricity markets can vary largely in price. Hence, many big DC providers choose to build data centers where electricity is cheap, or with low average temperature to save money on cooling.

Many electricity price markets provide a deregulated spot market where it is possible to buy electricity to a spot market price which is set by supply and demand. Effectively, this generates volatility in the market and prices can tenfold within just an hour [13]. For example the lowest observed price on the Swedish spot market 2017 was €1.59/MWh whereas the highest observed price was €111/MWh [13]. Together with the increased demand of cloud computing and the volatile electricity price there is an incentive to investigate in leveraging volatilities in the deregulated electricity price market to predict price spikes in order to reduce electricity consumption during these periods to lower electricity costs.

Content Deliver Networks (CDNs) are used by companies like Netflix to move

certain content closer to the end users in order to minimize transmission of large amounts of data and improve quality of service (QoS) [11]. This technique could potentially be used to offload storage from centralized DCs to nodes in the edge of the network in order to shut down some servers and lower electricity consumption and ultimately lower electricity costs.

Previous works [19], [20] investigate how electricity costs can be reduced in multi-geographical environments. In [17] the authors examine how servers should be activated with respect to the electricity price. [3] develops an energy-efficient data placement algorithm for node scheduling. There also exists research in energy efficient algorithms for virtual machine (VM) migration. E.g., see [14]. In [8] the authors optimize the route selection for data transmission. However, the mentioned papers suggest solutions to specific parts of a problem. For example, the authors rely on given electricity prices. This study suggests a full-scale solution including price forecasting and data storage optimization in a simple use-case environment.

## 1.2 Problem Formulation

Due to the increased interest and demand for cloud computing there is pressure on cloud providers to find new ways to better utilize existing resources to decrease electricity costs. To stay profitable and simultaneously meet service level agreements (SLAs). Due to the volatility in the deregulated electricity market there is an incentive to investigate whether these irregularities can be leveraged to lower electricity consumption and, hence, electricity costs. Potentially by offloading storage to nodes similarly to a CDN.

## 1.3 Aim

The aim of the research is to investigate whether it is beneficial or not to leverage rapid changes in electricity price as well as price-differences between different geographical regions to offload storage of data to reduce operational costs in data centers.

In order to accomplish the aim, the following set of research questions are investigated in this thesis:

1. Is it possible to reduce costs by offloading data storage to different geographical regions depending on the electricity price?
2. Does the electricity price fluctuate over time and between different geographical locations?
3. With what accuracy is it possible to forecast the electricity price?
4. How does different forecasting techniques affect the forecasting accuracy?

## 1.4 Method

The approach to reach the aim of the project was organized as follows. First, a research period was conducted to search for related work and effectively determine

what methods have been applied to similar projects and can be leveraged in this project. Second, data collection was conducted. More specifically, electricity prices in particular regions were retrieved for data analysis. Third, the retrieved data was used for data analysis in conjunction with various mathematical methods to forecast the electricity price. Finally, the results of the forecast were applied in an optimization model to investigate whether it was possible to minimize the operational electricity costs for data storage in data centers.

## 1.5 Scope

The data used in this thesis is hourly electricity price data from different spot markets in Canada - Ontario, Australia - New South Wales, Sweden and Austria. These regions were chosen were because of data availability, the existence of deregulated spot markets and their intercontinental locations which resembles a good real-life situation. The deregulated electricity price market is further explained in Chapter 2.

## 1.6 Structure

The rest of the thesis is structured as follows. Chapter 2 explains the theory that is necessary to understand the rest of the content. Chapter 3 explains the methodology of data collection and preparation, data analysis, forecasting and the optimization formulation. Chapter 4 shows the results of the performed analysis followed by a discussion of the work and a final conclusion and recommendations for future works in Chapter 5 and 6 respectively.

# Chapter 2

# Theory

*This section explains relevant theories that are essential to understand the following parts of the study*

## 2.1 Cloud Computing

In cloud computing, IT resources such as: storage, computing, networking, databases etc. can be flexibly consumed on an on-demand basis. Rather than building a private IT-infrastructure, these resources can be accessed through the internet. Consequently, investment costs and capital expenditures (CAPEX) for IT infrastructure can be reduced by paying on-demand. Traditionally, companies would have to make a guess of how much capacity the IT-infrastructure would be able to handle. This most likely ends up in one of two scenarios: (i) the capacity is overestimated, which means money and time have been wasted in building an oversized data center, or (ii) the capacity is underestimated leading to the inability to meet customer demand, hence, losses of revenue. With cloud computing, customers are able to pay for what they use only. Furthermore, economies-of-scale offer cloud providers the ability to launch services to a lower price by dynamically provision their resources. Another advantage with cloud computing is reduced time-to-market for new products or services. Rather than spending days or weeks to set up new IT resources, this can be achieved in minutes by subscribing to a service [7][22][4].

Typically cloud services can be divided into three categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). IaaS allows customer to rent infrastructure on a pay-as-you-go basis. PaaS, enables customers to develop, test, deploy and manage cloud applications without an existing IT infrastructure. SaaS is a method to deliver end-user off-the-shelf applications on a subscription basis [4]. The research presented in this study is related to IaaS exclusively.

A cloud can be characterized as a public, private or hybrid cloud. In a public cloud the IT-infrastructure is owned by a third party and may be used by several customers - such as Amazon Web Serives (AWS), Microsoft Azure or Google Cloud. The IT infrastructure in a private cloud is privately owned and is in contrast to a public cloud only used by the owner of the infrastructure. A hybrid cloud is essentially a combination of a private and a public cloud which allows data in applications to be moved among private and public clouds [4].

### 2.1.1 Virtualization

By creating a simulated environment with help of a piece of software on top of the hardware, it is possible to create several virtualized computing environments which can be run on a single physical machine. By creating multiple virtual environments from one physical machine, IT infrastructure can be utilized more efficiently in terms of the number of used physical servers and consumed energy and, hence, costs [5]. The concept of virtualization is illustrated in Figure 2.1.

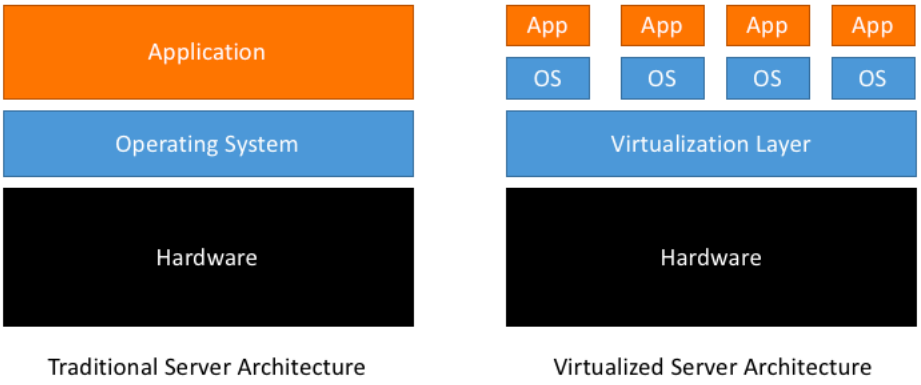


Figure 2.1: Traditional vs. virtualized server architecture

Figure 2.1 shows that, in the traditional architecture only one operating system (OS) and application can be mapped to one piece of hardware. Whereas in the virtualized architecture, the virtualization layer enables to instantiate several OSs with belonging application on one piece of hardware. Which facilitates a shared pool of resources allowing for a more flexible and effectively utilized infrastructure

### 2.1.2 Content Delivery Network (CDN)

Services that transmit large amounts of data and need to do so with low latency can benefit from CDNs. Netflix as an example, needs to stream large amounts of data all around the world and users require instant access. To be able to meet the customer demand of instantaneous access all around the world, Netflix has deployed over a thousand edge nodes in its CDN to be able to move data closer to customers and improve its QoS [11]. By predicting what content is popular in different regions, copies of that content are moved to the edge of the network. By doing so, the most frequently used data does not need to be transmitted over far distances or using expensive bandwidth over and over. Since the capacity in the nodes may be limited, only the content that is accessed frequently should be moved to the edge. By using this model, Netflix has increased their throughput from 8 Gbps from a single server in 2012 to 90 Gbps in 2016 from a single server [11].

## 2.2 Machine Learning

Machine learning is the concept of using data as fuel in sophisticated algorithms that learn patterns and behaviors of data to make decisions. These methods, have proven to be successful, especially for complex problems such as image recognition or spam detection for emails [15].

Machine learning is not a new concept. In 1959, Samuel Arthur formed a widely used definition of machine learning: "*Field of study that gives computers the ability to learn without being explicitly programmed*" [23]. But it was not until the 1990s it really got attention after discovering Moore's Law's effects on computing power [28]. Another triumph for machine learning and AI which increased the recognition was when IBM's computer Watson beat a human in the game Jeopardy [28].

Machine learning problems can typically be categorized as supervised learning or unsupervised learning. Supervised learning predicts the value of an outcome measure based on a set of input and output measurements. In contrast, no output measurements exist in unsupervised learning, whose goal is to describe the relationships and patterns between a set of input measures [15]. Just as AI is a wide-ranging topic with machine learning as a sub branch, machine learning itself consists of a great number of different methods and techniques; the methods that have been used in this study are explained in detail subsequently.

### 2.2.1 Support Vector Machines

Support vector machine (SVM) is a powerful supervised learning algorithm that is easy to implement and works well in many different cases. In fact, it is considered to be one the best "off-the-shelf" algorithms for supervised learning [18]. An SVM typically comes in two different configurations: support vector classifier (SVC) and support vector regression (SVR). SVC can be used for classification and SVR can be used to predict a future value with regression. The two methods differ slightly, hence, both are explained as following based on [15] and [9]. Before explaining the characteristics of an SVM, the concept - margin needs to be introduced.

The margin can be defined as the absolute distance between the data point closest to the decision boundary and the decision boundary [18]. Essentially, the goal of an SVM is to maximize the margin between two or more classes of data [15]. Figure 2.2 shows a simple example of two hyperplanes that separate the same data but with different results.

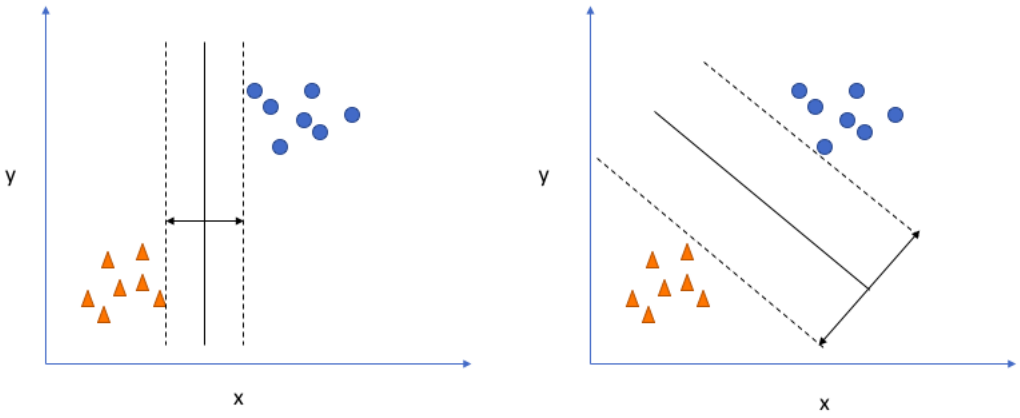


Figure 2.2: Two examples of a support vector classifier

Both hyperplanes separate the given data without any misclassifications in this case. However, it is clear that the margin is larger in the second classifier. This classifier is the one that will be chosen as a larger margin minimizes the generalization error. This example shows how the algorithm works in two dimensions. However, the principle is the same for higher dimensions as well. The mathematical formula for an SVC according to [15] can be stated as follows

Suppose the training data consists of $N$ pairs $(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)$ with $x_i \in \Re^p$ and $y_i \in \{-1, 1\}$. Then the hyperplane can be defined as

$$w^T \cdot x + b = 0, \tag{2.1}$$

Where $w$ is the vector and b is a scalar that defines the characteristics of the hyperplane. As the goal is to maximimize the margin it can be written as an optimization model with the objective function 2.2 and constraint 2.3.

$$\min \frac{1}{2}||w||^2 + C\sum_{i=1}^{N} \xi_i, \tag{2.2}$$

Subject to:

$$y_i(w^T \cdot x_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0,\ i = 1, ...., N \text{ and } C > 0, \tag{2.3}$$

Where $C$ is a regularization parameter defined by the error penalty and $\xi_i$ is a slack variable representing the distance between the decision boundary and a misclassified $x_i$. By adjusting $C$, it is possible to control the importance of a misclassification. A smaller $C$ leads to more misclassifications but, on the other hand, generates shorter training time.

The optimization problem can be solved by using Lagrange multiplier on Equation (2.2) and Equation (2.3). Sometimes, it may require to use a nonlinear separator to obtain a good result. This is possible by mapping $x_i$ into a function $\phi(x_i)$ to obtain a linearly separable hyperplane. To map a variable to another feature space, the kernel trick is applied. The Kernel trick uses a kernel function to map the variable to a desired feature space. The most common kernel function is the radial base function (RBF) which is also the one that we used in this thesis. The RBF is given by

$$K(x, x_i) = \langle \phi(x_i) \cdot \phi(x_j) \rangle = \exp\left( - \frac{||x - x_i||}{2\sigma^2} \right)^2, \tag{2.4}$$

For an SVR, the method deviates slightly, the optimization model looks as follows

$$\min \frac{1}{2}||w||^2 + C \sum_{i=1}^{N} \xi_i + \xi_i^*, \tag{2.5}$$

Subject to

$$\begin{gathered} y_i - (w^T \cdot x_i + b) \leq \varepsilon + \xi_i \\ (w^T \cdot x_i + b) \leq \varepsilon + \xi_i \\ \xi_i, \xi_i^* \geq 0, i = 1, ....N \text{ and } C > 0 \end{gathered} \tag{2.6}$$

In contrast to Equation 2.2, here, $\xi_i$ and $\varepsilon$ are the cost of an error in the prediction. The remaining variables are defined as for the SVC.

### 2.2.2 Linear Regression

Even if linear regression is an old and quite simple model, it is still widely used due to its ease of implementation and interpretation. In fact, in many cases for predictions, linear regression gives good results and outperforms fancier methods [15].

Lets assume an input vector $X^T = (X_1, X_2, ...., X_p)$ where the ouput vector $Y$ is to be predicted. Then, the linear regression model can be formulated as follows [15].

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j \tag{2.7}$$

$X_j s$ are the input variables, that is, the training data in a machine learning model and the $\beta_j s$ are the parameters to be learned by the model from some given training data $(x_1, y_1), ...., (x_N, y_N)$. $\beta_j$ can be estimated by different methods but the most common is *least squares* method [15]. The least squares method chooses the $\beta = (\beta_0, \beta_1, ..., \beta_p)^T$ to minimize the residual sum of the squares defined as

$$RSS(\beta) = \sum_{i=1}^{N} (y_i - f(x_i))^2 = \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{i_j} \beta_j)^2 \tag{2.8}$$

### 2.2.3 Cross Validation

When having limited amount of data, it is always a challenge to divide the data into training data and test data effectively. Too much training data and you risk overfitting and a high bias on the test data. Too small training set and you increase the risk of underfitting the model [9]. By diving the data into training and test data in different folds it is possible to re-use the same data, and, thus, improve the model and prevent high generalization and bias error. One way to do it is to use K-folds. The concept of K-folds is essentially to divide the data into K subsets and use K-1 subsets as training data and the remaining subset as test data [15]. This is repeated K times with alternating subsets as training and test data. Figure 2.3 shows the concept of K-folds.

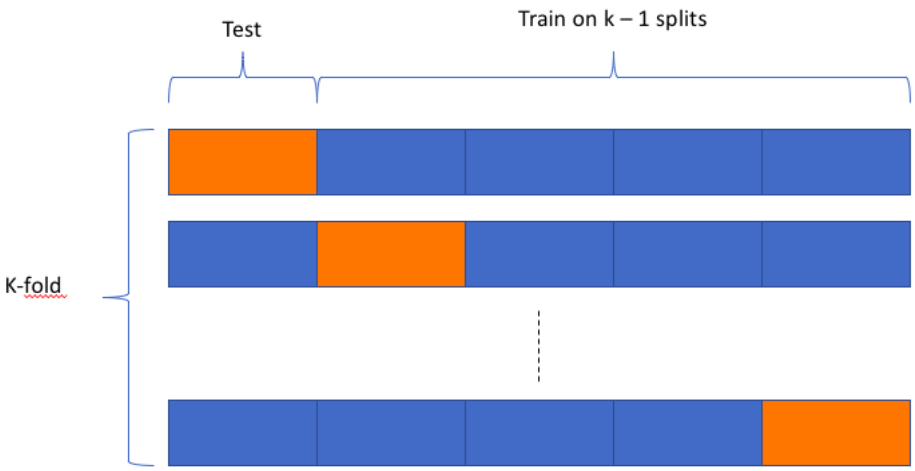


Figure 2.3: K-folds cross validation

As shown in Figure 2.3, the training data and test data are alternating for each fold but remain the same size. This allows the model to have more training data and learn more features as well as to get a more accurate error model as the model is tested on several test sets. Which also generates a lower bias error.

## 2.3 Time Series Analysis

Time series analysis covers statistical methods to analyze meaningful statistics and characteristics of data related to time [15]. Time series analysis is extensively used in financial and economical theory but can essentially be applied to any field. Typical time series may be stock prices, commodities or other assets that have a value that fluctuates over time. All further statistical methods in this section can be referred to [15] if nothing else is stated.

### 2.3.1 Autocorrelation

The basic idea with autocorrelation is to investigate if there exists any correlation between a value at time $t$ and $t+k$ [29]. This can be achieved by the autocorrelation function (ACF). According to [26] the autocorrelation function is given by

$$\hat{\rho}_k = \frac{\sum_{t=1}^{T-k}(y_t - \bar{y})(y_{t+k} - \bar{y})}{\sum_{t=1}^{T}(y_t - \bar{y})^2} \tag{2.9}$$

### 2.3.2 Partial Autocorrelation

In contrast to the autocorrelation, the partial autocorrelation function (PACF) also measures the correlation between two different time periods but also takes into account all the effects in between the different time periods [29]. For example, if $y_t$ and $y_{t+2}$ are to be compared, then $y_{t+1}$'s effect on $y_{t+2}$ will also be taken into account.

### 2.3.3 Stationarity

In order to use statistical models it may be required to make some assumptions. In time series analysis, a fundamental assumption is often that the time series which will be used for a model is stationary [15]. A stationary series is a stochastic process whose unconditional joint probability is invariant to time. Consequently, the mean, variance and covariance are also invariant to time [15]. These conditions can mathematically be formulated as follows.

$$F(y_t, ..., y_{t+k}) = F(y_{t+m}, ..., y_{t+m+k}), \tag{2.10}$$

$$\mu_y = E(y_t) = E(y_{t+k}), \tag{2.11}$$

$$\sigma_y^2 = E\Big[(y_t - \mu_y)^2\Big] = E\Big[(y_{t+m} - \mu_y)^2\Big], \tag{2.12}$$

$$\gamma_k = Cov(y_t, y_{t+k}) = Cov(y_{t+m}, y_{t+m+k}), \tag{2.13}$$

$$\forall t, k \text{ and } m$$

### 2.3.4 ARMA Models

Two popular models to represent time series are with moving average (MA) or autoregressive (AR) models. However, these models often become cumbersome as the require a high order of parameters to give a satisfactory representation of the data. To solve this problem, autoregressive moving average (ARMA) models may be used [26]. The ARMA model is a combination of the AR and MA models which allows for a smaller number of parameters for a satisfactory result. Assuming a time series is stationary and with expected value $E(y_t) = \mu$, an ARMA model of order $(p, q)$ can be represented as

$$y_t = \varphi_1 \cdot y_{t-1} + ... + \varphi_p \cdot y_{t-p} + \delta + \varepsilon_t - \theta_1 \cdot \varepsilon_{t-1} - ... - \theta_q \cdot \varepsilon_{t-q} \tag{2.14}$$

Where $(p)$ is the number of lagged parameters to include on the autoregressive part and $(q)$ is the number of lagged parameters to include from the moving average part of Equation (2.3.4).

## 2.4 Deregulated Electricity Market

In a deregulated market the price is not controlled by a central authority. The price is set by supply and demand and can be affected by factors such as weather, wind or underutilized power plants [13]. Typically, the market price is only available to retailers such as EON, Vattenfall and Fortum among others. However, smart grids make it possible for small customers to pay for electricity according to the market price on an hourly basis. For example, since October 1st 2012 in the Swedish electricity market, everyone has the right to pay for electricity by hour according to a spot market [25]. The purpose of a deregulated electricity market is to ensure electricity is produced at lowest possible cost at all time [16].

In a spot market, contracts are established between buyers and sellers for the delivery of an arbitrarily amount of electricity at a given time in the future [13].

Typically, the following day. Consequently, the market is driven by planning. A buyer, predicts how much power is needed in the future and states how much it is ready to pay for that amount of power. On the other side, the seller predicts how much power it can generate and for what price it is ready to sell it for. These information is fed into a central clearing system which sets the price [13]. Spot markets are known to be subject to sudden price spikes which may be attributed to extreme weather conditions, changes in demand, or supply among others [6]. When demand is low, cheaper power sources such as hydro or wind can provide power at a low price. When demand rises and these power sources cannot meet the demand, additional power sources need to be applied, such as gas turbines. The latter are expensive to operate which increases the price. Depending on the distribution of the power sources, different markets are sensitive to different scenarios. For instance, a power market which relies to a large extent on wind power, may experience a large increase in price when the wind speed slows down. For example, the Swedish power supply consists of 39% hydro whereas hydro only constitute for 23% of the power supply in Ontario [13][16]. These different scenarios can trigger price changes with a magnitude of several hundred per cent. For instance, in Sweden 2017 the lowest price for electricity reported was €1.59/MWh whereas the highest was €111/MWh [13]. Not only does this type of market generates sudden price increases, but also price decreases with prices falling below zero.

# Chapter 3

# Methods

*This section is intended to explain in detail to the reader how the work was carried out, and what methods were used.*

Essentially, the work was divided into four main parts. First, the data was collected from various sources and prepared for analysis. Second, a data exploration exercise was performed to explore how the characteristics of the data could be leveraged to achieve the aim of thesis. Third, forecasting of the data was performed which was used in the fourth and final step to optimize the operational cost of simulated environment consisting DCs and nodes in different geographical locations. Naturally, this section will follow the explained structure.

## 3.1 Data Collection and Preparation

Data from four different markets in four different countries was collected: New South Wales - Australia, Austria, Ontario - Canada and Sweden. Moreover, Data from these countries were retrieved partly because of data availability. But also because they can be used together to form a realistic use case scenario with DCs on both different continents with different time zones and also data centers on the same continent relatively close to each other in the same time zone. The data for New South Wales was retrieved from the Australian Energy Operator Market - AEMO [1], Ontario from IESO [16], Sweden from Nordpool [13] and Austria, from Energy Exchange Austria (EEA) [10].

Availability of historical data varied between the countries. However, because of performance reasons that are shown in Chapter 4 - Results, only data for 2017 was retrieved from all four countries. The data was downloaded in CSV format and stripped down to include only one column of hourly prices. The data also included date and time for each price but were removed because of different formats which messed up the reading of the file. Instead the dates were added in Python. The data was rather complete but in the Swedish data one extra empty line occurred at the transition from winter to summer time. This row was therefore removed. In the Austrian data, some missing prices occurred which were replaced with the most recent price. In total the data consisted of 8760 rows for each country, one for each hour for 365 days for year 2017.

The data was then imported to Python 3 which was used for both data preparation and the following steps of the method. Python was chosen because of its

convenience, nice offerings of packages for data management and visualizations like *Pandas* and *NumPy*. It may also be the "go-to" programming language for data visualization and machine learning with the support of a big community which can be leveraged to an advantage. In Python the data from all countries were prepared as Pandas dataframes in the same format with 8760 rows and three columns: date, hour and price. Furthermore, since prices appeared in different currencies they were all converted to Euro according to the exchange rate on the 10th of April 2018 from www.XE.com. A more realistic result could be achieved by using the actual exchange rates for each date. However, because of time limitation, exchange rates from the 10th of April were applied as previously explained. Noticeable, is that depending on how old the collected data is, it may be necessary to adjust the data for price inflation. This can be done by applying the consumer price index (CPI) for each country which OECD provides. Ultimately, this adjustment was not made as only data from 2017 was used. Figure 3.1 shows the ten first entries of data for Sweden after the data preparation.

| | Date | Hour | Price |
|---|---|---|---|
| 0 | 2017-01-01 | 1 | 22.394010 |
| 1 | 2017-01-01 | 2 | 22.394010 |
| 2 | 2017-01-01 | 3 | 22.384265 |
| 3 | 2017-01-01 | 4 | 21.611487 |
| 4 | 2017-01-01 | 5 | 22.459301 |
| 5 | 2017-01-01 | 6 | 23.307116 |
| 6 | 2017-01-01 | 7 | 23.875250 |
| 7 | 2017-01-01 | 8 | 24.108155 |
| 8 | 2017-01-01 | 9 | 24.397582 |
| 9 | 2017-01-01 | 10 | 23.987317 |
| 10 | 2017-01-01 | 11 | 25.059267 |

Figure 3.1: 10 first data entries for Sweden

The prices are in Euro per MWh, and this applies to all countries. Furthermore, this setup of data was ready to be used for the data exploration part.

## 3.2 Data Exploration

The goal of the data exploration was to get familiarized with the data through extracting several characteristics and statistics from the data. To achieve this goal, various visualizations were plotted from the data. To obtain visualizations the library *Matplotlib* was used, for other statistical operations such as retrieving the mean value or standard deviation using *Pandas* and *NumPy*.

## 3.3 Price Forecasting

The forecasting part accounted for a substantial part of the workload. Essentially the data was forecasted with five different methods in which the results were analyzed and compared. Four methods, were regression models and one model used classification as an initial step to predict a possible price spike and then depending

on the outcome, forecasted the price with a regression model. The approach for the five different forecasting methods are explained in detail as follows. For methods containing training and test data, the same separation of training and test data were used to ensure a fair comparison between the methods.

All forecasting methods were evaluated with the same metrics. The metrics that were used were mean square error (MSE) and mean absolute error (MAE) which can be calculated according to Equation (3.1) and Equation (3.2) accordingly. Both are widely used errors to evaluate forecast models according to [15][9][26]. For a set of data $(y_1, y_2, ..., y_n)$ and predictions $(\hat{y}_1, \hat{y}_2, ..., \hat{y}_n)$ MSE and MAE can be formulated as follows [26]

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.1}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3.2}$$

To evaluate the accuracy of the classification, the probabilities of generating a true positive and a false negative were used [21]. True positive (t/p) and false negative (f/n) are defined as follows:

$$\text{P(t/p)} = \text{P}(classified\ spike \mid spike) \tag{3.3}$$

$$\text{P(f/n)} = \text{P}(classified\ non\text{-}spike \mid spike) \tag{3.4}$$

With words, Equation (3.3) can be explained as the probability to classify a spike given a spike was observed. Likewise, Equation (3.4) the probability to classify a non-spike given a spike was observed.

### 3.3.1 Exponential Weighted Moving Average

Tyically, exponential weighted moving average (EWMA) is a smoothing technique but can also be used for forecasting [29].

EWMA was chosen as a measure of prediction as it is easy to implement and, hence, time-efficient to use as a benchmark to compare with other more time consuming methods that were used. EWMA, bases the prediction on an arbitrarily chosen number of previous values with different weights given to the previous values. The weights and number of previous values can be adjusted to obtain different results. The formula of an EWMA is given by

$$\hat{y}_{t+1} = \alpha \cdot y_t + \alpha(1-\alpha) \cdot y_{t-1} + \alpha(1-\alpha)^2 \cdot y_{t-2} + \alpha(1-\alpha)^3 \cdot y_{t-3} + ... \tag{3.5}$$

Where $\alpha$ controls the impact of previous values. The larger the $\alpha$, the larger is the weight on more recent values, which results in quicker response to rapid changes in the more recent data.

The method for the 1-hour ahead forecast is rather straight forward and was simply estimated by using Equation 3.5. Empirically the number of included parameters as well as the value of $\alpha$ were tested to find the combination which gave the best forecast.

The method for the 24-hour ahead forecast differs slightly. In this case the known value is lagged by k = 24, therefore first a forecast is made for lag k = 24. Subsequently, that forecast is used to predict the forecast for lag k = 23. This method iterates until we have a forecasted value for lag k = 1 which can then be used for predicting $\hat{y}_{t+1}$.

### 3.3.2 Linear Regression

The second forecasting method that was used is linear regression. Just as the EWMA, linear regression was implemented because of the easy and quick implementation to be able to use as a benchmark to compare with other algorithms. Another reason for using it refers to the theory about linear regression in Chapter 2 where [15] highlights that linear regression often outperforms more sophisticated algorithms.

Linear regression was implemented using *scikit-learn* - a machine learning package for Python. The algorithm was trained on a set of input data $(x_1, y_1), ...., (x_N, y_N)$ where $x_i$ is the input features and $y_i$ is the label, that is, the true price corresponding those set of features in $x_i$. By training the data, the algorithm can later be fed with only a set of $x_i$s to produce $y_i$s which is the prediction.

The prediction was made for 1-hour and 24-hours ahead forecasts. Since we want to predict a future state with based on a previous state the test input data for the 1-hour ahead forecast was shifted with one step backwards in relation to $y_i$ and for the 24-hours ahead forecast, the test input data was shifted 24 steps in relation to $y_i$.

### 3.3.3 ARMA

To find a representative model for the data to use for forecasting, Box-Jenkins method were followed according to [29]. The method consists of three steps:

1. Model identification
2. Parameter Estimation
3. Diagnosis Analysis

The first step in the model identification is to check if the data is stationary. If the data is not stationary, the data needs to be adjusted to become stationary before proceeding. As mentioned in Section 2.3.3 - stationarity, a series is stationary if the mean, variance and covariance are invariant to time. This was simply examined by plotting the data with respect to time and apply a simple linear regression based ordinary least-square. If the regression line has an angle of zero, the mean is constant. From the plotted data, it could be examined whether the variance and covariance were time invariant as well.

The next step in the model identification was to check whether there exists any serial correlation in the data. This was examined by plotting the ACF to check for any recurring correlations.

Once stationarity and seasonality were determined the next step was to estimate the order of the ARMA process. This was done by plotting the ACF and PACF. If the ACF decays rather quickly to zero in an exponential or sinusodial manner and

the PACF shows no significant spikes after lag $k = p$, the model may be represented as an ARMA($p$,0) process. On the contrary, if the ACF shows no other significant spikes after lag $k = q$ and the PACF decay rather quickly to zero exponential or sinusodial manner the model may be represented as an ARMA(0,$q$) process. If the ACF and PACF show a combination of these two events the model may be represented as an ARMA($p, q$) model [29].

Once it was determined whether the model follows an ARMA($p$,0), ARMA(0,$q$) or ARMA($p, q$) the next step was to determine the order of the parameters. This was done by selecting a range of possible lags from the ACF and PACF to test more in depth using the Akaike information criterion (AIC) and the Bayesian information criterion (BIC) [26][15][9]. These were automatically generated by a built-in time series package in Python called *statsmodels.tsa.arima_model*.

After the model was determined, the third and last step was to perform a diagnosis analysis to check whether the model is significant or not. This may be achieved in several ways with various different statistical methods with different difficulty. However, a simpler and less time consuming method is to plot the residuals between the fitted values and the real series; and was therefore used in this case. A perfect model would generate residuals of only zeros. Hence, residuals fluctuating around zero indicates a good model [26].

After the model was identified, estimated and diagnosed according to Box-Jenkins method, the model was used for forecasting for both 1 and 24 hours ahead. More in detail, the model was fed with a set of training data which the estimated model was used to forecast the next value. For the 1-hour ahead forecast, the training data was continuously updated with the next true observation until the entire test data set was consumed. For the 24-hour ahead forecast, the training data was updated continuously with the prediction as the observation. In the training data, the observations given by the predictions were exchanged to the true values 24 time units later continuously.

### 3.3.4 Support Vector Machines

The forecasting with the SVM was performed with two different methods. The first method, used SVR exclusively. In the second method however, SVC was first used to predict spike occurrence, afterwards, depending on the outcome SVR regression was performed. Both methods are described in detail subsequently.

#### Support Vector Regression

*sci-kit learn* for Python offers a built in SVR module which was used. The model was used with a set of training data $(x_1, y_1), ...., (x_N, y_N)$ where $x_i$ is the input features and $y_i$ is the label, that is, the true price. The SVR model can be tuned by selecting the kernel function (see Section 2.2.1), the constant $C$ and gamma. $C$ is the penalty parameter for the error term, gamma is the kernel choice (RBF, polynomial or sigmoid). Just like in the linear regression the input data in the test data was shifted with 1 and 24 steps for the 1 hour and 24 hours ahead forecast respectively.

### Hybrid Approach

In this method, rather than immediately predicting the price with an SVR, an SVC was first used to predict whether a price spike occurs. Depending on the outcome, an SVR was used on two different data sets, hence producing different results. All data was separated into two new data sets. One data set with all the prices that were not considered a spike, and one data set only with prices that were considered to be spikes. After the SVR was applied on the two different data sets, the results were merged into one market price forecast. Figure 3.2 shows a flowchart of the method.

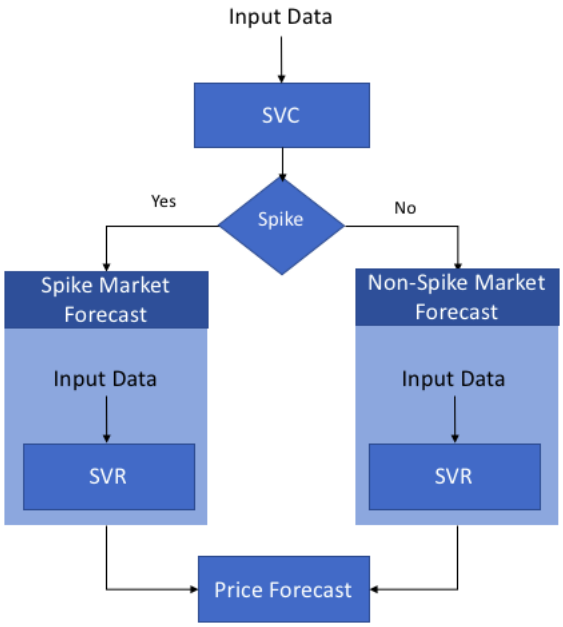


Figure 3.2: Flowchart for the hybrid SVM forecast method

By separating the data into two different datasets it was possible to extract the extreme volatility in one of the models.

Just like for the SVR, the SVC was used from a built-in function from *sci-kit learn* with the same input parameters as for the SVR. The only difference between the input data in the two different models are the labels. In the SVC, rather than using the true price as the label as for the SVR, a binary value was used as it is a classifier. 1, for a spike and 0 for non-spike. The threshold of a spike was defined as the mean plus one standard deviation of the prices in the training data set according to [21]. In [21], they define a spike as the mean plus or minus a standard deviation. However, this is not the case here as it is only desired to detect spikes that increases the price. More formally it can be formulated as

$$\text{spike} = \mu + \sigma \tag{3.6}$$

To get a more accurate prediction of the model and lower the generalization error, cross validation was used. Section 2.2.3 describes the most common method, K-fold. However, as the data in this case was a time series this is not an appropriate method to a full extent. Instead, a twist of K-fold was used. Since a model for time series forecasting relies on finding patterns and correlations with previous data it is

not appropriate to use a set of data as test data created earlier in time than the training data. Therefore, K-folds was used but with the constraint to only use test data created after the training data. The method is showed in figure 3.3.

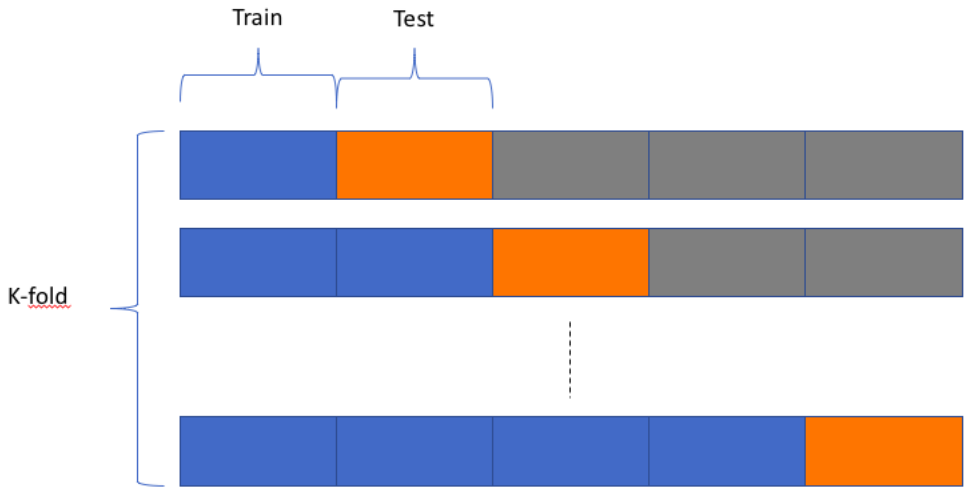


Figure 3.3: K-fold cross validation for time series

As figure 3.3 shows, it looks very similar to K-fold in figure 2.3. However, it is clear that, the test data never comes before the training data. Therefore, there are also some unused data in every fold except the last one. The grey boxes represent the unused data.

## 3.4 Optimization

The optimization was performed on two different use cases and hence two different optimization models were used. Each use case is explained in detail in two separate subsections following.

The data used in this exercise was the same price test data as used for the forecasting methods as well as the forecasts generated by the hybrid method. Since the test data was the last 20% of all data of 2017 this means that the results are for that period, which is roughly the last 2.5 months.

Both optimization problems were modelled in Python 3 and solved with Gurobi 8.0 via a Python API.

### 3.4.1 Use Case 1 - Offloading to Mobile Devices

In the first use case, a system of one data center and a number of nodes were considered. For each hour, the electricity price was assessed to investigate whether it was beneficial to offload storage to nodes. It was always cheaper, in terms of price to offload to nodes, but the data in the model is assumed to be updated frequently, for example, chat data like Messenger or Whatsapp and should therefore be stored as much as possible in the data center. Therefore, the data was only moved if a price spike occurred. In this use case, a node was considered to be a mobile device and therefore it was considered to be free to store data at the edge as it is not necessarily connected to an electricity supply and the actual price of the electricity to charge the node depends where and from which electricity provider the owner of the node uses to charge the node with. Figure 3.4 depicts the set-up of use case 1 on a map.

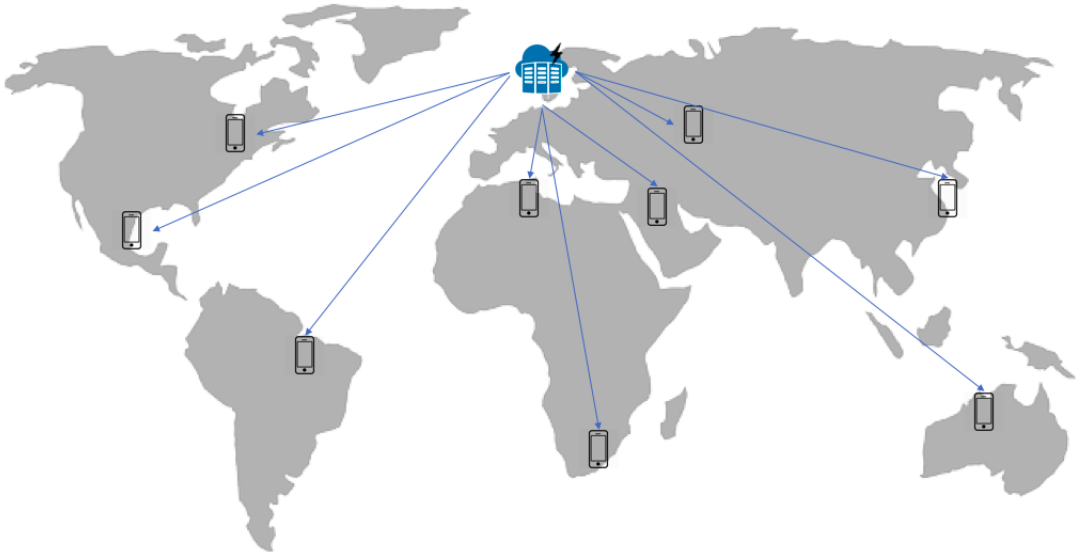

Figure 3.4: Illustration of the set-up for use case 1

The servers with a cloud around represent the data center and the mobile phones represent the nodes. The lightning icon on the data center represents the connection to a electricity supply. The arrows illustrates the capability of offloading the storage from the data center to the nodes.

Table 3.4.1 introduces the notations used in the optimization model for the first use case.

| **Notation** | **Definition** | **Type** |
|---|---|---|
| $i$ | Index for server $i$ | Index |
| $j$ | Index for node $j$ | Index |
| N | Total number of servers $i$ | Parameter |
| M | Total number of nodes $j$ | Parameter |
| $a_i$ | Capacity at server $i$ | Parameter |
| $b_i$ | Capacity at node $j$ | Parameter |
| $p_1$ | Predicted electricity price | Parameter |
| $p_2$ | True electricity price | Parameter |
| $c_j$ | Node storage cost | Parameter |
| $x_i$ | Amount of data to store on server $i$ | Decision variable |
| $y_j$ | Amount of data to store on node $j$ | Decision variable |

Table 3.1: Notations used in optimization model for use case 1

The optimization model for the first use case can be modelled as follows

P0:

$$\min \text{ w} = \sum_{i=1}^{N} x_i \cdot p_1 + \sum_{j=1}^{M} y_j \cdot c_j \tag{3.7}$$

subject to

$$x_i \leq a_i, \qquad \forall i = 1, ..., N \tag{3.8}$$

$$y_j \leq b_j, \qquad \forall j = 1, ..., M \tag{3.9}$$

$$\sum_{i=1}^{N} x_i + \sum_{j=1}^{M} y_j = \sum_{i=1}^{N} a_i \tag{3.10}$$

$$x_i, y_j \geq 0, \ \in \mathbb{Z}^+, \qquad \forall i = 1, ..., N, \forall j = 1, ..., M \tag{3.11}$$

Where $x_i$ is an integer variable which represents the amount of data to store at server $i$ and $y_j$ is the amount of data to offload to node $j$. $p$ is the electricity price, $c_j$ is the cost to store data at node j, $a_i$ the capacity of server $i$ and $b_j$ the capacity of node $j$. $c_j$ is a symbolic cost set to the threshold of a price spike in order for the algorithm to prefer storing data in the data center rather than in the edge. In the actual cost calculation, the cost is set to zero as it is the owner of the node that actually pays for the electricity.

The objective w minimizes the storage cost at every hour based on predicted prices. Constraint (3.8) makes sure that not more data than the capacity of the server can be allocated. Constraint (3.9) ensures the same but for each node. Constraint (3.10) ensures that the allocated storage distributed on both the servers in the data center and the nodes together sum up to exactly the amount of the initial capacity that was stored in the data center. The last constraint (3.11) ensures that negative storage cannot be allocated and that only integer values can be chosen on $x_i$ and $y_j$.

An algorithmic description of P0 can be explained as follows

*Step 0*: Generate electricity price forecasts using one of the methods in Chapter 3.

*Step 1*: For each hour, solve P0 with the predicted electricity prices.

*Step 2*: Calculate the cost from the given solution from P0 based on the true electricity prices.

The optimization is based on price predictions. However, no matter if the prediction is right or wrong, ultimately it is the real price that determines the actual cost. Therefore the price in the electricity price model was based on the predicted prices and the cost calculated from the true prices. The cost for each hour was calculated as follows

$$cost = \sum_{i=1}^{N} x_i \cdot p_2 \cdot E \cdot 10^{-6} \tag{3.12}$$

Where $E$ is the energy consumption in W per server per hour and $p_2$ the true electricity price. Since the prices were given in €/MWh they are transformed to €/Wh to have the same unit as $E$.

### 3.4.2 Use Case 2 - Offloading to Nodes Connected to Energy Supplies in Different Electricity Markets

In the second use case, similarly a system of one data center and a set of nodes were considered. The difference in this use case was the definition of a node. A node was considered to be a small storage unit connected to an electricity supply. Therefore, in this use case the electricity price in the different markets are also considered for which node to offload to. Similarly, the data to store was considered to be frequently updated and only offloaded in the event of a price spike occurrence. Figure 3.5 depicts the set-up of use case 2 with one data center and four smaller nodes in each of the four electricity markets.

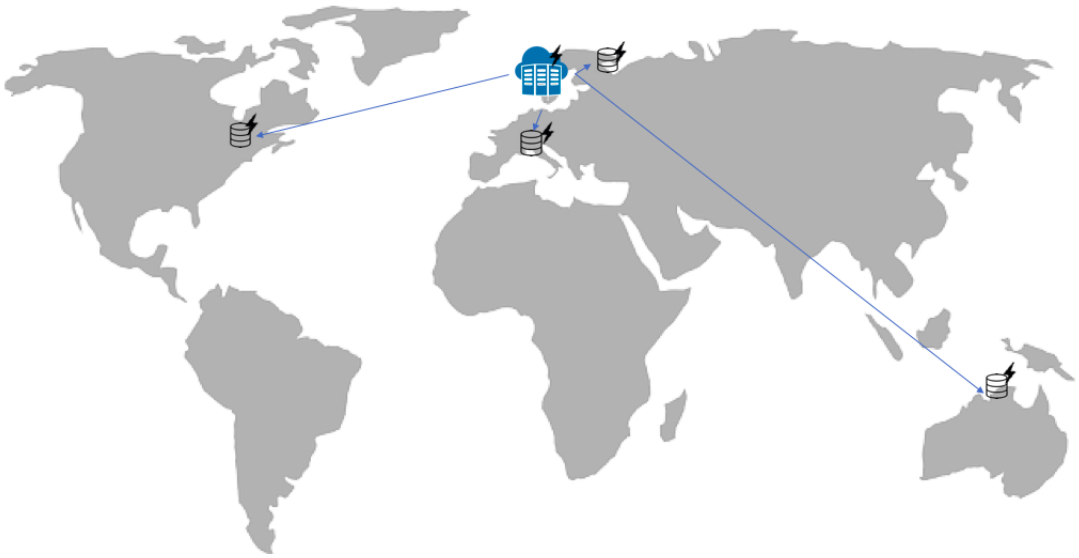

Figure 3.5: Illustration of the set-up for use case 2

The servers with a cloud around represents the data center and the smaller servers without clouds represent the nodes. The lightning icon represents that it is connected to an electricity supply. The arrows illustrates the capability of offloading the storage from the data center to the nodes.

Table 3.2 introduces the notations used in the optimization model for the second use case.

| Notation | Definition | Type |
|---|---|---|
| $i$ | Index for server $i$ | Index |
| $j$ | Index for electricity market $j$ | Index |
| M | Total number of markets $j$ | Parameter |
| N | Total number of servers $i$ for market $j$ | Parameter |
| $T$ | Total amount of data to be stored | Parameter |
| $C_{ij}$ | Capacity of server $i$ in market $j$ | Parameter |
| $p_{1j}$ | Predicted electricity price at market $j$ | Parameter |
| $p_{2j}$ | True electricity price | Parameter |
| $x_{ij}$ | Amount of data to store on server $i$ in market $j$ | Decision variable |

Table 3.2: Notations used in optimization model for use case 2

The optimization model for the second use case can be modelled as follows

P1:

$$\min \text{ w} = \sum_{i=1}^{N} \sum_{j=1}^{M} x_{ij} \cdot p_{1j} \tag{3.13}$$

subject to

$$\sum_{i=1}^{N} \sum_{j=1}^{M} x_{ij} = T \tag{3.14}$$

$$x_{ij} \leq C_{ij}, \qquad \forall i = 1, ..., N, \forall j = 1, ..., M \tag{3.15}$$

$$x_{ij} \geq 0, \in \mathbb{Z}^{+}, \qquad \forall i = 1, ..., N, \forall j = 1, ..., M \tag{3.16}$$

Where $x_{ij}$ is an integer variable which represents the amount of data to store at server $i$ in market $j$. The parameter $p_j$ is the electricity price in market $j$, $T$ is the total amount of data that needs to be stored and $C_{ij}$ is the size of each server.
The objective w in (3.13) minimizes the cost of offloaded storage with respect to the different predicted prices in the different markets for a given hour. Constraint (3.14) makes sure that the total distribution of storage equals the total amount of data needed to be stored. Constraint (3.15) ensures that the allocated storage not exceeds the capacity of server $i$ at location $j$. The last constraint, (3.16) ensures that negative storage cannot be allocated and that only integer values can be chosen on $x_{ij}$.

An algorithmic description of P1 can be defined as follows

*Step 0*: Generate electricity price forecasting with described methods in Chapter 3.

*Step 1*: For each hour, solve P1 with the predicted electricity prices.

*Step 2*: Calculate the cost from the given solution from P1 based on the true electricity prices.

The cost function for use case 2 follows the same principles as explained for use case 1 and can be calculated as

$$cost = \sum_{j=1}^{N} \sum_{j=1}^{M} x_{ij} \cdot p_{2_j} \cdot E \cdot 10^{-6} \tag{3.17}$$

# Chapter 4

# Results & Analysis

*The purpose of this chapter is to present the findings and results from the previously explained methods that have been used throughout the thesis. The chapter, follows the same structure as chapter 1.4 Methods except for the first section 3.1 Data Collection and Preparation which is excluded as it does not provide any value to this chapter.*

## 4.1 Data Exploration

Data was available at least from five years back at all locations. However, one year of data gave best results in the forecasting. Therefore, only results from data from 2017 are shown in this section if nothing else specified.

As a first step, to get an overview of the data, all data was plotted with respect to date as a time series. Figure 4.1 shows all prices for each hour during 2017 and Table 4.1 the maximum and minimum recorded prices for Ontario, Sweden, Australia and Austria respectively.

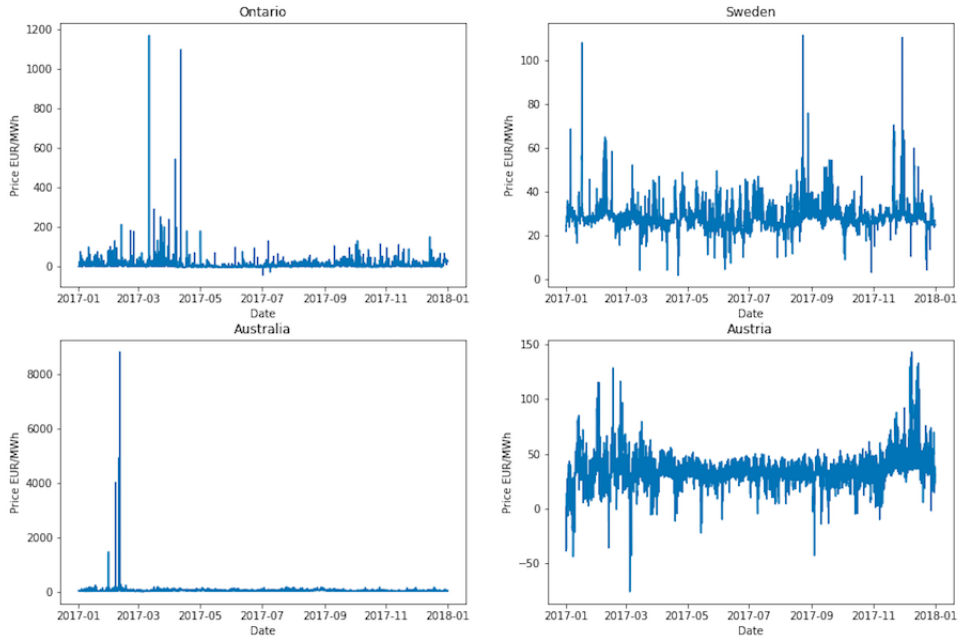


Figure 4.1: All prices for 2017 as a time series

| Market | Min | Max | Mean | Std |
|---|---|---|---|---|
| Ontario | -43.12 | 1171.80 | 9.08 | 23.3 |
| Sweden | 1.59 | 111.56 | 28.95 | 6.68 |
| Australia | 8.49 | 8809.10 | 60.47 | 150.04 |
| Austria | -76.00 | 141.30 | 34.50 | 16.28 |

Table 4.1: Extreme values in EUR

From figure 4.1 it is clear that prices in all locations have been relatively stable throughout the year. However, it is also clear that all locations suffer from extreme price spikes. This is in particular substantial to the markets in Ontario and Australia with price spikes reaching magnitudes €1171.80/MWh and €8809.10/MWh respectively. This also makes it harder to interpret these plots. For example, in the plot for Australia it looks like the price is close to zero, but referring to Table 4.1, it is clear that is not the case. The average price is around €60/MWh with a standard deviation of €150/MWh. In the data for Austria, negative prices occur several times. This is the result of over supply, further explained in Section 2.4 Deregulatad Electricity Market. It is also noticeable that the volatility is much greater in Ontario and Australia compared to Sweden and Austria. This may be due to the energy source distribution. In Sweden and Austria, large amounts of electricity is produced by more reliable energy sources like hydro and nuclear, whereas Australia and Ontario rely on other energy production sources such as wind, gas and coal [2]. For example, wind is a cheap source of energy but also very unstable. When the wind cannot deliver the predicted capacity more expensive sources like gas needs to be used to meet demand which can generate a higher price for some time [16].

In October 2017 Australia, introduced a price ceiling of €220.86. After adjusting the australian prices to the price ceiling, Figure 4.1 looks like Figure 4.2. and Table 4.1 changes to Table 4.2

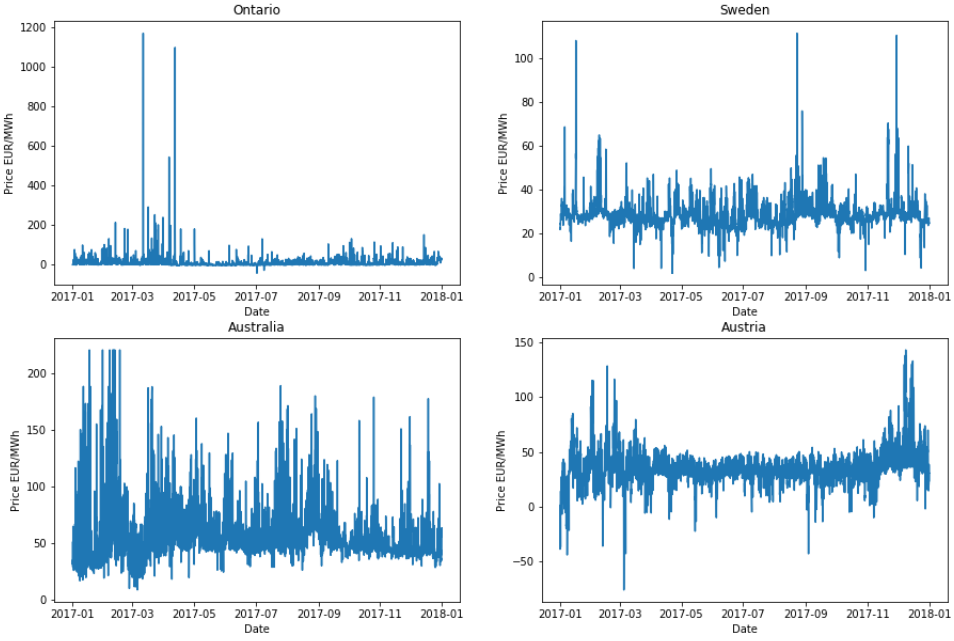


Figure 4.2: All prices for 2017 as a time series adjusted to the price ceiling

| Market | Min | Max | Mean | Std |
|---|---|---|---|---|
| Ontario | -43.12 | 1171.80 | 9.08 | 23.3 |
| Sweden | 1.59 | 111.56 | 28.95 | 6.68 |
| Australia | 8.49 | 220.86 | 57.01 | 22.38 |
| Austria | -76.00 | 141.30 | 34.50 | 16.28 |

Table 4.2: Extreme values in EUR after adjustment of price ceiling in Australia

After the adjustment it is visible that the extreme spikes have disappeared and the price distribution is more even. It is also clear, that the volatility in the Australian market is significant and seems to be larger than the other markets. In Table 4.2 the mean has decreased from 60.47 to 57.01 which should not affect the forecast significantly. However, because of the large change in the maximum value from 8809.10 to 220.86, the standard deviation has decreased from 150.04 to 22.38. This decrement should result in a higher accuracy of the forecast as the uncertainty of the data has decreased.

Figure 4.3 shows the average hourly price for 2017.

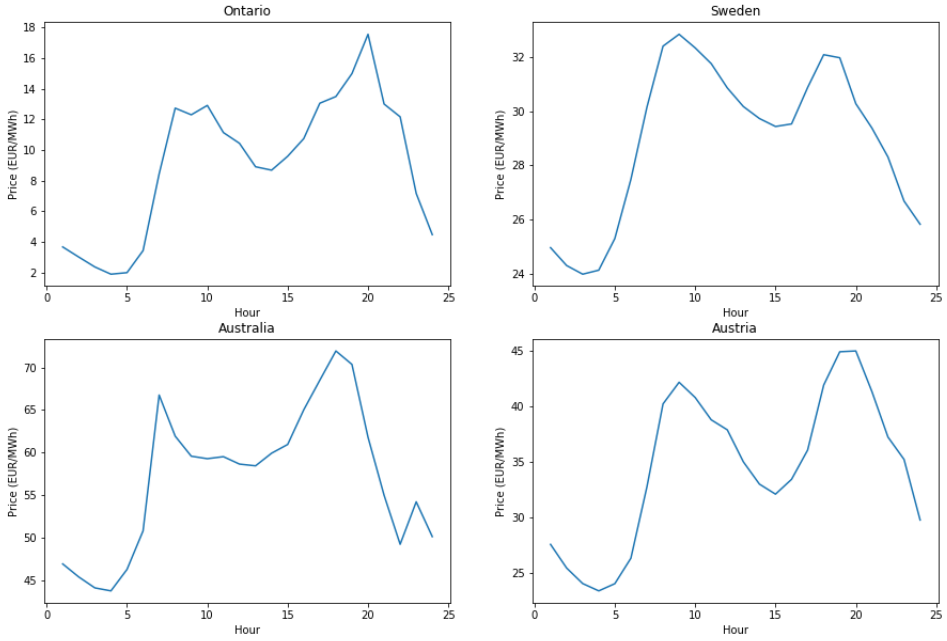


Figure 4.3: Average hourly price

All plots have in common a peak in the early morning around 10 am and a peak in the early evening around 7 to 8 pm. However, this peak occurs a bit earlier in Australia. Moreover, it is interesting that the morning peak is larger in Sweden than the evening peak since it is the opposite in the other markets. It is also noticeable that the difference in prices throughout a day is largest in Ontario, approximately the same in Australia and Austria and smallest in Sweden. This may, again be because of the energy source distribution. If the grid cannot meet the demand, it needs to use expensive energy sources to meet the customers' demand and the price increases more. Another reason could be because of the higher volatility which figure 4.2 shows with more extreme spikes in Ontario and Australia occurring in the peak hours.

Since the peak-pattern is more or less the same for the different countries and they are located in different time zones it may be possible to utilize these price variations together with the time zone differences. For example, since Ontario is five hours behind central European time (CET) it will record the lowest average hourly price throughout the day at the same time as a peak occurs in Sweden and Australia.

In figure 4.4 the prices for the local time zones are plotted in the left hand side plot. Essentially it is a combination of the subplots in figure 4.3. To visualize the effects of the time zones, the right hand side shows the prices in the different locations at UTC time zone. Table 4.3 shows the time zones used for each country.

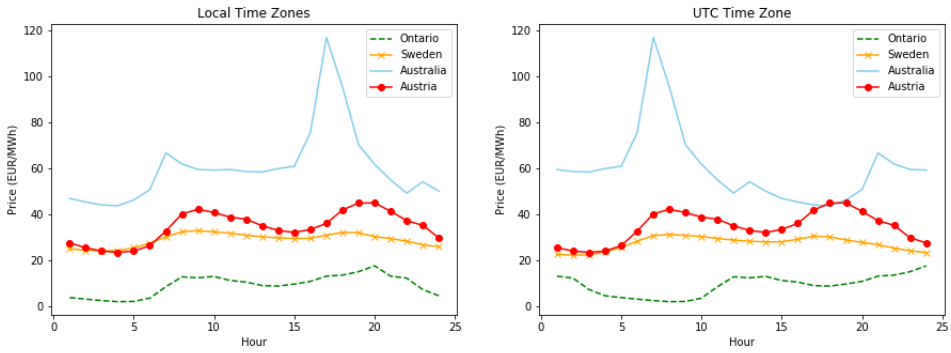


Figure 4.4: Average hourly price for local time zones and shifted time zones

| Market | Time Zone |
|---|---|
| Ontario | GMT -4 |
| Sweden | GMT +1 |
| Australia | GMT +10 |
| Austria | GMT +1 |

Table 4.3: Time zones

In the right hand side plot, Figure 4.4 shows the effect of the time zones. For example, Australia has its largest peak when Ontario has its cheapest price and when Austria and Sweden has their largest peak Australia records the lowest average hourly price. It would be beneficial to leverage the time zones where the lines cross. There are not really any clear intersections between the lines but it is worth to mention that the prices are average prices per hour throughout the year, therefore, in real life, the fluctuations are much larger and there will most likely occur several events when the lines would intersect. To illustrate this Figure 4.5 shows four different sample dates of the daily price for the different locations.

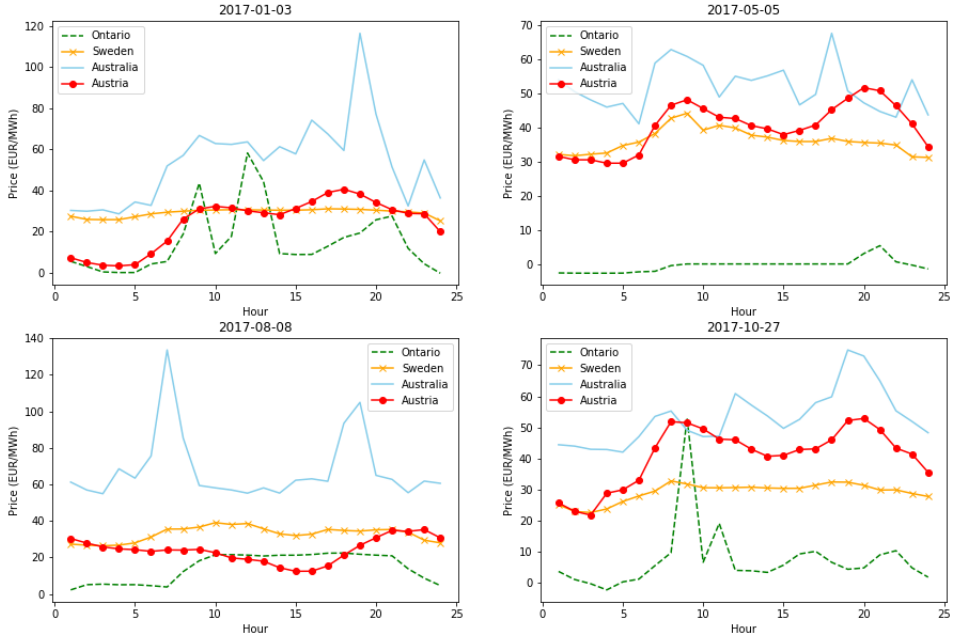


Figure 4.5: Hourly price for shifted time zones for four different days

When looking at specific price measurements rather than average price throughout an entire year there are clearly more fluctuations. Even if, for example Australia tend to always have a higher price than the other countries' lines cross each other. Therefore the plots indicate that the time zones together with the price fluctuations could potentially be used to lower the electricity cost.

A correlation analysis was performed to illustrate the impact on the price from different features as shown in Figure 4.7. To decide how many lags for the price that should have been included, the autocorrelation function was plotted for 100 lags as shown in Figure 4.6.

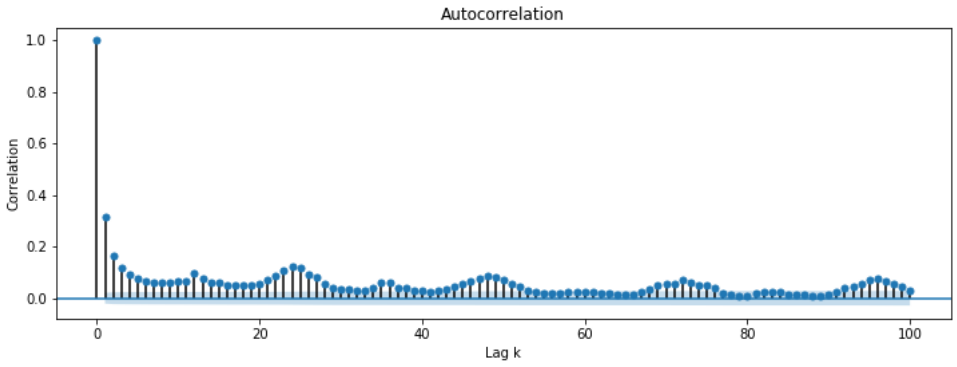


Figure 4.6: Autocorrelation function for Ontario

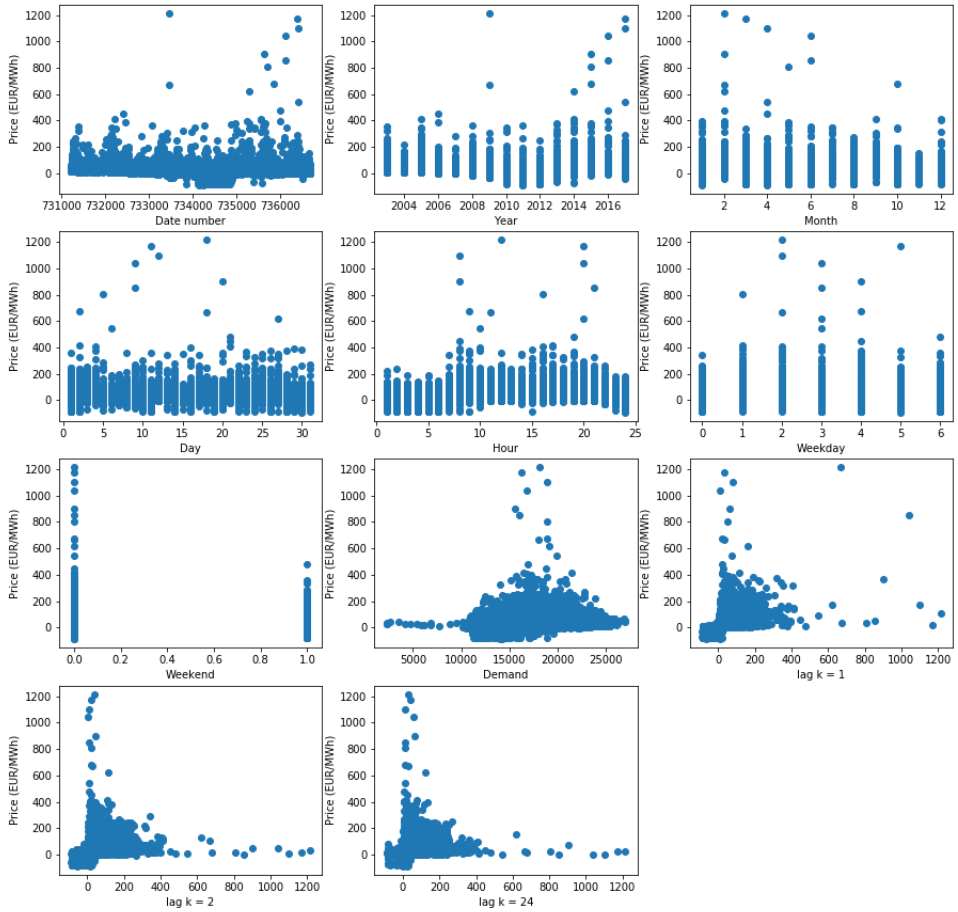


Figure 4.7: Correlation analysis for different features for Ontario

Figure 4.6 shows that the correlation is quite small between the price and different lags. Only, lag k = 1 and 2 seem to be interesting to include. We can also identify some seasonal dependency for each day, therefore lag k = 24 is also included. From 4.7 it is hard to identify any clear correlations. However, it can be extracted from the plots showing the correlation between the different lags that a high price does not tend to have had a price of similar magnitude previously. This tells, that these high prices show up unexpectedly and does not last for long. This is also something that has been shown in Figure 3.1 and Figure 4.5. Moreover, high demand also seems to be correlated to high prices.

## 4.2 Price Forecasting

Just like the different price forecasting methods were presented in separate sections in Chapter 3, the presentation of the results will follow the same structure. All different methods were tested on the same data. That is, the last 20 per cent of the

data from 2017.

Table 4.4 shows characteristics of the test data set.

| Country | Max | Min | Mean | Std | Spike threshold | Spike ratio |
|---|---|---|---|---|---|---|
| Ontario | 151.62 | -2.64 | 9.49 | 12.75 | 34.24 | 0.152 |
| Sweden | 110.58 | 2.82 | 29.29 | 6.82 | 35.51 | 0.066 |
| Australia | 179.1 | 28.27 | 49.59 | 13.59 | 42 | 0.099 |
| Austria | 143.1 | -10.03 | 43.89 | 20.24 | 46.32 | 0.12 |

Table 4.4: Test data statistics

Looking at Table 4.4, it resembles the results of the data exploration in Section 4.1. The all four data sets are quite similar with maximums and minimums significantly deviating from the mean. Sweden, seems to be the least volatile market and Ontario to be the cheapest market just like in the data exploration. Deviating from the full data sets, are Australia and Austria. The mean and standard deviation are quite different for those, which may affect the forecasting models that use training data forecast the price. The reason why the mean and std in the table do not add up is because the spike threshold is calculated from the training data set and not the test data set.

### 4.2.1 Exponential Weighted Moving Average

The EWMA was calculated for both a 1-hour and 24-hours ahead forecast. The forecast results are shown for each market in table 4.5.

| Forecast | MSE | MAE | P(t/p) | P(f/n) |
|---|---|---|---|---|
| 1 hour ahead Ontario | 0.82 | 0.4 | 0.96 | 0.04 |
| 24 hour ahead Ontario | 252.50 | 9.80 | 0 | 1 |
| 1 hour ahead Sweden | 0.96 | 0.35 | 0.95 | 0.05 |
| 24 hour ahead Sweden | 885.72 | 28.84 | 0 | 1 |
| 1 hour ahead Australia | 3.13 | 0.7 | 0.92 | 0.08 |
| 24 hour ahead Australia | 2592.16 | 48.89 | 0 | 1 |
| 1 hour ahead Austria | 1.76 | 0.58 | 0.97 | 0.03 |
| 24 hour ahead Austria | 2309.19 | 43.40 | 0 | 1 |

Table 4.5: Forecast results for EWMA

As expected, it is a big difference between the results for the 1-hour and 24-hours ahead forecast. The results vary between the countries too. Not surprisingly, the Swedish and Ontario forecast achieve the best results, most likely due to the low volatility. The reason why the difference between the 1-hour ahead and the 24-hour ahead forecasts are so big is because when using EWMA for a multi-step forecast it converges to zero. Hence, it generates a flat forecast with the value zero. Therefore, Ontario achieves the lowest MSE as it has the lowest average price. Likewise, Austria and Australia achieves the highest MSE due to higher volatility and higher average price. This is typically the problem with conventional forecasting methods. They perform very well on short-term forecasting but lack abilities to generate accurate forecasts on multi-step data.

### 4.2.2 Linear Regression

Table 4.6 shows the forecast results for the different markets with linear regression. For the one hour ahead forecast, using previous prices with lag k = 1 and k = 2 gave best results and where thus, used for all markets. For the 24-hours ahead forecast, only one feature - lag k = 24 was used.

| **Forecast** | **MSE** | **MAE** | **P(t/p)** | **P(f/n)** |
|---|---|---|---|---|
| 1 hour ahead Ontario | 104.25 | 6.54 | 0.06 | 0.94 |
| 24 hour ahead Ontario | 150.66 | 8.38 | 0 | 1 |
| 1 hour ahead Sweden | 7.97 | 1.19 | 0.81 | 0.19 |
| 24 hour ahead Sweden | 32.64 | 3.08 | 0.35 | 0.65 |
| 1 hour ahead Australia | 76.5 | 5.73 | 0.59 | 0.41 |
| 24 hour ahead Australia | 163.99 | 9.00 | 0.34 | 0.66 |
| 1 hour ahead Austria | 43.13 | 4.13 | 0.82 | 0.18 |
| 24 hour ahead Austria | 218.64 | 10.51 | 0.32 | 0.68 |

Table 4.6: Forecast results for linear regression

As Table 4.6 shows, in general linear regression forecasts with a lower accuracy, both in terms of MSE and MAE compared to EWMA. Especially significant, is the decrease in accuracy for Ontario compared to the forecast with EWMA. In terms of spike classification, the accuracy is low for Ontario. For the 24-hour forecast, linear regression does not classify a single spike correctly. However, it is important to understand that this does not mean that classification accuracy is zero on any set of data but the algorithm fails to classify spikes on this particular set of test data. On a different set of test data, the results may be different. However, for the other markets the classification results are better; especially for Sweden and Austria. The classification results are surprising for the Austrian market as the results are fairly good although a high MSE and MAE.

### 4.2.3 ARMA

Table 4.7 shows the results for a 1-hour and 24-hour forecast with an ARMA model.

| Forecast | MSE | MAE | P(t/p) | P(f/n) |
|---|---|---|---|---|
| 1 hour ahead Ontario | 52.15 | 3.94 | 0.51 | 0.49 |
| 24 hour ahead Ontario | 126.51 | 8.24 | 0.30 | 0.70 |
| 1 hour ahead Sweden | 3.98 | 1.00 | 0.80 | 0.20 |
| 24 hour ahead Sweden | 26.28 | 3.22 | 0.33 | 0.67 |
| 1 hour ahead Australia | 97.50 | 4.84 | 0.63 | 0.37 |
| 24 hour ahead Australia | 256.7 | 9.54 | 0.34 | 0.66 |
| 1 hour ahead Austria | 25.15 | 3.48 | 0.81 | 0.19 |
| 24 hour ahead Austria | 236.92 | 11.83 | 0.48 | 0.52 |

Table 4.7: Forecast results for ARMA

Looking at the results for the ARMA model in Table 4.7 the MSE and MAE show better results for Ontario and Sweden for both the 1-hour and 24-hour forecast compared to linear regression. However, ARMA performs worse for Australia and Austria compared to linear regression. Looking at the results for the classification we can see that, although the MSE and MAE is worse for Austria and Australia the classification is better for ARMA than for Linear Regression.One explanation to this can be because of the reason that ARMA classifies the spike occurrences better than linear regression but linear regression forecasts the magnitude of the values better than ARMA and hence achives lower MSE and MAE but lower accuracy for the spike classification.

Another explanation why the 1-hour forecast for Austria is better for the ARMA model compared to the linear regression may be due to the fact that the linear regression is estimated using the ordinary least square method which is very sensitive to outliers. The price spikes can be considered as outliers and therefore, those spikes may affect the overall forecast negatively.

### 4.2.4 Support Vector Machines

#### Support Vector Regression

Table 4.8 shows the forecast results with SVR. Best results were obtained with the features: past price with lag k = 1 and k = 24 for the 1-hour and 24-hour ahead forecast respectively. Hence, those features were applied for all markets. From

| Forecast | MSE | MAE | P(t/p) | P(f/n) |
|---|---|---|---|---|
| 1 hour ahead Ontario | 78.85 | 3.79 | 0.53 | 0.47 |
| 24 hour ahead Ontario | 144.97 | 7.6 | 0 | 1 |
| 1 hour ahead Sweden | 19.94 | 1.5 | 0.6 | 0.4 |
| 24 hour ahead Sweden | 35.77 | 2.96 | 0.12 | 0.88 |
| 1 hour ahead Australia | 90.3 | 4.75 | 0.59 | 0.41 |
| 24 hour ahead Australia | 138.04 | 7.05 | 0.29 | 0.71 |
| 1 hour ahead Austria | 368.74 | 9.42 | 0 | 1 |
| 24 hour ahead Austria | 414.09 | 12.7 | 0 | 1 |

Table 4.8: Forecast results for SVR

Table 4.8 we observe that the MSE is higher from the SVR for all markets except

Australia, compared to the previous methods. However, the MAE is lower for many of the markets compared to the other methods. This can be due to the fact that the algorithm fails to classify extreme values well, although it works quite well on average. This theory also reflects the results of the classification. The classification results are quite poor in comparison to the other methods. With SVR, the 24-hour ahead for Ontario, and both forecasts for Austria fail to classify a single spike correctly. Moreover, if we compare the results for the SVR and the ARMA we can observe a clearly higher MSE from the SVR whereas the MAE is more or less the same, morover, the SVR fails to classify spikes whereas ARMA still classifies decent accuracy despite the quite poor MSE and MAE. This further supports the previously suggested explanation of the results.

### Hybrid Approach

Table 4.9 shows the classification results for different features. Table 4.9 depicts that

| **Feature** | **P(t/p)** | **P(f/n)** |
|---|---|---|
| lag k = 1 | 0.78 | 0.22 |
| lag k = 24 | 0 | 1 |
| Date number | 0 | 1 |
| Month | 0.005 | 0.995 |
| Day | 0 | 1 |
| Hour | 0 | 1 |
| Weekday | 0 | 1 |
| Weekend | 0 | 1 |
| lag k = 1 & lag k = 24 | 0.14 | 0.86 |
| lag k = 1 & Date number | 0 | 1 |
| lag k = 1 & Month | 0.79 | 0.21 |
| lag k = 1 & Day | 0.67 | 0.33 |
| lag k = 1 & Hour | 0.76 | .24 |
| lag k = 1 & Weekday | 0.785 | 0.215 |
| lag k = 1 & Weekend | 0.76 | 0.24 |

Table 4.9: Classification result with different features

most of the features are redundant and can be excluded. As a single feature, one lagged value gives the best results. Adding other features to lag k = 1, downgrades the accuracy in all cases except for the month and weekday. The month as an additional feature improved the model the most and was therefore used as the final combination of features.

Table 4.10 shows the forecast results for the hybrid approach where the forecast is derived from two different data sets depending on a spike classification as an initial step described in Chapter 3.

| Forecast | MSE | MAE | P(t/p) | P(f/n) |
|---|---|---|---|---|
| 1 hour ahead Ontario | 20.23 | 0.55 | 0.35 | 0.65 |
| 24 hour ahead Ontario | 108.69 | 6.3 | 0 | 1 |
| 1 hour ahead Sweden | 5.6 | 0.33 | 0.83 | 0.17 |
| 24 hour ahead Sweden | 26.57 | 2.93 | 0.72 | 0.28 |
| 1 hour ahead Australia | 12.66 | 0.59 | 0.31 | 0.69 |
| 24 hour ahead Australia | 165.38 | 7.81 | 0.17 | 0.83 |
| 1 hour ahead Austria | 55.37 | 1.07 | 0.51 | 0.49 |
| 24 hour ahead Austria | 352.23 | 10.4 | 0.62 | 0.38 |

Table 4.10: Forecast results for the hybrid approach

Table 4.10 shows that the results improved significantly with the hybrid method to divide the data into two new data sets; one with only spikes and one with only data without spikes. By doing this, the MSE and MAE was lowered significantly on the non-spike dataset whereas the MSE and MAE was relatively high still on the spike dataset. However, since the number of spikes relative to the total number of measurements is low, this means that the overall accuracy can be improved even if the accuracy to determine the magnitude of a spike is quite low or misclassification of spikes occur. Most noticeable is the results for the 24-hour forecasts. Which, generates best results among the different methods. ARMA still performs better than the hybrid method for some markets but overall, these results show that the hybrid method with SVC and SVR are more robust tools that work better for more applications than just forecasts in very near future.

The reason why the classification results are not the same for the SVR and the hybrid method is because of the method. For the SVR, there has not been any classification applied before the SVR was applied but rather calculated afterwards from the forecast values. In contrast, in the hybrid method, an SVC was applied as an initial step to classify each value as a spike or as a non-spike as explained in Chapter 3.

In conclusion for the forecasting results, EWMA gave the best forecasts for a 1-hour ahead forecast and the hybrid method gave best forecasts for a 24-hour forecast. We can also observe that some methods work better on some markets than others. As the hybrid method gave the best results of the 24-hour forecast and second best results on the 1-hour forecast it was chosen as the most robust algorithm and was therefore used in the forecasting method in the optimization exercise. In addition, as the hybrid method is based on machine learning, there is potential that other features or adjustments could improve the accuracy of the model even further. On the contrary, the results of the EWMA and ARMA are hard to improved further as the methods do not include many variables to be varied. Rather, may extension of the methods exist with additional variables that may capture more behaviours.

## 4.3 Optimization

### 4.3.1 Use Case 1 - Offloading to Mobile Devices

Table 4.11 shows the results for the first use case with notatations that can be found in Table 3.4.1. To get a more realistic setup, the capacities for both the servers and

the nodes were randomized according to a normal distribution with different mean and standard distribution. N = 5 servers with capacities $a_i \in N(1000, 20)$ were considered. We make use of an one-and-off scheme were servers are considered being shut down in the occurence of offloading and hence, consuming zero energy. Moreover, all servers are considered to be homogeneous in that way that they consume equal amount of energy when active. The energy consumption for each server in active mode was set to 240 W/h according to [17]. 1000 nodes M with capacities $b_j \in N(5, 2)$ GB of storage each were considered. All nodes were assumed to always have maximum capacity available for offloading and in the first use case they were not connected to an energy supply and hence, the storage cost was zero to store data at the node. Moreover, the transmission cost between the data center and nodes was disregarded. The parameters were chosen with the logic that N $\gg$ M and $a_i \gg b_j$, $\forall i = 1, ...,$N and $\forall j = 1, ...,$M should be true. In addition to this, $b_j$ should be a realistic number that could apply for a example a mobile phone or some other personal device with a smaller storage space. When looking at the results it

| **Market** | **Cost Saving (EUR)** | **Cost Saving (%)** |
|---|---|---|
| Ontario | 18678.65 | 55.24 |
| Sweden | 15009.59 | 14.46 |
| Australia | 27885.39 | 15.74 |
| Austria | 34365.40 | 26.14 |

Table 4.11: Optimization results for use case 1

is important to remember that the potential saving is highly related to the number of predicted spikes. For example, the Ontario market suffered more then twice the amount of spikes as the Swedish market in the same time period. Hence, there are more opportunities in the Ontario market to offload storage and thus, reduce costs. Another factor, that affects the possible cost savings are, the mean price and standard deviation. For example a larger mean price with higher standard deviation indicates more fluctuations in the price and opportunities for more significant cost savings.

Looking at Table 4.11 it is clear that all markets have possible cost savings and that Ontario is subject for the biggest cost saving. We can see significant cost savings in all markets. Even if these results are only valid for a very simplified testing environment that in reality would suffer losses from other constraints that should be added that would affect the costs, for example communications costs, there are good opportunities for cost savings after those adjustments as well. We can also see that for example in Sweden which achieved the lowest cost reduction that, 14.46 % constitutes for more than 15000 Euro during roughly 2.5 months for a DC with only 5 servers with an approximate server size of 1000 GB. In a bigger DC with for example 5000 servers, during a whole year this cost saving would translate to 72 million Euro in cost savings. Therefore, even if in a real world scenario where the cost saving may fall to just a few per cent due to other additional constraints that are not considered here, there is still a lot of money that can be saved. The reason why the cost saving for Ontario is just slightly higher in Euro than for Sweden when the cost saving in per cent is almost four fold, is simply because the average electricity price in Sweden is higher than in Ontario. The same logic can be applied to a similar comparison of the results between Sweden and Australia - referring to

Table 4.4, the average price in the Australian market is almost twice as high as in the Swedish market.

The reasoning behind simulating a small system with five servers was because of computational limitations. Increasing the model size, greatly affects the computation time several hours per simulation. The reason why other combinations of set ups, with different capacities of the nodes, the servers or different number of nodes were not considered was simply, because it does not change the result. Since, the possible cost saving is related to the amount of storage that is possible to offload, the total summed capacity of the servers and the nodes were set to be roughly the same to reflect a scenario where all data can be offloaded in the data center. Therefore, a change of combination as for example with fewer nodes but with larger capacity, does not change the final cost saving, but just the allocation of the offloading. This is shown in Table 6.2 and Table 6.3 in Appendix A.

Table 4.12 shows the results where different node storage costs have been applied to reflect a more realistic use case where it may be more beneficial to offload storage to some nodes than other. This can for example be because of the distance between the DC and the node or connection quality.

The table shows results with three different degrees of additional node storage costs applied to each market. The different standard deviations (std) reflect different distributions of added additional node storage costs. The larger the std, the more nodes with higher additional node storage cost. The result of this additional node

| **Market** | **std** | **Cost Saving (EUR)** | **Cost Saving (%)** |
|---|---|---|---|
| Ontario | 0.1 | 17849.99 | 52.66 |
| Ontario | 0.5 | 13906.07 | 40.69 |
| Ontario | 1 | 11837.85 | 34.66 |
| Sweden | 0.1 | 11189.28 | 10.62 |
| Sweden | 0.5 | 5342.23 | 5.16 |
| Sweden | 1 | 3831.40 | 3.64 |
| Australia | 0.1 | 18590.69 | 10.57 |
| Australia | 0.5 | 9483.54 | 5.36 |
| Australia | 1 | 7845.38 | 4.44 |
| Austria | 0.1 | 28396.43 | 21.51 |
| Austria | 0.5 | 12495.68 | 9.40 |
| Austria | 1 | 8605.00 | 6.59 |

Table 4.12: Optimization results for uc1 with different costs on the nodes

storage cost is that it requires a higher price on the server side to be profitable to offload to some nodes. Therefore, with this added features there will not be as much storage from the servers allocated to the nodes as for the test used for the results in Table 4.11.

As expected, the cost savings are lower for all markets, and they also decrease when the std increases. However, the difference in the relative decrease for each new std is different between the different markets. For example, when the std increases from 0.1 to 0.5, the cost saving decreases with approximately 50 % for all markets except Ontario, where the decrease is near 20 %. A possible reason for this could be, that the spikes in the Ontario market may be more extreme and deviate more

from the spike threshold in general compared to the other markets. That would mean, that it requires larger additional node storage costs for the node to be used for offloading. This would also explain the higher cost savings for Ontario compared to the other markets in Table 4.11.

### 4.3.2 Use Case 2 - Offloading to Nodes Connected to Energy Supplies in Different Electricity Markets

In this section, results for the second use case where the offloading nodes are connected to different energy supplies in different electricity markets are presented. Table 4.13 shows the result for a scenario where each of the four markets: Ontario, Sweden, Australia and Austria have an equal normal distribution, N $\in N(50, 3)$ determining the number of nodes, a total amount of data to be stored T $\in N(1000, 20)$ and a capacity of each server in each market $C_{ij} \in N(100, 10)$. The notations referred to can be found in Table 3.2. In contrast to the results for use case 1 in Table

| **Market** | **Cost Saving (EUR)** | **Cost Saving (%)** |
|---|---|---|
| Ontario | 2065.83 | 6.36 |
| Sweden | 7481.29 | 7.51 |
| Australia | 21405.23 | 12.39 |
| Austria | 26113.53 | 19.78 |

Table 4.13: Optimization results for use case 2

4.11 where Ontario achieved the largest cost saving, Ontario achieves the lowest cost saving here. However, this makes as Ontario, has the lowest average price it means that in most cases, it is most beneficial to offload data to Ontario, i.e. not offloading at all (assuming the data center location is Ontario). Whereas, for example, Australia and Austria with higher average prices can benefit more to offload to markets with lower prices, such as Ontario and Sweden.

To create a more realistic scenario where the number of nodes may not be the same in each market. N was chosen with different distributions for the different markets as shown in Table 4.14. Set-up 1 to 4 represent scenarios where the logic is that Sweden and Austria are smaller countries compared to Ontario and Australia and hence, have less number of nodes and Ontario and Australia have more number of nodes. The difference between the number of nodes increase proportionally with the set-up as shown in Table 4.14. Set-up 5 to 8 represent a scenario where the logic is that Sweden and Ontario are the markets with the lowest average price and would thus, affect the cost savings more by having less number of nodes there.

| Setup | Ontario | Sweden | Australia | Austria |
|---|---|---|---|---|
| 1 | N(60,3) | N(40,3) | N(60,3) | N(40,3) |
| 2 | N(70,3) | N(30,3) | N(70,3) | N(30,3) |
| 3 | N(80,3) | N(20,3) | N(80,3) | N(20,3) |
| 4 | N(90,3) | N(10,3) | N(90,3) | N(10,3) |
| 5 | N(40,3) | N(40,3) | N(60,3) | N(60,3) |
| 6 | N(30,3) | N(30,3) | N(70,3) | N(70,3) |
| 7 | N(20,3) | N(20,3) | N(80,3) | N(80,3) |
| 8 | N(10,3) | N(10,3) | N(90,3) | N(90,3) |

Table 4.14: Optimization set-up for use case 2 with different number of nodes

Table 4.15 shows the results where different numbers of nodes have been used to represent the scenario where Sweden and Austria have less nodes due to smaller population and Ontario (referring to Canada) and Australia have more nodes due to greater population .

| Market | Set-up | Cost Saving (EUR) | Cost Saving (%) |
|---|---|---|---|
| Ontario | 1 | 1803.87 | 5.57 |
| Ontario | 2 | 1424.24 | 4.38 |
| Ontario | 3 | 1184.19 | 3.65 |
| Ontario | 4 | 897.26 | 2.79 |
| Sweden | 1 | 7708.77 | 7.51 |
| Sweden | 2 | 7694.66 | 7.51 |
| Sweden | 3 | 7741.19 | 7.51 |
| Sweden | 4 | 7717.76 | 7.51 |
| Australia | 1 | 21645.27 | 12.31 |
| Australia | 2 | 21693.45 | 12.20 |
| Australia | 3 | 21556.53 | 12.09 |
| Australia | 4 | 21223.62 | 12.09 |
| Austria | 1 | 25703.64 | 19.73 |
| Austria | 2 | 25409.84 | 19.55 |
| Austria | 3 | 25826.59 | 19.51 |
| Austria | 4 | 25281.39 | 19.42 |

Table 4.15: Optimization results for use case 2 with different number of nodes with set-ups 1-4

Looking at Table 4.15 we can see that the ranking of what markets achieve the highest cost saving remains the same as in Table 4.13. We can also see, if looking at set-up 1 for each market that the cost savings are lower than in Table 4.13. This is expected as the number of nodes has been decreased for Sweden and Austria. Hence, at those time points where either of these markets has been cheapest, some of the data has had been forced to be a stored in the second cheapest market. Constraining the number of nodes even more also lowers the cost savings. The reason why the effect is larger on Ontario than the other markets is because usually Ontario is the cheapest market hence, Sweden, Australia and Austria typically offload to Ontario. And since the number of nodes has only been decreased in Sweden and Austria

it does not affect the markets offloading to Ontario. In contrast, Ontario cannot offload to itself and therefore offloads to Sweden in most cases, which is typically the second cheapest market and therefore is more affected by the decrease in number of nodes in Sweden and Austria.

Looking at the results for Sweden, there is no change in results when the difference in number of nodes is increased. This is because at every time point there is a spike in Sweden, Ontario offers the cheapest price and hence, data is offloaded to Ontario. Therefore, a change in the number of nodes in Sweden and Austria does not change the result for the Swedish market.

To test these hypotheses further, another scenario where the number of nodes are decreased in the two cheapest markets and increased in the two most expensive markets was simulated.

Table 4.16 shows the results when the number of nodes are decreased in Ontario in Sweden and increased in Australia and Austria.

| **Market** | **Set-up** | **Cost Saving (EUR)** | **Cost Saving (%)** |
|---|---|---|---|
| Ontario | 5 | 1928.29 | 5.92 |
| Ontario | 6 | 1829.14 | 5.61 |
| Ontario | 7 | 1558.62 | 4.82 |
| Ontario | 8 | 821.60 | 2.52 |
| Sweden | 5 | 6725.90 | 6.55 |
| Sweden | 6 | 4791.39 | 4.71 |
| Sweden | 7 | 2758.71 | 2.66 |
| Sweden | 8 | 1623.36 | 1.55 |
| Australia | 5 | 20739.91 | 11.75 |
| Australia | 6 | 19985.12 | 11.27 |
| Australia | 7 | 18015.07 | 10.04 |
| Australia | 8 | 16418.46 | 9.22 |
| Austria | 5 | 24568.72 | 18.51 |
| Austria | 6 | 21407.20 | 16.39 |
| Austria | 7 | 18054.56 | 13.84 |
| Austria | 8 | 13872.00 | 10.65 |

Table 4.16: Optimization results for use case 2 with different number of nodes with set-ups 5-8

Comparing Table 4.16 with Table 4.15 we can see that the results for Sweden have changed. Rather than being constant as in Table 4.15 the results now decrease significantly as the change in number of nodes is increased. This indicates that, in case of a spike occurrence in Sweden, the cheapest market to offload to is Ontario, which also supports the previous hypothesis. We can also see significant changes in the Austrian market which as well indicates that the cheapest market to offload when a spike occurs is Ontario. However, looking at the Australian market, reductions can be observed, but not as significant as in the other markets. This indicates that, Australia is not as dependent on Ontario as Sweden and Austria. In the Ontario market, there are no significant changes, as it mostly depends on the Swedish market which possesses the same number of nodes as in the results for Table 4.15.

# Chapter 5

# Discussion

*This chapter analyzes and critiscises the thesis from an objective point of view as well as brings up subjective thoughts.*

It was rather surprising that in most cases adding additional features to the SVMs decreased the accuracy of the model. Typically, machine learning is known for performing good with many features in high dimensions and this applies in particular for SVMs. However, it is also suggested in [15] that simpler models with less features often perform well or even better. One way to improve the model could be to look for certain features related to the spikes. The results for the hybrid method, and in particular on the data without spikes were really good, therefore, looking at features that could lower the error for the spike data can be very beneficial.

Furthermore, it was surprising that the SVMs performed so well with just two features compared to the ARMA models which is an acknowledged algorithm popular for time series forecasting. Therefore, there is room for improvement for the SVMs by finding other features that can create added value and ultimately more accurate forecasts.

The results in this thesis are highly related to the particular data used here. Using the same methods on data other that has been used in this thesis will most likely not give the same results. However, using the same methods on the same data as used in this thesis, the results should be similar. As randomization is used in the method to some extent, the results may not be exactly the same but should be similar on average. Moreover, if using other data, it is important to take into account that the settings for the various algorithm are tailored for this exact data. For example it is not certain that the features used for this data are best for another set of data. It may also be the case that, features disregarded in this thesis may add value to another dataset and hence be relevant to include in the model.

The choice of how to set threshold for a price spike is crucial to the final results. In this thesis, the spike is set to the mean of the training data added by one standard deviation according to [21]. Changing the threshold, may change the results drastically. For example, a lower threshold would generate more spikes and thus more opportunities for offloading and ultimately higher cost savings. Whereas a higher spike would result in the opposite - less offloading opportunities and lower cost reductions. Therefore, it is important to decide what the definition of a spike is in the particular use case to achieve realistic results as that is a central part of method.

One major limitation was neglecting the consequences of moving the storage to

the node and vice versa. It does not only apply an extra cost for the routing and bandwidth but also, to make this useful in real life, the DCs must be able to move a lot of data instantaneously. This requires a very good internet connection, not only on the server side but also at the node. Even if this may be a problem today, one can argue that in future years, new technology will allow for transmission at much higher speeds than possible today. But even if high speed bandwidth is available on both server and node side, one must take into account that if the nodes are mobile phones they will move around and not always have high speed internet connection. Or even worse, no connection at all. This may be a problem, if the data is some type of data that needs to be often and instantaneously accessed. For example, chat data as considered in this thesis.

Furthermore, in this project, the effects of activating and deactivating servers were not considered. First, when shutting down a server, it takes some time and energy to re-activate the server. Second, turning the servers on and off may damage the disks in the server and affect the lifetime and hence, the investment cost of a server in the long run. In a real case scenario, these are factors that should be included. Another approach rather than the on-and-off scheme could be to just put the servers in an idle mode where the servers consume less energy but not zero. By using this method, the damage on the servers and the time to re-activate a server may not be as significant.

Most of the theoretical sources for the mathematical methods are well known books used at many universities in courses for statistics, machine learning and time series analysis. However, some central parts of methods are inspired by one particular source - Novel Hybrid Market Price Forecasting Method With Data Clustering Techniques for EV Charging Station Application written by Sarikprueck et al. [21]. That does not necessarily mean it is a weakness, however, it is discussable whether that is enough to use as a reliable source. In this case, it was considered as necessary as the reference has been cited many times and seem to be an accepted article.

# Chapter 6

# Conclusions

The aim with thesis was to investigate whether it is beneficial or not to leverage volatilities and differences in the different geographical electricity price markets to offload storage of data to reduce costs in DCs. The electricity price fluctuate greatly over time as well as between different geographical regions. By using different forecasting methods, different results can be obtained on different data. Moreover, there was no method that was best on all tests, but different methods work better on different applications and data sets. For a 1-hour ahead forecast it was possible to predict the price with a MSE and MAE of 3.98 and 1.00 respectively and for a 24-hour ahead forecast with a MSE and MAE 26.22 and 2.93 respectively. With help of these forecasts it was possible to reduce costs up to 55.24 %. Moreover, it has been shown that in a simple testing environment it is possible to significantly reduce the costs which indicates that it may be possible to reduce costs in a more advanced testing environment that resembles a real case scenario.

In the future it may be interesting to investigate a system with more parameters and constraints that can relate to a more realistic case. In addition, it would be interesting to investigate more in the forecasts. More specifically, to look at more data that could be used as features in the machine learning models or more advanced models related to time series. Examples of other data that could be used are wind power, forecasts, load forecasts, capacity, weather data among others depending on availability. Another way to improve the machine learning model and achieve better results on the spike data is to use clustering. By grouping the data into clusters may be possible to determine the magnitude of the spikes with a lower error and thus a lower overall error on the forecast.

Furthermore, in this thesis only one type of data is considered to be moved between the data center and the nodes. In future studies it would also be interesting to investigate how the allocation would change with different types of data. For example, data related to a geographical area, or data that does not need to updated so often.

# Bibliography


[1] Aemo. 2018. URL: https://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Data-dashboard#aggregated-data.

[2] International Energy Agency. 2014. URL: https://www.iea.org/publications/freepublications/publication/Austria2014.pdf.

[3] Y. Xiao. et. al. "An Energy-Efficient Data Placement Algorithm and Node Scheduling Strategies in Cloud Computing Systems". In: *Internation Conferences on Advances in Computer Science and Engineering* 2 (2013).

[4] Microsoft Azure. 2018. URL: https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/.

[5] Microsoft Azure. 2018. URL: https://azure.microsoft.com/en-us/overview/what-is-virtualization/.

[6] Christensen. "Forecasting Spikes in Electricity Prices". In: *Working paper 70* (2011).

[7] Google Cloud. 2017. URL: https://cloud.google.com/what-is-cloud-computing/.

[8] A. Deylamsalehi, P. Afsharlar, and V. M. Vokkarane. "Real-time energy price-aware anycast RWA in optical data center networks". In: *2016 International Conference on Computing, Networking and Communications (ICNC)*. Feb. 2016, pp. 1–6. DOI: 10.1109/ICCNC.2016.7440723.

[9] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley, 2000.

[10] EEA. 2018. URL: https://www.exaa.at/en/marketdata/historical-data.

[11] K. Florance. 2016. URL: https://media.netflix.com/en/company-blog/how-netflix-works-with-isps-around-the-globe-to-deliver-a-great-viewing-experience.

[12] P. A. Garcia. "Proactive Power and Thermal Aware Optimizations for Energy-Efficient Cloud Computing". PhD thesis. Technical University of Madrid, 2017.

[13] Nordpool Group. 2018. URL: https://www.nordpoolgroup.com/the-power-market/.

[14] A. Gupta et al. "Cost-efficient live VM migration based on varying electricity cost in optical cloud networks". In: *2014 IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS)*. Dec. 2014, pp. 1–3. DOI: 10.1109/ANTS.2014.7057275.

[15] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.

[16] Ieso. 2018. URL: https://www.ieso.ca.

[17] Shifeng Lin and Yinliang Xu. "Distributed coordination of internet data centers for energy cost minimization". In: *2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*. Oct. 2016, pp. 1460–1464. DOI: 10.1109/APPEEC.2016.7779732.

[18] A. Ng. *Support Vector Machines*. Tech. rep. Stanford, 2017.

[19] L. Rao et al. "Distributed Coordination of Internet Data Centers Under Multiregional Electricity Markets". In: *Proceedings of the IEEE* 100.1 (Jan. 2012), pp. 269–282. ISSN: 0018-9219. DOI: 10.1109/JPROC.2011.2161236.

[20] L. Rao et al. "Minimizing Electricity Cost: Optimization of Distributed Internet Data Centers in a Multi-Electricity-Market Environment". In: *2010 Proceedings IEEE INFOCOM*. Mar. 2010, pp. 1–9. DOI: 10.1109/INFCOM.2010.5461933.

[21] P. Sarikprueck et al. "Novel Hybrid Market Price Forecasting Method With Data Clustering Techniques for EV Charging Station Application". In: *IEEE Transactions on Industry Applications* 51.3 (May 2015), pp. 1987–1996. ISSN: 0093-9994. DOI: 10.1109/TIA.2014.2379936.

[22] Amazon Web Services. 2018. URL: https://aws.amazon.com/what-is-cloud-computing/?nc1=f_cc.

[23] P. Simon. *Too Big to Ignore: The Business Case for Big Data*. Wiley, 2013.

[24] Z. Song, X. Zhang, and C. Eriksson. "Data Center Energy and Cost Saving Evaluation". In: *ICAE2015* 75.1 (2015), pp. 1255–1260.

[25] Ny teknik. 2012. URL: https://www.nyteknik.se/energi/klart-kopa-el-per-timme-6416520.

[26] R.S. Tsay. *Analysis of Financial Time Series*. Wiley, 2010.

[27] S. Walker. *Three ways to boost datacenter utilization*. 2016. URL: https://cloudblog.ericsson.com/digital-services/3-ways-to-boost-datacenter-utilization.

[28] D. Wellers et al. *Why Machine Learning? And Why Now?* SAP. 2018.

[29] Prof. A. Werwatz. *Lecture notes in Time Series Analysis*. Dec. 2017.

# Appendix A

| Data | P(t/p) | P(f/n) |
|---|---|---|
| 2017 | 0.35 | 0.65 |
| 2016-2017 | 0.25 | 0.75 |
| 2015-2017 | 0.27 | 0.73 |
| 2014-2017 | 0.21 | 0.79 |
| 2013-2017 | 0.11 | 0.89 |
| 2012-2017 | 0.16 | 0.84 |

Table 6.1: SVC results for different amount of training data

| Set-up | No. of Servers | C Server | No. of Nodes | C Node |
|---|---|---|---|---|
| 1 | 5 | N(1000,20) | 1000 | N(5,2) |
| 2 | 3 | N(1000,20) | 1000 | N(3,2) |
| 3 | 10 | N(1000,20) | 2000 | N(5,2) |
| 4 | 5 | N(1000,20) | 2000 | N(2.5,2) |
| 5 | 5 | N(1000,20) | 500 | N(10,2) |

Table 6.2: Optimization set-ups for use case 1 with different combinations of the test environment

| Market | Set-up | Cost Saving (EUR) | Cost Saving (%) |
|---|---|---|---|
| Sweden | 1 | 15009.59 | 14.46 |
| Sweden | 2 | 9085.62 | 14.46 |
| Sweden | 3 | 29950.13 | 14.44 |
| Sweden | 4 | 14799.42 | 14.46 |
| Sweden | 5 | 14913.51 | 14.44 |

Table 6.3: Optimization results for use case 1 with different combinations of the test environment