

University of Stuttgart
Germany

University of Stuttgart
**Institute for Architecture of
Application Systems**

Service Computing

Master Thesis

Development of a web-based application for analysis
and visualization of CO₂ emissions of European power
systems

Sandesh Kenjana Ashok

Course of Study: INFOTECH

Examiner: Prof. Dr. Marco Aiello

Supervisor: Ms. Laura Fiorini

Commenced: 18/06/2020

Completed: 18/12/2020

Acknowledgement

This work is a craft and a product of six months of dedication. Being part of such a work is the best thing that happened to me in my academics and I would like to thank my supervisor Ms.Laura Fiorini for her constant support, feedback, and advice throughout this work. Also, I would like to extend my gratitude and thanks to Professor Dr. Marco Aiello for guiding me under his project. Department of service computing has supported in official work involving registering and submitting my work and I would like to thank them for making life easy.

I am indebted to the great support from my parents throughout my life and especially during my thesis days. They were constantly motivating me to accomplish my goals. Also, life at Stuttgart was joyful thanks to all of my friends and colleagues who played a larger role in making my days here memorable.

Abstract

Climate change is the foremost concern that is demanding the attention of the world to take strategic actions to combat the adverse effects. Countries around the globe are passing amendments to monitor global warming. CO₂ emissions are at alarming levels. According to the International Energy Agency, the electricity sector contributes to nearly 40% of greenhouse gas emissions. With an ever-rising demand for electricity and to meet technological innovations, production plants are producing larger amounts of electricity. For decades, fossil fuels have been the major source of electricity generation, causing higher amounts of CO₂ emissions. To take important measures in reducing the CO₂ emissions, there is a need for an accountable framework and platform that provide intuitive and critical data for CO₂ assessment. Currently, most of the existing platforms take emissions from a production plant into consideration. However, the production of electricity is to meet consumer requirements that make consumers share equal responsibility for emissions. In order to take consumers' demand into account, few methods are proposed and are currently in use. One of them is power flow tracing, a method that traces the origin of power generated at the consumer end. In this thesis, we use power flow tracing methodology to account for the CO₂ emissions due to consumption of power. This dashboard also provides users additional data associated with electricity like production values due to different generation techniques, imports and exports of the country, and the price trend of electricity. For a better assessment, support for the comparison between two countries helps in evaluating the performance of each country.

Contents

List of Tables	6
List of Figures	8
List of Algorithms	9
1 Introduction	10
1.1 Motivation	11
1.2 Problem Statement	13
1.3 Methodology	15
2 Related Work	16
2.1 Assessment of CO ₂ accounting methods	16
2.2 Available platforms	19
3 Background	25
3.1 Proportional Sharing	25
3.2 Nodal Color Mix	27
3.3 Matrix Computation	30
3.4 Test Case	35
4 Methodology and Implementation	45
4.1 Flow Chart	45
4.2 Algorithm	48
4.3 Technology stack	52
4.3.1 NumPy	52
4.3.2 Pandas	52

4.3.3	Dash	54
4.3.4	SQL	56
4.4	Implementation	57
4.4.1	Database Preparation	57
4.4.2	Dashboard implementation	60
5	Results	62
5.1	Dashboard	62
5.2	Evaluation	70
5.2.1	Larger period	70
5.2.2	Real time and Historical Data	70
5.2.3	Consumption Based Accounting	71
5.2.4	Comparison between two countries	71
6	Conclusion and Future Scope	73
6.1	Usecase	73
6.2	Limitations	74
6.3	Future Scope	76
	Bibliography	83

List of Tables

2.1	Summary of the available platforms and their features.	24
3.1	Proportional Sharing Flow Balance	27
3.2	Contribution of each node to the load	28
3.3	Generation types, values, the normalized value, Nodal color Mix of IT. All values in MWh	38
3.4	Generation types, values, the normalized value, and the con- tribution of AT to Italy's Load. All values in MWh	39
3.5	Generation types, values, the normalized value, and the con- tribution of CZ to Italy's Load. All values in MWh	40
3.6	Generation types, values, the normalized value, and the con- tribution of DE to Italy's Load. All values in MWh	41
3.7	Life cycle CO ₂ equivalent from selected electricity supply tech- nologies. Values in the table represents (gCO ₂ eq/kWh) values [1] [2]	42
3.8	Total CO ₂ emission per fuel type per country. (All values are in kgCO ₂)	43
3.9	Production-based accounting, CO ₂ emission due to generation in Italy.	44
5.1	Summary of evaluation of important features of our dashboard against existing platforms.	72

List of Figures

2.1	France Electricity and CO ₂ dashboard	20
2.2	Electricity Generation of Germany for year 2019 in Energy Charts dashboard [3]	23
2.3	Electricity exchange of Germany with its neighbours in 2019 [3]	23
3.1	Proportional Sharing Concept Illustration	26
3.2	Sample network showing principle of color mixing.	28
3.3	Color mixing inside node following proportional sharing.	29
3.4	Flows to one node to explain flow tracing [4].	30
3.5	Each row shows where the marginal electricity of that row area is generated. For example, Eastern Denmark (DK-DK2) has marginal electricity coming from local generation (the diagonal term) as well as originating from Norway (although it has no direct connection to it) and Sweden [5].	34
3.6	Test case explaining flow tracing methodology.	35
4.1	Dashboard application architecture	46
4.2	Work flow for updating database	47
4.3	Dash architecture [6]	55
4.4	Dash callback interaction with layout	56
4.5	ENTSO-E platform used as primary data source [7]	59
4.6	Source code tree of dashboard application	61
5.1	User selection for options	62
5.2	Selecting Multiple Countries for Comparison and time period	63
5.3	Production Share of different techniques and total generated value	64

5.4	Price trend showing max, min and average price for selected days	65
5.5	Total Load, Total Imports and Total Exports for the countries	66
5.6	Import statistics for the countries	66
5.7	Import to a country from different generation sources of it's neighbors	67
5.8	Export statistics for the countries	67
5.9	Import and Export statistics for the countries	68
5.10	CO ₂ emissions panel for different categories	68
5.11	CO ₂ trend of selected time frame for different categories . . .	69

List of Algorithms

1	Solving Flow Tracing in Python	48
2	Computing Diagonal Elements for Matrix A	48
3	Computing Off - Diagonal Elements for Matrix A	49
4	Computing Diagonal Elements for Matrix B	49
5	Computing nodal color mix	50
6	Computing nodal power mix	50
7	Updating the database to be in sync with dashboard	51

Chapter 1

Introduction

CO₂ emissions are the by-product of electricity generation and are the major contributor to the global greenhouse gas (GHG) emissions. GHG emissions are at an alarming level and authorities across the globe are raising concerns over climate change. Electricity production is one of the major contributors to the CO₂ emissions and it is estimated that roughly about 40% of CO₂ emission directly depends on the way electricity is produced [?]. The primary source of electricity generation is the use of fossil fuels. Nearly 80% of electricity is generated using the fossil fuels [8]. For instance, electricity generated using coal, emits about 950 grams of CO₂ for every kWh, whereas that of nuclear power plants emit 12g CO₂/kWh. The year 2018 saw the highest ever recorded CO₂ emission with global emission of 33.1GtCO₂, credits to the ever raising high energy demands. International energy agency (IEA) conducted a study and found out that CO₂ emissions due to coal alone contributed to 0.3°C of 1°C rise in global average temperature before the pre-industrial era. Strong actions were taken to combat these alarming levels of emission by many countries. European Union saw the decline of CO₂ emission by 160Mt to reach to 2.9Gt. Germany stood upfront in taking important measures which saw the emission of the country fell by 8% in 2019 mainly due to a 25% decrease in the coal-powered plants. These emission levels are last observed in 1950 in Germany [9].

The currently existing CO₂ emission accounting methodologies are production-based. These production based accounting frameworks hold producers re-

sponsible for the emissions. Here the generation of electricity from domestic usage including production and export are taken into consideration for evaluation [10]. Various carbon accounting methods are proposed in [11] which mainly emphasizes power and heat generation of the countries. These methods show the larger emission levels for the countries with heat and power plants combined.

The correct measurement of total CO₂ emissions depends on the amount of production, consumption, import, and export levels. It is due to multiple sources involved in the generation and the cumulative of all sources are either stored in the power grid or transferred across the borders for different countries. With the growing electricity demand, the proper accounting system of the CO₂ emission is of high importance to perform a quantitative measurement. There exist underlying challenges in measuring the end-to-end consumption of electricity. The process of measuring the path of generated electricity from the producer to the end-user is complex to track down. A consumption-based accounting model has been proposed for Chinese regions which evaluated the emission levels for the connected electricity network of China [12]. This approach has been extended for the European transmission network as proposed in real-time carbon accounting methodology [4]. Hence, we use the method proposed in [4] to develop an application to analyze the energy flow in interconnected large-scale electricity systems and assess the carbon emissions tied to electricity production and consumption using flow tracing methodology [13].

1.1 Motivation

Statistics of the electricity production, consumption, and associated CO₂ due to consumption are of preeminent importance in evaluating the CO₂ accountability. The data currently available are distributed over multiple platforms. It becomes troublesome to analyze the data from different places and to arrive at an outcome. There are few readily available platforms, which gives users a good representation of electricity data and CO₂ emission. But the

advantages of these platforms are masked by the price associated with it, which are tough to afford because they come with monthly subscriptions. Since there is already a toolset available in the form of flow tracing, all that lacks is free data that is available to work on.

The privatization and the complete restructuring of the electricity market brought many private players to the electricity market. The data provided by each member is of the highest importance. To provide a common ground for every member, the ENTSO-E platform has been created. This platform has become a meeting point for all other Transmission System Operator(TSO) who abide by the rules laid out by the ENTSO-E institution. This organization has set up a few objectives and principles which are respected and followed by all the TSOs. To name a few of them, objectives involve, setting up internal energy markets, optimal functionality, supporting other TSOs, adapting the functionality to meet the climate agenda, and other important goals. Amongst all the key principles, transparency stands out. Every country has agreed to share its operations data and it is made publicly available in the ENTSO-E platform [14].

Data available in the ENTSO-E platform are fed to the web-based dashboard we develop. The raw data is divided into multiple sections in the ENTSO-E transparency platform is fetched, and the same data is used in the flow tracing algorithm which is the backbone of the dashboard [15]. With the data freely available at ENTSO-E and the flow tracing as a toolset, our dashboard application, that gives useful information to the user is constructed systematically meeting all the user requirements. The idea of our dashboard is to help in enhancing the analysis of the user, or a researcher to carry out better evaluation or the implementation of the work one currently pursues. Features like downloading the graphs, data at the considered period would further benefit the work.

1.2 Problem Statement

As mentioned in the previous section, there are several platforms available that give the statistics of CO₂ emission levels. Currently, various existing platforms either provide statistics only on CO₂ accounting or just about electricity trades. The platforms like electricityMap provide CO₂ accounting for both production and consumption. However, there are not metrics or statistics shown for generated electricity for different techniques, imported and exported values. This platform provides a historical dataset for evaluation, but it comes with a license and charged based on subscription [16]. Carbon Intensity by Great Britain provides the emission levels as well as predicts the forecast for the next 96 hours [17]. Carbon Intensity platform uses the approach proposed in [18] for measuring the carbon intensity of the consumed electricity within each region of Great Britain. CO₂ accounting framework for measuring demand-side emission as proposed in [12] calculates the emission levels due to consumption of electricity for the Chinese region.

So, there exists no platform which accounts for the statistics of both real-time and historical data for a user-defined period. A tool that can be used for comparison between the two countries in the European interconnect network. Alongside comparison, there is a need for a platform that shows trends, a good and simple visualization of production, consumption, import, and export values using charts, and the summarized reports. The reports and pictures should be downloadable, an open database that would be accessed for academic research.

Therefore goal of this project is to develop a user-friendly platform that provides insights of the following aspects:

- **Energy production**
- **Energy consumption**
- **Transmission between regions**
- **Market prices**

- Statistical information of both (almost) real-time data and historical data
- Comparison between two countries based on user selection

1.3 Methodology

This web-based dashboard application gives end-user, customers, or researchers the complete account of electricity production, consumption, and CO₂ emission. The production data involves the amount of electricity generated using different generation methodologies, the price of electricity in Euros, and the forecast of electricity production for the days. Consumption accounts for the total load of the country for the period, the total amount of power imported from neighboring countries, and the total power exported to the connected countries. These metrics also involves the result of the flow tracing algorithm, using which the contribution of each generation technique involved in the imported power are traced down. For example, assume a hypothetical scenario where Austria's load of 600MWh is satisfied by the import of 125 MWh from the Czech Republic. Later, we find out the share of each generation technique that satisfied Austria's load. Once each country's respective generation technique share is calculated, we find out the total carbon emission for that technique and by summing together individual values, we get the total CO₂ emission for that country due to consumption. The dashboard also gives the user the trend for the particular period.

Chapter 2

Related Work

This chapter provides a brief overview of the available literature on energy flow analysis, frameworks available for CO₂ accounting, and the platforms which are currently available showing information about electricity production, consumption of the country, imports from neighbors, export values, price of the electricity in the market, and CO₂ emissions.

2.1 Assessment of CO₂ accounting methods

CO₂ emission occurs during the production, consumption, or transmission of electricity. However, the amount of emission significantly varies. The amount at production constitutes the level of the CO₂ emission due to the usage of fossil fuels during the production phase. Currently, available methods consider the production values for calculating the CO₂ emissions. References [19] and [20] discuss about the importance of paradigm shift from production-based to consumption-based accounting. This enables the sharing of responsibility between both consumer and a producer. In order to meet the consumer demand, production plants opt for faster mode of production by using fossil fuels and thus resulting in rise in the use of fossil fuels in electricity. To meet the demands, a larger amount of non-renewable sources are used in production techniques. Hence, the emission from the industry should be understood from a consumer perspective which will enhance the interpretation of the association between CO₂ emission and electricity consumption. The case study [21] advocates the consumption-based analysis to understand the

relationship of the life-cycle carbon emissions between the type of fuel used in electricity generation.

To allocate for the transmission loss in the network, J.Bialek proposed a method based on proportional sharing principle called Power Flow tracing (PFT), that formulates the algebraic way of solving the transmission network basing of solving the linear system of equations of power flow in the network [22]. Kirschen proposed flow tracing in parallel with Bialek as a tool set to perform analytical analysis using a graph-based approach [23]. Bialek's method states the primary assumption that the flow at each node in the network is balanced, the inflow to the node is same as outflow. This assumption is fundamental in applying the proportional sharing principle and adapting the method logy in evaluating the electricity transmission network [13]. This method was initially proposed for accounting transmission usage allocation as explained in references [24, 25]. This method gained popularity over time and gained many researchers attention that enabled it to be used for cross-border trades, and inter-transmission network of different system operators inside European transmission network [26, 27, 28].

The initially proposed flow tracing method focuses on the power flow from individual nodes and their decomposition in the network. The power flow from the node is generated using various techniques. To account for the various account and to implement this method for the larger network like European transmission network, reformulated method has been proposed where a new in-partition $q_{m,\alpha}$ representing the power injection in the node due to a generation technique α is introduced [13]. For the modification to adapt to transmission network, as proposed in the flow tracing method, it is assumed that all the power contributions to the node are mixed evenly. This assumption of even mixing of power, the balance of net inflow and outflow are fundamental for tracing method proposed by Bialek and Kirschen [22, 23]. Using the proportional sharing principle, a loss allocation illustration has been shown where two generators feed into the same line [29].

The flow tracing method follows the power flows from point of generation

to the point of consumption along with the transmission network mapping. In this method, each power flow from each generation technique from each node[country] is assigned a unique color. Though assigning a color is impossible in the real transmission network, for easier mathematical analysis, a color is assigned. This method is implemented in electricity map dashboard as explained in reference [4]. The local generation inside the node as well as the imported power is mixed evenly. The resulting color mix determines the outgoing power to the load and the exported flows. The proportional sharing concept helps in analyzing the contribution of each node to the line flow. By identifying the power contribution of each node, we go a step deeper in calculating the share of each production technique. More about the usage of proportional sharing and method to realize the concept is explained in detail in Section 3.1.

Various case studies [30, 31, 32], have been performed on evaluating the flow tracing methodology, it is concluded that the correctness of the proportional sharing principle can neither be proved nor be disproved. There are other power tracing methodologies exists and are widely implemented that are evaluated in detail in [33]. Our dashboard is based on the flow tracing technique built on proportional sharing method.

2.2 Available platforms

Many organizations and individuals are working on carbon accounting methodologies. These groups have developed their custom version of dashboards for their evaluation. Some of these dashboards are publicly available whereas few are licensed. In the section, the available platforms are listed along with a brief explanation of work. Later these dashboards are evaluated against the functionality of our dashboard.

A platform `electricitymap`, <https://www.electricitymap.org/map>, a real-time visualization of the CO₂ footprint due to electricity consumption and production [16] [4]. The platform considers the flow tracing as a toolset in calculating the CO₂ emissions due to consumption. Emissions due to production are directly associated with the generation technologies. This platform gives the statistics of production, consumption, imports to/from the country, and utilization within the country. Data shown are near real-time data along with the trends for the past hours. This is an open-source project, contributed by various developers all over the globe. The zones exchange information, carbon intensities keep changing over time, are being updated regularly. This platform also provides historical data for every country. However, that comes with a license and subscription-based service and it is expensive for academic research.

The platform `carbonintensity` <https://carbonintensity.org.uk/> provides APIs to measure the CO₂ emission level for different geographical regions of Great Britain. This platform also contains the dashboard, which provides metrics on CO₂ emissions due to different generation techniques. The forecast predicts the CO₂ intensities for the next 96 hours. The trend pattern shows the day's high and low levels of CO₂ emission. The supported APIs enable anyone to build applications for their customers to use cases. The methodology used to measure the CO₂ emissions are described in this work [17].

Platform mentioned in [21] is restricted particular to China region. In this work, carbon flow is treated as energy flow, and flows are traced to find

out the origin of the power sources. France region has a platform of its <https://www.rte-france.com/en/eco2mix>. This platform furnishes the complete data of France’s electricity usage. The production panel shows the information about all the generation methodologies and their respective amount of production. The demand of the country and the forecast are displayed in the consumption panel. The CO₂ emission of the country and its corresponding levels are shown. The cross-border electricity trading accounts for the imports and exports. This platform provides adequate information involving a systematic breakdown for each country’s contribution for a while. By default the user views the real-time data, however, the data up to 8 weeks can be viewed. The trends show comparison against neighboring countries. The platform is well organized and hands out all the required information for the evaluation. However, all the information is restricted to the France geographical location. As shown in figure 2.1, the summary panel represents all the necessary information in a concise manner [34].

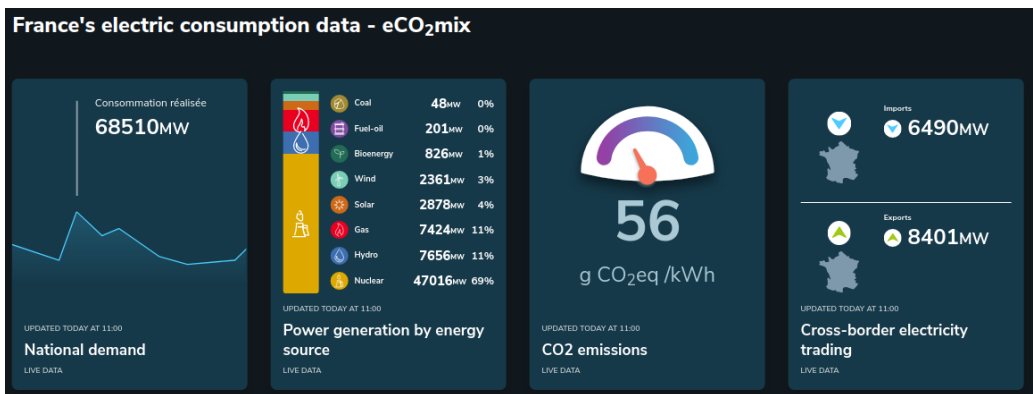


Figure 2.1: France Electricity and CO₂ dashboard

Australian Energy Market Operator, another well developed and informative dashboard representing all the information regarding electricity of the Australian energy market. This platform compares the demand and the generation of different regions, the price demand, the generation mix of different methodologies, average price information. The panel shows the real-time data for each region and the 7-day outlook for analyzing the larger time

frame [35]. However, this platform does not show any metrics accounting for CO₂ emissions. With the flow tracing algorithm available, one can develop the carbon accounting feature as we built it in this project.

Universities like Western Washington university [36] and the university of Berkeley [37] have their energy management dashboard which gives information about electricity generation at different locations within the university premise, consumption of each building, CO₂ emission associated with the electricity, the cost vs energy trend analysis. The former has a detailed representation of data using the charts based on the selected period and the location inside the campus. Supported periods are either months or year long. UC Berkeley has production and consumption metrics and building performance that shows the expected vs actual usage in the buildings.

Carbon monitor dashboard [38] aims at providing near real-time CO₂ emission data of different countries across the globe. Since the study considers the emission due to fossil fuels and usage of cement are significant contributors to the raising climate concerns. Being a recently developed platform, this gives the comparison of CO₂ emissions before and after the COVID19 pandemic. The methodology used for this work is described in this paper [39]. The idea behind this method is to account for the amount of emission due to combustion from fossil fuels and the production of cement. Hourly, monthly and yearly averages are considered in the calculation process. This accounting framework mainly considers the production techniques using fuels across various industries like electricity production, aviation, marine and road transportation, and cement production.

United States Energy Information Administration (EIA), provides tools and data sources in their dashboard. This dashboard provides complete data required for evaluation of electricity production, consumption, pricing, and CO₂ emissions associated with electricity. Data provided are per individual states considering different production techniques, transmission between states. The data shown are in real-time, and past years are considered in historical data analysis [40].

Energy charts [3] is the fully functional dashboard providing a complete breakdown of the information of Germany. Data comprises on a weekly, monthly, or yearly basis and it is up to user selection. Figure 2.2 shows energy production using different techniques in Germany for the year 2019. The charts can be viewed as both pie charts or in the form of bar charts. This platform provides information on electricity production, installed power plants, renewable share, storage data, imports and exports of Germany, power trade statistics, and emissions of the power plants. The majority of the data is pulled from the ENTSO-E platform. Cross border physical flow data is consolidated on a monthly or yearly basis. The evaluated charts can be downloaded after analyzing based on the user selection filters. However, hourly values are present only for wind speed across the country. The exchanges between neighbors coalesced for one calendar year as shown in the figure 2.3.

Summary of all the features in the dashboards, the regions they cover, the availability of data to the user are recorded in the table 2.1. As defined in Section 1.3, we need a dashboard that gives all the metrics related to electricity in one place. Production values, import, and export, CO₂ emission of the countries are visualized in the dashboard. This dashboard should be working in sync with the data provided by transmission operators, hence we need a real-time dashboard. However, for analyzing the pattern and behavior of each country, there is a need to evaluate the historical data. At last, the countries should be compared against each other to know the conduct. Above listed dashboards are evaluated against the dashboard we build to compare the features and the benefits that our dashboard additionally provides. This in-depth evaluation is under Section 5.2.

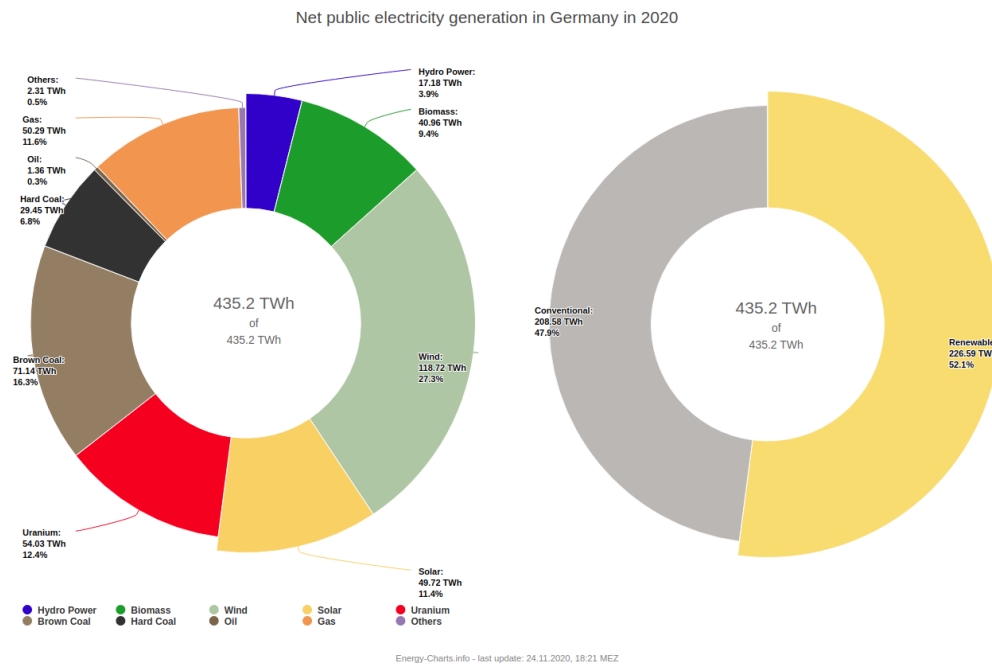


Figure 2.2: Electricity Generation of Germany for year 2019 in Energy Charts dashboard [3]

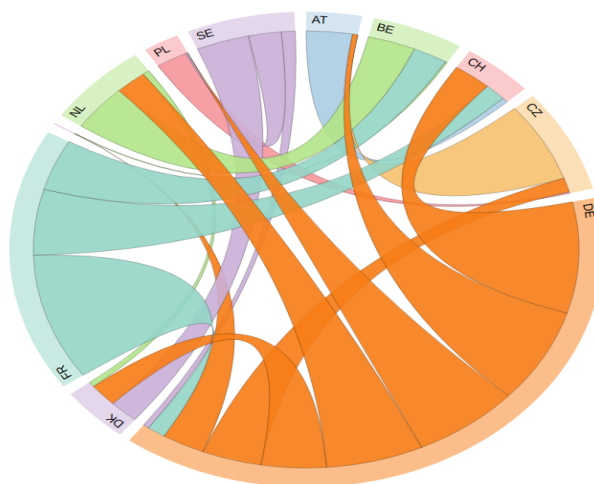


Figure 2.3: Electricity exchange of Germany with its neighbours in 2019 [3]

Platform	Region	Production, Consumption	CO₂ Emissions due to Consumption	Real Time / Historical	Pricing
Electricity Map	Global	Yes	Yes	Yes/Yes	Subscription for Historical Data
Carbon Intensity	Great Britain	No	Yes	Yes/Yes (From 2017-09-26)	Open Platform
Eco2mix	France	Yes	No	Yes/Yes (Upto 8 Weeks)	Open Platform
AEMO	Australia	Yes	No	Yes/Yes	Open Platform
Energy Manager	Western Washington University	Yes	No	No/Yes	Open Platform
Carbon Monitor	Global	No	No	No/Yes	Open Platform
US EIA	United States	Yes	No	Yes/Yes	Open Platform Still in Beta
Energy Charts	Deutschland	Yes	No	Yes/Yes	Open Platform

Table 2.1: Summary of the available platforms and their features.

Chapter 3

Background

This chapter highlights the importance of proportional sharing, a concept behind flow tracing, implementation of proportional sharing, and a test case to realize the concept. Later, the linear system of equations obtained using the flow tracing method are solved using matrices. The method of solving equations is explained in detail. Last Section 3.4 implements a sample test case involving four countries with sample generation, load, import, and export values. We use the flow tracing method to find out the CO₂ emissions of this sample network.

3.1 Proportional Sharing

The idea of proportional sharing is analogous to the flow of water in the pipe. The inflow to the junction is always balanced by the outflow. The proportional sharing aids in measuring the contribution of each flow and transmission network usage. This scheme is introduced and well illustrated by J. Bialek in his work [22]. The contribution of each line in the transmission system lies behind the idea of proportional sharing.

Consider a loss-less network as shown in figure 3.1, it has nodes with nodal inflow and outflows. Lines $L_{1,2}$ and $L_{1,3}$ are outgoing lines from node 1 to node 2 and node 3 respectively. With the assumption of a loss-less network, the inflows are proportional to the outflow from each node. Hence, this obeys Kirchhoff's current law maintaining the power balance at the node.

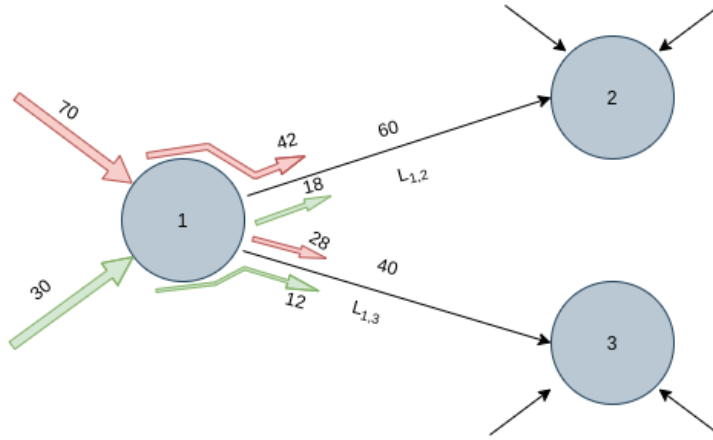


Figure 3.1: Proportional Sharing Concept Illustration

As J.Bialek explained in the work, the nodal inflows are proportionally distributed to the outgoing nodal flows [22].

By analyzing the above Figure 3.1, we see that node 1 has an inflow of 100 units, from two different sources each contributing 70 and 30 units. These inflows are proportionally distributed to node 2 and node 3 with each node getting 60 and 30 units respectively. Now, by using this principle, the contribution of each inflow to the outflows are calculated as shown below,

Contribution of 70 units of inflow at Node 1 to

- $L_{1,2}$ is $\frac{70}{100} * 60 = 42$ units
- $L_{1,3}$ is $\frac{70}{100} * 40 = 28$ units

Similarly, Contribution of 30 units of inflow at Node 1 to

- $L_{1,2}$ is $\frac{30}{100} * 60 = 18$ units
- $L_{1,3}$ is $\frac{30}{100} * 40 = 12$ units

Once the contribution of each line is known, it has to be verified that the inflow and outflow values are matches and flows are equally balanced as shown in the Table 3.1.

Nodes	Line	Line	Total Inflow = Outflow
Inflow 70 at Node 1	42	28	70
Inflow 30 at Node 1	18	12	30

Table 3.1: Proportional Sharing Flow Balance

3.2 Nodal Color Mix

In this section, the use of proportional sharing principle in the nodal color mix will be understood. For the sake of simplicity, a 5 node network is considered. The links between nodes are named in the form $L_{m,n}$, showing the connection between the node m and node n . The direction of link represents the direction of flow and the value against it is the magnitude of the flow as shown in Figure 3.2. Nodes in the green represents the positive inject of the node to the network, where as the nodes in red are consuming the flow from network.

The coloring vector is assigned to each node, showing the state of the node. For nodes 1, 2 and 5, the assigned coloring vector is $\vec{c}_1 = (1, 0, 0, 0, 0)^T$, $\vec{c}_2 = (0, 1, 0, 0, 0)^T$ and $\vec{c}_5 = (0, 0, 0, 0, 1)^T$ respectively. Since these 3 nodes does not have any ingoing links, the color of the vector will be same as the color of the node. Different colors has been assigned to evaluate the coloring vector. Blue, orange, and red are the colors for the nodes 1, 2 and 5 respectively as explained in [41]. These nodes are pure upstream nodes and have positive injections. The links represents the color of the flow from the node. $L_{1 \rightarrow 3}$, $L_{2 \rightarrow 3}$, and $L_{5 \rightarrow 4}$ hold the colors of the upstream nodes.

Now for the nodes 3 and 4, there is no own generation or positive injection and these two nodes rely on the imports from the other node to satisfy its own load. $L_{1 \rightarrow 3}$, $L_{2 \rightarrow 3}$ contribute to the node 3, hence the color mix at node 3 is the equal mix from node 1 and node 2. Therefore the coloring vector for node 3 is $\vec{c}_3 = (\frac{1}{2}, \frac{1}{2}, 0, 0, 0)^T$. As shown in Figure 3.3, the node 3 has 50%

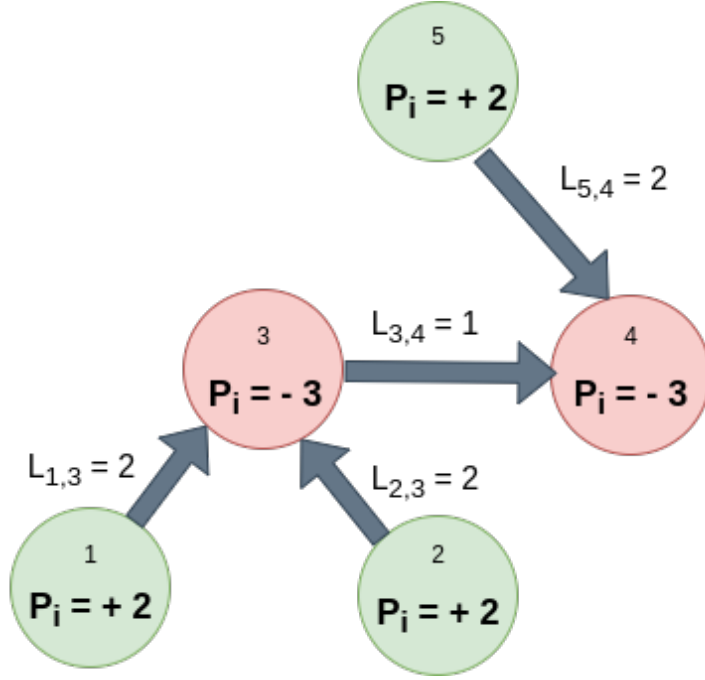


Figure 3.2: Sample network showing principle of color mixing.

blue and 50% orange. Similarly, $L_{3 \rightarrow 4}$, $L_{5 \rightarrow 4}$ contribute to the node 4. But, since node 3 does not have any contribution on its own, node 3 will share the additional imported flow from node 1 and node 3 to the $L_{3 \rightarrow 4}$. Hence the color mix at node 4 is the proportional mix from node 1, node 2 and node 5. Therefore the coloring vector for node 4 is $\vec{c}_1 = (\frac{1}{6}, \frac{1}{6}, 0, 0, \frac{2}{3})^T$.

Nodes	3	4
	Load = 3	Load = 3
1	$L_{1 \rightarrow 3}$ 2	$L_{3 \rightarrow 4}$ 0.5
2	$L_{2 \rightarrow 3}$ 2	$L_{3 \rightarrow 4}$ 0.5
5	-	$L_{5 \rightarrow 4}$ 2

Table 3.2: Contribution of each node to the load

By analyzing the color mix at node 4, we see that, though there is no direct link between node 4 and node 1, and node 4 and node 2, there is a power consumption at node 4 generated at node 1 and 2. Hence, with the proper

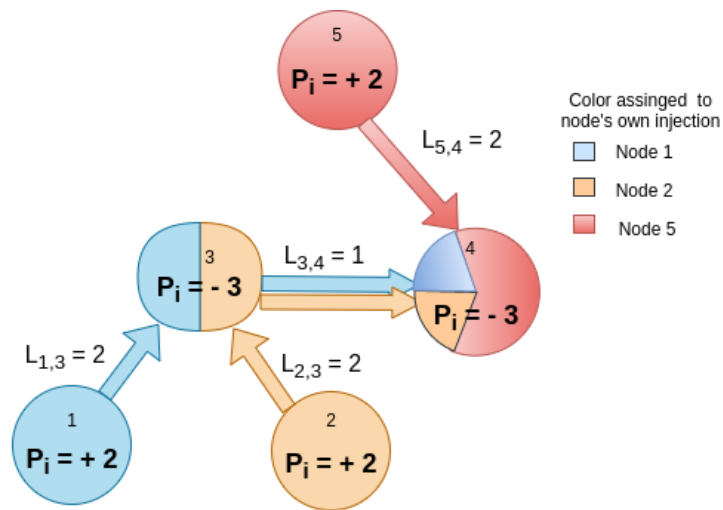


Figure 3.3: Color mixing inside node following proportional sharing.

understanding of this principle, we can trace down the contribution of each node in the network. The flows are balanced at each node and the share of generating node at a contributing node is shown in Table 3.2.

Node 4 gets 0.5 units of flow from node 1 and node 2 through node 3. Now, the generation techniques at node 1 and node 2 can be normalized to find out each contribution of each generation technique in satisfying the 3 units of load at node 4. This method and the above shown Figure 3.2 and Figure 3.3 are illustrated in the measurement of renewable power in the European electricity network using the power flow tracing method [41].

3.3 Matrix Computation

The flow tracing method is performed every hour to identify the nodal color mix at each node in the network. At each node, the generated power from different technologies, the imported power, the power due to storage discharging are mixed to obtain a color mix of the exports and the power serving the internal load of the node.

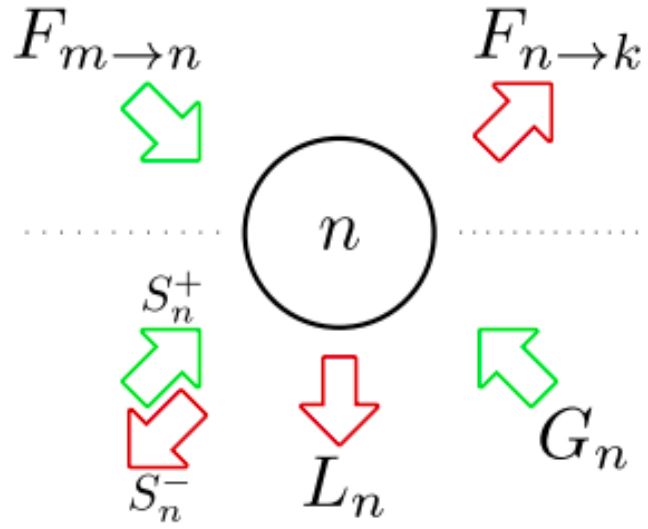


Figure 3.4: Flows to one node to explain flow tracing [4].

Figure 3.5 shows one node and considered power flows to obtain a color mix. As seen from the previous section, the proportional sharing principle will balance the power at each node. So by balancing the flows at each node, we arrive at an equation,

$$L_n + S_n^- + \sum_k F_{n \to k} = \sum_\alpha (G_{n,\alpha} + S_{n,\alpha}^+) + \sum_m F_{m \to n} \quad (3.1)$$

Equation 3.1 shows that the left-hand side(LHS) and right-hand side (RHS) are balanced for the node. LHS represents the power outflow from the node and RHS is indicated in the flow of the power at node n. This is the equation for nodal power balance. Once the power is balanced, include the nodal color mixes for every node, then we arrive at an equation,

$$q_{n,\alpha}(L_n + S_n^- + \sum_k F_{n \rightarrow k}) = G_{n,\alpha} + S_{n,\alpha}^+ + \sum_m (q_{m,\alpha} F_{m \rightarrow n}) \quad (3.2)$$

The above Equation 3.2 is an equation to find out nodal color mix per country n and per generation technology α inside the country. By rearranging the above Equation 3.2 and applying it to every node/country in the network, we arrive at a matrix formula for the entire network,

$$\sum_m [\delta_{n,m}(L_m + S_m^- + \sum_k F_{m \rightarrow k}) - F_{m \rightarrow n}] q_{m,\alpha} = G_{n,\alpha} + S_{n,\alpha} \quad (3.3)$$

where:

- α = set of all generation/storage technologies.
- L_m = nodal load.
- $G_{n,\alpha}$ = nodal generation for all technologies.
- $F_{n \rightarrow k}$ = nodal outflow to direct neighbors.
- $F_{m \rightarrow n}$ = nodal inflow from direct neighbors.
- $S_{n,\alpha}$ = storage discharge for each storage technology α at node n.
- S_m^- = sum of storage charging at node n.
- $q_{m,\alpha}$ = nodal color mix
- $\delta_{n,m}$ = kronecker delta

Solving the above equation 3.3, we arrive at a nodal color mix for every hour for all the countries with different generation techniques. Equation 3.3 is of the form $\mathbf{Ax} = \mathbf{B}$. Left-hand side corresponds to \mathbf{Ax} and right-hand side of the equation is \mathbf{B} . Here matrix A is a sparse matrix, meaning most of the elements of the matrix are zero. The diagonal elements of the matrix are the load(L) of the node also with charging storage(S) of the nodes and the exports(E) by the node to its corresponding connected countries. The non-diagonal elements are the imports to the node from its neighboring connected countries. Matrix A will be a square matrix with an equal number of rows and columns accounting for every country in the network.

$$\begin{matrix}
& AT & BE & \dots\dots & SI & SK \\
AT & \left(L_{AT} + S + E \right. & -F_{BE \rightarrow AT} & \dots\dots & -F_{SI \rightarrow AT} & -F_{SK \rightarrow AT} \\
BE & -F_{AT \rightarrow BE} & L_{BE} + S + E & \dots\dots & -F_{SI \rightarrow BE} & -F_{SK \rightarrow BE} \\
\cdot & & & \cdot & & \\
\cdot & & & \cdot & & \\
\cdot & & & \cdot & & \\
\cdot & & & \cdot & & \\
SI & -F_{AT \rightarrow SI} & -F_{BE \rightarrow SI} & \dots\dots & L_{SI} + S + E & -F_{SK \rightarrow SI} \\
SK & -F_{AT \rightarrow SK} & -F_{BE \rightarrow SK} & \dots\dots & -F_{SI \rightarrow SK} & L_{SK} + S + E \left. \right)
\end{matrix}$$

where:

- L_m = Load for the node m.
- E = Total export from the node m
- $-F_{n \rightarrow m}$ = Flow/Imports from node n to node m

Matrix B is a diagonal matrix where every diagonal entry represents the generation and discharging storage of the country as shown below.

$$B_{mn} = \begin{matrix}
& AT & BE & \dots\dots & SI & SK \\
AT & \left(G_{AT} \right. & 0 & \dots\dots & 0 & 0 \\
BE & 0 & G_{BE} & \dots\dots & 0 & 0 \\
\cdot & & & \cdot & & \\
\cdot & & & \cdot & & \\
\cdot & & & \cdot & & \\
\cdot & & & \cdot & & \\
SI & 0 & 0 & \dots\dots & G_{SI} & 0 \\
SK & 0 & 0 & \dots\dots & 0 & G_{SK} \left. \right)
\end{matrix}$$

G_m = Total generation for the node m

$$x_{mn} = \begin{matrix} & AT & BE & \dots\dots\dots & SI & SK \\ \begin{matrix} AT \\ BE \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ SI \\ SK \end{matrix} & \left(\begin{array}{ccccc} x_{AT} & x_{AT, BE} & \dots\dots\dots & x_{AT, SI} & x_{AT, SK} \\ x_{BE,AT} & x_{BE} & \dots\dots\dots & x_{BE, SI} & x_{BE, SK} \\ & & \cdot & & \\ & & \cdot & & \\ & & \cdot & & \\ & & \cdot & & \\ x_{SI,AT} & x_{SI, BE} & \dots\dots\dots & x_{SI} & x_{SI,SK} \\ x_{SK,AT} & x_{SK, BE} & \dots\dots\dots & x_{SK, SI} & x_{SK} \end{array} \right) \end{matrix}$$

where:

$x_{m,n}$ = Contribution of node n in satisfying the load at m

$x[m,n]$ is the nodal color mix shows how much of the consumption at the node m is satisfied by the node n. Contribution of each technique(α) from Node n, can be calculated by using the normalized value of each generation technique, multiplied by nodal color mix and total load of node m.

$$\frac{G_{n,\alpha}}{\sum G_{n,\alpha}} * x_{m,n} * Load_m \tag{3.4}$$

These values give the contribution of each generation technique in satisfying the load at node m. At last, we multiply this value with the associated carbon emission for this generation technique of this node n to know the intensity of CO₂ emitted due to the consumption of electricity at node m. Figure 3.5 shows the nodal color mix matrix for one hour. These equations are solved every hour to obtain the nodal color mix. The nodal color mix is a matrix, which shows the contribution of the countries in satisfying the load. The contributing countries need not be directly linked. The contribution factor is multiplied with the total load of the country and normalized generation

vector to find the nodal power mix as shown in Algorithm 6. The nodal power mix is the contribution of a generation technique(α) in satisfying the consumption node. This is completely traced down from the consumer end to the generation plant. Next Section 3.4, will show an example of calculating the power mix for a simple 4 country network.

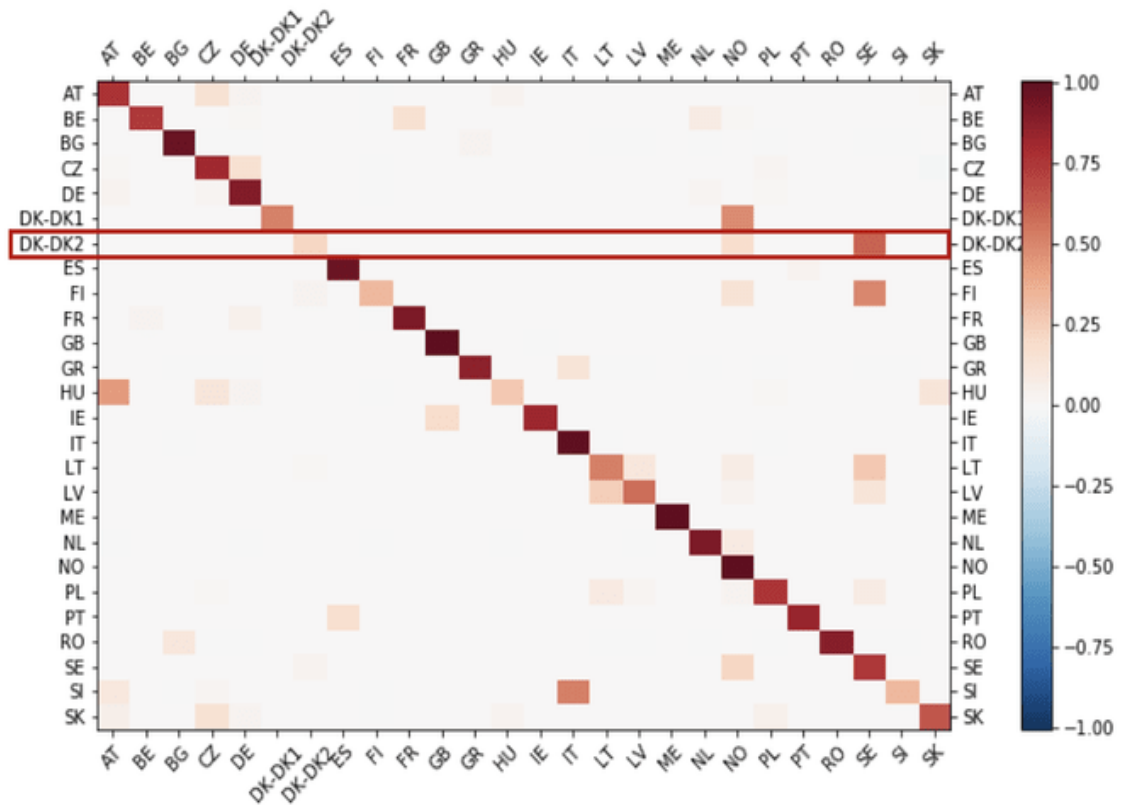


Figure 3.5: Each row shows where the marginal electricity of that row area is generated. For example, Eastern Denmark (DK-DK2) has marginal electricity coming from local generation (the diagonal term) as well as originating from Norway (although it has no direct connection to it) and Sweden [5].

3.4 Test Case

This section explains the flow tracing methodology in detail with an example. The figure 3.6 shows the interconnect of 4 countries namely Austria(AT), Czech Republic(CZ), Germany(DE) and Italy(IT). Here AT imports from CZ and DE, IT imports from AT, and CZ imports from DE. Every country has its intrinsic generation and load values. Since the focus is on consumption, we evaluate how every country's load is satisfied by its generation and the imported values.

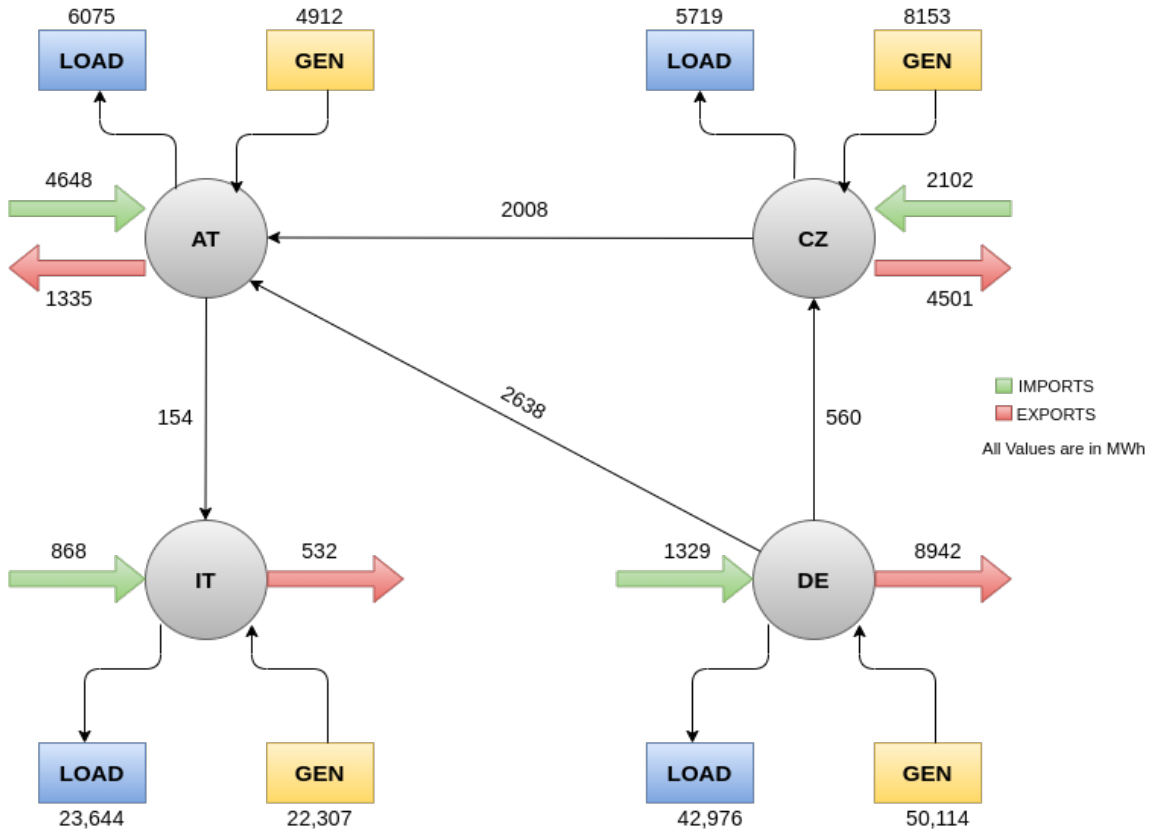


Figure 3.6: Test case explaining flow tracing methodology.

Equation 3.3 is the base for finding out the nodal color mix. As per the equation, we solve the matrix and find out the nodal color mix matrix showing each country's contribution. As explained in the previous section, the

equation is simplified to the $\mathbf{Ax} = \mathbf{B}$ form, where both A and B are square matrices. Matrix A represents the load, storage charging at a node, imports, exports of each country and Matrix B constitutes of total generation for each country. Storage charging values for AT, CZ, DE, IT are 1621, 0, 876, 1 respectively.

By referring to the above figure 3.6, we construct matrix A. The diagonal elements of the matrix represent the total load of the country, total exports, and total storage value due to charging at node n as seen in equation 3.3. The off-diagonal elements are the imports from the respective neighbor and are negated indicating the inflow to the country.

$$A_{4,4} = \begin{matrix} & AT & CZ & DE & IT \\ \begin{matrix} AT \\ CZ \\ DE \\ IT \end{matrix} & \left(\begin{array}{cccc} 6075 + 1335 - 1621 & -2008 & -2638 & 0 \\ 0 & 5719 + 4501 + 0 & -560 & 0 \\ 0 & 0 & 42976 + 8942 - 876 & 0 \\ -154 & 0 & 0 & 23644 + 532 + 1 \end{array} \right) \end{matrix}$$

Sum of the diagonal elements results in a simplified matrix as shown below

$$A_{4,4} = \begin{matrix} & AT & CZ & DE & IT \\ \begin{matrix} AT \\ CZ \\ DE \\ IT \end{matrix} & \left(\begin{array}{cccc} 9031 & -2008 & -2638 & 0 \\ 0 & 10220 & -560 & 0 \\ 0 & 0 & 52794 & 0 \\ -154 & 0 & 0 & 24177 \end{array} \right) \end{matrix}$$

The matrix B is a diagonal matrix that has total generation values for every country as diagonal elements. For the test case that is under evaluation, the constructed matrix B is as shown below

$$B_{4,4} = \begin{matrix} & AT & CZ & DE & IT \\ \begin{matrix} AT \\ CZ \\ DE \\ IT \end{matrix} & \begin{pmatrix} 4912 & 0 & 0 & 0 \\ 0 & 8153 & 0 & 0 \\ 0 & 0 & 50114 & 0 \\ 0 & 0 & 0 & 22307 \end{pmatrix} \end{matrix}$$

Solve the matrix for equation $X = A^{-1}.B$. These matrices are implemented in python using panda dataframes, once they are constructed, the matrices are converted into NumPy arrays. These NumPy arrays are solved using a **linalg.splu** utility from the SciPy package.

$$[X] = \begin{matrix} & AT & CZ & DE & IT \\ \begin{matrix} AT \\ CZ \\ DE \\ IT \end{matrix} & \begin{pmatrix} 0.5439 & 0.1773 & 0.2888 & 0 \\ 0 & 0.79775 & 0.05201 & 0 \\ 0 & 0 & 0.94923 & 0 \\ 0.0034645 & 0.0012982 & 0.0018398 & 0.92265 \end{pmatrix} \end{matrix}$$

By analyzing the above nodal color mix matrix, one can understand that, even though there is no direct connection between IT and DE, DE has contributed some amount in satisfying the IT's load. This is because of the large amounts of imports at AT from DE. As learned before, $x[n,m]$ shows how much of the consumption at the node n is satisfied by the node m. In this case, $x[IT, DE]$ is 0.001839, which means that 0.18% of IT's loads are satisfied by the electricity generated at DE. Now, in order to determine the nodal power mix, calculate the norm of the generation at DE as shown below in Table 3.4 - 3.6.

Once the nodal power mix for every generation technique of each contributing country is calculated, multiple the life cycle CO₂ equivalent value for each technique. In this case, values provided by working group III to the Fifth Assessment Report of the IPCC [1] is used as shown in Table 3.7.

Fuel Type	Value	Normalised Value	Contribution (0.92265)
Biomass	495	0.0221903	484.08442
Fossil Coal-derived gas	776	0.0347872	758.88751
Fossil Gas	8053	0.3610077	7875.4322
Fossil Hard coal	1938	0.086878	1895.2554
Fossil Oil	8	0.0003586	7.822907
Geothermal	674	0.030214	659.12253
Hydro Run-of-river and poundage	1891	0.0847715	1849.3018
Hydro Water Reservoir	289	0.0129555	282.62600
Other	3070	0.1376249	3002.3059
Waste	37	0.0016586	36.182585
Wind Onshore	5076	0.2275518	4964.0736
Total	22307	-	-

Table 3.3: Generation types, values, the normalized value, Nodal color Mix of IT. All values in MWh

Fuel Type	Value	Normalised Value	Contribution (0.0034645)
Biomass	304	0.0618892	5.06887
Fossil Brown coal/Lignite	-	-	-
Fossil Gas	1269.75	0.2584995	21.1741049
Fossil Hard coal	155.5	0.0316571	2.5931704
Fossil Oil	0	0	0
Geothermal	0	0	0
Hydro Run-of-river and poundage	2693.75	0.5484018	44.921966
Hydro Water Reservoir	122	0.0248371	2.034512
Nuclear	-	-	-
Other	22	0.0044788	0.366813
Other renewable	-	-	-
Waste	100	0.0203583	1.66761742
Wind Offshore	-	-	-
Wind Onshore	245	0.0498778	4.08572
Total	4912	-	-

Table 3.4: Generation types, values, the normalized value, and the contribution of AT to Italy's Load. All values in MWh

Fuel Type	Value	Normalised Value	Contribution (0.0012982)
Biomass	251	0.0307862	0.9449639
Fossil Brown coal/Lignite	3175	0.3894272	11.95331
Fossil Coal-derived gas	185	0.0226910	0.6964911
Fossil Gas	241	0.0295596	0.907320
Fossil Hard coal	234	0.0287010	0.8809657
Fossil Oil	3	0.0003679	0.0112925
Hydro Run-of-river and poundage	51	0.0062553	0.1920039
Hydro Water Reservoir	23	0.0028210	0.0865894
Nuclear	3485	0.4274500	13.1204067
Other	94	0.0115294	0.353890
Other renewable	271	0.0332392	1.020263
Waste	20	0.0024530	0.075293
Wind Onshore	120	0.0147185	0.4517784
Total	8153	-	-

Table 3.5: Generation types, values, the normalized value, and the contribution of CZ to Italy's Load. All values in MWh

Fuel Type	Value	Normalised Value	Contribution (0.0018398)
Biomass	4132	0.0824520	3.586679
Fossil Brown coal/Lignite	6135	0.1224208	5.325330
Fossil Gas	3273	0.0653110	2.841042
Fossil Hard coal	2411	0.0481103	2.092795
Fossil Oil	482	0.0096180	0.418385
Geothermal	3	0.0000598	0.002601
Hydro Run-of-river and poundage	1321	0.0263598	1.14623
Hydro Water Reservoir	86	0.0017160	0.074646
Nuclear	8653	0.1726663	7.5108767
Other	383	0.0076425	0.332428
Other renewable	107	0.0021351	0.0928772
Waste	497	0.0091738	0.399062
Wind Offshore	3136	0.0625773	2.72212
Wind Onshore	19495	0.3890130	16.92214
Total	50114	-	-

Table 3.6: Generation types, values, the normalized value, and the contribution of DE to Italy's Load. All values in MWh

Technology	Min	Median	Max
Coal PC	740	820	910
Gas Combined Cycle	410	490	650
Biomass – Dedicated	130	230	420
Solar Utility scale	18	48	180
Solar – rooftop	26	41	60
Geothermal	6.0	38	79
Concentrated Solar Power	8.8	27	63
Hydro power	1.0	24	2200
Offshore	8.0	12	35
Nuclear	3.7	12	110
Wind Onshore	7.0	11	56

Table 3.7: Life cycle CO₂ equivalent from selected electricity supply technologies. Values in the table represents (gCO₂eq/kWh) values [1] [2]

Fuel Type	IT (kgCO ₂)	AT (kgCO ₂)	CZ (kgCO ₂)	DE (kgCO ₂)	Total Emission due to Consumption
Biomass	111320	1164	216	823	113523
Fossil Brown coal/Lignite	0	0	9838	4320	14158
Fossil Coal-derived gas	622380	0	570	0	622950
Fossil Gas	3858750	10365	444	1391	3870950
Fossil Hard coal	1553900	2123	722	1715	1558460
Fossil Oil	8993	0	12	478	9483
Geothermal	25042	0	0	0.1	25042.1
Hydro Run-of-river and poundage	44376	1076	0.44	27	45479.44
Hydro Water Reservoir	6768	46	1.943	1.79	6817.73
Nuclear	0	0	157	90	247
Other	2101400	256	247	230	2102133
Other renewable	0	0	35	3.23	38.23
Waste	8280	383	17	91.11	8771.11
Wind Offshore	0	0	0	27.27	27.27
Wind Onshore	54604	44	5	154	54807
Total	8395813	15457	12265.38	9351.5	8432886.88

Table 3.8: Total CO₂ emission per fuel type per country. (All values are in kgCO₂)

For calculating the CO₂ emissions, only the generation values of different techniques are considered as shown in Table 3.9. Final CO₂ emissions are in kgCO₂ units. Production based accounting shows that 8,586.7 Tonnes of CO₂eq is emitted due to production. From Table 3.8, total emission due to consumption is 8,432.88 tonnes of CO₂eq.

Fuel Type	Value in MWh	CO₂ Emission (kgCO₂)
Biomass	495	113850
Fossil Coal-derived gas	776	636414
Fossil Gas	8053	3945970
Fossil Hard coal	1938	1589999
Fossil Oil	8	9196
Geothermal	674	25607
Hydro Run-of-river and poundage	1891	45384
Hydro Water Reservoir	289	6936
Other	3070	2149000
Waste	37	8510
Wind Onshore	5076	55836
Total	22307	8586702

Table 3.9: Production-based accounting, CO₂ emission due to generation in Italy.

Chapter 4

Methodology and Implementation

In this chapter, the design of the dashboard and implementation of the flow tracing methodologies are explained. Algorithms represent the implementation of the flow tracing method in code. The sequence of working of update scripts that are used to gather data from the ENTSOE platform is explained using a flow chart. The different tools, libraries, and software utilities that are used in implementing this dashboard are discussed in detail.

4.1 Flow Chart

Figure 4.1 defines the overall design of the dashboard architecture. This application is split into two modules namely update scripts and dashboard scripts. These two modules work in tandem to complete the functional application. The update scripts and the dashboard backend are abstracted for the user who only interacts with the application through the web browser. The application is completely developed using python as the development language, the front-end and back-end of the application are powered by the dash, a framework to realize the data and build a web application using python as the primary language.

The user interacts with widgets rendered by the front-end side of the appli-

cation in the browser. These interactions are bundled into user requests that trigger the callback functions in the backend. The computation logic function performs all the calculations and returns the values as user responses that are shown as graphs or panel in the browser.

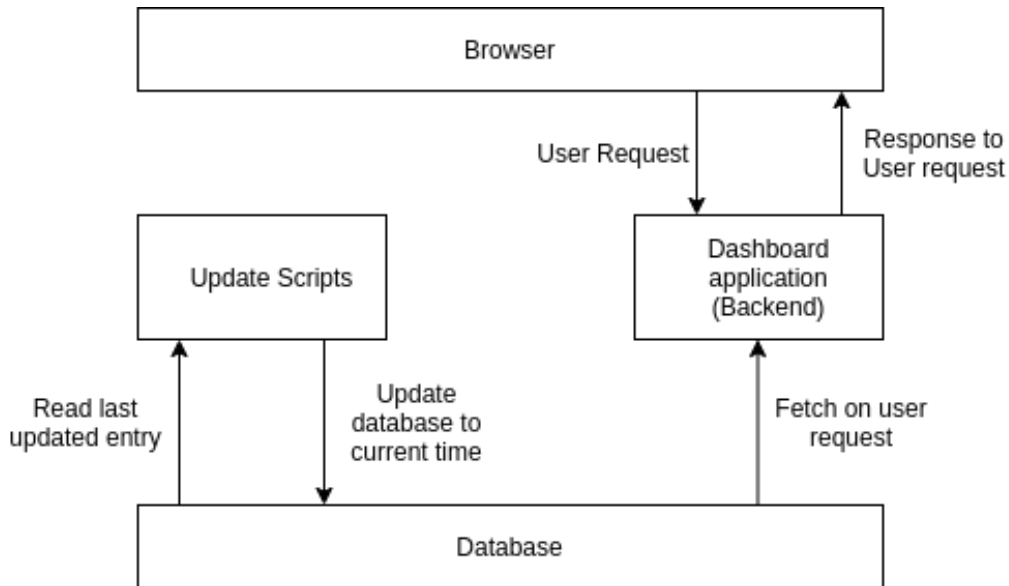


Figure 4.1: Dashboard application architecture

The update scripts work in sync with the dashboard application. The dashboard script and update scripts share the dashboard. The functioning of the update scripts is as shown in Figure 4.2. The index of the table in the database represents the date. The last records from the database are fetched, the date from these records is calculated against the current date and time. The different entries are requested from the ENTSO-E platform are updated in the local database. Update scripts executed on an hourly basis to fetch. The databases are updated to provide faster response to user requests.

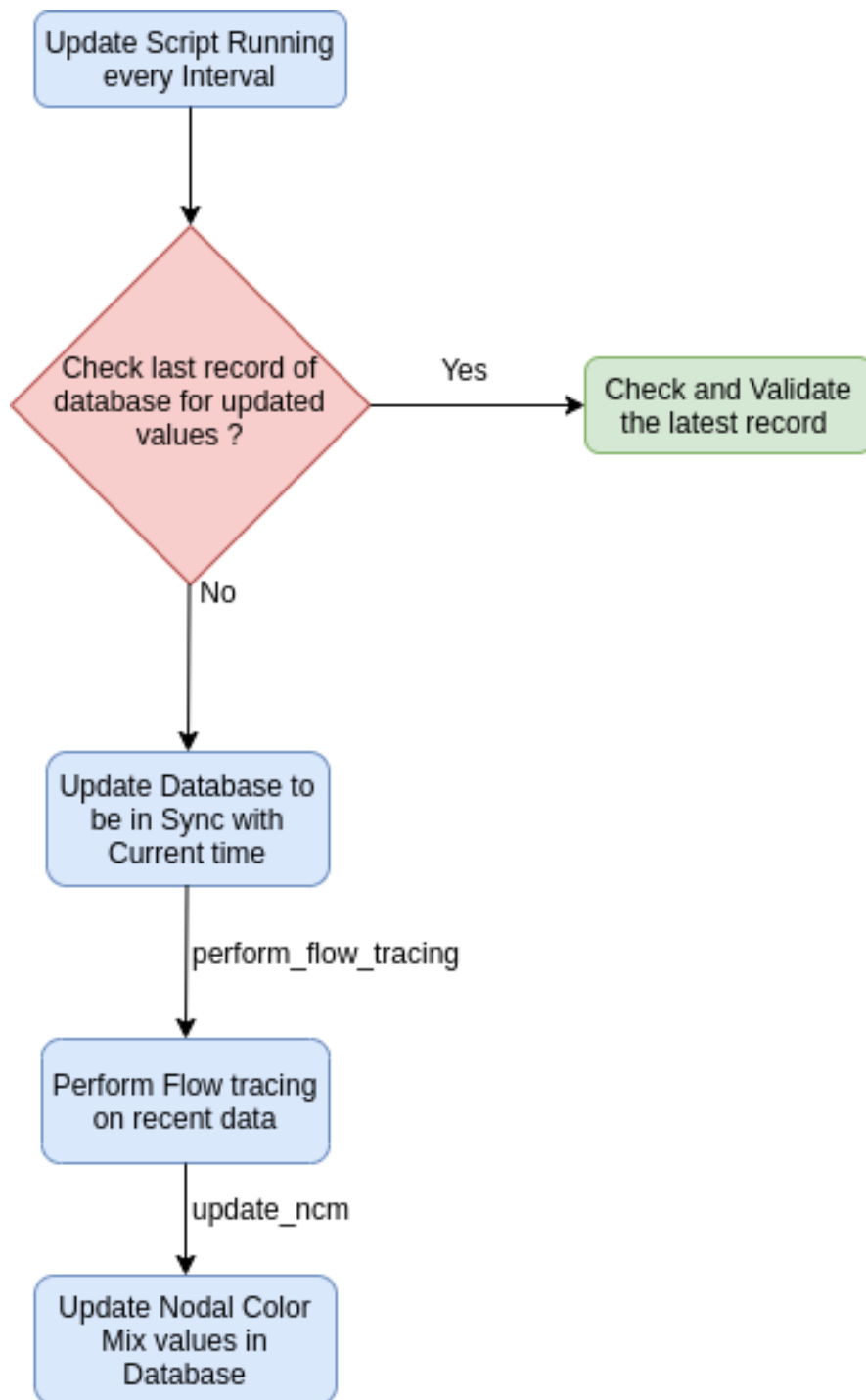


Figure 4.2: Work flow for updating database

4.2 Algorithm

This section explains the implementation of flow tracing methodology using different algorithms. The algorithms defined here are implemented in Python as the development language. Algorithm 1 shows the core logic to find out the nodal color mix. The nodal color mix for every country is found for every hour. The data worked on is both historical data fetched from the ENTSO-E platform to a local database or through real-time web query. As explained in Section 3.3, the system of linear equations is solved with the aid of matrices. The matrix is of form $\mathbf{Ax} = \mathbf{B}$, and matrix A and matrix B has to constructed before solving. The matrices are constructed using the pandas for better computational benefits. To solve the matrix, a helper function from the scipy python package is used.

Algorithm 1 Solving Flow Tracing in Python

- 1: Create empty matrix_A
 - 2: Create empty matrix_B
 - 3: **for** Every Country **do**
 - 4: matrix_A = calculate_diagonal_elements_A(matrix_A)
 - 5: matrix_A = calculate_off_diagonal_elements_A(matrix_A)
 - 6: matrix_B = calculate_diagonal_elements_B(matrix_B)
 - 7: Solve $x = A^{-1}.B$
 - 8: **end for**
-

Algorithm 2, shows the implementation of matrix A. This matrix takes the load, country imports and exports, and storage into consideration. The diagonal elements of the matrix consist of load, total exports, and charging storage of each country.

Algorithm 2 Computing Diagonal Elements for Matrix A

- 1: Fetch Load,total exports, charging storage from database
 - 2: `matrix_A.iloc[country] = Load+total_exports+ total_storage_charging`
 - 3: return updated dataframe
-

Algorithm 3, explains the methods of populating the non-diagonal elements of the matrix A. These nondiagonal elements are the imports to the country. As seen in Section 3.3, the imports to the country have a negative sign indicating the flow to the node. All the nondiagonal elements are updated for every country in the loop and the final matrix A obtained will be a sparse matrix. A sparse matrix is a matrix that has most of the elements as zero. Sparse matrices are used extensively to speed up data processing, and helpful in saving memory. Operations like addition, subtraction, multiplication, division, and matrix power are performed using sparse matrix [42].

Algorithm 3 Computing Off - Diagonal Elements for Matrix A

- 1: Fetch imports of country from database for selected time frame
 - 2: `matrix_A.iloc[country] = - (import_value)`
 - 3: return updated dataframe
-

Matrix B is a diagonal matrix, where the diagonal elements are generation values for each country. The diagonal elements are the sum of all generation techniques. The sign of values remains positive indicating the indigenous generation inside the node. Algorithm 4 explains the construction of matrix B.

Algorithm 4 Computing Diagonal Elements for Matrix B

- 1: Fetch total generation of country from database for selected time frame
 - 2: `matrix_B.iloc[country] = total_generation`
 - 3: return updated dataframe
-

Once both the matrices are constructed. The nodal color mix is calculated. Matrix A, as shown before, is a sparse matrix. The matrix is converted from pandas dataframe to NumPy array. The reason for conversion is because of the supported lower-upper(LU) decomposition of a sparse matrix in scipy package [43]. After converting the matrix A to a numpy array, the sparse matrix is converted into a compressed parse column (CSC) matrix. This conversion holds an advantage as matrix-vector products become faster, CSC matrix supports faster arithmetic operations by performing efficient column slicing [42]. Algorithm 5 explains the procedure to compute a nodal color

mix.

Algorithm 5 Computing nodal color mix

- 1: Convert the pandas dataframe to numpy array
 - 2: $\text{matrix_A_numpy} = \text{matrix_A.to_numpy}()$
 - 3: $\text{matrix_B_numpy} = \text{matrix_B.to_numpy}()$
 - 4: $\text{csc_matrix_A} = \text{csc_matrix}(\text{matrix_A_numpy})$
 - 5: $\text{nodal_color_mix} = \text{linalg.splu}(\text{csc_matrix_A}).\text{solve}(\text{matrix_B_numpy})$
 - 6: return updated dataframe
-

The nodal color mix is a matrix, which shows the contribution of the countries in satisfying the load. The contributing countries need not be directly linked. The contribution factor is multiplied with the total load of the country and normalized generation vector to find the nodal power mix as shown in Algorithm 6. The nodal power mix is the contribution of a generation technique in satisfying the consumption node. This is completely traced down from the consumer end to the generation plant.

Algorithm 6 Computing nodal power mix

- 1: Identify color mix for each country
 - 2: Find out the sum of all generation techniques, refer Section 3.2
 - 3: Find out the normalised generation vector
 - 4: $\text{nodal_power_mix} = \text{load_of_country} * \text{color_mix} * \text{normalised_generation}$
 - 5: Perform the above operation for all contributing country
-

The final traced down contribution of each generation technique, also known as nodal power mix, represents the type of fuel used in the production of electricity that has been consumed in the consumer end. The IPCC assessment has given guidelines and the level of CO₂ emissions for electricity sources. The Table 3.7 summarizes the minimum, maximum and median values for different techniques [1]. The total sum of the product of nodal power mix and these IPCC values give the final CO₂ emissions due to the consumption.

The database is updated at regular intervals in order to provide a faster user experience while using the dashboard. In order to keep the database updated, the new data from the ENTSO-E platform is requested using update functions. Algorithm 7, explains the procedure of updating the database. The last updated date is read, compared against the current date, and the difference in days or hours is updated. The updated data is fed to the flow tracing algorithm to compute the nodal color mix. The nodal color mix values are updated for each country in the database as shown in Figure 4.2.

Algorithm 7 Updating the database to be in sync with dashboard

- 1: Read the last element of all the tables in database
 - 2: Fetch the last updated date
 - 3: Web query to pull the data from ENTSO-E platform to the current date
 - 4: Update the database with newly queried data
 - 5: Run flow tracing methodology to identify the nodal color mix
 - 6: Update the nodal color mix database for every country
-

4.3 Technology stack

4.3.1 NumPy

NumPy is an open-source project which makes numerical and scientific computing with Python easier. It is one of the fundamental packages in Python which can be installed with the help of any python package manager. NumPy helps in solving basic linear algebra equations, Fourier transforms, and other mathematical equations. NumPy arrays are powerful and support N-dimensional computations. Along with their ease of usage for computational, they can also act as containers holding generic data types. The fact that it facilitates the N-dimensional data type, the operations on a large number of data. and complex advanced mathematical equations are computed efficiently with less code.

We used following functions of NumPy for our computation.

- **np.zeros:**

For creating the initial empty dataframe for all the EU28 countries. This function accept shape of the array to be created and datatype held by the dataframe. This function returns a new uninitialized empty array.

Usage: `np.zeros((28, 28))`

- **np.where:**

Numpy provides functionality select among multiple options based on the conditions. If there are two conditions like x and y, Where returns ndarray of x elements when conditions are true, or y elements otherwise. Here, a matrix is checked for zero rows using where function of NumPy.

Usage: `np.where(matrix_A.any(axis=1))[0]`

4.3.2 Pandas

Pandas is an open-source and free software library written for the Python programming language which is mainly used for data manipulation and analysis of large data. It provides functionalities and offers data structures for

handling time series and manipulating the larger data in the form of tables. The processed data can be of any form i.e, either heterogeneous or homogeneous data. The tables consist of rows and columns and they have inbuilt indexing which labels each row and column. Pandas do not limit to a single dimension of data. It supports 1-Dimensional and 2-Dimensional data in the form of time Series and DataFrame respectively. Time Series is 1D, labeled, and work on homogeneous data, Usage of pandas makes it easier for the managing the larger dataset due to inbuilt functionalities like merging, dropping, group by, slicing, indexing, reshaping, and joining of the data structures. Pandas are reliable and are very fast in terms of execution which makes it a viable option in financial and data analysis applications. In addition to the data handling, pandas provide various number of input-output operations. It is easier to export the pandas DataFrame to files like CSV, SQL, XLSX, and TXT [44].

We used following functions of Pandas for our computation.

- **pd.DataFrame:**

As stated earlier, DataFrame operates on 2D heterogeneous data. Usually, data is a form of array or table. This dataframe is mutable, meaning the size can be varied at runtime. Users can create a labeled index or by default index will be enumerated. This dataframe can be created from other data structures like a dictionary or records using functions like DataFrame.from_dict. This function takes data in form of array or dictionary or any iterable, index, columns as input and returns the dataframe [45].

Usage: pd.DataFrame(my_data, index=indices, columns=col_names)

- **pd.DataFrame.iloc:**

Stands for integer location. This function is used to address the entry inside the dataframe. This function can select multiple entries, based on the argument given. If arguments are the list of numbers, then all the entries associated with the indices are returned. This function becomes handy while modifying certain entries in the already existing dataframe.

Usage: pd.DataFrame.iloc[index_num] = user_defined_value

- **DataFrame.to_csv:**

Pandas provide various input-output functions, which can be used to export the computed dataframe into user-defined files. CSV is one of the popular and widely used formats. The dataframe object can be exported to the CSV file. IDE supports the scientific view window to analyze the dataframe in a tabular format. However, CSV files when opened in XLSX editors like LibreOffice or Microsoft Excel, make it easier for the user to understand the dataframe.

Usage: `pd.DataFrame.to_csv("filename.csv")`

- **Dataframe.read_csv:**

Read is an input utility provided by pandas. With the help of this function, CSV files are loaded onto the dataframe. This function takes a file's path as arguments and returns a dataframe. This function enables users to work on locally stored CSV files as well as CSV files from the internet using the valid URL.

Usage: `pd.DataFrame.read_csv("filename.csv")`

4.3.3 Dash

Dash is a python framework used to build web applications. Dash is ideally used to build an application that involves data analytics and visualizations. This framework abstracts backend and frontend for user convenience and allows one to develop applications quickly using pure python skills. Dash uses flask for enabling backend webserver and plotly.js and react.js for front-end visualizations. Apps built using dash are rendered at the browser end, these apps are easily deployed which generates the valid URL and shared with others. Dash has a vibrant community of developers enabling easy and faster development. An individual developer who has python language skills does not have to waste time learning new tools or development language to use the dash [46].

Dash apps are visually pleasing, require fewer number lines of code, and are reactive in usage. User interactions are done using the utilities like drop-downs, checkboxes, buttons, radio buttons, and text areas. Dash supports

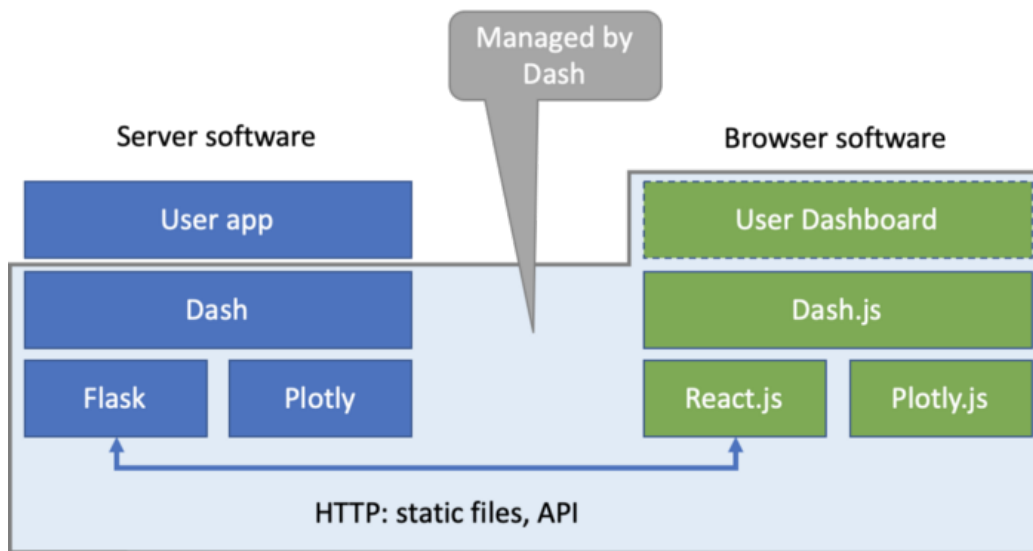


Figure 4.3: Dash architecture [6]

DCC, which has a wide range of inbuilt utilities. DCC comes as a python package, which is imported directly to the python code. Dash apps have `app.layout`, which defines the layout of the application which is rendered in front-end. The interactions with the layout are processed at the backend using app callbacks. Callbacks are developed using python decorators, which takes inputs and outputs. These callbacks are sensitive to input lists and whenever there is a change of value in the input items, these callbacks are triggered and the functions which are wrapped under these callbacks are executed. Each item in the layout has a specific id, which is used to address the input and output for the callback functions.

As shown in Figure 4.4, the user interacts with the date picker component is tagged by the id: `selection`. Callback function `display_user_data` uses this id as the input. So, whenever there is a change in the value, the callback function is triggered and the function is called. The Output of the callback function is using the id: `display`, which updates the text field with user selection value. The layout can be dynamically updated with new components by created panels in the callback function and return them to the HTML

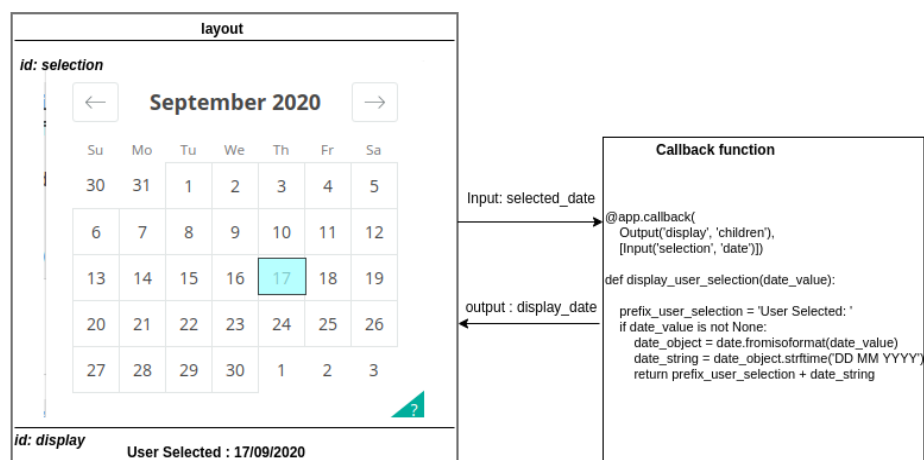


Figure 4.4: Dash callback interaction with layout

generic container called div. This dashboard makes use of this feature for updating the charts dynamically which are processed and created in the callback functions [47].

DCC contains a graph component that is used mainly to build and display figures. Graph component accepts figure argument and renders graphical figures like charts, maps, and diagrams to display. Rendering is managed by a JavaScript library Plotly.js which is abstracted by a dash. This app makes use of bar charts, pie charts, and panels for displaying the summarized report. All these figures are created in the callback functions and are passed displayed on the layout [48].

4.3.4 SQL

Structured Query Language is used in conjunction with programming languages for managing the data, structured data in particular. This is a non-procedural language, although standard SQL extended to have procedural language with extensions. SQLAlchemy enables developers with flexibility and ways to harness the full power of SQL. This toolkit is efficient and persistent and adapted to be used just as python language, so developers would not face any unintended overhead. Very often, SQLAlchemy considers

the database as a relational algebra engine, not as a collection of row and columns. The queries to the standard SQL are replaced by the functions, which makes it more like a pythonic way. Execution interface offers to work with DBAPI through an engine created using `create_engine` function. The connection created maintains the scope of the connection and binds it to the context. The queries which are executed using the `execute` function return `ResultProxy`.

`ResultProxy` is used to fetch data based on the data requirement. `Fetchone`, `fetchmany`, and `fetchall` are the functions available in `Resultproxy` which can be used to fetch data according to the usecase. For the larger dataset, `.fetchmany` or `.fetchall` are used. One of the interesting features which make `SQLAlchemy` prominent is its ability to convert the result list to `pandas dataframe` [49].

Tables are created for every country. Each country has a table for generation data, import data, day-ahead price, and forecast. These tables hold data since 2015. Queries are constructed to fetch the data for each entity based on user selection. The date for which data is requested is selected by the user from the user interface. Function `.fetchmany` is used to fetch all the data of the particular date. The results are then processed using user-defined data structures for convenience. `Services` are created which runs based on user-defined time interval which fetches data from the `ENTSO-E` platform and updates each table for every country. Later these data are used in future queries.

4.4 Implementation

4.4.1 Database Preparation

The application works on a large dataset to give the user statistics of electricity production techniques, consumption metrics, imports to, and exports from the country. Data is available since 2015 in the platform that accounts for all these variants, there is a need for a larger database. Data from 2011

to 2015 exists in the form of XML and Xls format. However, there is no data related to generation values from 2011 to 2015. If there is a missing data for an hour or for a month, it is safe to average it by using previous values. However, averaging for 4 years doesn't yield in right amount of accounting. Hence flow tracing is performed only for data from 2015.

Some countries provide the values in 15-minute intervals, in this case, the average of values for one hour is taken for calculations. Data for some countries are often not present in the platform for unknown reasons. However, we fill the missing values with the previous values. In case of missing data, once the data are updated in the local database, the missing values are filled with the average values of the previous day. For example, if a generation data is missing on for Austria(AT) on January 21st at 09:00, in that case, the previous hourly data is used to fill the missing values. We observed that Serbia doesn't have any generation data until 2016 December. A country with missing data for a longer timeframe is ignored from flow tracing. However, if there is a missing value for some hours, the values are averaged to fill the missing values. The control areas interconnected system operated by a single operator whereas bidding zones are the largest involving exchanges by market participants. The country is the geographical border for each country in the transmission network. Since we are evaluating each country, the data related to each country is considered as shown in Figure 4.5. We have cross-checked the data between bidding zone values, and country-wise values, the data integrity is maintained. The country-wise exchanges are nothing but a sum of each bidding zone data. In case of pricing, each bidding zone has its own prices, in that case the average price of each bidding zone is taken into consideration. For example, Norway or Italy or Sweden region is split into different zones, in that case, the average price of all the zones are considered.

A Web API Security Token, assigned by ENTSO-E platform [7] are required to access the data through web requests. Databases are periodically updated by using update scripts. The scripts run at pre-defined time intervals in the backend and append newly available data in the platform to the database. This Database has multiple tables namely *GENERATION*, *IMPORT*, and

DAY AHEAD PRICE for each country. Data query APIs used in these scripts are developed by EnergieID as part of the ENTSO-E project [50]. This package provides clients namely RawClient and PandaClient which return data in XML or zip of XML formats and pandas respectively.

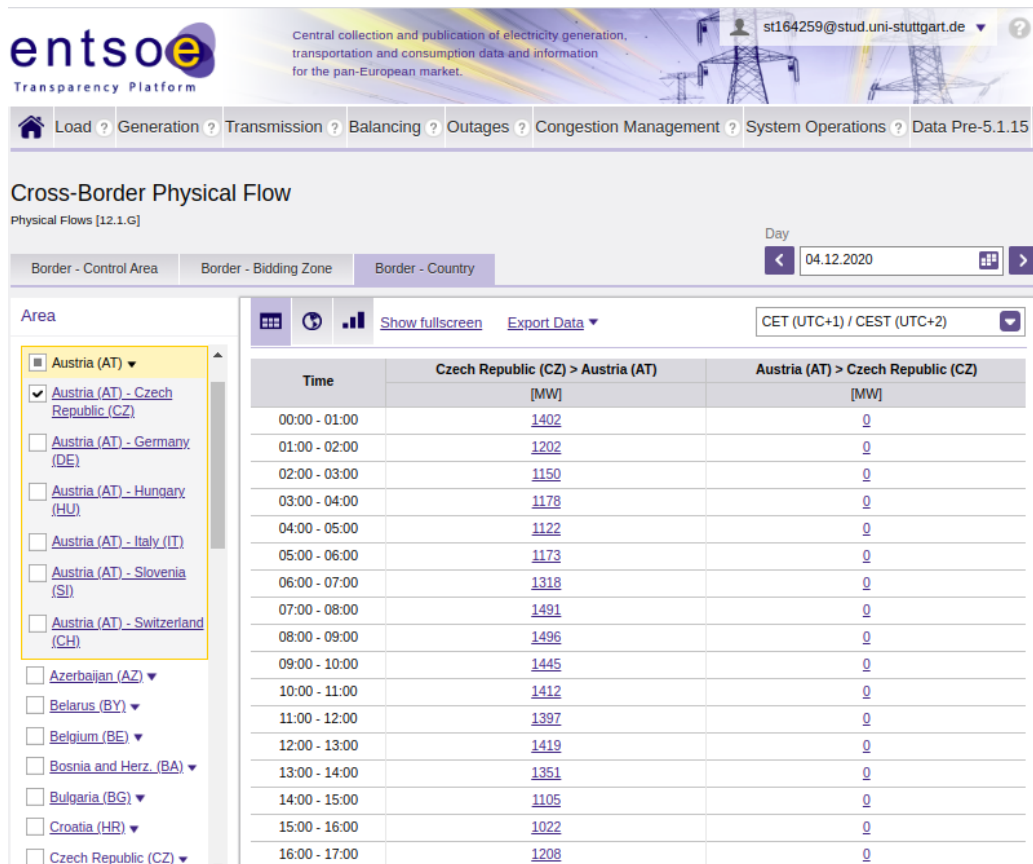


Figure 4.5: ENTSO-E platform used as primary data source [7]

Flow tracing algorithm takes nearly 36 hours to perform calculations on 1 year worth of data. These calculated values are stored in the nodal color mix database. An update script runs in parallel with the dashboard scripts. This update scripts, performs web query every hour, updates the local database, performs the flow tracing algorithm, updates the nodal color mix database, and computes the CO₂ emission values. Based on the user query, the data will be served from the local database. The advantage of creating a database

beforehand is directly dependent on user experience and response time while using the application. Requesting data via APIs by doing web queries and computing the logic hinders the performance and results in a longer waiting time for the user. Hence, a database is created which supplies the necessary data to the computation logic.

4.4.2 Dashboard implementation

The dashboard source code structure is as shown in 4.6. The `index.py` file initializes the navigation bar, support multiple page application, and explore options. This file also creates the dash app and initializes the server. The server is initialized with the debug option by default, it can be turned off by setting the debug filed to `False`. The bootstrap theme used in this application is a LUX theme. Reference [51] is used as a base to create the bootstrap components for this dashboard and are initialized in the `app.py` file.

Each of production, consumption, and CO₂ have their implementation in separate files. These files create their layout structures and callback functions. File `database_query.py` provides the interface functions to interact with the database by querying. This file takes care of all the operations related to the database. Any modifications to the database structures is done using this file. Instructions for initial setup, required packages, and the running of the dashboard are provided in the `README.md` of the source code.

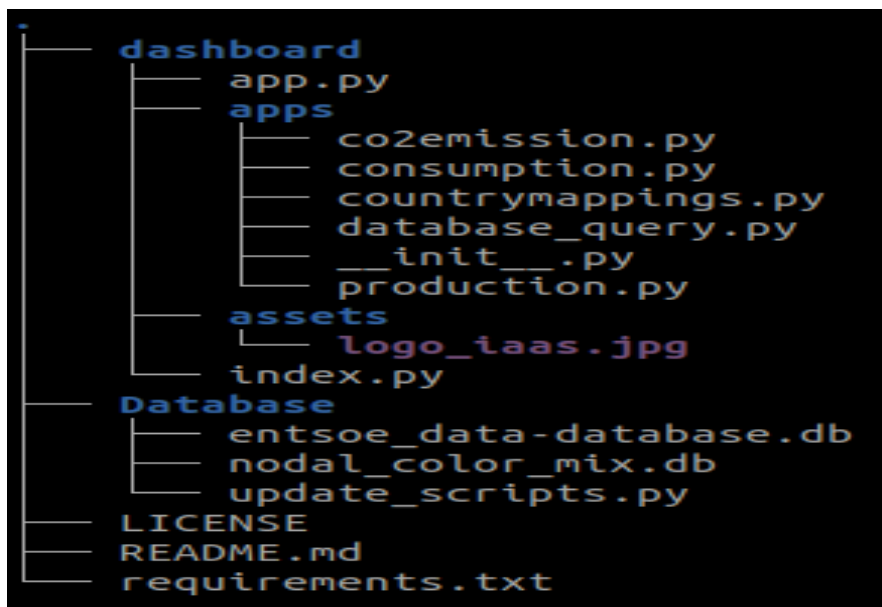


Figure 4.6: Source code tree of dashboard application

Chapter 5

Results

5.1 Dashboard

Once the dashboard application is started, the dashboard can be viewed on the browser. By default, the Production page is shown as the home page. However, the user can select the required page by selecting one of the options from the **Explore** icon as shown in figure 5.1. Based on the user selection, the corresponding dashboard page will be opened.

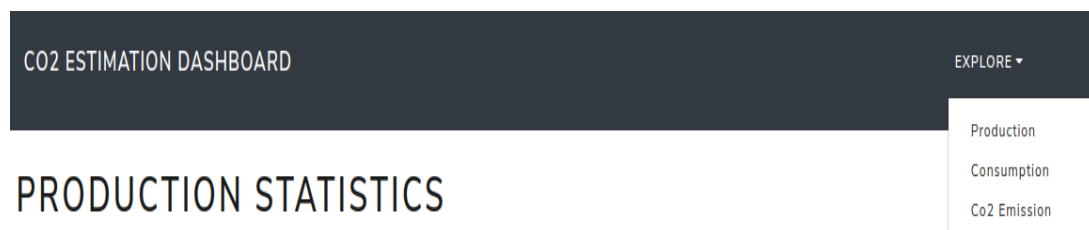


Figure 5.1: User selection for options

Every page follows an identical operating procedure, that it requires a user to select countries for comparison, the metric to be compared against, and the period i.e, start date and end date. Once the options are selected, these options serve as inputs to the callback functions. The functions are triggered, the application logic corresponding to the options are executed and the results are returned in form of charts, figures, or panels.

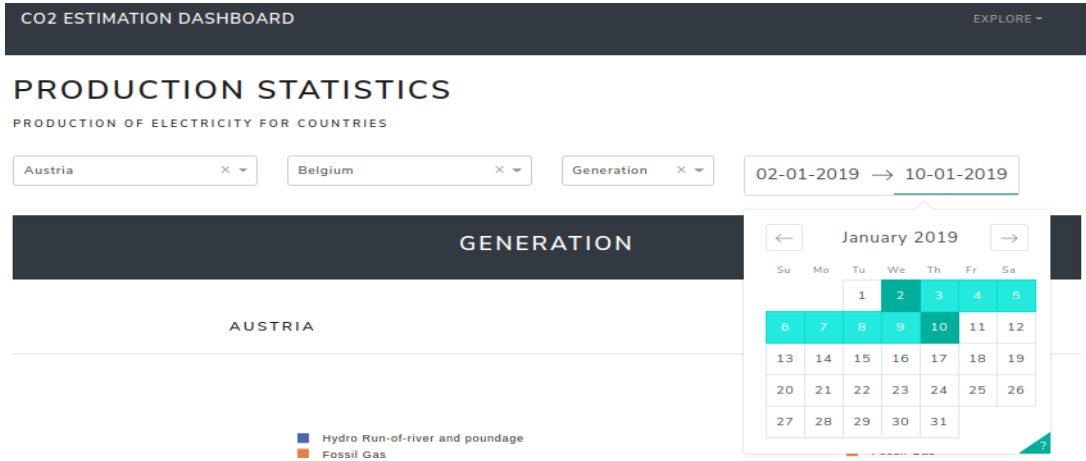


Figure 5.2: Selecting Multiple Countries for Comparison and time period

As described above, the application opens the production page by default. Once the page is loaded, it waits for a user to select the countries for comparison, the type of metrics to be compared, and the time period. Supported types are the type of generation(Generation) and day-ahead price. Once the required options are selected, based on the amount of time, the application displays two sections. The first sections show the pie chart involving all the modes of generation techniques involved in the production of electricity for each country. The second section is a panel, which displays the summarized output for the user. Here, the number of days under consideration, the total power generated in the country under that time frame, and the storage are displayed as shown in Figure 5.3. All the power values are in the MWh units.

Day ahead price shows the trend based analysis about varying price over selected days. The panel also shows the maximum, minimum and average price for the selected dates as shown in figure 5.4. The bars represents the price for individual days, and the line graph follows the trend across different days.

Consumption statistics page gives all the information about each country load, imports from and exports to the connected countries. As performed before, once the consumption page is loaded, user will select the countries

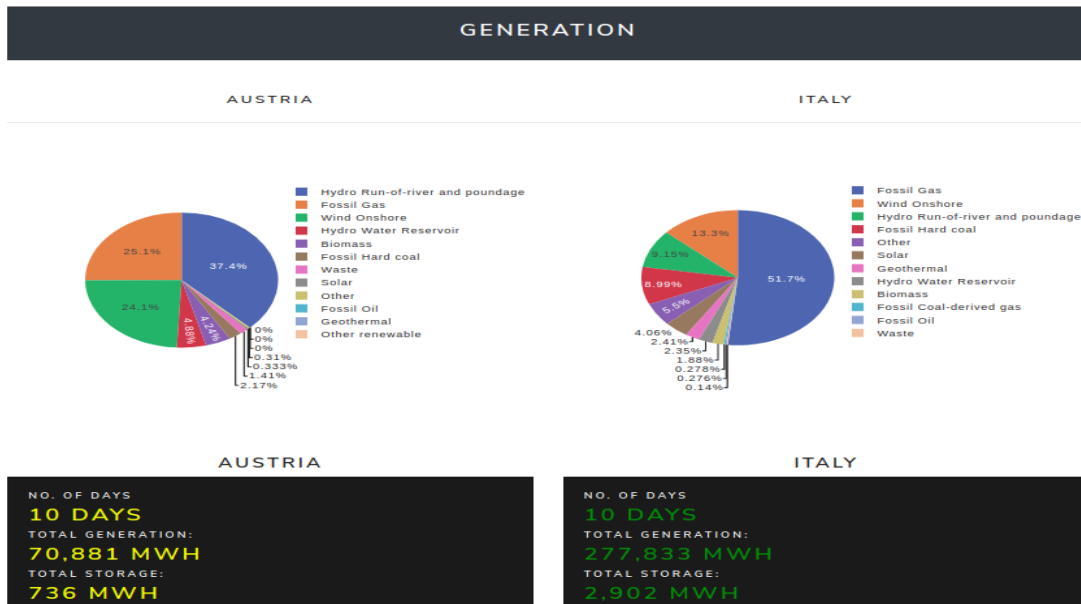


Figure 5.3: Production Share of different techniques and total generated value

and time period. The page displays the summarised information in the panel for each country as shown in Figure 5.5.

Next section, named as **IMPORTS FROM COUNTRY** in the page displays the pie charts explaining the imported values from the neighboring countries as seen in Figure 5.6.

As we have the results of flow tracing, the generation technique used in the imported values are traced down and the same are displayed in the section **IMPORTS BY GENERATION TECHNIQUE** as shown in Figure 5.7.

Exported power from country is shown as similar to that of imports. A pie chart representing the percentage share exported to each country. These exported values are the direct exports from the country as shown in Figure 5.8.

At last, to know the total exchanges of the country, the import and export



Figure 5.4: Price trend showing max, min and average price for selected days values are plotted together to get the better understanding of the exchanges. This is shown in Figure 5.9.

CO₂ emissions are calculated using IPCC values [1] as shown in the example in Section 3.4. The values are classified into different categories namely emissions due to renewable sources, non-renewable sources, and nuclear emissions. These values are displayed in a summarised panel as shown in Figure 5.10.

The trend shows the emission levels over the selected period. The trend graph consists of all the categories of emissions. These emissions can be analyzed separately by clicking on the desired graph as shown in Figure 5.11

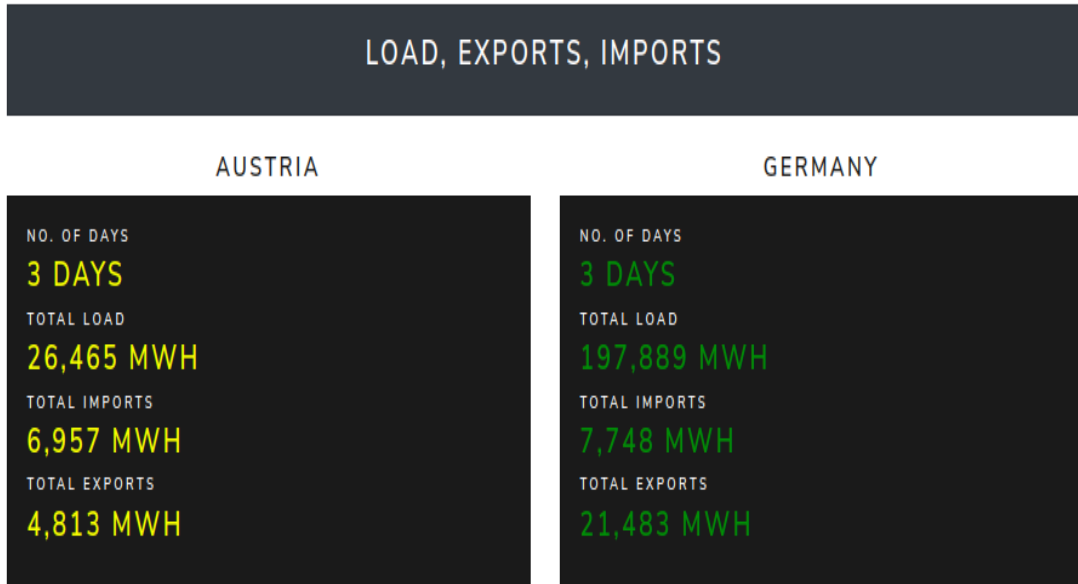


Figure 5.5: Total Load, Total Imports and Total Exports for the countries

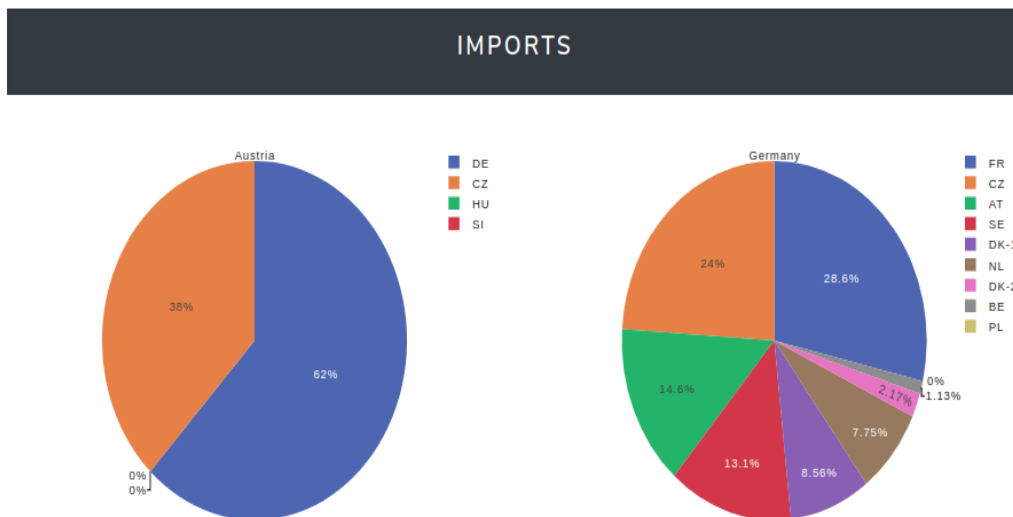


Figure 5.6: Import statistics for the countries

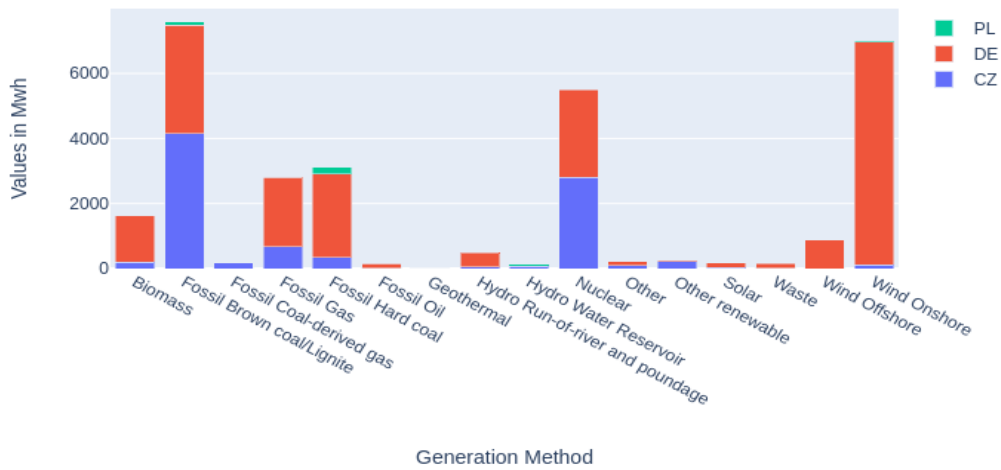


Figure 5.7: Import to a country from different generation sources of it's neighbors

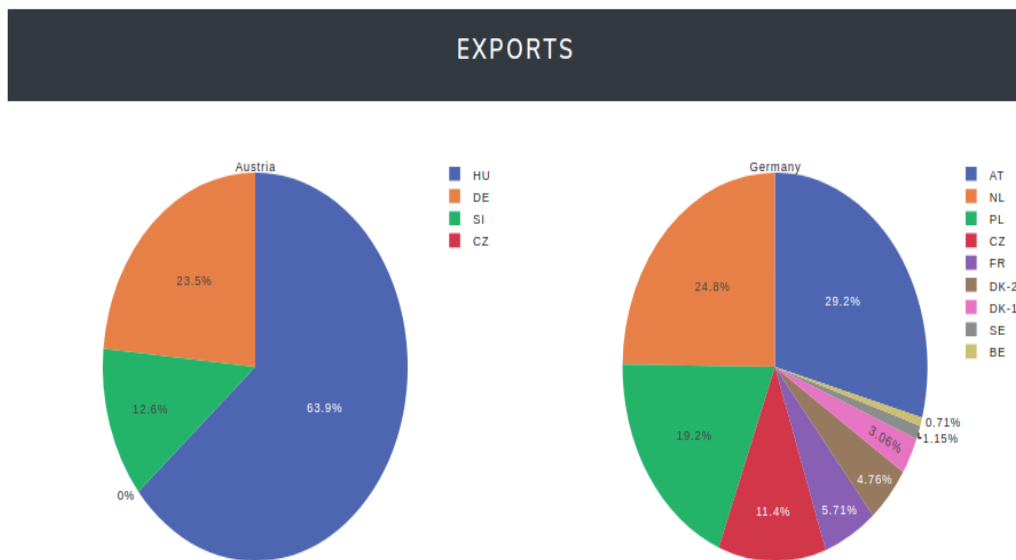


Figure 5.8: Export statistics for the countries

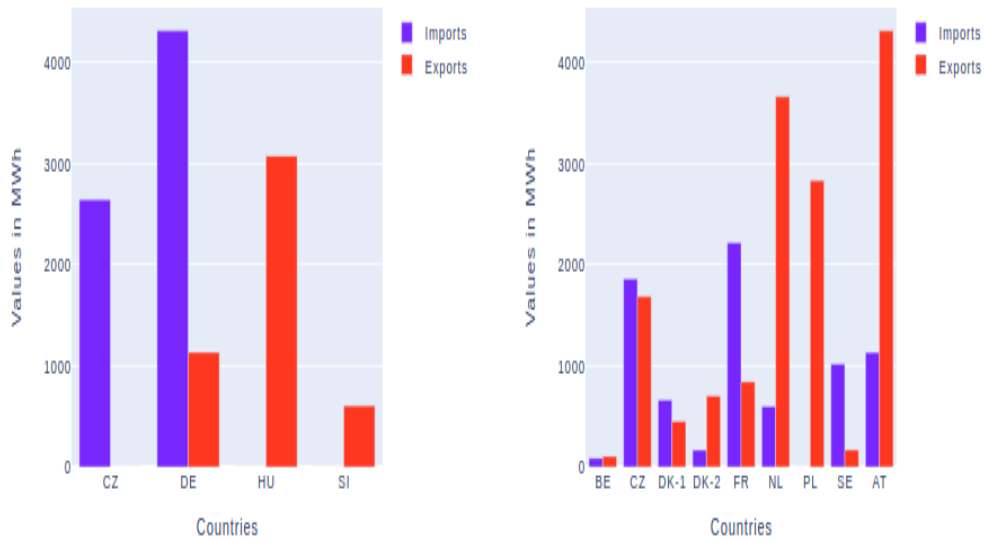


Figure 5.9: Import and Export statistics for the countries

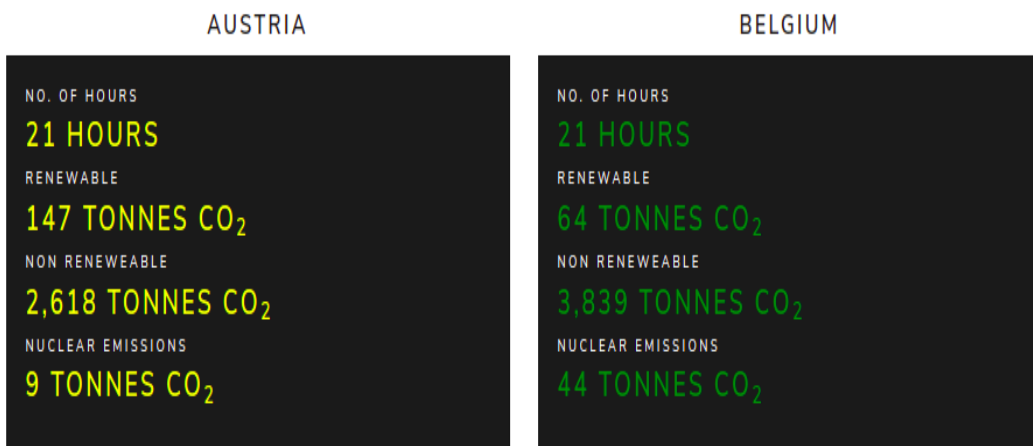


Figure 5.10: CO₂ emissions panel for different categories

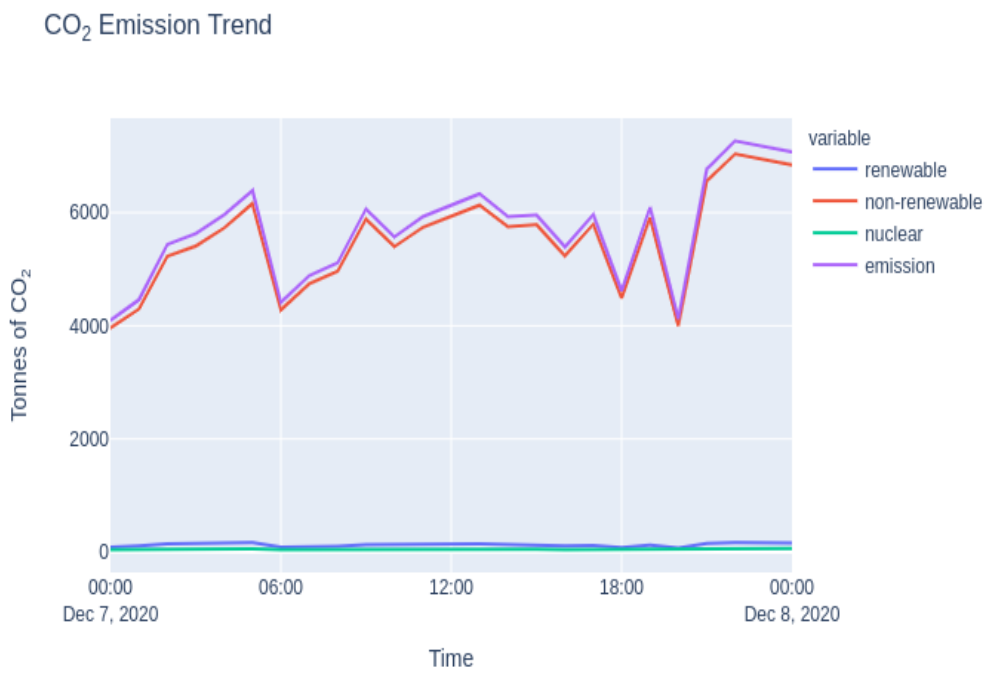


Figure 5.11: CO₂ trend of selected time frame for different categories

5.2 Evaluation

In this Section, the different features of the dashboard are tested for different countries. The evaluation parameters are the unique features our dashboard provides against the existing dashboards. At the end of this section, the unique features of our dashboard are compared against the features of the other currently existing dashboard which we evaluated in Section 2.2.

5.2.1 Larger period

This dashboard enables the user to select the period ranging from days, weeks, months, and also over years. This range of evaluation is one of the unique features of our dashboard. However, by selecting the larger period, the performance can be a little slower. By default, the end date is set to the current date. The start date is of user choice. There are no cross yearly limitations. The year of the start date and end date can be different. Carbon Intensity [52] supports the download of data for 31 days, however, this data provides information only on carbon emission levels. There is no information about electricity generation and consumption. The historical data from the electricity map has not been evaluated as data is not openly available. Eco2mix [34] supports larger periods of up to 8 weeks. Eco2mix supports production, consumption imports, and CO₂ emission due to generation values. Energy charts, provide yearly, monthly and weekly data, but it is limited to Germany.

5.2.2 Real time and Historical Data

Carbon Intensity, electricitymap, Eco2mix supports real-time forecasts and analysis on past data. However, the time frame of evaluation varies according to each platform. Electricitymap historical data is charged, hence not considered for evaluation here. Carbon Intensity supports data in their platform since 2017 and also supports real-time CO₂ emission of Great Britain areas, also providing the next 96 hours forecast. Our dashboard provides analysis on historical data related to production, consumption, imports, exports, and CO₂ emissions from 2015. The real-time visualizations are shown until the

latest available data in a day. If the selected start and end dates are the same, the data for the current day is displayed.

5.2.3 Consumption Based Accounting

While many dashboards are providing production-based accounting, platforms that take consumption into account are very limited. Our dashboard, developed using flow tracing methodology, accounts for CO₂ emission due to consumption. Electricitymap is another platform that offers consumption-based accounting.

5.2.4 Comparison between two countries

It is an exclusive feature of our dashboard providing a comparison between two countries for different types of data. For a better analysis, two countries are compared against each other for the same criteria and period. The behavior of each country is learned and can be examined further.

Most of the distinguishing features are considered for evaluation here and summarised in the Table 5.1, few other commonalities exist across platforms that are not taken into consideration.

Features	Our Dashboard	Other Platforms
Real Time	Yes	Electricity Map, Carbon Intensity
Historical Statistics	Yes, from 2015	Electricity Map(priced), Carbon Intensity(Only from 2017)
Period of evaluation	No Limit	Limit on time period.
Consumption Based Accounting	Yes	Electricity Map, Carbon Intensity (Limited toGreat Britain Area)
Comparison Between Countries	Yes	No

Table 5.1: Summary of evaluation of important features of our dashboard against existing platforms.

Chapter 6

Conclusion and Future Scope

This chapter highlights the usecase of the dashboard, field of use, the limitations of our dashboard and concludes with the future scope and improvements.

6.1 Usecase

The data available in this platform can be used by administrative authorities to monitor the electricity exchange pattern. The CO₂ emissions over the period indicate the supply and demand pattern, and emission levels associated with it. Based on analyzing this pattern, measures can be planned to tackle the electricity demand by using renewable sources over fossil fuels. It helps in reducing the CO₂ emissions.

The platform can be a central entity for all the countries to learn the usage and emissions trend of other countries and take actions to adapt to reduce the emission levels. Also, this dashboard can be extensively used for academic research purposes. There are a large number of projects going on across different institutes in developing a robust framework for measuring CO₂ emissions. In that case, this dashboard comes in handy to give researchers data comprehensively.

6.2 Limitations

With all the highlighted advantages over the existing platforms, this dashboard carries certain limitations. This is an experimental dashboard developed by an individual, unlike other platforms that have several developers working across the globe. This application is completely developed in Python, using Dash as the framework. Though it is a web-based application, it lacks the major flexibility provided by web development languages like JavaScript and Ruby. The animations that make the app look rich in the user interface cannot be done using Dash. Dash has a certain limitation on the supported animations. Country-wise data visualization is better done using maps, highlighting the selected country on the map for better analysis. The real-time exchanges between countries can be animated on maps. However, with limited capabilities, those visually enriching features cannot be accommodated in this dashboard. This dashboard performs is limited to a single platform like the web. The app layout misbehaves when accessed using via phone.

ENTSO-E regulatory board conducts consists of updates to the platform. The bidding zones can be split into additional zones, a map is getting updated over certain intervals, new zones are added. This information is available as part of their technical reports that can be accessed by public [53]. This information has to be updated manually to the platform. Also, with the addition or deletion of countries and exchanges between countries, the database structure changes. New tables have to be accommodated for the newly added countries. Also, the existing data in the ENTSO-E platform are updated regularly in case of erroneous past data. Our scripts update the data based on the difference between the last updated record and the current time. Scripts do not perform the validation of the past data that has been modified in the ENTSO-E platform.

The sum of the nodal color mix should be equal to 1 if the equations are perfectly balanced as per equation 3.3. However, due to missing data, it sometimes doesn't sum up to 1. This is mainly due to the missing of country data for a particular hour. The missing data happens for small countries like

Montenegro, Estonia, or Ireland. This data inconsistency should be taken care of for the proper accounting. However, as of now, we are not handling this situation separately. We just consider the case to calculate the carbon intensities.

Python packages are consistently tuned implementing new features and deprecating certain existing functions. The functions used in the study are of a certain package version. These changes have to be revised in the project source files.

6.3 Future Scope

The present work represents most of the data related to electricity. The assessment of CO₂ emissions due to the consumption of electricity is of foremost importance, considering the ever-rising demand for electricity. This dashboard meets most of the requirements that are helpful in the evaluation of the behavior of the electricity market. However, there is scope for improvement which can be incorporated in future works. Barring the smooth and good user interface, this dashboard excels in implementing functionalities. But, the dashboard applications are high performance-oriented and visually pleasing. Hence, in the future, this dashboard can be improved to meet the industry standards.

This project emphasizes on flow tracing methodology for accounting for the CO₂ emissions due to consumption. All the functionalities are consumption-oriented. In the future, the dashboard can implement the production-based accounting and use the same to compare against the consumption-based emissions providing in-depth analysis. The dashboard can be extended to run on multiple platforms like an application on the phone and web. The dashboard can support event triggers in the future, where an emission level exceeding certain threshold levels can be notified to the users through email or using any other messaging protocol. Also, this dashboard can have yearly, monthly and weekly reports about all the data that can be directly downloaded instead of computing it every time.

Bibliography

- [1] 2014 IPCC, global warming potential of electricity sources. Last Accessed: 2020-12-06. [Online]. Available: https://www.ipcc.ch/site/assets/uploads/2018/02/ipcc_wg3_ar5_annex-iii.pdf#page=7
- [2] IPCC 2014 Working Group III, Life Cycle Emissions. Last Accessed: 2020-12-06. [Online]. Available: https://en.wikipedia.org/wiki/Life-cycle_greenhouse_gas_emissions_of_energy_sources#2014_IPCC_2C_Global_warming_potential_of_selected_electricity_sources
- [3] Energy Charts, Fraunhofer Institute for Solar Energy Systems ISE. Last Accessed: 2020-12-06. [Online]. Available: <https://energy-charts.info/?l=en&c=DE>
- [4] B. Tranberg, O. Corradi, B. Lajoie, T. Gibon, I. Staffell, and G. Andresen, "Real-time carbon accounting method for the european electricity markets," *Energy Strategy Reviews*, vol. 26, p. 100367, 08 2019.
- [5] Estimating the marginal carbon intensity of electricity with machine learning. Last Accessed: 2020-12-06. [Online]. Available: <https://www.tmrow.com/blog/marginal-carbon-intensity-of-electricity-with-machine-learning/>
- [6] Dash architecture and design. Last Accessed: 2020-12-06. [Online]. Available: <https://towardsdatascience.com/plotly-dash-or-react-js-plotly-js-b491b3615512>
- [7] ENTSO-E transparency platform. Last Accessed: 2020-12-06. [Online]. Available: <https://transparency.entsoe.eu/>

- [8] Fossil fuels in generation of electricity. Last Accessed: 2020-12-06. [Online]. Available: <https://www.iea.org/data-and-statistics?country=WORLD&fuel=Energy%20supply&indicator=TPESbySource>
- [9] Global CO₂ emissions in 2019. Last Accessed: 2020-12-06. [Online]. Available: <https://www.iea.org/articles/global-co2-emissions-in-2019>
- [10] J. Munksgaard and K. A. Pedersen, “CO₂ accounts for open economies: producer or consumer responsibility?” *Energy Policy*, vol. 29, no. 4, pp. 327 – 334, 2001. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0301421500001208>
- [11] W. Graus and E. Worrell, “Methods for calculating CO₂ intensity of power generation and consumption: A global perspective,” *Energy Policy*, vol. 39, no. 2, pp. 613 – 627, 2011, special Section on Offshore wind power planning, economics and environment. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0301421510007858>
- [12] B. Li, Y. Song, and Z. Hu, “Carbon flow tracing method for assessment of demand side carbon emissions obligation,” *IEEE Transactions on Sustainable Energy*, vol. 4, no. 4, pp. 1100–1107, 2013. [Online]. Available: https://www.researchgate.net/publication/260582358_Carbon_Flow_Tracing_Method_for_Assessment_of_Demand_Side_Carbon_Emissions_Obligation
- [13] J. Hörsch, M. Schäfer, S. Becker, S. Schramm, and M. Greiner, “Flow tracing as a tool set for the analysis of networked large-scale renewable electricity systems,” in *International Journal of Electrical Power and Energy Systems*, ser. vol. 96, no. April 2017, pp.390–397, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0142061517308517>
- [14] ENTSO-E objectives. Last Accessed: 2020-12-06. [Online]. Available: <https://www.entsoe.eu/about/inside-entsoe/objectives/>
- [15] ENTSO-E objectives. Last Accessed: 2020-12-06. [Online]. Available: <https://transparency.entsoe.eu/dashboard/show>

- [16] Electricity map. Last Accessed: 2020-12-06. [Online]. Available: <https://www.electricitymap.org/map>
- [17] A. Bruce and L. Ruff, “GB Regional Carbon Intensity Forecast,” 05 2018. [Online]. Available: https://www.researchgate.net/publication/328202500_GB_Regional_Carbon_Intensity_Forecast
- [18] I. Staffell, “Measuring the progress and impacts of decarbonising british electricity,” *Energy Policy*, vol. 102, pp. 463 – 475, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0301421516307017>
- [19] M. Lenzen, J. Murray, F. Sack, and T. Wiedmann, “Shared producer and consumer responsibility — theory and practice,” *Ecological Economics*, vol. 61, no. 1, pp. 27 – 42, 2007. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0921800906002953>
- [20] G. P. Peters, “From production-based to consumption-based national emission inventories,” *Ecological Economics*, vol. 65, no. 1, pp. 13 – 23, 2008. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0921800907005162>
- [21] C. Kang, T. Zhou, Q. Chen, Q. Xu, Q. Xia, and Z. Ji, “Carbon emission flow in networks,” *Scientific Reports*, vol. 2, no. 1, p. 479, Jun 2012. [Online]. Available: <https://doi.org/10.1038/srep00479>
- [22] J. Bialek, “Tracing the flow of electricity,” *IEE Proceedings - Generation, Transmission and Distribution*, vol. 143, no. 4, pp. 313–320, 1996.
- [23] D. Kirschen, R. Allan, and G. Strbac, “Contributions of individual generators to loads and flows,” *IEEE Transactions on Power Systems*. [Online]. Available: <https://ieeexplore.ieee.org/document/574923>
- [24] J. Bialek, “Topological generation and load distribution factors for supplement charge allocation in transmission open access,” *IEEE Transactions on Power Systems*, vol. 12, no. 3, pp. 1185–1193, 1997. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/630460>

- [25] J. W. Bialek, "Allocation of transmission supplementary charge to real and reactive loads," *IEEE Transactions on Power Systems*, vol. 13, no. 3, pp. 749–754, 1998. [Online]. Available: <https://ieeexplore.ieee.org/document/708576?reload=true&arnumber=708576>
- [26] J. W. Bialek and S. Ziemianek, "Tracing based transmission pricing of cross-border trades: fundamentals and circular flows," in *2003 IEEE Bologna Power Tech Conference Proceedings*, vol. 3, 2003, pp. 8 pp. Vol.3-. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/1304456>
- [27] J. W. Bialek, S. Ziemianek, and R. Wallace, "A methodology for allocating transmission losses due to cross-border trades," *IEEE Transactions on Power Systems*, vol. 19, no. 3, pp. 1255–1262, 2004. [Online]. Available: <https://ieeexplore.ieee.org/document/1318658>
- [28] L. O. Camacho and I. J. Pérez-Arriaga, "An assessment of inter-tso compensation algorithms in the internal electricity market of the european union," *International Journal of Electrical Power Energy Systems*, vol. 29, no. 10, pp. 699 – 712, 2007. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0142061507000622>
- [29] P. Kattuman, R. Green, and J. Bialek, "Allocating electricity transmission costs through tracing: a game-theoretic rationale," *Operations Research Letters*, vol. 32, no. 2, pp. 114 – 120, 2004. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167637703000956>
- [30] Zhaoxia Jing and Fushuan Wen, "Discussion on the Proving of Proportional Sharing Principle in Electricity Tracing Method," in *2005 IEEE/PES Transmission Distribution Conference Exposition: Asia and Pacific*, 2005, pp. 1–5. [Online]. Available: https://www.researchgate.net/publication/224623724_Discussion_on_the_Proving_of_Proportional_Sharing_Principle_in_Electricity_Tracing_Method
- [31] A. J. Conejo, J. M. Arroyo, N. Alguacil, and A. L. Guijarro, "Transmission loss allocation: a comparison of different practical algorithms," *IEEE Transactions on Power Systems*, vol. 17, no. 3, pp. 571–576, 2002.

- [32] S. S. and T. Dhadbanjan, “Comparison of virtual flow approach with proportional sharing methods for tracing of network power flows,” *International Journal of Emerging Electric Power Systems*, vol. 12, no. 4, 21 Jul. 2011. [Online]. Available: <https://www.degruyter.com/view/journals/ijeeps/12/4/article-ijeeps.2011.12.4.2667.xml.xml>
- [33] K. Berg, “Power Flow Tracing: Methods and Algorithms - Implementation Aspects.” [Online]. Available: <https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2452452>
- [34] ECO2mix en. Last Accessed: 2020-12-06. [Online]. Available: <https://www.rte-france.com/en>
- [35] Australian energy market operator. Last Accessed: 2020-12-06. [Online]. Available: <https://aemo.com.au/>
- [36] Energy manager western washington university. Last Accessed: 2020-12-06. [Online]. Available: <https://energy.dudesolutions.com/BETA/?bbID=WWU1DASH>
- [37] UC Berkeley Energy Dashboard. Last Accessed: 2020-12-06. [Online]. Available: <https://engagementdashboard.com/ucb/ucb>
- [38] Carbon monitor dashboard. Last Accessed: 2020-12-06. [Online]. Available: <https://www.carbonmonitor.org.cn/>
- [39] Z. Liu, P. Ciaïis, Z. Deng, S. J. Davis, B. Zheng, Y. Wang, D. Cui, B. Zhu, X. Dou, P. Ke, T. Sun, R. Guo, H. Zhong, O. Boucher, F.-M. Bréon, C. Lu, R. Guo, J. Xue, E. Boucher, K. Tanaka, and F. Chevallier, “Carbon monitor, a near-real-time daily dataset of global CO₂ emission from fossil fuel and cement production,” *Scientific Data*, vol. 7, no. 1, p. 392, Nov 2020. [Online]. Available: <https://doi.org/10.1038/s41597-020-00708-7>
- [40] United states energy information administration. Last Accessed: 2020-12-06. [Online]. Available: <https://www.eia.gov/tools/>
- [41] B. Tranberg, A. B. Thomsen, R. A. Rodriguez, G. B. Andresen, M. Schäfer, and M. Greiner, “Power flow tracing in a simplified

- highly renewable european electricity network,” *New Journal of Physics*, vol. 17, no. 10, p. 105002, oct 2015. [Online]. Available: <https://doi.org/10.1088%2F1367-2630%2F17%2F10%2F105002>
- [42] Compressed sparse column matrix. Last Accessed: 2020-12-06. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.csc_matrix.html#scipy.sparse.csc_matrix
- [43] Lu decomposition of a sparse, square matrix. Last Accessed: 2020-12-06. [Online]. Available: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.linalg.splu.html#scipy.sparse.linalg.splu>
- [44] Pandas. Last Accessed: 2020-12-06. [Online]. Available: https://pandas.pydata.org/pandas-docs/version/0.25.0/getting_started/overview.html
- [45] Pandas dataframe. Last Accessed: 2020-12-06. [Online]. Available: <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html>
- [46] Plotly dash. Last Accessed: 2020-12-06. [Online]. Available: <https://dash.plotly.com/dash>
- [47] Plotly app callbacks. Last Accessed: 2020-12-06. [Online]. Available: <https://dash.plotly.com/basic-callbacks>
- [48] Plotly graph. Last Accessed: 2020-12-06. [Online]. Available: <https://dash.plotly.com/dash-core-components/graph>
- [49] M. Bayer, “Sqlalchemy,” in *The Architecture of Open Source Applications Volume II: Structure, Scale, and a Few More Fearless Hacks*, A. Brown and G. Wilson, Eds. aosabook.org, 2012. [Online]. Available: <http://aosabook.org/en/sqlalchemy.html>
- [50] ENTSO-E API. Last Accessed: 2020-12-06. [Online]. Available: <https://github.com/EnergieID/entsoe-py>
- [51] Dashboard bootstrap components. Last Accessed: 2020-12-06. [Online]. Available: <https://towardsdatascience.com/beginners-guide-to-building-a-multi-page-dashboard-using-dash-5d06dbfc7599>

- [52] Carbon intensity api. Last Accessed: 2020-12-06. [Online]. Available: <https://carbonintensity.org.uk/>
- [53] ENTSO-E public consultations. Last Accessed: 2020-12-06. [Online]. Available: https://www.entsoe.eu/network_codes/consultations/

Disclaimer

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

Stuttgart, 18 December, 2020

Sandesh Kenjana Ashok