SciGRID_gas: The filled INET gas transmission network data set

Release 1.0

J.C. Diettrich & A. Pluta & W. Medjroubi

Jul 23, 2020

CONTENTS

How to cite

J.C. Diettrich, A. Pluta, W. Medrjoubi
SciGRID_gas: The INET gas transmission network data set
DLR Institute for Networked Energy Systems
Germany
doi: ???

Impressum

DLR Institute for Networked Energy Systems
Carl-von-Ossietzky-Str. 15
26129 Oldenburg
Germany
Tel.: +49 (441) 999 060

LIST OF FIGURES

LIST OF TABLES

Summary

The goal of SciGRID_gas is to develop methods to create an automated process that can generate a gas transmission network data set for Europe. Gas transmission networks are fundamental for simulations by the gas transmission modelling community, to derive major dynamic characteristics. Such simulations have a large scope of application, for example, they can be used to perform case scenarios, to model the gas consumption, to minimize leakages and to optimize overall gas distribution strategies. The focus of SciGRID_gas will be on the European transmission gas network, but the principal methods will also be applicable to other geographic regions.

Data required for such models are the gas facilities, such as compressor stations, LNG terminals, pipelines, etc. One needs to know their locations, in addition to a large range of attributes, such as pipeline diameter and capacity, compressor capacity, configuration, etc. Most of this data is not freely available. However throughout the SciGRID_gas project it was determined, that data can be found and grouped into two fundamental different groups: a) OSM data, and b) non-OSM data. The OSM data consists of geo-referenced facility data that is stored in the OpenStreetMap (OSM) data base, and is freely available. However, the OSM data set currently contains hardly any other information than the location of the facilities. The Non-OSM data set can fill some of those gaps, by supplying information such as pipeline diameter, compressor capacity and more. Part of the SciGRID_gas project is to mine and collate such data, and combine it with the OSM data set. In addition heuristic tools are required to fill data gaps, so that a complete gas network data set can be generated.

Here, this document describes a non-OSM data set, called the “INET” data set, which is one of the fundamental building blocks of the SciGRID_gas project. The INET data set is test data sets, and a previous version of the INET data set has been published. However here, it is attempted to publish a data set where all missing values have been estimated. Hence this this is the filled INET data set. This document explains the origin and structure of the data set, and the processes undertaken to fill the missing values.

In this document, the chapter “Introduction” will supply some background information on the SciGRID_gas project, followed by the chapter “Data structure“, which gives a detailed description of the data structure that is being used in the SciGRID_gas project. Chapter “Data sources” describes the raw INET data set. This is followed by chapter “Heuristic”, which describes the steps that were implemented to fill the missing values of the INET data set, resulting in a gas transmission network data set.

The appendix contains a glossary, references, location name alterations convention and finishes with the table of country abbreviation.

CHAPTER

ONE

INTRODUCTION

SciGRID_gas is a three-year project funded by the German Federal Ministry for Economic Affairs and Energy [BMWi20] within the funding of the 6. Energieforschungsprogramm der Bundesregierung [BMWi11].

The goal of SciGRID_gas is to develop methods to generate and provide an open-source gas network data set and code. Gas transmission network data sets are fundamental for the simulations of the gas transmission within a network. Such simulations have a large scope of application, for example, they can be used to preform case scenarios, to model the gas consumption, to detect leaks and to optimize overall gas distribution strategies. The focus of SciGRID_gas will be the generation of a data set for the European Gas Transmission Network, but the principal methods will also be applicable to other geographic regions.

Both the resulting method code and the derived data will be published free of charge under appropriate open-source licenses in the course of the project. This transparent data policy shall also help new potential actors in gas transmission modelling, which currently do not possess reliable data of the European Gas Transmission Network. It is further planned to create an interface to [MMK16] or heat transmission networks. Simulations on coupled networks are of major importance to the realization of the German *Energiewende*. They will help to understand mutual influences between energy networks, increase their general performance and minimize possible outages to name just a few applications.

This project was initiated, and is managed and conducted by DLR Institute for Networked Energy Systems.

1.1 Project information

- **Project title**: Open Source Reference Model of European Gas Transport Networks for Scientific Studies on Sector Coupling (*Offenes Referenzmodell europäischer Gastransportnetze für wissenschaftliche Untersuchungen zur Sektorkopplung*)
- **Acronym**: SciGRID_gas (Scientific GRID gas)
- **Funding period**: January 2018 - December 2020
- **Funding agency**: Federal Ministry for Economic Affairs and Energy (*Bundesministerium für Wirtschaft und Energie*), Germany
- **Funding code**: Funding Code: 03ET4063
- **Project partner**: DLR Institute for Networked Energy Systems

DLR Deutsches Zentrum für Luft- und Raumfahrt
German Aerospace Center
Institute of
Networked Energy Systems

Gefördert durch:
Bundesministerium für Wirtschaft und Energie
aufgrund eines Beschlusses des Deutschen Bundestages

1.2 Background

As of today, only limited data of the facilities of the European Gas Transmission Networks is publicly available, even for non-commercial research and related purposes. The lack of such data renders attempts to verify, compare and validate high resolution energy system models difficult, if not impossible. The main reason for such sparse gas facility data is often the unwillingness of transmission system operators (TSOs) to release such commercially sensitive data. Regulations by EU and other lawmakers are forcing the TSOs to release some data. However, such data is sparse, and too often not clearly understandable for non-commercial operators, such as scientists.

Hence, details of the gas transmission network facilities and their properties are currently only integrated in in-house gas transmission models which are not publicly available. Thus, assumptions, simplifications and the degree of abstraction involved in such models are unknown and often undocumented. However, for scientific research those data sets and assumptions are needed, and consequently the learning curve in the construction of public available network models is rather low. In addition, the commercially sensitivity also hampers any (scientific) discussion on the underlying modelling approaches, procedures and simulation optimization results. At the same time, the outputs of energy system models take an important role in the decision making process concerning future sustainable technologies and energy strategies. Recent examples of such strategies are the ones under debate and discussion for the Energiewende [BundesregierungDeutschland20] in Germany.

In this framework, the SciGRID_gas project initiated by the research centre DLR Institute of Networked Energy Systems in Oldenburg aims at building an open source model of the European Gas Transmission network. Releasing SciGRID_gas as open-source is an attempt to make reliable data on the gas transmission network available. Appropriate (open) licenses attached to gas transmission network data ensures that established models and their assumptions can be published, discussed and validated in a well-defined and self-consistent manner. In addition to the gas transmission network data, the Python software developed for building the model SciGRID_gas are published under the GPLv3 license.

The main purpose of the SciGRID_gas project is therefore to open the door to new gas transmission network models and innovative ideas in energy system modelling by providing freely available and well-documented data on the European gas transmission network.

The input data itself is based on data available from openstreetmap.org (OSM) under the Open Database License (ODbL) as well as Non-OSM data gathered from different sources, such as Wikipedia pages, fact sheets from TSOs or even newspaper articles.

The main workload of this project is to:

- retrieve the OSM and Non-OSM data sets for the gas infrastructure
- merge all available data sets

- build a gas transmission component data set
- generate missing data using heuristic methods
- remove all gas facilities, that are not connected to pipelines.

The first step of the project was to collate a Non-OSM data set by searching the web for metadata that will be useful for the project. This included information, such as pipelines, compressors, LNG terminals, and their attributes, such as diameters, capacities, etc.

This data set is called the **InternetDaten** data set (INET). The raw data set has been published previously. However, here the missing values have been determined using heuristic processes. At later stages, descriptions of other data sources will follow, and will be made available on the project webpage.

This multi-stage release will allow us to easily and effectively incorporate feedback from potential users during the lifetime of the project. Those releases can be downloaded from the SciGRID_gas webpage with documentation, and can be seen as a snapshot of the current research project state.

Further information on the project can be found on the SciGRID_gas web page: https://www.gas.scigrid.de/pages/imprint.html.

The web page is maintained throughout the project lifetime, and will contain information on:

- General project information
- Contact details
- Presentations
- Bug/data fixes
- Data, code and documentation releases
- Publications.

As part of the SciGRID_gas webpage, one can also sign up to the SciGRID_gas newsletter by sending an email to

news.gas-subscribe@scigrid.de

1.3 Project goal

The overall goals of the SciGRID_gas project are:

- **Data output**: Creation of customisable gas transmission network data sets.
- **Open source**: Any one can download the data, make changes to it, pass it on to others, or even use it in commercial projects, as long as the SciGRID_gas project is mentioned as the original source of the data (CC by).
- **Application**: The outcome of the project can be used for a variety of scientific applications (e.g. sector coupling, entry-exit models, etc.).
- **Transparency**: The Python code, the documentation and the data (that can be passed on under copyright licences) is supplied.
- **Extendability**: Every user can extend the software code to their needs. However, we would encourage users to update and maintain the original git-repository and documentation for others.
- **Feedback**: Through constant data releases, it is hoped that the output data set will improve in quality and quantity by constantly incorporating feedback from the research community.

1.4 Document overview

This is an overview of this SciGRID_gas documentation, as this will help the user to better understand the overall project, its aims and the steps that were taken to obtain/model the resulting data set.

SciGRID_gas has been coded in Python, and hence, with that came the overall data structure that was selected for the project. As this is the most fundamental aspect for anyone wanting to use the data and the code, it is described first. In the chapter **Data Structure** we define the terms *Components*, *Elements*, and *Attributes*. We also give an overview on the internal workings of the SciGRID_gas source code.

A fundamental building block for the SciGRID_gas project is the data itself. Overall, we have classified the data into two groups: OSM and non-OSM data. The chapter **Data Sources** contains background information on the InterDaten (INET) data set only. Information is supplied on how the data was collated and how it was implemented into the SciGRID_gas data sets structure. In addition, an overview of the extent of the data will also be given for the data, e.g. the number of elements or the list of attributes that the data contains.

The raw INET data is "incomplete", as not all attribute values could be found for all elements. This results in missing elements and missing attribute values. Hence, the chapter **Heuristic Methods** will describe the different methods that have been implemented to estimate the missing attribute values.

The document also contains the chapter **Appendix** that contains sub-sections, such as *Glossary*, *References*, etc.

CHAPTER

TWO

DATA STRUCTURE

A well designed and documented data structure is fundamental in any large scale project. Good data structure in combination with tools, based on algorithms, improve the performance of any project output.

This structure needs to represent the gas flow facilities as good as possible, Hence, it needs to include components, such as pipelines, compressors, etc. A finite number of components have been identified, that are required as building blocks of a gas network. In addition each component will contain attributes, such as pipeline diameter, maximal operating pressure, maximal capacity, number of turbines etc.

It is anticipated, that the adopted data structure can be implemented in different types of gas flow models and will be used by the research community for topics, such as sector coupling or identifying gas transmission bottlenecks.

Within the SciGRID_gas project, the structure of the data model is part of classes defined within the Python code. Alterations may occur over the duration of the project, but it is envisaged, that those will be small, and that compatibility will be assured.

The goal of this section is to describe in details the data structure that has been adopted and implemented into the Python code. This will be important in understanding other aspects of this document, such as exporting the data into CSV files or generating missing values.

Prior to the description of the data structure, the overall pathway of the data flow within the SciGRID_gas project will be explained, as it is believed, that such overview will help the reader.

2.1 Data structure description

This section contains information about the SciGRID_gas data structure, the format, and the code that can be used to import publicly available data into the project, so that it can be used in subsequent steps. Paramount for an understanding of the data structure is a good understanding of the terminology used throughout this section and the document in general. Hence, terminology will be introduced in the following sub-section.

2.1.1 Terminology

Throughout this document certain terms will be used, which will be described below and summarized as a picture in Figure 2.1.

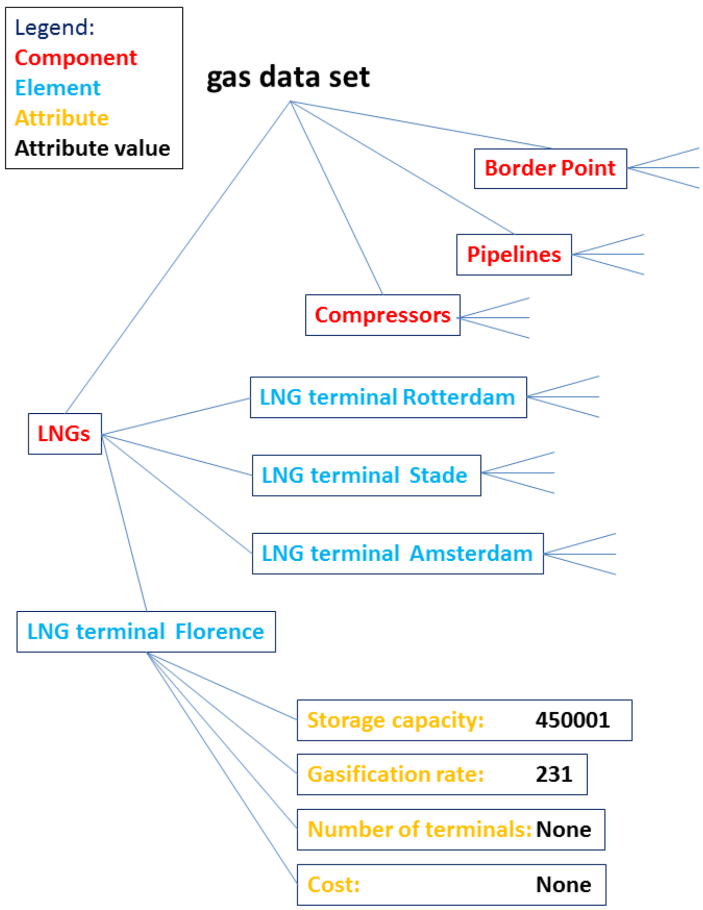

Figure 2.1: Data structure for the SciGRID_gas data set

Gas transmission network

The term "gas transmission network" describes the physical gas transmission grid. This does not include the distribution of gas through gas distribution companies, but includes the long distance transmission of gas from producer countries to consumer countries, as carried out by the Transmission System Operators (TSO) [Wik20g].

Gas component data set

The term "gas component data set" is used for all raw data of objects/facilities that have been loaded using SciGRID_gas tools into a Python environment. Gas component data sets are used as input into our SciGRID project. Several data sources can be loaded as gas component data sets, and then combined into a single gas component data set. However, not all elements (e.g. compressors) must be connected to pipelines, Hence, such a data set is referred to as a "gas component data set", and the emphasis is on the term **component**.

Gas network data set

A "gas component data set" can be converted into a "gas network data set", by connecting all non-pipeline elements to nodes and all nodes are connected to pipelines, and as part of the process all network islands have been connected or removed, resulting in a single network. Therefore the network contains nodes and edges which are coherently connected, and all objects with the exception of pipelines are associated with nodes in this network, whereas pipelines are associated with edges. Hence, the emphasis here is on the term **network**.

Component

There are several component types in a gas transmission network, such as compressors, LNG terminals, or pipelines. In Figure 2.1 they are coloured red. Hence, whenever the word "component" is mentioned, it refers to one of these components. There are roughly a dozen different components that will form a gas network data set. They will be briefly explained below.

Element

The term "element" refers to individual facilities, e.g. the LNG Terminal in Rotterdam, or the compressor in Radeland. In Figure 2.1 they are coloured blue. The first one is an element of the component LNG terminals, whereas the second one is an element of the component compressors. Hence, many elements make up a component. However, all elements are referring to different facilities by default. This means in a single network, one cannot have two elements of a component describing the same facility. The structure of elements is described below.

Attribute

"Attribute" is a term that is being used for the individual labels of the values that are associated with the elements. Examples for this term are gas "pipeline diameter", "maximum capacity", "max gas pipeline pressure", to name just a few and in Figure 2.1 they are coloured yellow. Overall there will be several hundred attributes in the SciGRID_gas project. However, the same attributes can occur in more than one component, e.g. "max flow capacity" exist for pipelines and also for compressors. Throughout the project, we have tried to keep the units of such attributes the same, so that there is no unit conversion required.

Attribute value

Each attribute has a value, most likely a number or a string. In Figure 2.1 they are coloured black. While boolean (*True*/*False*) is also allowed, more likely a "1" will stand for *True* and "0" for *False*. However, some attribute values might not be given in the data source, therefore a no value attribute value does exist. In the Python code it is termed *None*.

The Figure 2.1 depicts the relationships between the terms "gas data set", "component", "element", "attribute", and "attribute value". As can be seen, a single gas data set consists of several components, where each component contains several elements, and each element has several attributes, which each come with a value, where "None" stands for unknown value. The heuristic processes described in this document at a later stage will fill those "None" values with generated values.

Gas component types

A gas transmission network consists of different components, such as pipelines, compressors, etc. For the SciGRID_gas project a hand-full of components have been implemented, and will be described here briefly:

- *Nodes*: In a gas network, gas flows from one point to another point, which are given through their coordinates. All elements of all other components (such as compressor stations and power plants) have an associated node, which allows for the geo-referencing of each element. Overall the term "nodes" will be used throughout this document, as it aligns with graph theory aspects.
- *PipeLines*: *PipeLines* are one of the main components of the gas pipeline network. *PipeLines* allow for the transmission of the gas from one node to another. However, each pipe is unique. They might have different diameter, capacity or max pressure. In addition, a single *PipeLine* can connect several nodes. Therefore it could go from "Radeland" to "Bottrub" and then follow on to "Frankfurt". However, *PipeLines* do not need to connect more than 2 nodes, but can. The order of those nodes is important, and indicates the flow direction.
- *PipeSegments*: *PipeSegments* are almost identical to *PipeLines*, However, are only allowed to connect two nodes. Thus they have one start node and one end node, and are not passing via other nodes or other component elements in between, such as compressors or LNG terminals. Hence, any pipeline can easily be converted to multiple pipe-segments.
- *Compressors*: *Compressor* represent compressor stations, which are important. Gas travelling through the gas pipeline loses pressure due to friction on the pipeline walls and other factors. This will reduce the throughput of the gas amount. Hence, every so often (~ every 150 km), a compressor station is required, which increases the pressure of the gas, and Hence, allows the gas to flow through the gas pipeline. A gas compressor station contains several gas compressors units (turbines). Knowing the individual gas turbines is of an advantage, as those turbines can be combined in different ways, such as in series, or parallel, or combinations of those two options.
- *LNGs*: (LNG terminals and LNG storages) Some of the gas, which is being used throughout Europe, is supplied via ships to LNG terminals and LNG storage facilities. (From here onwards the acronym "*LNGs*" will be used instead "LNG terminals and LNG storage facilities".) As the transmission of gas would be extremely inefficient due to its volume, the gas state is changed to the liquid form (LNG gas), and then shipped. Ships arriving in Europe need special LNG terminals that can store LNG gas and subsequently re-gasify it. The storage and re-gasification of the gas are combined in the *LNGs* component and need to be part of any gas network for Europe.
- *Storages*: Part of the gas network will be gas storages. Gas storages are being used as gas pipeline capacities or gas production capacities might not be able to cover high demand periods, such as during the winter. Hence, large gas storage units are being filled during the summer periods while the overall demand is low, and if capacities of net supply allow it. This gas is then used during the winter period, and can compensate for shortcomings of the gas network or gas supply. Almost every country has their own gas storage units, ranging from smaller units to compensate for daily fluctuations to larger units, which compensate seasonal fluctuations. For the SciGRID_gas model the larger seasonal storage units are of more importance than the smaller ones,

as we are interested in the transmission gas pipeline network. However, any gas storage can be added and implemented into the gas network data set.

- *Consumers*: Part of the gas pipeline network is the knowledge of gas demand. Gas is added to the network at LNG terminals and European boundary cross border points. One type of users is the gas power plants. These can be added to the SciGRID_gas model, as this will specify local gas demand. In addition other consumers, such as city gas providers and large industries can also be added to the network data set.
- *Production*: These can be wells inside a country where gas is pumped out of the ground. Most of the gas used in Europe comes from outside of the EU, However, there are several smaller gas production sites scattered through Europe.
- *BorderPoints*: BorderPoints are cross border points (between different countries), which are mostly for the purpose of accounting the gas flow. Most large gas pipelines have cross border stations, e.g. Ellund (lat/long: 54.80181, 9.289079) at the border between Germany and Denmark, with gas facilities on both sides of the border.
- *EntryPoints*: These are special border points, as they are at the borders of the European Union and will be the gas entry points for the SciGRID_gas model data set.
- *InterConnectionPoints*: These are points between gas transmission operators, and will be found mainly within Europe, in particular at country borders. However, they can also be found within a single country, if there is more than one gas transmission operator.

Element structure

As described above, elements are describing individual facilities, such as compressors or LNG terminals. However, the overall structure of those elements is the same for all elements of all components. The overall structure of those elements is described in the following part:

- *id*: A string, that is the ID of the element, and must be unique.
- *name*: A string that is the name of the element, such as "Compressor Radeland".
- *source_id*: A list of strings that are the sources of the element. As several elements from different sources could have been combined in a single element, one might need to know which are the original ids of the original sources.
- *node_id*: This is the ID of a geo-referenced point to which an element of the network is associated to. For a compressor, this will be just a single node_id, However, for a gas pipeline, that starts at one point and finishes at a different point, this entry would be a list of at least two node_id values.
- *lat*: This is the latitude value of an element. For pipelines, lat is a list of latitude values if known. The geo-referenced projection of the element that is being used in the SciGRID_gas project is: World Geodetic system 1984 (epsg:4326).
- *long*: The longitude, analogue to lat.
- *country_code*: This is a string indicating the official 2-digit country code (Alpha-2 code, see Chapter 6.5 for list of countries and their code). It represents the location of the element. As pipelines can pass through more than one country, the country code for pipes is the list of country codes of the countries the pipeline is passing through.
- *comment*: This is an arbitrary comment that is associated with the element.
- *tags*: This dictionary is reserved for OpenStreetmap data. It contains all associated key:value-pairs of an OpenStreetMap item.

In addition, there are three further groups of attributes to each element. Throughout the SciGRID_gas project, they have been coded as "dictionaries". They are called:

- *param*
- *method*
- *uncertainty*.

The structure within each dictionary is the same, However, their meaning is different. First of all the dictionary *param* (short for "parameter") contains a list of attributes and their values. This list of attributes will be different for each component. For the component *PipeLines* they might be pipeline diameter, max pipeline pressure, and max pipeline capacity. For the component *Compressors* they might be , such as number of turbines, overall turbine power, energy source of turbine and more.

So the other two attribute dictionaries are *method* and *uncertainty*. Each of those two dictionaries contains exactly the same list of attributes as the "param" dictionary. However, their attribute values reflect the name of the dictionary. E.g. the attributes in the dictionary *method* contain the information on the method used to derive the attribute value that is stored in the param dictionary. Here methods of value generation can include heuristic methods names (in form of strings) that have been implemented in the SciGRID_gas project. However, if attribute values are not being generated by the SciGRID_gas project, but originate from one of the input data sources, then the attribute values in the *method* dictionary is set to "raw".

Similar is the content of the *uncertainty* dictionary. It contains information on the uncertainty of the attributes from the *param* dictionary of that component. Again all attributes listed in the *param* dictionary are also present in the *uncertainty* dictionary. The attribute values here reflect the uncertainty of the attribute. Here, it is assumed, that attributes with a method of "raw" have an uncertainty of zero. Only for those attributes, which were generated during heuristic SciGRID_gas methods an uncertainty larger than zero will be specified.

2.2 Summary

The SciGRID_gas software is designed to construct a gas transmission network data set form different open source gas component data sets. The gas transmission data set needs to be available and stored in a precise and predefined way, which was described in this section. We have identified several *component*-types of a gas transmission network grid, like pipelines, compressor stations, LNG-terminals, etc. Each specific facility that falls under such a component is considered an *element* of that component. Each element is described by a list of *attributes* and correspondent *attribute values*.

CHAPTER THREE

DATA SOURCES

Two thirds of the gas used in Europe is imported from non-EU states, and all gas required for the consumption needs to be distributed through the existing gas transmission pipelines in Europe. In the future gas consumption might rise, leading to additional pressure on the current infrastructure. In addition, gas facilities could play a vital role in reducing CO_2 emission, as excess electricity could be converted to gas, that could be stored and transmitted throughout Europe with the existing gas network. Hence, a reliable network data set for the European transmission network is essential. The data required for such models ranges from pipeline diameter, gas pressure within the pipeline, actual pipeline length, pipeline capacity, and underground storage volume to name just a few.

However, such data is the property of the transmission system operators (TSOs) and is therefore generally not freely available in the form and depth that is required for modelling purposes. The major reason for the difficulty of obtaining of such data is that most of the gas network infrastructure, namely pipelines, is buried underground. Thus a pipeline diameter is hard to estimate locally. In addition, almost all of the data is commercially sensitive.

However, there is a public drive to gather such data and subsequently make it available. The major platform through which this is occurring is the Open Street Map database [Hel18]. OSM is a geo-referenced database through which people can supply geo-referenced information on all man-made and natural structures, ranging from mountains to buildings. To achieve this, people throughout the world wander the globe and geo-reference everything that they can find. This also includes gas-pipeline markers, compressor stations or LNG terminals. However, the major problem remains that one cannot measure or estimate the diameter of the underground pipelines, or the number and size of the compressor turbines, as compressors are within buildings, which are fenced off. Hence, such information is hardly supplied to the OSM platform.

Nevertheless, some data is made available by gas transmission network operators, through different channels. E.g. information on the size and number of compressors could be made public through a press release, as part of a refurbishment. An example is given below (https://www.maz-online.de/Lokales/Teltow-Flaeming/Neue-Verdichterstation-entsteht-in-Radeland):

> "Die Eugal-Pipeline dient dazu, Gas aus der neuen Ostseepipeline Nord Stream 2 bis zur tschechischen Grenze zu leiten. 275 Kilometer von ihr verlaufen in Brandenburg. Grundsätzlich soll die neue Leitung parallel zur bestehenden Opal-Pipeline gebaut werden."

In addition some information can be found on company web pages, (https://www.open-grid-europe.com/cps/rde/SID-752BB6B5-E0A975F2/oge-internet-preview/hs.xsl/NewsDetail.htm?rdeLocaleAttr=en&newsId=50190C3B-E14F-4685-9E64-E40EEAB57A28):

> Open Grid Europe (OGE) is investing roughly EUR 150 million at its compressor station in Werne to improve the security and flexibility of energy supply for North Rhine-Westphalia and Germany. The upgrade of the station, which is one of the hubs of the pipeline network, will allow gas flows to be switched (reversed) from north to south and south to north. In addition, OGE is preparing the station for the upcoming transition from L- to H-gas. Through this fitness programme, the station's transmission capacity will increase by about 500,000 to 6.5 million m^3/h, which is equivalent to the annual consumption of more than 2,100 single-family homes. The project, which is due for completion at the end of 2018, is fully on track."

The data available can be separated into two different groups:

- OSM data: Data can be found in the OSM data base. OSM data is well geo-referenced, but contains little meta-information (information on the facility attributes, such as pipeline diameter or pipeline capacity). OSM data is very helpful to obtain accurate routes of pipelines.
- Non-OSM data: Non-OSM data have in general lower geographical accuracy but contain a lot of meta-information. Unfortunately, such information is only known for a few facilities. One exception to this rule are shapefiles from TSOs. They are rare, but well geo-referenced. However, the resolution of the meta information can vary from TSO to TSO.

One of the main challenges for SciGRID_gas is that, gas transmission data is incomplete and accumulated from different sources. Also such different sources can have different properties for one and the same facility. Hence, it is important to know, which data set supplies which information. Hence, this chapter here will introduce the relevant data sets (e.g. INET), starting off with the components, the elements for each component and then the attributes for each element.

3.1 Non-OSM data

Non-OSM data includes data from internet research, TSO press releases, TSO transparency platform, TSO public data, national open-source gas network data sets[1], etc.

Some of the TSO information had to be made available due to EU-regulations. Other information has been made public as part of a company's self presentation and advertisement. The information used by the SciGRID_gas project focuses on:

- the quality of the data
- the format of the data
- the level of representation of the data
- and the copyright restrictions on the data.

In addition, each data source is unique. Source specific tools need to be developed, so that all data sources can be made accessible for the SciGRID_gas project in the format as described in later chapter releases.

A significant portion of the project was spent on finding non-OSM data sets . Further data sources might be available, but unknown to the authors. If the authors are made aware of additional sources, the project will try to incorporate those, as this would only increase the depth of the data available and increase the applicability of the gas network data set and model.

Non-OSM data sources are very specific, addressing only certain aspects of the entire gas infrastructure. E.g. the GIE [GasIEurop20] data set supplies information on the daily gas flow in and out of gas storages in LNG terminals. However, they fall short on specifying the fundamental information of the actual physical location. Other data sets, such as the LKD [FMWP+17] data set is quite detailed in respect of pipelines, compressors and consumptions, however, only available for Germany.

Hence, the main task is to look closely at each data source, distil which data attribute values can be used, how it can be downloaded and incorporated into our SciGRID_gas model, and identify the copyright restrictions on the data source.

Due to copyright regulations, there are roughly two groups of data:

- Non copyright restrictive data (N-CRRD): here the copyright does not restrict the download, use and distribution of the data.
- Copyright restrictive data (CRRD): here the data can be downloaded and used internally, but not re-distributed to others.

[1] An entire gas network data set is only available from the UK, see https://www.nationalgridgas.com/land-and-assets/network-route-maps'.

The following is a list of the data sources that will be used throughout the project and an indication into which group of copyright restriction they fall:

- **OSM** (https://www.openstreetmap.org) (N-CRRD)
- **GB** (https://www.nationalgridgas.com/land-and-assets/network-route-maps) (CRRD)
- **NO** (https://www.npd.no/en/about-us/information-services/available-data/map-services/) (N-CRRD)
- **LKD** (https://tu-dresden.de/bu/wirtschaft/ee2/forschung/projekte/lkd-eu) (N-CRRD)
- **ENTSOG** (https://transparency.entsog.eu/) (CRRD)
- **EMap** (https://www.entsog.eu/sites/default/files/2020-01/ENTSOG_CAP_2019_A0_1189x841_FULL_401.pdf) (CRRD)
- **GIE** (https://www.gie.eu/) (N-CRRD)
- **GSE** (https://www.gie.eu/index.php/gie-publications/databases/storage-database) (N-CRRD)
- **IGU** (https://www.igu.org/) (CRRD)
- **GasLib** (http://gaslib.zib.de/) (N-CRRD)
- **INET** (see `Refs_InternetData`) (N-CRRD).

Each data set and source comes with a different copyright regulation. The copyright can be rather non-restrictive (e.g. INET) or can be restrictive (IGU). It is attempted to use only freely available data, so that such data can be re-distributed. In more restrictive data cases (IGU, GB), it is not allowed to download the data and distribute it to others. However, it is allowed to let other potential users know of the location of such data and supply them with tools, that allow them to carry out the same data download and subsequent incorporation of the data into a gas network data set.

Note:

In case that other users are aware of other data sources, that might be useful to this project, please get in touch and supply us with a brief description of the data and the location of such data, so that additional tools can be developed to incorporate the data in this project. Please use the following email address: developers.gas(at)scigrid.de

3.2 The InternetDaten (INET) data set

Here we have another one [Mandela] and so on

This section contains information on the content and nature of the so called **InternetDaten** data set (**INET**), how this data was generated, its format, and the tools that have been designed to import the INET into the SciGRID_gas project.

The INET data set is a special data set, as it was collated from many www sources and the information has been collated into CSV files. Please note, that throughout the project, the separator within CSV files will need to be ";". This section here will give an overview on the INET, its components and how the data is stored in INET specific CSV files. Further the processing of the data into Python will be described.

Prior to the description of those processes, a general overview of the INET data set is given first, so that the reader gets a better understanding of the size and depth of the data.

3.2.1 Overview of the INET data set

The INET data set contains geographical and meta information on gas facilities that were found through Internet searches. The data originated from www pages, such as Wikipedia, gas transmission system operators, fact sheets and press releases and more. Hence, most of the data had to be extracted manually out of text pages. To make this data available throughout the project, the data is being stored in CSV files. This also allows others to add additional properties and values to the INET data set at any stage. Tools have been written to load the INET from those CSV files and make them accessible throughout the project[1].

The Table 3.1 summarises the number of elements for each component that has been found so far. However, this does not imply that there is no missing data. In contrary, this data set comes with a lot of missing data:

Table 3.1: INET component summary.

Component Name	Count
BorderPionts	119
Compressors	249
ConnectionPoints	0
Consumers	0
EntryPoints	37
InterConnectionPoints	118
LNGs	32
Nodes	907
PipeSegments	920
Production	0
Storages	199

In addition, a map (see Figure 3.1) visualizes these components for Europe in the figure below.

[1] These tools will be made available during an upcoming release, where the INET data set will be jointly released with the GIE data set most likely through our project web page.

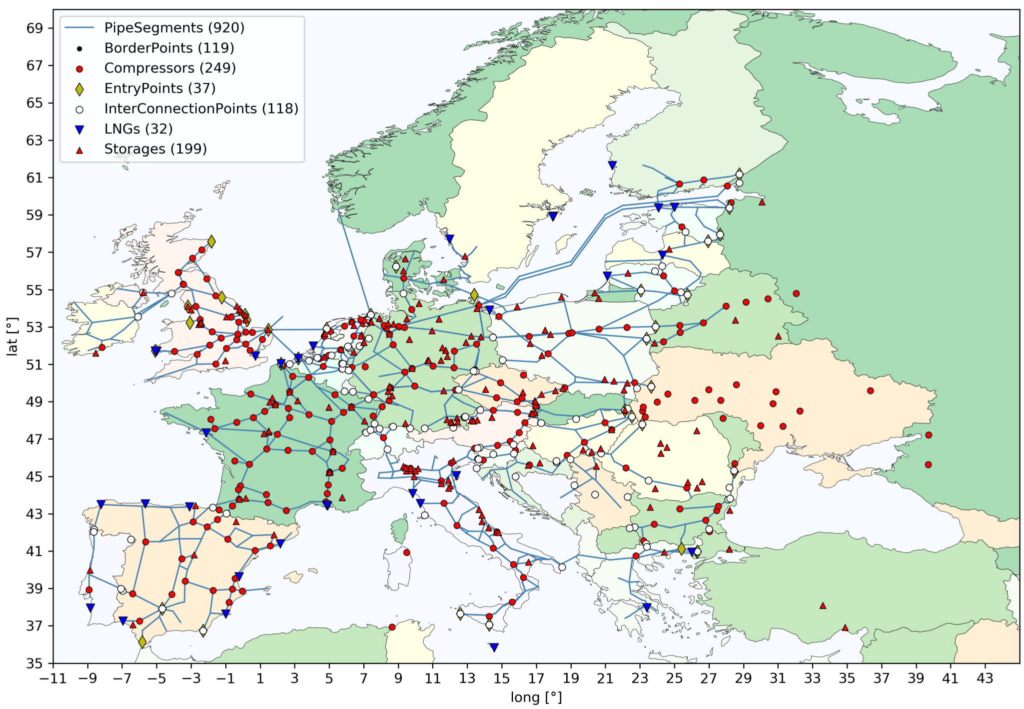

Figure 3.1: Map of the INET data set. The legend contains the number of elements for each component.

3.2.2 Origin of the data

As has been stated before, the resulting INET data set originated from text sources found on the www. Here, for the pipeline JAGAL [Wik20h] an example from a Wikipedia page is given (https://en.wikipedia.org/wiki/JAGAL):

Commissioned	1999
Technical information	
Length	338 km (210 mi)
Maximum discharge	24 billion cubic meters per year
Diameter	1,200 mm (47 in)
No. of compressor stations	1
Compressor stations	Mallnow

Figure 3.2: Screenshot of part of the Wikipedia page for the pipeline JAGAL.

As one can see, some information is given, such as location name of the compressor (Mallnow), total pipeline length (338 km), pipeline diameter (1200 mm) and maximum pipeline capacity (24 billion $\mathrm{m}^3\mathrm{a}^{-1}$). This is the information that is manually extracted from such pages and put into the CSV files. Other sources of such data are: gas facility operator press releases, fact sheets and other documents, newspaper articles, federal departmental web pages etc.

To collate the data in an orderly manner, a system of CSV files has been created and will be described below.

3.2.3 INET CSV file description

Each component of the INET data set is represented by a single CSV file. Each of those files has a single header line, and it is very important to know, that entries in the first line should only be changed if one knows, what one is doing, as the first row labels (the actual words) are imported and used as variable names in the SciGRID_gas project Python programs. Hence, if certain labels would be missing, the program would fail. In addition, each label needs to be unique within each file. It is advised to incorporate the units of the attributes into the label name, where possible.

Nodes.csv file

This is a unique file, and contains information on the nodes of the INET data set. Nodes are such entities, to which and from where pipelines can run, or to which other facilities can be associated to. Nodes supply information on a location including its name, its latitude and longitude, and the country in which it is located. Additionally, they supply information on how exact is the location of the node could be determined. The nodes component data is supplied to the SciGRID_gas data model only via a single CSV file.

- *id*: This is a unique id of a node of type string. Most likely this will be the name of an element. White spaces are allowed in this string.
- *comment*: Here the user can place additional information on the location node.
- *country*: Here the user needs to write the 2 letter abbreviation of the country, in which this node is located (see Table 6.5 for a list of country codes used).
- *lat*: This is a number of the best estimate of the latitude of the location. Best latitude value (and long value) were attempted to be generated by using metadata of the facility node and satellite maps. Using the satellite data, address information etc., it was tried to visually find the facility of the node.

- *long*: This is the corresponding best estimate of the longitude of the location derived during the same process as described under **Lat**.
- *node_id*: This is an identifier of a node.
- *source_id*: This is a unique identifier describing the source of the element. Here **INET** is the abbreviation for **InternetDaten**. Hence, all elements originating from the INET data set starts with the letters **INET**.
- *name*: A string containing the name of the location. It is allowed to contain white spaces.
- *exact*: This is a number in the range of 1 to 5, indicating how accurate the lat/longs were supplied for each node. Options are as follow:
  - "1": The exact location of this node is known, as one was able to verify the facility through satellite data.
  - "2": Here the lat/long is not known exactly, however one assumes that the location is within a small region (e.g. Krummhoern), Hence, not being much larger than 10 km.
  - "3": Here so little is known about the exact location, and one only knows, that the location is within a large region (e.g. Hamburg). Hence, the actual location could be out by 10 km or more but less than 100 km.
  - "4": Here so little is known about the exact location, and one only knows, that the location is within a state (e.g. Niedersachsen). Hence, the actual location could be out by 100 km or more but less than 1000 km.
  - "5": Here so little is known about the exact location, and one only knows, that the location is within a country (e.g. Ukraine). Hence, the actual location could be out by 1000 km or more.

All other components need two files, the location file and the metadata file, which will be described next.

Compressor CSV meta file

The compressor file ("Compressors.CSV") contains all the metadata for all known compressor stations.

In addition to the seven mandatory columns introduced above, the following columns are currently implemented, and contain the following data:

- *end_year*: Integer number of the year, when the operation of this element stopped. If it contains a value of "None", than it is still operational.
- *start_year*: Integer number of the year, when the operation of this element started. If it contains a value of "None", then this year is not known.
- *operator_name*: This is a string, containing the name of the operator of the compressor station.
- *pipe_name*: This is a string containing the label of the pipeline that the compressor is connected to.
- *source*: Information on where the information of this element originated from.
- *is_H_gas*: This is a boolean, indicating if the gas is of high calorific gas type ("1") or of low calorific gas ("0").
- *max_cap_M_m3_per_h*: This is a number, which is the overall capacity of gas that can be compressed by the compressor station. Values need to be supplied in units of $[\mathrm{Mm}^3/\mathrm{h}]$.
- *max_pressure_bar*: This is a number, which is the maximum pressure that the gas can be compressed to. Values need to be supplied in units of [bar].
- *max_power_MW*: This is a number, which is the sum of the power of all compressor units that are installed at the compressor station. Values need to be supplied in units of [MW].
- *num_turb*: This is the number of compressor turbines installed at the compressor facility. This number is including the reserve turbine unit.
- *turbine_fuel_isGas_1*: This is a boolean, indicating if the turbine is powered by gas ("1"), or by electric ("0").

- *turbine_type_1*: This is a string containing additional information on the type of turbine unit, e.g. name of the turbine.
- *turbine_power_1*: This is a number, indicating the power of the turbine unit. The value needs to be supplied in units of [MW].
- *turbine_fuel_isGas_2*: This is the information for the second turbine unit. Same as for *turbine_fuel_isGas_1* applies. Currently up to 6 individual units can be stored in the database, Hence, the last digit in the identifier can be as large as 6.
- *turbine_type_2*: This is the information for the second turbine unit. Same as for *turbine_type_1* applies. Currently up to 6 individual units can be stored in the database, Hence, the last digit in the identifier can be as large as 6.
- *turbine_power_2*: This is the information for the second turbine unit. Same as for *turbine_power_1* applies.
- . . .
- *turbine_power_6*: This is a number, indicating the power of the sixth turbine unit. The value needs to be supplied in units of [MW].

LNG CSV meta file

The LNG terminal metafile ("LNGs.CSV") contains all the metadata for the LNG terminals.

Next to the above described first seven columns the following columns are currently implemented, and contain the following data:

- *end_year*: Integer number of the year, when the operation of this element stopped. If it contains a value of "None", then it is still operational.
- *start_year*: Integer number of the year, when the operation of this element started. If it contains a value of "None", then this year is not known.
- *source*: Information on where the information of this element originated from.
- *max_workingGas_M_m3*: This is a number, indicating the maximum amount of liquid gas that can be stored, after having been brought in by ship. Values need to be supplied in units of $[\mathrm{Mm}^3]$
- *max_cap_store2pipe_M_m3_per_a*: This is a number, indicating the maximum amount of gas that can leave the LNG terminal. This gas is in gas phase. Values need to be supplied in units of $[\mathrm{Mm}^3/\mathrm{a}]$

BorderPoints CSV meta file

The metafile for * BorderPoints * elements ("BorderPoints.CSV") contains all the metadata for each border point.

Next to the above described first seven columns the following columns are currently implemented, and contain the following data:

- *pipe_name*: This is a string, the name of the pipe that is passing the border point.
- *end_year*: Integer number of the year, when the operation of this element stopped. If it contains a value of "None", then it is still operational.
- *start_year*: Integer number of the year, when the operation of this element started. If it contains a value of "None", then this year is not known.
- *source*: Information on where the information of this element originated from.

EntryPoints CSV meta file

The metafile for *EntryPoints* elements (“EntryPoints.CSV”) contains all the metadata for entry points of gas pipelines into Europe.

Next to the above described first seven columns the following columns are currently implemented, and contain the following data:

- *end_year*: Integer number of the year, when the operation of this element stopped. If it contains a value of “None”, then it is still operational.
- *start_year*: Integer number of the year, when the operation of this element started. If it contains a value of “None”, then this year is not known.
- *source*: Information on where the information of this element originated from.

InterConnectionPoints CSV meta file

The *InterConnectionPoints* metafile (“InterConnectionPoints.CSV”) contains all the metadata for interconnection points between the different operators within Europe.

Next to the above described first seven columns the following columns are currently implemented, and contain the following data:

- *end_year*: Integer number of the year, when the operation of this element stopped. If it contains a value of “None”, then it is still operational.
- *start_year*: Integer number of the year, when the operation of this element started. If it contains a value of “None”, then this year is not known.
- *source*: Information on where the information of this element originated from.
- *pipe_name*: This is a string, the name of the pipe that is passing the border point.

Storages CSV meta file

The metafile “Storages.CSV” contains all the metadata for gas storage within Europe.

Next to the above described first seven columns the following columns are currently implemented, and contain the following data:

- *access_regime*: String indicating the access of the storage facility, TPA or not TPA (nTPA), and could be used for heuristic processes at a later stage.
- *end_year*: Integer number of the year, when the operation of this element stopped. If it contains a value of “None”, then it is still operational.
- *start_year*: Integer number of the year, when the operation of this element started. If it contains a value of “None”, then this year is not known.
- *store_type*: This is a string, indicating the type of storage, such as “Leeres Gas Feld” (empty gas field), “Salz Kaverne” (salt cavern) etc.
- *source*: Information on where the information of this element originated from.
- *is_H_gas*: This is a boolean, that indicates if the gas is of high calorific nature (“1”) or of low calorific nature (“0”).
- *is_onShore*: this is a number, indicating if this gas store is on land or not. Options are “1” and the gas store is on land, whereas the second option “0” indicates, that the gas store is not on land, Hence, will be off shore.

- *operator_name*": String, containing the name of the operator.
- *max_workingGas_M_m3*: A number indicating the maximum amount of gas that can be stored and worked with in that gas field. Values need to be supplied in units of $[\mathrm{Mm}^3]$.
- *max_cap_store2pipe_M_m3_per_d*: A number indicating the maximum amount of gas that can move from the gas store into a gas pipe. Values need to be supplied in units of $[\mathrm{Mm}^3\mathrm{d}^{-1}]$.
- *max_cap_pipe2store_M_m3_per_d*: A number indicating the maximum amount of gas that can move from the gas pipeline into a gas store. Values need to be supplied in units of $[\mathrm{Mm}^3\mathrm{d}^{-1}]$.

PipeSegments CSV meta file

The metafile "PipeSegments.CSV" contains all the metadata for gas pipe lines within Europe.

Next to the above described first seven columns the following columns are currently implemented, and contain the following data:

- *is_bothDirection*: This is a boolean with value of '1' or '0'. If set to '1', then the gas pipeline can be operated in both directions, whereas if set to '0', then the gas can only flow from the start point to the end point. Hence, here the order of the point_lables in the pipes file is important.
- *length_km*: This is the overall length of the pipeline, and NOT of the segment. The value needs to be supplied in units of [km].
- *diameter_mm*: This is the diameter of the pipe in units of [mm].
- *max_pressure_bar*: This is the maximum pressure of the gas within the gas pipeline in units of [bar].
- *max_cap_M_m3_per_d*: This is the maximum annual gas volume that the pipe can transmit in units of $[\mathrm{Gm3/d}]$.
- *num_compressor*: This is the number of compressors along the pipeline.
- *end_year*: Integer number of the year, when the operation of this element stopped. If it contains a value of "None", then it is still operational.
- *is_H_gas*: This is a boolean, that indicates if the gas is of high calorific nature ("1") or of low calorific nature ("0").
- *source*: Information on where the information of this element originated from.
- *lat_mean*: Average mean latitude value of the pipe-segment.
- *lat_mean*: Average mean longitude value of the pipe-segment.

3.2.4 INET data density

Now that the structure of the INET has been described in detail above, the "data density" of data will be presented. Here "data density" is defined as follow: it is the ratio of the number of usable (not missing, e.g. filled or raw values) attribute values over the number of all elements of the component. Supposedly the INET would have two LNG terminals. One of the facilities has a storage volume, whereas the other one does not. Hence, the data density would be 50% for the attribute storage volume. Here, the data density for the most relevant attributes will be given next for all components. At a later stage, missing values might be estimated through heuristic processes, to complete the data set.

PipeSegments

Overall there are 920 *PipeSegments* elements in the INET data set.

Table 3.2 summarizes the data densities for the most important pipe-segment attributes:

Table 3.2: INET *PipeSegments* data density

Attribute name	Data density [%]
diameter_mm	43
is_H_gas	94
is_bothDirection	9
length_km	100
max_cap_M_m3_per_d	16
max_pressure_bar	31
num_compressor	3

Compressors

Overall there are 249 *Compressors* elements in the INET data set. The data densities for the most important attributes is given in Table 3.3 below:

Table 3.3: INET *Compressors* data density

Attribute name	Data density [%]
is_H_gas	99
max_cap_M_m3_per_d	7
max_power_MW	16
max_pressure_bar	7
num_turb	15
operator_name	11
pipe_name	10
turbine_power_1_MW	15
turbine_power_2_MW	147
turbine_power_3_MW	9
turbine_power_4_MW	3
turbine_power_5_MW	1
turbine_power_6_MW	0
turbine_fuel_isGas_1	14
turbine_fuel_isGas_2	14
turbine_fuel_isGas_3	10
turbine_fuel_isGas_4	7
turbine_fuel_isGas_5	1
turbine_fuel_isGas_6	0
turbine_type_1	9
turbine_type_2	9
turbine_type_3	6
turbine_type_4	3
turbine_type_5	1
turbine_type_6	0

Nodes

Overall there are 907 nodes. As described above, the information supplied is an "id", latitude and longitude values, the country code and a value indicating the accuracy of the node location. Hence, Table 3.4 summarizes the relative number of nodes within the possible value range of 1 to 5:

Table 3.4: Summary for the attribute "exact" of component *Nodes* of the INET data set.

Exact value	ration of data with exact value [%]
1	20
2	45
3	9
4	12
5	14

Storages

Overall there are 199 storage elements in the INET data set. The data densities for the most important attributes is given in Table 3.5 below:

Table 3.5: INET *Nodes* data density

Attribute name	Data density [%]
access_regime	90
is_H_gas	12
is_onShore	22
max_cap_pipe2store_M_m3_per_d	77
max_cap_store2pipe_M_m3_per_d	77
max_workingGas_M_m3	80
operator_name	99
source	95
store_type	94

BorderPoints

Overall there are 119 *BorderPoints* elements in the INET data set. The data densities for the most important attributes are given in Table 3.6 below:

Table 3.6: INET *BorderPoints* data density

Attribute name	Data density [%]
pipe_name	9

EntryPoints

Overall there are 37 *EntryPoints* elements in the INET data set. This component does not contain any further major attribute of interest.

InterConnectionPoints

Overall there are 118 *InterConnectionPoints* elements in the INET data set. The data density for the most important attributes is given in Table 3.7 below:

Table 3.7: INET *InterConnectionPoints* data summary

Attribute name	Data density [%]
pipe_name	15

LNGs

Overall there are 32 *LNGs* elements in the INET data set. The data densities for the most important attributes is given in Table 3.8 below:

Table 3.8: INET *LNGs* data density

Attribute name	Data density [%]
max_cap_store2pipe_M_m3_per_d	90
max_workingGas_M_m3	97

Overall one can see that a lot of data has been collated and is made available through the INET data set. However, as presented in the data density tables, a lot of attributes have low data density. Chapters later in this documentation will demonstrate how missing values can be estimated, so that the generated SciGRID_gas data set has a data density of 100% for each attribute.

3.2.5 Copyright and disclaimer for the INET data set

Copyright

A list of the sources used for the generation of the INET data set can be found in Chapter 6.3.

Disclaimer

The INET data set is supplied on a best-effort basis only, using available information as documented gathered from the Internet. While every effort is made to make sure the information is accurate and up-to-date, we do not accept any liability for any direct, indirect, or consequential loss or damage of any nature–however caused–which may be sustained as a result of reliance upon such information.

3.3 Data summary

SciGRID_gas is based on open source data. To generate a gas pipeline network data set, one needs to access different data sets that were found throughout the project and presented here. Emphasis was given to depict the number of elements per component and the data density for each data set.

3.4 Summary

Gas component data sets come in different forms, licenses, formats and detail. The SciGRID_gas project can process such data and combine them to a consistent and reliable network data set.

The underlying gas component data sets were categorized into two different groups:

- OSM data: This is data originating from the OSM data base, containing well geo-referenced locations of gas facilities, such as pipe locations or gas storage facilities. However, it comes with very few meta information.
- Non-OSM data: These are all other data sources, which can "supply" detailed information on some of the gas facilities attributes. However, this information is sparse, as published only for a few facilities. Here, the INET data set was introduced as an example of the non-OSM data set, and the pathway of converting the raw data from the www into SciGRID_gas project component structure.

Here detailed information on one or several data sources have been given, and should be used as a reference for later data processes.

CHAPTER

FOUR

HEURISTIC ATTRIBUTE VALUE GENERATION

Gas facility data sources have been described in Chapter 3. However, those data sources might not contain values for all attributes, and hence, those values need to be generated. This chapter here will describe the current implemented heuristic methods that can be used to estimate any missing attribute value.

4.1 Attribute value generation

The SciGRID_gas project has been set up to deliver a data set of the European gas transmission network. Despite merging several data sources of gas facilities, the resulting gas component data set will contain a large number of missing values. This section here describes how missing attribute values can be generated through different heuristic methods. The tools developed here have been designed for the non-OSM data sets. However, they should be applicable to any gas component data set.

In this section, the problem of missing data is described with the help of some synthetic data set. This is followed by the description of the heuristic methods that have been implemented, and the general pathway that the user needs to undertake to eliminate missing values.

Problem description

Figure 4.1 depicts the problem that the SciGRID_gas project is facing. The data sets contain different elements of different components, where many attributes values could not be found. The example given in the following sections shall depict the gas pipelines, where the attributes in question are *diameter*, *capacity* and *pressure*. The data is summarized in Table 4.1.

Table 4.1: Summary of data of the nine sample pipelines from Figure 4.1.

Pipeline name	*capacity* [M m ^{3} d $^{-1}$]	*pressure* [bar]	*diameter* [mm]
Jagal	76	80	1200
RHG		84	800
Midal 1	40		900
Midal 2	50		1000
Midal 3	35		800
Midal 4	34		800
Stegal_1			800
Stegal_2			900
Wedal	27		

As can be seen, all but one *PipeSegments* contain an attribute value for the attribute *diameter*. For the attribute *capacity* three values are missing, and for the nine *PipeSegments*, only two have a value for the attribute *pressure*. So the task will be to fill as many missing attribute values as possible. The corresponding data densities for the attributes *capacity*,

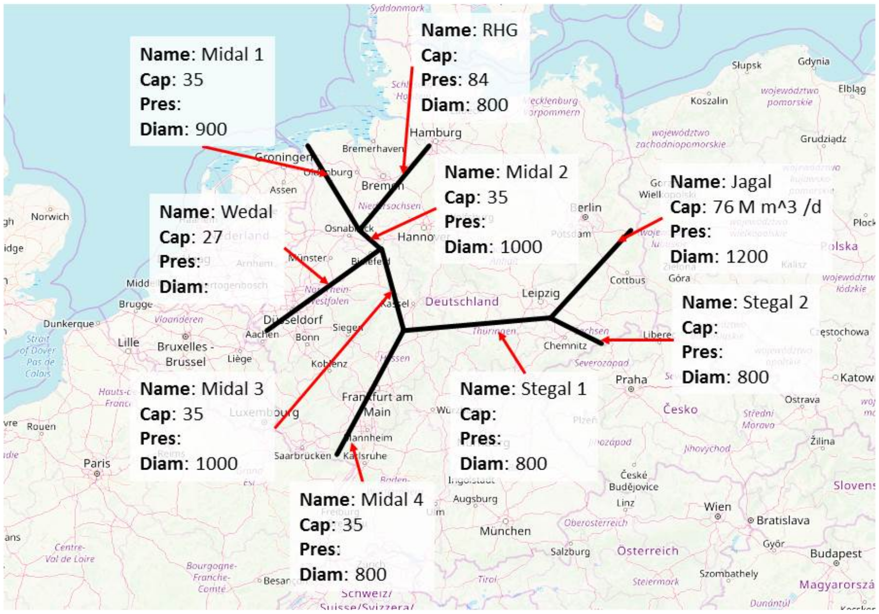

Figure 4.1: Map of some of the larger pipelines in Germany, with corresponding attributes *capacity* (cap), *pressure* (pres), and *diameter* (diam).

pressure and *diameter* are 67 %, 22 % and 89 % respectively. The overall goal will be to achieve a data density of 100 % for all attributes.

4.1.1 Fill value methods

As one can see, the *capacity* attribute value is given for six of the nine facilities. Several options exist in determining the missing values. A simple solution would be to use the average or median of the input values as a method of estimating any missing value. Here the **mean** and **median** value are $41\,\mathrm{Mm^3d^{-1}}$ and $35\,\mathrm{Mm^3d^{-1}}$, respectively. However, selecting the best approach can be difficult, and needs to be transparent. Hence, an "estimation uncertainty" term could be used as decision criteria to determine the best method.

Conventionally, in the worlds of data enginers and big data, one splits the data into a training data set and a test data set. Normally a 70/30 rule is applied, where 70 % of the data ends up in the training data set, and 30 % in the test data set. In the first step a method (e.g. **median**) is applied to the training data set. In the second step, the fitted method results are used to predict the values of the test data set. In the third step one calculates the absolute error between the method results and the test data set. The smaller the absolute error, the better the method. This error value could be used as the "estimation uncertainty" and could be used to choose the method, that would estimate any missing values with the smallest error values.

However, the SciGRID_gas project only contains relatively small data sets. Any splitting of the input data set into training and test data sets would create a data set too small for training and testing purposes. As an example, there are roughly 35 *LNGs* terminals in Europe, and splitting such data set would result in roughly 10 values for testing purposes only. Hence, throughout the SciGIRD_gas project the "Leave-one-out" method will be used (see Chapter 6.6), and the error is calculated using the "mean absolute error" (*MAE*), where the absolute error is the absolute difference between a single raw input data value and the model estimation of that value instance. This means, instead of having a 70/30 percent split, one uses all but one data value for training the model and then uses the trained model to estimate the one data value that was not part of the training process. This is being repeated for all data values.

The *MAE* for the **mean** and the **median** method is $25\,\mathrm{Mm^3d^{-1}}$ and $16\,\mathrm{Mm^3d^{-1}}$, respectively. Hence, based on the *MAE*, it would be best to use the **median** method approach, and one could fill all missing values with the value of $35\,\mathrm{Mm^3d^{-1}}$, with an *MAE* of $16\,\mathrm{Mm^3d^{-1}}$. The **median** method is normaly selected for data sets, which have outlyers or the data is not normal distributed. However, the sampel size is small, and one could argue, to select the method with the smalles *MAE*. However, overall, the *MAE* is very large in respect of the actual *capacity* value. Therefore other method approaches also need to be investigated.

An attribute could have a linear correlation with one or several other attributes. Here, one could use a linear regression. However, linear regressions tend to weight the independent feature data equally, if more than one is given. Other methods, such as the Lasso-linear regression, tend to weight the independent variables unequally and can even indicate that it would be better to remove some independent variables [Wik20d]. Therefore, if not stated otherwise, the Lasso-linear regression will be used here, instead of a simple linear regression.

Here, the Lasso-linear regression is applied to the *capacity* variable (also referred to as the "predictor" or "regression input"), and the variable *diameter* is the independent variable (also referred to as the "feature" variable). For the pipe RHG, where the *capacity* value is missing and a *diameter* value of 800 mm is given, the Lasso-linear regression estimated the following *capacity* value: $33.9\,\mathrm{Mm^3d^{-1}}$ with a *MAE* of $2.4\,\mathrm{Mm^3d^{-1}}$. As one can see, the *MAE* of the Lasso method is significantly smaller when compared with the *MAE* of the **mean** and the **median** methods. However, this example should not lead to the assumption, that a Lasso-linear regression is always better than a simple estimation using a **mean** or a **median** value. For example, an attribute data set could be unrelated to any other attribute; hence using a Lass method would be wrong. In addition, for some attribute values the methods of **mean** or **median** might have to be used, due to lack of feature data.

The processes described above have been applied to the example data presented in Table 4.1. Table 4.2 and Table 4.3 summarize the input values, estimation attribute values, estimation method, and the corresponding *MEA* based on the "Leave-one-out" approach, for the attribute *capacity* and *diameter* respectively. As the attribute *pressure* only contained two input values, no values could be estimated with the above process, as the system has been set up that it needs at least four values. For the other two attributes, the input and estimated values are being presented, and the difference between estimated and input value is close to the given uncertainty. As one can see, the estimated values agree better with the input data for the method of "Lasso", when compared with the method of "mean". However, not all values could be estimated using the Lasso method, due to missing values (e.g. *diameter* for pipeline Stegal_2).

Table 4.2: Input and estimated *capacity* data of the example, including the method of estimation and the corresponding estimated error. Values given in units of [M m ^{3} d $^{-1}$].

Pipeline name	Input *capacity*	Estimated *capacity*	Method	Uncertainty
Jagal	76	69.7	Lasso	2.4
RHG		33.9	Lasso	2.4
Midal 1	40	42	Lasso	2.4
Midal 2	50	51.8	Lasso	2.4
Midal 3	35	33.9	Lasso	2.4
Midal 4	34	33.9	Lasso	2.4
Stegal_1		33.9	Lasso	2.4
Stegal_2		42.8	Lasso	2.4
Wedal	27	43.7	Mean	12.9

Table 4.3: Input and estimated *diameter* data of the example, including the method of estimation and the corresponding estimated error. Values given in units of [mm].

Pipeline name	Input *diameter*	Estimated *Diameter*	Method	Uncertainty
Jagal	1200	1233	Lasso	23
RHG	800	900	Mean	100
Midal 1	900	875	Lasso	23
Midal 2	1000	975	Lasso	23
Midal 3	800	826	Lasso	23
Midal 4	800	816	Lasso	23
Stegal_1	800	900	Mean	100
Stegal_2	900	900	Mean	100
Wedal		746	Lasso	23

Hopefully the above example and description can be used as a blueprint of the problem that the SciGRID_gas project is facing, and how the missing value generation can be approached. The following section will describe the implemented method pathway within the SciGRID_gas project code.

4.1.2 Attribute value generation pathway

This section describes how the generation of the missing attribute values has been implemented. Overall, there are six steps that need to be carried out in order. They are described in more detail in the following sub-sections:

1) Loading network data
2) Configuration of the setup files
3) Generation of plots for data QA
4) Parameters generation for the heuristic methods
5) Selecting individual estimation methods for each attribute
6) Simulation of missing attribute values

1) Loading network data

As the first step, the data needs to be loaded into memory. Functions have been designed as part of the SciGRID_gas project, and will be introduced in an upcoming documentation.

2) Configuration of the setup files

In the next step the user needs to set up the two required setup files. In the first one ("StatsMethodsSettings.csv") meta information for each method (e.g. **mean**, **median**) is being supplied in addition to other settings. The second setup file ("StatsAttribSettings.csv") contains a list of attributes, including attribute specific metadata. Both setup files are described in more detail below.

StatsMethodsSettings.csv

In this file the user selects which methods (e.g. "Lasso") shall be used for testing the data and their relationships. A sample method setup file is given in Figure 4.2.

	A	B	C
1	MethodNam	Param	ToBeApplied
2	Lasso		1
3	LogisticReg	{'solver':'lbfgs'}	1
4	Mean		1
5	Median		1
6	Min		1
7	Max		1

Figure 4.2: Sample file of the file "StatsMethodsSettings.csv"

Column "A", which has the label "MethodName", contains the heuristic method names that have been implemented into the SciGRID_gas project code. Currently the following methods have been implemented:

- **Lasso**: A type of linear regression that uses shrinkages. It has been described as [sl19]: "The Lasso is a linear model that estimates sparse coefficients. It is useful in some contexts due to its tendency to prefer solutions with fewer non-zero coefficients, effectively reducing the number of features upon which the given solution is dependent. For this reason Lasso and its variants are fundamental to the field of compressed sensing. Under certain conditions, it can recover the exact set of non-zero coefficients."
- **LogisticReg**: Here the attributes to be predicted are of binary type or are multiple discrete values. The Logistic-regression is described by the scikit-learn.org web portal as [sl19]: "Logistic regression, despite its name, is a linear model for classification rather than regression. In this model, the probabilities describing the possible outcomes of a single trial are modelled using a logistic function.".
- **Mean**: Calculation of the **mean** attribute value of the predictor attribute values.
- **Median**: Calculation of the **median** value of the predictor attribute values.
- **Min**: Calculation of the **min** value of the predictor attribute values.
- **Max**: Calculation of the **max** value of the predictor attribute values.

The second column with the label "Param" contains possible parameters that are applied to the method within the SciGRID_gas Python code. Here for the LogisticReg a solver needed to be specified. As can be seen, the following entry was supplied:"{'solver':'lbfgs'}" (See [Wik20e] for an explanation of the 'lbfgs' solver). All other methods currently do not need additional parameter settings.

The column "ToBeApplied" describes if the method should be a part of the test suit (1) or not(0).

StatsAttribSettings.csv

In this CSV file (**StatsAttribSettings.csv**), additional information in respect of the attributes is being supplied. Here, the user selects the attributes (e.g. *max_cap_M_m3_per_d*), which shall be used during the heuristic testing suit. A sample of such file is presented in Figure 4.3.

The file consists of seven columns and they are described as follow:

- **CompName**: This column contains the component name. Options are all component names as introduced in Chapter 2.

	A	B	C	D	E	F
1	CompName	AttribName	Features	Predictors	Convert2Float	RegressionType
2	Compressors	max_cap_M_m3_per_d	1	1	0	lin
3	Compressors	is_H_gas	1	1	0	log
4	Compressors	max_power_MW	1	1	0	lin
5	Compressors	max_pressure_bar	1	1	0	lin
6	Compressors	num_turb	1	1	0	lin
7	Compressors	turbine_fuel_isGas_1	1	1	0	log
8	Compressors	turbine_fuel_isGas_2	1	1	0	log
9	Compressors	turbine_fuel_isGas_3	1	1	0	log

Figure 4.3: Sample file of the file "StatsAttribSettings.csv"

- **AttribName**: This column contains the attribute names of the component given under "CompName", that can be part of the heuristic testing process.
- **Features**: This indicates that the attribute values shall be a feature variable ("1") or not ("0").
- **Predictor**: This indicates if this variable shall be tested ("1") or not ("0"). Attributes with value settings of "1" will be loaded, independent of the settings under "Feature".
- **Convert2DiscreteValue**: This indicates, if the loaded data is to be converted from string variables to numbers. Below in Figure 4.4 an example is given of a column of "gender" entries and "age" values, where the attribute *gender* is being converted.

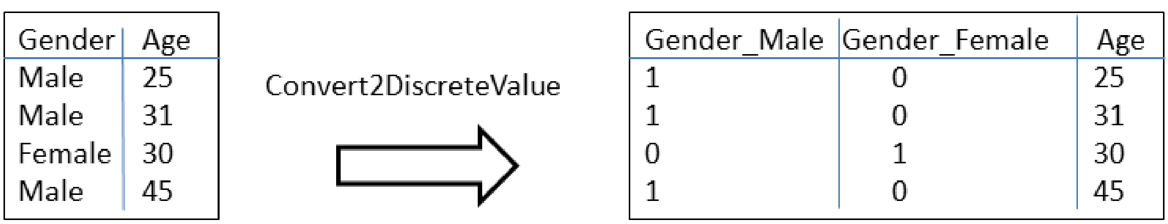

Figure 4.4: Example of converting strings attributes to number attributes.

RegressionType This is a string, indicating the regression method to be applied to the data. The following two options are currently implemented:

- "lin": This stands for "linear regression", and includes the Lasso linear regression, median, min and max sample values.
- "log": This stands for "logistic regression", and refers to a logistic regression.

Hence, with the above settings, the user can supply all information required for the testing phase, and the user has the option of modifying the testing runs by excluding certain variables. The following section will describe further required steps the user will need to undertake as part of the attribute value generation.

3) Generation of plots for data QA

Having a good understanding of the quality and quantity of the data is very important, before one can apply any heuristic methods for generating missing attribute values. Hence, **THE USER NEEDS TO (VISUELLY) INSPECT THE DATA THAT CAN BE USED AS INPUT FOR THE HEURISTIC PROCESSES!**

This is carried out with the function **M_Stats.gen_DataHists(** *Netz, CompNames, AttribNames, StatsInputDirName, DataStatsOutput* **)**. It requires the following inputs:

- *Netz*: A copy of the network.
- *CompNames*: A list of components to be visualized.
- *AttribNames*: A list of attribute names to be visualized.
- *StatsInputDirName*: A relative path to the above setup file "StatsAttribsSettings.csv".
- *DataStatsOutput*: A relative path to the main folder, where the plots will be stored to. After the plot generation, this folder will contain the following:
  - **HistPlots**: This is a subfolder containing further subfolders, one for each component, and each subfolder will contain the corresponding plots. Each plot consists of a scatter-plot of the data, and a histogram, see Figure 4.5 as an example. In addition, the title contains the name of the attribute next to information in respect of the attribute data density.
  - **Overview.png**: This is a file with the name "Overview.png" and contains pair-plots of attributes. (See Figure 4.6 as an example). Here each attribute of a component is plotted against all other attributes of the same component. This can be used to investigate if there are correlations between individual attributes already.

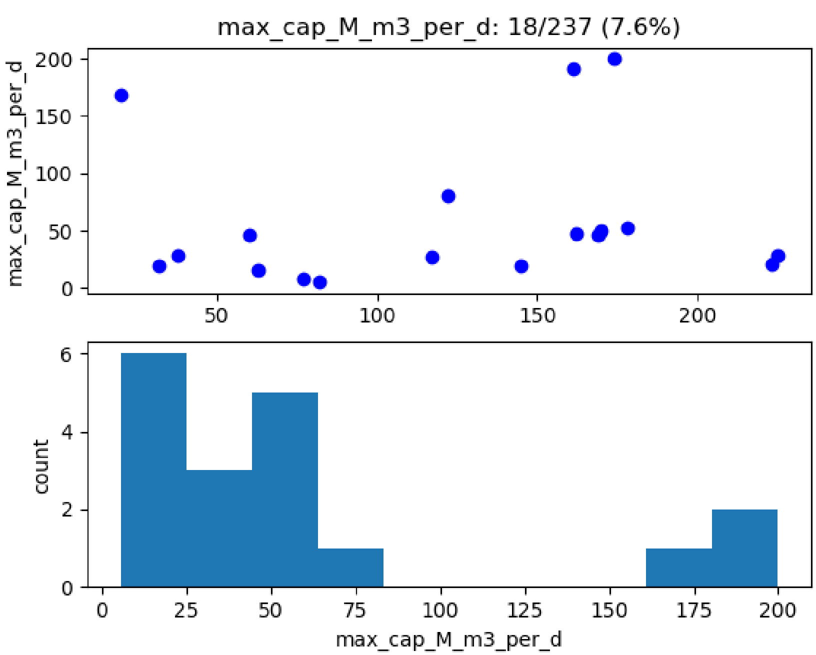

Figure 4.5: Example histogram plot of the *Compressors* attribute *max_cap_M_m3_per_d*.

With the help of the plots and data density values, the user can embark on the following steps:

- Correct any wrong data.

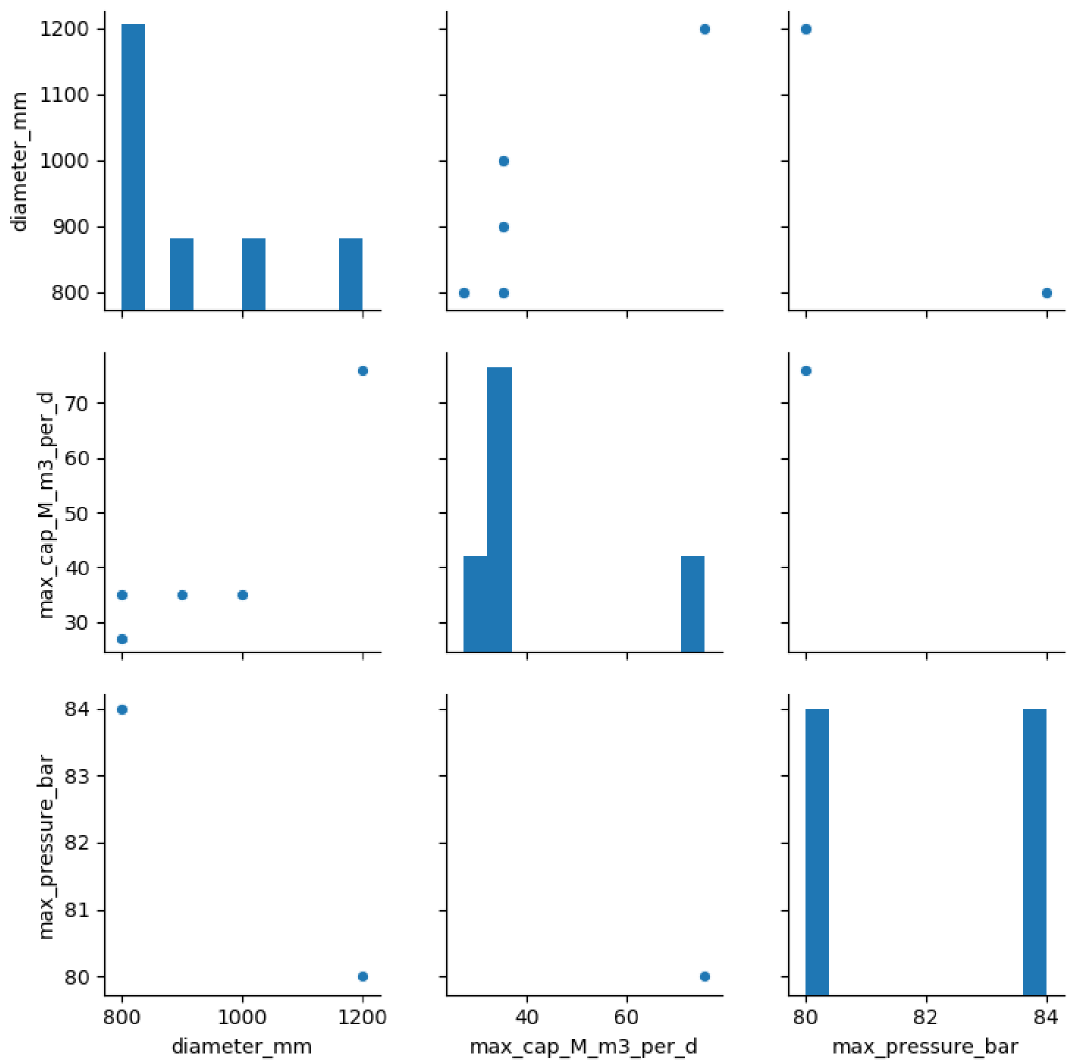

Figure 4.6: Overview of the mutual attribute relations for the component *Compressors*.

- Adding more data where data density is low, if possible.
- Select and unselect attributes from the subsequent steps due to data distribution and data density issues, resulting in changes in the setup files.

4) Parameters generation for the heuristic methods

After the data has been loaded into the Python memory and the setup files have been adjusted, the methods in conjunction with the feature attributes will be used to test to generate missing values (predictors). For this the function **M_Stats.gen_StatsParam(** *Netz, CompNames, StatsInputDirName, DataStatsOutput, MaxCombDepth* **)** has been generated. It needs the following inputs:

- *Netz*: A copy of the gas component data set.
- *CompNames*: A list of component names for which this process needs to be carried out.
- *AttribNames*: A list of attribute names for which this process needs to be carried out.
- *StatsInputDirName*: A relative path, where both input setup files can be found.
- *DataStatsOutput*: A relative path, where output information will be written to.
- *MaxCombDepth*: This gives the number of independent attributes, which can be used by each estimation method. The larger this value, the more combinations exists for a given list of independent attributes. However, more significant is that larger "MaxCombDepth" can lead to over-fitting. The number of resulting combinations of attributes can be estimated using $n!/(r!(n-r)!)$, where n the number of attribute variables to choose from, and r of them are chosen, where repetition is not allowed and order does not matter.

The output of the function **M_Stats.gen_StatsParam()** will be twofold: additional plots and measures of fit values.

A sample of such a plot is given in Figure 4.7. Each predictor (here *max_cap_M_m3_per_d*) is plotted against the selected features that were used to determine the predictor attribute (here *max_power_MW*). Here the predictor is plotted on the y-axis, whereas the feature is plotted on the x-axis. The solid line is the estimation of the method used. The title contains information on the method selected, and an R-square value of the fit that was determined using the method.

In addition to the graphical output, additional simulation information is stored in individual CSV files for each component. These files are described in more detail below.

An example output file is given in Figure 4.8 and Figure 4.9 for the component *LNGs*.

These files are written to the folder "/Ausgabe/Sample/StatsData/". All generated files start with the name "RetSummary" and are followed by the name of the component, separated by an underscore. Therefore, the CSV file name is "RetSummary_LNGs.csv" for the component *LNGs*.

The files contain information on attributes, methods, errors and parameter settings, where each line is a single run/test result. The columns are as follow:

- *CompName*: Name of the component.
- *AttribName*: Name of the predictor attribute.
- *NumElements*: Number of elements of this component.
- *MethodName*: Name of the method used, that was selected through the setup file "StatsMethodsSettings.csv".
- *NumFeatures*: Number of features used to estimate the predictor.
- *FeatureNames*: List of feature attribute names, that were used to estimate the predictor.
- *Plots*: A link to the individual plots of the features and attribute relationship, where the hyperlink currently works under Excel on Windows only.

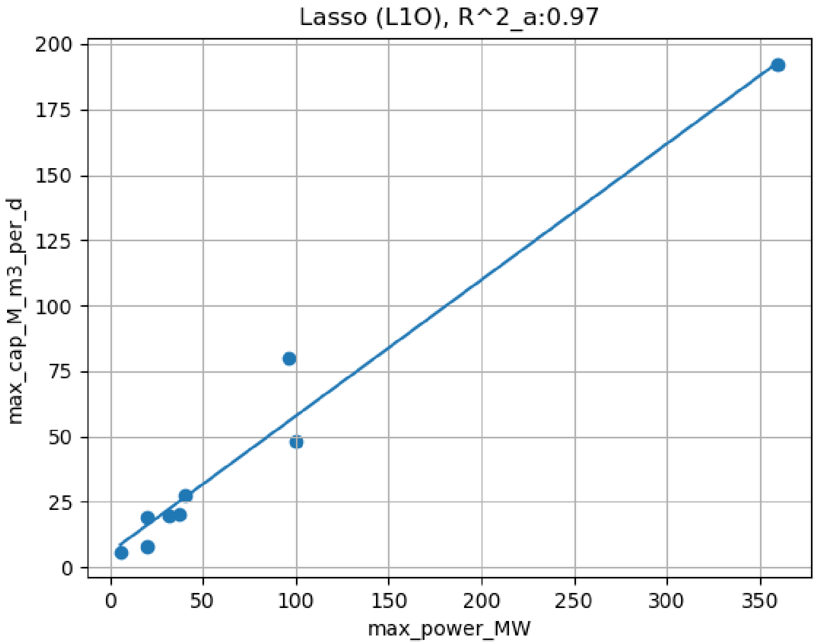

Figure 4.7: Example of attribute *max_power_MW* versus *max_cap_M_m3_per_d* from the component *Compressors*. The solid line represents the fit of the Lasso method to the data.

	A	B	C	D	E	F
1	CompName	AttribName	NumElements	ModelName	NumFeatures	FeatureNames
2	LNGs	max_cap_store2pipe_M_m3_per_d	32	Lasso	1	["Pipe_max_cap_M_m3_per_d"]
3	LNGs	max_cap_store2pipe_M_m3_per_d	32	Lasso	1	["median_cap_store2pipe_M_m3_per_d"]
4	LNGs	max_cap_store2pipe_M_m3_per_d	32	Lasso	1	["max_workingGas_M_m3"]
5	LNGs	max_cap_store2pipe_M_m3_per_d	32	Mean	1	["max_cap_store2pipe_M_m3_per_d"]
6	LNGs	max_cap_store2pipe_M_m3_per_d	32	Median	1	["max_cap_store2pipe_M_m3_per_d"]
7	LNGs	max_workingGas_M_m3	32	Lasso	1	["median_cap_store2pipe_M_m3_per_d"]
8	LNGs	max_workingGas_M_m3	32	Lasso	1	["max_cap_store2pipe_M_m3_per_d"]
9	LNGs	max_workingGas_M_m3	32	Mean	1	["max_workingGas_M_m3"]
10	LNGs	max_workingGas_M_m3	32	Median	1	["max_workingGas_M_m3"]

Figure 4.8: Example CSV output of heuristic model results for the component *LNGs*, depicting columns A - F.

G	H	I	J	K	L	M	N	O
Plots	NumSamples	NumFill	BIC	MeanAbsError	R_2	R_2_adj	ReplaceType	ModelParam
../StatsDa	4	0	14.3057177	4.845552454	0.72296201	0.584443015		{"SC_Mean": [34.5343801375], "SC_Scale": [28
../StatsDa	21	0	85.2509268	4.756034276	0.822467826	0.813124027		{"SC_Mean": [29.326994301428574], "SC_Scale
../StatsDa	28	4	109.381584	5.140814158	0.846444198	0.840538205		{"SC_Mean": [256364758.0], "SC_Scale": [1687
../StatsDa	29	3	166.939796	12.85032496	0	-0.037037037		{"SC_Mean": [0], "SC_Scale": [0], "Intercept":
../StatsDa	29	3	168.158632	12.42324043	-0.04292456	-0.081551394		{"SC_Mean": [0], "SC_Scale": [0], "Intercept":
../StatsDa	21	0	764.014679	69016010.55	0.786511086	0.775274828		{"SC_Mean": [29.326994301428574], "SC_Scale
../StatsDa	28	2	1011.07223	56118773.21	0.849922646	0.84415044		{"SC_Mean": [24.677103718199607], "SC_Scale
../StatsDa	31	1	1180.27533	149945270.6	0	-0.034482759		{"SC_Mean": [0], "SC_Scale": [0], "Intercept":
../StatsDa	31	1	1181.666	147160879	-0.04588166	-0.081946546		{"SC_Mean": [0], "SC_Scale": [0], "Intercept":

Figure 4.9: Example CSV output of heuristic model results for the component *LNGs*, depicting columns G - O.

- *NumSamples*: Number of samples of the feature data, which were used as part of this method evaluation (this number can never be larger than the value in column *NumElements*).
- *NumFill*: Number of elements for which the predictor attribute can be simulated with the method, where the attribute had missing values. (This value also includes the value given under *NumSamples*.)
- *BIC*: Indicator for the goodness of fit of the model using the BIC (Bayesian information criterion) value. The lower the BIC value the better the method fit.
- *MeanAbsError*: Measure of goodness of fit of the model using the mean absolute error (MAE).
- *R_2*: R-square model value.
- *R_2_adj*: Adjusted R-square value.
- *ReplaceType*: An empty column that will be used at a later stage.
- *ModelParam**: An entry containing all the fitting parameters used by the methods and attributes, the scaling values of the features attributes ("SC_Mean", and "SC_Scale"), and the method parameters ("Intercept", "Coef").

With the above information, visual and values, the user can make an informed decision in respect of which method could be used with which attribute to fill missing values.

5) Selecting individual estimation methods for each attribute

As described in the above processes, the function **M_Stats.gen_StatsParam()** generates plots and CSV files containing information on the goodness of the methods for each attribute. With this information the user can decide which method in combination with feature attribute values can be used to generate the missing attribute values.

Hence, in the next step the user needs to create a setup file, which contains the settings for methods and attributes that will be used for the generation of those missing attribute values. For this the user can use the output files from the previous step, by carrying out the following actions:

- Copy the above output CSV files to a new location.
- Open one of those files after the other, and carry out the next steps for each file:
  - With the information of the graphs and the indicator of goodness of fit values (e.g. MAE), remove all those method attribute combinations that shall not be used for any heuristic processes.
  - Place one of the following keywords into the column *ReplaceType*:
    * "replace": All values will be replace with the simulated value. Even the original input data will be replaced by the newly simulated values.
    * "fill": Here only the missing attribute values will be determined, therefore, the original input data will not be overwritten, in contrast to option "replace".
    * "fill_ARR": Here missing attribute values are being filled, and values that stem from copyright protected sources are also being overwritten.

However, independent on the replace type setting, some attribute values might not be estimated with a single method and single feature attribute set. This can be due on missing feature attribute values required during the estimation process. Hence, the user will need to select several different methods for the generation of all missing attribute values, by retaining several different method lines for a single attribute in the CSV file. This can be seen in Figure 4.8 in the depicted rows two to four. The attribute to be estimated and the model method are the same. However, the feature input variables differ for each line. With this approach one should be able to estimate all missing values. To retain the highest confidence in the estimated values, the user will need to select only those methods and feature attribute value combinations, that results in smallest errors.

Here it is important to notice, that the attribute values are generated in the order as they appear in the CSV input file. Hence, the user should order the methods in such a way, that the method with the "best" predictions are being carried out first. This could be followed by methods that generate attribute values with larger errors. To assure that there are no further missing values one could retain the **mean** or **median** method as the last method, filling any values that were left unfilled by any previous estimation.

6) Simulation of missing attribute values

The actual simulation and filling of the attributes is carried out with the function **M_Stats.pop_Attribs(** *Netz, CompNames, StatsInputDirName* **)** and is the last step in getting attribute values filled in a gas component data set. This function requires the following input:

- *Netz*: A copy of the component data set.
- *CompNames*: A list of components for which the element's attributes shall be generated and filled.
- *StatsInputDirName*: A relative path name of the location of the above modified setup files.

The return of this function is a component data set, where all missing attribute values have been generated.

4.2 Example value estimation

As an example, a result for the attributes *max_cap_pipe2store_M_m3_per_d* and *max_cap_store2pipe_M_m3_per_d* of the component *Storages* will be presented for the combined IGG data set. The table contains three columns with numbers, with the following definition:

- "N": This is the number of raw input values; hence this number is equal or smaller than the number of all facilities of this component.
- "A": This is the overall average value after all missing values have been estimated, using input and estimated values.
- "M": This is the mean absolute error, of those elements, of which the attribute value had to be determined.

Overall, there are 216 *Storages* elements in the data set. 48 values were missing for both attributes. After the attribute value estimation, the overall mean ()A of the attributes *max_cap_pipe2store_M_m3_per_d* and *max_cap_store2pipe_M_m3_per_d* was 13.9 and 14.6 respectively. The mean absolute error (M) calculated to be 10.9 and 11.2 for the attributes *max_cap_pipe2store_M_m3_per_d* and *max_cap_store2pipe_M_m3_per_d*. This seems large in comparison with the overall average value A. However, the individual values range from 0 to more than 100 for both attributes.

Table 4.4: List of attributes of the *Storages* component for the IGG data sets, with some statistical properties.

Attribute name	N	A	M
max_cap_pipe2store_M_m3_per_d	168	13.9	10.9
max_cap_store2pipe_M_m3_per_d	168	14.6	11.2

A histograms of the raw and the estimated values, depicted in Figure 4.10, gives the distribution for both attributes.

The estimated values are roughly distributed the same way that the raw values are distributed. The exceptions are the larger counts of the bin containing the median value of the raw data set. Here the median value is the values used, if there was no other means of determining a missing value. As can be seen, the median value was used substantially for several elements.

Here a quick statistical Z-score was carried out [UoO14], giving Z-score values of around -1.45 for both attributes. This indicates that the raw and the estimated distributions are the same, as their absolute values are samller than two.

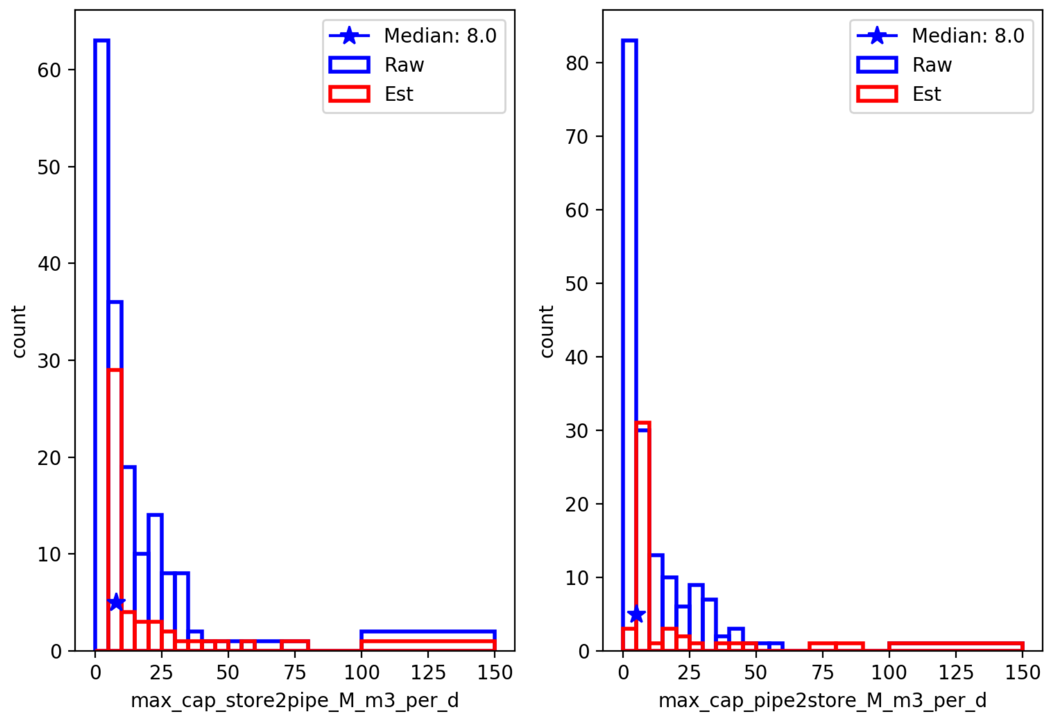

Figure 4.10: Histogram of raw (blue) and estimated (red) values for *max_cap_store2pipe_M_m3_per_d* (left) and *max_cap_pipe2store_M_m3_per_d* (right) of the *Storages* component. Both subplots also indicate the location of the median value for the raw data (star).

4.3 Automated attribute value generation

In addition to the manual process described in Chapter 4.1, an automated process has also been developed and will be briefly explained below. The automated process has been implemented for users with little statistical background, and to get first results fast.

However, the user should be aware, that by applying the automated process, some methods might be selected, that lead to incorrect results (e.g. negative *Storages* capacity for increasing max pipeline pressure of connected pipes). Hence, the user should carry out the manual process, instead of relying on the automated process results, where those bad relations could be eliminated.

The difference to the manual process described in Chapter 4.1 is that the selection of the attribute generation methods, as described in Chapter 4.1.2, has been automated. As was described in Chapter 4.1.2, one is supposed to select those methods with the best goodness of fit, e.g. BIC or MAE. Here for the automated process, the MAE value is the value that has been selected to determine which attribute generation method is to be used. The automated process selects the methods with increasing MAE value. Up to four different methods are automatically selected. In addition, if the median method is not part of the selected methods, then this method will be added to the list of methods to be executed, and will be executed last.

All other processes are as described in Chapter 4.1.

Values supplied with this data set here have been generated using the automated attribute value generation process.

4.4 Single network generation

So far raw data has been loaded, and converted into a SciGRID_gas data set. Any missing attribute values have been generated with the help of implemented regression methods. The last step is to assure, that all elements of all components are connected with each other into one large network. Here a method has been created that looks for facilities, such as *Storages* elements, that are not connected to a pipeline. Then the closest pipeline is determined, and checked, if the pipeline is closer than a user specified distance. In case that the facility is closer than this distance, the facility is moved to the pipeline. This is carried out with the function **M_Shape.moveComp2Pipe(** *Netz, CompName, PipeName, maxDistance_km* **)**. The inputs are as follow:

- *Netz*: A copy of the gas component data set.
- *CompName*: A name of the component for which this process needs to be carried out.
- *PipeName*: A name of the type of pipeline that are used in the network. Options are “PipeSegments” and “PipeLines”.
- *maxDistance_km*: The maximum distance by which the facility will be moved in units of [km].

The table below (Table 4.5) shows the number of elements prior and after this process, whereas the number of segments and length of overall network did not change in this process. Here the value for *maxDistance_km* was set to 100 km.

Table 4.5: INET number of elements prior and post connection with pipelines.

Component name	# of elements prior to process	# of elements post process
Compressors	249	230
Storages	199	187
InterConnectionPoints	118	117
EntryPoints	37	37
BorderPoints	119	118
LNGs	32	29

As can be seen, this resulted in discarding several elements, as their distance to the nearest pipeline was larger than the set 100 km. However, this assures now, that the entire data set is a gas network data set, where all elements are connected with each other and all attribute values have been estimated.

The final gas network data set is also been presented in Figure 4.11.

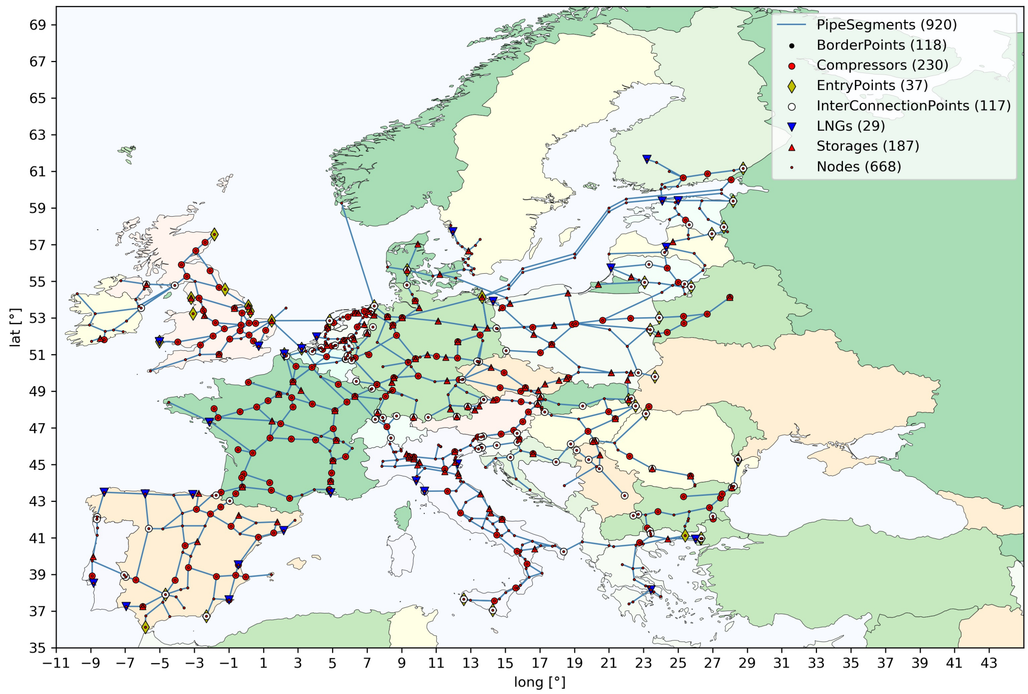

Figure 4.11: Final INET gas transmission network data set.

As can be seen when compared with Figure 3.1, all those facilities, that are more than 100 km away from the next pipeline, e.g. in the Ukraine, have been removed.

4.5 Summary

Here a method pathway has been described to fill missing attribute values. This is a complex process, where the Python code generates plots and model output values, which need to be considered by the user. With the information on hand, the user can decide if certain missing attribute values should be estimated using implemented regression methods. In addition, an automated process has been described, that can be used to generate all missing values without any additional user tasks, however the user has been made aware, that this can lead selecting incorrect attribute relationships, leading to incorrect values, as expert input is missing.

In addition, a process to generate a single network was briefly introduced, so that all elements, such as LNG terminals, are connected with pipelines with each other. Such generated data set is ready to be used by modellers.

CHAPTER

FIVE

CONCLUSION

This document here is the documentation of one of the data sets that is part of the SciGRID_gas project. This document started off with the introduction of the SciGRID_gas project, such as funding, duration and goals. In a subsequent chapter the data structure within the SciGRID_gas project was described, such as components, elements, attributes and attribute values, so that the transmission data set could be an input to some gas flow models. The third chapter introduced the INET data set, which is a data set that was generated as part of the SciGRID_gas project by collating all possible attributes and attribute values from the internet. As this data set contained missing attribute values, the fourth chapter introduced heuristic methods, that can be used to generate missing values. This resulted in the filled-INET transmission gas network data set, containing: 29 LNG terminals, 187 storage locations, 229 compressors, 118 interconnection points, 118 border point, 37 entry points and 918 pipe segments (60,000 km of gas pipelines through Europe).

CHAPTER

SIX

APPENDIX

6.1 Glossary

Dataset abreviations can be found in Table 6.1.

Table 6.1: Dataset abreviations

Name	Abbreviation	Description
Raw InternetDaten data set	INET	This is the label/name for the raw InternetDaten data set
Raw Gas Infrastructure Europe data set	GIE	This is the label/name for the raw Gas Infrastructure Europe data set
Raw Gas Storage Europe data set	GSE	This is the label/name of the raw Gas Storage Europe data set
Raw Norwegian data set	NO	This is the label/name for the raw Norwegian data set
Raw Long-term planning and short-term optimization data set	LKD	This is the label/name for the raw Long-term planning and short-term optimization data set
Raw International Gas Union data set	IGU	This is the label/name for the raw International Gas Union data set
Raw EntsoG-Map data set	EMAP	This is the label/name for the raw EntsoG-Map data set
Merged and filled IGG data set	IGG	This is the filled data sets, for which the **I**NET, **G**IE and **G**SE data sets were merged
Merged and filled IGGI data set	IGGI	This is the filled data sets, for which the **I**NET, **G**IE, **G**SE and **I**GU data sets were merged
Merged and filled IGGIG data set	IGGIG	This is the filled data sets, for which the **I**NET, **G**IE, **G**SE, **I**GU and the **GB** data sets were merged

The glossary terms can be found in Table 6.2.

Table 6.2: Glossary (A)

Name	Abbreviation	Description
component		A gas network consists of different components, such as pipelines, compressors LNG terminals and more. However, for a gas transmission network, there is a handful of components only: pipeline, compressor, LNG terminal, storage, entry point, border point, connection point, consumer, node, and production
element		Elements are instances of component. Hence, we speak of 10 compressor elements, if we have a data set that has 10 compressors. Here then we can refer to the first or the last or any element of such component
attribute		Gas facilities, such as pipelines or compressors, can be described with a large number of parameters, such as pipeline diameter, or compressor capacity. Those parameters are referred to as attributes. Hence, each component has a list of properties, which are different from one component to another component
facility		General term used for a gas appliance, such as compressor element, or LNG terminal
PipeLine		This is a gas pipeline entity, which has one start and one end point, however can run via many nodes, compressors and other gas network elements
PipeSegment		This is a gas pipeline, that has only one start and one end point, but no nodes in-between, Hence, only goes from one node to another node
LNG	LNG	Liquefied natural gas
CNG	CNG	Compressed natural gas
flow duration curve	FDC	It is the cumulative frequency curve that shows the percent of time specified flow were equal or exceeded during a given period. The information on occurrence of events is lost
Energiewende		German term for the change in using primary energies, the move away from coal to renewable energies, such as wind or solar
gas component data set		Raw input data, associated with components of the gas transmission grid
gas network data set		Output data, a coherent network of gas transmission components
OSM	OSM	Data that is available from the openstreetmap.org
non-OSM	Non-OSM	Data that is not part of the OSM data set
gas type		There are two types of gas High (H) and Low (L) calorific gas
mean absolute error	MAE	mean difference between input values and estimated values
data density		This is the ratio of the number of usable (not missing) attribute values over number elements of the component, in units of [%]
Transmission System Operators	TSO	This is an entity entrusted with the transportation of natural gas/electricity, as defined by the European Union
gas transmission network		This describes the physical gas transmission grid, however excludes any facilities/components that would be part of a distribution network and their facilities. This projects goal is to create an open source gas network data set that can be used to describe the European gas transmission network

Table 6.3: Glossary (B)

Name	Abbreviation	Description
gas component data set		The term "gas component data set" is used for all raw data of objects/facilities that have been loaded using SciGRID_gas tools into a Python environment. However, not all elements (e.g. compressors) must be connected to pipelines, Hence, such a data set is referred to as a "gas component data set", and the emphasis is on the term **component**
gas network data set		A "gas component data set" can be converted into a "gas network data set", by connecting all non-pipeline elements to nodes and all nodes are connected to pipelines, and as part of the process all network islands have been removed, resulting in a single network. Therefore the network contains nodes and edges which are connected, and all objects with the exception of pipelines are associated with nodes in this network, whereas pipelines are associated with edges. Hence, the emphasis here is on the term **network**

6.2 Unit conversions

Table 6.4: Unit conversions

From Unit	To Unit	MultiVal
LNG Mt	LNG Mm^3	2.47
gas tm^3/h	gas Mm^3/d	24/1000
LNG Mm^3	gas Mm^3	584
LNG t	gas Mm^3	1442.48

6.3 References for INET data set

Below a list of those sources used to generate the INET data set.

- Google (n.d.). [Google Maps of Europe]. Retrieved between 15.July.2018 and 31.June.2021
- http://belarus-tr.gazprom.ru/
- http://corporate.vattenfall.com/about-vattenfall/operations/market-transparency/gas-storage/
- http://en.gaz-system.pl//
- http://gaslager.energinet.dk/EN/Pages/default.aspx
- http://interfaxenergy.com/article/19138/lng-better-than-norway-pipeline-ex-polish-pm
- http://ir.gasplus.it/home/show.php?menu=00002
- http://italgasstorage.it/eng/progetto.html
- http://media.edfenergy.com/Misc/AboutUs.aspx
- http://mmbf.hu/en/company/gas_storage
- http://mndgsgermany.com/
- http://mysolar.cat.com/cda/files/870985/7/Solar_Turbines_5000_Gas_Compressor_News_Rel_Sent.pdf

- http://sse.com/whatwedo/ourprojectsandassets/thermal/Aldbrough/
- http://sse.com/whatwedo/wholesale/gasstorage/
- http://www.bayernugs.de/4-1-Home.html
- http://www.bendisenergy.com.tr/tr/Projelerimiz/24-toren-dogalgaz-depolama-ve-madencilik-as
- http://www.berliner-erdgasspeicher.de/en/Pages/default.aspx
- http://www.botas.gov.tr/
- http://www.bulgartransgaz.bg/en/pages/transstorge-110.html
- http://www.caythorpegasstorage.com/caythorpe/
- http://www.centrica-sl.co.uk/
- http://www.ceskaplynarenska.cz/en/ultimate-speed-underground-gas-storage
- http://www.dea-speicher.de/en
- http://www.depomures.ro/
- http://www.edisonstoccaggio.it/en
- http://www.ekb-storage.de/de/home/
- http://www.emplpipeline.com/en/the-gas-pipeline/
- http://www.emplpipeline.com/en/the-gas-pipeline/,Extra
- http://www.enagas.es/enagas/en/Transporte_de_gas/Almacenamientos_Subterraneos
- http://www.energystock.com/
- http://www.ewe-gasspeicher.de/english/index.php
- http://www.fluxys.com/belgium/en/Services/Storage/Storage
- http://www.gasnaturalfenosa.com/en/activities/lines+of+business/1285338591925/supply+and+transportation+of+gas.html
- http://www.gasspeicher-hannover.de/startseite.html
- http://www.gasstorage.cz/en/operation-information/available-firm-capacity/
- http://www.gasstoragebergermeer.com/gas-storage-bergermeer-2/
- http://www.gastrade.gr/en/the-company/the-project.aspx
- http://www.gas-union-storage.de/
- http://www.gatewaystorage.co.uk/
- http://www.gazprom.com/about/production/underground-storage/
- http://www.geogastock.it/ITA/Home.asp
- http://www.grtgaz.com/en/major-projects/beynes-compressor-station/presentation/news/compressor-station-at-the-beynes-site.html
- http://www.grtgaz.com/fileadmin/plaquettes/en/2017/Essentiel-plaquette-institutionnelle-EN-2017.pdf
- http://www.gsa-services.ru/
- http://www.halite-energy.co.uk/our-project/project-overview/
- http://www.hradf.com/en/portfolio/south-kavala-natural-gas-storage
- http://www.humblyenergy.co.uk/about-us#useful-information

- http://www.islandmageestorage.com/
- http://www.kge-gasspeichergesellschaft.de/
- http://www.kgsp.co.uk/
- http://www.kingstreetenergy.com/
- http://www.kinsaleenergy.ie/gas-storage.html
- http://www.le.lt/index.php/projects-in-progress/syderiai-underground-gas-storage/535
- http://www.lg.lv/index.php?id=3376&lang=eng
- http://www.magyarfoldgaztarolo.hu/en/Lapok/default.aspx
- http://www.mnd.eu/en/2014-11-26-12-13-02/mnd-group-companies
- http://www.nafta.sk/en/about-gas-storage
- http://www.nam.nl/en/about-nam/facts-and-figures.html
- http://www.omv.com/portal/01/com/gas/storage
- http://www.petroceltic.com/operations/bulgaria.aspx
- http://www.pgnig.pl/reports/annualreport2012/en/ar-obrot-magazynowanie-2.html
- http://www.pozagas.sk/en/?PHPSESSID=8d83cc7890abb7980f6793cf56633a72
- http://www.psp.hr/home
- http://www.rag-energy-storage.at/en.html
- http://www.romgaz.ro/en/content/ugs-n-366-new-gas-storage-facility-romania
- http://www.romgaz.ro/en/content/ugs-n-371-sarmasel-storage-facility-upgrading
- http://www.romgaz.ro/en/inmagazinare
- http://www.rwe.com/web/cms/de/37110/rwe/presse-news/pressemitteilungen/?pmid=4005467,http://digitalnewsservice.net/clients/net4gas-nimmt-neue-hochdruck-gas-pipeline-gazelle-in-betrieb/
- http://www.rwe.com/web/cms/en/531750/rwe-gasspeicher/
- http://www.scottishpower.com/pages/hatfield_moor_gas_storage_facility.asp
- http://www.snam.it/en/transportation/Thermal_Year_Archive/Thermal_Year_2013_2014/Info-to-users/index.html
- http://www.sppstorage.cz/#
- http://www.srbijagas.com/naslovna.1.html
- http://www.stogit.it/en/index.html
- http://www.storengy.com/countries/deutschland/en/products-services.html
- http://www.storengy.com/countries/france/en/
- http://www.storengy.com/countries/unitedkingdom/en/oursites.html
- http://www.taqaglobal.com/our-regions/netherlands/gas-storage/peak-gas-installation/overview?sc_lang=en
- http://www.terranets-bw.de/en/gas-transmission/we-transport-your-natural-gas/
- http://www.thueringerenergie.de/Unternehmen/Ueber_uns/Geschaeftsfelder/Speicher.aspx
- http://www.tigf.fr/en/what-we-can-offer/storage.html
- http://www.tpao.gov.tr/eng/?tp=m&id=31

- http://www.tpao.gov.tr/eng/?tp=m&id=84
- http://www.trianel-gasspeicher.com/
- http://www.ugs-katharina.de/en/unternehmen.html
- http://www.vng-gasspeicher.de/content/en/Speicher/index.html
- https://de.wikipedia.org/wiki/Baumgarten_an_der_March
- https://de.wikipedia.org/wiki/Erdgasleitung_Jamal%E2%80%93Europa,https://en.wikipedia.org/wiki/Yamal%E2%80%93Europe_pipeline
- https://de.wikipedia.org/wiki/Hungaria-Austria-Gasleitung
- https://de.wikipedia.org/wiki/MIDAL,https://www.gascade.de/netzinformationen/unser-leitungsnetz/midal/
- https://de.wikipedia.org/wiki/Mittel-Europ%C3%A4ische_Gasleitung,https://en.wikipedia.org/wiki/MEGAL_pipeline
- https://de.wikipedia.org/wiki/NEL_(Pipeline),https://en.wikipedia.org/wiki/NEL_pipeline,https://www.fluxys.com/nel/en/NELSystemInfo/AboutNEL
- https://de.wikipedia.org/wiki/Norddeutsche_Erdgas-Transversale,https://en.wikipedia.org/wiki/Netra
- https://de.wikipedia.org/wiki/OPAL_(Pipeline),https://en.wikipedia.org/wiki/OPAL_pipeline,https://www.opal-gastransport.de/netzinformationen/ostsee-pipeline-anbindungsleitung/
- https://de.wikipedia.org/wiki/Penta-West
- https://de.wikipedia.org/wiki/Rehden-Hamburg-Gasleitung,https://en.wikipedia.org/wiki/Rehden%E2%80%93Hamburg_gas_pipeline
- https://de.wikipedia.org/wiki/Trans_Austria_Gasleitung,https://en.wikipedia.org/wiki/Trans_Austria_Gas_Pipeline
- https://de.wikipedia.org/wiki/Trans-Adria-Pipeline,https://en.wikipedia.org/wiki/Trans_Adriatic_Pipeline
- https://de.wikipedia.org/wiki/Trans-Europa-Naturgas-Pipeline
- https://de.wikipedia.org/wiki/Transgas-Pipeline,Extra
- https://de.wikipedia.org/wiki/WEDAL,https://www.gascade.de/netzinformationen/unser-leitungsnetz/wedal/
- https://de.wikipedia.org/wiki/West-Austria-Gasleitung
- https://en.wikipedia.org/wiki/Arad%E2%80%93Szeged_pipeline
- https://en.wikipedia.org/wiki/BBL_Pipeline
- https://en.wikipedia.org/wiki/BRUA_Pipeline
- https://en.wikipedia.org/wiki/Europipe_II
- https://en.wikipedia.org/wiki/Giurgiu%E2%80%93Ruse_pipeline
- https://en.wikipedia.org/wiki/MEGAL_pipeline,https://de.wikipedia.org/wiki/Mittel-Europ%C3%A4ische_Gasleitung
- https://en.wikipedia.org/wiki/National_Transmission_System
- https://en.wikipedia.org/wiki/Scotland-Northern_Ireland_pipeline,E127
- https://en.wikipedia.org/wiki/South_Wales_Gas_Pipeline
- https://en.wikipedia.org/wiki/Transitgas_Pipeline,https://web.archive.org/web/20070808033932/http://www.swissgas.ch/en/3_2.php

- https://en.wikipedia.org/wiki/Transitgas_Pipeline,transitgassystem_en.gif,https://web.archive.org/web/20070808033932/http://www.swissgas.ch/en/3_2.php
- https://en.wikipedia.org/wiki/V%C3%A1rosf%C3%B6ld%E2%80%93Slobodnica_pipeline
- https://www.fnb-gas.de/media/FNB_GAS_Projekte_2014_02_17_anlage_6_nep-gas-2014_projekt-steckbriefe.pdf
- https://globallnghub.com/wp-content/uploads/2018/09/King.pdf
- https://globallnghub.com/wp-content/uploads/2018/09/King.pdf
- https://news.err.ee/610905/vopak-to-build-initially-4-000-cubic-meter-lng-terminal-at-muuga
- https://globallnghub.com/wp-content/uploads/2018/09/King.pdf
- https://www.lngworldnews.com/tallinna-sadam-alexela-to-work-on-paldiski-lng-terminal/
- https://globallnghub.com/wp-content/uploads/2018/09/King.pdf
- https://www.sourcewatch.org/index.php/Delimara_Malta_LNG_Terminal
- https://theodora.com/pipelines/france_and_belgium_pipelines.html
- https://theodora.com/pipelines/france_and_belgium_pipelines.html, E21
- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E10
- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E106
- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E11
- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E112
- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E113
- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E13
- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E14
- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E16_1
- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E16_2
- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E17_Skikda
- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E22_Arzew_1
- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E22_Arzew_2
- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E48
- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E64
- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E65
- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E78
- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E80_1
- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E80_2
- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E81
- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E85
- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E86
- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E87
- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E88

- https://theodora.com/pipelines/france_and_belgium_pipelines.html,E91
- https://transparency.entsog.eu/
- https://www.astora.de/storage-locations/haidach-storage-facility.html?L=1
- https://www.astora.de/storage-locations/jemgum-storage-facility.html?L=1
- https://www.astora.de/storage-locations/jemgum-storage-facility.html?L=2
- https://www.astora.de/storage-locations/rehden-storage-facility.html?L=1
- https://www.baltic-pipe.eu/de/das-projekt/
- https://www.enagas.es/stfls/ENAGAS/Transporte%20de%20Gas/Documentos/CAT_English.pdf
- https://www.enbw.com/unternehmen/konzern/geschaeftsfelder/speicher/gasspeicher/zahlen-daten-fakten.html
- https://www.eneco.nl/over-ons/projecten/gasspeicher/voorraadniveau
- https://www.energiefachmagazin.de/Branchen-News/VNG-Gasspeicher-GmbH-legt-Erdgasspeicher-Buchholz-still
- https://www.enovos.de/industrie/ueber-uns/erdgasspeicher
- https://www.eugal.de/eugal-pipeline/,https://www.mdr.de/nachrichten/politik/regional/baubeginn-erdgastrasse-eugal-umstritten-100.html
- https://www.europipe.com/de/referenzen/referenzprojekte/
- https://www.eustream.sk/en_transmission-system/en_transmission-system
- https://www.fluxys.com/belgium/en/about%20fluxys/infrastructure/network/network
- https://www.gascade.de/en/our-network/compressor-stations/radeland/
- https://www.gascade.de/en/our-network/our-pipelines/jagal/
- https://www.gascade.de/netzinformationen/unser-leitungsnetz/stegal/
- https://www.gascade.de/netzinformationen/unser-leitungsnetz/stegal/,https://en.wikipedia.org/wiki/STEGAL, https://de.wikipedia.org/wiki/STEGAL
- https://www.gascade.de/presse/presseinformationen/pressemitt,http://www.friedrich-vorwerk.de/de/aktuell/projekte/neubau-nowal-dn-1000.htmleilung/news/bau-der-nord-west-anbindungsleitung-nowal-startet-anfang-maerz/
- https://www.gasinfocus.com/en/indicator/existing-and-planned-lng-terminals/
- https://nauticor.de/lng-terminal-nynaeshamn
- https://www.gazprom.com/f/posts/86/569604/portovaya_eng.pdf
- https://www.habau.at/de/projekte/erdgasleitung-wag-expansion-3
- https://www.habau.at/de/projekte/erdgasleitung-wag-plus-600-phase-1
- https://www.hydrocarbons-technology.com/projects/swedegas-lng-facility-port-gothenburg/
- https://www.ign.ren.pt/en/armazenamento-subterraneo3
- https://www.ign.ren.pt/en/armazenamento-subterraneo4
- https://www.innogy-gasstorage.cz/en/index/
- https://www.innogy-gasstorage.cz/en/stramberk/
- https://www.n-ergie.de/geschaeftskunden/produkte/erdgas/erdgasspeicher.html
- https://www.ontras.com/fileadmin/user_upload/Dokumente_Download/Publikationen/ONTRAS_Netzpufferanlage_Burggraf_Bernsdorf.pdf

- https://www.osm.pgnig.pl/en
- https://www.rnf.de/hintergrund-die-erm-gasleitung-von-ludwigshafen-nach-karlsruhe-58182/
- https://www.shz.de/lokales/landeszeitung/wissenschaftler-unter-zeitdruck-id5845016.html
- https://www.sourcewatch.org/index.php/Lithuania-Latvia_Interconnection_Gas_Pipeline
- https://www.storengy.com/countries/france/en/nos-sites/beynes.html
- https://www.storengy.com/countries/france/en/nos-sites/cere-la-ronde.html
- https://www.storengy.com/countries/france/en/nos-sites/manosque.html
- https://www.storengy.com/countries/france/en/our-sites/saint-clair-sur-epte.html
- https://www.swedegas.com/Our_services/services/Storage
- https://www.uniper-energy-storage.com/cps/rde/xchg/ust/hs.xsl/3252.htm?rdeLocaleAttr=en
- https://www.uniper-energy-storage.com/cps/rde/xchg/ust/hs.xsl/3437.htm?rdeLocaleAttr=en
- https://www.vng-gasspeicher.de/storage_locations
- https://www.wesernetz.de/netznutzung/bremen/gasnetz-speicheranlagen.php
- https://www.wingas.com/storage-uk-ltd/home.html
- https://www.PLE
- https://www.www.gascade.de/Verdichterstation_Bunde_2016.pdf
- https://www.www.gascade.de/Compressor_station_Eischleben_2016.pdf
- https://www.www.gascade.de/Verdichterstation_Lippe_72dpi060614.pdf
- https://www.www.gascade.de/Anlandestation_Greifswald_Lubmin_2016.pdf
- https://www.www.gascade.de/Verdichterstation_Mallnow_2016.pdf
- https://www.www.gascade.de/Compressor_station_Olbernhau_2016.pdf
- https://www.www.gascade.de/Verdichterstation_Reckrod_2016.pdf
- https://www.www.gascade.de/Verdichterstation_Rehden_2016.pdf
- https://www.www.gascade.de/Compressor_station_Rueckersdorf_2016.pdf
- https://www.www.gascade.de/Verdichterstation_Weisweiler_201609.pdf
- https://www.opal-gastransport.de/Compressor_Station_Radeland_72dpi.pdf
- https://www.opal-gastransport.de/Verdichterstation_Radeland_2016.pdf
- https://www.pipelinesystems.com/: "NETRA compressor station Wardenburg"
- https://www.grtgaz.com: Plan_decennal_2017-2026.pdf
- https://www.streicher.de
- https://www.open-grid-europe.com
- https://www.open-grid-europe.com
- https://www.open-grid-europe.com, Pressinformation, May 2017, "Herbstein compressor station new build project"
- https://www.fluxys.com/tenp/de
- Presseinfo Bayernets, WinGas, 19-Sep-2008

- NWZonline.de, 22-04-2010: "ExxonMobil gibt kräftig Gas" by Tanja Mikulski
- Porzerleben.de: 6-sep-2011, "Open Grid Europe investiert in europäischen Netzverbund"
- Christoph Edler, Bachelorarbeit PR 370005, Technische Universität Wien, "Das österreichische Gasnetz", Juli 2013

6.4 Location name alterations

Location names should be changed into the 26 letters used in the English language.

For names from the individual countries please follow the suggested approach:

- Germany/Austria: *Umlaute* to be replaced with the letter followed by an 'e', e.g.: ü = ue.
- France/Belgium: Omit accent de gues and accent de graphs, e.g.: ó = o.
- Sweden: Please change the last three letters of the Swedish alphabet and replace e.g.: ä = a.
- Poland: Please change any letter, that cannot be found in the English alphabet, knowing that for some letters, that one can only use a single letter instead of the three different letters used in the Polish alphabet, e.g.: z = z.
- Spain/Portugal: Please change any letter, that cannot be found in the English alphabet, e.g.: ñ = n.
- Greece: Please do not use Greek letters. Please try to write the Greek words with Latin letters.
- Denmark: Please change any letter that contains non-English letters, e.g.: "å" with "aa".
- Slovakia, Czech Republic, Hungary, Rumania, Latvia, Lithuania, Estonia, Bulgaria, Slovenia, Croatia: PLEASE use your common sense, based on the examples from the other countries above.

6.5 Country name abbreviations

For convenience we provide a short list of names and two-digit codes (see Table 6.5) for the probably most important countries associated with the European Transmission Grid.

Table 6.5: Country codes

Country name	Country code	Country name	Country code
Albania	AL	Kosovo	XK
Armenia	AM	Latvia	LV
Austria	AT	Liechtenstein	LI
Azerbaijan	AZ	Lithuania	LT
Belarus	BY	Luxembourg	LU
Belgium	BE	Malta	MT
Bosnia and Herzegovina	BA	Moldova	MD
Bulgaria	BG	Montenegro	ME
Croatia	HR	Netherlands	NL
Cyprus	CY	Norway	NO
Czech	CZ	Poland	PL
Denmark	DK	Portugal	PT
Estonia	EE	Romania	RO
Finland	FI	Serbia	RS
France	FR	Slovakia	SK
Georgia	GE	Slovenia	SI
Germany	DE	Spain	ES
Greece	GR	Sweden	SE
Hungary	HU	Switzerland	CH
Iceland	IS	Turkey	TR
Ireland and Northern Ireland	IE	Belarus	UA
Italy	IT	Great Britain	GB
Russia Federation	RU	Europe	EU

6.6 Statistical background

Here some of the statistical methods mentioned in the document are described briefly. This is so that actions described in this document can be understood better by the user, and is not thought of giving a full explanation. Most descriptions have been copied from the Wikipedia pages, or other internet pages, and will be referenced accordingly.

6.6.1 Out-of-bag

This is a method of measuring the prediction error of random forests, boosted decision trees, and other machine learning models utilizing bootstrap aggregating (bagging) to sub-sample data samples used for training. [Wik20f]

6.6.2 Leave p-out cross-validation

The following has been copied from [Wik20b]:

"Leave-p-out cross-validation (LpO CV) involves using p observations as the validation set and the remaining observations as the training set. This is repeated on all ways to cut the original sample on a validation set of p observations and a training set.

LpO cross-validation requires training and validating the model C_p^n times, where *n* is the number of observations in the original sample, and where C_p^n is the binomial coefficient. For *p > 1* and for even moderately large *n*, LpO CV can become computationally infeasible. For example, with *n = 100* and *p = 30 %* of *100*, $C_{30}^{100} \approx 3 \times 10^{25}$."

6.6.3 Leave one-out cross-validation

The following has been copied from [Wik20b]:

"Leave-one-out cross-validation (LOOCV) is a particular case of leave-p-out cross-validation with $p = 1$.

The process looks similar to jackknife; however, with cross-validation one computes a statistic on the left-out sample(s), while with jackknifing one computes a statistic from the kept samples only.

LOO cross-validation requires less computation time than LpO cross-validation because there are only $\mathrm{C}_1^n = \mathrm{n}$ passes rather than C_k^n. However, n passes may still require quite a large computation time, in which case other approaches such as k-fold cross validation may be more appropriate."

6.6.4 Jackknifing

The following text has been copied from [Wik20c]:

"In statistics, the jackknife is a resampling technique especially useful for variance and bias estimation. The jackknife pre-dates other common resampling methods such as the bootstrap. The jackknife estimator of a parameter is found by systematically leaving out each observation from a data set and calculating the estimate and then finding the average of these calculations. Given a sample of size n , the jackknife estimate is found by aggregating the estimates of each

The jackknife technique was developed by Maurice Quenouille (1924-1973) from 1949, and refined in 1956. John Tukey expanded on the technique in 1958 and proposed the name "jackknife" since, like a physical jack-knife (a compact folding knife), it is a rough-and-ready tool that can improvise a solution for a variety of problems even though specific problems may be more efficiently solved with a purpose-designed tool.

The jackknife is a linear approximation of the bootstrap."

6.6.5 Bootstrap

The following has been copied from [Wik20a]:

"Bootstrapping is any test or metric that uses random sampling with replacement, and falls under the broader class of resampling methods. Bootstrapping assigns measures of accuracy (bias, variance, confidence intervals, prediction error, etc.) to sample estimates. This technique allows estimation of the sampling distribution of almost any statistic using random sampling methods.

Bootstrapping estimates the properties of an estimator (such as its variance) by measuring those properties when sampling from an approximating distribution. One standard choice for an approximating distribution is the empirical distribution function of the observed data. In the case where a set of observations can be assumed to be from an independent and identically distributed population, this can be implemented by constructing a number of resamples with replacement, of the observed data set (and of equal size to the observed data set)."

6.7 Acknowledgement

We acknowledge the contribution of Dr. Ontje Luensdorf from the DLR Institute of Networked Energy System to the SciGRID_gas project.

BIBLIOGRAPHY

[AFW14] M. Ahmed, B.T. Fasy, and C. Wenk. *New Techniques in Road Network Comparison*. Penguin Random House, New York, NY, 2014.

[AG99] H. Alt and L. Guibas. *Discrete geometric shapes: matching, interpolation, and approximation-a survey*. Sack JR, Urrutia J, Handbook of Computational Geometry, Elsevier, New York, NY, 1999.

[DPM20a] J.C. Diettrich, A. Pluta, and W. Medjroub. *SciGRID_gas: The INET gas transmission network data set*. DLR-Institut für Vernetzte Energiesysteme e.V., Oldenburg, Germany, 2020.

[DPM20b] J.C. Diettrich, A. Pluta, and W. Medjroub. *SciGRID_gas: The combined IGG gas transmission network data set*. DLR-Institut für Vernetzte Energiesysteme e.V., Oldenburg, Germany, 2020.

[FMWP+17] Kunz F., Kendziorski M., Schill W.-P., Weibezahn J., Zepter J., von Hirschhausen C., and Hauser P. *Electricity, Heat, and Gas Sector Data for Modeling the German System*. Deutsches Institut für Wirtschaftsforschung, Daten Documentation 92, Berlin, 2017.

[GH85] R.L. Graham and P. Hell. On the history of the minimum spanning tree problem. *Annauls of the History of Computer*, 7(1):43–57, 1985. doi:{10.1145/2729977}.

[Hel18] D. Helle. OpenStreetMap - Deutschland. https://www.openstreetmap.de/, 2018. Accessed: 2019-12-12.

[Kha13] Y. Khalid. What is Pickle in python? https://pythontips.com/2013/08/02/what-is-pickle-in-python/, 2013. Accessed: 2019-10-10.

[LSS+19] P. Lustenberger, F. Schumacher, M. Spada, P. Burgherr, and B. Stojadinovic. Assessing the performance of the european natural gas network for selected supply disruption scenarios using open-source information. *Energies*, 12(4685):1–28, 2019. doi:{10.3390/en12244685}.

[MMK16] C. Matke, W. Medjroubi, and D. Kleinhans. SciGRID - An Open Source Reference Model for the European Transmission Network (v0.2). https://power.scigrid.de, 2016. Accessed: 2019-09-09.

[San19] B. Sandvik. World Borders. http://thematicmapping.org/downloads/world_borders.php, 2019. Accessed: 2019-07-07.

[SAB+17] M. Schmidt, D. Aßmann, R. Burlacu, J. Humpola, I. Joormann, N. Kanelakis, T. Koch, D. Oucherif, M.E. Pfetsch, L. Schewe, R. Schwarz, and M. Sirvent. *GasLib—A Library of Gas Network Instances*. 2017. doi:{10.3390/data2040040}.

[sl19] scikit-learn. 1.1. Linear Models (scikit learn). https://scikit-learn.org/stable/modules/linear_model.html, 2019. Accessed: 2019-08-08.

[UoO14] USA University of Oregon. Comparing distributions: Z Test. http://homework.uoregon.edu/pub/class/es202/ztest.html, 2014. Accessed: 2020-07-07.

[Wik20a] Wikipedia. Bootstrapping (statistics). https://en.wikipedia.org/wiki/Bootstrapping_(statistics), 2020. Accessed: 2019-06-06.

[Wik20b] Wikipedia. Cross-validation (statistics). https://en.wikipedia.org/wiki/Cross-validation_(statistics)#Exhaustive_cross-validation, 2020. Accessed: 2019-07-07.

[Wik20c] Wikipedia. Jackknife resampling. https://en.wikipedia.org/wiki/Jackknife_resampling, 2020. Accessed: 2019-08-08.

[Wik20d] Wikipedia. Lasso (statistics). https://en.wikipedia.org/wiki/Lasso_(statistics), 2020. Accessed: 2020-04-04.

[Wik20e] Wikipedia. Limited-memory BFGS. https://en.wikipedia.org/wiki/Limited-memory_BFGS, 2020. Accessed: 2020-06-06.

[Wik20f] Wikipedia. Out-of-bag error. https://en.wikipedia.org/wiki/Out-of-bag_error, 2020. Accessed: 2019-07-07.

[Wik20g] Wikipedia. Transmission system operator. https://en.wikipedia.org/wiki/Transmission_system_operator/, 2020. Accessed: 2019-09-09.

[Wik20h] Wikipedia. JAGAL. https://en.wikipedia.org/wiki/JAGAL, 2020. Accessed: 2020-01-01.

[YEZ20] I. Yueksel-Erguen and J. Zittel. Approach of converting a PDF map into shape file. priv. comms, 2020.

[BMWi11] BMWi. Forschung für eine umweltschonende, zuverlässige und bezahlbare Energieversorgung. https://www.bmwi.de/Redaktion/DE/Publikationen/Energie/6-energieforschungsprogramm-der-bundesregierung.pdf?__blob=publicationFile&v=12, 2011. Accessed: 2019-02-02.

[BMWi20] BMWi. Home page of BMWi. https://www.bmwi.de/Navigation/DE/Home/home.html, 2020. Accessed: 2020-03-03.

[BundesregierungDeutschland20] Bundesregierung Deutschland. Home page of Bundesregierung Deutschland. https://www.bundesregierung.de/Webs/Breg/DE/Themen/Energiewende/_node.html, 2020. Accessed: 2020-01-01.

[GasIEurop20] Gas Infrastructure Europ. Home page of Gas Infrastructure Europ. https://agsi.gie.eu, 2020. Accessed: 2020-01-01.

[Gassco20a] Gassco. Data page of facilities from Gassco. https://www.npd.no/en/about-us/information-services/available-data/map-services/, 2020. Accessed: 2020-01-01.

[Gassco20b] Gassco. Home page of Gassco Norway. https://www.gassco.no/en/, 2020. Accessed: 2020-01-01.

[IGU20] IGU. Home page of International Gas Union. https://www.igu.org/, 2020. Accessed: 2018-10-01.

[nationalGrid20] nationalGrid. Home page of National Grid UK. https://www.nationalgrid.com/uk/, 2020. Accessed: 2018-10-01.