VesselAI, 2022: Information Extraction from PDFs, User Guide

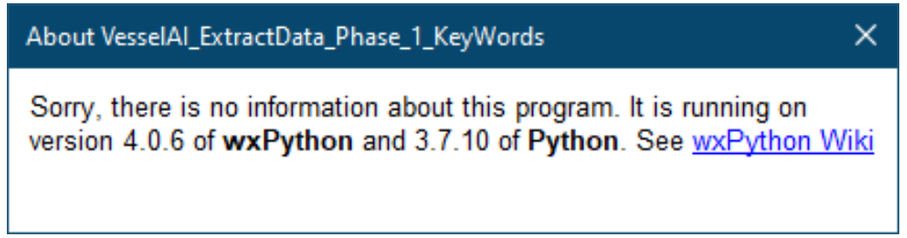
16/12/2022 VTT – beyond the obvious

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957237.

Trying to find marine engine specs & manuals from Internet sources (PDF format)

- We want to test how automatic extraction of information from PDF files would work
- The PDF files describe the performance of ship engines

- This presentation describes 4 Python programs, 3 first of which together implement a workflow for such information extraction from PDF files; NN version is separate
  - VesselAI_ExtractData_1.py
  - VesselAI_ExtractData_2.py
  - VesselAI_ExtractData_3.py
  - VesselAI_ExtractData_NN.py



- The programs were written using Anaconda Python, and wxPython UI library

- Introduction / general information in slides 2-12, the workflow & programs are described in slides 13-19, and installation instructions are in slides 20-22.

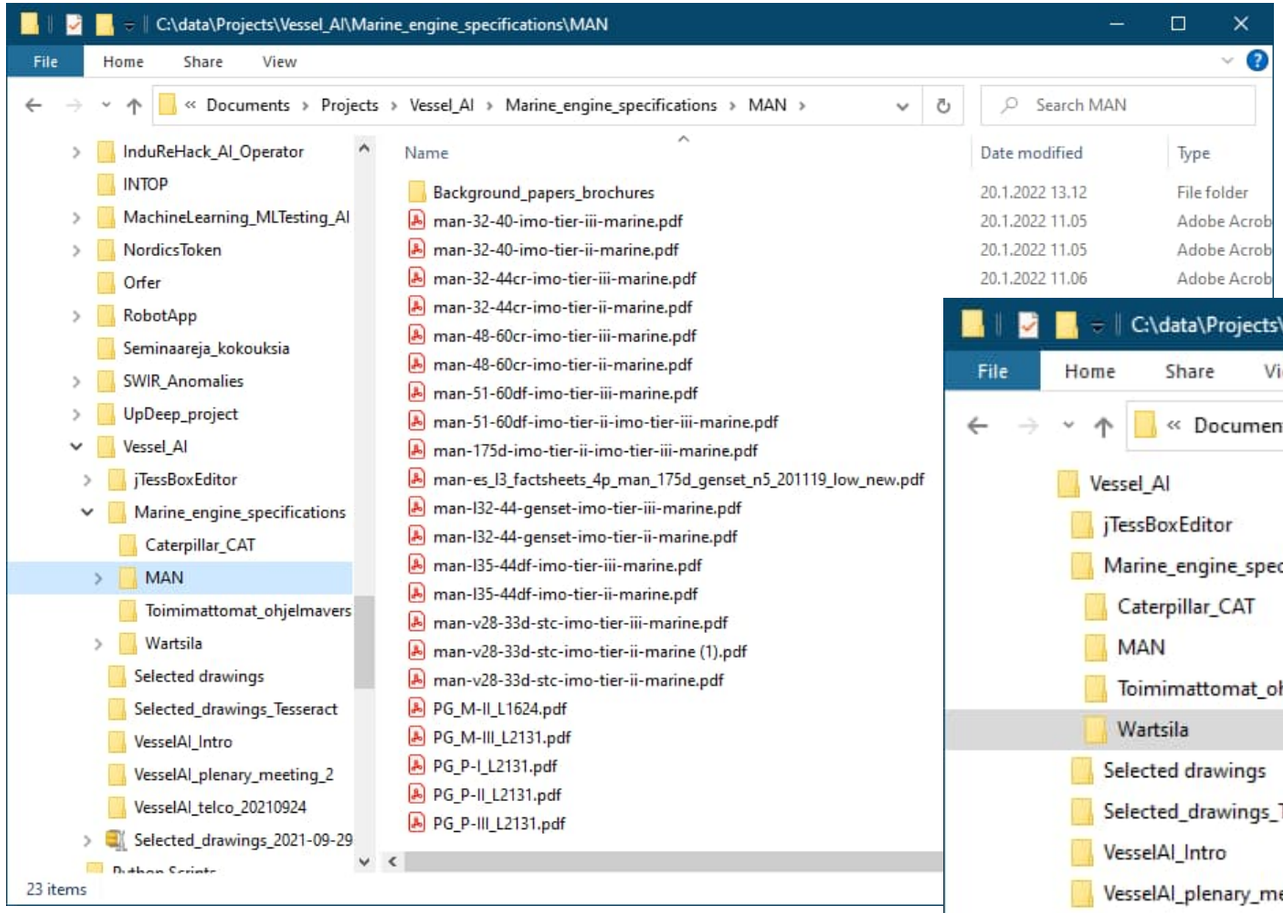
PDFs from largest ship engine manufacturers

- It was decided that we start with the 3 largest marine engine companies: Wärtsilä, MAN, and Caterpillar
- We downloaded all “technical looking” PDFs that consider marine engines
  - From Wärtsilä we downloaded 14 PDFs
  - From MAN we downloaded 22 PDFs
  - Alas, for Caterpillar / CAT there were no PDF specifications that could be freely downloaded…
- To make sure that our code is versatile enough, we later searched for PDF files in WinGD and HHI (Hyundai Heavy Industries) web pages also
  - Google search: engine specification “pdf” site:www.wingd.com
  - Google search: engine specification “pdf” site:english.hhi.co.kr
  - We downloaded 27 files from WinGD, and 11 files from HHI

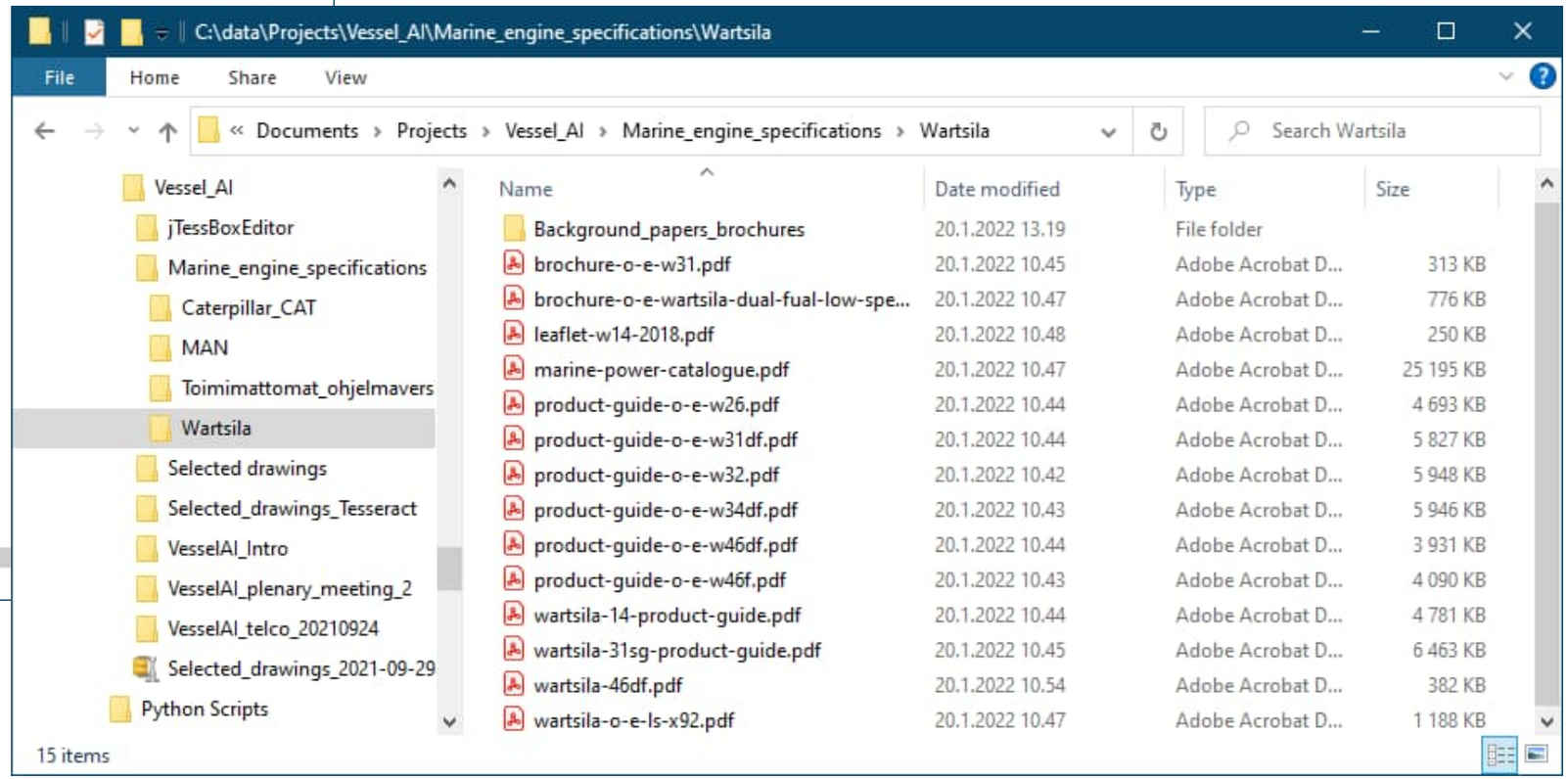
- This was our test material!

PDF specifications from Wärtsilä and MAN

- These are the Wärtsilä and MAN PDF files that we considered:



- 22 PDFs from MAN, 14 from Wärtsilä…
- 27 files from WinGD, 11 files from HHI



PDF files NOT included!

- There are 73 PDF files in total, containing 18 019 pages
- These files are (or at least were in spring 2022) publicly available
- However, due to copyright issues we cannot distribute them
- Lists of these files are presented in the next slides
- You must try to find them from the vendors' web sites and download them yourself
- Google searches for MAN, Wärtsilä, HHI, WinGD PDF files (**or:** "engine manual"…):
  - engine specification "pdf" site: www.man-es.com/marine/
  - engine specification "pdf" site: www.wartsila.com
  - engine specification "pdf" site:english.hhi.co.kr
  - engine specification "pdf" site:www.wingd.com
- Save the found files into subfolders 'MAN', 'Wartsila', 'Hyundai_HHI', and 'WinGD'
- However, the Python programs contain hard-wired lists of the files to be processed. If you cannot find all these listed files, you must modify these lists
- But the NN (neural network) program version also contains the indices of documents and pages that the human expert considered relevant ("Ground Truth"). These index lists must be modified also, which is tedious…

PDF files NOT included! MAN files to find

- MAN/man-175d-imo-tier-ii-imo-tier-iii-marine.pdf
- MAN/man-32-40-imo-tier-ii-marine.pdf
- MAN/man-32-40-imo-tier-iii-marine.pdf
- MAN/man-32-44cr-imo-tier-ii-marine.pdf
- MAN/man-32-44cr-imo-tier-iii-marine.pdf
- MAN/man-48-60cr-imo-tier-ii-marine.pdf
- MAN/man-48-60cr-imo-tier-iii-marine.pdf
- MAN/man-51-60df-imo-tier-ii-imo-tier-iii-marine.pdf
- MAN/man-51-60df-imo-tier-iii-marine.pdf
- MAN/man-es_l3_factsheets_4p_man_175d_genset_n5_201119_low_new.pdf
- MAN/man-l32-44-genset-imo-tier-ii-marine.pdf
- MAN/man-l32-44-genset-imo-tier-iii-marine.pdf
- MAN/man-l35-44df-imo-tier-ii-marine.pdf
- MAN/man-l35-44df-imo-tier-iii-marine.pdf
- MAN/man-v28-33d-stc-imo-tier-ii-marine.pdf
- MAN/man-v28-33d-stc-imo-tier-iii-marine.pdf
- MAN/PG_M-III_L2131.pdf
- MAN/PG_M-II_L1624.pdf
- MAN/PG_P-III_L2131.pdf
- MAN/PG_P-II_L2131.pdf
- MAN/PG_P-I_L2131.pdf

PDF files NOT included! Wärtsilä files to find

- Wartsila/brochure-o-e-w31.pdf
- Wartsila/brochure-o-e-wartsila-dual-fual-low-speed.pdf
- Wartsila/leaflet-w14-2018.pdf
- Wartsila/marine-power-catalogue.pdf
- Wartsila/product-guide-o-e-w26.pdf
- Wartsila/product-guide-o-e-w31df.pdf
- Wartsila/product-guide-o-e-w32.pdf
- Wartsila/product-guide-o-e-w34df.pdf
- Wartsila/product-guide-o-e-w46df.pdf
- Wartsila/product-guide-o-e-w46f.pdf
- Wartsila/wartsila-14-product-guide.pdf
- Wartsila/wartsila-31sg-product-guide.pdf
- Wartsila/wartsila-46df.pdf
- Wartsila/wartsila-o-e-ls-x92.pdf

PDF files NOT included! HHI files to find

- Hyundai_HHI/2016_HHI_en.pdf
- Hyundai_HHI/2017_HHI_en.pdf
- Hyundai_HHI/2018_HHI_en.pdf
- Hyundai_HHI/2019_HHI_en.pdf
- Hyundai_HHI/2020_KSOE_IR_EN.pdf
- Hyundai_HHI/2021_KSOE_IR_EN.pdf
- Hyundai_HHI/HHI_EMD_brochure2017_1.pdf
- Hyundai_HHI/HHI_EMD_brochure2017_2.pdf
- Hyundai_HHI/HHI_EMD_brochure2017_3.pdf
- Hyundai_HHI/HHI_EMD_brochure2019_1.pdf
- Hyundai_HHI/speclal_NSD2020.pdf

PDF files NOT included! WinGD files to find

- WinGD/16-06-pres-kit_x92.pdf
- WinGD/cimac2016_120_kyrtatos_paper_thedevelopmentofthemodern2strokemarinedieselengine.pdf
- WinGD/cimac2016_173_brueckl_paper_virtualdesign-and-simulation-in2strokemarineenginedevelopment.pdf
- WinGD/cimac2016_233_ott_paper_x-df.pdf
- WinGD/dominikschneiter_tieriii-programme.pdf
- WinGD/Fuel-Flexible-Injection-System-How-to-Handle-a-Fuel-Specturm-from-Diesel_CIMAC2019_paper_404_Andreas-Schmid.pdf
- WinGD/marcspahni_generation-x-engines.pdf
- WinGD/MIM_WinGD-RT-flex48T-D.pdf
- WinGD/MIM_WinGD-X72.pdf
- WinGD/MM_WinGD-RT-flex58T-D.pdf
- WinGD/MM_WinGD-X35-B_2021-09.pdf
- WinGD/MM_WinGD-X52.pdf
- WinGD/MM_WinGD-X72.pdf
- WinGD/MM_WinGD-X72DF.pdf
- WinGD/MM_WinGD-X82-B.pdf
- WinGD/Motorship-May-2018-VP-R-D-Leader-Brief.pdf
- WinGD/OM_WinGD-X62_2021-09.pdf
- WinGD/OM_WinGD-X72DF_2021-09.pdf
- WinGD/OM_WinGD-X82-B.pdf
- WinGD/OM_WinGD_RT-flex50DF.pdf
- WinGD/WinGD-12X92DF-Development-of-the-Most-Powerful-Otto-Engine-Ever_CIMAC2019_paper_425_Dominik-Schneiter.pdf
- WinGD/wingd-paper_engine_selection_for_very_large_container_vessels_201609.pdf
- WinGD/WinGD-WiDE-Brochure.pdf
- WinGD/WinGD_Engine-Booklet_2021.pdf
- WinGD/wingd_moving-inlet-ports-concept-for-optimization-of-2-stroke-uni-flow-engines_patrick-rebecchi.pdf
- WinGD/X-DF-Engines-by-WinGD.pdf
- WinGD/X-DF-FAQ.pdf

Wärtsilä PDF files: Relevant tables

- Then a few of the Wärtsilä PDF files were inspected
- The first one that contains relevant numerical information was ‘product-guide-o-e-w26.pdf’
- It contains many tables of the same format
- The page that contains the first such table is shown here:
- The four last lines give **fuel consumption** at 100% / 85 % / 75 % / 50 % engine load [g/kWh]
- The four first lines in group “Exhaust gas system” give **exhaust gas flow** at 100% / 85 % / 75 % / 50 % engine load [kg/s]
- There was no **charge air flow**; however, the first line in group “Combustion air system” gives “Flow of air at 100 % load” [kg/s]; we will use these numbers instead
- Header data above is crucial; we want to know the engine model etc. to which the numbers apply!

3.2 IMO Tier 2

3.2.1 Wärtsilä 6L26

Table 3-3

Wärtsilä 6L26		AE/DE IMO Tier 2	AE/DE IMO Tier 2	ME IMO Tier 2	ME IMO Tier 2
Cylinder output	**kW/cyl**	**325**	**340**	**325**	**340**
Engine speed	**rpm**	**900**	**1000**	**900**	**1000**
Engine output	kW	1950	2040	1950	2040
Mean effective pressure	MPa	2.55	2.4	2.55	2.4
Combustion air system (Note 1)					
Flow of air at 100% load	kg/s	3.7	4.1	3.9	4.1
Temperature at turbocharger intake, max.	°C	45	45	45	45
Air temperature after air cooler, nom. (TE601)	°C	55	55	55	55
Exhaust gas system (Note 2)					
Flow at 100% load	kg/s	3.8	4.2	4.0	4.2
Flow at 85% load	kg/s	3.3	3.7	3.4	3.5
Flow 75% load	kg/s	3.0	3.4	3.0	3.1
Flow 50% load	kg/s	2.1	2.3	1.9	2.2
Temp. after turbo, 100% load (TE517)	°C	343	324	318	324
Temp. after turbo, 85% load (TE517)	°C	343	319	327	328
Temp. after turbo, 75% load (TE517)	°C	349	321	339	341
Temp. after turbo, 50% load (TE517)	°C	367	344	402	388
Backpressure, max.	kPa	3.0	3.0	3.0	3.0
Exhaust gas pipe diameter, min	mm	500	500	500	500
Calculated exhaust diameter for 35 m/s	mm	492	507	493	507
Heat balance (Note 3)					
Jacket water, HT circuit	kW	348	372	336	372
Charge air, LT-circuit	kW	611	723	689	723
Lubricating oil, LT-circuit	kW	288	306	282	306
Radiation	kW	92	97	92	97
Fuel system (Note 4)					
Pressure before injection pumps (PT101)	kPa	700±50	700±50	700±50	700±50
Engine driven pump capacity at 12 cSt (MDF only)	m³/h	2.9	3.2	2.9	3.2
Fuel flow to engine (without engine driven pump), approx.	m³/h	1.6	1.8	1.7	1.8
HFO viscosity before engine	cSt	16...24	16...24	16...24	16...24
HFO temperature before engine, max. (TE 101)	°C	140	140	140	140
MDF viscosity, min	cSt	2.0	2.0	2.0	2.0
MDF temperature before engine, max. (TE 101)	°C	45	45	45	45
Fuel consumption at 100% load	g/kWh	190.6	194.4	191.5	194.4
Fuel consumption at 85% load	g/kWh	189.6	193.4	188.7	191.5
Fuel consumption at 75% load	g/kWh	192.0	195.3	190.6	193.4
Fuel consumption at 50% load	g/kWh	202.3	207.0	196.6	201.3

PDF files: Relevant tables

- After much testing we settled for the following rules:
- We first search for these keywords (can be inside () or []):
  - g/kWh
  - kJ/kWh
  - kg/s
  - kg/kWh
- But with two upper keywords the following other words must also be found in the same page:
  - fuel **AND** consumption
  - **OR** sfoc **OR** bsfc
- And with two lower keywords the following other words must also be found in the same page:
  - charge air **OR** combustion air **OR** exhaust gas
  - **AND** flow
- This produced good enough results!

3.2 IMO Tier 2

3.2.1 Wärtsilä 6L26

Table 3-3

Wärtsilä 6L26		AE/DE IMO Tier 2	AE/DE IMO Tier 2	ME IMO Tier 2	ME IMO Tier 2
Cylinder output	**kW/cyl**	**325**	**340**	**325**	**340**
Engine speed	**rpm**	**900**	**1000**	**900**	**1000**
Engine output	kW	1950	2040	1950	2040
Mean effective pressure	MPa	2.55	2.4	2.55	2.4
Combustion air system (Note 1)					
Flow of air at 100% load	kg/s	3.7	4.1	3.9	4.1
Temperature at turbocharger intake, max.	°C	45	45	45	45
Air temperature after air cooler, nom. (TE601)	°C	55	55	55	55
Exhaust gas system (Note 2)					
Flow at 100% load	kg/s	3.8	4.2	4.0	4.2
Flow at 85% load	kg/s	3.3	3.7	3.4	3.5
Flow 75% load	kg/s	3.0	3.4	3.0	3.1
Flow 50% load	kg/s	2.1	2.3	1.9	2.2
Temp. after turbo, 100% load (TE517)	°C	343	324	318	324
Temp. after turbo, 85% load (TE517)	°C	343	319	327	328
Temp. after turbo, 75% load (TE517)	°C	349	321	339	341
Temp. after turbo, 50% load (TE517)	°C	367	344	402	388
Backpressure, max.	kPa	3.0	3.0	3.0	3.0
Exhaust gas pipe diameter, min	mm	500	500	500	500
Calculated exhaust diameter for 35 m/s	mm	492	507	493	507
Heat balance (Note 3)					
Jacket water, HT circuit	kW	348	372	336	372
Charge air, LT-circuit	kW	611	723	689	723
Lubricating oil, LT-circuit	kW	288	306	282	306
Radiation	kW	92	97	92	97
Fuel system (Note 4)					
Pressure before injection pumps (PT101)	kPa	700±50	700±50	700±50	700±50
Engine driven pump capacity at 12 cSt (MDF only)	m³/h	2.9	3.2	2.9	3.2
Fuel flow to engine (without engine driven pump), approx.	m³/h	1.6	1.8	1.7	1.8
HFO viscosity before engine	cSt	16...24	16...24	16...24	16...24
HFO temperature before engine, max. (TE 101)	°C	140	140	140	140
MDF viscosity, min	cSt	2.0	2.0	2.0	2.0
MDF temperature before engine, max. (TE 101)	°C	45	45	45	45
Fuel consumption at 100% load	g/kWh	190.6	194.4	191.5	194.4
Fuel consumption at 85% load	g/kWh	189.6	193.4	188.7	191.5
Fuel consumption at 75% load	g/kWh	192.0	195.3	190.6	193.4
Fuel consumption at 50% load	g/kWh	202.3	207.0	196.6	201.3

PDF files: Learning Program

- Another version of the original Python program was written
- Instead of selecting keywords (e.g. “kJ/kWh”, “fuel”, “consumption”; see previous slide) we select a few of our data pages as “Model Pages”
- For example, some Model Pages could be related to fuel consumption, some to charge air / exhaust gas / heat balance
- Then the program computes the relative frequencies of all the words in each Model Page
- And when a candidate page is processed, a similar word frequency histogram is computed, and compared to the frequency histograms of all the Model Pages
- Words that can be converted to floating-point numbers are excluded from the diff. sums
- If the best difference is small enough (e.g. <=0.5), i.e. that candidate page matches at least one of the Model Pages, then that candidate page is accepted

PDF files: Whole workflow

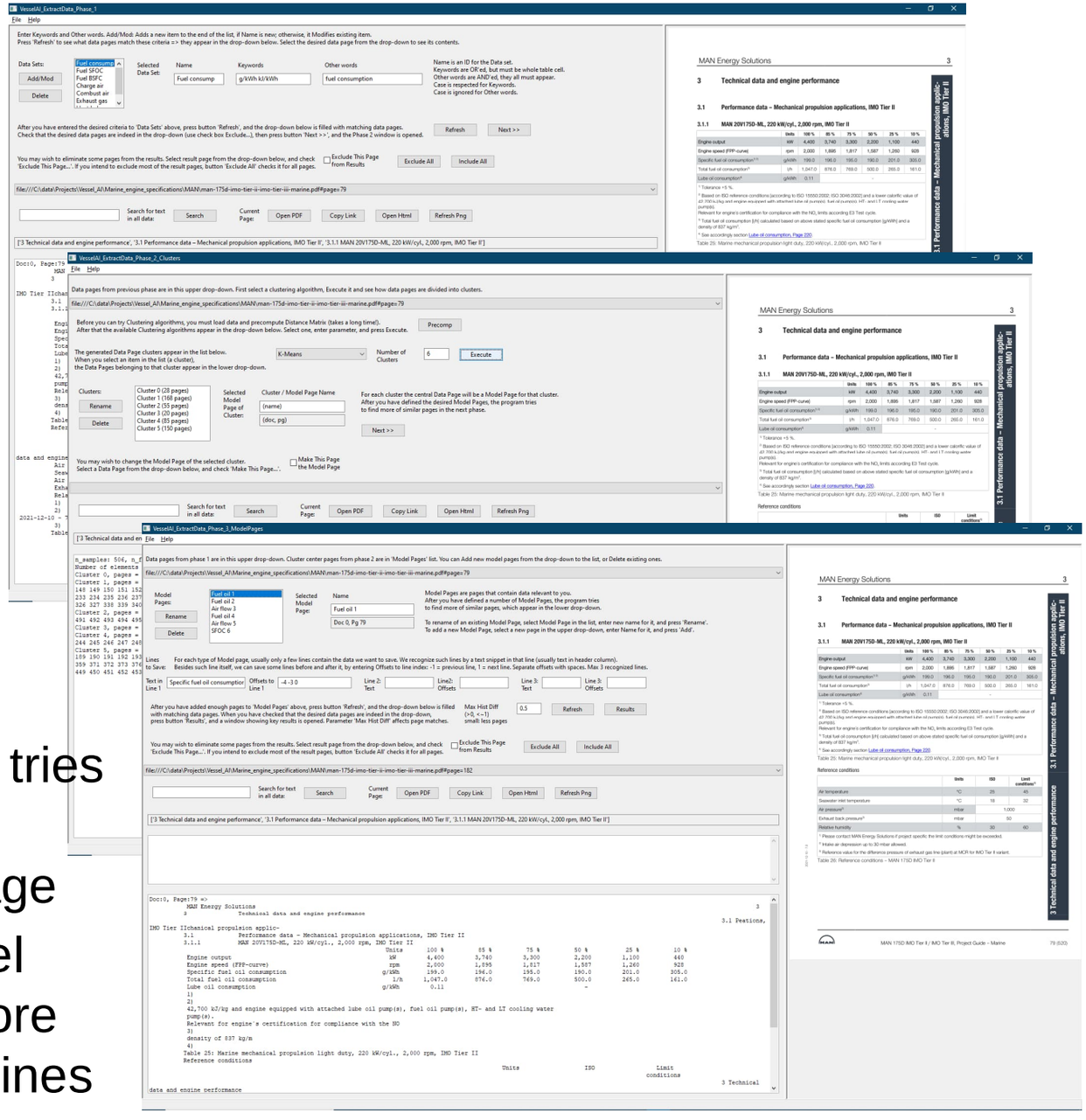
- **New versions of the Python programs discussed previously were written, so that together the new programs create a whole workflow for information extraction:**

1. We extend the first version of our program (using keywords) so that the user enters some keywords (e.g. fuel consumption). Then the program finds those *.pdf file **pages** that match these keywords. Optionally the user can eliminate some (irrelevant) pages. The remaining pages go to phase 2.
2. We cluster the found pages into groups (i.e. clusters); group center pages will be the proposed Model Pages. Optionally the user can eliminate some (irrelevant) Model Pages. The remaining pages go to phase 3.
3. We extend the second version of our program (learning version) to use these Model Pages, and then to find more of similar pages within the PDF files. Besides, for each model page the user can define rules for lines where the relevant data values are, thus reducing the amount of irrelevant information even further.

- And now we have a toolchain that can extract information from a large number of PDF files: Return desired parameter values for desired marine engine types.

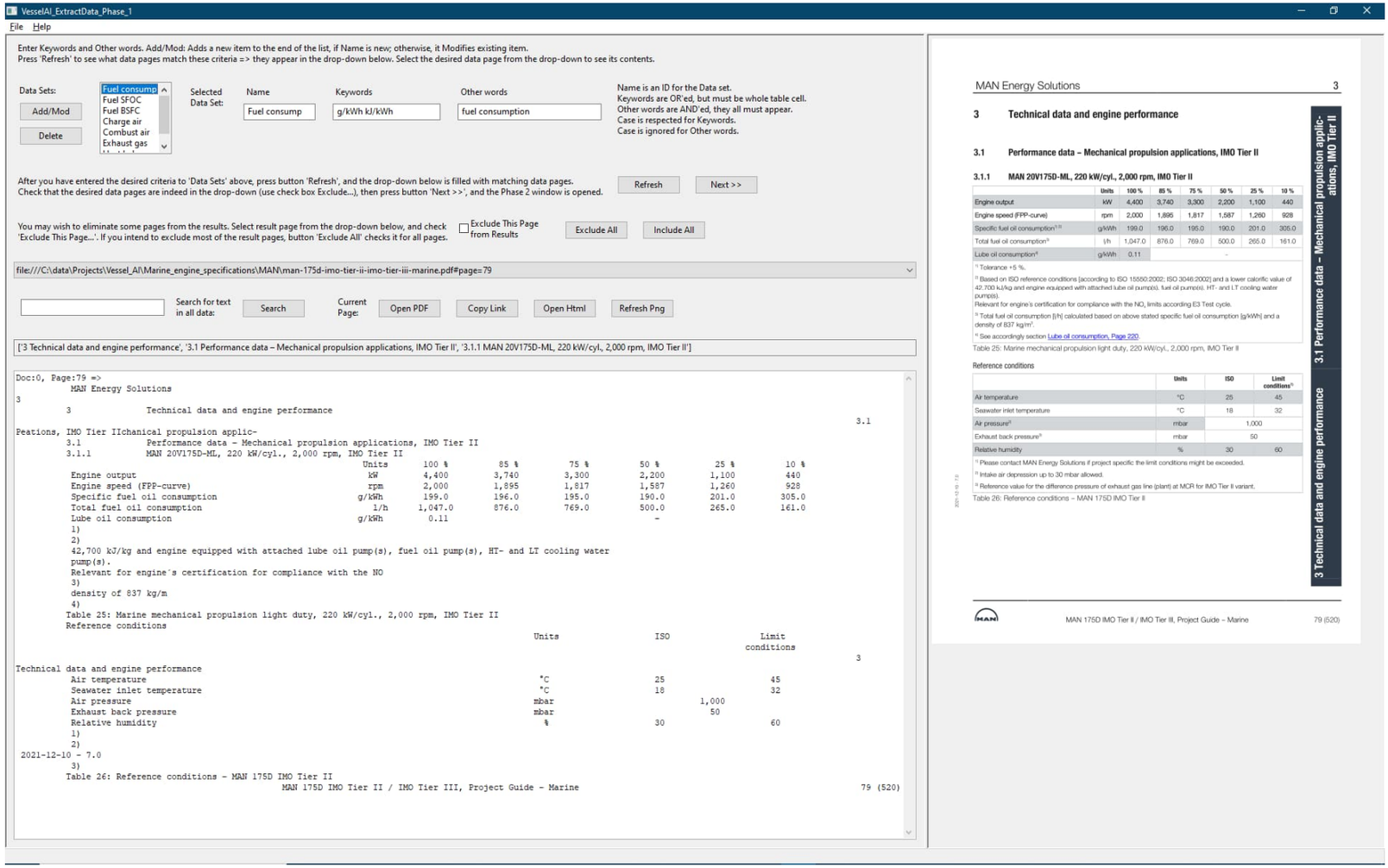
Implementation of the workflow (1/4)

- Still using our set of downloaded PDF files
- Can be later extended to web crawlers…
- Three separate Python programs to realize the workflow:
  - VesselAI_ExtractData_1.py
  - VesselAI_ExtractData_2.py
  - VesselAI_ExtractData_3.py
- In Phase 1 the user enters keywords and other words, and the program finds the matching files/pages; links to these files/pages are saved
- The Phase 2 program reads these files, and tries to group into clusters (4 algorithms); central page in each cluster is saved as a Model Page
- In Phase 3 the user can edit this set of Model Pages; and then the program tries to find more of similar pages, as well as extract relevant lines



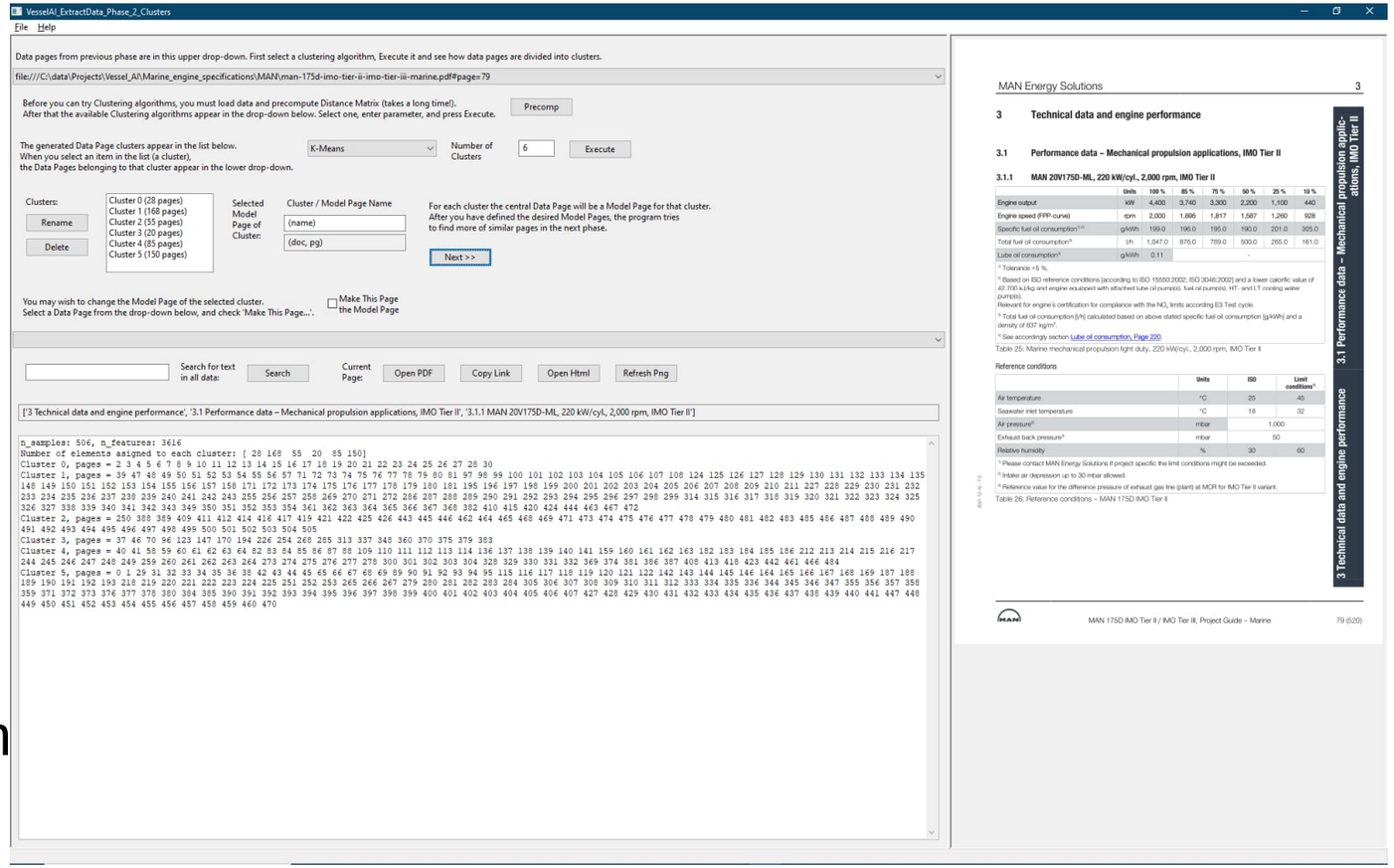
Implementation of the workflow (2/4): Phase 1

- The user enters a number of Data Sets, each contains Name (=ID for DS), Keywords, Other words; the list of defined Data Sets is shown
- The Data Set definitions are saved for next use. At first use: reasonable default values
- The algorithm of this program is executed with button ‘Refresh’, and the found files/pages are shown in a drop-down, and the selected page can be seen as an image and as text
- User can exclude some of the result pages from results
- Phase 2 program is started with button ‘Next >>’



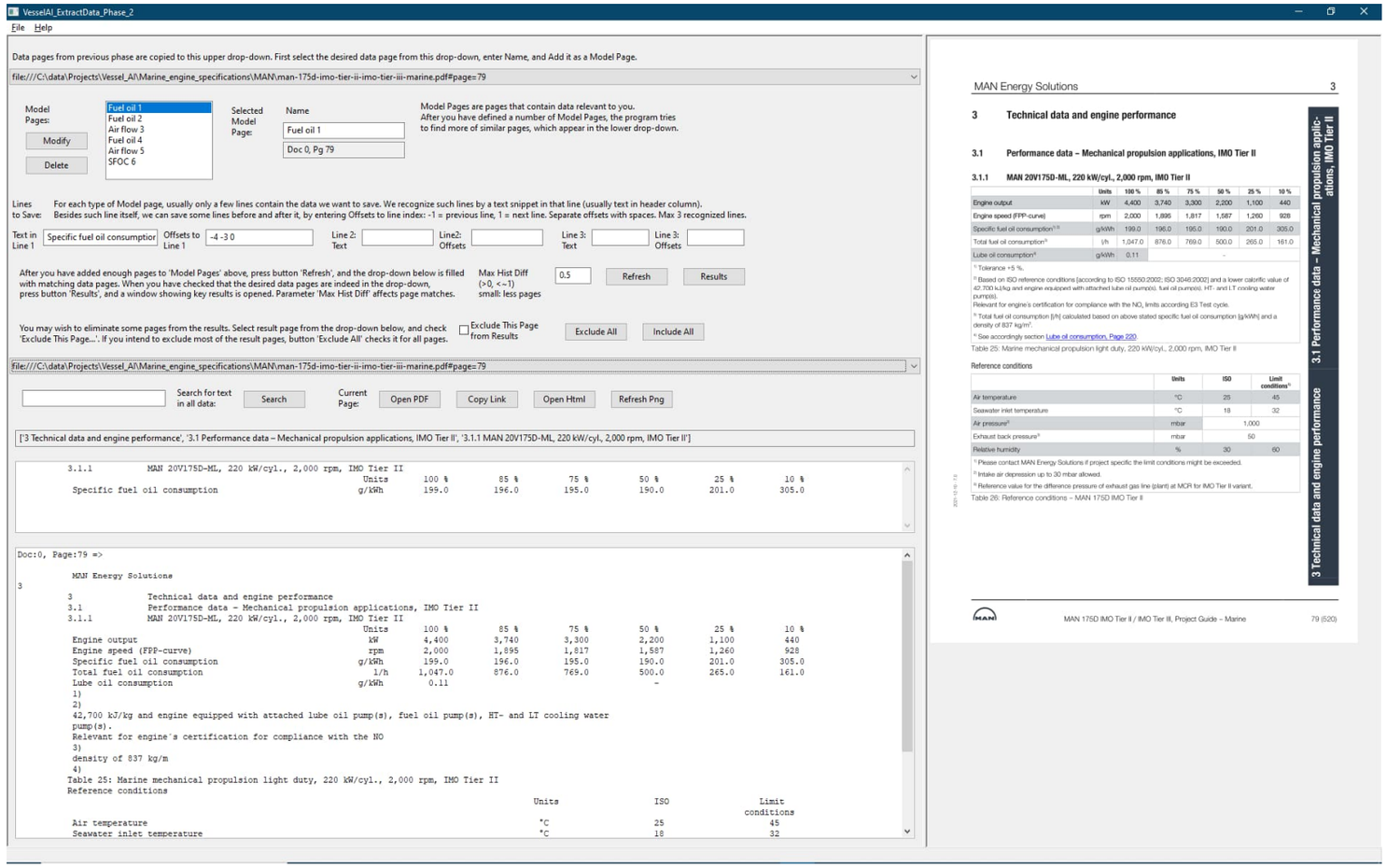
Implementation of the workflow (3/4): Phase 2

- The program tries to group Phase 1 result pages into clusters
- The central page of each cluster becomes a Model Page for Phase 3
- At first two algorithms for page clustering were available: **K-means** and **DBSCAN**; both from scikit learn (a.k.a. sklearn) program library:
  - https://scikit-learn.org/stable/auto_examples/text/plot_document_clustering.html#sphx-glr-auto-examples-text-plot-document-clustering-py
  - (“Clustering text documents using k-means”)
  - https://programmer.group/61755e726f9b3.html
  - (“Self defined distance measurement method of clustering algorithm in scikit learn Library”)
  - Later **AffinityPropagation** and **SpectralClustering** algorithms from scikit learn were also added!
- A lengthy Precomputation step is necessary before the 4 algorithms can be tried, but clustering itself is fast



Implementation of the workflow (4/4): Phase 3

- The user selects a number Phase 2 result pages as Model Pages, each contains Name (=ID for MP) and rules for lines of relevant data; the list of defined Model Pages is shown
- The contents of this list comes from Phase 2 (no data: reasonable default contents)
- For each Model Page, the user must define the lines where relevant data values are found
- The algorithm of this program is executed with button ‘Refresh’, and the found files/pages are shown in a drop-down, and the selected page can be seen as an image and as text. Data in relevant lines shown separately
- User can exclude some of the result pages from results
- Result data (=relevant lines) are shown with button ‘Results’. New Notepad window is opened

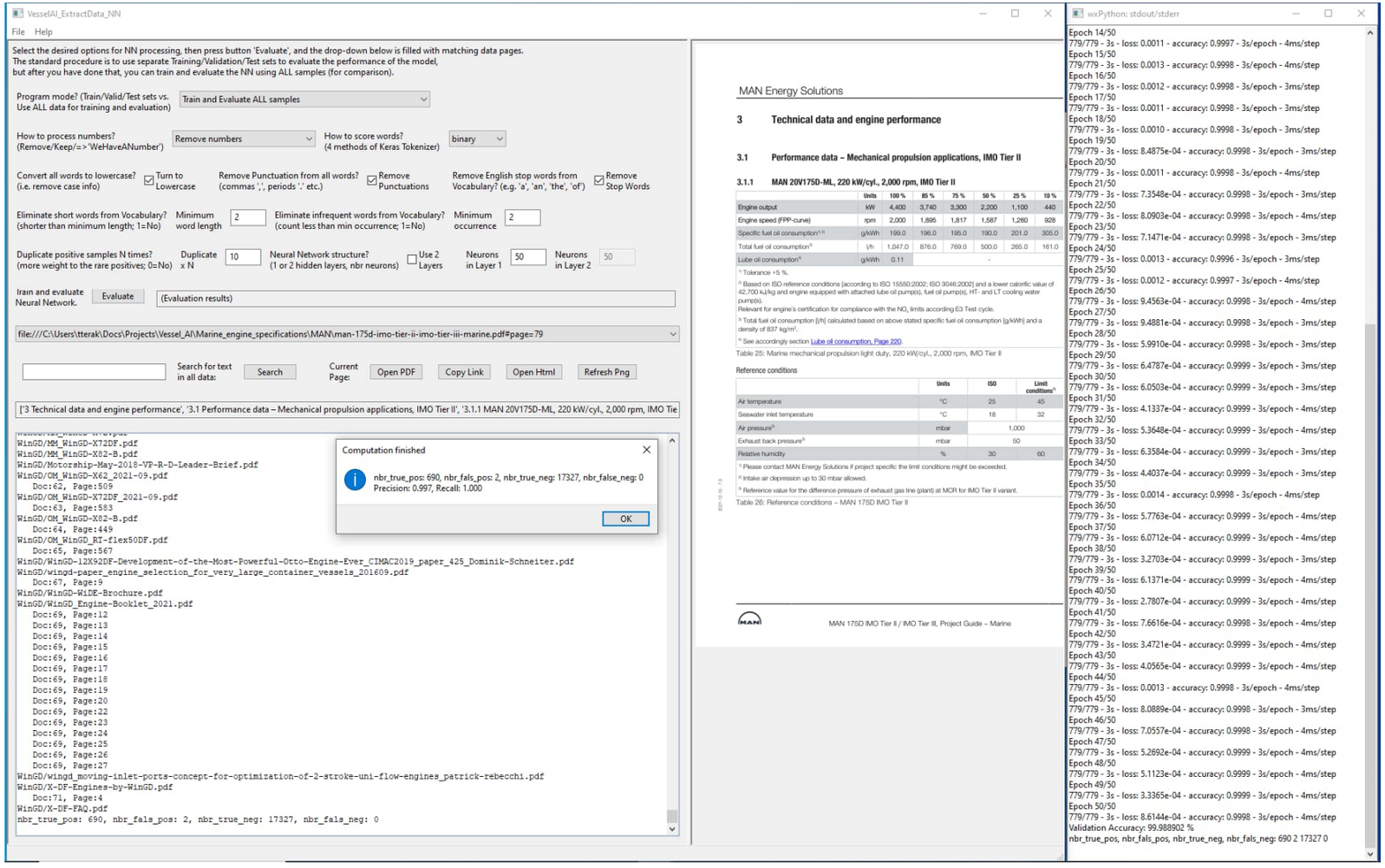


Notes

- Please note that when you open the window of the next phase (button ‘Next >>’), a new window is opened, but the original window remains open behind this new window; however, because the new window uses the same Python/wxPython environment as the original one, the original window remains inactive until the new window is closed
- Lengthy operations (e.g. ‘Precomp’ in Phase 2, ‘Refresh’ in Phases 1 and 3) are executed in the main thread, so the UI is frozen until the operation is completed. Progress is shown in the bottom subwindow, but this window is not always updated properly; especially if you move the focus into another window, the UI becomes frozen until the computation ends
- The programs have been tested in Windows 11, Anaconda Python; in other environments (e.g. Linux) there may be problems…

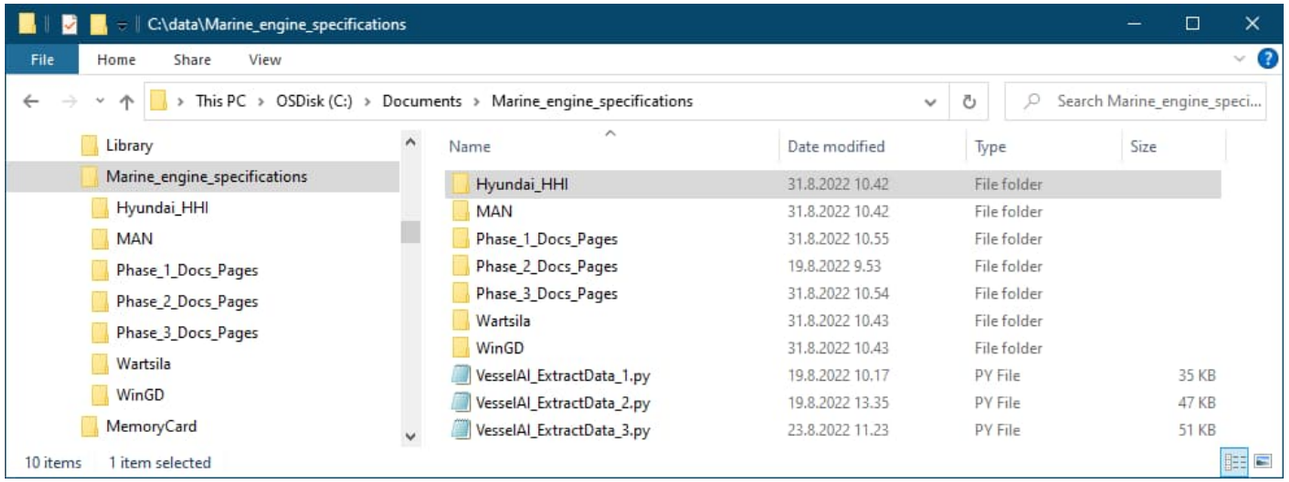
Neural network program

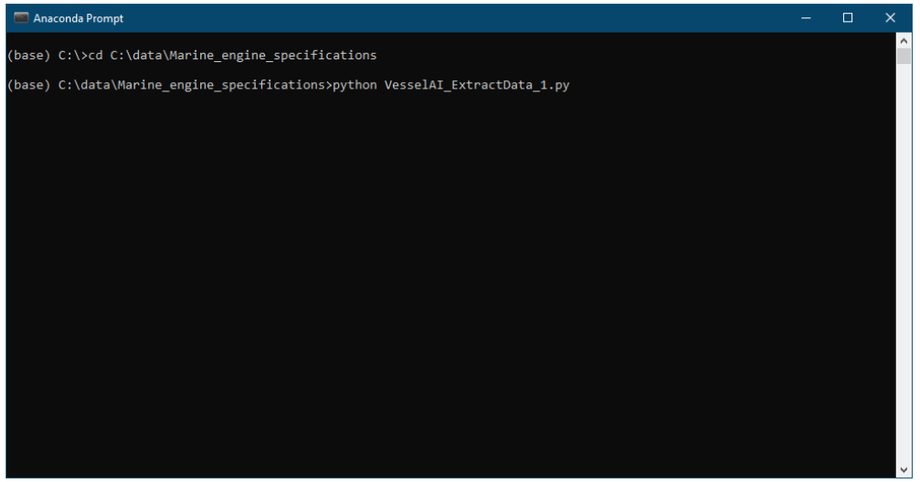
- Ground truth of relevant pages by human expert: 690 pages were deemed relevant
- NN program: many settings were tried, results of best one
  - Precision = 0.998, Recall = 0.979, Accuracy = 0.999
- NN produced much better results than either Phase 1 or Phase 3
- However, for new data (e.g. new marine engine manufacturer) both NN and Phase 3 are likely to fail
- But Phase 1 could still work: SI units (our keywords) and technical terms (our other words) are the same



Installation

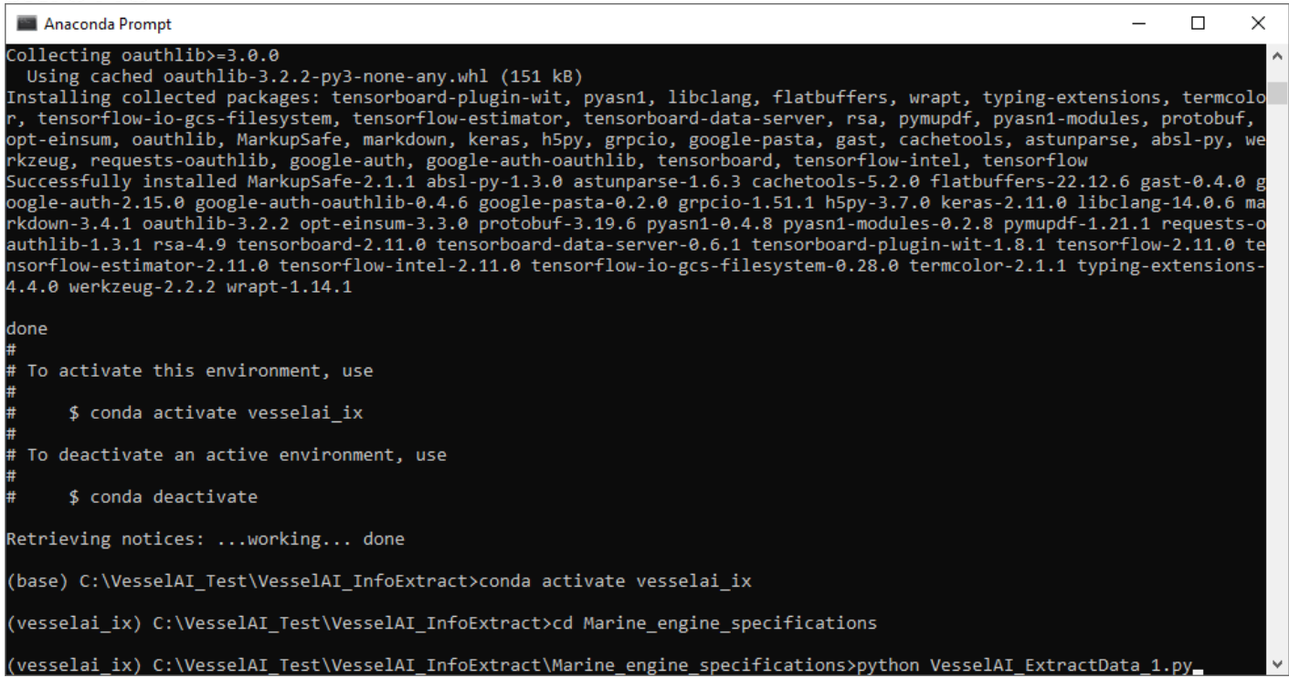
- The *.zip file contains a folder ‘Marine_engine_specifications’
- This folder contains 4 Python files:
  - VesselAI_ExtractData_1.py
  - VesselAI_ExtractData_2.py
  - VesselAI_ExtractData_3.py
  - VesselAI_ExtractData_NN.py
- …and also subfolders containing PDF files (Hyundai_HHI, MAN, Wartsila, WinGD) as well as empty subfolders where result data will be written (Phase_?_Docs_Pages)
- The PDF files must be downloaded from Internet, see slides 5-9





Installation, continued

- Extract the contents of the *.zip file into a local folder, and then go to the folder ‘VesselAI_InfoExtract’ (inside the new folder) with your Python Prompt
- In Anaconda Python environment, enter command:
  - conda env create -f env.yml
- This will create Anaconda environment ‘vesselai_ix’, containing the necessary Python program libraries; then activate this new environment:
  - conda activate vesselai_ix
- Go to subfolder ‘Marine_engine_specifications’
  - cd Marine_engine_specifications
- Remember that subfolders ‘MAN’, ‘Wartsila’, ‘Hyundai_HHI’, and ‘WinGD’ must contain PDF files you have downloaded, as instructed earlier
- Start the Phase 1 program, *_1.py
  - python Vesselai_ExtractData_1.py



Installation, continued

- If you do not have the Anaconda Python, this will probably fail at first, because you have not installed all the necessary libraries to your Python environment
- But when starting the python program fails, the name of the missing library is shown. Then install this library (pip install <name_of_lib>). Continue until *_1.py starts
- Close this *_1.py immediately. Then try *_2.py and *_3.py. Continue installing missing libraries until all 3 programs start
- Then try to go through the whole workflow with default values: start *_1.py:
  - >python VesselAI_ExtractData_1.py
  - *_Phase_1: Maximize, move vertical slider right, press ‘Refresh’, (wait…), press ‘Next >>’
  - *_Phase_2: Maximize, move vertical slider right, press ‘Precomp’, (wait…), select algorithm ‘DBSCAN’ from the drop-down, press ‘execute’, press ‘Next >>’
  - *_Phase_3: Maximize, move vertical slider right. Next you should enter relevant line definitions for each cluster, the press ‘Refresh’, (wait…), press ‘Results’.
  - => However, because this is tedious (DBSCAN produces 41 clusters!), delete all the files in folder ‘Phase_3_Docs_Pages’, and then start ‘VesselAI_ExtractData_3.py’ directly. And now:
  - *_Phase_3: Maximize, move vertical slider right, press ‘Refresh’, (wait…), press ‘Results’
  - And now Notepad shows file ‘Phase3_key_results.txt’, containing relevant data lines only!