

|  |
| --- |
| Coventry University  5011CEM Big Data Programming Project Specification Document  visualization and simple Ensemble comparison |

**Author: Armandas Barkauskas**

**Document Version: 2.0**

**21/03/2020**

**Table of Contents**

[1. Introduction 3](#_Toc35632211)

[2. Project Requirements 3](#_Toc35632212)

[2.1 Related documents 4](#_Toc35632213)

[2.2 Terms/Acronyms and Definitions 4](#_Toc35632214)

[3. Risks and Assumptions 5](#_Toc35632215)

[4. Out of Scope 5](#_Toc35632216)

[5. System/ Solution Overview 5](#_Toc35632217)

[5.1 Context Diagram/ Interface Diagram/ Data Flow Diagram, Application Screen Flow, Sitemap, Process Flow 6](#_Toc35632218)

[6. Project Management 7](#_Toc35632219)

[7. References 8](#_Toc35632220)

[8. Open Issues 8](#_Toc35632221)

[Appendices 8](#_Toc35632222)

# Introduction

The aim of this project is to help the scientists visualize huge amounts of data (Ozone/O3) over the area of the Europe. This can be done by allowing the scientist to compare the models for the same time period side by side. The data is provided in a grid of 700 by 400 for Ozone levels in 7 different models. Furthermore, in the words of Hyde, it was found that the accuracy of model ensembles can be improved up to 18% when using the DDC clustering algorithm (Hyde and Angelov 2014). More details about DDC algorithm can be found at <https://ieeexplore.ieee.org/document/6930157>. Consequently, there is a need to make a simple comparison between mean ensemble and cluster-based ensemble (CBE) to identify and obtain the errors made by the predictions. Project is expected to use some of the big data techniques such as data streaming from the disk, parallel processing and pretty form of visualization. Matlab is a great tool to perform mathematical calculations and plot various plots, which is going to be used to achieve the required goals by the customer.

# Project Requirements

Originally, there are 14 climate models worldwide, but for this project only 7 of them are going to be used due to the long execution time. The program is expected to support the data input provided in 2 formats: NetCDF (standard climate science format) or CSV (table like format). Moreover, we are going to analyse Europe area only (in 0.1-degree precision). It is required to use some big data techniques to reduce the processing time to less than 2 hours. These techniques will include optimized data streaming from the disk, parallel processing, display of the big data. The flowchart depicts the core idea of program flow and expected working principles. All project plans and completed tasks with comments are going to be recorded in the excel logbook which can be reviewed by the business at any time. Some of the PSEUDOCODE for program can be found in **Appendix A**. Also add an option to choose colour schema including blind support. Ultimately, this sub-project **is not** aimed to produce the CBE ensemble while using DDC algorithm. This step is expected to be done by the other team, and then CBE data to be passed to us to produce a visual comparison.

Suggested systems requirements:

|  |  |  |
| --- | --- | --- |
| **Item** | **Minimum** | **Recommended** |
| Operating system | Windows 7 | Windows 10 |
| Processor | Intel i3-2120 | Intel i5-4440 |
| RAM | 4 GB | 16 GB |
| Graphics | None | NVIDIA GTX 1050 Ti |
| Disk space | HDD drive with 8 GB | SSD drive with 32 GB |

## Related documents

|  |  |  |
| --- | --- | --- |
| **Component** | **Name (with link to the document)** | **Description** |
| Code base / Version control | <https://github.coventry.ac.uk/barkausa/big-data> | Program code for this project excluding the big data files |
| DDC report | <https://ieeexplore.ieee.org/document/6930157> | Find out more about the DDC algorithm from its authors |
| Flowchart | /flowchart.drawio and /images/flowchart.png | Find the program flowchart as a project file or image |
| Installation guide | /INSTALL.md | Contains a short description how to run the program |
| README file | /README.md | Contains a description on the written program such as file structure, source of big data files etc. |

## Terms/Acronyms and Definitions

|  |  |  |
| --- | --- | --- |
| **Term/Acronym** | **Definition** | **Description** |
| observations |  | A model that is an accurate measurement extracted by the real sensors. This data model is interpreted as 100% accurate. |
| SME / SE | Simple mean ensemble or simple ensemble | An average of all provided models (in our case 7) for every location point within Europe |
| DDC algorithm | Data density clustering algorithm | An algorithm that clusters data points. Its output is being used to create a CBE |
| CBE | Cluster based ensemble | More accurate ensemble then SME, where generation of it is much more complicated than SME |
| Sub-spacing |  | Dividing a current data set to smaller data chunks |
| Parallel processing |  | Usually, splitting independent tasks to run on a different thread to execute faster. Closely related to sub-spacing. |
| flowchart |  | Depicts the main logical program steps visualized using a special symbol syntax |
| logbook |  | An excel sheet, were the major project steps are being recorded |
| IDE | Integrated development environment | A software that provides facilities for the program developers. Usually it is a place where developers write code. |

# Risks and Assumptions

To begin with, a Matlab programming language is a great choice for a scientific research but it has a couple of drawbacks such as old IDE which might cause some inconveniences for developers and unclear documentation. In addition, Matlab is a paid software which requires a pay licence to be used. The visual output may not be suitable for small screens such as small laptops and mobile phones. The different program functions might use different window sizes. Also, this project is going to use Matlab with addons, which is brings to a concern that it takes big amount of disk space and processing power while installing.

The hardware will be limited up to 8 cores due to the financial constraint, meaning that the performance still can be scaled horizontally later. Data is encoded in uncommon data format: netCDM, which leads to the point that the developers’ community is smaller. Thus, as an alternative program is going to accept a CSV format as well. Furthermore, data files are huge in size and any operational error in the data file might cause data corruption. As a result, developers need to careful when handling the data. In addition, an error can occur in programming files leading to incorrect program behaviour. Some of the 3rd party libraries can be unstable. Ultimately, data files might change in the future and thus developers must write some tests to ensure that the program deals with all possible cases.

# Out of Scope

This section will discuss the things that are not going to be achieved by this project. To begin with, some of the visual projections are going to use CBE ensemble for comparison against a predicted mean model but this program is not intended to produce a CBE ensemble itself. Moreover, it is not expected to handle the data bigger than 1 GB size in this project due to the time limitations given for this project. Consequently, this might produce slightly imprecise results.

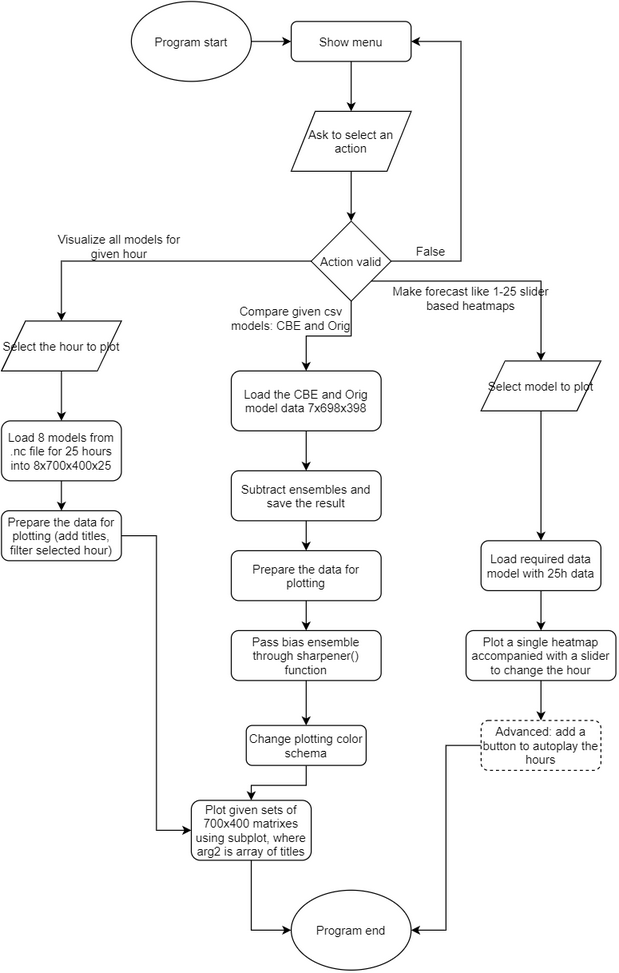
# System/ Solution Overview

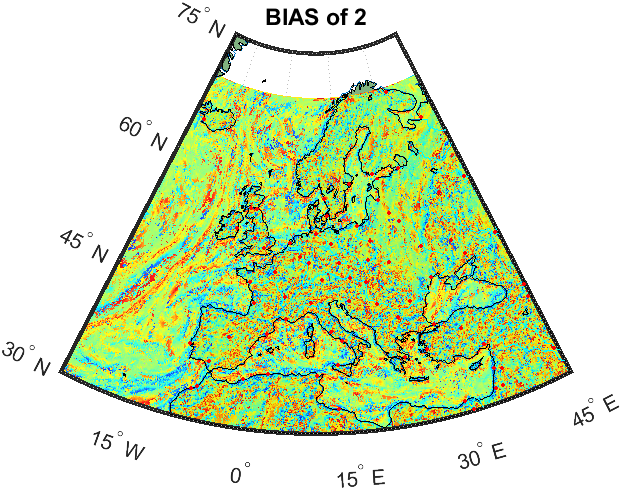
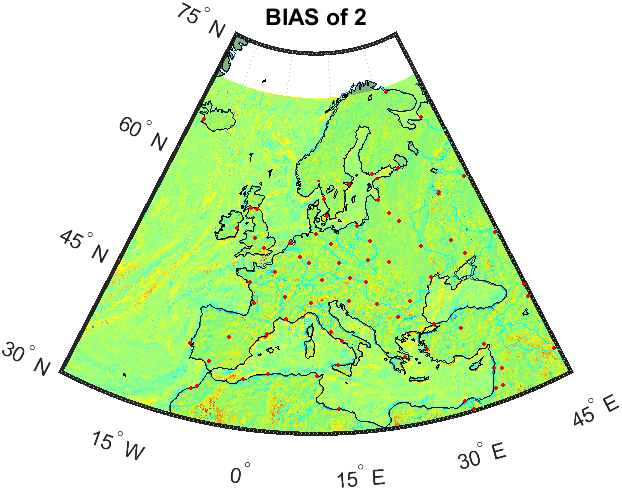
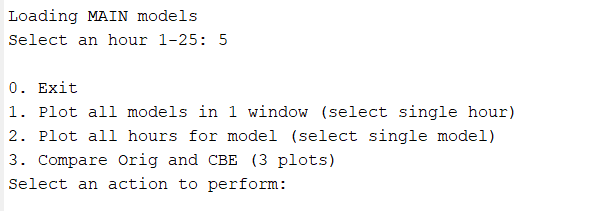
The software is going to be written as a Matlab script running in Matlab studio. Software will run in a console menu, initially displaying you the menu with possible actions. Let’s discuss the options.

1. Exit
2. Plot all 8 models in 1 window by selected hour
3. Plot model with a slider allowing to change selected hour
4. Compare given ensembles (Org & CBE)
5. Change a colour scheme for plots

When selecting any of the options, the program will save the loaded ensembles data in the memory so the user can use it again with a much faster access. Furthermore, option 3 will calculate the difference between the 2 models (Orig and CBE) and then run a *sharpener* algorithm to highlight the errors. The *sharpener* function will have an option to run a parallelized version of that function. All results are going to be plotted in a form of heatmap allowing the user to see the difference between model values.

# Context Diagram/ Interface Diagram/ Data Flow Diagram, Application Screen Flow, Sitemap, Process Flow

****

****

# Project Management

Main project managements strategies:

* **Logbook** as excel sheet
* Git **version control**
* Estimated program behaviour in **flowchart**
* Clear **documentation** in the project (README.md, INSTALL.md, code comments)
* Use of **SMART** framework

All completed tasks in this project will be recorded and timestamped in the logbook, which later can be reviewed by the business or other developers. The other component is a logical folder structure splitting the files into these folders: ***data***, ***images***, ***src*** (development code), ***dist*** (production code).

Project is going to be well documented including code comments and markdown files. Code will be regularly uploaded to the Github. Lastly, the core logic of the program is explained in the Data Flow Diagram.

**SMART** (acronym) targets are:

* Help the scientists to visualize and compare the data models in a convenient manner
* Track the current progress of the project in a single file (logbook)
* The model rendering/plotting should take less than 15 seconds
* The CBE models for comparison should already be precompiled by other program
* Must be realistic in terms of system resources such as CPU, RAM and memory
* Increment development includes delivery of new features by an agreed deadline
* The data exploration concerns single pollutant (in this case O3)

# References

* Matlab language documentation: <https://www.mathworks.com/help/matlab/>
* Data density-based clustering: <https://ieeexplore.ieee.org/document/6930157>
* University material/ data source (restricted access): <https://cumoodle.coventry.ac.uk/course/view.php?id=70387&sectionid=1011258>
* Code base website: <https://github.coventry.ac.uk/>
* Colour maps: <https://uk.mathworks.com/help/matlab/ref/colormap.html>
* SMART goals guide: <https://corporatefinanceinstitute.com/resources/knowledge/other/smart-goal/>

# Open Issues

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Issue** | **Raised On** | **Solution/ Decision** | **Resolved On** | **Status** |
| Specification must be improved | 19/02/2020 | Go over the text again and improve the quality. Add more references | 20/03/2020 | Done |
| Missing SMART targets | 19/02/2020 | Analyze the project and add the SMART targets as they might impact the velocity of the project | 21/03/2020 | Done |

# 

# Appendices

**APPENDIX A: pseudocode**

Pseudocode is included only for some of the algorithms that can also be found in project files.

* Sharpener
* Parsharpener (parallelized)
* Program (index file)

**SHARPENER**

TITLE: sharpener function pseudocode

DESCRIPTION: this algorithm takes model (matrix) and number 1-5 for strength and diminishes the values that are outstanding

FUNCTION sharpenEnsemble(ensemble, strength)

sz <- the dimentions of ensemble in array

average <- set initial average to 0

FOR set i: 1 to sz[1]

FOR set j: 1 to sz[2]

average <- add every value in the matrix

END

END

average <- average divided by the metrix cells count

FOR set i: 1 to sz[1]

FOR set j: 1 to sz[2]

value <- ensemble[i][j]

percent <- calculate value relation with average by: value \* 100 / average

IF absolute value of "percent" is more than 1000(%)

SET ensemble[i][j] to be equal itself divided by "strength"

END

END

END

RETURN ensemble

END

**PARSHARPENER**

TITLE: parallel sharpener function pseudocode

DESCRIPTION: this algorithm takes model (matrix) and number 1-5 for strength and diminishes the values that are outstanding in parallel

FUNCTION parSharpenEnsemble(ensemble, strength)

numProcessors <- SET to 6

IF check if we dont have parallel pool, THEN

start parallel pool passing "numProcessors"

END

sz <- the dimentions of ensemble represented as array

result <- create empty output matrix with the same dimentions as ensemble

PARFOR set i: 1 to sz[1]

FOR set j: 1 to sz[2]

# Average call is executed every time to make system stress and prove

# that parallel processing can be implemented in this algorithm

# in case it would be of a bigger data set

average <- SET function return value: average(ensemble)

value <- ensemble[i][j]

percent <- calculate value relation with average by: value \* 100 / average

IF absolute value of "percent" is more than 1000(%)

value <- SET to be equal ensemble[i][j] divided by "strength"

ELSE

value <- SET to be equal ensemble[i][j] (copy)

END

result[i][j] <- SET to be equal to "value"

END

END

RETURN ensemble

END

FUNCTION average(ensemble matrix)

sz <- the dimentions of ensemble represented as array

sum <- set initial average to 0

FOR set i: 1 to sz[1]

FOR set j: 1 to sz[2]

sum <- add every value in the matrix

END

END

RETURN the sum divided by the metrix cells count (n)

END

**PROGRAM**

TITLE: Main(index) project file

DESCRIPTION: This is a starting file to run the program. It handles menu logic and loads the files

PRINT: program started!

# CREATE model matrix variables globally

CBE <- NULL

ORIG <- NULL

MAIN <- NULL

choice <- -1

WHILE choice is not equal to 0

PRINT: menu options

choice <- get user input

SWITCH choice

CASE 0: exit the program by CONTINUE

CASE 1:

MAIN <- load main data file

pass that data to plot all models

CASE 2:

MAIN <- load main data file

pass that data to plot single model with slider to pick hour

CASE 3:

CBE <- load cluster based ensemble

IF CBE loaded successfully THEN

ORIG <- load ORIG ensemble

END

DEFAULT:

PRINT: unknown action error

END

END

FUNCTION loadMainDataFile

result <- -1

IF ORG is empty

IF file models-combined.nc EXIST

modelNames <- get model names

create result 4D matrix 7x700x400x25

read opened file and SET this to a variable MAIN

result <- 1

END

END

RETURN result

END

PRINT: program finished!