**Constraint Data Processing Workflow of Constraint Analyzer Project Transmission Outage Data Processing**

Transmission outages hourly data pulled **from Yes Energy (2014-2019)**

**Code file - Get\_Historical\_Trans\_Outage\_v1.py**

Constraint-Contingency Summary data pulled **from Yes Energy (2014-2019)**

**Manual process of pulling data from Yes Energy Portal**

Column values of outage data (all files) and mapping document are changed in order to make mapping easier

**Code file – TransmissionOutagesMapping.py**

Constraint-contingency unique pair extraction

**Code file – ConstraintContingencyUniquePairCreation.py**

Column values of constraint data and mapping document are changed in order to make mapping easier

**Code file – ConstraintContingencyMapping.py**

With the obtained excel sheet, vlookup is performed to map constraints to the respective from bus and to bus number

With the obtained excel sheet, vlookup is performed to map constraints to the from bus and to bus number respectively.

All the files are combined in such a way that only unique outages which are mapped are stored in one sheet and unmapped outages are stored in another of resultant file

**Code file – UniqueTransmissionOutagesConcat.py**

Separation of mapped constraints, mapped contingency, unmapped constraints, unmapped contingency and mapped both

**Code file – ConstraintContingencySeparation.py**

Mapped outages are verified using approximate string matching and a histogram is created to show verification results

**Code file – MappedOutagesVerification.py**

**Code file – MappedOutagesVerificationHistogram.py**

Above obtained sheet containing unmapped constraint data is used to perform approximate string matching between unmapped constraints and mapping document

**Code file – UnmappedConstraintsApproximateStringMatching.py**

Unmapped outage data is used to perform approximate string matching between unmapped outages and mapping document

**Code file – ApproximatematchingUnmappedTransmissionOutages.py**

Mapping of unmapped constraints is performed using approximate string-matching results

**Manual Process**

Mapping of unmapped constraints is performed using approximate string-matching results

**Manual Process**

A final file is created containing unique constraint-contingency pair which are mapped

**Code file – ConstraintContingencyFinal.py**

After the manual process unmapped and mapped outages are separated into different sheets of an excel file

**Code file – SeparationOfUnmappedTransmissionOutages.py**

Hourly constraint-contingency data is pulled from **Yes Energy (2014-2019)**

**Code file - Get\_Historical\_ConstraintData.py**

A final file (combining already mapped and newly mapped outages) consisting unique transmission outages data is created where mapped and unmapped outages are separated into different sheets

**Code file – TransmissionOutagesFinal.py**

A final file consisting of hourly constraint-contingency data is created which is mapped using the above mapping file.

Only mapped constraint-contingency pairs are stored in an excel sheet

**Code file – ConstraintContingencyList.py**

A super list consisting of hourly outage data (subset = outage and start date unique) is created which contains mapped outages of all years

**Code file – TransmissionOutagesList.py**

Above obtained excel file data is extracted and formed such a way that it matches the powerworld case file format

**Code file – ConstraintContingencyInterfaceCreation.py**

Result obtained from above step is used to define interface elements in powerworld case file which is stored after making changes to it.

**Code file – ConstraintContingencyInterfaceDefinitionPowerworld.py**

**Note – detailed description of all code files can be found in CodeFiles\_Documentation.docx**

Result of above step is used to create the training dataset

**Code file – TraningDatasetNewMethod.py**

Only mapped constraint-contingency pairs are stored in an excel sheet

**Code file – ConstraintContingencyList.py**

A super list consisting of hourly outage data (subset = outage and start date unique) is created which contains mapped outages of all years

**Code file – TransmissionOutagesList.py**

Dependent LODF calculation is performed taking constraint-contingency and transmission outage files.

**Code file – CalculationNewMethod.py**

Independent LODF calculation is performed taking constraint-contingency and transmission outage files.

**Code file – CalculationTrial.py**

Result of above step is used to create the training dataset

**Code file – TraningDatasetTrial.py**

Only mapped constraint-contingency pairs are stored in an excel sheet

**Code file – ConstraintContingencyList.py**

A super list consisting of hourly outage data (subset = outage and start date unique) is created which contains mapped outages of all years

**Code file – TransmissionOutagesList.py**