WMRAT USER MANUAL

Overview 2

Development 2

Usage 3

1..Model integration 3

2..Pipe failures.......... 4

2.1. Single pipe failure (EPANET) 4

2.2. Single pipe failure (graph-based) 7

2.3. Multi pipe failure (graph-based) 9

3. Single pipe leakages 12

4. Segment criticality.......... 17

5. Valve criticality 19

6. Downloading files 22

Netowork configurations.......... 23

License 25

Overview

WMRAT (Water Management Resilience Analysis Toolkit) is a proof-of-concept web application that showcases and implements several scenarios in the context of urban water management resilience analysis. It is part of the project RESIST.

Development

This section explains how to start the local development server in a virtual Python environment ("venv"). Make sure that you also have Redis installed and running (`systemctl start redis`). The following third-party dependencies are used and are installed into the virtual environment:

- Django (v5.0)
- Django RQ (v1.15.1)
- WNTR (v1.1.0)

First check out this repository and switch into it:

```
$ git clone git@github.com:iut-ibk/wmrat.git
$ cd wmrat
```

Make the environment:

```
$ python -m venv venv
```

Activate it (observe how your shell prompt changes):

```
$ source venv/bin/activate
```

Install dependencies (we will use requirements later):

```
$ pip install django django_rq wntr
```

If everything went well you have now successfully installed the (Django) web application in your virtual environment. There is a script to reset the database (NOTE: this deletes all files, but is useful for debugging):

```
$ ./ops/reset_dev_env.sh
```

The login credentials for the test admin are `admin` and `pass`. Then use the following command to run the development environment:

```
$ ./ops/run_dev_env.sh
```

This starts the Django development server on `localhost:8000` (the Django default port) and runs the required workers (that run the actual analyses).

Usage

This section explains the usage (Input functions, sample results (software based), sample results (GIS based)) of each criticality analysis with an artificially created water distribution network (WDN). The results/output is explained by an artificially created WDN in an Austrian Alpine city.

1. Model integration

Upload the required EPANET (.inp) file into the software with user specified coordinates for street view overlay.

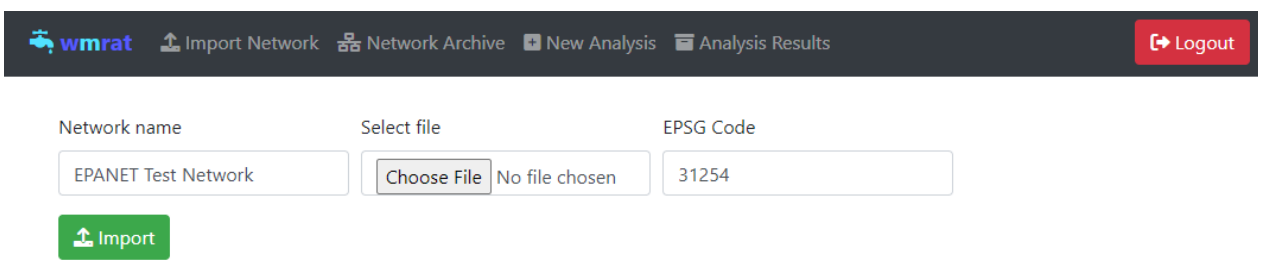


Figure 1: EPANET. inp File upload in WMRAT

2. Pipe failures

Single pipe failure analysis uses hydraulic-based criticality to assess node pressure, graph-based methods to identify overloaded pipes, and a hybrid approach for multiple failures to efficiently rank and analyze critical pipe combinations.

2.1. Single pipe failure (EPANET)

Allows users to estimate the number of critical nodes (nodes below the required pressure) during a single pipe failure using hydraulic-based demand driven EPANET criticality analysis. Results of pipes with no critical nodes are excluded from the results.

2.1.1 Input

The users are required to input the required values for duration of EPANET analysis, required pressure for evaluation and minimum diameter below which are excluded in the analysis as seen in Table 1 and Figure 2.

Name	Units	Purpose	Default	Provider
Duration	Seconds	Assess impact duration	0 seconds	User
Required Pressure	Meters	Critical nodes are estimated below required pressure	35.0 meters	User
Minimum Diameter	Meters	Excludes diameters below this threshold	0.001 meters	User

Table 1: Input value details for single pipe failure (EPANET)

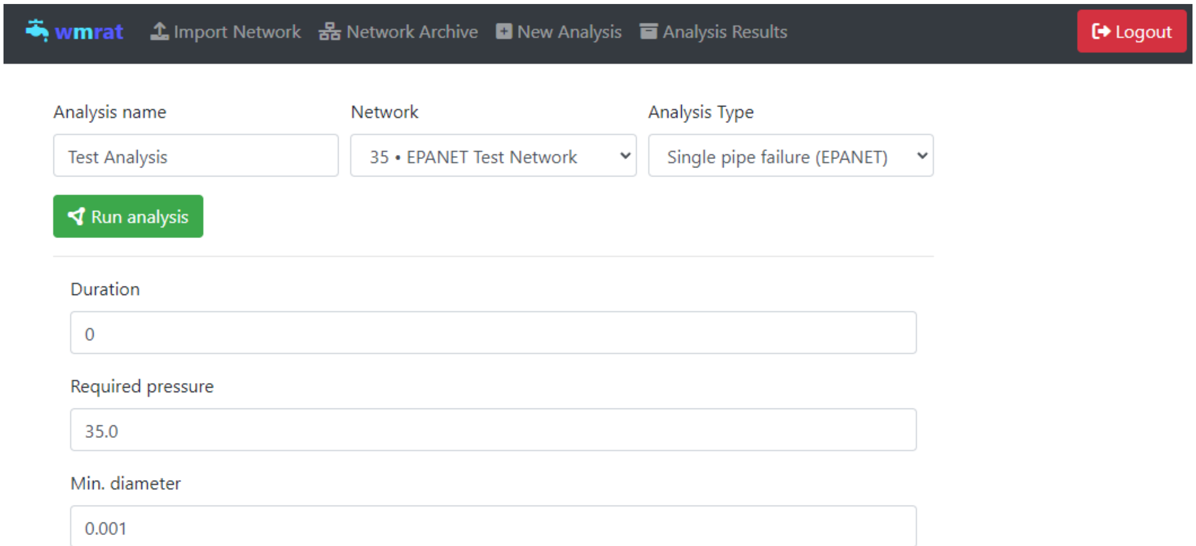


Figure 2: User input for single pipe failure (EPANET) on the web-platform

2.1.2 Sample results (Software based)

The software-based results display the failed pipe in orange and the number of impacted junctions in red as illustrated in Figure 3. Users can either select pipes directly from the web platform table or click on them in the visual map.

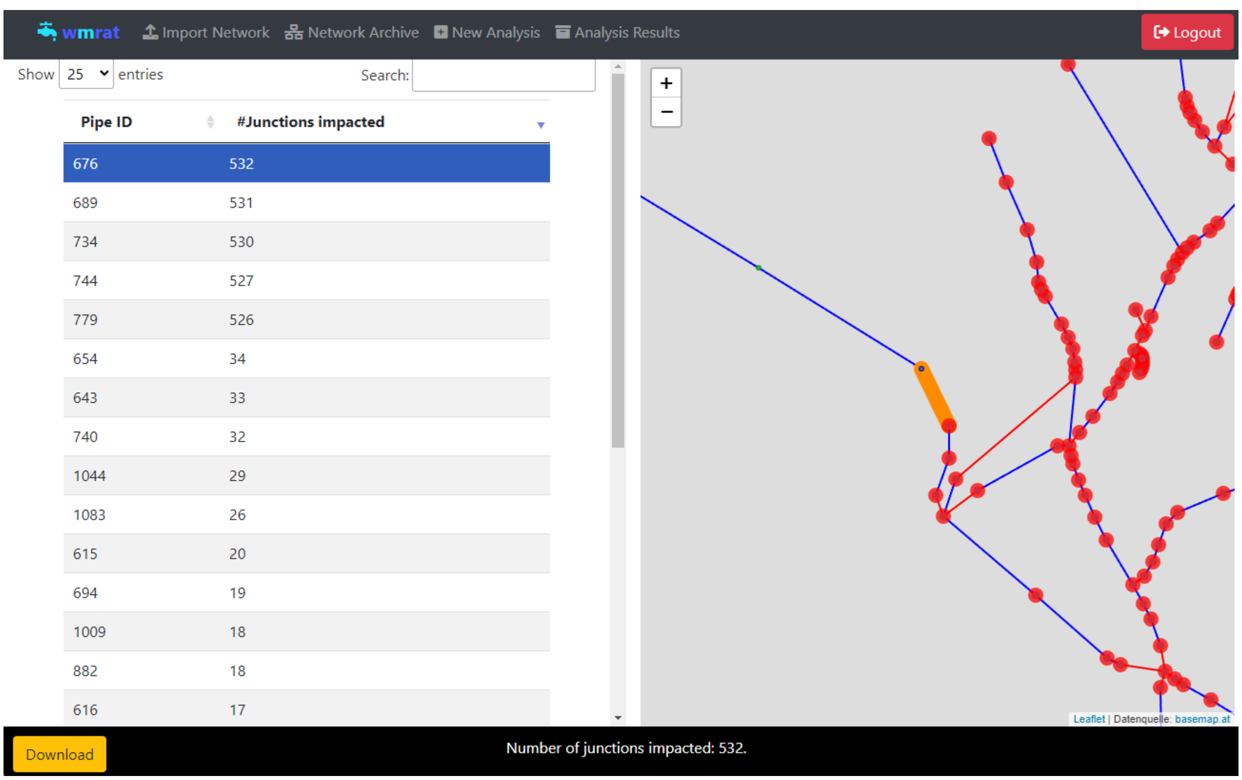


Figure 3: Sample results (software based) screenshot for the sample case study - single pipe (EPANET)

2.1.3 Sample results (GIS based)

Downloading the .json file and exporting it to GIS can facilitate comprehensive visualization of the WDN and the critical pipelines as illustrated in Figure 4.

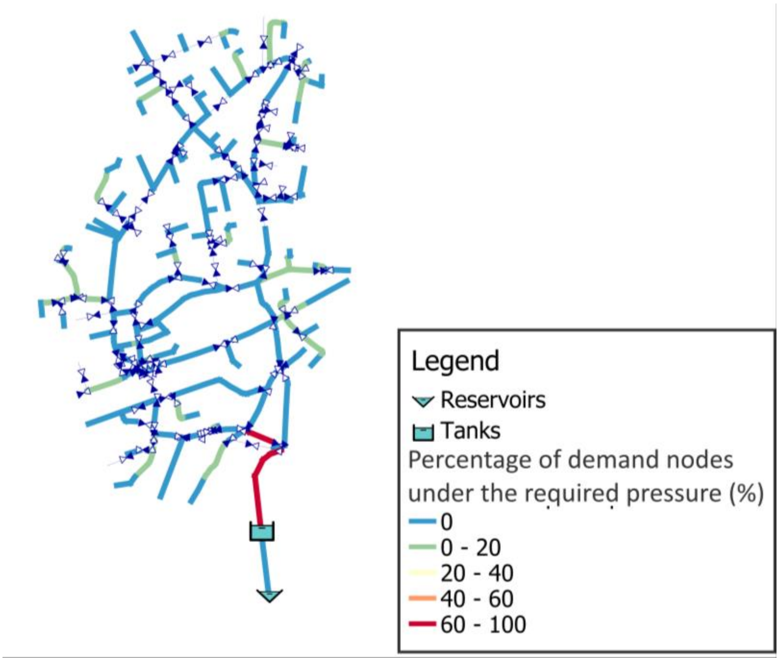


Figure 4: GIS visualization of the single pipe failure (EPANET) for the sample case study.

2.2. Single pipe failure (graph-based)

Allows users to estimate the rank of each pipe during a single pipe failure using graph-based criticality analysis.

2.2.1. Input

The users are required to input the Tank name as per the .inp file as illustrated Table 2 and Figure 5. The number of tanks is user defined by pressing on Add as illustrated in Figure 5.

Name	Purpose	Default	Provider
Source	Calculate the shortest route and failure magnitudes to this node	534	User
Number of sources	For networks with multiple sources	1	User

Table 2: Input value details for single pipe failure (graph-based)

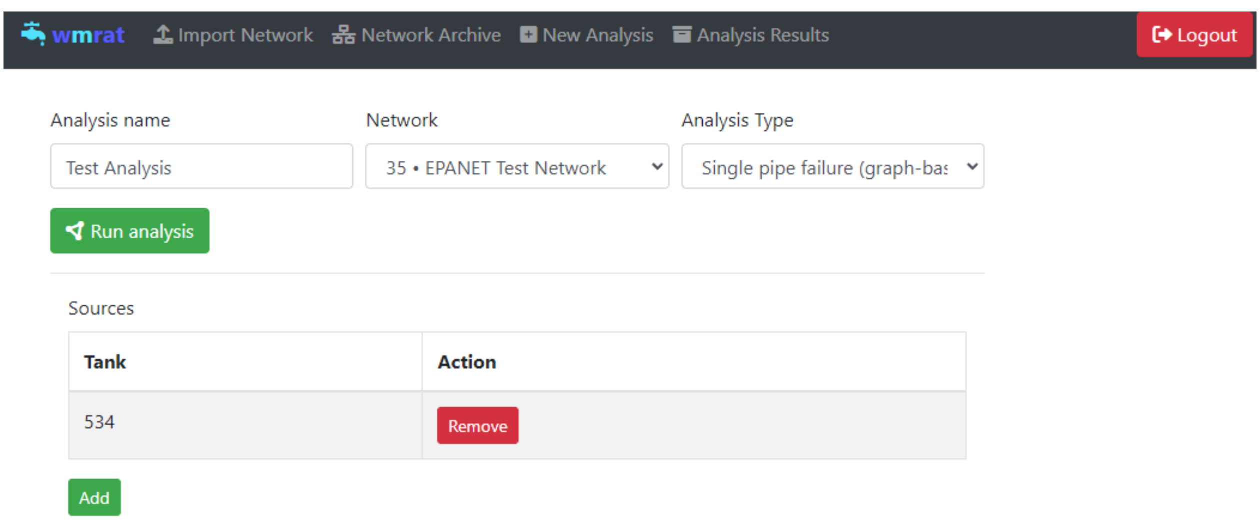


Figure 5: User input for single pipe failure (graph-based) on the web-platform

2.2.2. Sample results (Software based)

The software-based results display the failed pipe in orange and the rank of failed pipe based on the failure magnitudes in Figure 6.

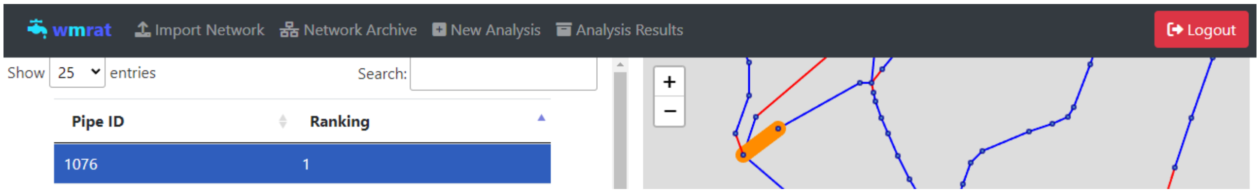


Figure 6: Sample results (software based) screenshot for the sample case study – single pipe failure (graph-based)

2.2.3. Sample results (GIS based)

Downloading the .json file and exporting it to GIS facilitates a detailed visualization of the WDN, highlighting pipes based on their rankings derived from graph-based failure magnitudes (GBFM), as illustrated in Figure 7.

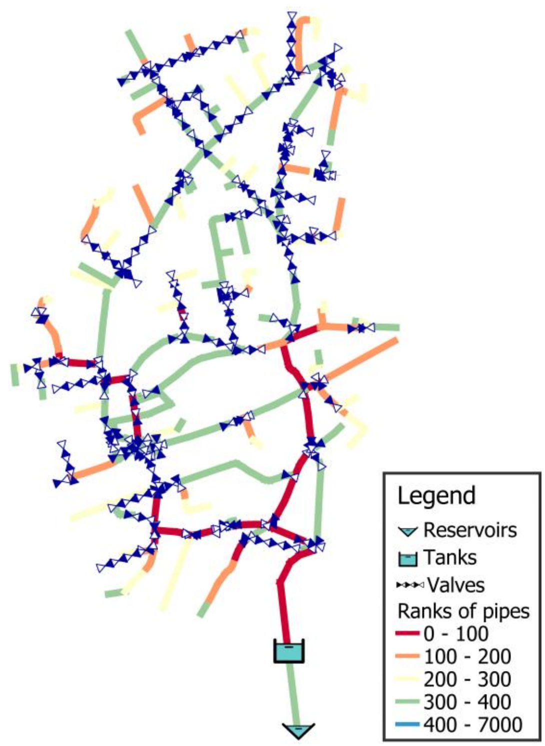


Figure 7: GIS visualization of single pipe failure (graph) for the sample case study.

2.3. Multi pipe failure (hybrid-based)

Allows users to estimate the rank on each pipe combinations during the respective pipe failure using graph-based criticality analysis and also rank and compute the amount of demand fulfilled in the network for the combinations based on hydraulic-based pressure driven analysis. The graph-based criticality analysis effectively filters out the combinations with no impact on the WDN making it faster than hydraulic-based criticality analysis.

2.3.1. Input

The users are required to input the Tank name as per the .inp file as illustrated in Table 3 and Figure 8. The number of combinations of pipes that is required to be analyzed as per user requirements.

Name	Purpose	Default	Provider
Source	Calculate the shortest route and failure magnitudes to this node	534	User
Number of combinations	To create the required number of pipe combinations	2	User

Table 3: Input value details for multi pipe failure (graph-based)

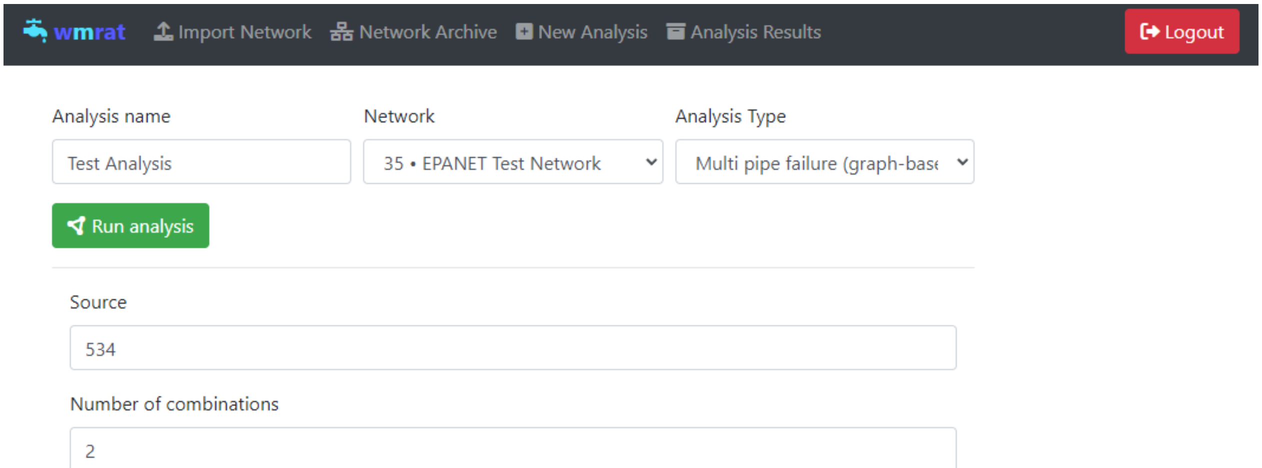
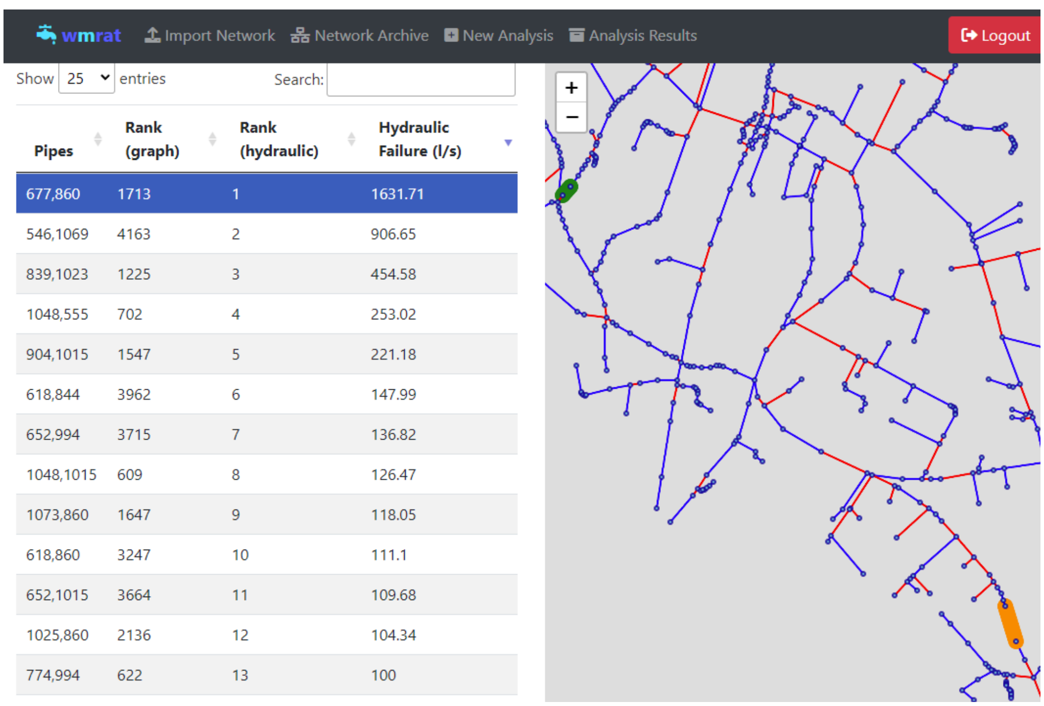


Figure 8: User input for multi pipe failure (graph-based) on the web-platform

2.3.2. Sample results (Software based)

The software-based results display the failed pipes in orange and green as illustrated in Figure 9 with ranks of graph and hydraulic-based criticality analysis and demand fulfilled from hydraulic-based criticality analysis in the software table.



Pipes	Rank (graph)	Rank (hydraulic)	Hydraulic Failure (l/s)
677,860	1713	1	1631.71
546,1069	4163	2	906.65
839,1023	1225	3	454.58
1048,555	702	4	253.02
904,1015	1547	5	221.18
618,844	3962	6	147.99
652,994	3715	7	136.82
1048,1015	609	8	126.47
1073,860	1647	9	118.05
618,860	3247	10	111.1
652,1015	3664	11	109.68
1025,860	2136	12	104.34
774,994	622	13	100

Figure 9: Sample results (software based) screenshot for the sample case study – multi pipe failure (graph-based)

2.3.3. Sample results (GIS based)

The results for GIS based visualization is complicated for the entire network as the results are based on combinations and not on a single source from the attributes table.

3. Single pipe leakages

Allows users to estimate the pipe leakages in seconds, minutes and time to empty 10 m^3 of water from the tank due leaks in single pipes.

3.1. Input

The users are required to input the Tank name, tank outflow pipe ID (pipe in direction of flow) as per the .inp file, number of tanks and material/area/exponent values as illustrated in Table 4 and Figure 10. The material/area/exponent values are literature-based values as illustrated in Table 5 and Table 6.

Name	Unit	Purpose	Default	Provider
Tank name	String	Calculate the volume of tank	TANK_1	User
Tank outflow pipe ID	String	To access the main outflow pipe	1086_b	User
Number of tanks	Number	Calculate volume and leakage affect for multi sources	1	User
Material/Area /Exponent	Varied	Calculation of emitter coefficient and emitter exponent	Figure 10	User

Table 4: Input value details for single pipe leakage

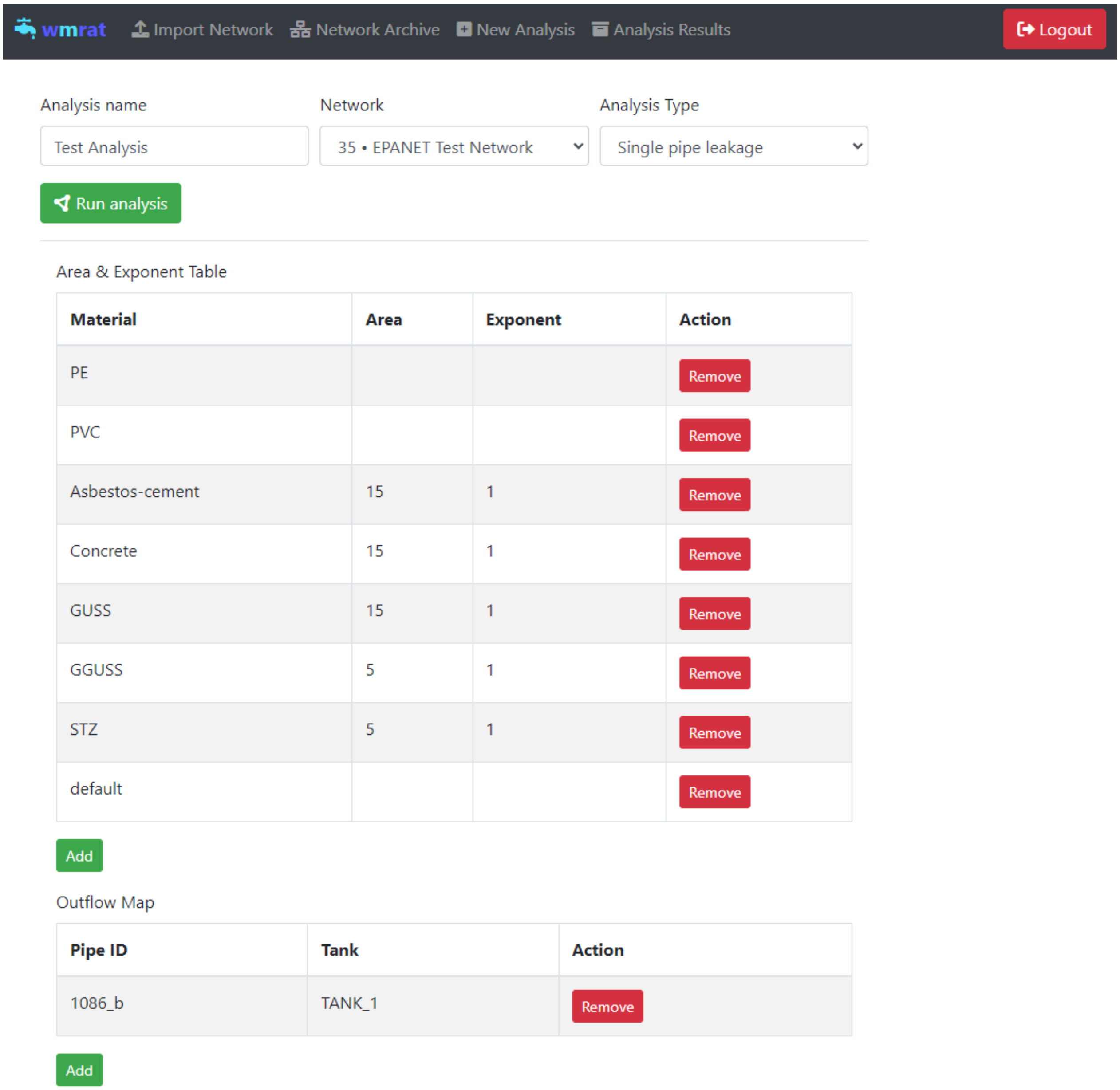


Figure 10: User input for single pipe leakages on the web-platform

Table 5: Aperture area A (cm²) for different fault modes for an Austrian transmission grid (modified from Friedl, et al. 2012)

Aperture area							
Failure mode	PE	PVC	Asbestos -cement	Concrete	GUSS	GGUSS	STZ
Circumferential crack	10	10	25	-	25	-	-
Longitudinal crack	40	40	40	-	40	-	-
Corrosion cluster	-	-	-	15	15	15	50
Leaking joint	-	-	15	15	15	5	5

Table 6: Emitter exponents derived from the literature for different failure modes (modified from Friedl et al., 2012)

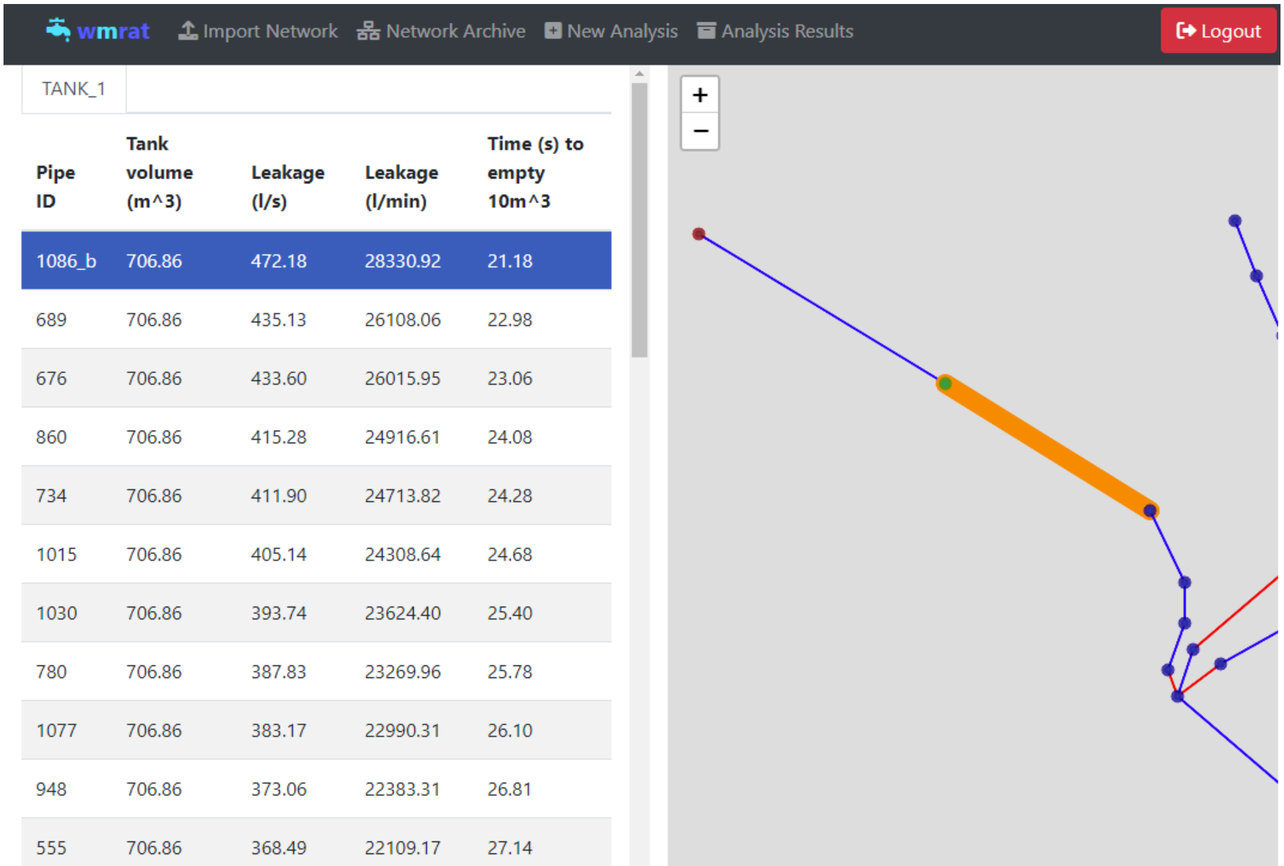
Emitter exponent							
Failure mode	PE	PVC	Asbestos -cement	Concrete	GUSS	GGUSS	STZ
Circumferential crack	0.5	0.5	0.5	-	0.5	-	-
Longitudinal crack	1.5	1.5	0.9	-	0.85	-	-
Corrosion cluster	-	-	-	1.5	1.5	1.5	1.5
Leaking joint	-	-	1	1	1	1	1

References:

Friedl, F., Möderl, M., Rauch, W., Liu, Q., Schrotter, S., Fuchs-Hanusch, D., 2012. Failure propagation for large-diameter transmission water mains using dynamic failure risk index, World environmental and water resources Congress 2012: Crossing boundaries, pp. 3082-3095.

3.2. Sample results (Software based)

The software displays the failed pipe in orange, along with the tank volume to which the pipe is connected. The leakage rates are shown in liters per second and per minute, as well as the time required to empty 10 m³ of water from the connected tank, as illustrated in Figure 11.



Pipe ID	Tank volume (m^3)	Leakage (l/s)	Leakage (l/min)	Time (s) to empty 10m^3
1086_b	706.86	472.18	28330.92	21.18
689	706.86	435.13	26108.06	22.98
676	706.86	433.60	26015.95	23.06
860	706.86	415.28	24916.61	24.08
734	706.86	411.90	24713.82	24.28
1015	706.86	405.14	24308.64	24.68
1030	706.86	393.74	23624.40	25.40
780	706.86	387.83	23269.96	25.78
1077	706.86	383.17	22990.31	26.10
948	706.86	373.06	22383.31	26.81
555	706.86	368.49	22109.17	27.14

Figure 11: Sample results (software based) screenshot for the sample case study – single pipe leakage.

3.3. Sample results (GIS based)

Downloading the .json file and exporting it to GIS facilitates a comprehensive visualization of the WDN and identifies critical pipelines that may be vulnerable to various types of leaks, as depicted in Figure 12.

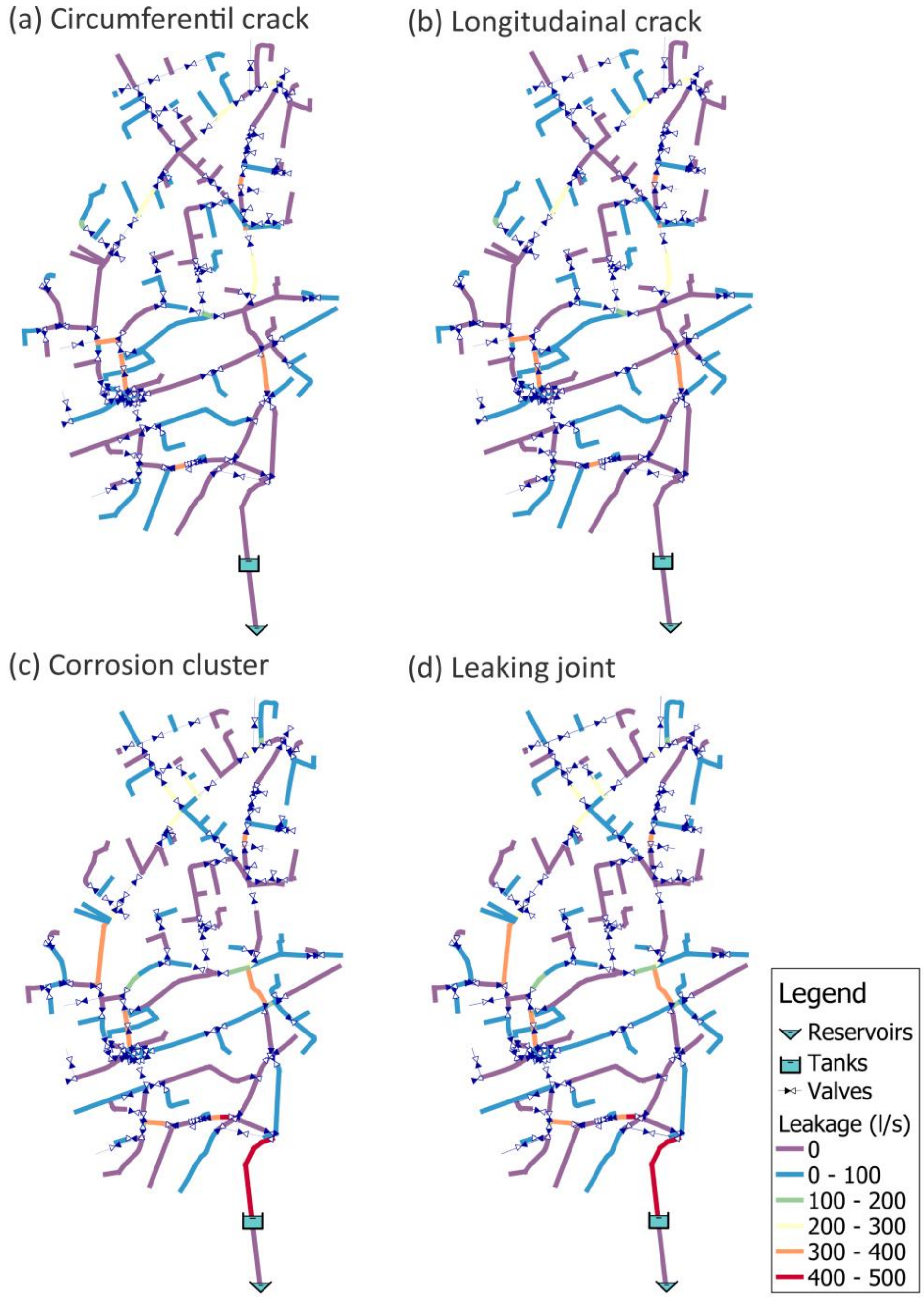


Figure 12: Results for single pipe failure (EPANET) for artificially created WDN visualized on a GIS platform.

4. Segment criticality

Allows users to estimate the criticality of each segment based of pressure driven hydraulic EPANET analysis by estimating the amount demand fulfilled during a segment failure.

4.1. Input

There is no user input required; however, the WDN must have isolation values (GPV or FCV) to function properly. Otherwise, the entire WDN will be considered as a single segment.

Name	Purpose	Provider
Isolation valves	Creation of segments	🖥 System

Table 7: Input value details for single pipe leakage

3.2. Sample results (Software based)

The software displays the segment in green, directly impacted nodes in red (inside the segment), indirectly impacted nodes due to lack of supply of water demand compared to the original state. The software table also displays the demand not fulfilled, number of directly and indirectly impacted nodes as illustrated in Figure 13.

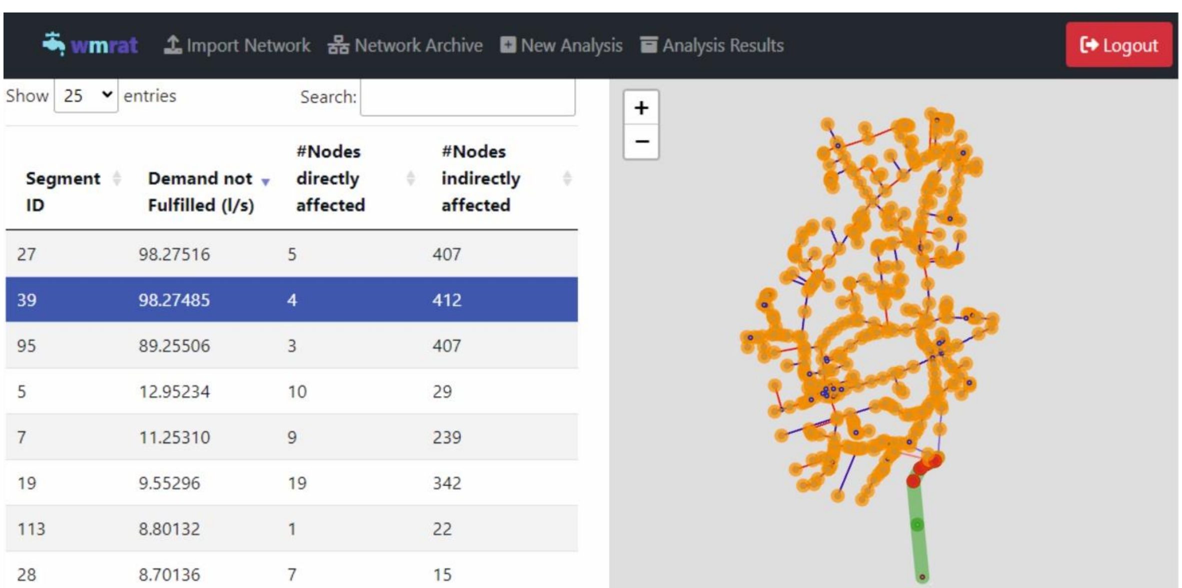


Figure 13: Sample results (software based) screenshot for the sample case study – segment criticality.

4.3. Sample results (GIS based)

The results for GIS based visualization is complicated for the entire network as the results are based on combinations (segments are combinations of pipes, nodes and valves) and not on a single source from the attributes table.

5. Valve criticality

Allows users to estimate the criticality of each valve based of pressure driven hydraulic EPANET analysis by estimating the amount demand fulfilled during a valve failure.

5.1. Input

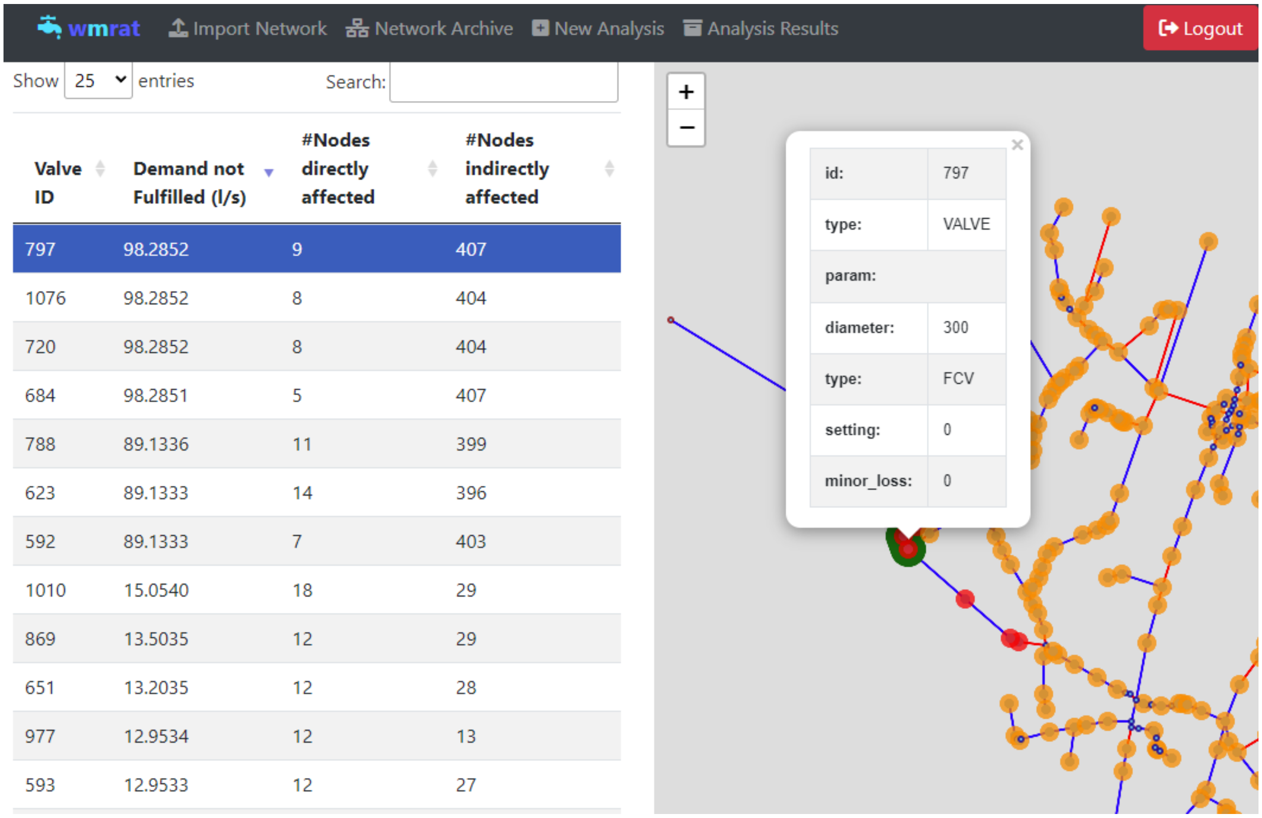
There is no user input required; however, the WDN must have isolation values (GPV or FCV) to function properly. During analysis segments connected to the valves are closed for finding the demand not fulfilled in nodes. Closing the segments would isolate the valve for repairs.

Name	Purpose	Provider
Isolation valves	Creation of segments	System

Table 8: Input value details for single pipe leakage

5.2. Sample results (Software based)

The software displays the failed valve in green, directly impacted nodes in red (inside the two segments closed), indirectly impacted nodes due to lack of supply of water demand compared to the original state. The software table also displays the demand not fulfilled, number of directly and indirectly impacted nodes as illustrated in Figure 14.



Valve ID	Demand not Fulfilled (l/s)	#Nodes directly affected	#Nodes indirectly affected
797	98.2852	9	407
1076	98.2852	8	404
720	98.2852	8	404
684	98.2851	5	407
788	89.1336	11	399
623	89.1333	14	396
592	89.1333	7	403
1010	15.0540	18	29
869	13.5035	12	29
651	13.2035	12	28
977	12.9534	12	13
593	12.9533	12	27

Figure 14: Sample results (software based) screenshot for the sample case study – valve criticality.

5.3. Sample results (GIS based)

Downloading the .json file and exporting it to GIS enables a detailed visualization of the WDN and highlights critical valves based on the percentage of demand that is not met under current conditions, as illustrated in Figure 15. The percentages of unmet demand are represented on a color gradient, with blue indicating no impact and red signifying a significant percentage of unmet demand.

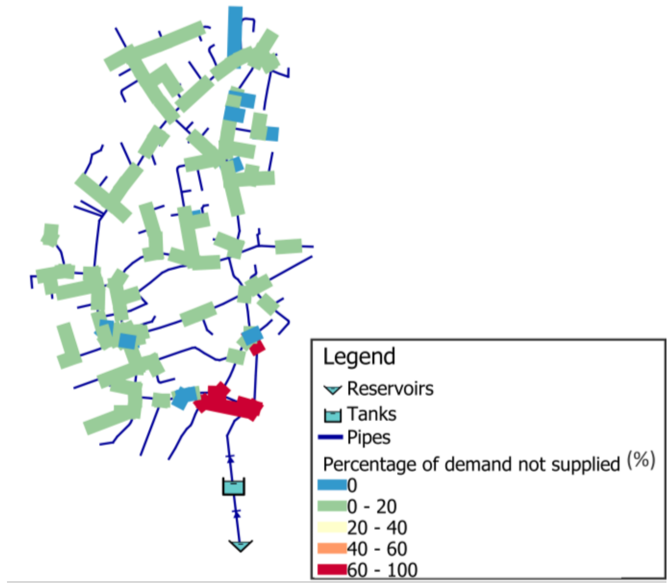


Figure 15: Results for valve criticality for artificially created WDN visualized on a GIS platform.

6. Downloading files

The results can be viewed on the web platform by clicking the "Show" button, and downloaded as an Excel or JSON file by clicking the "Download" button, as shown in Figure 16.

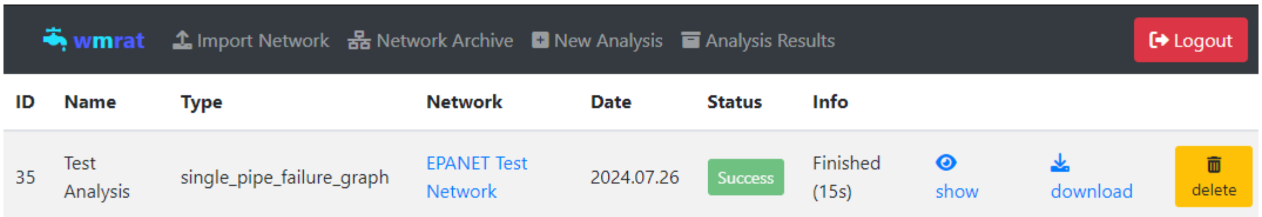


Figure 16: Analysis results output tab in WMRAT

Network configurations

Certain analyses require specific configurations to ensure the proper functioning of the WMRAT software. This section outlines both general configurations and Analysis component specific configurations.

General configuration:

- WMRAT operates on single-step analysis, meaning the demand is always considered at the first second of the demand pattern.
- Negative pressure nodes may not significantly impact scenarios, as the software primarily utilizes PDA, and DDA is employed to identify pressures lower than or equal to the specified threshold.
- The software is designed with minimal control settings (e.g., tank and pump settings). Introducing additional control settings may result in runtime errors.

Analysis component specific configuration:

Single pipe failure (graph-based)

- Failure to specify the correct tank or source node may result in an error.
- The network must contain at least one source node.
- Control settings within the network may interfere with the execution process.

Multi pipe failure (hybrid-based)

- Suitable for single-source networks.
- Execution time varies significantly based on the number of pipes.
- Control settings within the network may interfere with the execution process.

Single pipe leakages

- Material data is required to obtain leakage values; otherwise, the software will produce null results.

- Martials data have to defied according to the software readable nomenclature.

Software readable nomenclature	Real nomenclature pipes
PV	Polyethylene
PVC	Polyvinyl chloride
Asbestos-cement	Asbestos-cement
Concrete	Concrete
GUSS	Ductile iron
GGGUSS	Caste iron
STZ	Steel pipes

- A tank must be present in the network, as the volume is calculated based on the tank's dimensions.
- The analysis is conducted for singe time step. To conduct analysis for peak demands the demand patterns may have to be adjusted

Segment criticality

- Networks without valves fail to produce results and give a error sign

Valve criticality

- Networks without valves fail to produce results and give a error sign

License

AGPLv3