Semi-Autonomous Operation (SAO) in Ultrafiltration Membrane Process

Technical Information

by:

Mahdi Faramarzi

Kaushik Ghosh

Gokula Krishnan Sivaprakasam

Bijuraj Pandiyath

Sathik Bhasha

SGDC-ISRD, CIC-AI LAB

April - 2022

For Internal Use Only

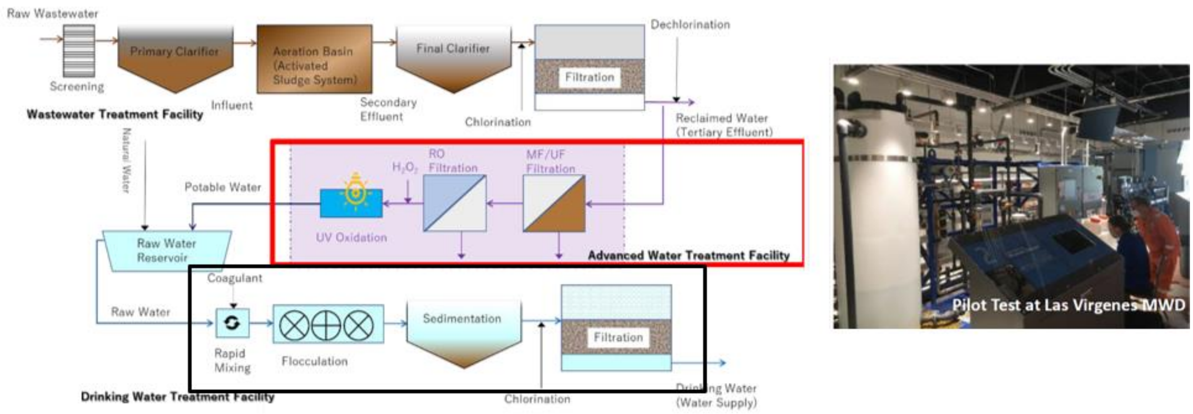
Table of Content

For Internal Use Only

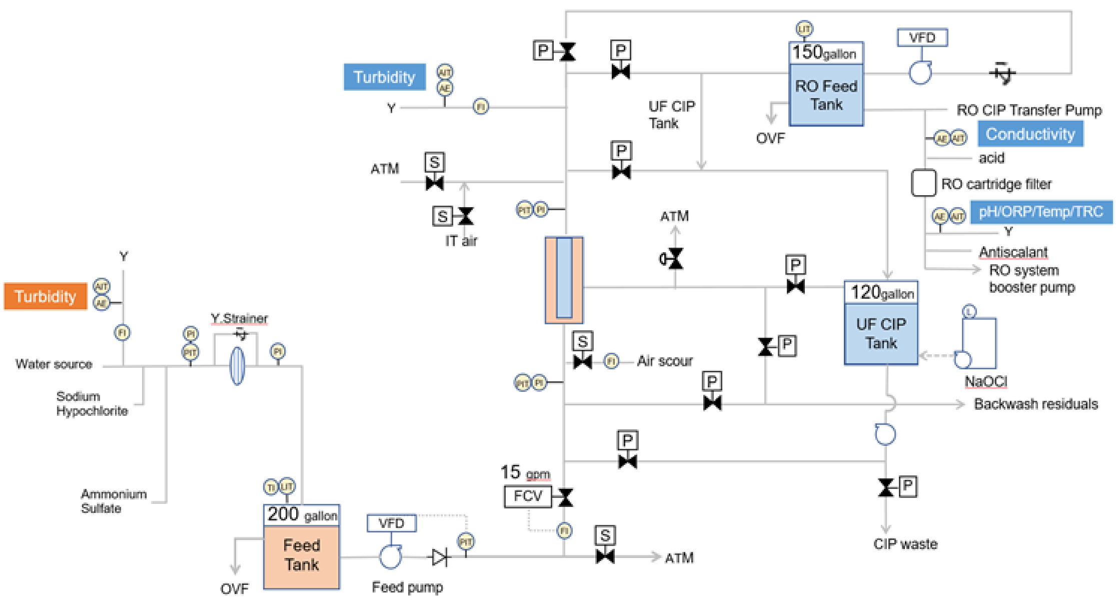
List of figures

1. Overall Process of Water Treatment

The following images show the overall water treatment plant (a) and its control system (b).



(a)



(b)

Figure 1. (a) overall water treatment plant and (b) plant control system

Raw water flows through different processes including Ultrafiltration (UF) and Reverse Osmosis (RO). This project focuses on UF membrane and its control system. Ultrafiltration membrane filtration (UF) is a low-pressure membrane process for water treatment that is designed to remove turbidity causing particles including those comprised of suspended solids, bacteria, colloidal matter and proteins. UF effectively removes particles in the size range of less than 0.01 to 0.1 mm. Effective removal of these contaminants in a source water results in a filtrate well suited for further treatment by downstream RO. During the ultrafiltration process the permeability of the ultrafiltration membrane diminishes gradually due to deposition of cake on the membrane surface and clogging of the membrane pores. Hence, more and more driving force (energy) is required to produce the filtered water. To avoid this situation, ultrafiltration membrane is cleaned frequently to improve the permeability of membrane by removing the clogging and deposited cake. It is therefore important to optimize the filtration-cleaning cycle in ultrafiltration process so that the required amount of permeate is produced at minimum overall operating costs. However, Optimizing the membrane ultrafiltration process poses novel challenges with respect to its modelling and simulation as it is very difficult to derive traditional first principle-based model for very complicated membrane fouling and cleaning behaviour.

To overcome these difficulties data driven AI based models are developed leveraging on large volume of operational data of the ultrafiltration process collected by Process Control System (DCS or SCADA) without building a first principles process model. State-of-the art AI based ML models are built to reveal the hidden relationship among the process operating parameters. The derived ML models are used in optimization to minimize the overall operating costs – based on chemicals consumption in cleaning, energy consumption in filtration and investment costs – over a fixed time horizon, guaranteeing production of a specified volume of permeate, where the number of cycles, the net production flux, the duration of a filtration phase and the duration of a subsequent cleaning phase are calculated.

2. SAO Objectives

The main objectives of Semi-Autonomous Operation (SAO) project for Ultrafiltration membranes are to optimize the performance of Ultrafiltration membrane, decrease the cost of operation and extend the life of membrane while satisfying the production demand. In this regards the SAO project identifies the optimum maintenance cleaning (MC), recovery cleaning (RC) frequency, Chemical concentration, cost of production per unit of water produced, flux and the estimated remaining useful life of the UF membrane using water quality and control system parameters.

The goal of the first phase of the project is to collect historical data from a popular UF membrane module based upon designed experimental plan and find the best flux set point that can maintain the permeability level, given the MC interval and chemical concentration, taking various water quality parameters (turbidity, viscosity, Total Organic Carbon) into consideration.

3. Unique challenges of SAO

Companies manufacture Ultrafiltration membranes in a variety of materials for many applications. Ultrafiltration membrane modules come in plate-and-frame, spiral-wound, and tubular configurations. In all configurations the optimum system design must take into consideration the flow velocity, pressure drop, power consumption, membrane fouling and module cost. However, membrane factors made it difficult to have an optimum setup for all configurations. Therefore, a lot of efforts have been made to optimize the control system for each UF membrane including different chemical cleaning approaches (Maintenance and Recovery), frequency of chemical cleaning, frequency and duration of backwash and air scouring, different flux set points for different production demands, etc. These different options give the operator flexibility and make their operation non-optimal without support of optimization algorithm. Usually, these factors are set up manually by the operator using experts' knowledge. Typically, MC is carried out with Hypo of 500 ppm concentration and RC is carried out with Hypo and Citric acid at 2000 PPM in sequence. However, SAO tries to make this process easier by using historical data and expert knowledge to build AI/ML system to help operator for best decision.

There are lots of parameters from water quality and chemical consumption to UF control system parameters which can be used for UF operation optimization process. Identifying the most relevant parameters is one of the main challenges in this process.

Another challenge of the UF membrane is real time data transfer and analysis. To overcome this issue, we proposed transferring data only from the selected tags that are required for modelling from the on-premises Process Control System (DCS or SCADA) located at UF membrane site to cloud.

Nature of process data also possess additional challenges in UF modelling and optimization as the collected historical data is very noisy and it is also including sensor reading errors, fluctuation in data even with same condition, consisting of outliers and unpredicted moves from point to point

The active nature of the UF membrane and its performance during UF life cycle makes it difficult to simulate and optimize its performance.

For Internal Use Only

4. Technology comparison

The approach for SAO project is to use AI/ML models for prediction of UF behavior and optimization algorithms to optimize its performance. Different machine learning algorithms have been investigated to find the best models for simulation and prediction of the UF membrane for a predefined time. These machine learning models include linear and non-linear regression models, Decision Trees, Multi-Layer Perceptron (MLP) and other classical ML models. Apart from prediction modelling, other algorithms have been explored for optimizing the process of UF membrane, reducing fouling, and extending the life cycle of the membrane. These algorithms are deterministic and stochastic optimization approaches e.g., Simplex, Particle Swarm Optimization (PSO) and Genetic Algorithm (GA).

5. Technology selection

The Linear Regression model has been selected for production mode prediction and Decision Tree is used to predict permeability after cleaning based on RMSE results. These models show lower RMSE than the others.

GA optimization algorithm has been selected for UF membrane performance optimization. It is a stochastic algorithm which is used to find the best solution where the search area is very huge and consists of many different solutions. Due to different concentration of chemicals, period of operation, cost of chemicals and energy consumption, the search area for this problem is huge and simple deterministic optimization algorithms like Simplex is not applicable while GA shows better performance in such problem. Another reason for the selection of GA is that it is a common option to use as it is simple to implement and avoid trapping in local optima due to GA operators like crossover and mutation.

6. Technology details (Literature):

There are some GA definitions as follows:

- Search Space - All possible solutions to the problem
- Population - Group of all individuals

- Chromosome - Blueprint for an individual (the length of Chromosome in our problem is the operation period)
- Gene – smallest part of a chromosome
- GA Operators
    - Selection
    - Cross over
    - Mutation
- Fitness Function – evaluate the cost of individuals

Following are GA steps used in this study to optimize the MC and RC intervals and total cost.

1. Initialization
    i. Binary
    ii. Numeric
2. Selection
    i. Roulette Wheel
3. Cross Over (CO)
    i. One-point CO
    ii. Two-points CO
4. Mutation (MU)
    i. One-point
    ii. Multi-point
5. Termination
    i. A solution is found that satisfies minimum criteria
    ii. Fixed number of generations reached
    iii. Allocated budget (computation time/money) reached

Beside these major steps, elitism also can be used to transmit the best individual of population to the next generation. This step speeds up the algorithm convergence.

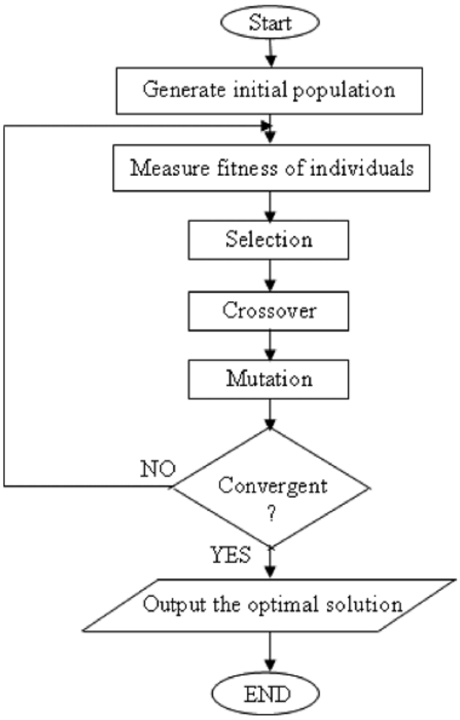


Figure 2. Block diagram of GA algorithm

Figure 1 shows the block diagram of a GA algorithm. In a genetic algorithm, a population of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem is evolved toward better solutions. Each candidate solution has a set of properties (its chromosomes or genotype) which can be mutated and altered; traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. Evolution usually starts from a population of randomly generated individuals, and is an iterative process, with the population in each iteration called a generation.

In each generation of GA, the fitness of every individual in the population is evaluated; fitness is usually the value of the objective function in the optimization problem being solved. The more fit individuals are stochastically selected from the current population, and everyone's genome is modified (recombined and possibly randomly mutated) to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.

A typical genetic algorithm requires:

- a genetic representation of the solution domain,
- a fitness function (objective function) to evaluate the solution domain.

Once the genetic representation and the fitness function are defined, a GA algorithm proceeds to initialize a population of solutions and to improve it through repetitive application of the mutation, crossover, and selection operators.

7. Application of selected technology on SAO

Each UF membrane works in two modes: operation and cleaning. Operation mode is when the UF membrane filtrate the water and cleaning mode is when chemical is used to clean clogging from surface and pores of the membrane. To optimize the performance of the membrane, an algorithm must find the best values for the control system parameters and chemical concentration and schedule. The SAO app consists of two parts: Prediction and Optimization. AI models have been used in prediction mode to predict the membrane permeability for a predefined period while optimization algorithm is used to find the best concentration and schedule of chemicals. Three AI models have been developed and trained to predict operation, maintenance, and recovery cleaning modes. Model 1 is used to find permeability during operation mode, Model 2 is used to find permeability after MC cleaning and Model 3 is used to find permeability after RC cleaning as shown in Figure 3. Historical data is collected from a popular UF membrane to train the models.

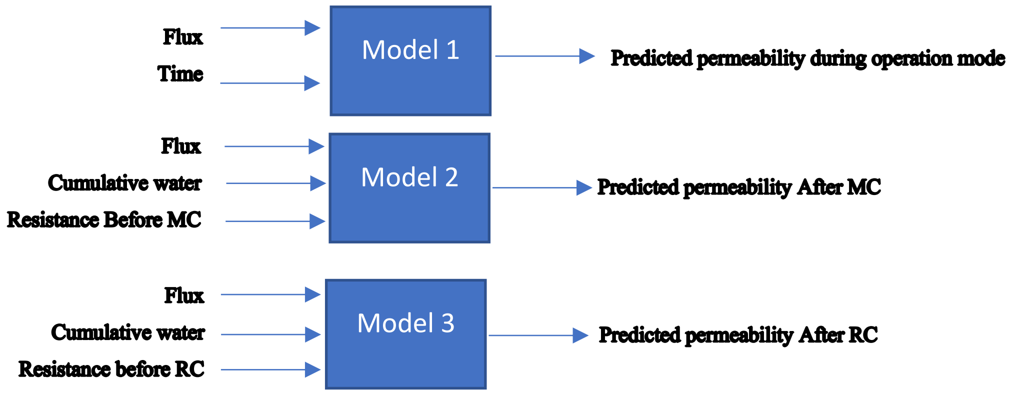


Figure 3. (a) Developed model for UF permeability prediction at different stages

These trained models are used to simulate a membrane operation from current moment to a predefined time (e.g., next one month). Figure 4 shows one simulation of the membrane using three models:

For Internal Use Only

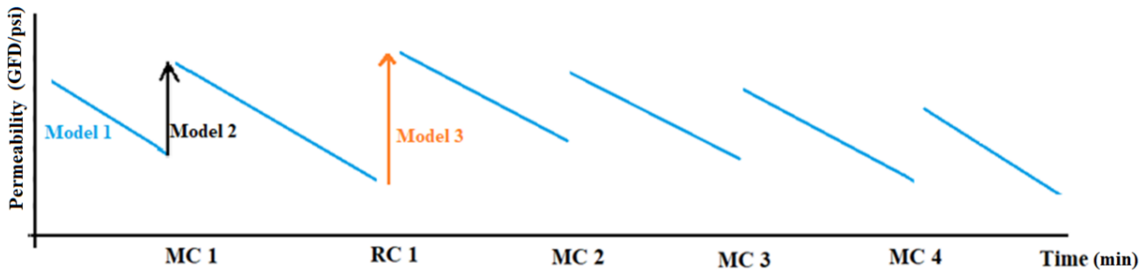


Figure 4. (a) Simulation of the membrane using three mod

Figure 4 only shows one example of many different configuration of MC and RC cleaning schedule. Finding the best configuration of MC and RC for the UF membrane is the biggest challenge. To find the best schedule and concentration for MC and RC, a GA algorithm is developed with the objective of higher production at lower resistance. GA randomly generates schedules like figure 1 and calculates these parameters to find the best solution.

8. Future expansion / adaptability to other use cases

The current algorithm tries to optimize the UF membrane performance by considering RC and MC cleaning. However, it can also improve to cover backwash cleaning or air souring. Another good aspect of the current approach is that by changing the objective of the GA optimization algorithm we can consider other options to the algorithm like cost of operation. The current application can also be developed to adopt with RO operation.

9. SAO Solution Architecture

SAO application is implemented using various components / features available on Yokogawa Cloud Platform (YCloud). SAO collects data from iGreen system periodically via FTP interface. Received Data processed by ETL flows and stored in to timeseries data base of YCloud. On a scheduled interval ETL flows process the data in time series database and runs various python scripts (preprocessing, event detection, etc.) and send the results to VTSCADA systems using RestFul API.

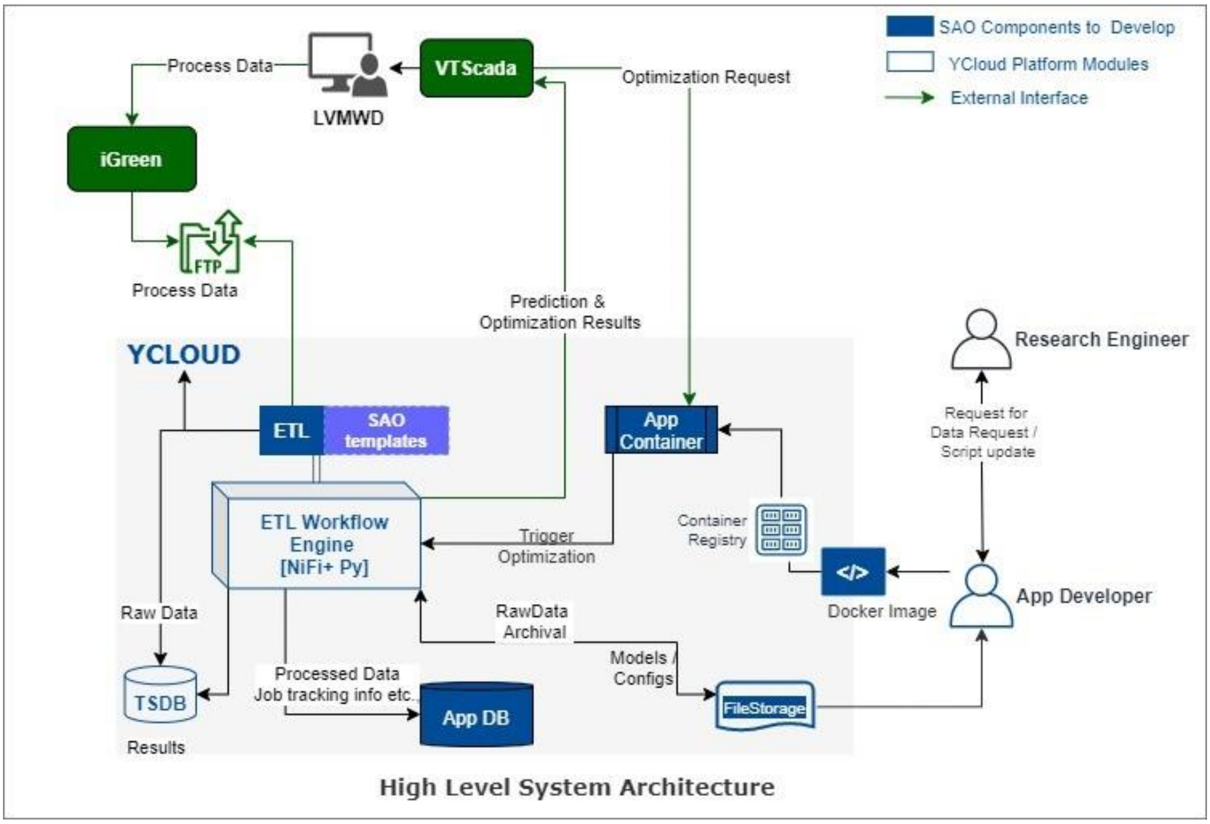


Figure 5. SAO Solution Architecture

SAO can receive optimization / what if scenario simulation request via the RESTful API interface hosted in the App container. It triggers ETL flow for optimization. Optimization results send back to VT SCADA using RestFul API.

Advantages of this Architecture:

- Low code visual ETL flows are easy to understand and maintain.
- Good for Agile ML development where ML model building is evolving. Data science team able to update the scripts / models without changing the application code.

For Internal Use Only

Overview of all Flow groups in ETL Screen can be seen in Figure 6.

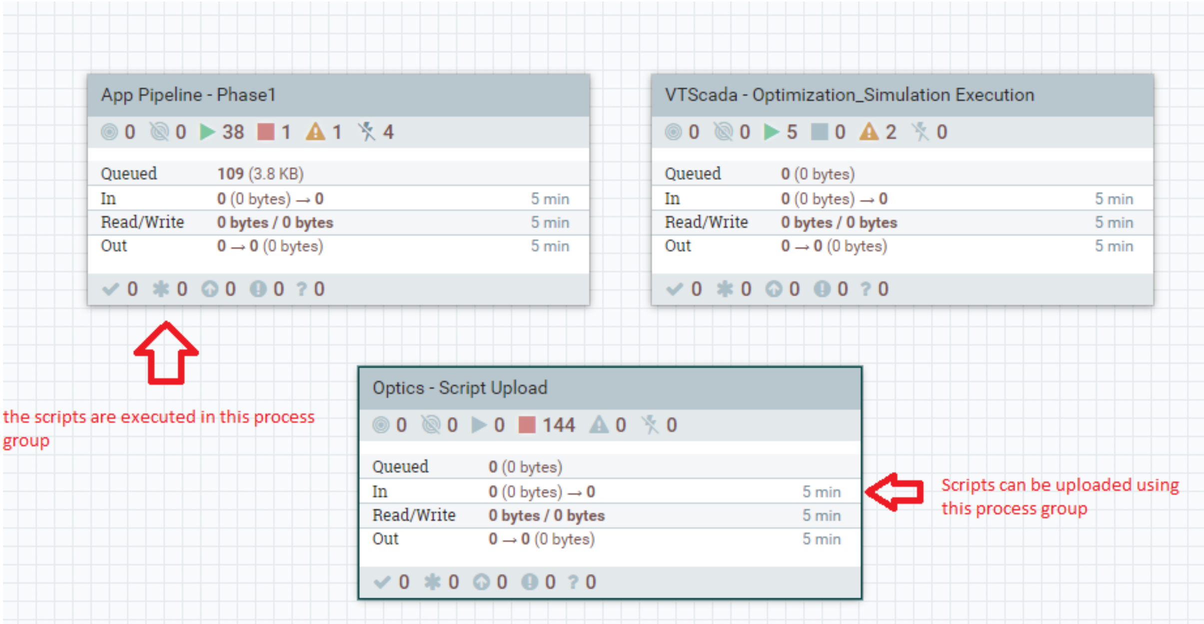


Figure 6. Overview of all Flow groups in ETL Screen

The application flow can be seen in process group App Pipeline-Phase 1

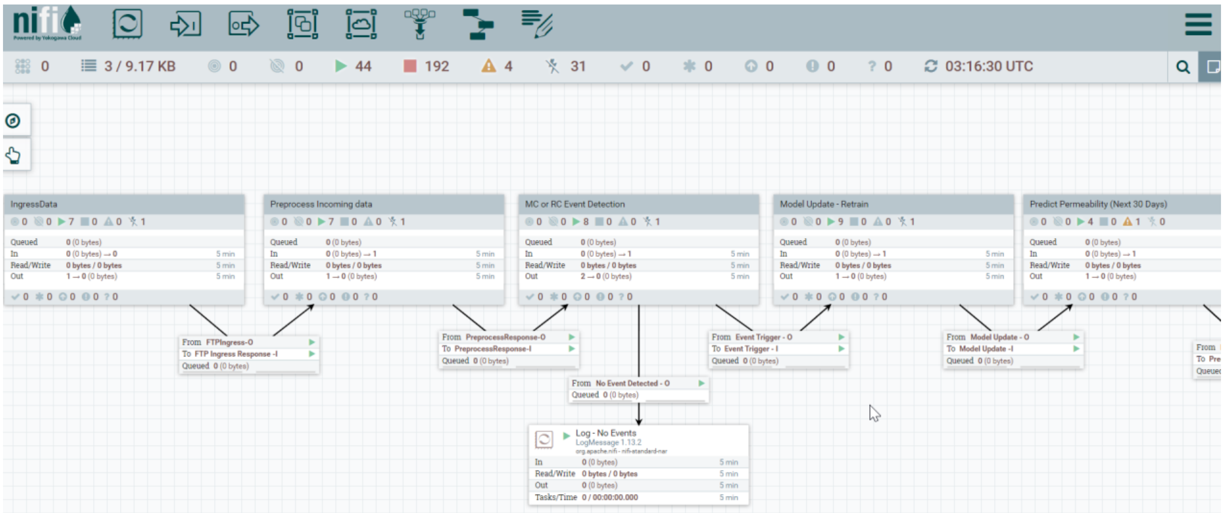


Figure 7. SAO App Pipeline

Short description on what each flow group perform is as follows

- Ingress data - shall read data from ftp server and store it to TSDB periodically (every hour). This schedule can be configured within the ETL screen
- Preprocess incoming data - shall read data from TSDB and do necessary preprocess transformations and save the result data to application DB (Postgres)

- MC, RC event detection - will read preprocess data from DB and detect if any MC/RC happened within defined time window and proceed to next flow. If no events were detected, the flow will not proceed to further downstream flows and will resume again in the next processing cycle.
- If an event happened model update- retrain will be called to retrain the model. This new model will be used for permeability prediction and results will be sent to VTScada Endpoints

Following block diagrams show the details of different SAO modules:

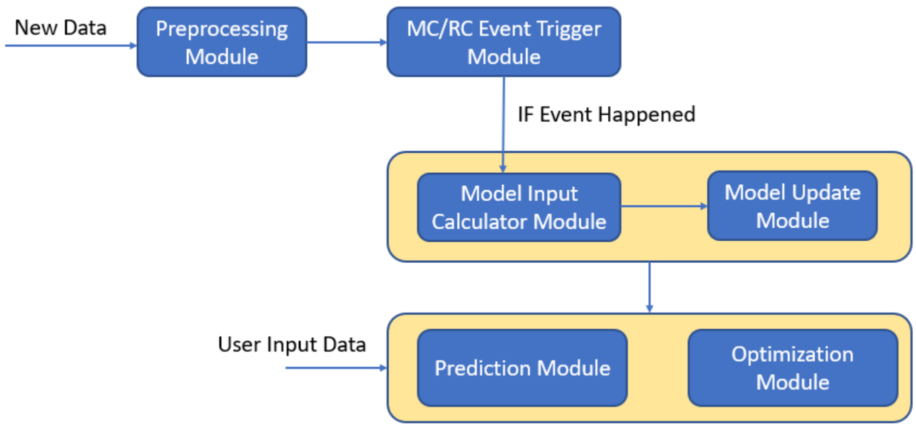
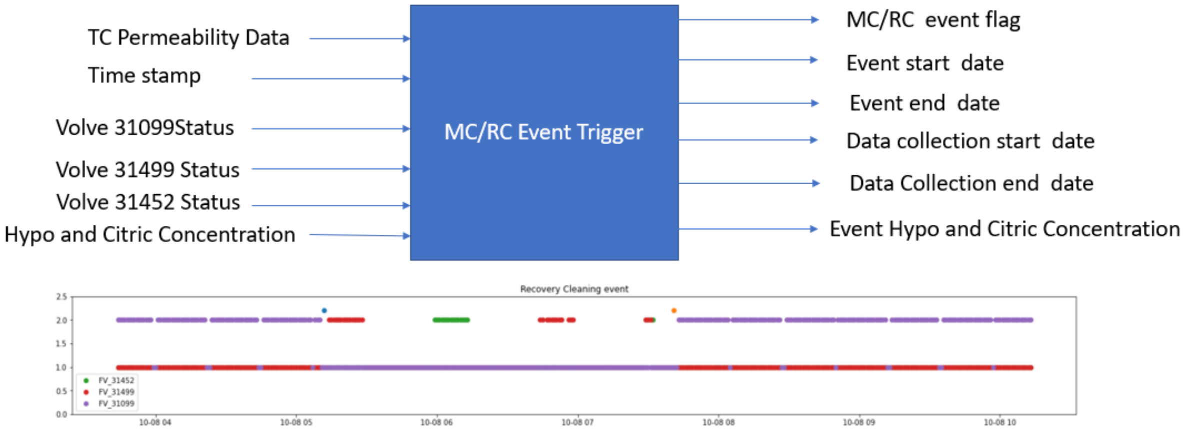


Figure 8. SAO overall flow including different modules

t_stamp	feed_temp	tmp	flux	perm	tc_tmp	tc_flux	tc_perm	FV_31099	FV_31452	FV_31499
2021-10-11 18:44:00	77.918777	5.320611	43.485073	8.289364	6.011001	38.553738	7.139628	2	1	1
2021-10-11 18:45:00	77.918777	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1	1	1
2021-10-11 18:46:00	77.918777	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	2	1	1

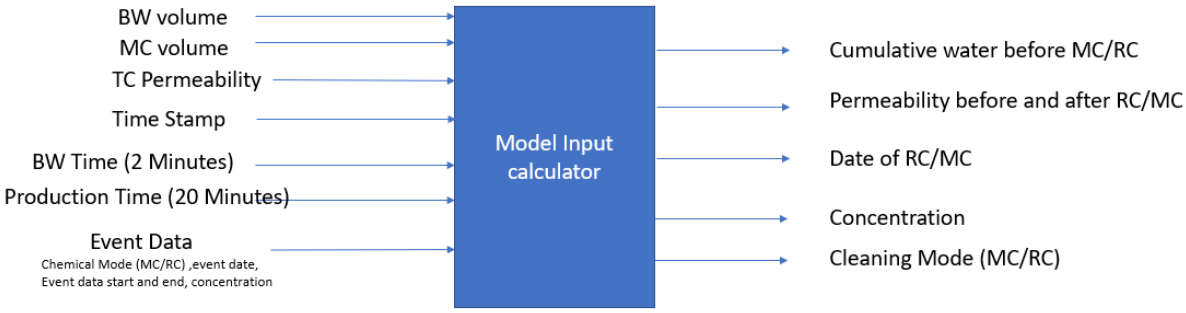
Figure 9. Preprocessing Module with its output data

For Internal Use Only



CC	Start Date	End Date	Data Start	Data End	Hypo Concentration	Citric Concentration
MC	2020-06-18 17:45:00	2020-06-18 18:22:00	2020-06-18 16:45:00	2020-06-18 20:22:00	500.0	
MC	2020-06-22 06:41:00	2020-06-22 07:17:00	2020-06-22 05:41:00	2020-06-22 09:17:00	500.0	
MC	2020-06-25 18:24:00	2020-06-25 19:01:00	2020-06-25 17:24:00	2020-06-25 21:01:00	500.0	

Figure 10. Event Trigger Module



CC	Start Date	End Date	Data Start	Data End	Hypo Concentration	Citric Concentration	Perm_before_cc	Perm_after_cc	Cummulative Water
MC	2020-06-18 17:45:00	2020-06-18 18:22:00	2020-06-18 16:45:00	2020-06-18 20:22:00	500.0		4.986255	6.385851	124.184160
MC	2020-06-22 06:41:00	2020-06-22 07:17:00	2020-06-22 05:41:00	2020-06-22 09:17:00	500.0		5.979871	6.878609	118.142646
MC	2020-06-25 18:24:00	2020-06-25 19:01:00	2020-06-25 17:24:00	2020-06-25 21:01:00	500.0		6.724669	6.989098	147.536355

Figure 11. Model Input Calculator module

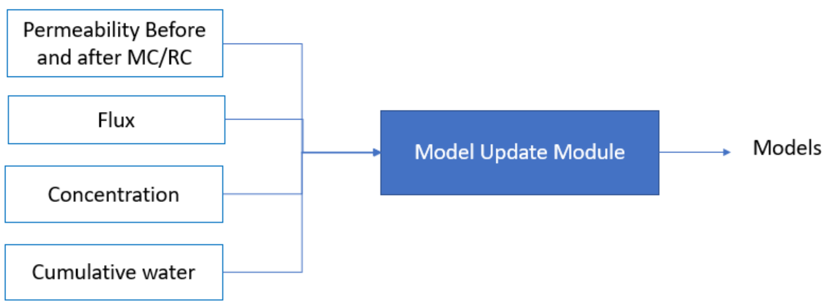


Figure 12. Model Update Module

For Internal Use Only

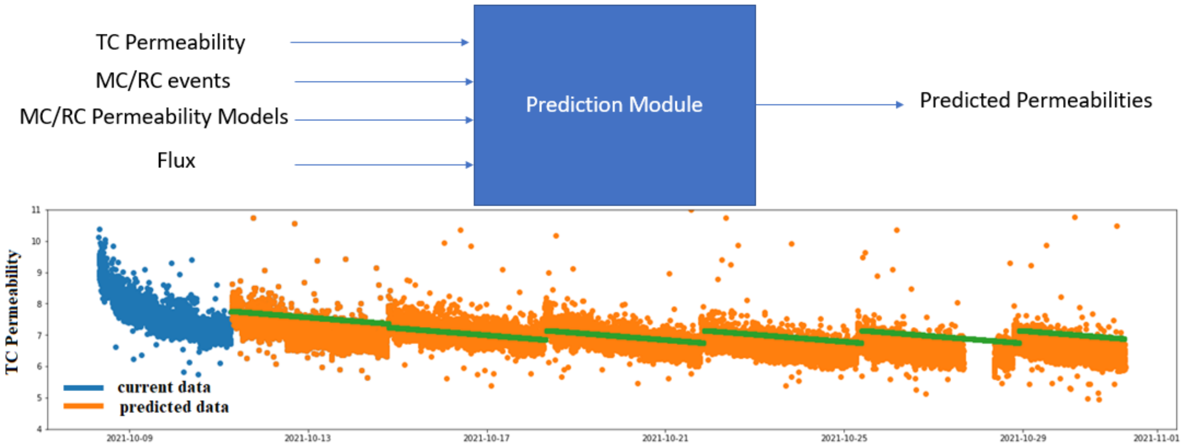


Figure 13. Prediction Module using different models (Production, RC and MC models)

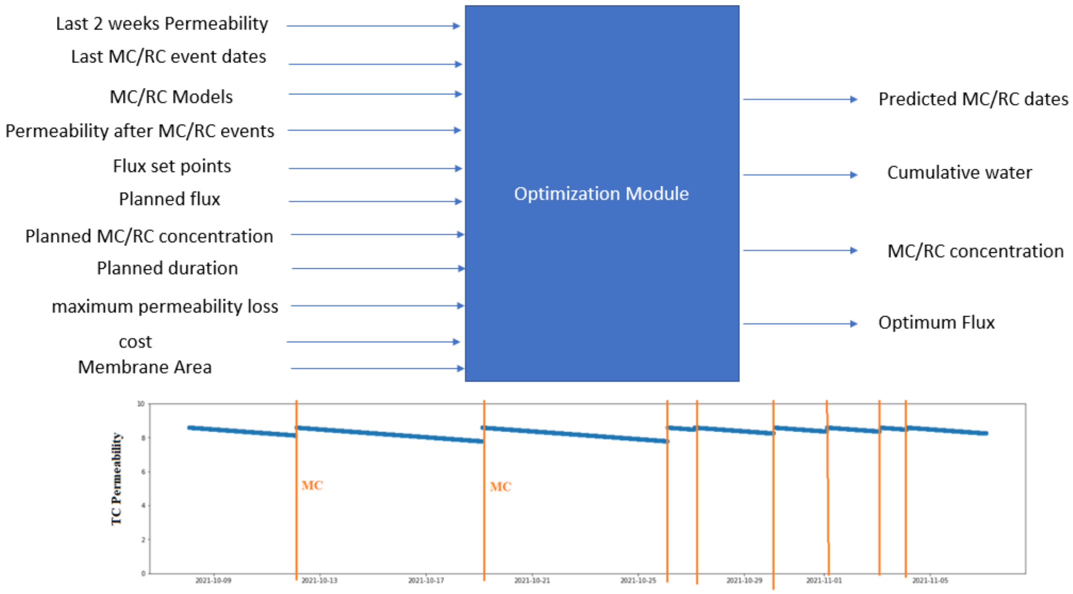


Figure 14. Optimization module

10. SAO Edge deployment Architecture (Investigation in progress)

Initial investigations for executing SAO scripts on eRT3 shows that existing SAO scripts should be refactored and recompiled to be able to run on eRT3 and all the scripts cannot be run on eRT3 due to following.

- eRT3 supports 32-bit OS unlike 64bit systems used to develop existing programs.
- eRT3 runtime memory is limited, so that existing program needs to be changed to handle data in batches for inferencing. Entire data frame cannot be loaded into the memory.

Due to the above issues following alternative architecture is preferred considering the effort required to overcome the limitations and risk to fail to convert the entire application.

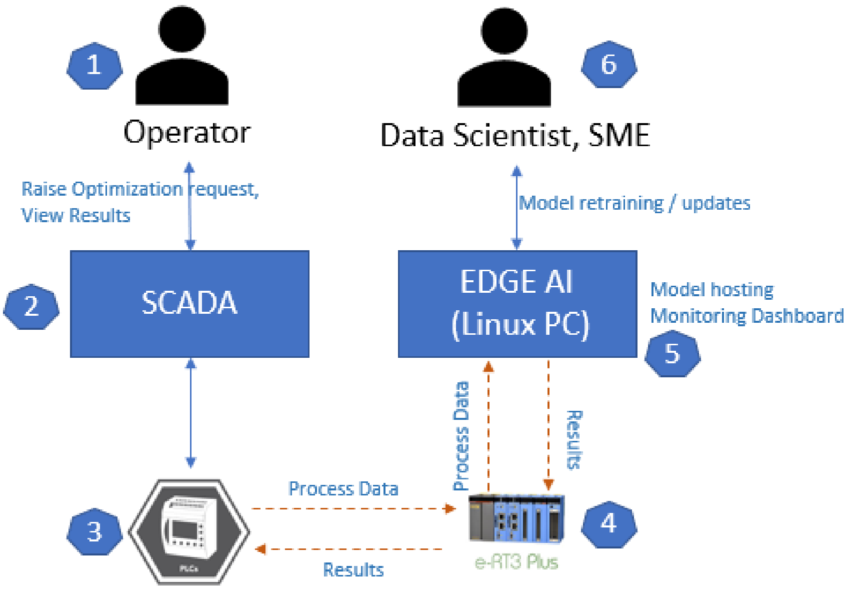


Figure 15. SAO application on eRT3

a) Operators use SCADA graphic to view results or raise optimization request
b) SCADA system send the optimization request to eRT3 plus or set the values at PLC tags and set flag to trigger optimization (depends on capability of site SCADA system)
c) eRT3 plus continuously collect data via PLC runs preprocessing scripts and send to Edge AI machine for inferencing or advanced data transformation / processing.
d) Edge AI machine host the AI models and inferencing runtime. It provides interface for model retraining, configuration, and monitoring.
e) Data scientist / SME use EDGE AI machine for configuration and performance monitoring.

PS: Assume eRT3 can collect data from PLC at site. Connectivity of eRT3 to SCADA and its capability to collect data from existing PLC at site is not tested due to non-availability of hardware modules.

For Internal Use Only