Final Model Report- AquaFlow Technologies

MLOps course, Reichman University
February, 2024
Students names: Shahar Ehrenhalt, Oren Avida, Mayan Stroul, Alexander Gorelik

Analytic Approach

Target definition:
Following the baseline report, in this report we will focus on the LGBM model only, although implementing also the MSET.

The goal of the final model was to reduce the FN ("missed anomalies") alerts, or in other words, to improve the F-beta score of the model. We used beta=2 as the improvement in the recall score is twice as important as improvement in the precision score.

The final model was built on top the baseline model (LGBM and MSET, seperately), but with an improved feature selection process. The new pipeline takes the baseline model (LGBM and MSET) and runs it as usual. After having the model trained with all of the original features, the improved pipeline uses SHAP to reduce less significant features and then trains the model again with the remaining features.

SHAP is used in 2 different ways:

- **SHAP average impact** - using the SHAP model to identify the average impact on model output magnitude, and then removing the least 15% contributing features (4 features in this case).
- **Heuristic SHAP** - in this improvement, we defined a new concept of "directional SHAP" in which we calculated for each feature the correctness of contribution to the score according to the sample's true label. Then we calculated the F-beta score for each feature.
  Under the second method we removed 7 features out of 24 (F-beta score < 0.15).

The inputs of the final model are the same for the base model, but in the second training the input will include only the best features.

As mentioned in the baseline report, the data used is an 8-dimensional Time Series (TS) data obtained from sensors. Following is an example of the raw dataset with all of the features:

datetime	Accelerometer1RMS	Accelerometer2RMS	Current	Pressure	Temperature	Thermocouple	Voltage	Volume Flow RateRMS
2020-03-09 10:14:33	0.026588	0.040111	1.330200	0.054711	79.3366	26.0199	233.062	32.0000
2020-03-09 10:14:34	0.026170	0.040453	1.353990	0.382638	79.5158	26.0258	236.040	32.0000
2020-03-09 10:14:35	0.026199	0.039419	1.540060	0.710565	79.3756	26.0265	251.380	32.0000
2020-03-09 10:14:36	0.026027	0.039641	1.334580	0.382638	79.6097	26.0393	234.392	32.0000
2020-03-09 10:14:37	0.026290	0.040273	1.078510	-0.273216	79.6109	26.0420	225.342	32.0000
...	...	...	...	...	...	...	...	...
2020-03-09 13:59:09	0.028142	0.039445	0.981651	0.382638	66.9901	24.7381	232.774	32.0000
2020-03-09 13:59:10	0.028283	0.040615	0.966102	0.054711	66.8982	24.7404	227.030	32.0226
2020-03-09 13:59:11	0.028778	0.041955	0.827491	-0.601143	66.8706	24.7499	228.784	32.9781
2020-03-09 13:59:12	0.028282	0.041094	1.204300	0.054711	66.9742	24.7464	235.070	32.0000
2020-03-09 13:59:13	0.028482	0.040634	1.524940	-0.273216	66.9312	24.7440	244.074	32.0226

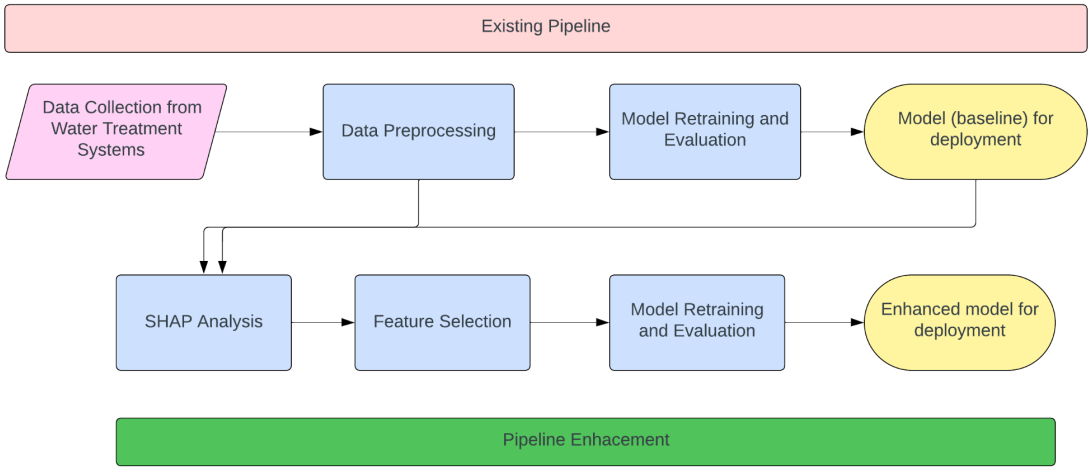
12712 rows × 8 columns

Same preprocessing steps were performed in the improved model.

Solution Description

Solution components -

The following scheme shows the flow of the process. The baseline model is the top row:



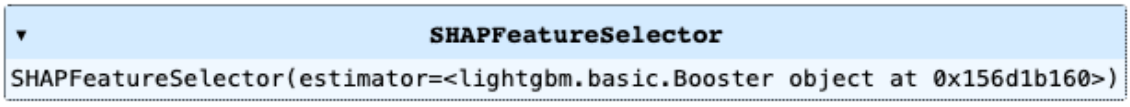
In the improved model we used SHAP to improve the feature selection process. SHAP (SHapley Additive exPlanations) is a machine learning interpretability method that explains the output of any model by quantifying the impact of each feature on the prediction. It uses game theory principles, attributing an average contribution of each feature across all possible combinations.

The pipeline starts with the existing model, that includes data collection from the water treatment systems, preprocessing of the data, model training, and model evaluation. After that, the SHAP part comes in (with either the **SHAP average impact** or the **Heuristic SHAP**), and helps in finding the most relevant features. After the SHAP analysis, the feature selection process takes place, and after that the model is trained and evaluated again.

As mentioned, we used the **SHAP average impact** to identify the average impact on model output magnitude, and then removing the least 15% contributing features (4 features in this case).

After seeing that this did not improve the F-beta score, we decided to also use the **Heuristic SHAP**.

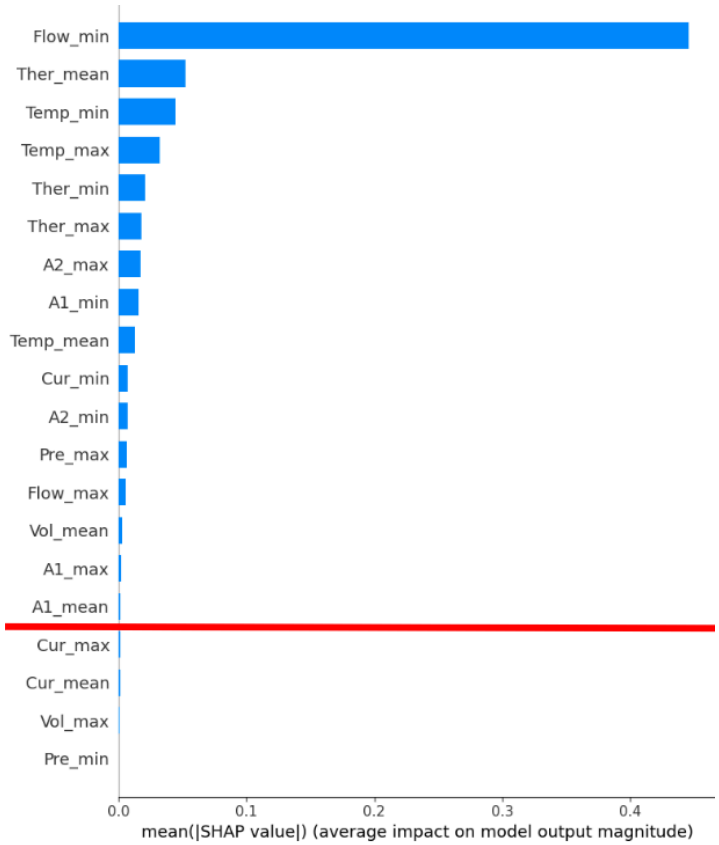
Under the **SHAP average impact**, we are implementing the custom SKLearn pipeline step:



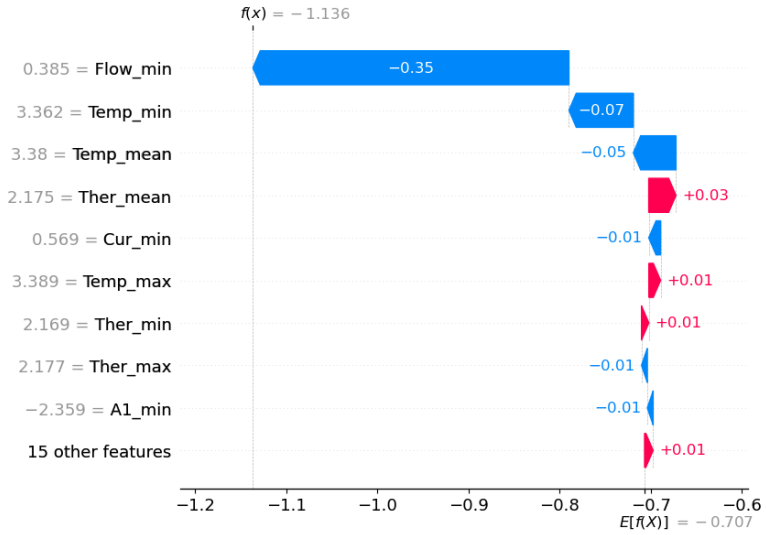
Under the **Heuristic SHAP** we enhanced the “SHAPFeatureSelector” shown above:

SHAP average impact:

Under this solution we calculated the average impact on model output magnitude for each feature and dropped the 15% of the features with the lowest contribution:



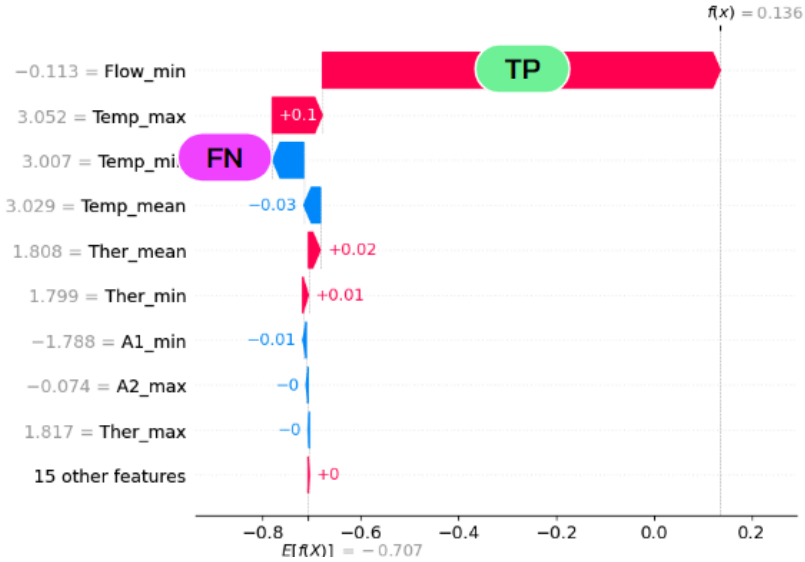
Heuristic SHAP:



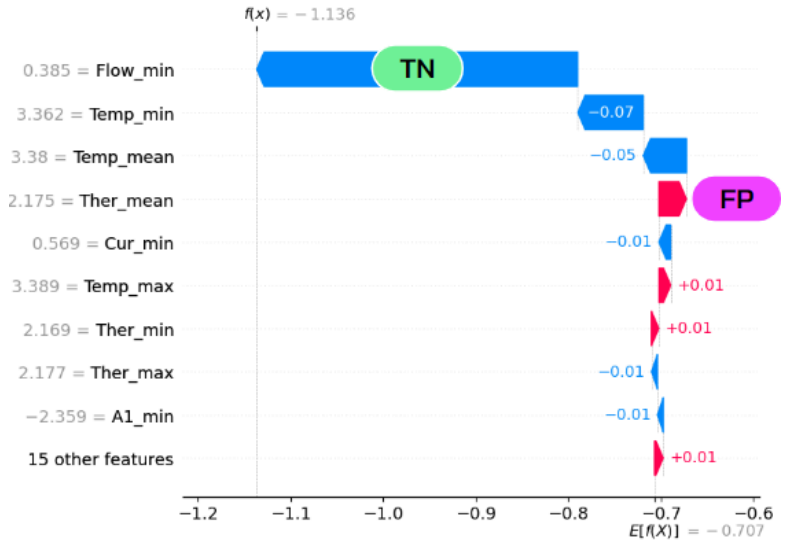
After noticing that the **SHAP average impact** results did not improve compared to the baseline model, we came up with a new approach to use the SHAP values.

We noticed that the **SHAP average impact** calculation uses the absolute SHAP value of each feature. Thus, we decided to use a more heuristic method (i.e. " **Heuristic SHAP**"), in which we tried to understand which feature contributes the most to FNs, and remove those features from the dataset. We preferred to look at the impact of the feature, but also to take into account the label of the sample. In other words, we wanted to see a positive impact when the label is positive and a negative impact when the label is negative. To do so, we calculated for each feature its contribution to the measured metrics (FN/FP/TN/TP). Based on that, we calculated the F-beta score for each feature.

For example, in a case of an anomaly, if the SHAP value is positive (i.e. "pushes" towards the "right" direction), the feature will get a "TP point", and in case the value is negative, the feature will get a "FN point":



If the true label is not anomaly, then in case of a negative SHAP value (i.e. “pushes” towards the “right” direction), the feature will get a “TN point”, while in case the value is positive, the feature will get a “FP point”:



Then we calculated the F-beta score for each feature based on their points. To ensure fair assessment of feature contributions in our heuristic SHAP method, we normalize the F-beta score by the count of non-zero SHAP "votes" for each feature. This step prevents inflated scores for features with sporadic correct SHAP directions, prioritizing features that consistently contribute to model accuracy. This normalization balances impact with reliability, refining feature selection and model enhancement strategies.
Then we removed the features with F-beta normalized score that is less than 0.15:

24 rows | 24 rows × 9 columns

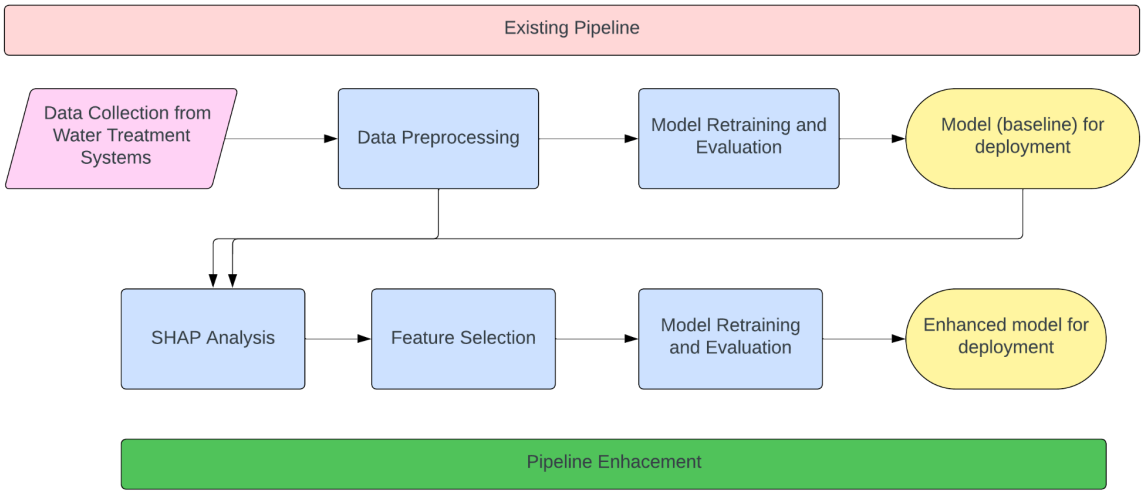
	TP	FP	TN	FN	recall	precision	f_beta	contrib_coef	f_beta_score_normed
Flow_min	3858	175	8219	451	0.895335	0.956608	0.906954	1.0	0.906954
Temp_mean	3759	6115	2279	550	0.872360	0.380697	0.693287	1.0	0.693287
Temp_min	3760	6286	2108	549	0.872592	0.374278	0.689099	1.0	0.689099
A1_mean	3471	6518	1876	838	0.805523	0.347482	0.637466	1.0	0.637466
Vol_mean	2970	5273	3121	1339	0.689255	0.360306	0.582833	1.0	0.582833
Cur_min	2954	5386	3008	1355	0.685542	0.354197	0.577495	1.0	0.577495
Cur_mean	2765	5541	2853	1544	0.641680	0.332892	0.541265	1.0	0.541265
A2_max	2573	4523	3871	1736	0.597122	0.362599	0.528728	1.0	0.528728
Ther_max	2447	4053	4341	1862	0.567881	0.376462	0.515462	1.0	0.515462
A2_min	2605	5716	2678	1704	0.604549	0.313063	0.509645	1.0	0.509645
A1_min	2341	3760	4634	1968	0.543282	0.383708	0.501564	1.0	0.501564
Ther_min	1904	2495	5899	2405	0.441866	0.432826	0.440028	1.0	0.440028
Pre_max	1538	3033	5361	2771	0.356927	0.336469	0.352639	1.0	0.352639
Ther_mean	1082	1198	7196	3227	0.251102	0.474561	0.277208	1.0	0.277208
Temp_max	956	5258	3136	3353	0.221861	0.153846	0.203838	1.0	0.203838
Cur_max	764	1415	6979	3545	0.177303	0.350620	0.196755	1.0	0.196755
A1_max	450	903	7491	3859	0.104433	0.332594	0.121039	1.0	0.121039
Flow_max	504	4617	3777	3805	0.116964	0.098418	0.112716	1.0	0.112716
Vol_max	142	296	8098	4167	0.032954	0.324201	0.040172	1.0	0.040172
A2_mean	0	0	0	0	0.000000	0.000000	0.000000	0.0	0.000000
Pre_min	0	0	0	0	0.000000	0.000000	0.000000	0.0	0.000000
Pre_mean	0	0	0	0	0.000000	0.000000	0.000000	0.0	0.000000
Vol_min	0	0	0	0	0.000000	0.000000	0.000000	0.0	0.000000
Flow_mean	0	0	0	0	0.000000	0.000000	0.000000	0.0	0.000000

Data and Features

The raw data is an 8-dimensional time series data obtained from the company's water filtration systems. The data includes 34,000 samples splitted 34 files of 1,000 samples.
The features include Accelerometer1RMS, Accelerometer2RMS, Current, Pressure, Temperature, Thermocouple, Voltage, and Volume Flow RateRMS (After preprocessing, the input data includes 24 features).

Algorithm & Learner

As a baseline model, we used the LightGBM model as given. LightGBM (Light Gradient Boosting Machine) is a gradient boosting framework developed by Microsoft that uses tree-based learning algorithms. It's designed for speed and efficiency with the capability of handling large datasets. LightGBM is popular for its performance and lower memory usage compared to other gradient boosting frameworks.



The learner is the LGBM.

The hyperparameters for this model include learning rate, min_data_in_leaf, max_depth, and num_leaves. The baseline model includes optimization of those hyper-params.

```
lgb_params = {
        'objective':'binary',
        'metric':'binary_error',
        'force_row_wise':True,
        'seed':0,
        'learning_rate':0.0424127,
        'min_data_in_leaf':15,
        'max_depth':24,
        'num_leaves':29
    }
```

As mentioned, we focused in this report only on the LGBM although implementing both LGBM and MSET.

Results

After running the Classic SHAP on the LGBM model, we saw no improvement in the results, and thus we decided to use a heuristic improvement of the SHAP model. This led to improvement in the results -

LGBM -

LGBM (baseline model)	Predicted Non-Anomaly	Predicted Anomaly
True Non-Anomaly	**1041**	**0**
True Anomaly	**145**	**658**

SHAP average impact + LGBM	Predicted Non-Anomaly	Predicted Anomaly
True Non-Anomaly	**1041**	**0**
True Anomaly	**145**	**658**

Heuristic SHAP + LGBM	**Predicted Non-Anomaly**	**Predicted Anomaly**
True Non-Anomaly	**1041**	**0**
True Anomaly	**139**	**664**

Results comparison:

Metric	**LGBM (baseline model)**	**SHAP average impact + LGBM**	**Heuristic SHAP + LGBM**
Accuracy	0.92	0.92	0.93
Precision	1	1	1
Recall	0.81	0.81	0.83
F-2 Score	0.85	0.85	0.86

MSET -

In our attempt to integrate our SHAPFeatureSelector step with the MSET model pipeline, we faced several challenges but aimed to maintain the consistency of the pipeline with the baseline. For model-agnostic adaptation, we utilized KernelExplainer with MSET, chosen for its compatibility with gradient-opaque models, offering broad applicability. However, its detailed feature subset evaluations necessitated extensive computational effort, requiring about 75 hours to explain the training dataset. This exhaustive process did not yield performance improvements, attributed mainly to training data selection practices. The baseline's reliance on the initial 400 samples from each file, primarily non-anomalous, likely impeded effective feature explanation. This led to negligible TP and FN rates, rendering F-beta score derivation impractical due to the scarcity of anomalous cases in the training set. Consequently, in cases there we shift back to assessing features' overall impact on model output, sidestepping individual F-beta scores often nullified by these constraints.

MSET (baseline model)	**Predicted Non-Anomaly**	**Predicted Anomaly**
True Non-Anomaly	**19,970**	**4,364**
True Anomaly	**2,101**	**10,966**

SHAP + MSET	**Predicted Non-Anomaly**	**Predicted Anomaly**
True Non-Anomaly	**21,810**	**2,524**
True Anomaly	**3,844**	**9,223**

Results comparison:

Metric	**MSET (baseline model)**	**SHAP + MSET**
Accuracy	0.827	0.830
Precision	0.715	0.785
Recall	0.839	0.706
F-2 Score	0.811	0.720