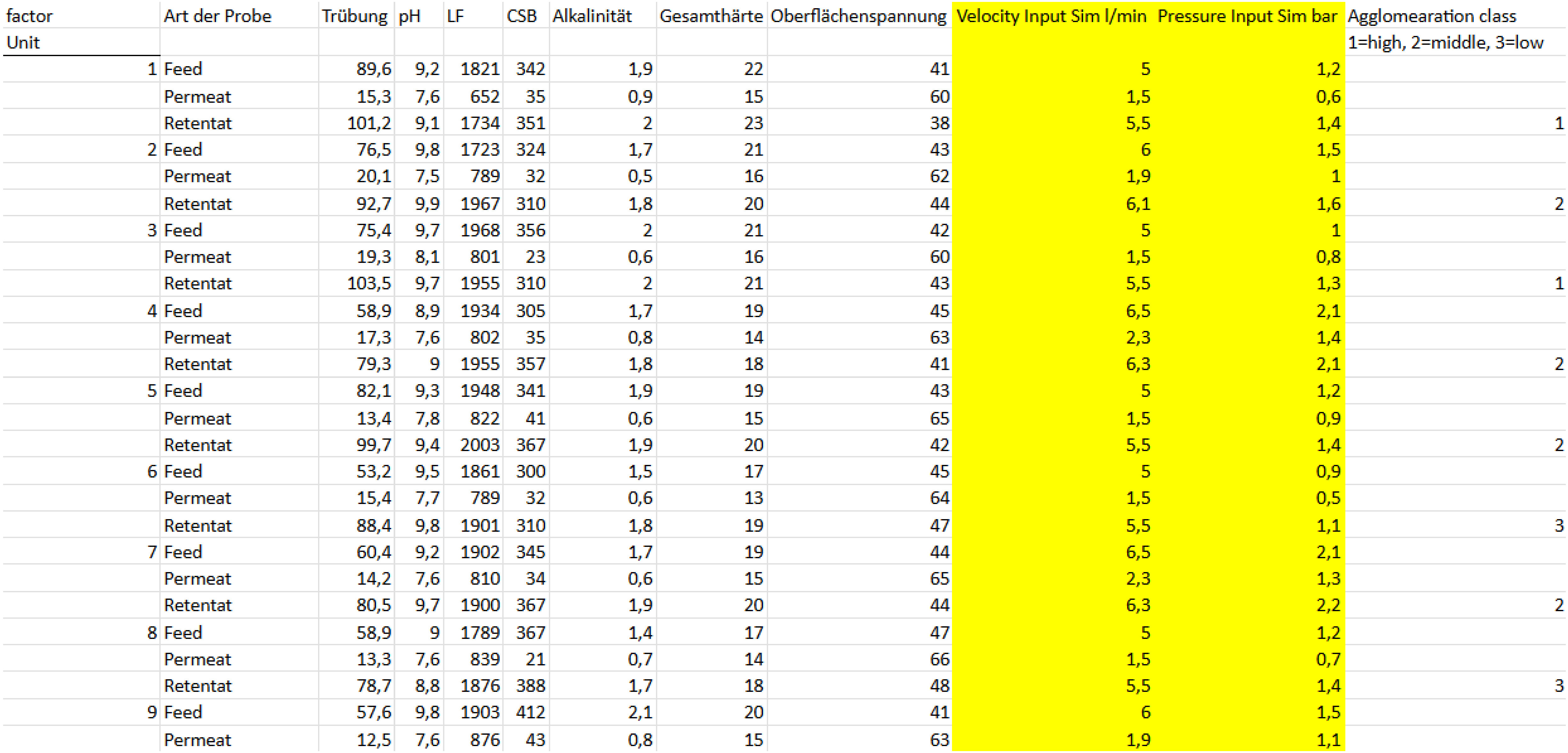
AIMFiltech Software Platform and Intelligence

POLITECHNIKA WARSZAWSKA

WARSAW UNIVERSITY OF TECHNOLOGY

Illia Rohalskyi
Jan Krzysztoforski

Data Overview



factor	Art der Probe	Trübung	pH	LF	CSB	Alkalinität	Gesamthärte	Oberflächenspannung	Velocity Input Sim l/min	Pressure Input Sim bar	Agglomearation class
Unit											1=high, 2=middle, 3=low
1	Feed	89,6	9,2	1821	342	1,9	22	41	5	1,2	
	Permeat	15,3	7,6	652	35	0,9	15	60	1,5	0,6	
	Retentat	101,2	9,1	1734	351	2	23	38	5,5	1,4	1
2	Feed	76,5	9,8	1723	324	1,7	21	43	6	1,5	
	Permeat	20,1	7,5	789	32	0,5	16	62	1,9	1	
	Retentat	92,7	9,9	1967	310	1,8	20	44	6,1	1,6	2
3	Feed	75,4	9,7	1968	356	2	21	42	5	1	
	Permeat	19,3	8,1	801	23	0,6	16	60	1,5	0,8	
	Retentat	103,5	9,7	1955	310	2	21	43	5,5	1,3	1
4	Feed	58,9	8,9	1934	305	1,7	19	45	6,5	2,1	
	Permeat	17,3	7,6	802	35	0,8	14	63	2,3	1,4	
	Retentat	79,3	9	1955	357	1,8	18	41	6,3	2,1	2
5	Feed	82,1	9,3	1948	341	1,9	19	43	5	1,2	
	Permeat	13,4	7,8	822	41	0,6	15	65	1,5	0,9	
	Retentat	99,7	9,4	2003	367	1,9	20	42	5,5	1,4	2
6	Feed	53,2	9,5	1861	300	1,5	17	45	5	0,9	
	Permeat	15,4	7,7	789	32	0,6	13	64	1,5	0,5	
	Retentat	88,4	9,8	1901	310	1,8	19	47	5,5	1,1	3
7	Feed	60,4	9,2	1902	345	1,7	19	44	6,5	2,1	
	Permeat	14,2	7,6	810	34	0,6	15	65	2,3	1,3	
	Retentat	80,5	9,7	1900	367	1,9	20	44	6,3	2,2	2
8	Feed	58,9	9	1789	367	1,4	17	47	5	1,2	
	Permeat	13,3	7,6	839	21	0,7	14	66	1,5	0,7	
	Retentat	78,7	8,8	1876	388	1,7	18	48	5,5	1,4	3
9	Feed	57,6	9,8	1903	412	2,1	20	41	6	1,5	
	Permeat	12,5	7,6	876	43	0,8	15	63	1,9	1,1	

Data – Details and Challenges

Details:

1. Data was collected in was collected by Dr. Igor Kogut using internal methods
2. Target variable is agglomeration class, 1 being highest and 3 being lowest
3. There is only 40 data points and around 29 features
4. The data quality is very good, no data to be imputed or cleaned

Challenges:

1. One datapoint takes up 3 different rows. Weird formatting
2. The need to integrate OpenFOAM data as part of the project
3. OpenFOAM simulations take a lot of time, not scalable unless parallelized
4. Very small dataset with a lot of features. Prone to overfitting

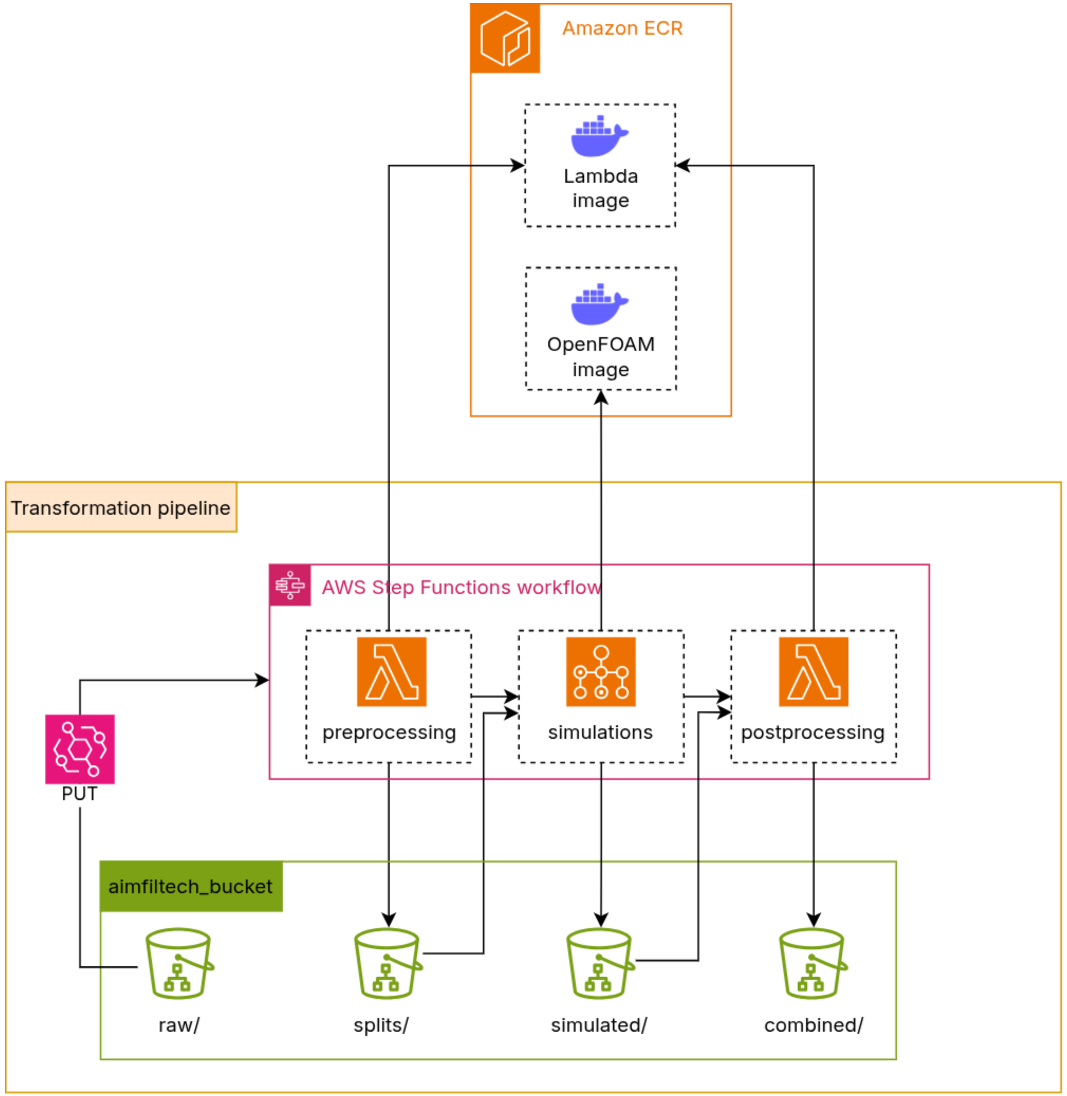
Data Transformation Implementation

To prepare the data, we built a small, efficient pipeline that automates data formatting, simulation, and post-processing — all in the cloud

- Lambda (Preprocessing): Merges 3-row entries into single rows, renames for OpenFOAM

- OpenFOAM on AWS Batch: Runs simulations in parallel (1 job per 10 samples)

- Lambda (Postprocessing): Combines all output CSVs into a single dataset



Modeling – XGBoost with Optuna Tuning

Why XGBoost?

- Proven winner in many real-world ML tasks
- Relatively fast to train, easy to interpret
- Robust to overfitting with good hyperparameter tuning

DMLC XGBoost

Why Optuna?

- Modern, uncomplicated hyperparameter optimization library
- Has a lot of search algorithms implemented
- Improves XGBoost results by finding better set of hyperparameters

OPTUNA

Modeling – Hyperparameter optimization

Tree-structured Parzen Estimator (TPE):

TPE is Bayesian-inspired, using two probability models: one for good results, one for bad. It samples more from areas that previously gave good results.

Hyperparameter choices:

❶ n_estimators: how many trees

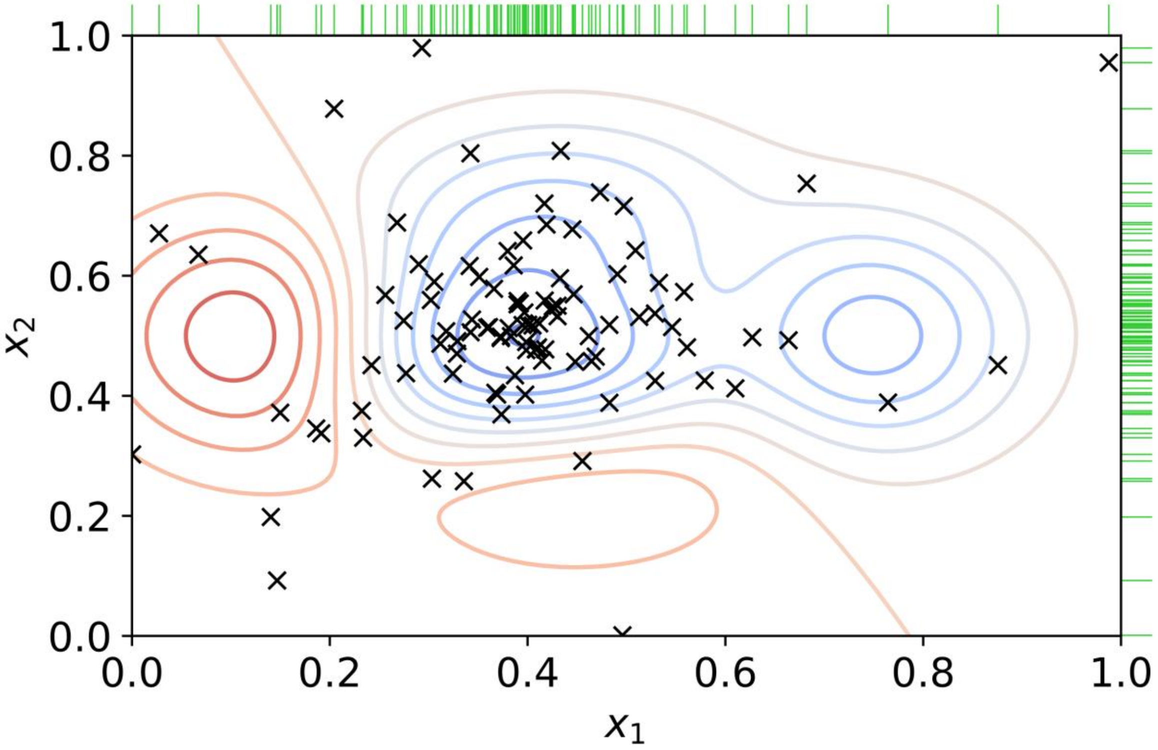
❷ max_depth: tree depth

❸ min_samples_split: min samples to split

❹ min_samples_leaf: min samples at leaf node

❺ max_features: how many features to consider per split

❻ bootstrap: whether to use bootstrapped samples



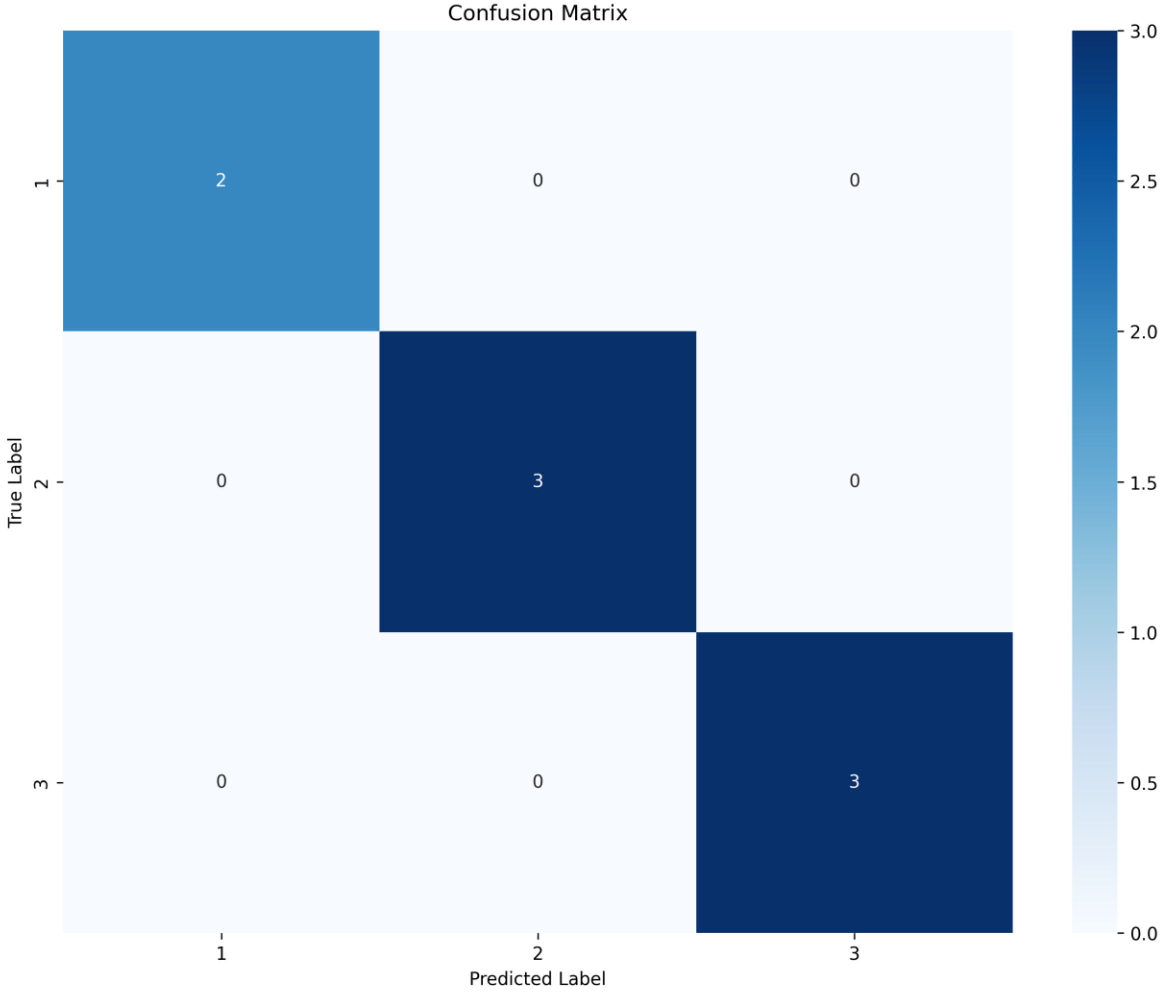
Alexander Elvers via Wikipedia Commons

Modeling – Performance

The trained model achieved perfect accuracy on the test set — every prediction matched the true label

However, this result should be taken with caution — the dataset is small, and the test set had just 8 points.

Still, it's a very promising signal that the approach and features are working.



Deployment and Machine Learning Operations

❶ Batch-based deployment instead of online deployment

Why?

- Cost optimization: no need for servers to “wait” for requests
- Simpler infrastructure management

❷ Infrastructure-as-Code

Terraform builds the entire cloud system in one go.
No manual server configuration, just `terraform apply`

❸ CI/CD Pipeline – Automation and Tests using Github Actions

Small Continious Integration / Continious Deployment (CI/CD) script handles:

- Linting (Code style checks on Python files)
- Building Docker container and testing OpenFoam module

Testing and CI/CD could be extended given more time and commitment.
The focus was on delivering functional system over ideal unfinished system

Deployment and Machine Learning Operations

❹ Deployment via bash script instead of CI/CD

Why?

- Can spin up servers when directly working with them, and destroying when not in use
- Not the best practice, but very budget efficient

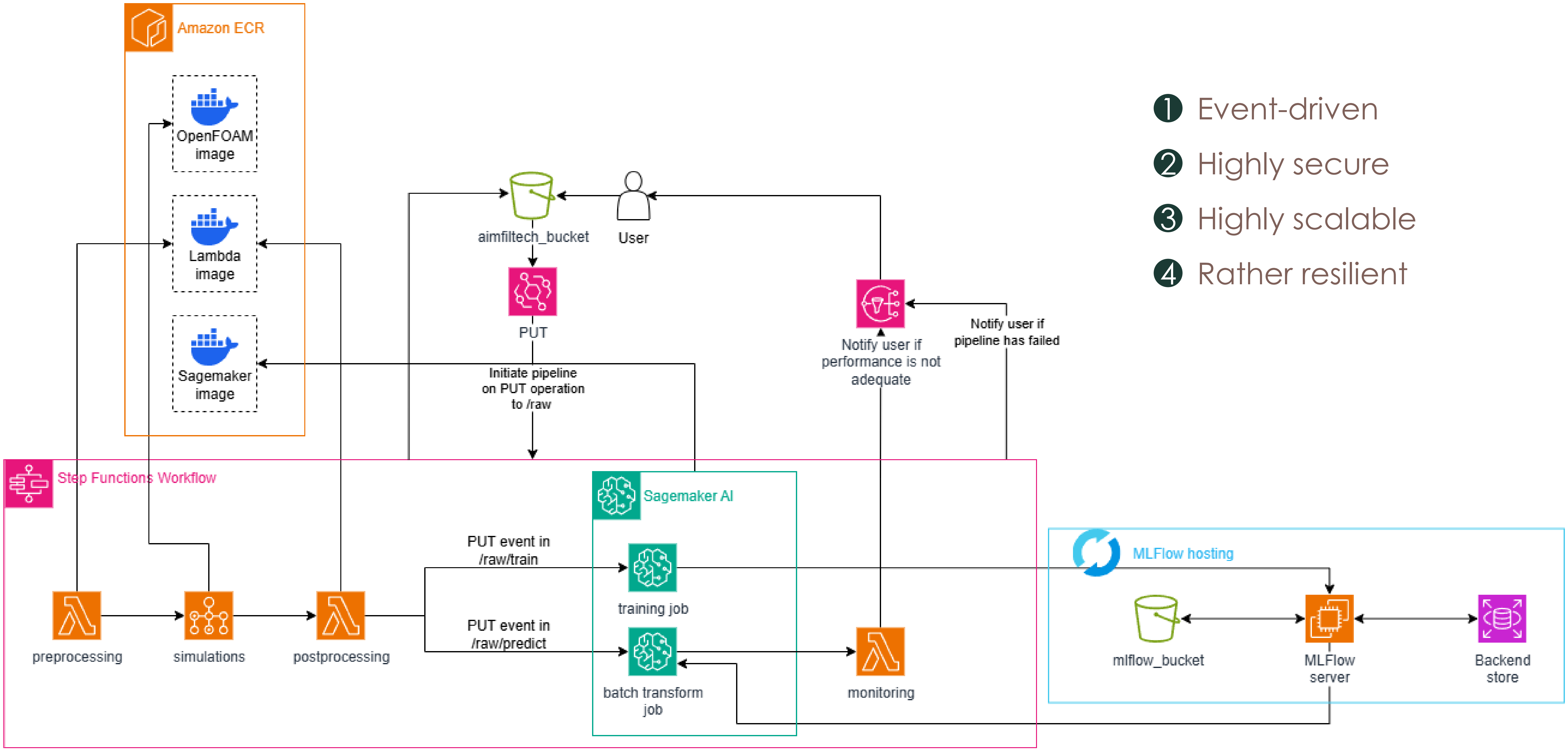
❺ Logging and CloudWatch logs

Core components (Lambda, SageMaker, Batch) have basic CloudWatch logging. There's room to expand with more detailed logs and alerts.

❻ Code quality

The code is generally clean with clear separation of concerns and understandable structure. Some documentation is included, and most components are logically organized. Still, it would benefit from consistent linting and more complete docstrings to improve readability and maintainability

High-Level Overview of The Cloud Architecture



1. Event-driven
2. Highly secure
3. Highly scalable
4. Rather resilient

AWS Well-Architected Pillars Ranking (subjective)

- 🟡 Operational Excellence (6/10):
  The system is automated and modular, aiding maintainability. Logging, monitoring, and the CI/CD pipeline are basic and need improvement.

- 🟢 Security (9/10):
  Strong security with VPC isolation and strict IAM roles ensures good protection of resources and data.

- 🟡 Reliability (7/10):
  Generally reliable, but MLFlow on EC2 lacks fault tolerance, and the missing comprehensive test suite reduces stability confidence.

- 🟢 Performance Efficiency (9/10):
  AWS Batch and SageMaker enable efficient resource use and good scalability with workload demands.

- 🟢 Cost Optimization (9/10):
  On-demand server spin-up saves costs; further savings possible via spot instances and container size optimization.

- 🟢 Sustainability: (8.5/10):
  This project leverages efficient AWS managed services and runs in a region powered by renewable energy, but the always-on EC2 prevent a higher rating.

Thank you for your attention

POLITECHNIKA WARSZAWSKA

WARSAW UNIVERSITY OF TECHNOLOGY