vår energi

Data-Driven Estimation of Individual Branch Flow in Multilateral Wells

Author: Tord Aarø, Petroleum Engineer & Student

MSc Research Project at Imperial College London & Vår Energi

EWPF Stavanger, August 2025

Table of Contents

Introduction

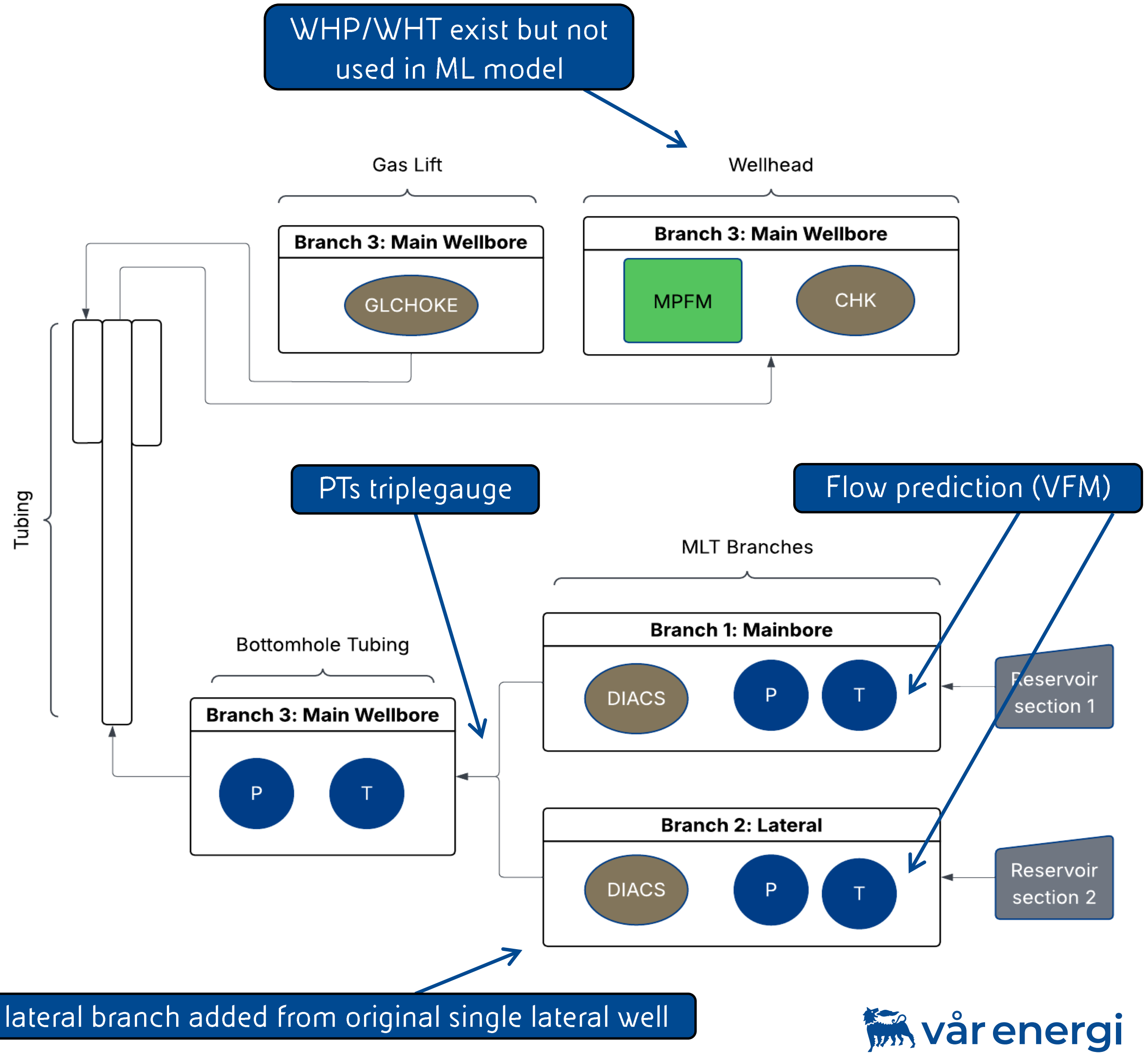
Problem Statement

Background

- Retrofitted Multilateral wells (rMLTs) are increasingly common, targeting different reservoir sections
- Flow estimates from each branch is important for reservoir modelling and production optimization
- Multiphase flow meter (MPFM) is only available topside, measuring commingled production
- Branch chokes (DIACS) controls branch flow and adds complexity to commingled flow
- Attempts to use analytical solutions (e.g. Bernoulli's) have not been successful with current pressure drop
- Machine learning methods can be applied to capture the complex and non-linear relationships present in these conditions

Proposed Solution

- Develop a semi-supervised deep learning model to predict the multiphase flow for each individual branch in one rMLT well, using 3 other MLTs in the training set



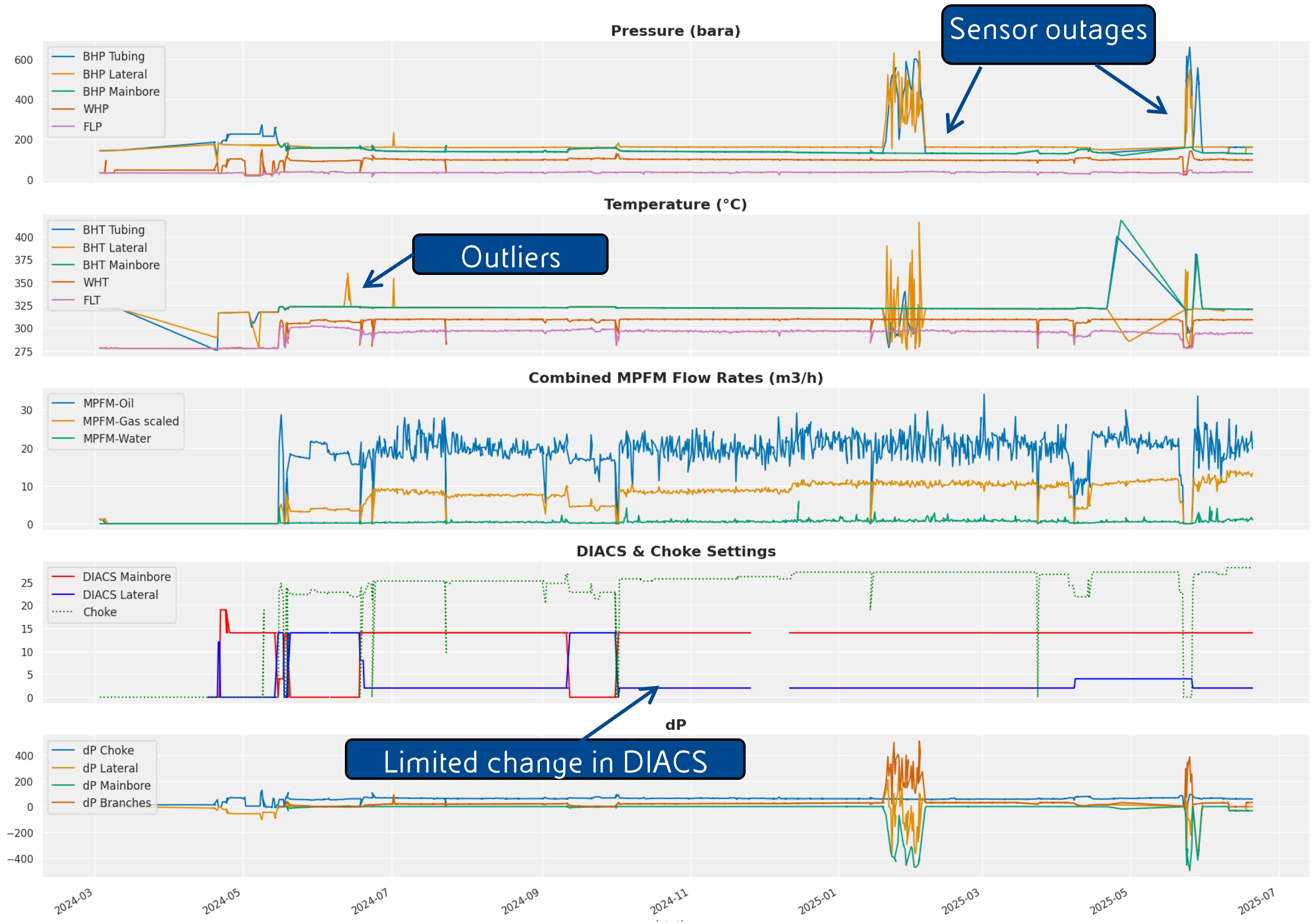
Data Analysis

The available data has several problems:

- Sensor drifting, sensor mapping, sensor outages and sensor measurement error
- Years of time series data available from 4 wells, but with limited variation, such as static DIACS settings
- Minor pressure sensor offset/error becomes large during pressure drop calculations, especially for fully open DIACS

Measures taken to address problems:

- Sensor remapping
- Remove time interval of outages
- Imputing missing data
- Filter out sensor measurements to be within sensible operating limits
- Apply moving median filter to smooth out unstable sensor readings



vår energi

Previous Work

Numerical Modelling in GAP

Background

- Sr. Reservoir Engineer (ENI), Claudio Cannone, developed a static numerical simulation of individual branch flow contribution in rMLTs

Method

- Completion schematic simplified into 3 parts, one for each branch and one for the main wellbore
- By adjusting oil rates, Gas-oil ratio (GOR) and water cut (WCT), each parts pressures and temperatures are matched
- The new adjusted oil rates, GOR and WCT obtained are corrected for the MPFMs measurement error
- Obtain corrected comingled production flow rates for calculating the branch contribution ratios

Results

- Gap Model returns a branch split for a specific combination of choke and DIACS settings
- Model results only valid for present GOR, WCT, pressures, temperatur
- Water rates are not included

High mainbore gas MPFM measurement error

Wellbore / date	ibution			
	Oil Rate [%]		Gas Rate [%]	
	Mainbore (14 DIACS)	Lateral (2 DIACS)	Mainbore (14 DIACS)	Lateral (2 DIACS)
Mainbore MPFM error / 19th May 2024	96%	0%	216%	0%
Lateral MPFM error / 16th Jun 2024	0%	100%	0%	114%
Commingle Branch Split / 29th July 2024	**26%**	**74%**	**68%**	**32%**

Lateral flow contribution: 74% oil and 32% gas

Methods

Deep Learning Model

Graph Autoencoder (GAE)

- An autoencoder based on a graph neural network with node attention
- Branches communicate with the main wellbore unilaterally
- Nodes are defined with a set of features (column of data, e.g. pressure)
- Data is transformed to a latent space using a Graph Attention (GAT) encoder with various heads
- Data is transformed back to the input dimension with 3 additional features per branch using the Multilayer Perceptron (MLP) decoder
- Main Wellbore predictions are used for evaluation only
- Results are clipped to be > 0
- Predictions are processed in the loss function

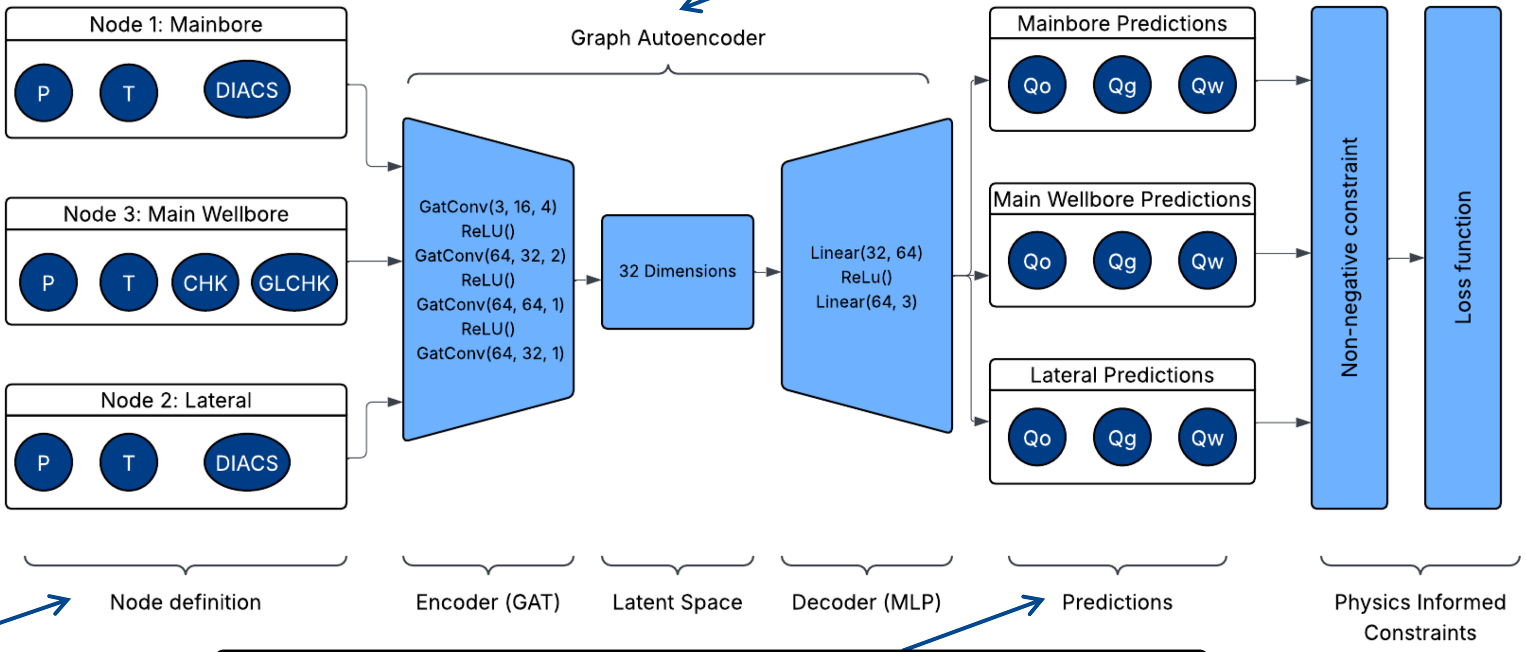
Give data to model

GAE version 2:

Check how good the predictions are

- Global data processing using additional linear transformations in the encoder
- Allows the mo a, at once

Perform some mathematical transformations to find a pattern in the data



Model returns a result based on the pattern it has found

vår energi

Loss Function

Learning

- The GAE model learns by minimizing the loss function and updating the model weights and biases (parameters)
- Defined as the mean squared error (MSE) between the actual (MPFM) and predicted using two different methods (1 & 2)

1. Semi-supervised target (Some answer exist):

- Branch flow is unknown, but the commingled flow is assumed to be accurate (MPFM)
- $Q_{MPFM} = \widehat{Q_1} + \widehat{Q_2}$
- *The model learns to predict a multiphase flow sum equal to the measured (MPFM)*

2. Supervised target (Answer exist):

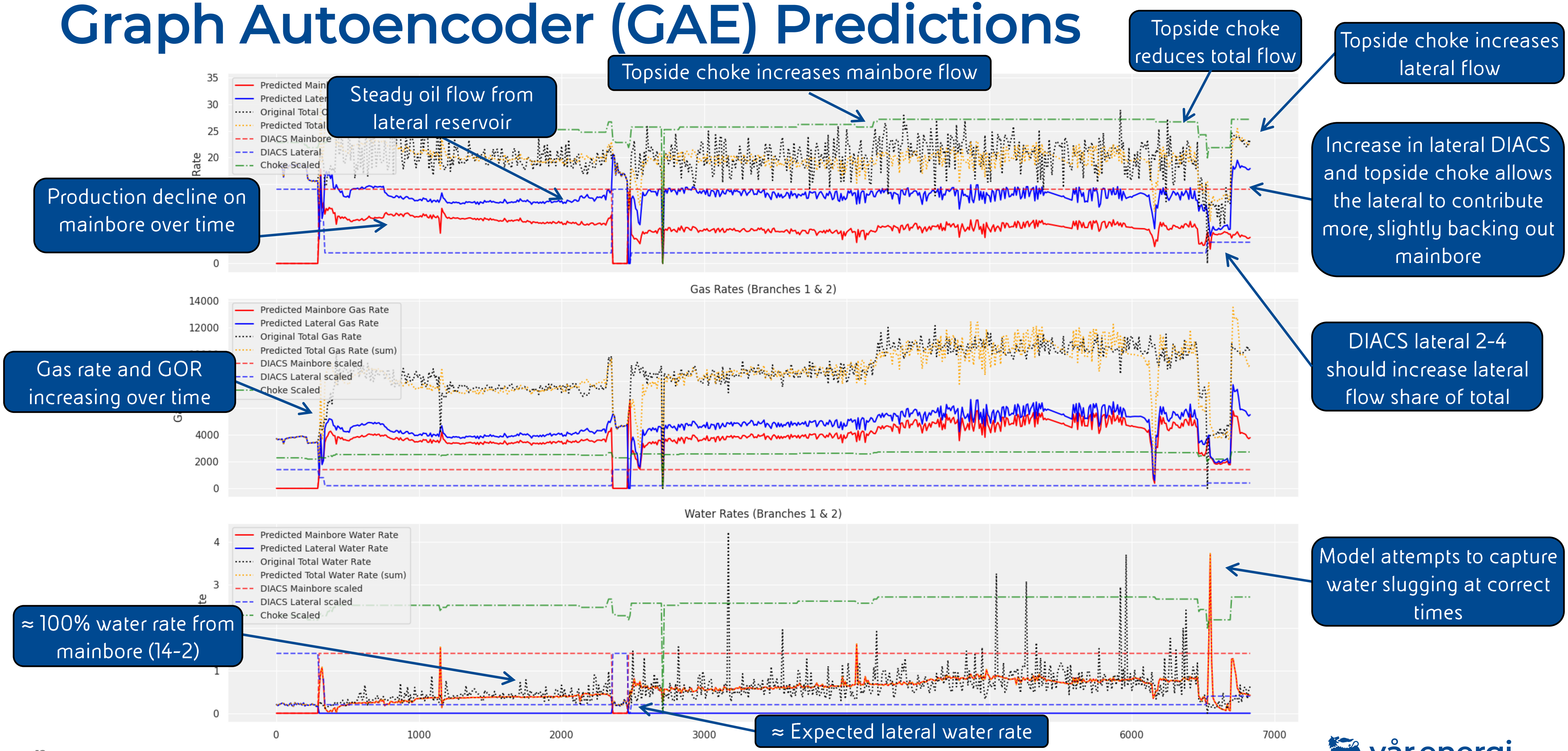
- If the lateral is closed: $Q_{MPFM} = \widehat{Q_1} + 0$
- If the mainbore is closed: $Q_{MPFM} = 0 + \widehat{Q_2}$
- *The model learns the contribution of each branch when one branch is producing*

```
for i in range(batch_size):
    km = k_main[i]
    kl = k_lat[i]
    dp_m = dp_main[i]
    dp_l = dp_lat[i]

    # Extract predictions from model
    o_b1, o_b2 = pred_oil_b1[i], pred_oil_b2[i]
    g_b1, g_b2 = pred_gas_b1[i], pred_gas_b2[i]
    w_b1, w_b2 = pred_water_b1[i], pred_water_b2[i]

    # Get targets labels from actual measured MPFM
    t_o, t_g, t_w = target_oil[i], target_gas[i], target_water[i]

    # 2. Supervised loss (one branch is closed)
    if km == 0:  # Mainbore closed -> all flow through Lateral
        total_loss += ( # MSE loss
            loss_fn(o_b2, t_o) + loss_fn(g_b2, t_g) + loss_fn(w_b2, t_w)
        )
    elif kl == 0:  # Lateral closed -> all flow through Mainbore
        total_loss += ( # MSE loss
            loss_fn(o_b1, t_o) + loss_fn(g_b1, t_g) + loss_fn(w_b1, t_w)
        )

    # 1. Semi supervised learning (both branches flow)
    else:
        # MSE loss on total flows
        o_total = o_b1 + o_b2
        g_total = g_b1 + g_b2
        w_total = w_b1 + w_b2

        mse_loss_total += (
            loss_fn(o_total, t_o) + loss_fn(g_total, t_g) + loss_fn(w_total, t_w)
        )
```


vår energi

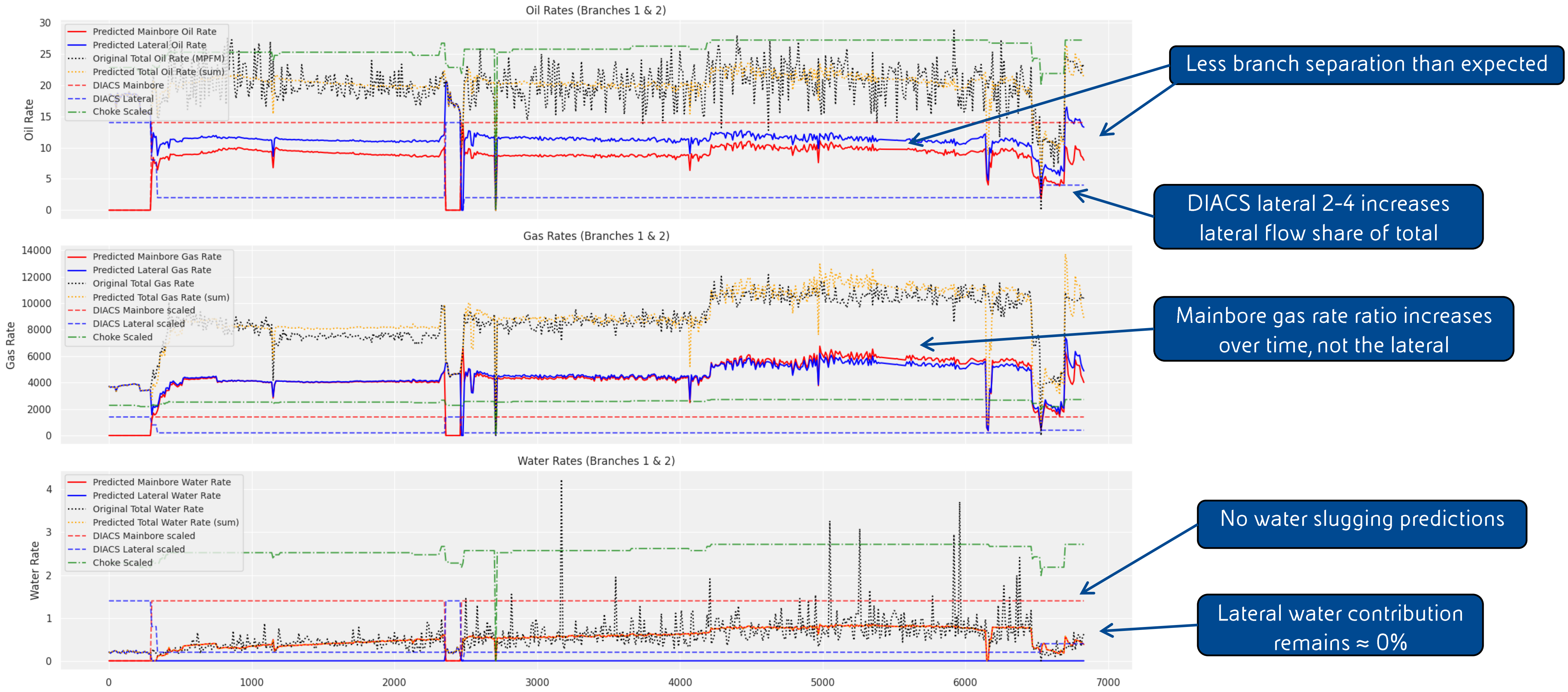
Results

Graph Autoencoder (GAE) Predictions



vår energi

Graph Autoencoder (GAE) Predictions – Version 2



vår energi

Evaluation

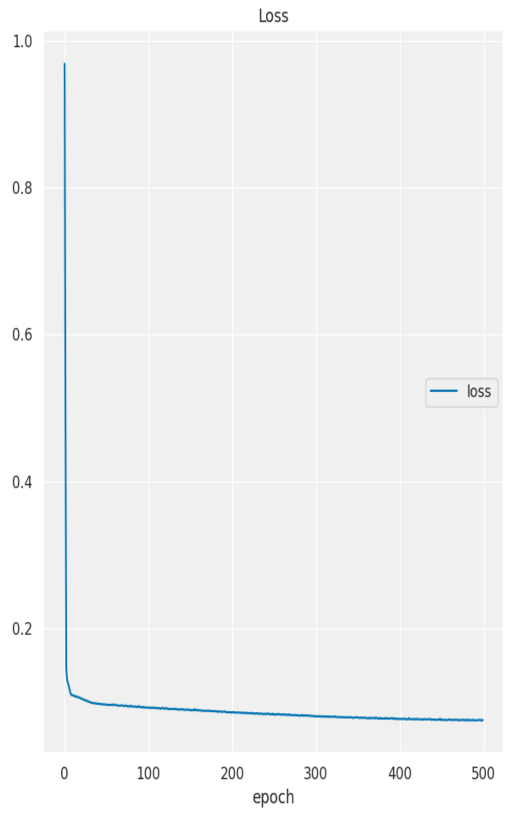
Graph Autoencoder (GAE) Metrics

Loss function MSE (Mean Squared Error)

- Smooth and decreasing loss curve with MSE: 0.968 to 0.075
- Proves the model is guided towards the correct solution
- Efficient loss function 1 and 2

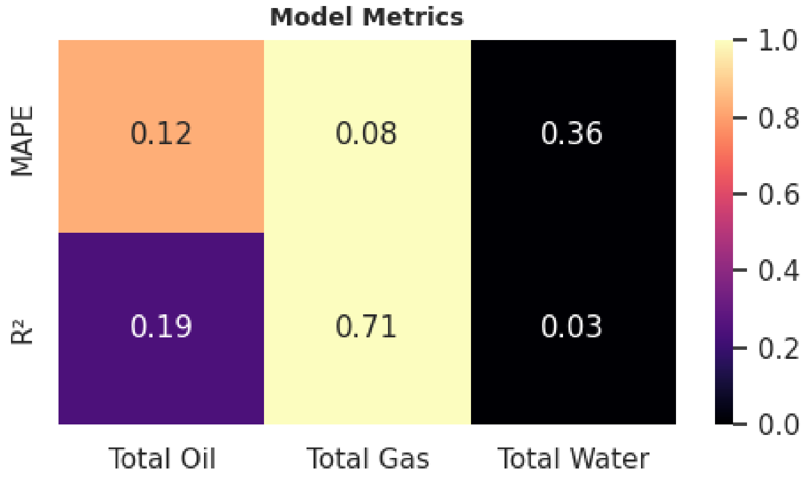
Contribution Ratios

- Lateral contribution at (14-2): 64% Oil and 53% Gas
- Aligns with GAP model results although GAE predicts higher gas and less oil.



Evaluation Scores

- Output reconstruction error: 1.11% (MPFM prediction)
- Good mean average percentage error (MAPE) for oil and gas, poor for water
- R^2 score shows the model fails to capture fluctuations in the oil and water rates
- MAPE and R^2 calculated using sum of branch flow



vår energi

Discussion

vår energi

Model Performance

GAE model sensitivity

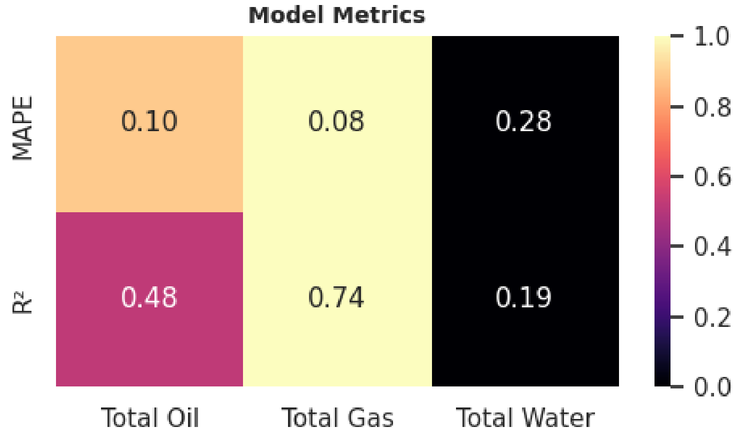
Best results obtained with:

  - 8e-5 step learning rate / 64 hidden dim / 32 dim latent space / 128 batch size

- Changes in model hyperparameters caused:
  - Even branch split ratio
  - Overfitting/underfitting
  - No mainbore production decline predicted
  - Less responsive to changes in DIACS chokes
- Model is struggling to predict > 0 for all phases – a frequent occurrence during development
- Training on multiple wells caused more responsive results

GAE Version 2 – global processing effects:

- More even branch split (55% lateral oil rate at 14-2)
- Improved responsiveness to chokes
- Improved metrics
- Reconstruction error: 0.7%
- Indicates a trade-off between correct branch ratio, or better responses to choke settings



Conclusion

Conclusion

Data quality and Preprocessing

- The results have been highly sensitive to the quality of the training data and preprocessing, as expected
- Even a small training set size (4 wells) with little variation generated valuable predictions for oil and gas flow

Modelling

- Node learning and attention mechanisms has proven to be effective for predicting branch flow
- Additional global processing in the encoder is beneficial, but reduces branch separations
- All model versions have been highly sensitive to the hyperparameters, and features chosen for training

Evaluation

- Predicting branch flow without labels adds uncertainty to the evaluation methods (no real answer available)
- Existing GAP results comparison and petroleum production analysis have been the most important evaluation methods

Usage

- The GAE branch flow predictions provide a dynamic insight into multilateral well performance and can be used for reservoir management and production optimization
- The next step will be to investigate zero valued water phase predictions for the lateral branch, compensate for MPFM measurement errors and to create a generalized model capable of predicting on unseen data

Acknowledgement

I would like to thank,

Main Supervisor & Mentor: Marius Stene - Principal Petroleum Engineer at Vår Energi

Second Supervisor: Dr. Samuel Krevor - Professor of Subsurface Carbon Storage at Imperial College London

Support and Guidance: Erik Brøndbo – Senior Principal Petroleum Engineer at Vår Energi

Support and Guidance: Kristoffer Sæby – Senior Petroleum Engineer at Vår Energi

Support and Guidance: Mohammad Sohrab Hossain – Senior Petroleum Engineer at Vår Energi

Gap Model: Claudio Cannone – Senior Reservoir Engineer at ENI

EWPF Stavanger, August 2025

vår energi

Q&A

References

1. *Khemani, B., Patil, S., Kotecha, K. and Tanwar, S., 2024. A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions. Journal of Big Data, 11(1), p.18.*

2. *Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P. and Bengio, Y., 2017. Graph attention networks. arXiv preprint arXiv:1710.10903.*

3. *Sandnes, A.T., Grimstad, B. and Kolbjørnsen, O., 2021. Multi-task learning for virtual flow metering. Knowledge-Based Systems, 232, p.107458.*

4. *Bikmukhametov, T. and Jäschke, J., 2020. First principles and machine learning virtual flow metering: a literature review. Journal of Petroleum Science and Engineering, 184, p.106487.*

5. *Claudio Cannone's work on GAP model for branch contributions (Unpublished)*

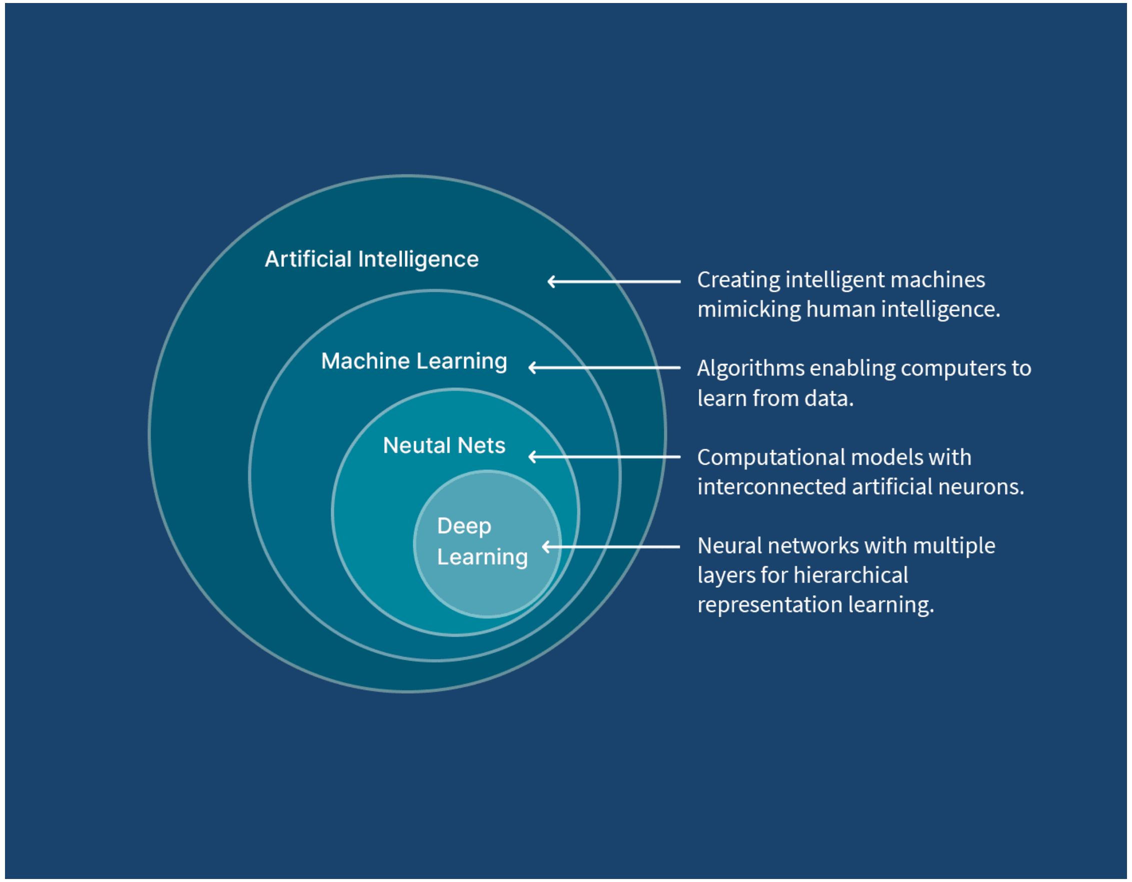
6. *SLB's work on the Bernoulli's equation for retrofitted MLT's (unpublished)*

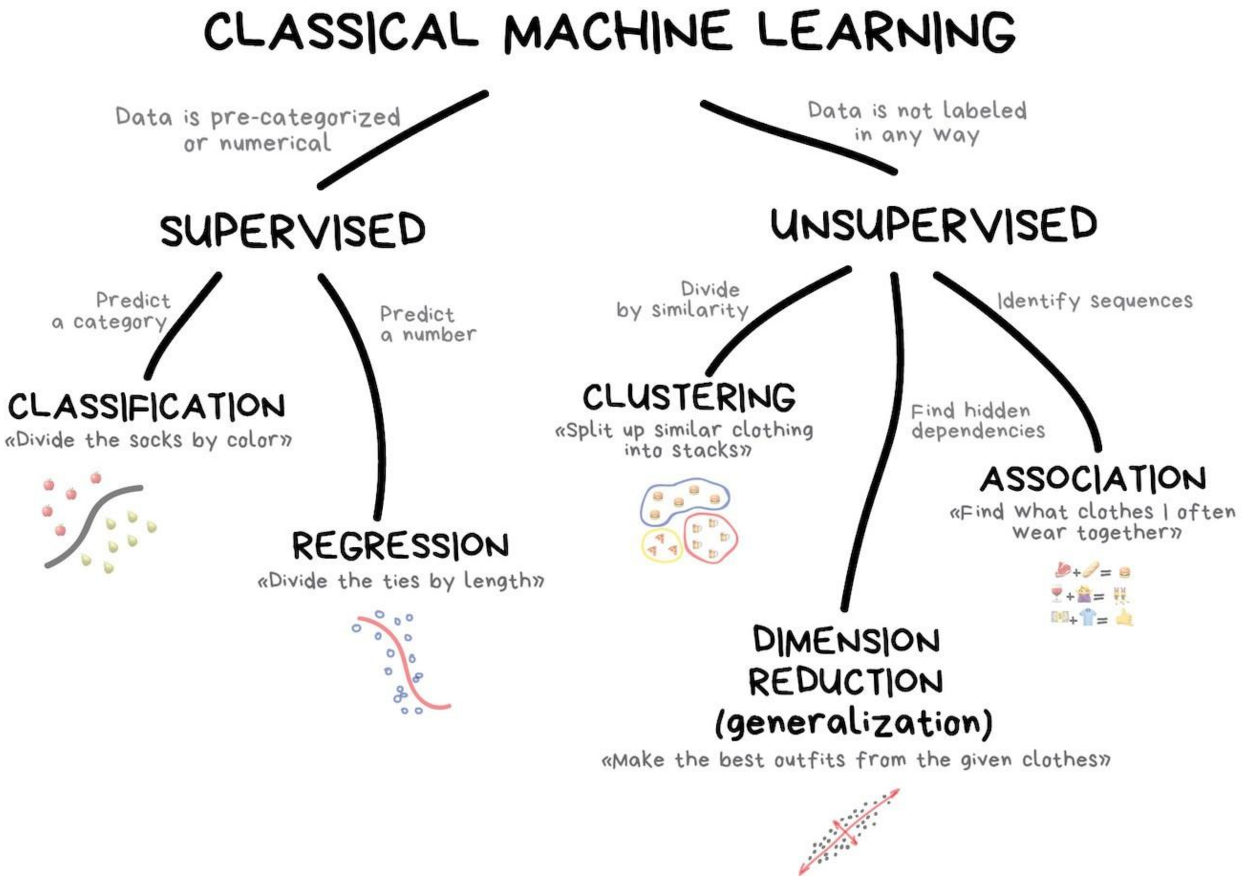
vår energi

Appendix

vår energi

Machine Learning Definition

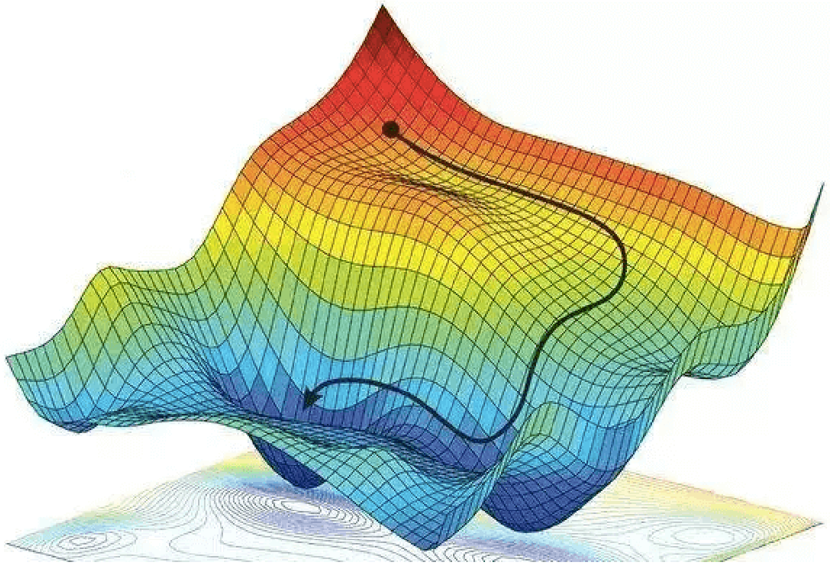




Loss function – How does the model learn?

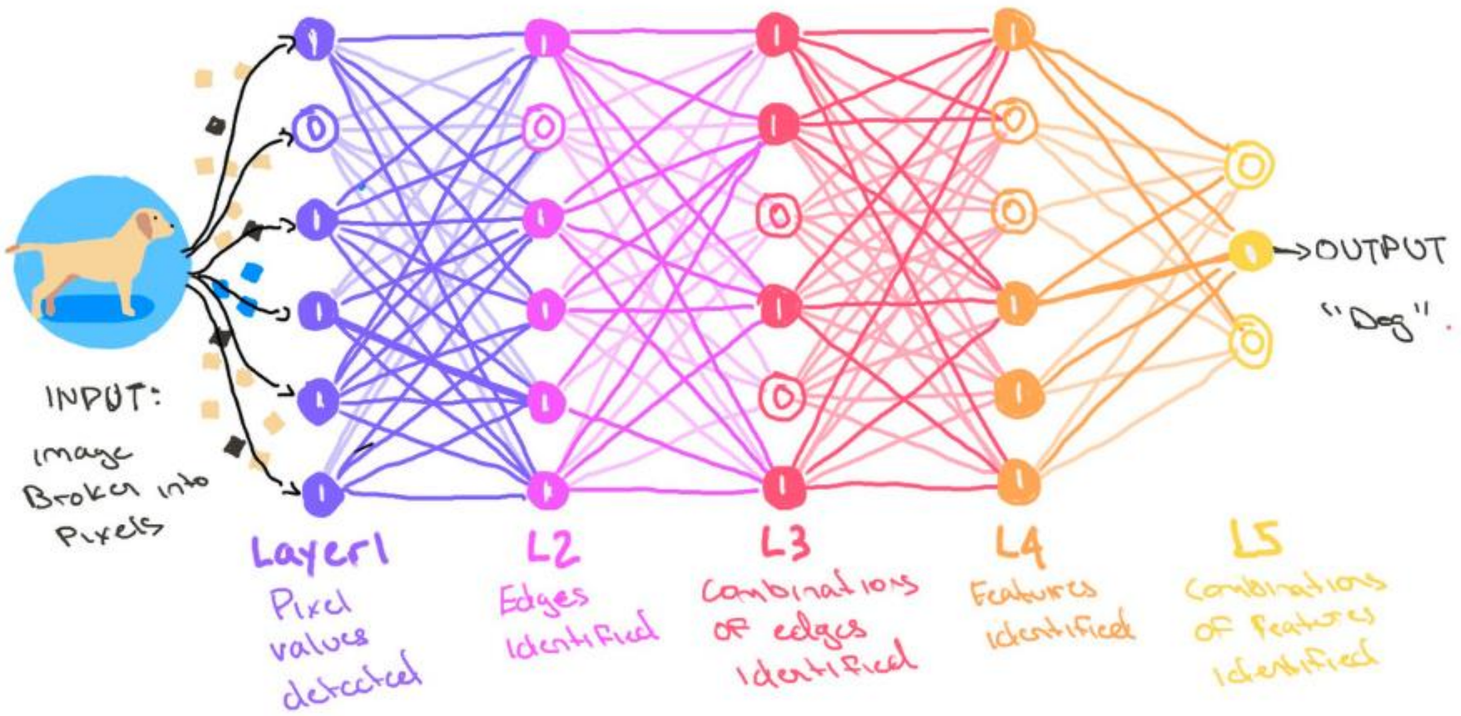
Gradient Descent

- Calculates the gradient (direction) to find the "road" to the lowest MSE in the loss function (loss space)
- If MSE increases, it adjusts its weights and biases to find a new "road"
- If MSE decreases, it adjusts its weights and biases to continue down this "road"
- Finally, the MSE can not be reduced further, and the model has been trained with the optimal weights and biases

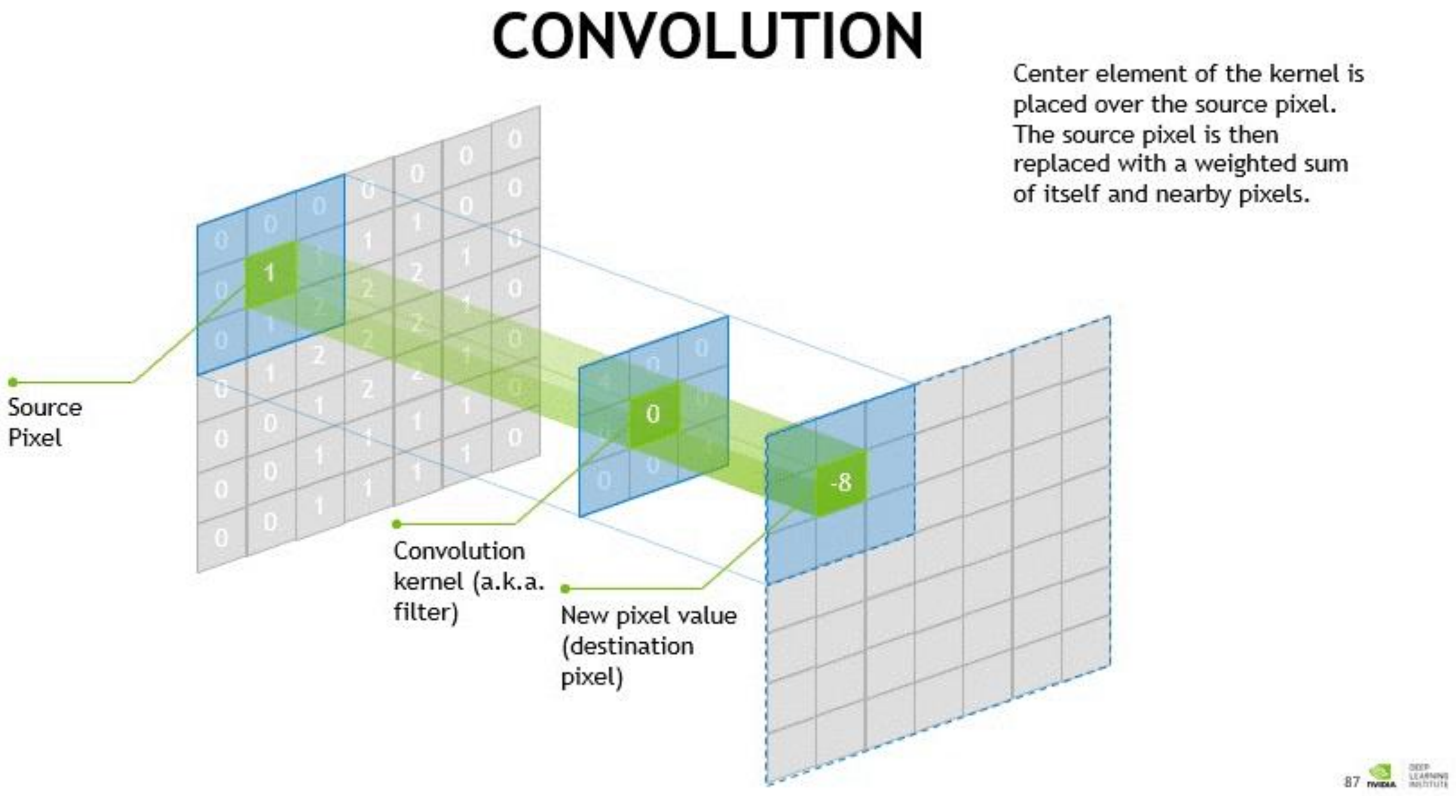


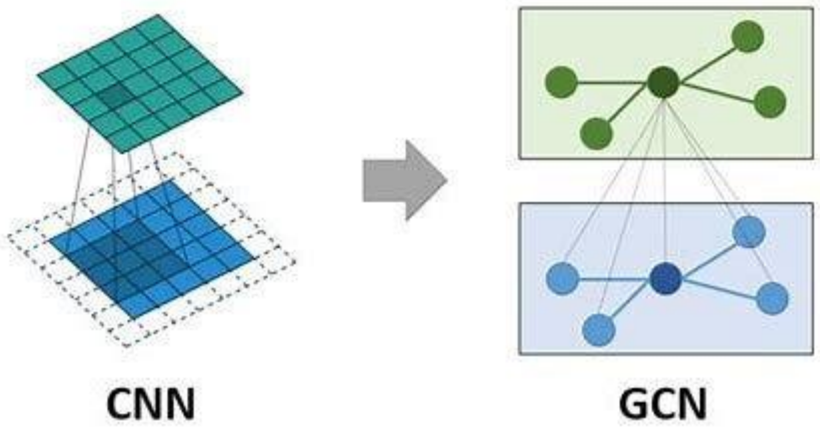
vår energi

Neural Networks – What does the model learn?



Convolutional Transforms



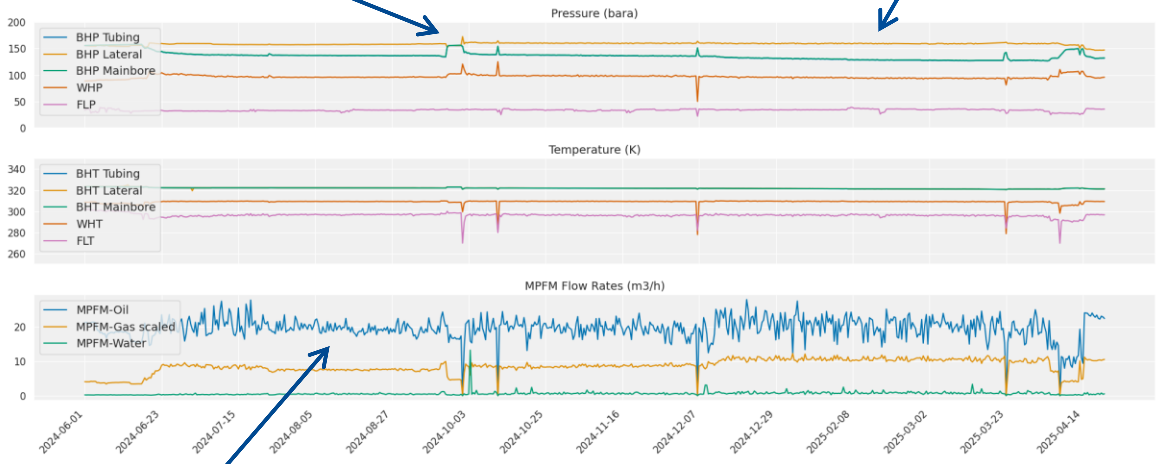


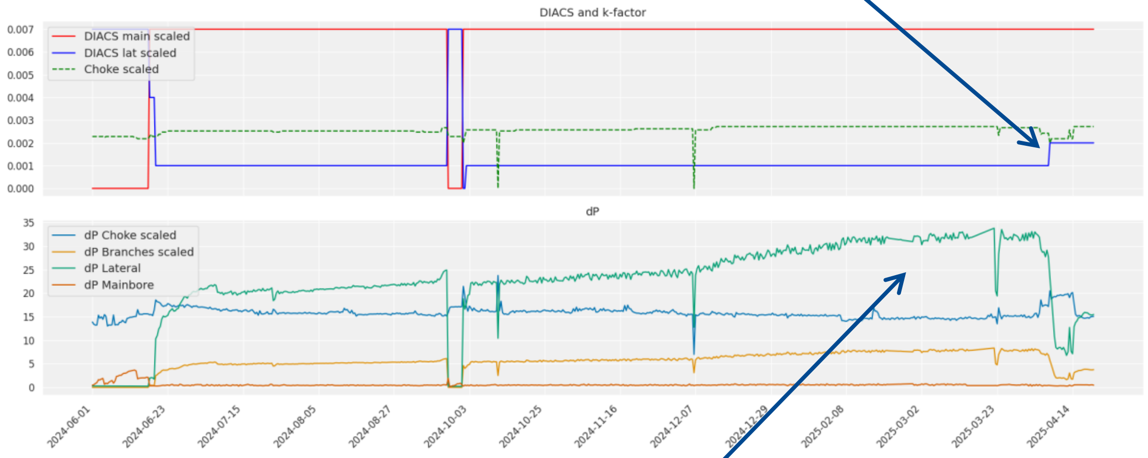
Cleaned Data

Sensor readings clipped to sensible operating range

Sensor outages manually removed and filled using last value

Changes in DIACS still scarce





Curve smoothed with median average to dampen outliers

Result of forward filling pressure

Exploratory Data Analysis

Flow Correlations

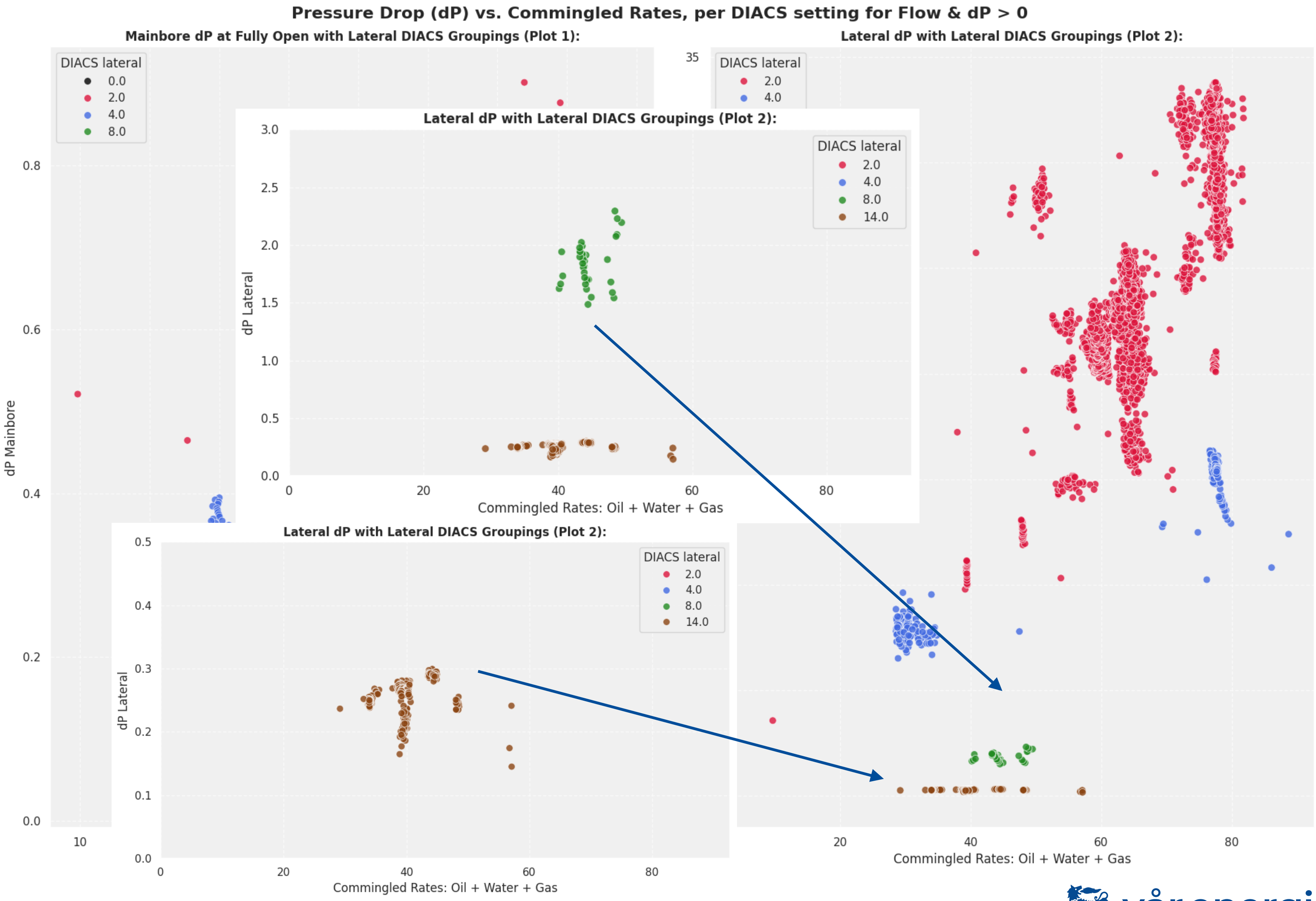
- Expecting quadratic correlations
- Combined, dP correlates quadratic with flow rates, approximately
- Wide range of measurements at DIACS 2, displaying clear correlation
- DIACS 4 jump due to topside choke adjustments
- Measurement errors, outliers, potential complex flow patterns prevents perfect quadratic curve fit
- Individual oil, gas and water rates displayed varying, caotic or no correlations

Plot 1:

- DIACS 8 almost no correlation due to few measurements

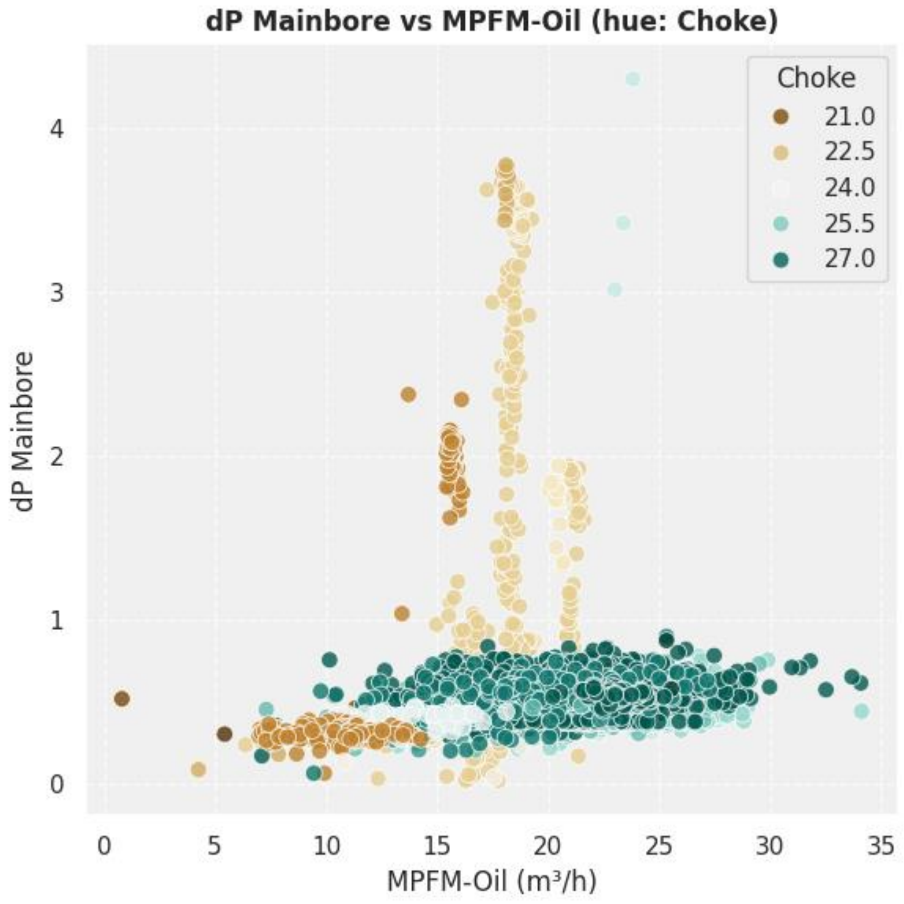
Plot 2:

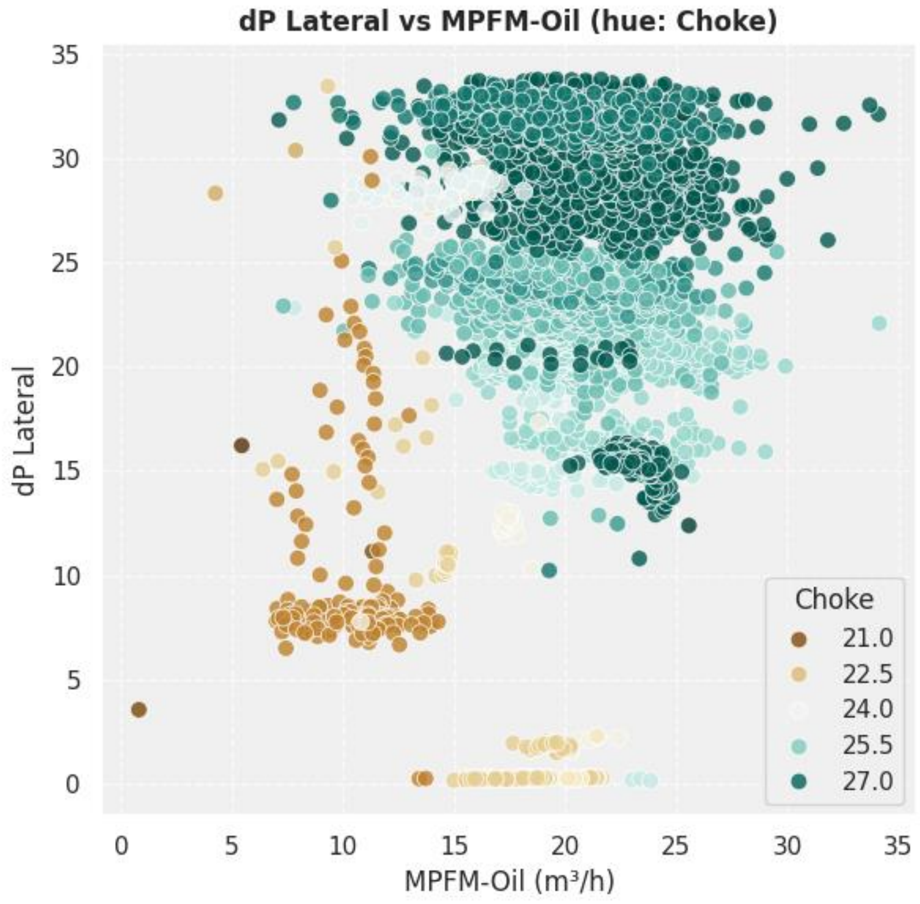
- Some correlation at DIACS 8
- DIACS 14 distorted as topside choke increases and sensor error present



vår energi

EDA: Choke Correlations





vår energi

Bernoulli's

Sandnes et al., (2021) derived an analytical model from Bernoulli's equation:

$$Q = AC\sqrt{\frac{P_1 - P_2}{\rho}}$$

Anders T. Sandnes, Bjarne Grimstad, and Odd Kolbjørnsen. Multi-task learning for virtual flow metering. Knowledge-Based Systems, 232:107458, 2021.

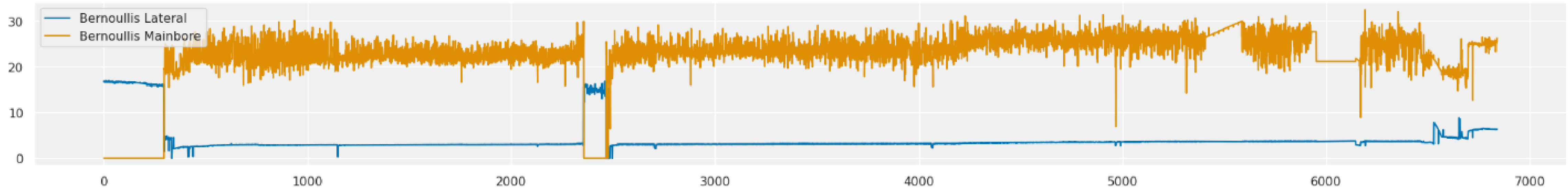
A – effective valve opening | C – Coefficient | P – Pressure | ρ – fluid density

SLB derived an analytical model for Bernoulli's, fitted to DIACS branch valve openings (unpublished):

$$Q^2 = N\frac{C_d A^2}{\rho}(P_1 - P_2)$$

N - number of nozzles | A - Area of Nozzles | C_d - Coefficient | P - Pressure | ρ – fluid density

- Multiple assumptions are made: Uniform velocity profiles, horizontal, inviscid, steady and incompressible single-phase flow
- Flow control chokes, multiphase flow behavior, friction forces and commingled flow makes these assumptions invalid
- The well of interest is a special case - Bernoulli's results are poor due to comparing extremely restricted DIACS with fully open, and pressure sensor errors
- Bernoulli's in the loss can work as a regularizer, for some wells, to guide the model to learn which branch should flow more



vår energi

Preprocessing

Transformation methods:

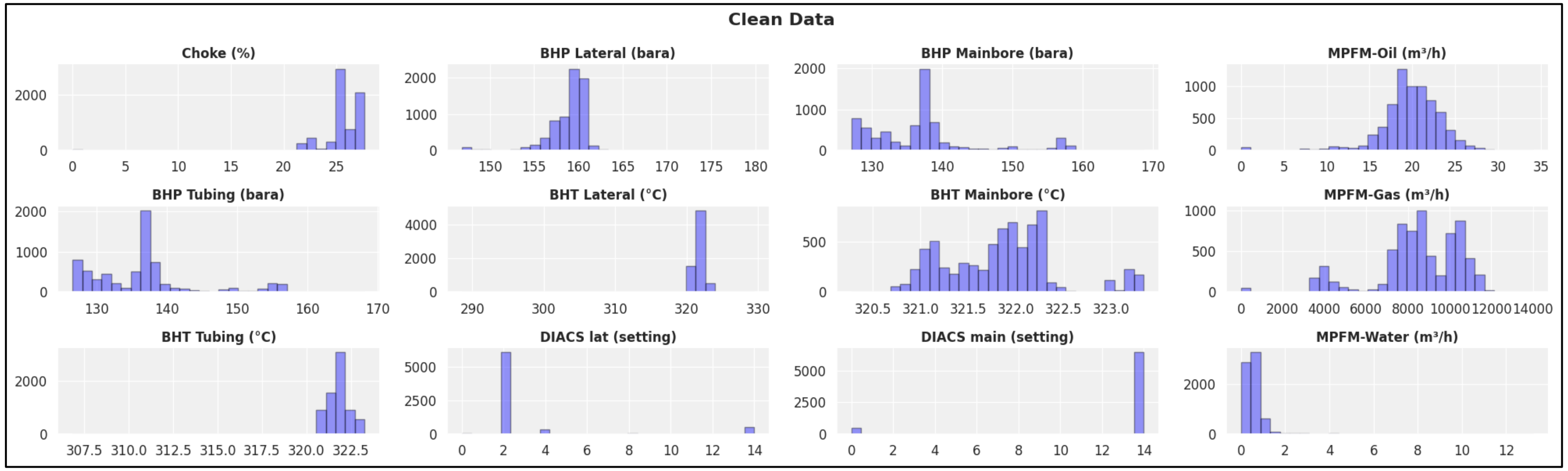
- Log transform non-gaussian features
  - Excluding MPFM-Oil, MPFM Gas, DIACS, Choke and k-factor
- Apply min-max scaler:

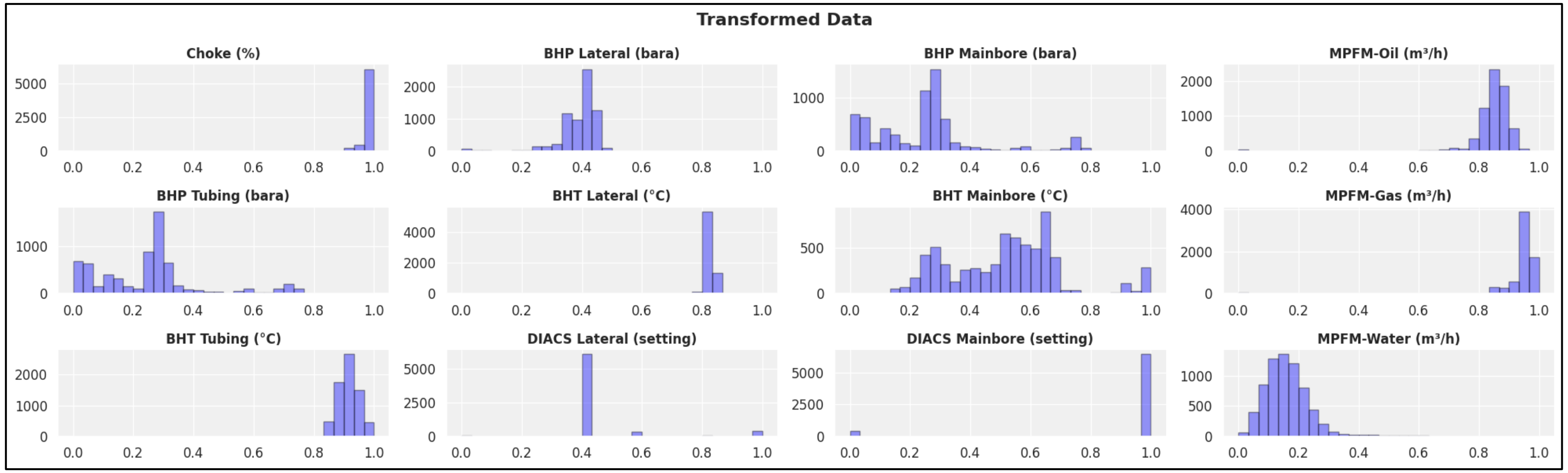
$$x_{scaler} = \frac{(x - x_{min})}{(x_{max} - x_{min})}$$

- Provides faster learning and improved predictions

Data assumptions:

- No sensor measurements error
- No sensor drifting

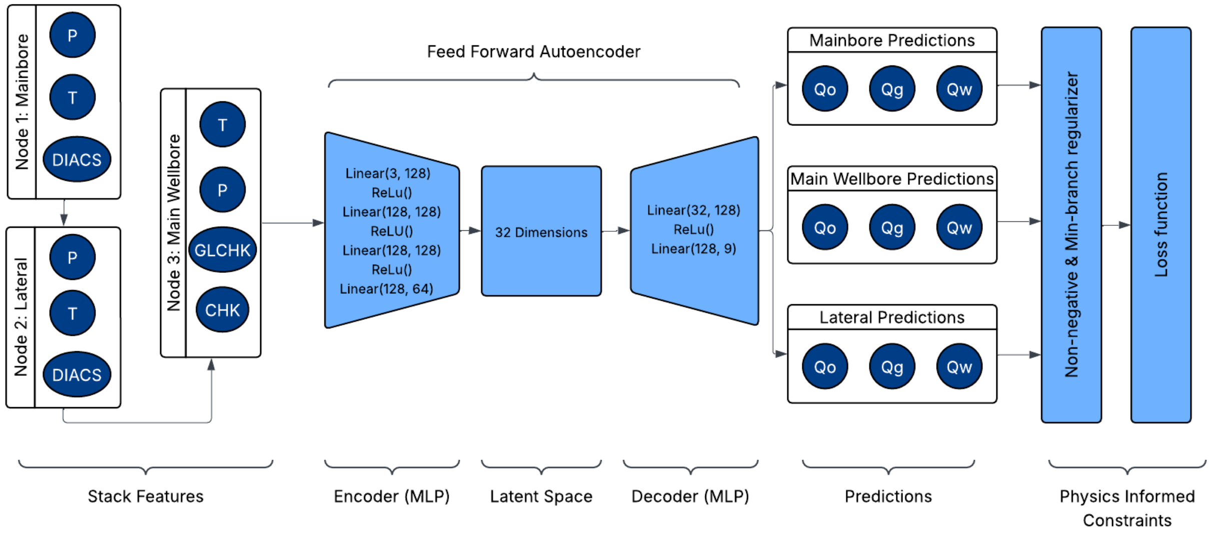




Deep Learning Model

Feed Forward Autoencoder

- A simpler model, FFA, was developed for comparison
- Stacked features for global processing
- Ignores relationships between nodes
- FFA contains a 3 layered MLP in the encoder, and the same decoder




```python
# Minimum ratio penalty (k > 0)
min_ratio_loss = 0.0
if min_ratio_lambda > 0:
    both_open_mask = (k_main > 0) & (k_lat > 0)  # [batch_size]
    if both_open_mask.sum() > 0:  # Only if both valves open
        open_flows = flows[both_open_mask]  # [num_open_samples, 3, 3]
        total_flows_open = open_flows[:, 0, :] + open_flows[:, 1, :] + 1e-8  # [num_open_samples, 3]

        # Calculate proportions
        prop_b1_open = open_flows[:, 0, :] / total_flows_open
        prop_b2_open = open_flows[:, 1, :] / total_flows_open

        # Apply penalty
        penalty_b1 = F.relu(min_ratio - prop_b1_open)
        penalty_b2 = F.relu(min_ratio - prop_b2_open)
        min_ratio_loss = torch.mean(penalty_b1 + penalty_b2)
```

Minimum branch contribution

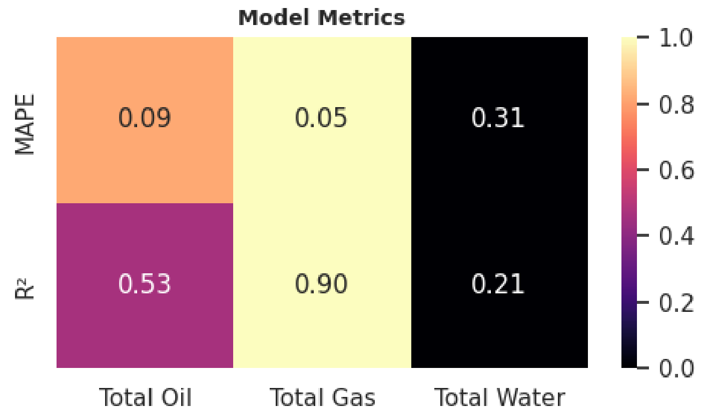
- Model cheats by predicting one branch to be 0
- Minimum branch flow is enforced to avoid predicting commingled flow

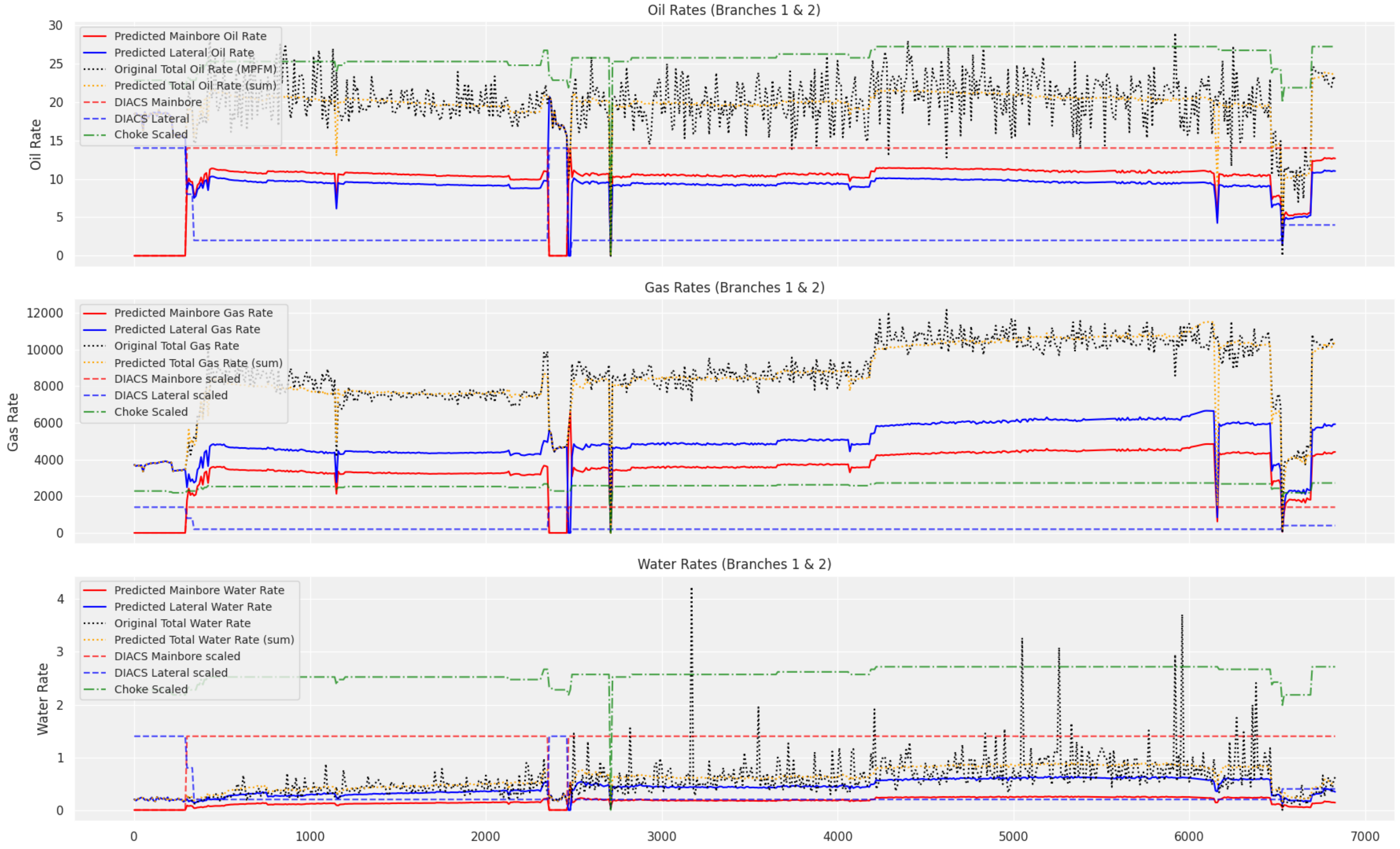
vår energi

Feed-Forward Autoencoder

Comparison

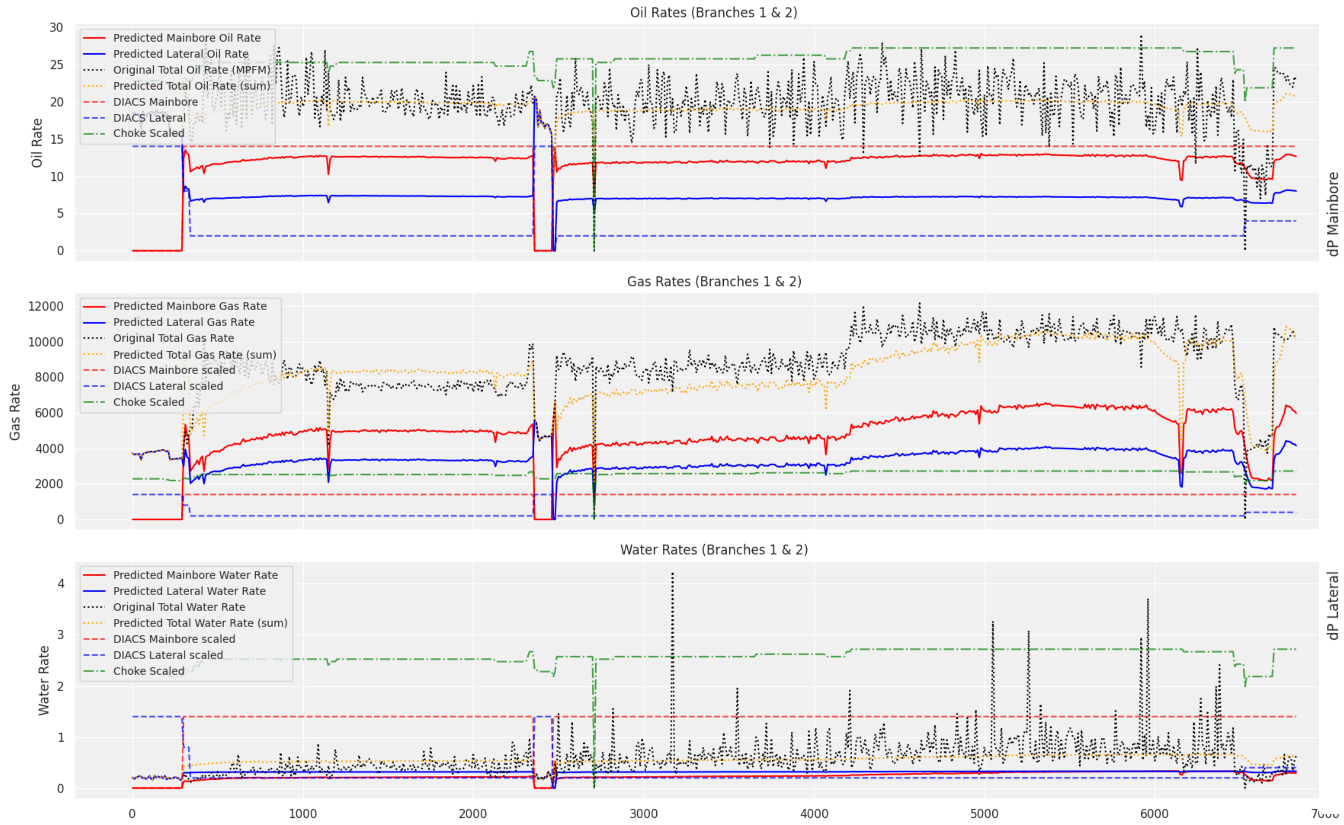
- Predicts unexpected rates
- No mainbore production decline predicted
- Poor response to Lateral DIACS increase
- Topside choke adjustment causes equal branch rate change
- Improved metrics
- Water rate branch predictions > 0

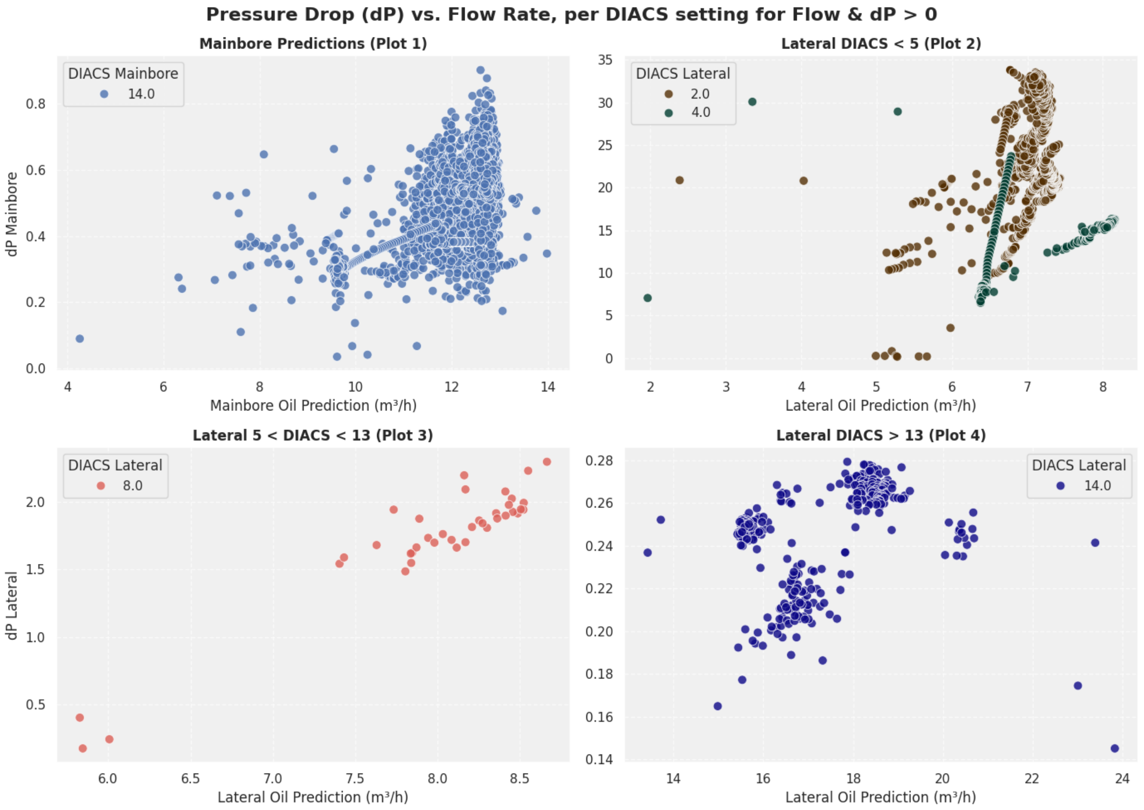




vår energi

GAE – Training on one well

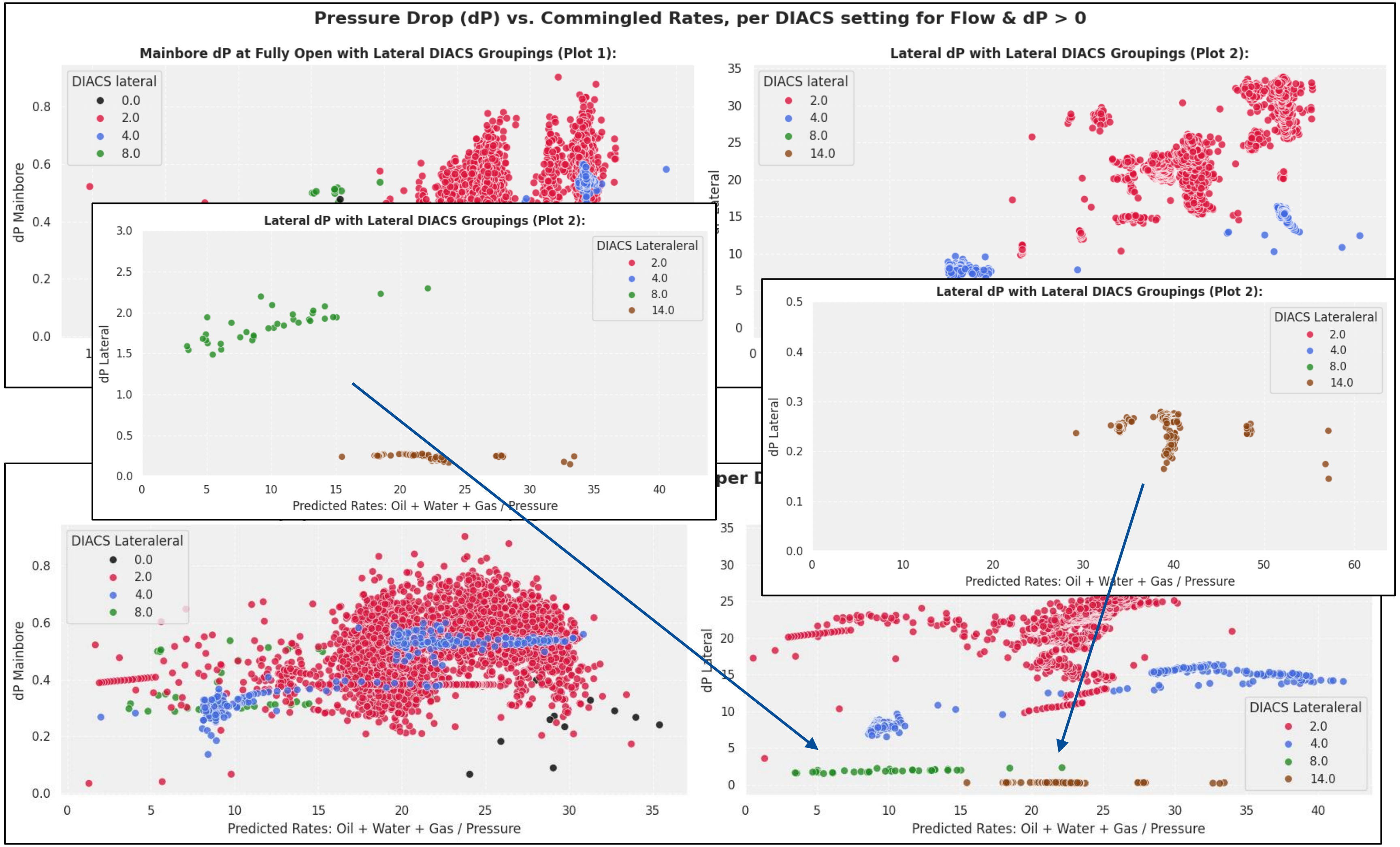




Graph Autoencoder (GAE) Flow Correlations

Flow Correlations

- The predicted rates generates dispersed correlations
- Correlation patterns have changed, but remains approximately linear/quadratic
- Stronger correlation on DIACS 8
- Predicting commingled at DIACS 14



vår energi