PREDICTING AIRPLANE WEIGHT AND BALANCE

Data provided by: A-ICE Aviation Information and Communication Engineering

Presented by: Tommaso Ferrari (xbl574)
Alicia Mosquera Rodríguez (dlx861)
Andreas Malthe Faber (qzj517)

The Database: Inputs

The database contains data referring 64’000 flights to and from the MPX airport in Milan.

Most of the features are various parameters that indicate weights and centers of gravity in commercial and passenger flights. To cite a few of the most important examples:

- DOW_W/I: Dry Operating Weight, weight/index (float that expresses COG)
- TOW/I: TakeOff weight/index
- PAX_W/I: Passenger weight/index
- Dist: Distance of the flight

The Database: Outputs

There are four features that we are trying to predict with our machine learning algorithms:

- ZFW/I: Zero Fuel, weight/index ⇨ These are the ideal values of weight and center of gravity to maintain the ideal flight attitude, which reduces wear on the aircraft.
- Blockfuel/Burneoff: Weight of fuel at takeoff/Weight of consumed fuel ⇨

  Fuel is one of the main expenses for an airline

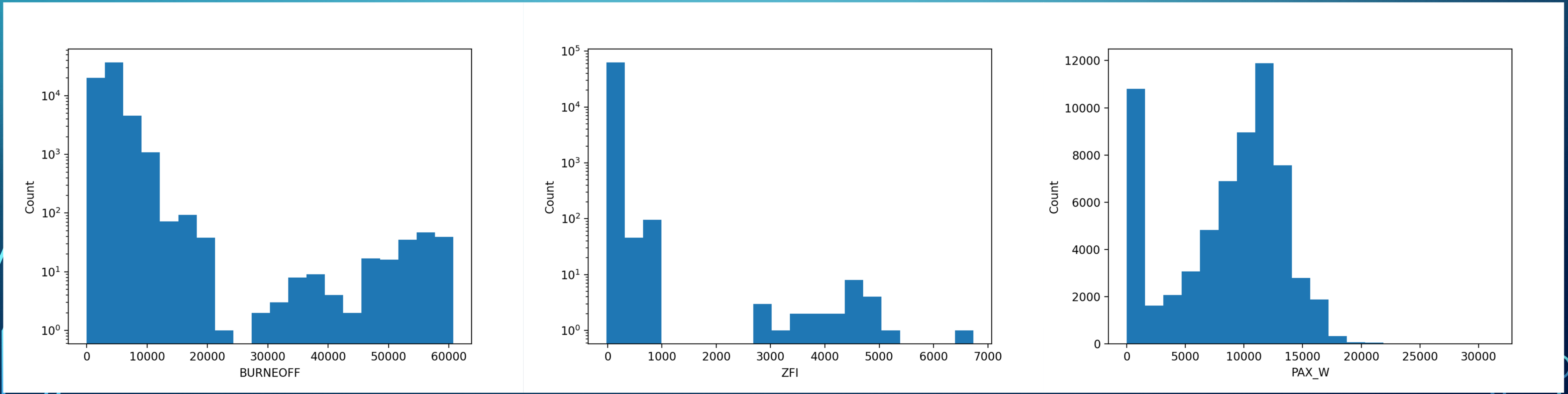
Pre-Processing

Categorical Inputs: ACtype, Config, Version.

- ACtype and Config turned into integers ⇨ Ordinal Encoder
- Version divided into 3 integers (NaNs as well): F (first class), C (club), Y (economy)

Normalization of non-categorical data (input and prediction):

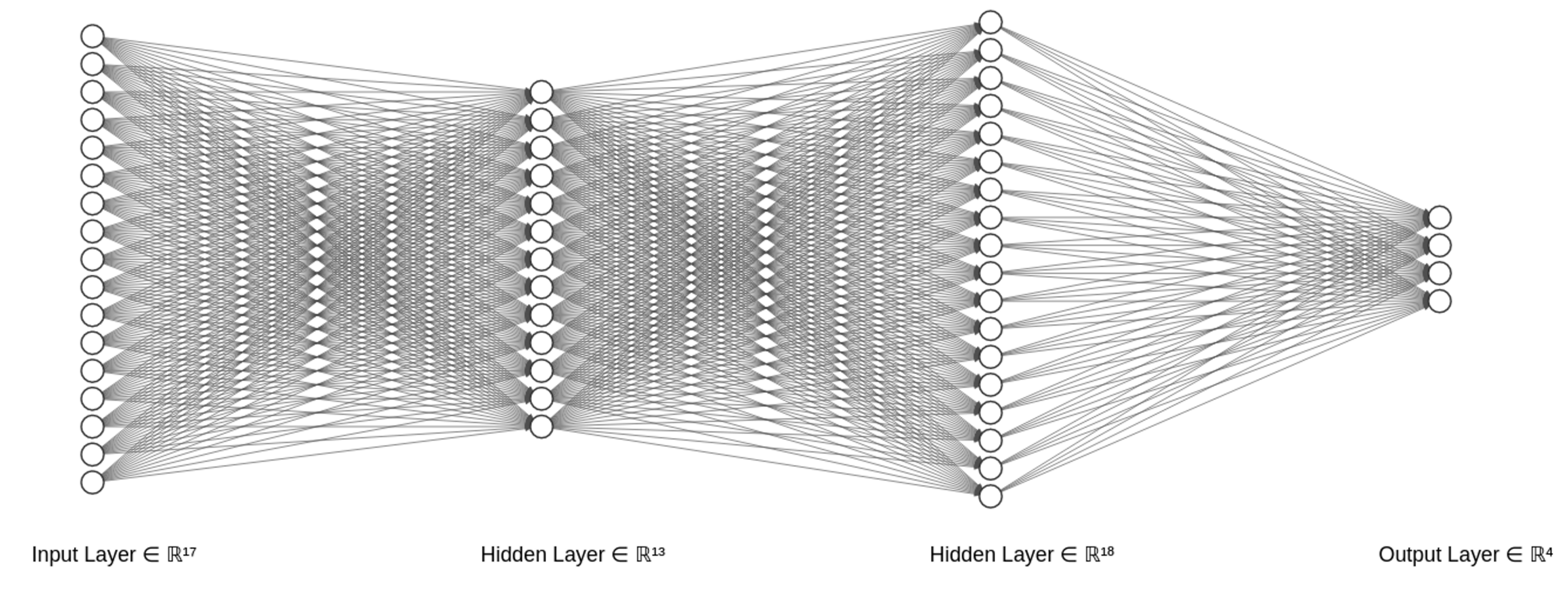
- Multi-peak data ⇨ Quantile range 10/90%
- Scaled * 200 (to match categorical)



Neural Network Structure

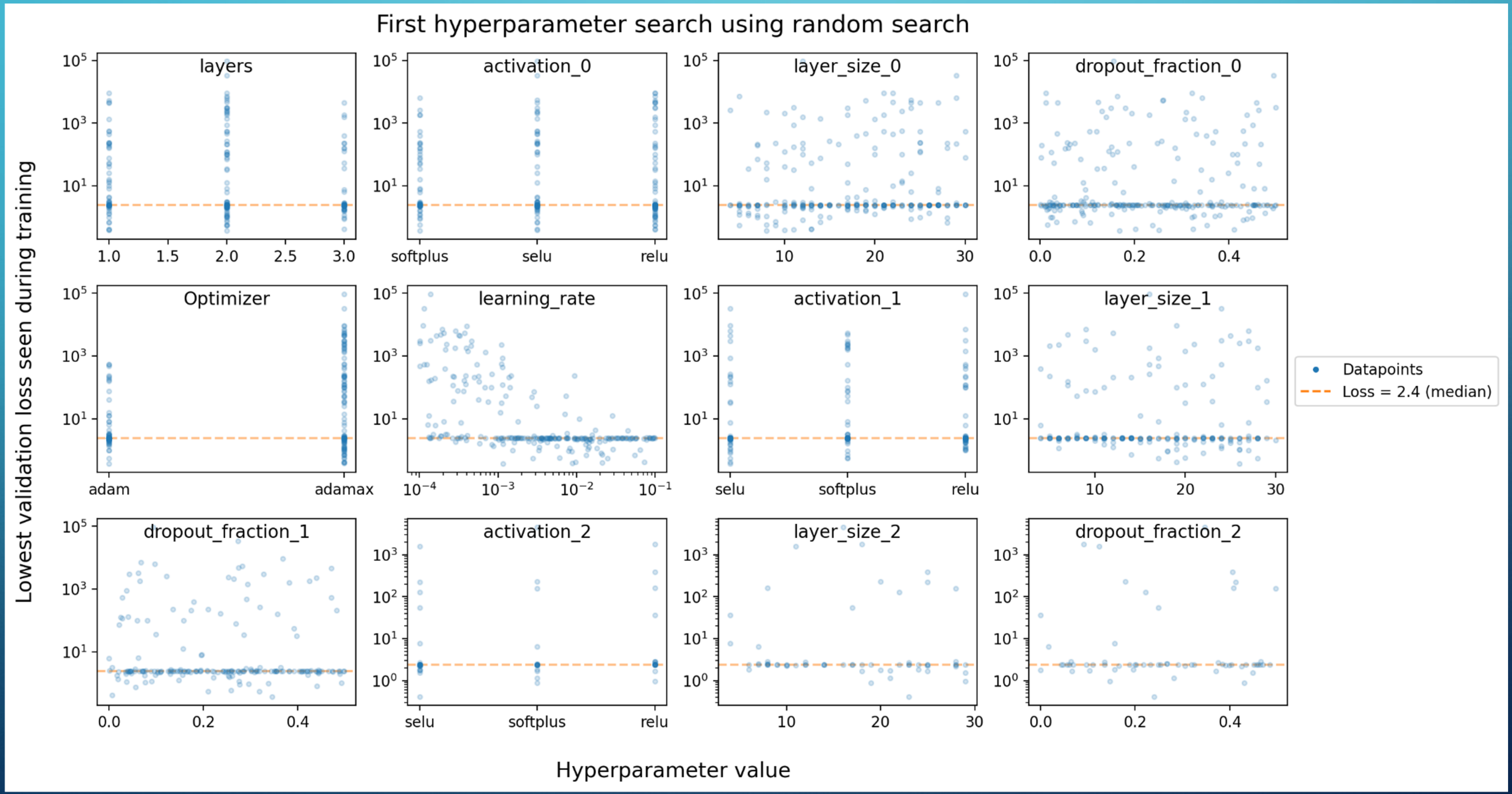
Input Layer: 17 inputs
Number of Dense layers: 1-3
Layer Size: 4-30
Dropout: 0-0.5
Output Layer: 4 regressions

Activation: relu, softplus, selu
Optimizer: Adam, AdaMax
Learning rate: 0.0001-0.1 (log sampling)
loss=MSE



Results of First Hyperparameter Search

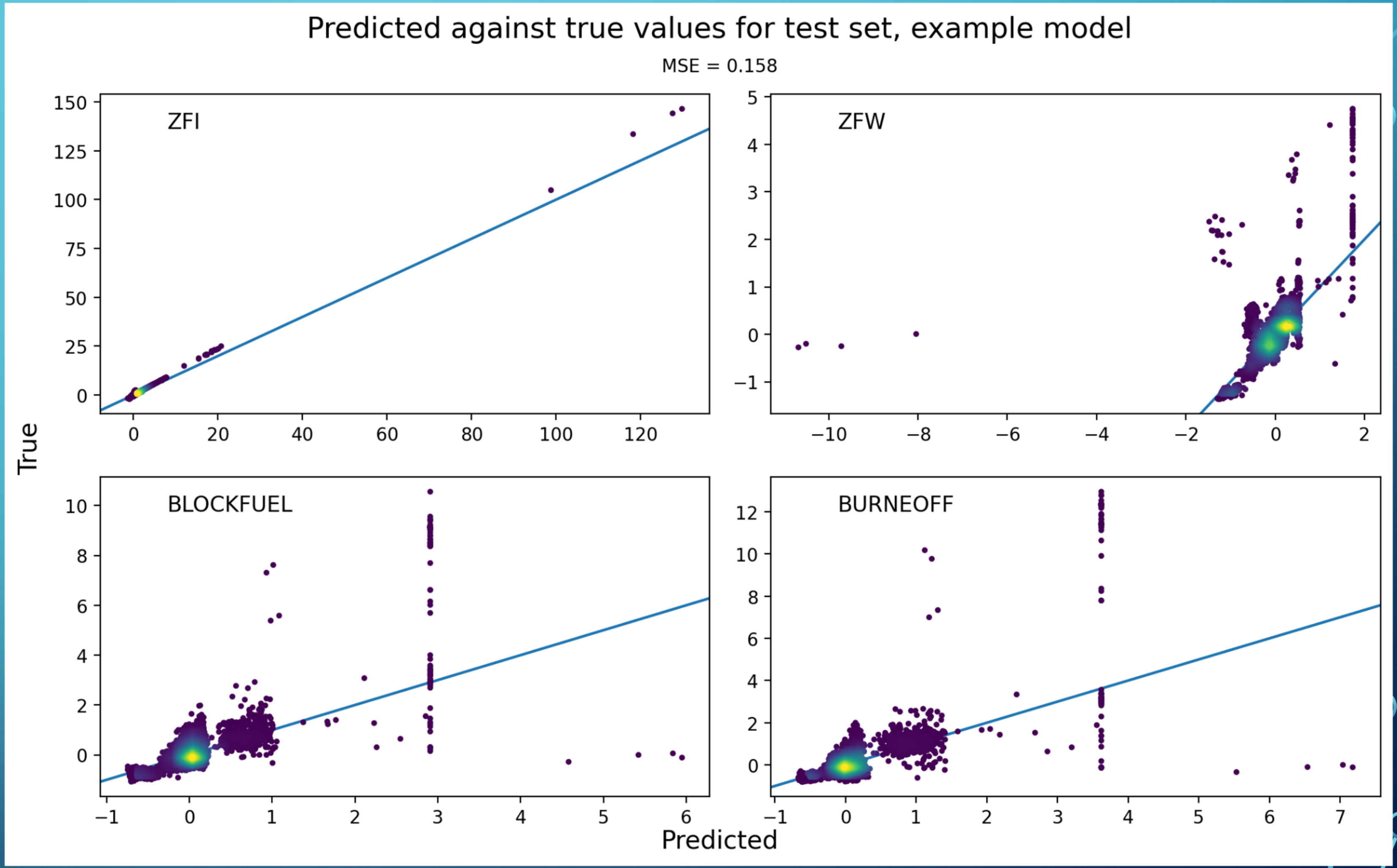
- A random search of 250 trials, training for 20 epochs max but with early stopping and a patience of 4 epochs



Example Model

- After finding good HP candidates in the search, models were trained with cross validation

- Even when the MSE wasn't terrible, many of the networks struggled severely with the outliers



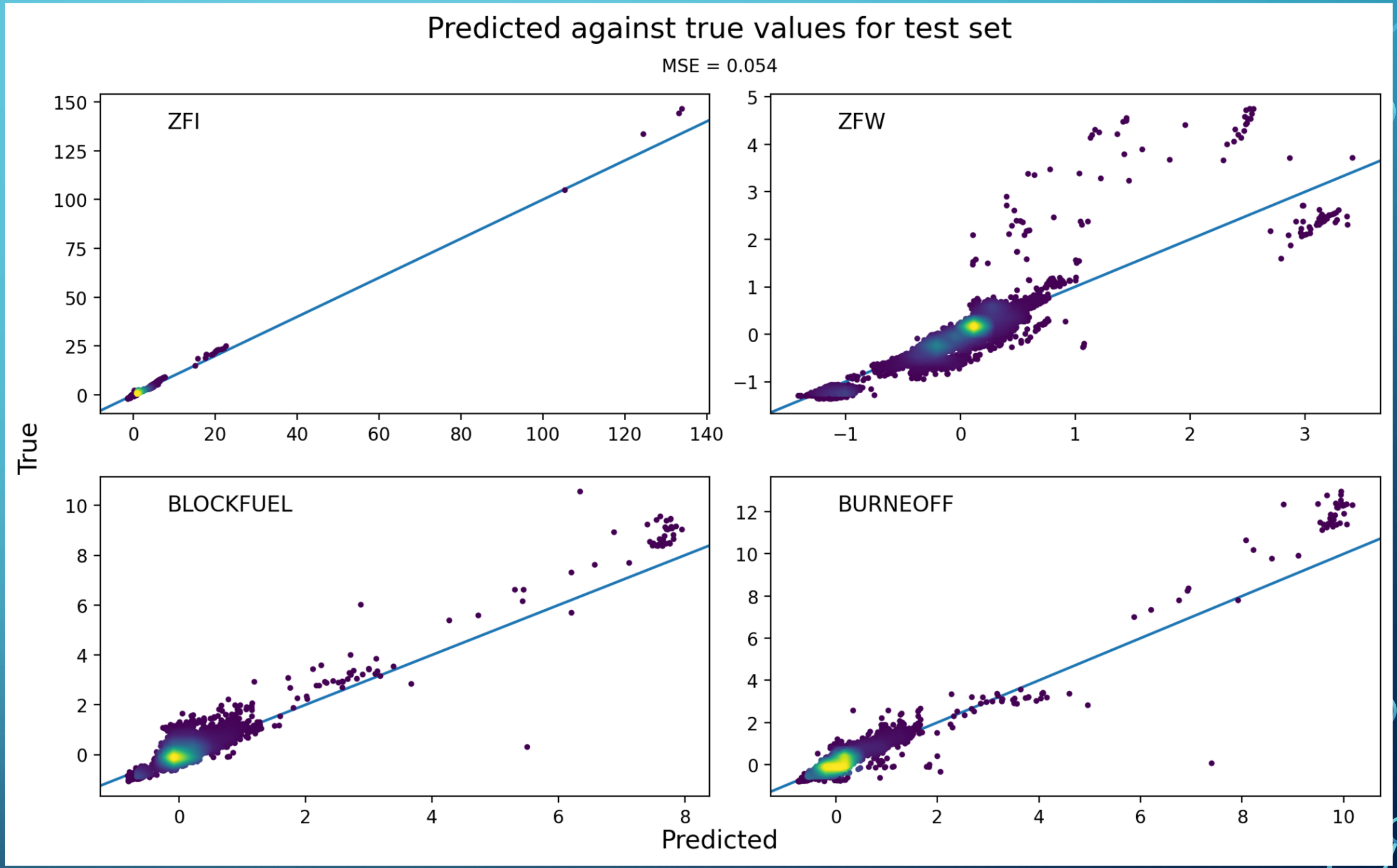
Further Searches and Training

- 3 total further HP searches were performed
- A random search of 500 trials with reduced search space from the last time
- A hyperband search of 270 total trials using the same space as previously
- A hyperband search of 270 total trials exploring deeper and broader networks, increasing the number of layers from 1-3 to 2-5 and layer size from 4-30 to 4-60

Training the Top Contenders of Each Search

The Final Neural Net Model

- An ensemble model was constructed using the top n models from a total of 150 trained models
- The ensemble size (n) and aggregation method was chosen to minimize the loss on the training set
- n = 4, aggregation = median



XGBoost Regressor (1)

Implementation of gradient boosting that produces a prediction model in the form of an ensemble of decision trees.

1. HPO through **random search** of 50 trials

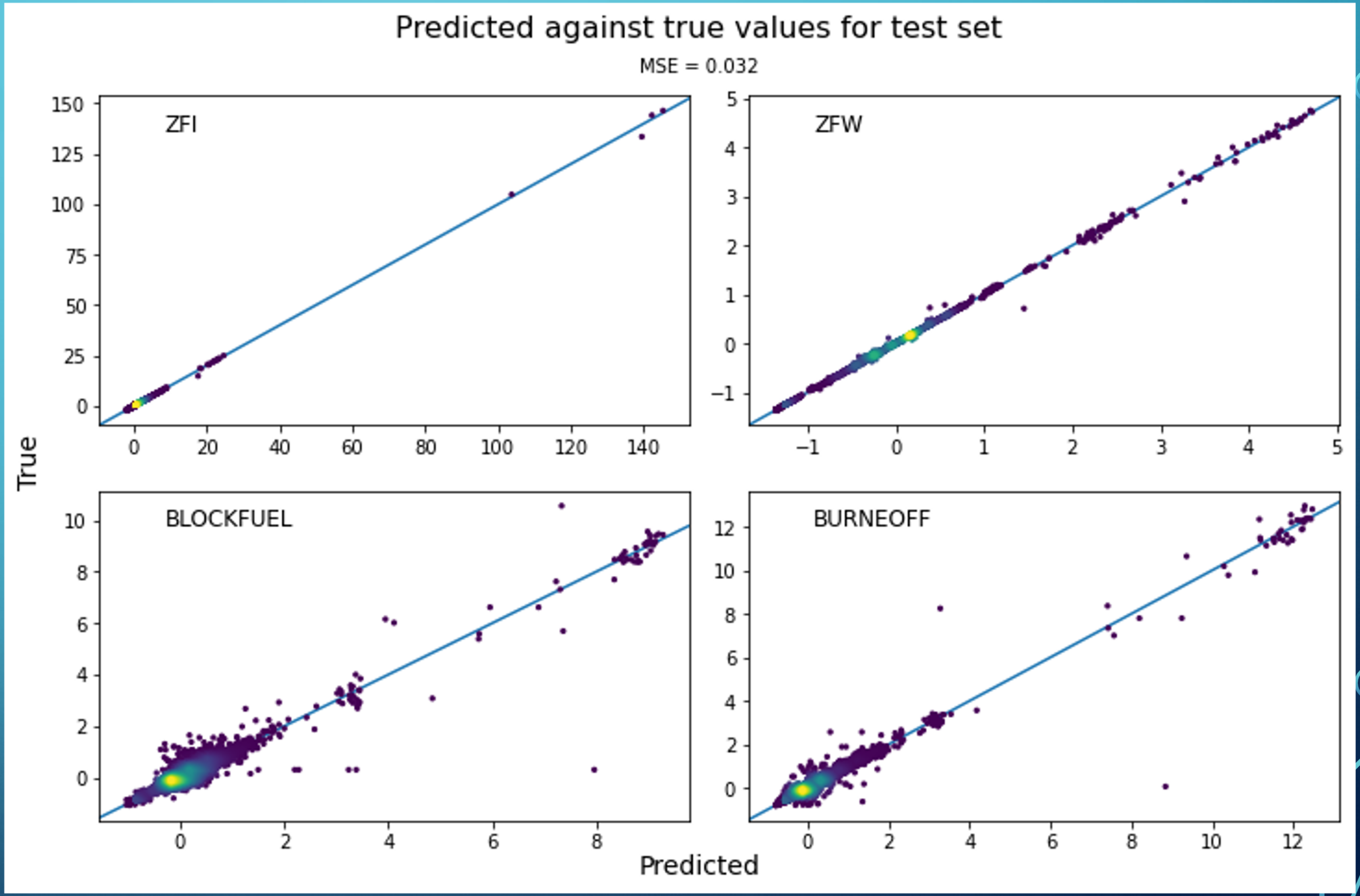
Hyperparameter	Range	Optimized Hyperparameter
Learning rate	0.1-0.3	0.3
Number of estimators	100-300	200
Maximum depth	5-15	10

1. Train the model
2. Perform **cross-validation**
3. Score on test set: **MSE = 0.032**

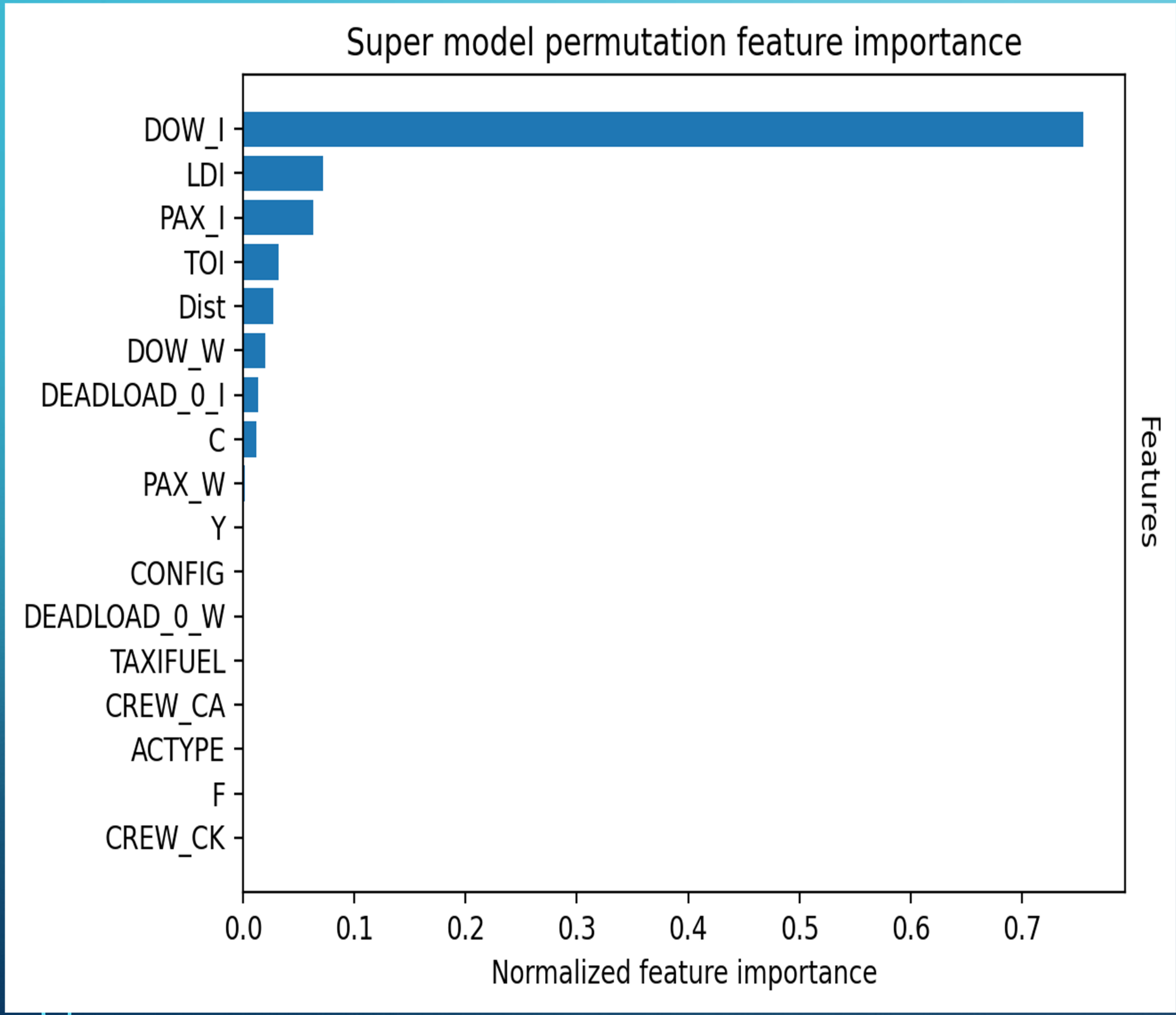
   Score on individual decision tree: MSE = 0.442

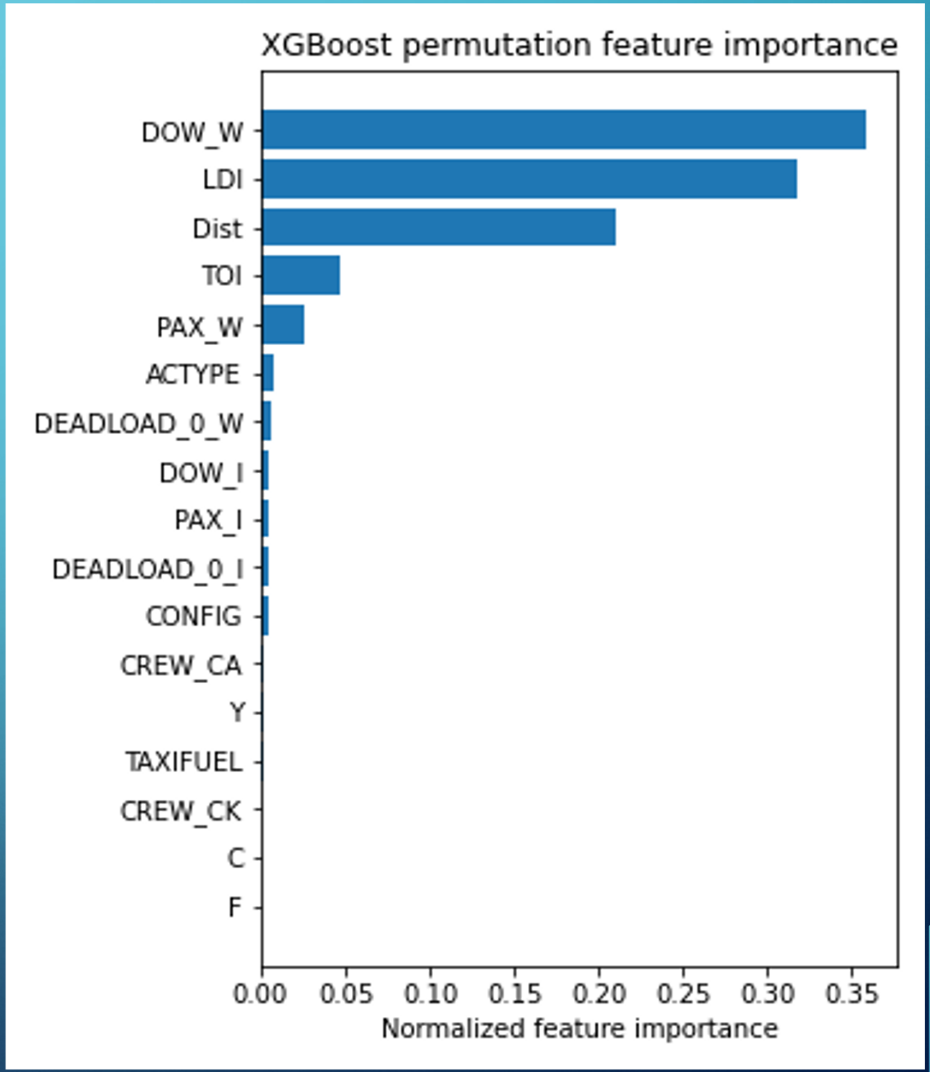
XGBoost Regressor (2)

Precise predictions and better treatment of outliers



Feature Permutation Importance





Conclusion and outlook

- Least MSE found in the tree (0.032) followed very closely by the NN ensemble (0.054)
- Always try a tree first :D
- Final super-ensemble: tree + NN
- The decisions in the tree are not differentiable, so they cannot be optimized ⇨ final model would be the NN
- Optimization problem can be done with the final model

THANK YOU FOR YOUR ATTENTION