Transfer Learning for Short-Term Load Forecasting; Comparing CNN and LSTM

Bachelor's Project by Antony Krymski (s4478177)

rijksuniversiteit groningen

Introduction

- **STLF:** The prediction of electrical power/energy demand for a short time horizon, typically ranging from a few minutes to a few days.
- Deep Learning models used for STLF require **large** amounts of data
- **New buildings** lack this historical data
- **Transfer Learning** can be used
- **Fine-tuning**

Research Questions

Main Research Question:

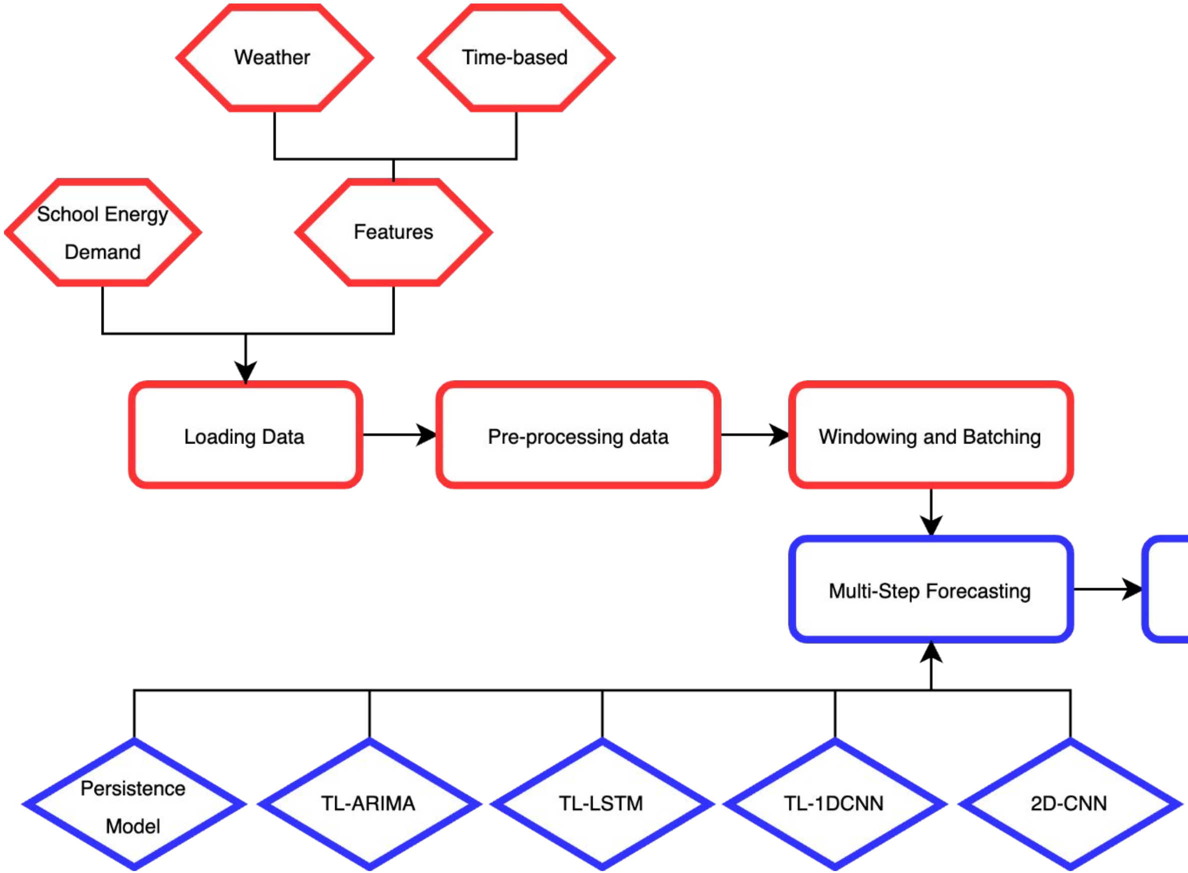
Which, if any, out of TL-LSTM, TL-1DCNN and TL-2DCNN models can effectively predict the short-term energy consumption of selected school buildings?

Sub-questions:

Does adding temperature as a feature improve the accuracy of the models?

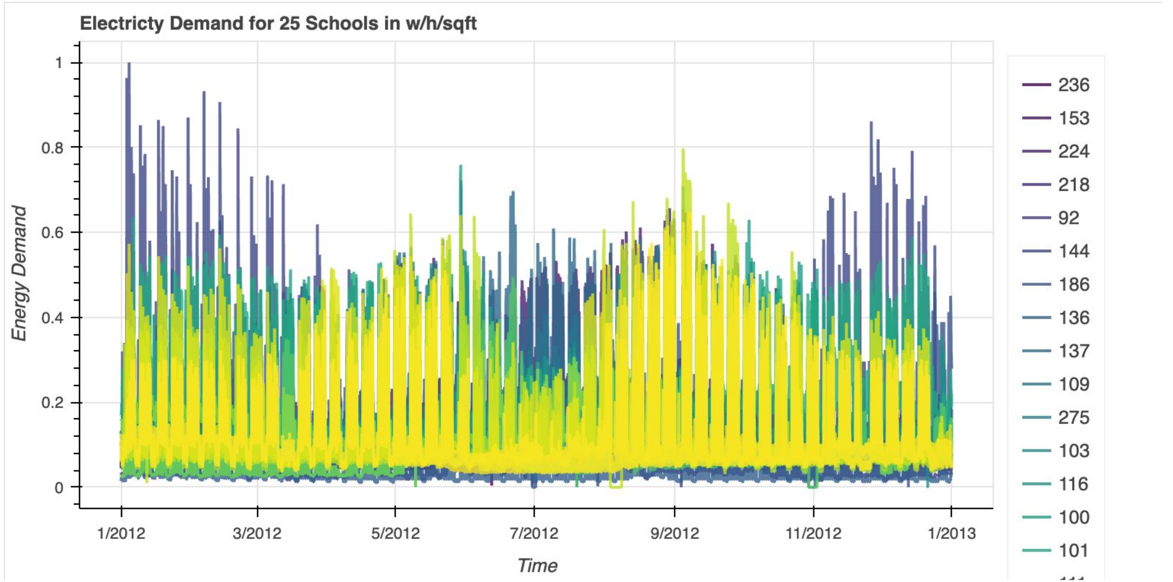
Does applying fine-tuning on the best-performing model result in an increase in accuracy?

Design of Experiments



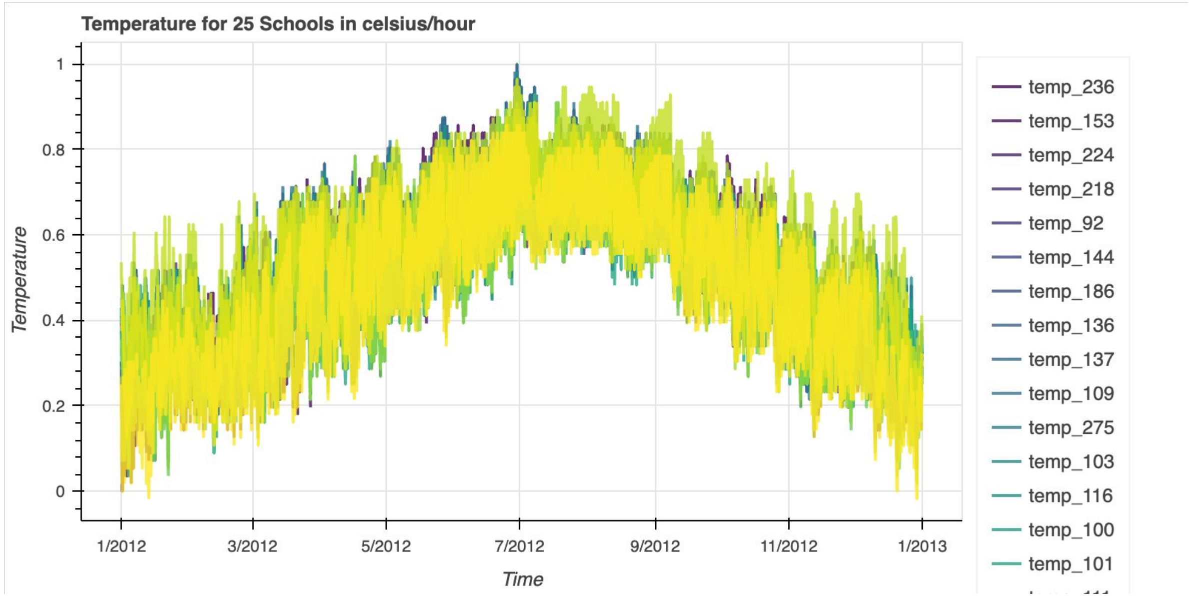
Pre-Processing

- 5-minute energy usage data for 100 commercial/industrial sites for 2012-2013 from EnerNOC.
- Dataset was filtered for schools.
- Converted from kw/5 min to w/hour/sq foot.
- Min-max scaling of electricity demand to 0 to 1.



Pre-Processing

- Data from weather API open meteo in celsius per hour
- Min-max scaling of temperature data to 0 to 1.



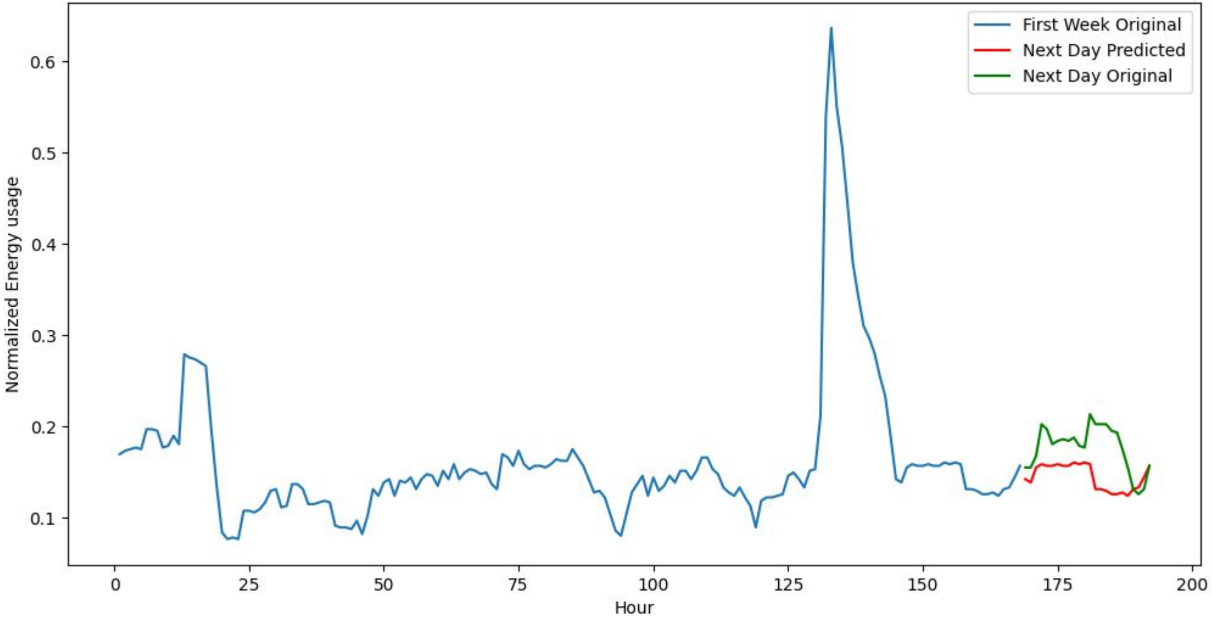
Feature Engineering

- One-hot encoded day of the week, month and school holidays
- Circularly encoded the hour of the day
- Data was windowed by weeks (24*7 timesteps)
- 24 features in total
    - Last week of electricity demand
    - Last week of temperature
    - 22 time features of last week

Naive Model

Persistence Model (Naive Forecast):

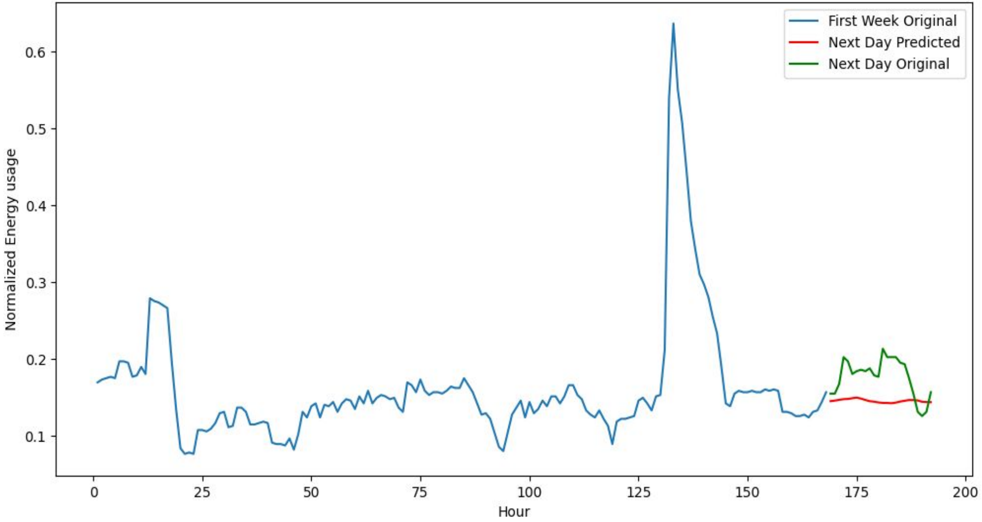
- Typical benchmark
- Forecast next 24 time steps in test as the last 24 time steps.



Arima Model

ARIMA Forecasting:

- Model Choice: Utilized Statsforecast MSTL with daily and weekly seasonality, and AutoARIMA for trend forecasting.
- Rolling Forecast: Iteratively trained on data and predicted the next 24 hours, updating the model each time.



Evaluation on Test Schools:

- Applied the forecasting model on each school in the test dataset.
- Computed the average MAE and MSE for all the forecasts.

LSTM

LSTM Model (based on an implementation from Dongsu Kim et al.):

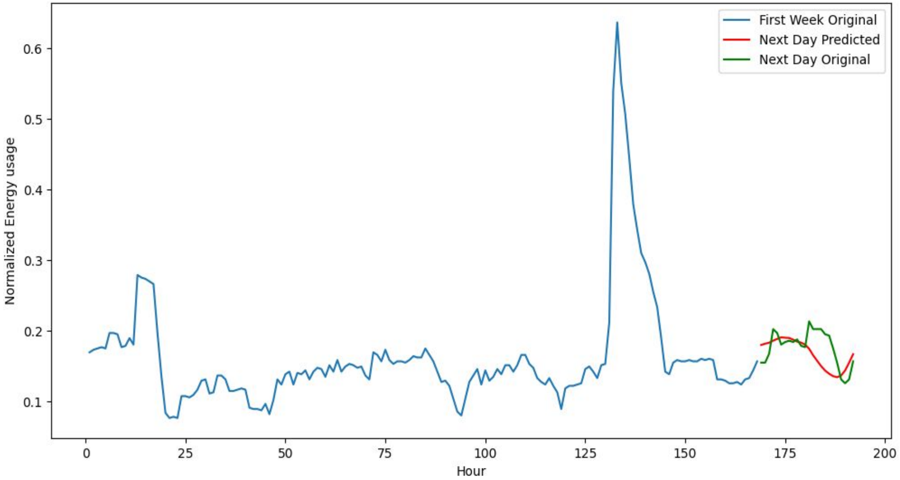
Input Layer: Accepts time series data with defined time steps and features.

LSTM Layers:

- Layer 1: 8 units with sequences returned.
- Layer 2: 16 units with sequences returned.
- Layer 3 & 4: 32 units each, both returning sequences.
- Layer 5: 64 units without returning sequences.

Regularization: Dropout layer with a 0.5 rate.

Output Layer: Dense layer with 24 units and a sigmoid activation function.



TL-1DCNN

1D-CNN Model (based on an implementation from Tim Oates et al.):

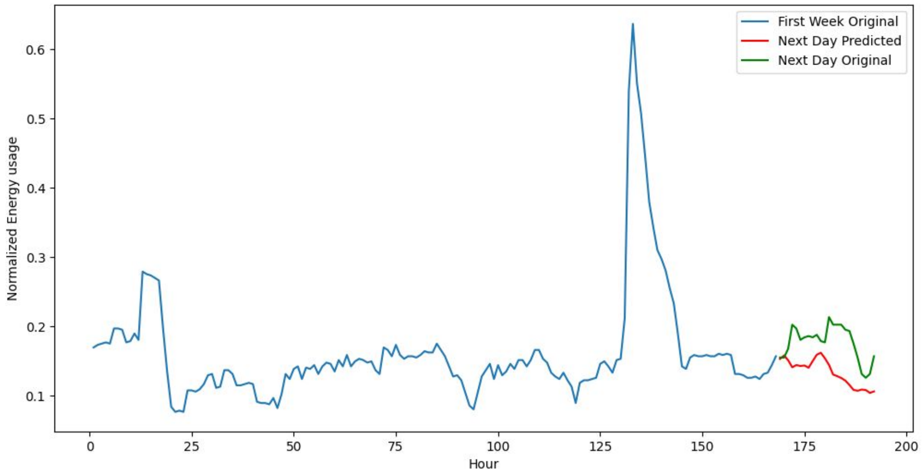
Input Layer: Accepts time series data with defined time steps and features.

Convolutional Blocks:

- Block 1: 128 filters with kernel size of 8.
- Block 2: 256 filters with kernel size of 5.
- Block 3: 128 filters with kernel size of 3.

Pooling Layer: Global average pooling across time steps.

Output Layer: Dense layer with 24 units and a sigmoid activation function.



TL-2DCNN

2D-CNN Model (based on the VGGNET16 implementation from Karen Simonyan et al.):

Custom VGG16-inspired Architecture for reduced GPU usage:

Input Layer: Accepts 2D data (time steps x features) with a single channel.

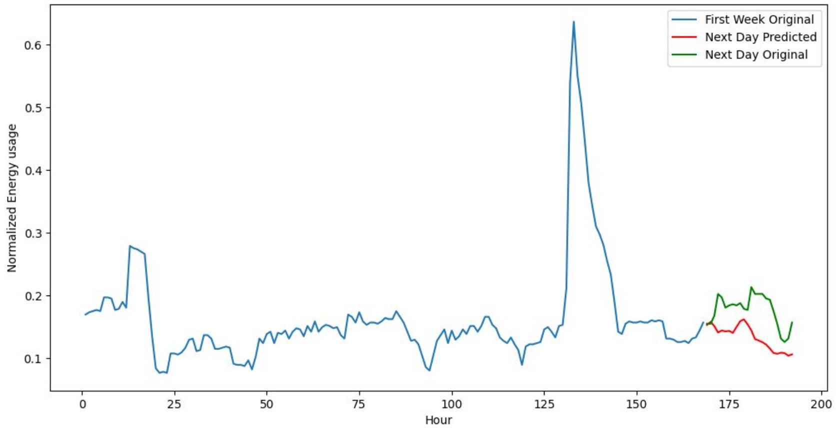
2 VGG Blocks:

- Block 1: 2 convolutional layers with 64 filters, kernel size (3x3).
- Block 2: 2 convolutional layers with 128 filters, kernel size (3x3).

Pooling: Each VGG block concludes with a max pooling layer, reducing spatial dimensions by half.

Fully Connected (FC) Layers:

- Dense layer with 256 units and ReLU activation.
- Dropout layer with 50% drop rate.
- Final dense layer with 24 units and a sigmoid activation function.



Results

Model	MAE	MSE	Approx. MAPE(%)	Percentage Decrease Against ARIMA MAE (%)
Naive	16.4073674	1086.6172515	21.29	-
ARIMA	14.4435396	443.7503036	18.74	0.00%
TL-1DCNN	15.6585732	579.4733276	20.32	8.41%
TL-1DCNN-W	15.0681953	503.6648865	19.55	4.32%
TL-LSTM	13.7397156	467.0726013	17.83	-4.87%
TL-LSTM-W	13.1282921	419.3503418	17.03	-9.11%
TL-2DCNN	12.7136984	385.3051147	16.50	-11.98%
TL-2DCNN-W	12.6932468	380.1078186	16.47	-12.12%

Adding Temperature as a Feature

Models	Addition of weather on MAE (%)	Addition of weather on MSE (%)
TL-1DCNN	3.41%	15.21%
TL-LSTM	3.30%	6.15%
TL-2DCNN	-0.24%	1.05%

Analysis

- ARIMA outperforms 1DCNN on MAE and MSE
  - Complexity vs Simplicity
  - Feature Sensitivity
  - Seasonality & Trends
- ARIMA outperforms LSTM on MSE
  - Error Magnitude
  - Model Sensitivity
  - Noise Handling
- Adding temperature as a feature improved all DL models
  - Model Simplicity Benefits More
  - Diminishing Returns
- 2DCNN-W is an overall winner

Hyperparameter Tuning TL-W-2DCNN

- Implemented a hyperparameter tuning process .
- Used Hyperband method to efficiently search for the best parameters.
- Tuned parameters include depth of the network, kernel size, dilation rate, and initial number of filters.

Hyperparameter Tuning for Custom VGG-inspired Architecture:

Base Structure:

- Input Layer: Accepts 2D data (time steps x features) with a single channel.
- VGG Blocks: Convolutional layers followed by max-pooling.

Hyperparameters Tuned:

- Initial Filters: Choices of [16, 32, 64, 128].
- Depth: Choices of [2, 3] VGG blocks.
- Dilation Rate: Choices of [1, 2] for each block.
- Kernel Size: Choices of [1, 3, 5, 8] for both height and width in each block.

Fully Connected (FC) Layers:

- Dense layer with 256 units and ReLU activation.
- Dropout layer with 50% drop rate.
- Output layer: 24 units with a sigmoid activation function.

Tuning:

- Method: Hyperband optimization.
- Objective: Minimize validation loss.

Resulting Tuned TL-W-2DCNN

- The tuned model is only different in kernel size to original.

Custom VGG16-inspired Architecture (Tuned):

Input Layer: Accepts 2D data (time steps x features) with a single channel.

VGG Blocks (Dilation Rate =1):

- Block 1: 2 convolutional layers with 64 filters, kernel size (3x3).
- Block 2: 2 convolutional layers with 128 filters, kernel size (1x3).

Pooling: Each VGG block finishes with a max pooling layer, halving spatial dimensions.

Fully Connected (FC) Layers:

- Dense layer with 256 units and ReLU activation.
- Dropout layer with 50% drop rate.
- Output layer: 24 units with a sigmoid activation function.

Results

Model	MAE	MSE	MAPE(%)	Improvement over TL-2DCNN-W MAE (%)	Improvement over TL-2DCNN-W MSE (%)
HT-TL-2DCNN-W	12.5299072	367.2881165	16.2578240	-1.29%	-3.37%

Fine-Tuning TL-2DCNN

- Tune set is based on test set.
- The number of weeks was parameterized.
- The model was fed 2, 4, 8, 24 weeks of test data

Preparation:
- Load pretrained HT-TL-2DCNN-W model.
- Freeze all layers except the last one.

Callbacks:
- Checkpoint: Save the model with the lowest training loss

Compilation:
- Optimizer: Adam with a learning rate of 0.00001
- Loss: Mean Squared Error (MSE).
- Metrics: Mean Absolute Error (MAE).

Training:
- Train on the tuning dataset (X_tune, y_tune).
- Validate on (X_val, y_val).
- Train for 10 epochs.

Results

No. of weeks used to fine-tune	MAE	MSE	MAPE(%)
2 weeks	19.0292664	781.8784180	15.22
4 weeks	18.6608410	760.6349487	15.01
8 weeks	17.8768368	713.5093994	14.57
24 weeks	19.0304451	735.7049561	14.07
Overall	18.6493473	747.9319305	14.72

- 24 weeks performed the best

Analysis

- Overfitting Risk:
  - 2 weeks: Highest risk of overfitting due to very limited data.
  - 8 weeks: More generalized than both 2 and 4 weeks.
  - 24 weeks: Best generalization and lowest risk of overfitting.
- Data Variety & Seasonality:
  - 2 weeks: Captured very recent patterns but lacked broader perspective.
  - 24 weeks: Best at capturing seasonality by providing the broadest perspective.
- Training Stability:
  - Longer durations like 24 weeks have less noise and provide better stability.
- Relevance of Historical Data:
  - 4 weeks: Extended slightly into the past.
  - 8 weeks: Considered more historical data.
- Transfer Learning Dynamics:
  - Models trained on longer durations, especially 24 weeks, benefit from understanding extended patterns. This is most transferable when fine-tuning on newer data.

Conclusion

- Temperature reduced the MAE and MSE in DL models
  - For all DL models
    - MAE on average improved
    - MSE on average improved
- Fine-tuning
  - The longer the duration of data to be tuned the better it performs

Acknowledgements

- Would like to thank Professor Düstegör and Andres for supervising my thesis!