DTU Compute 02456 Deep Learning
s201441 , s201792, s201232
Yevhen, Cedric and Tim

Crowdedness prediction in public transport under Covid-19

Problem Statement

During Covid-19 pandemic social distancing in public transport is an important matter to prevent spreading the virus. Thus, it would be beneficial to know when and where there are bottlenecks in the public transport network. Our goal is to reduce the capacity problem by predicting the crowdedness for a specified time interval with RNNs. As a data basis, we mainly use check-in/check-out numbers of Rejsekort users. We supplement this information with data from counting sensors that are installed in a small fraction of buses.

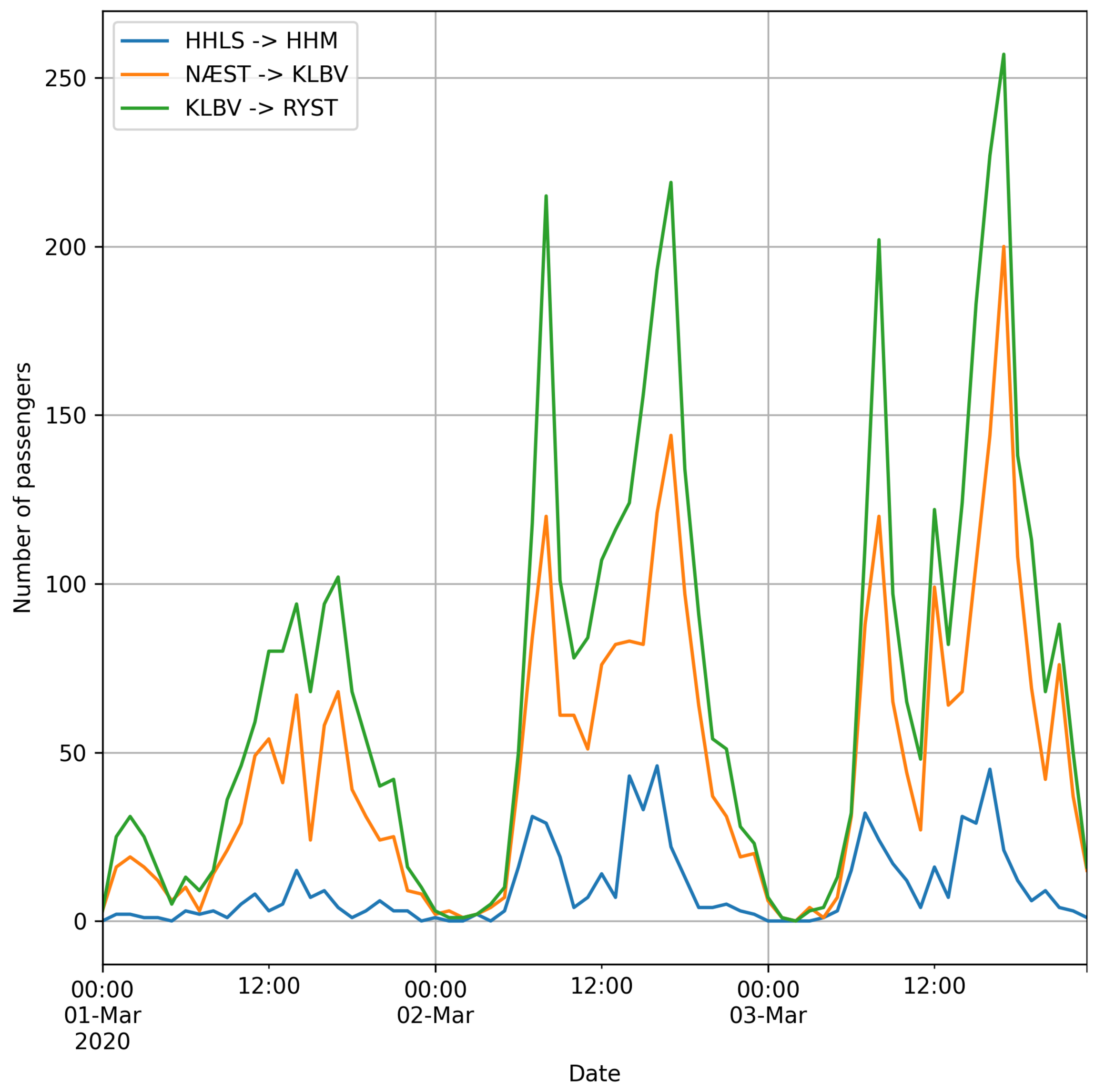

Figure 1: Number of passengers for three of nine intervals within a 72-hour time interval.

Preprocessing

- Rejsekort passenger numbers entering and leaving at a specific timing point of the busline ‘150s’ (scalable to other bus lines)
- We want passenger numbers *between* two timing points
- Reshape the data to a multivariate time series that represents the 9 intervals between the 10 timing points
- Clear patterns in the passenger numbers can be noticed → see Figure 1
- Combine this with sensor data to get more realistic predictions

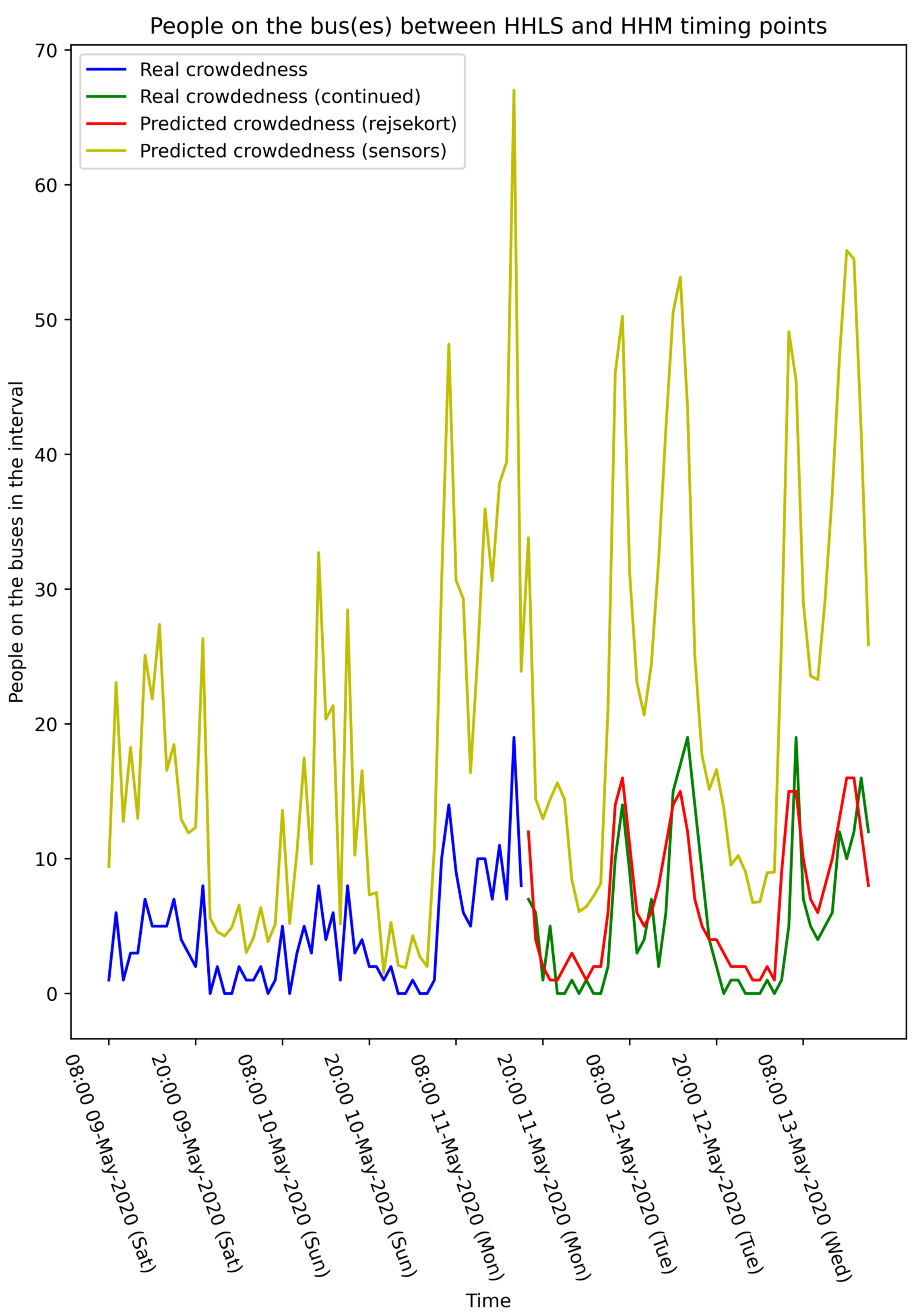

Figure 2: Final predictions of Seq2Seq and rejsekort-to-sensor models

Approach 1: Multistep LSTM

- A LSTM is the obvious choice for time series prediction
- We extended the lecture model to multiple input and output features (for the 9 intervals)
- Problem: lecture model only predicts next element in sequence
- Repeated calling of model gives some result, but uses model for something it’s not trained for
- Multistep LSTM with teacher forcing gave bad results

Approach 2: Seq2Seq model

- We decided to make our model more complex in order to capture more of the pattern
- Encoder-Decoder model for Seq2Seq predictions first encodes an input sequence into a context vector and then tries to decode the vector into an output sequence
- Both Encoder and Decoder consist of one LSTM and additional linear layers
- We experimented with adding more LSTM layers and switching to bidirectional LSTMs which led to overfitting and longer training times
- To help the model better concentrate on certain parts of the input sequence, we added the attention layer
- We created a second model that predicts the real crowdedness by finding a relation between rejsekort and sensor data
- The output of Seq2Seq model is fed into the second model to make the final prediction

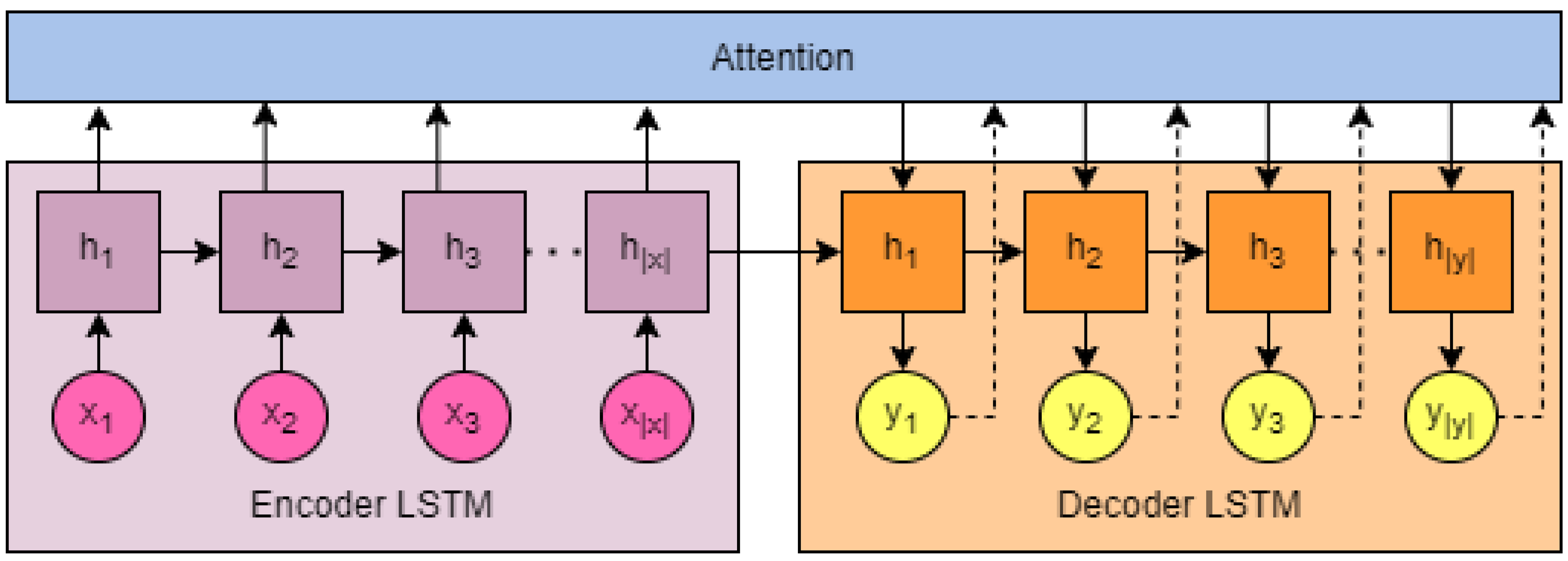

Figure 3: Model of the final network

Further improvements

- Adding more important features like weather and holiday dates
- Adding CNN-like layers that could capture spatial patterns in the data (locations of the stops)
- Trying other types of attention for Seq2Seq model