**MIE324 Project Final Report: HighHOEPs**
Word count: 1991
Penalty: 0%

Sam Harrison – 1001297777
Yanchen Ma – 1003423048

## Goal and Motivation

In Ontario's wholesale electricity market, import and export transactions are scheduled based on predicted "pre-dispatch" prices, but are compensated based on the real-time Hourly Ontario Energy Price (HOEP). When the HOEP differs significantly from pre-dispatch prices, there is a possibility for significant financial gain or loss. The goal of this project was to use publicly available electricity market data to build a model for predicting the HOEP better than the publicly available pre-dispatch prices.

## Sources of Data

This project relied on publicly available data from Ontario's Independent Electricity System Operator (the IESO). The IESO publishes more than 70 "reports" (available at http://reports.ieso.ca/) in xml or csv format. For this project, a data scraper was used to download thousands of files from the IESO's online repository, which provides data for the past 30 days. The following reports were used:

- *Pre-dispatch Market Prices* – the IESO's projected prices for future hours
- *Real-time Market Prices* – the actual energy price, issued every 5 minutes
- *Hourly Ontario Energy Price*[1] – the Hourly Ontario Energy Price (the average of 12 real-time market prices.
- *Pre-dispatch Market Totals* – the IESO's forecast demand for future hours
- *Real-time Market Total* – actual market demand, with 5-minute granularity
- *Variable Generation Forecasts* – wind and solar generation forecasts for future hours
- *Generator Output and Capability* – historic hourly generator output from all generators in the IESO market

The data from these 7 reports was manipulated to combine data points about a given hour in a table. Each hour of data had the following features:

| Feature | Description |
|---|---|
| Timestamp | Time index for hour |
| HOEP | Hourly Ontario Energy Price in hour |
| MCP_1 | Market Clearing Price for 1st 5-minute interval of hour |
| … | … |
| MCP_12 | Market Clearing Price for 12th 5-minute |

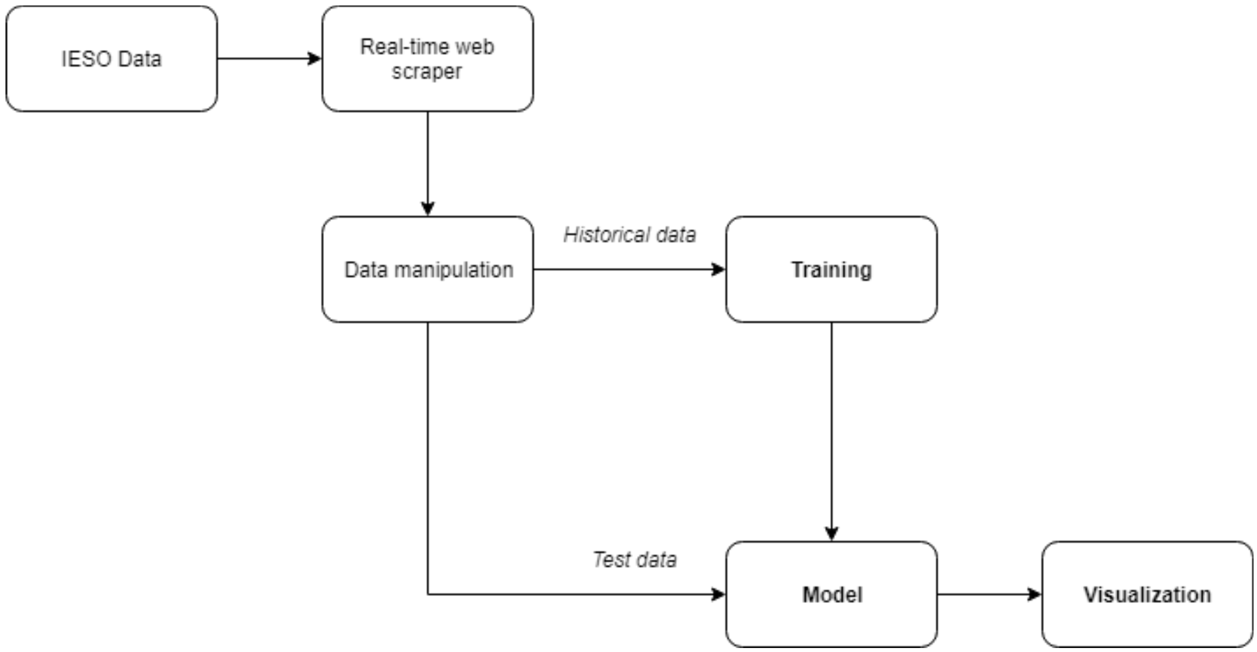
[1] The real-time market price was included in addition to the HOEP because the data is published sooner, thus providing additional features about the current hour when making real-time predictions.

| | |
|---|---|
| | interval of hour |
| Price_PD1 | Pre-dispatch price for this hour, issued one hour in advance |
| … | … |
| Price PD5 | Pre-dispatch price for this hour, issued five hours in advance |
| RT_Market_Totals_1 | Demand for 1st 5-minute interval of hour |
| … | … |
| RT_Market_Totals_12 | Demand for 12th 5-minute interval of hour |
| Market_totals_PD1 | Forecast for demand in this hour, issued one hour in advance |
| … | … |
| Market_totals_PD5 | Forecast for demand in this hour, issued five hours in advance |
| Wind_output | Output of wind generators in this hour |
| Wind_output_PD1 | Forecast output of wind generators in this hour, issued one hour in advance |
| … | … |
| Wind_output_PD5 | Forecasted output of wind generators in this hour, issued five hours in advance |
| Solar_output | Output of solar generators in this hour |
| Solar _output_PD1 | Forecasted solar of wind generators in this hour, issued one hour in advance |
| … | … |
| Solar _output_PD5 | Forecast solar of wind generators in this hour, issued five hours in advance |
| Gas_output | Output of gas generators |
| Nuclear_output | Output of nuclear generators |
| Hydro_output | Output of hydro generators |

Given different feature points become available at different times (e.g. the PD-5 price for hour j is available at j – 5:00, and HOEP and generation output during hour j are available at j + 1:00), certain elements of future hours were withheld from the network to mimic the availability of data in real-time. For example, the network would be marking a prediction about the hour between 1 and 2 PM at 11 AM, only data available at 11 AM would be used. A 2-hour in advance prediction was selected because electricity market participants are required to submit their bids 2 hours in.

**Description of Overall Software Structure**

A block diagram representing the elements of this software project is presented below.




The overall software structure revolves around two distinct data paths – one for historical report data (for training) and one for test data (for inference). This involves five major modules:

- *Web scraper* – this module pulls the desired reports from the IESO's website using the BeautifulSoup4 python library. Since a new file is generated on the IESO every 5 minutes, this volume is much greater than can be manually processed. The web scraper automates the filtering and downloading of these reports.

- *Data manipulation* – this module takes the CSV and XML files downloaded by the web scraper and performs the required manipulations to achieve the output format detailed in the previous section. For each report, the individual files are parsed into one Pandas dataframe. This is done using a separate function for each report as their structure differs. Data from each of the seven dataframes is then stitched together with the previous 5 hours of predictions for forecast features (pre-dispatch market prices, market totals, and wind/solar output predictions) to create the final data used for training. All of the non-price features are then normalized by standardizing over time to counter the issue encountered in training of exploding gradients. This takes place in two scripts: *parse_data.py* and *data_manipulation.py*.

- *Training* – this module splits historical data into appropriate training/validation splits and creates the Dataset class used for the training data. Code for the training loops, evaluation functions, and additional data manipulation required for each model is also contained in this component. This is contained in two files: *RNN_train.py* and *train_linear_and_LSTM.py.*

- *Model* – the neural network models, contained in *models.py*.

- *Visualization* – the final module is a visualization layer. This module uses matplotlib to show predictions generated by the model with the HOEP and the IESO's price projections to allow for more intuitive assessment of model performance.

## Training, Validation, and Test

### *The Model*

Experimentation with several neural networks was conducted throughout the course of this project – primarily with a multilayer perceptron (MLP) as a baseline and varying sizes of recurrent neural networks (RNNs). As the MLP model served only as our baseline and produced relatively poorly in terms of prediction accuracy, this section of the report will primarily discuss the RNN models.

Since the HOEP is a time-series where past behaviour is often a good indicator of future values, the design choice to use an RNN was made to leverage this sequential property. Two separate RNN models using LSTM cells were developed and tested – one that took in a sequence of the past 5 hours and one that used a variable-length window of up to 100 past hours. The purpose of this comparison was to determine the proper balance between model performance and memory usage in training. The performance of these two models is shown in the comparison below.

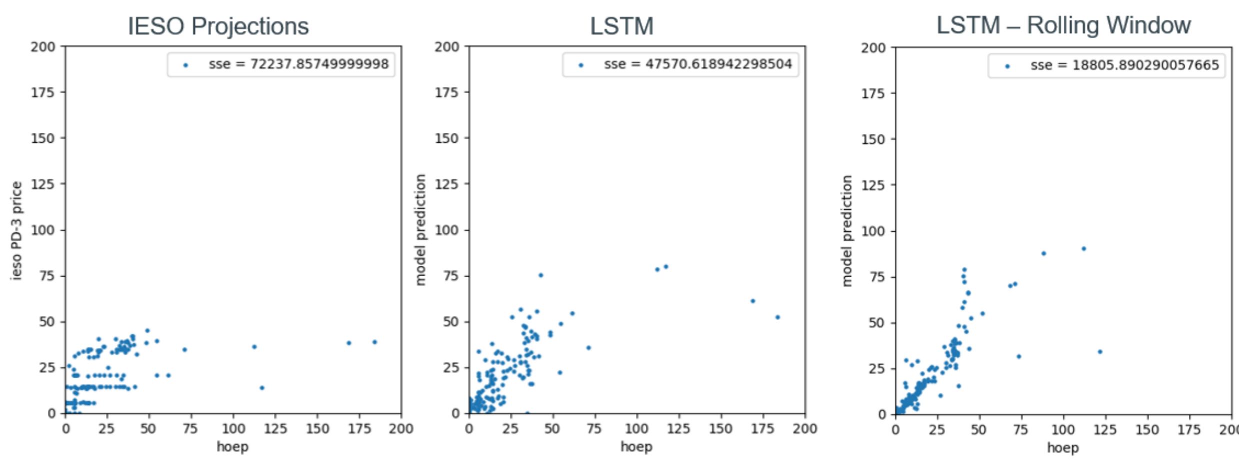



*Figure 1: From left to right: predictions from the IESO, the LSTM model with 5 hours of past data, and the LSTM model with 50 hours of past data.*

This is a view of the accuracy of both models' price predictions compared to the IESO's projections. The x-axis represents the actual HOEP while the y-axis represents the predicted values. A perfect model would produce a diagonal line. It is clear that the RNN using the larger rolling window of 50 hours produces much more accurate predictions.

The final structure of our model can be seen below.

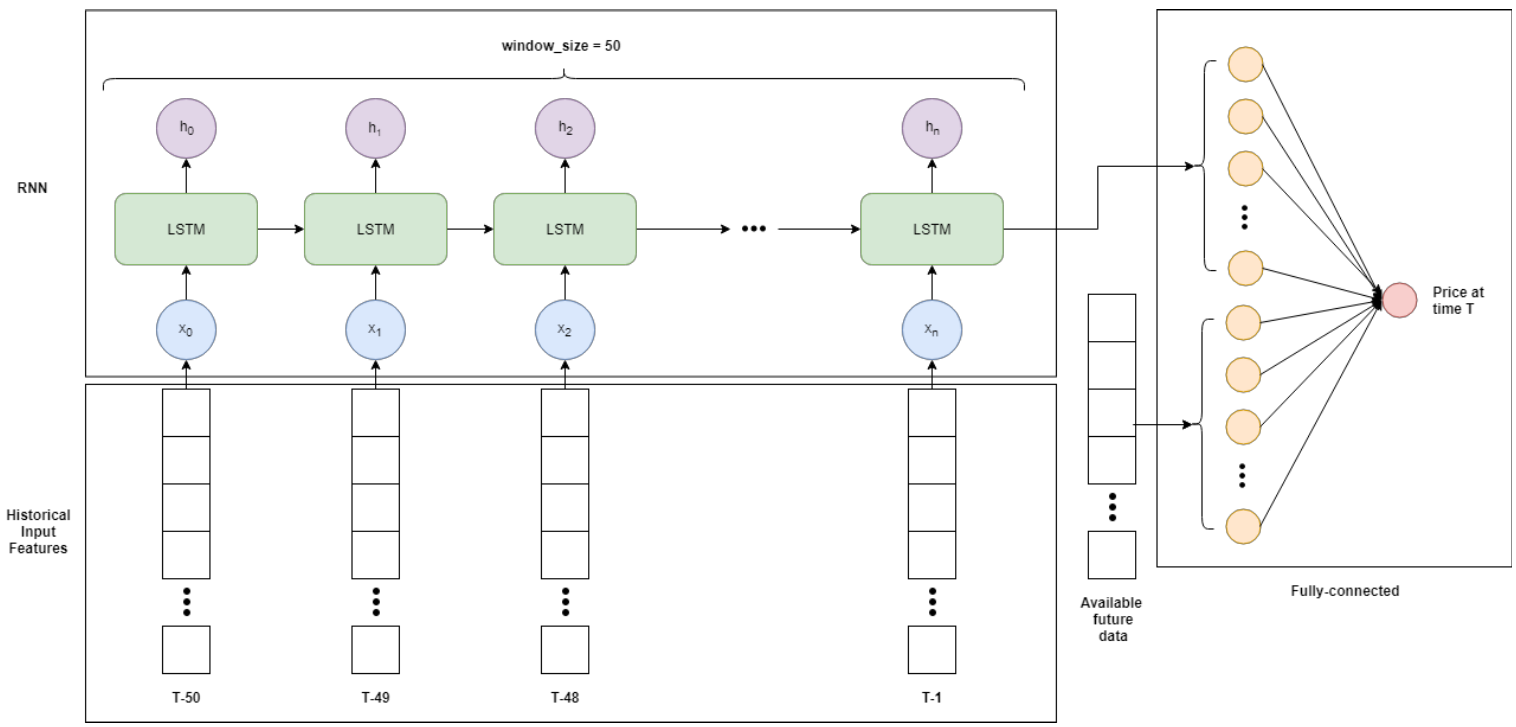



*Figure 2: Model structure. Hours T-50 to T-1 contained all data for these hours, whereas only features of future hours that would have been available at the time of the prediction are used in the fully-connected layer.*

### ***Validation and Test Results***

Our final model was trained using the Adam optimizer, using a decaying learning rate starting at 0.01 and decreasing by a factor of 10 at 50 and 100 epochs. The batch size used was 64 and the model was trained for 200 epochs.

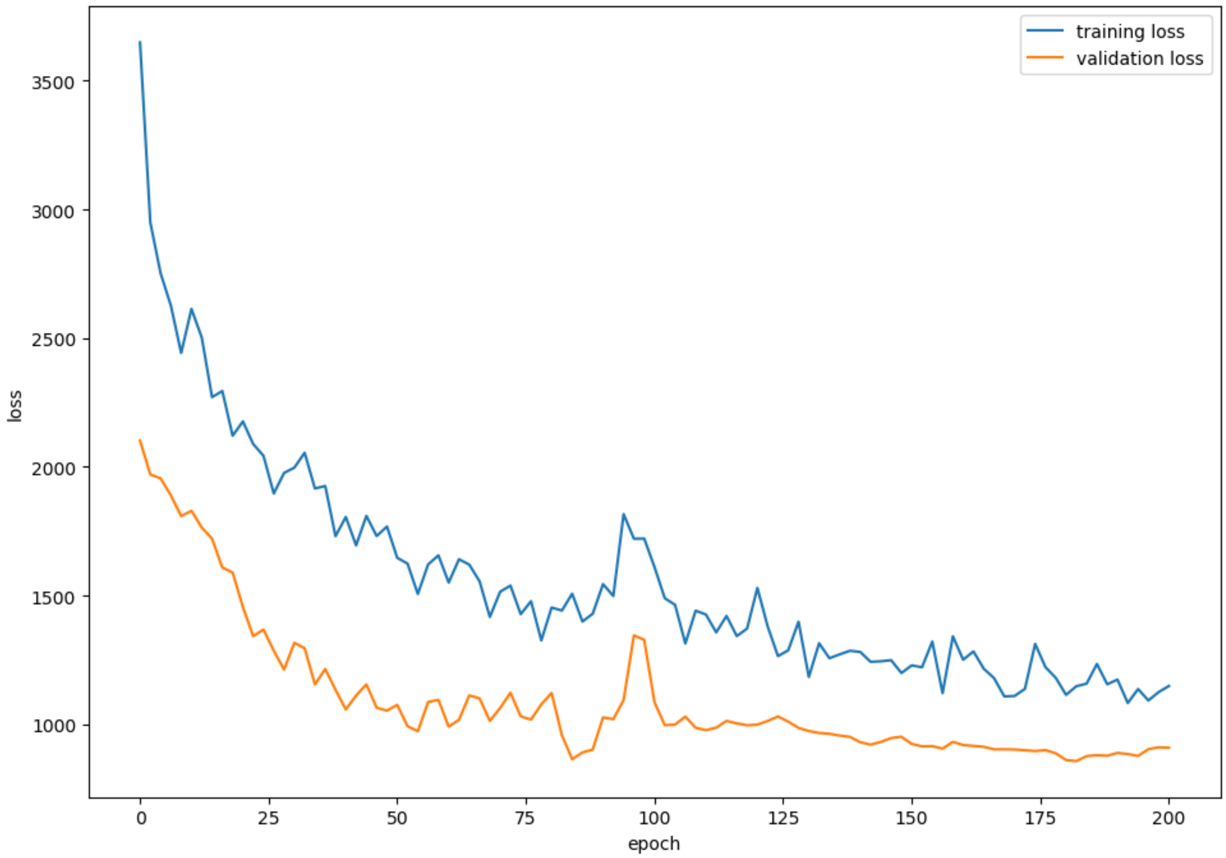



*Figure 3: Training and validation loss.*

This plot of training and validation loss vs. epoch allows interesting insight into the training process and the justification behind the hyperparameter choices.

Throughout the training process, the validation loss is lower than the training loss due to the small dataset size. Due to the fact that the dataset only encompassed 39 days, there were a small number of extreme price spikes in the dataset. This made the training/validation loss sensitive to where these spikes fell in the training/validation split (and consequently the seed of the split). This is discussed further in the key learnings section.

The number of epochs was determined by the point where the validation loss began to plateau. The batch size and optimizer were the ones observed to produce the lowest validation loss with a fixed validation set and the learning rate was chosen such that the training process could be significantly sped up in the early stages where the loss values fall quickly while retaining the ability for the model to find a better minimum in the later stages.

***Impact of Loss Function Selection***

Selection of the loss function for model training provided an interesting choice for this project. The selection of a mean squared error loss emphasized outliers in the dataset;

models trained with this loss function performed much better in predicting price spikes, but had greater loss in most hours. This effect can be seen below:

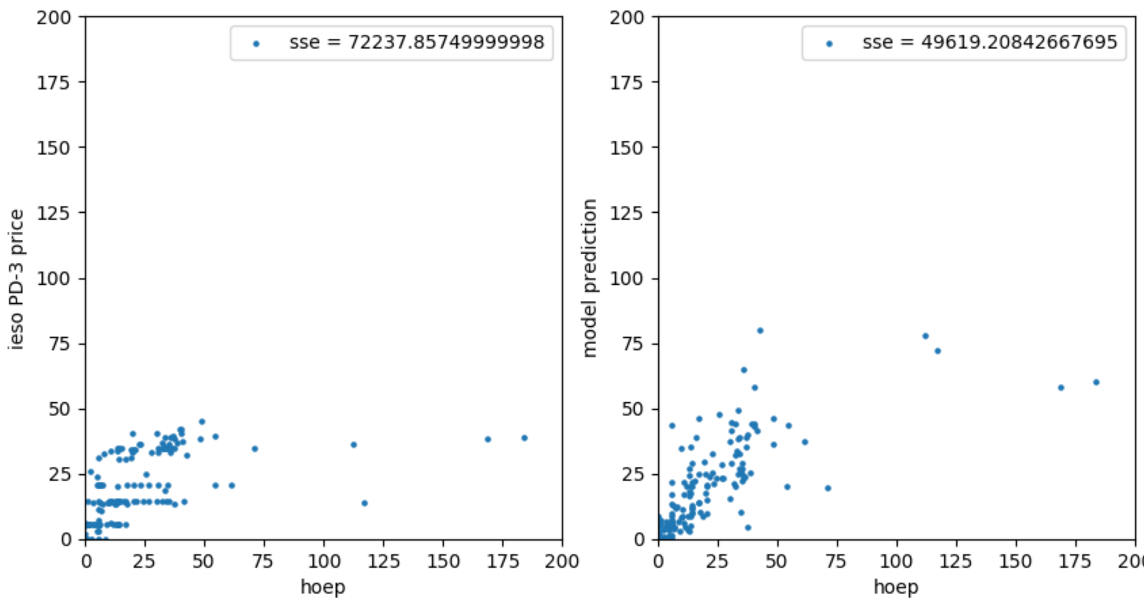



*Figure 4: Predictions from an LSTM model (right) trained with squared-error loss compared to HOEP. This model preformed far better than the IESO PD-3 price during high price hours. All values provided in $CAD/MWh.*

Conversely, the use of a linear loss function placed less weight in the loss calculation on outlier hours; models trained with this loss performed better during regular hours but did a poor job of predicting outliers.

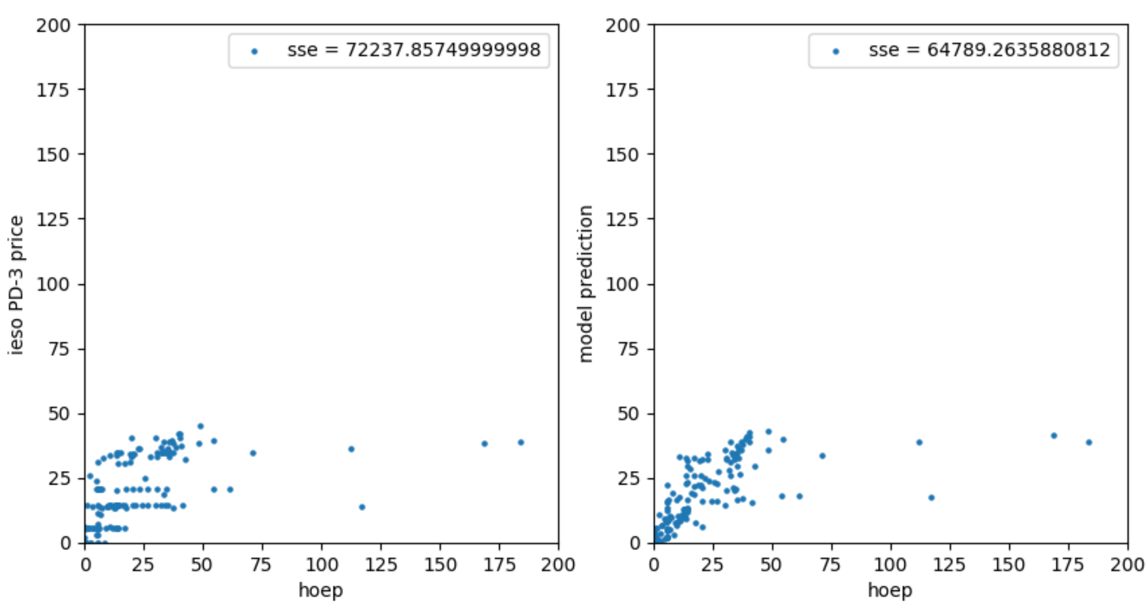



*Figure 5: Prediction from an LSTM model (right) trained with linear loss compared to HOEP. This model preformed far better than the IESO PD-3 price during most hours, though performed no better than the PD-3 price during price spikes. All values provided in $CAD/MWh.*

## Ethical Issues

The potential for ethical conflicts in this project is relatively limited compared to many other machine learning applications. The reduction of pre-dispatch price risk in the wholesale market could lead to increased profits for energy traders and retailers, which could increase wealth inequality. Long-term changes in the wholesale electricity price have distributional impacts between large and small consumers; an increase in wholesale prices typically reduces electricity costs for small consumers and increases costs for industrial-scale consumers.

As a potential positive ethical effect of this project, reduction of price risk could lead to more efficient inter-jurisdictional trade and therefore lower average costs.

## Key Learnings

### *Seek more data early*

The IESO's governing documents require that the IESO provide market participants and individuals with access to public market data without discrimination. 30 days of data is kept online, with some data sources going back to 2002. A request was submitted for all the market data data from 2014-18, which would have increased the amount of training data about forty-fold. Unfortunately, the IESO did not fulfil this request until three days before this report was due.

### *Create a more thoroughly engineered training-validation split*

Due to the relatively small size of the dataset (39 days or 926 hours), the training and validation loss were sensitive to the seed used for a random train-validation split. This was due to the small number of high-priced hours in the dataset, which would dominate the loss function when training.

A solution that should be considered for future iterations of this model is to divide hours into classes by HOEP. A random split could be performed on each class and the results could be merged into one training and one validation set that contained a similar distribution of prices.

### *Better define the design problem*

This project set out to create a predictor for prices. However, significant emphasis was placed on prediction of hours where the HOEP deviates significantly from the IESO's pre-dispatch prices. As discussed in the *Training, Validation, and Test* section, hyperparameter selection resulted in a trade-off between prediction of extreme price spikes and general price prediction. While this trade-off could really only be assessed by how the user – an energy trader or retailor – intends to use this model, having insight into these unknown user preferences would lead to a clearer "right answer" for hyperparameter selection.