Comparison of LSTM and GRU

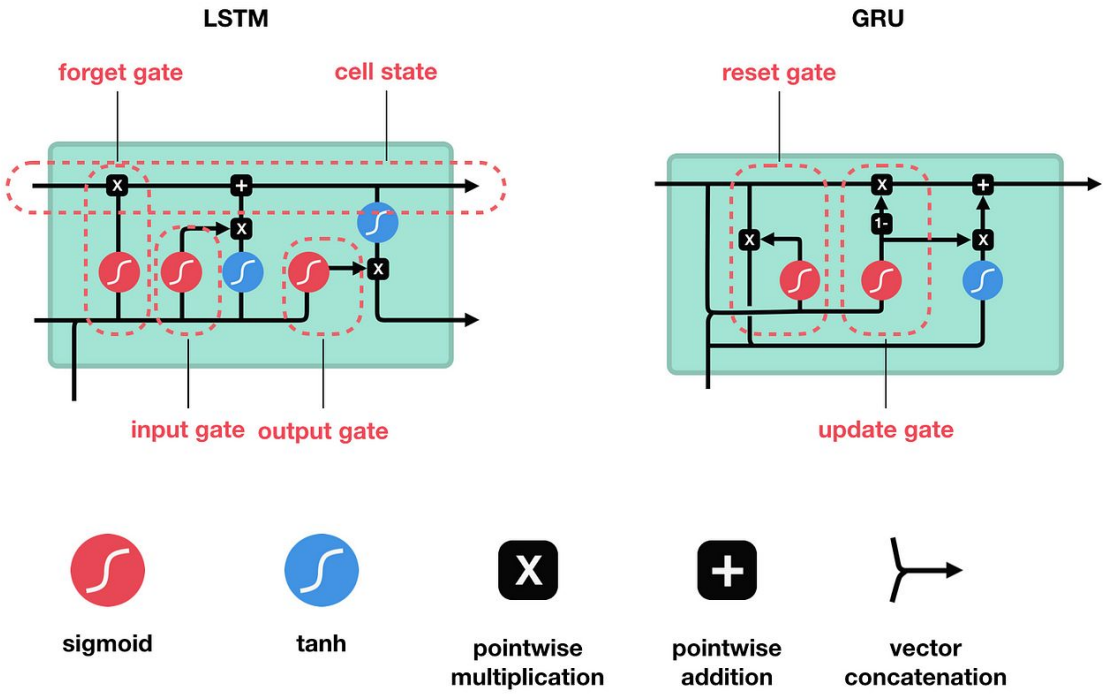
Maciej Lorens

LSTM and GRU

Introduction

- Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) were introduced as a solution to the short-term memory and vanishing gradient problems of the Recurrent Neural Networks (RNN).
- LSTM and GRU excel at processing long sequences, thanks to the gates in their architecture.
- Gates control the flow of information and can learn which parts of the sequence to keep or remove from memory.
- Nowadays, both LSTM and GRU are used in natural language processing, speech recognition or time-series forecasting.

Architectures of LSTM and GRU



Quick overview of the characteristics of LSTM and GRU

LSTM:

- More complex structure
- Training can be time consuming
- Ability to retain long sequences of information

GRU:

- Simpler structure (fewer parameters to learn)
- Training is faster than LSTM
- Will not perform as well as LSTM on very complex tasks

Dataset

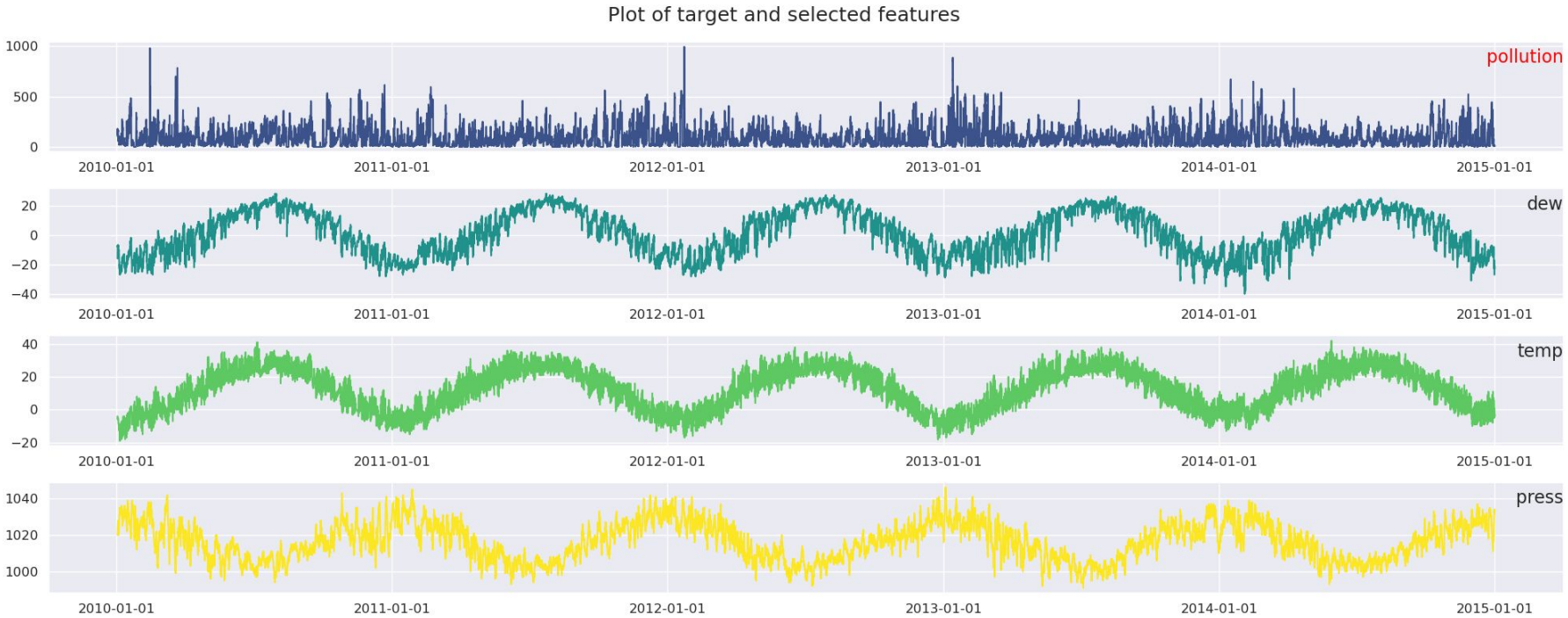
Dataset description

Data comes from hourly reports on the weather and the level of pollution between 2010 and 2015 at the US embassy in Beijing. Altogether, there are 43800 rows and 9 columns:

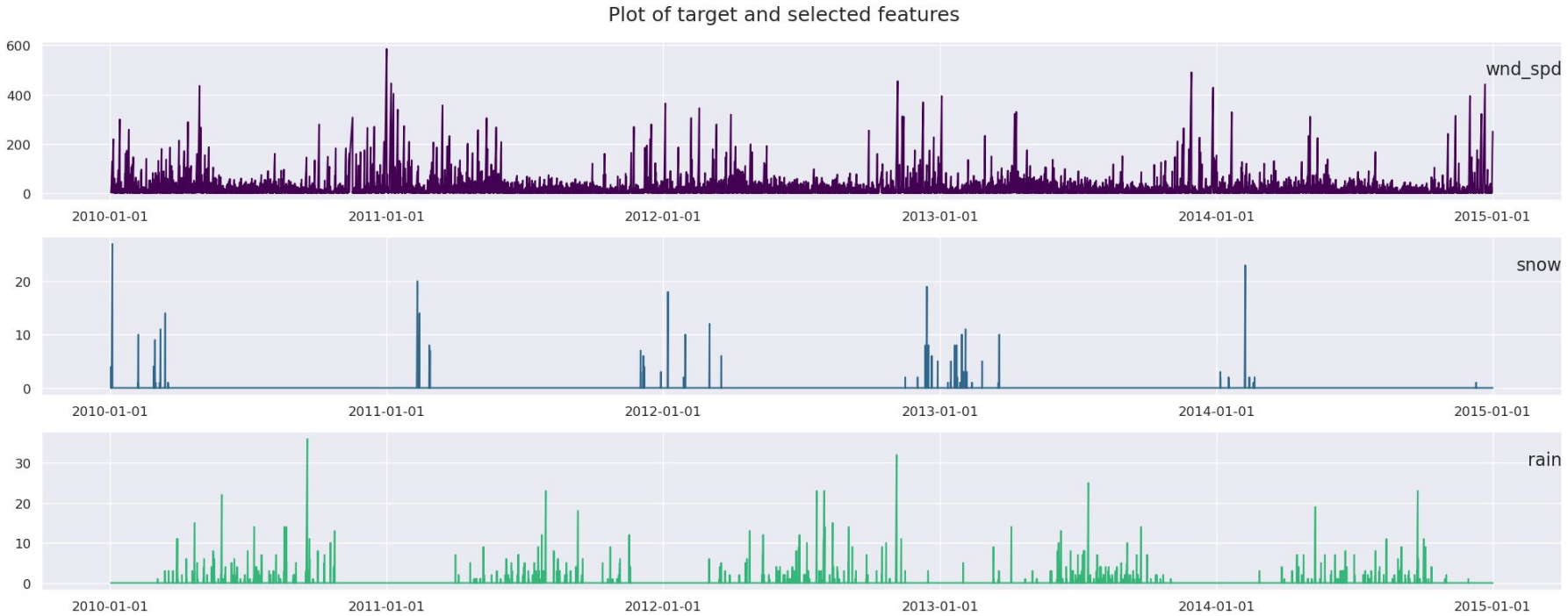
- *date* - date and hour of measurement
- *dew* - temperature to which air must be cooled in order to become saturated with water vapor
- *temp* - temperature
- *pres* - pressure
- *wnd_dir* - wind direction, denoted with NE (north-east), NW (north-west), SE (south-east) or CV (calm and variable)
- *wnd_spd* - cumulated wind speed
- *snow* - cumulated hours of snow
- *rain* - cumulated hours of rain
- *pollution (target variable)* - concentration of PM2.5, which is a mixture of different pollutants (with a diameter lesser than 2.5 μm) that can damage lungs when inhaled

Link to the data: https://www.kaggle.com/datasets/rupakroy/lstm-datasets-multivariate-univariate

Time series visualization



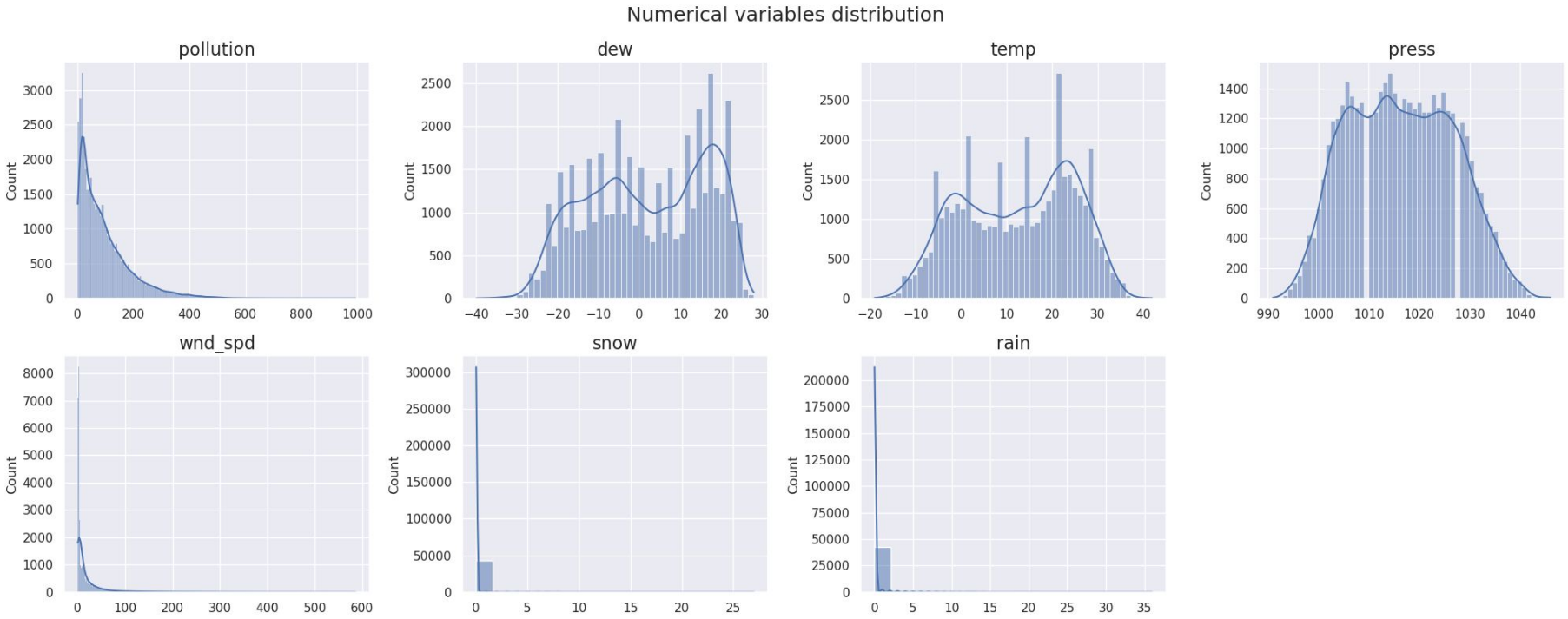
Time series visualization



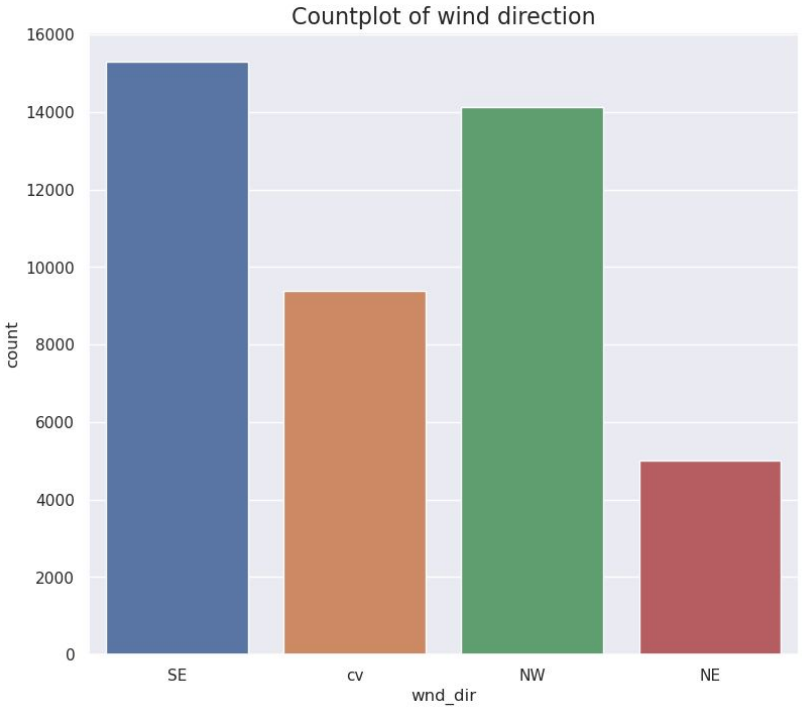
Descriptive statistics

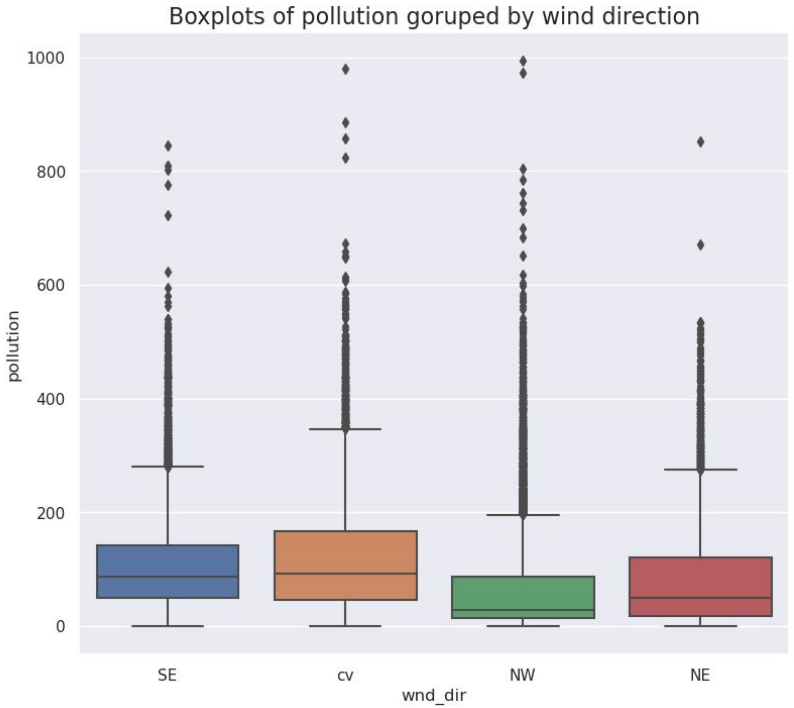
	pollution	dew	temp	press	wnd_spd	snow	rain
count	43800.0000	43800.0000	43800.0000	43800.0000	43800.0000	43800.0000	43800.0000
mean	94.0135	1.8285	12.4590	1016.4473	23.8943	0.0528	0.1950
std	92.2523	14.4293	12.1934	10.2714	50.0227	0.7606	1.4162
min	0.0000	-40.0000	-19.0000	991.0000	0.4500	0.0000	0.0000
25%	24.0000	-10.0000	2.0000	1008.0000	1.7900	0.0000	0.0000
50%	68.0000	2.0000	14.0000	1016.0000	5.3700	0.0000	0.0000
75%	132.2500	15.0000	23.0000	1025.0000	21.9100	0.0000	0.0000
max	994.0000	28.0000	42.0000	1046.0000	585.6000	27.0000	36.0000

Descriptive statistics



Descriptive statistics





Hyperparameter tuning and model training

Preprocessing steps

- Transforming the data, so that:
  - independent variables consist of features and target values 48 hours before the first target measurement time
  - if data is out of scope, padding is applied (unavailable time steps are filled with 0s)
  - the dependent variable consists of pollution values over the next 24 hours
- Wind direction mapped into numeric values with an ordinal encoding (NE=0, NW=1, SE=2, cv=3)
- All data is transformed with MinMaxScaler. Formula: $x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$
- The last available month (720 rows) is kept as a test set
- The remaining data is split into 70% training and 30% validation sets

Results of hyperparameter tuning

I applied grid search to find the best hyperparameters. Each model was run for 10 epochs, with ASHA (Asynchronous Successive Halving) Scheduler and Adam optimization. The grid consisted of the following hyperparameter values:

- **hidden size**: [8, 16, 32]
- **number of layers (stacked models)**: [2, 4, 8]
- **dropout rate**: [0.2, 0.4]
- **activation function**: [None, ReLU]
- **learning rate**: [0.001, 0.01]
- **batch size**: [128, 256]

Results of hyperparameter tuning - LSTM

config/h_size	config/num_l	config/dropout_rate	config/activ	config/lr	config/batch_size	val_loss
32	2	0.4	ReLU	0.001	128	0.004387
32	2	0.2	ReLU	0.001	128	0.004395
8	2	0.2	ReLU	0.01	256	0.004421
32	2	0.4	Identity	0.001	128	0.004461
16	2	0.2	Identity	0.01	256	0.004474

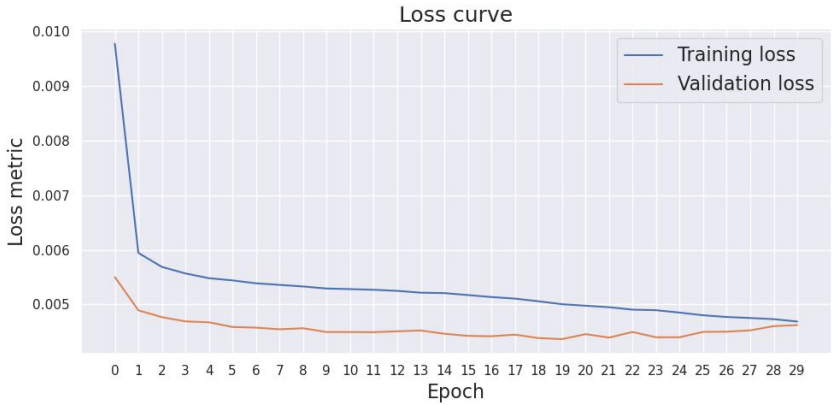
Results of hyperparameter tuning - GRU

config/h_size	config/num_l	config/dropout_rate	config/activ	config/lr	config/batch_size	val_loss
32	2	0.2	Identity	0.001	256	0.004366
32	2	0.4	ReLU	0.001	128	0.004390
32	2	0.4	Identity	0.001	256	0.004414
16	2	0.2	Identity	0.001	128	0.004450
32	2	0.2	ReLU	0.001	256	0.004494

Training results - 50 epochs with early stopping

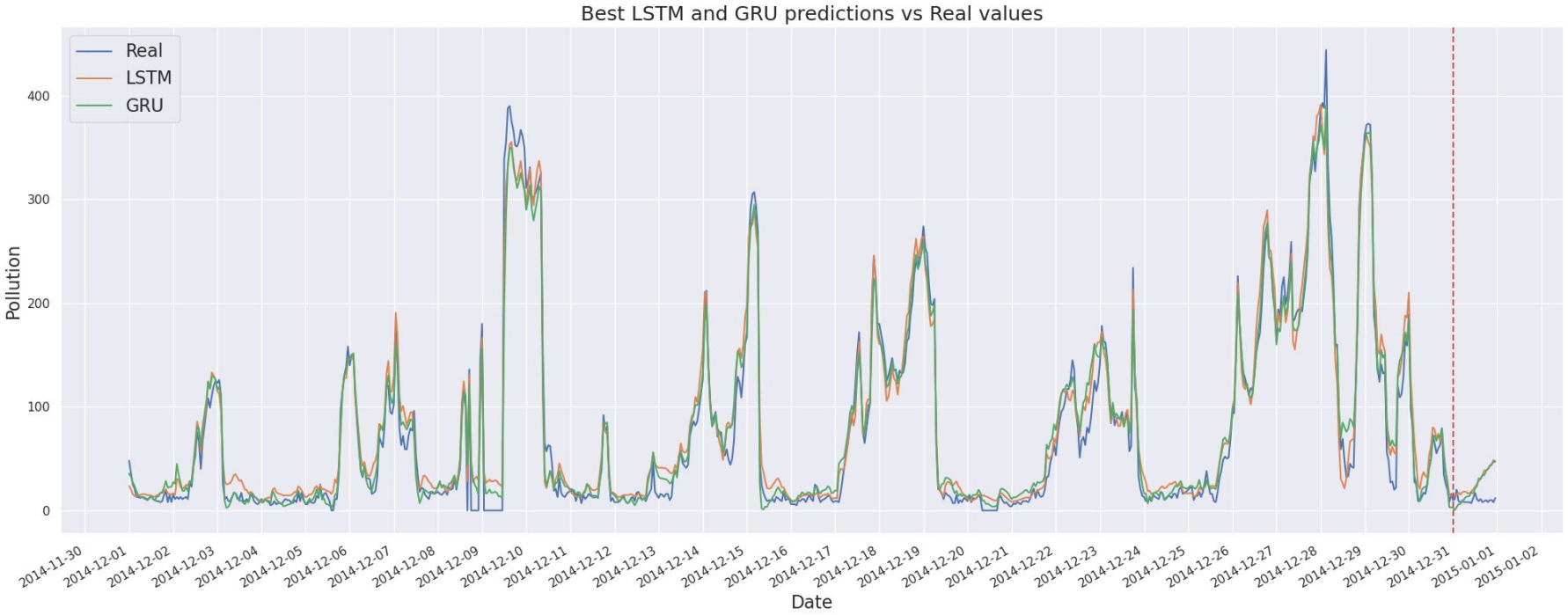
LSTM - stopped after 31 epochs

GRU - stopped after 30 epochs

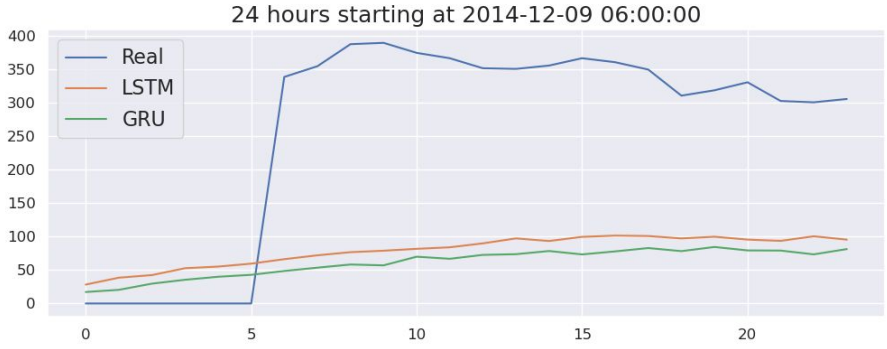


Model comparison

Comparing predictions with the real values

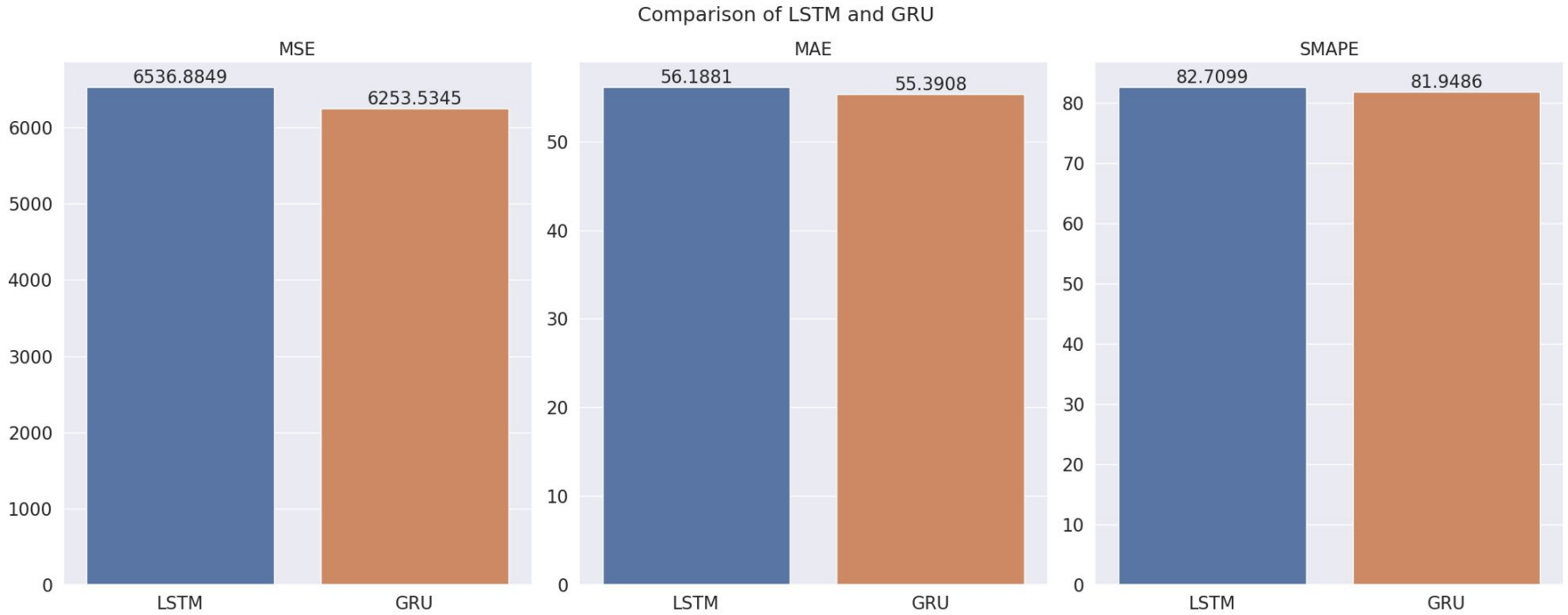


Comparing predictions with the real values



Comparison of chosen performance metrics

$$\text{SMAPE} = \frac{1}{n}\sum_{t=1}^{n}\frac{|F_t - A_t|}{(A_t + F_t)/2}$$



Comparing training time

For the comparison to be fair, I created a “twin” GRU model with the same hyperparameters as LSTM. It was trained for the same number of epochs as LSTM (31 epochs). Surprisingly, “twin” GRU model was slower.

- **Best LSTM:** 6min 30s
- **Twin GRU:** 9min 22s
- **Best GRU:** 5min 55s

Related research paper

(2021) “**PM2.5 concentration forecasting at surface monitoring sites using GRU neural network based on empirical mode decomposition**”, Guoyan Huang, Xinyi Li, Bing Zhang, Jiadong Ren

- PM2.5 concentration was decomposed and fed into a GRU model along with meteorological features
- Features included dew point, historical PM2.5, temperature, air pressure, wind direction and wind speed

Table 5
PM2.5 prediction error results under the each of different feature selections.

Parameter name	Value
Training set	35,040
Test set	8756
Number of GRU units	200&100
Batch size	128
Loss	MAE
Optimizer	Adam
Epochs	30
Sample weight mode	1D
Dropout	0.2
Learning rate	0.001

Table 6
The EMD-GRU model evaluation index under the setting of different time steps.

	Method	Parameter setting	RMSE	MAE	SMAPE(%)	R-square
ML	SVM	Kernel = RBF C = 14, gamma = 0.01	30.627±0.000	23.346±0.000	39.790±0.000	0.8934±0.0000
	DTR	Criterion = ‘mse’ Max depth = 3, Max leaf nodes = 8	26.299±0.000	16.578±0.000	26.553±0.000	0.9117±0.0000
	GBDT	Loss = ‘huber’ Min samples split = 2, Learning rate = 0.1	20.872±0.028	11.447±0.012	17.188±0.001	0.9478±0.0001
	RF	n estimators = 50 Max depth = 6 Min samples split = 3	24.841±0.175	15.188±0.162	24.773±0.175	0.9210±0.0011
DL	RNN	See Table 5	21.225±0.044	11.358±0.098	17.357±0.044	0.9488±0.0002
	LSTM	See Table 5	20.872±0.038	11.184±0.023	16.759±0.116	0.9506±0.0006
	GRU	See Table 5	20.309±0.053	11.039±0.049	16.758±0.227	0.9531±0.0002
Proposed	EMD-GRU	See Table 5	11.372±0.145	6.532±0.073	14.809±0.646	0.9852±0.0004

Thank you for your attention!