Predictive Process Monitoring

Eric Näser

Munich, 17.01.2023

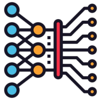
Table of Contents

1. Introduction of Data and Tasks

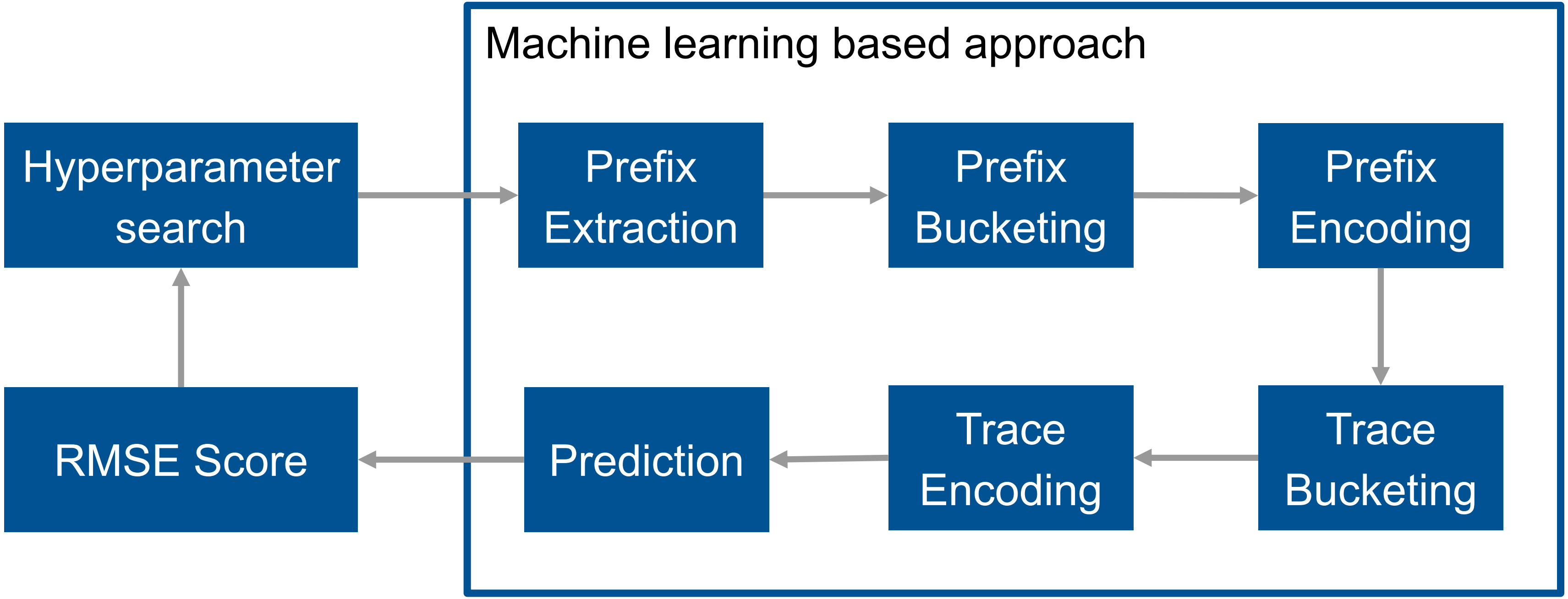
Data

- Sepsis_Cases_1
- Event log with treatments of patients with scepsis symptoms in hospital
- Number of cases: 782
- Number of activities: 15
- Trace duration between 3 hours and 114 days with an average of 7.5 days

Tasks

- Analyse data 
- Machine learning based remaining time prediction 
- Improve explainability 
- Deep learning based remaining time prediction 

2. Hyperparameter Search with Ray Tune



2. Hyperparameter Search with Ray Tune

Table representation of search space

Parameter	Distribution	Values
temporal_split_sort_by	Choice	x ∈ ["timesincecasestart", "time:timestamp"]
bucketing_technique	Grid	x ∈ ["SingleBucket", "PrefixLength", "Clustering"]
bucketing_upper_bound	Uniform integer	x ∈ [2, 20]
…	…	…

Code representation of search space

```
{
    "temporal_split_sort_by": tune.choice(
    ["timesincecasestart", "time:timestamp"]),

    "bucketing_technique": tune.grid_search(
    ["SingleBucket", "PrefixLength", "Clustering"]),

    "bucketing_lower_bound": tune.randint(2, 20),
    …
}
```

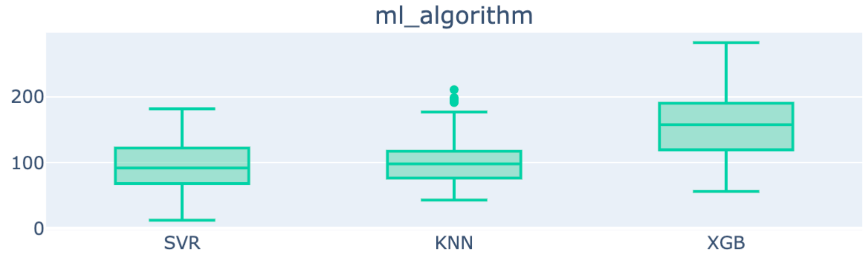
3. Visualizations with Plotly for Explainability

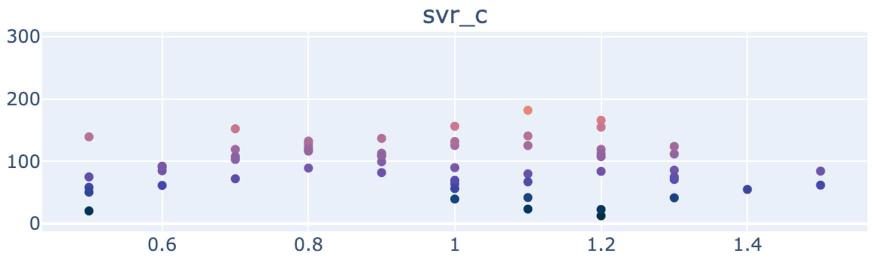
Plotly is an:

- Free
- Interactive
- Well documented

Graphing library with publication-quality graphs

Examples





4. Deep learning with Pytorch Lightning

Pytorch Lightning is a

- Relatively simple
- Flexible
- Well documented

Deep learning framework

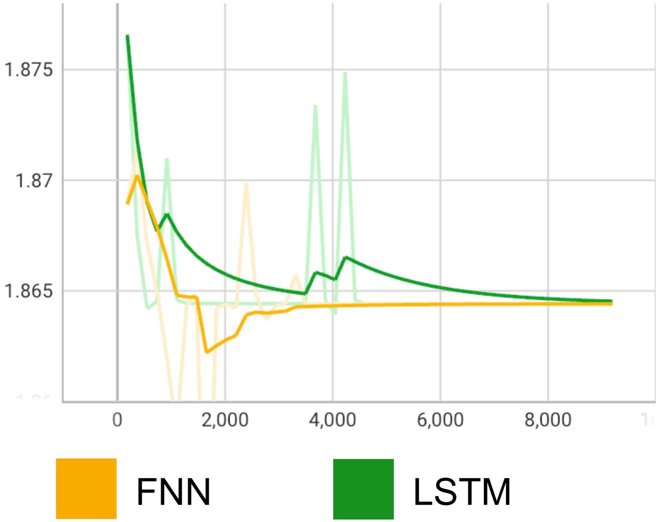
Easy implementation of neural networks

```
self.dfnn = nn.Sequential(
    nn.Linear(self.input_dim, self.hidden_dim1),
    nn.BatchNorm1d(self.hidden_dim1),
    nn.ReLU(),
    nn.Dropout(self.dropout),

    nn.Linear(self.hidden_dim, self.hidden_dim2),
    nn.BatchNorm1d(self.hidden_dim2),
    nn.ReLU(),
    nn.Dropout(self.dropout),
    …
```

5. Challenges: Develop Neural Networks that learn

Problem



Solution

- Keep it simple at first
- Stick to best practices instead to papers at first
- Revisit slides of Introduction to Deep Learning
- Visualize predictions with Tensorboard

5. Challenges: Common data structure for prefixes

Problem

- Machine learning and deep learning based approach follow similar structure
- Different techniques can be used for same steps in approaches
- Code should be clear, maintainable and reusable
- Data structure for storing prefixes should be the same for all settings

Solution

- Reordering of steps in approaches
- Idea testing with pen and paper

	ER	CRP	LEU	Remaining Time
2:	1	1	0	0.4
	0	1	1	-0.1
	1	1	0	0.2

Prefix Length / Cluster

Dataframe

5. Challenges: Understanding key concepts

Problem

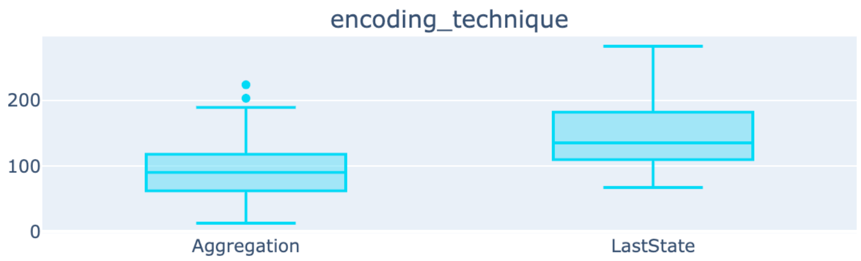
- What is prefix extraction?
- What is prefix bucketing?
- What is last state encoding?
- What is aggregation encoding?
- …

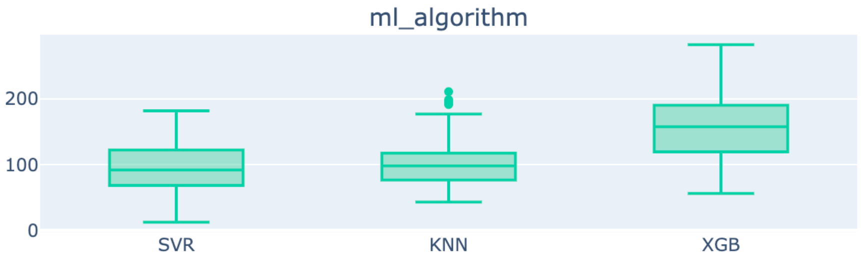
Solution

- Reading multiple sources about concepts
- Finding explanations with helpful visualizations
- Asking the same questions multiple times during consultations ☺

6. Results: Best Machine Learning Approaches

RMSE	MAE	Prefix length	Bucketing technqiue	Encoding technique	ML algorithm
0.008	761.00	14	Single Bucket	Aggregation	SVM
0.009	1366.20	10	Prefix Length	Aggregation	SVM
0.018	1227.12	17	Prefix Length	Aggregation	SVM





6. Results: Deep learning architectures

Feedforward Neural Network architecture

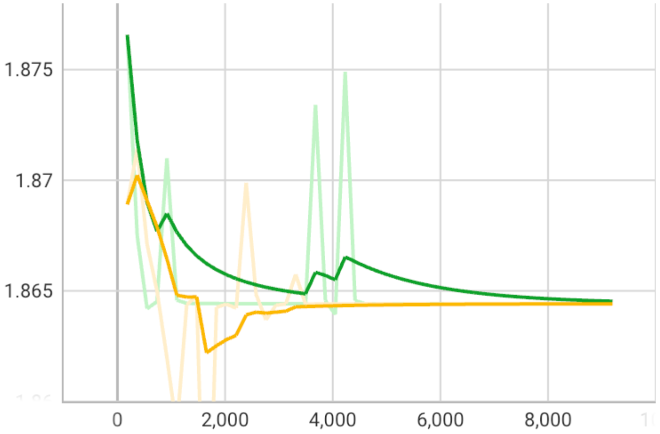
Parameter	Value
# layers	5
# neurons per layer	[input_dim, hidden _dim, hidden_dim * 1.2, hidden_dim * 0.6, hidden_dim * 0.3, output_dim]
# dropout layer	5
Activation functions	[relu, relu, relu, relu, tanh]

Long Short Term Memory architecture

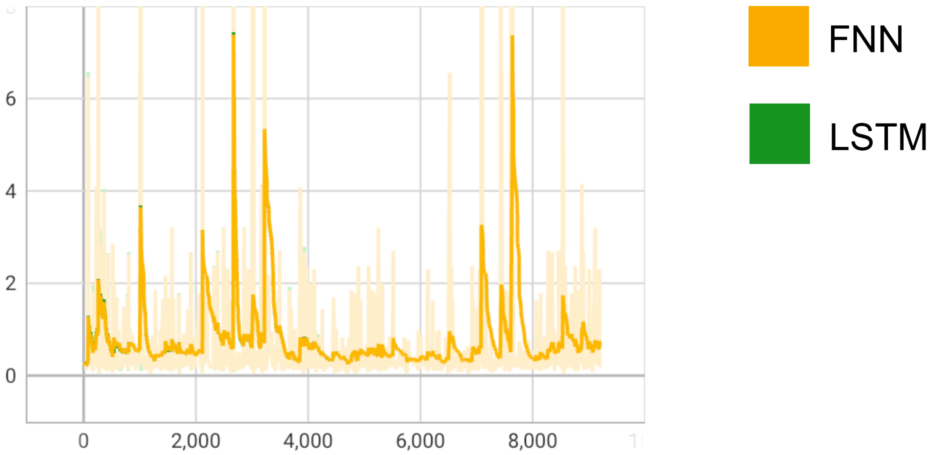
Parameter	Value
# layers	[2 lstm, 2 fully connected]
# neurons per layer	[input_dim, hidden_dim, hidden_dim, hidden_dim * 0.3, output_dim]
# dropout layer	2
Activation functions	[relu, tanh]

6. Results: Deep learning with frequency encoding

Validation loss



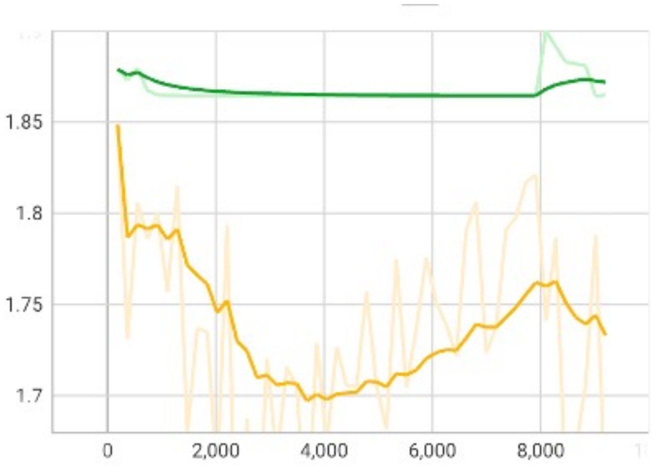
Test loss



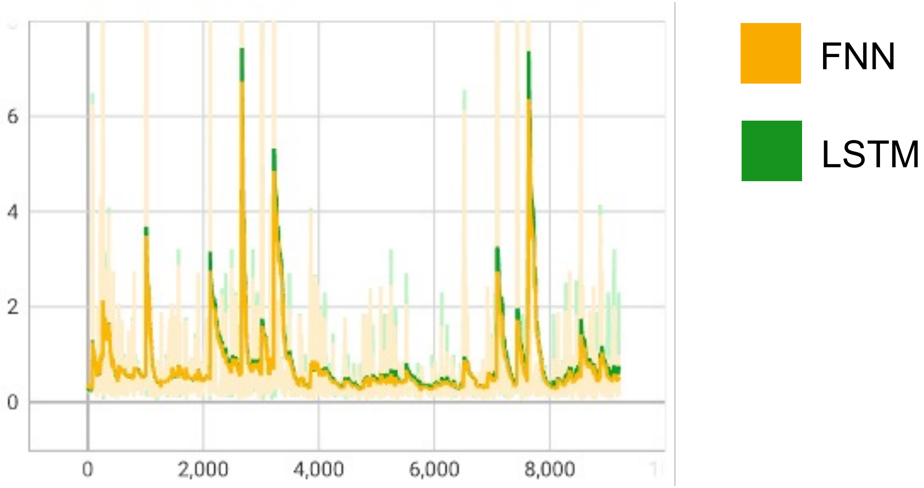
→ No learning of remaining time prediction evident, bad RMSE and MAE scores

6. Results: Deep learning with Batch Normalization

Validation loss



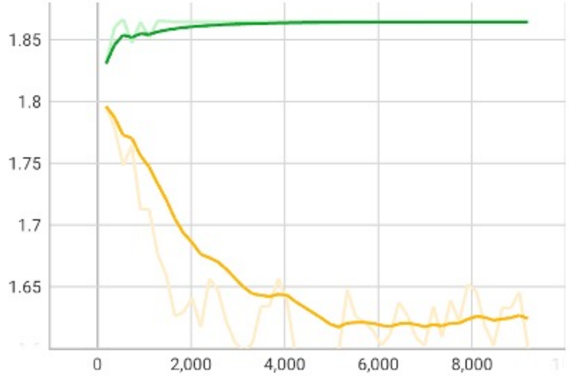
Test loss



→ Improved results for Feedforward neural network

6. Results: Further experiments with deep learning

Decrease in number of layers



→ Improved results for Feed-forward neural network

Experiments without improvements

- Decrease in batch size (50 → 20)
- Increase in batch size (50 → 70)
- Decrease in size of „hidden_dim“ (100 → 50)
- Increase in size of „hidden_dim“ (100 → 150)
- One-Hot encoding for event encoding

FNN

LSTM

6. Results: Summary

Best results per approach

Approach	RMSE	MAE
Machine Learning	0.02	762.94
Feedforward Neural Network	5.64	16677.14
Long Short Term Memory	7.14	19190.72

Key takeaways

- Machine learning based approach outperformed deep learning based approach
- Single bucketing, aggregation encoding and Support Vector Machines were main contributors to success of machine learning
- Results of Feedforward Neural Network were improved, but still not useful
- Long Short Term Memories tended to not learn anything