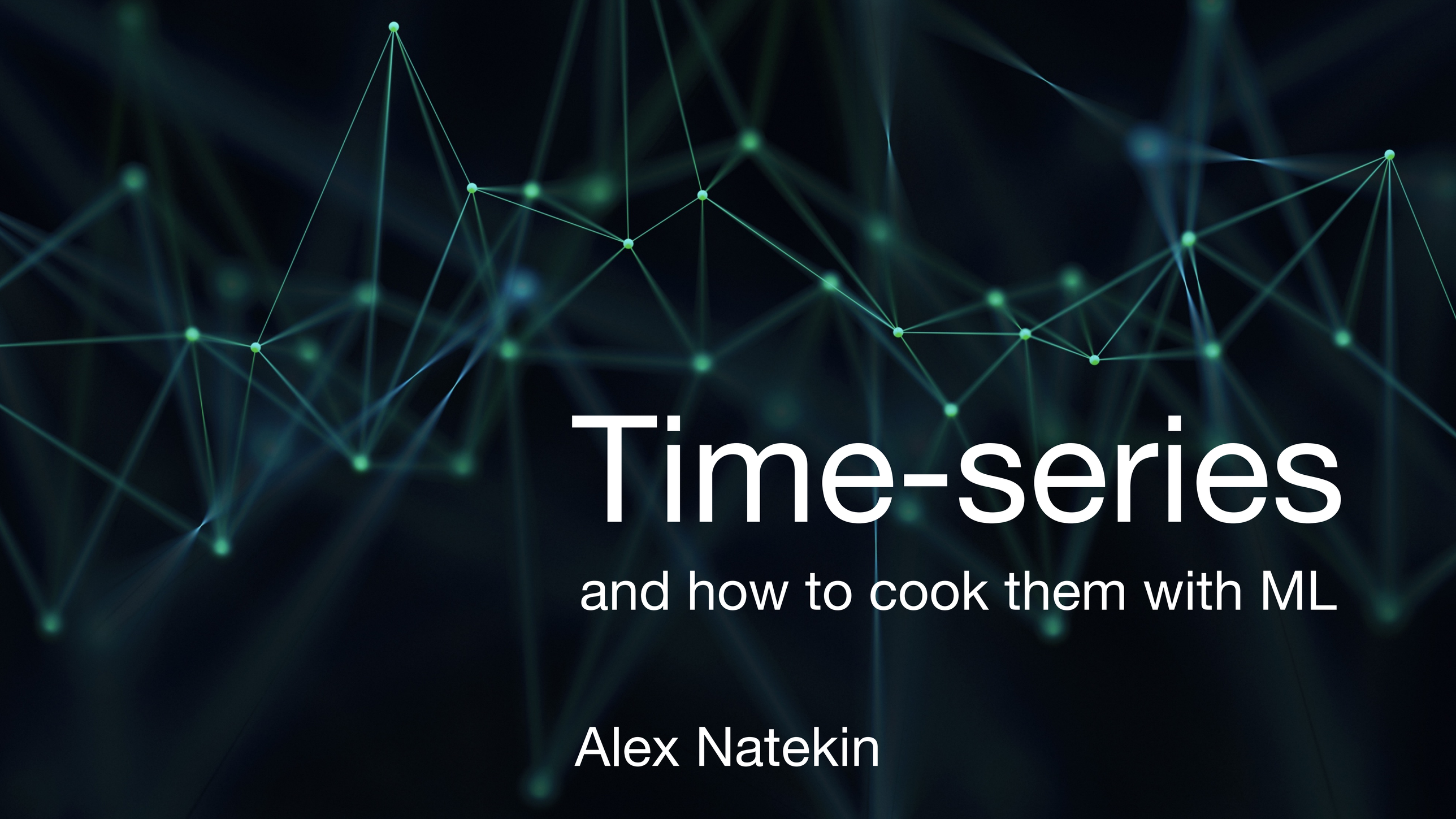
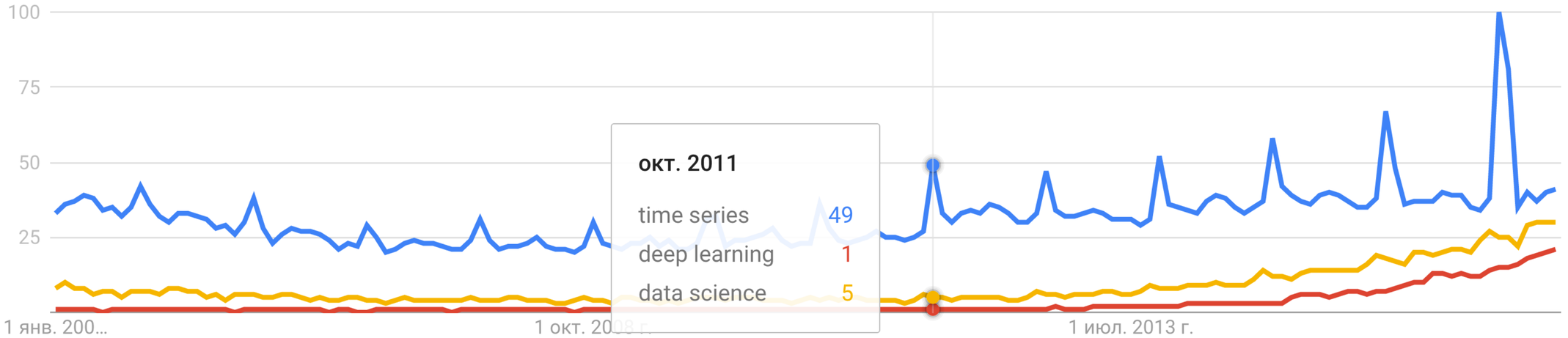

Time-series
and how to cook them with ML
Alex Natekin

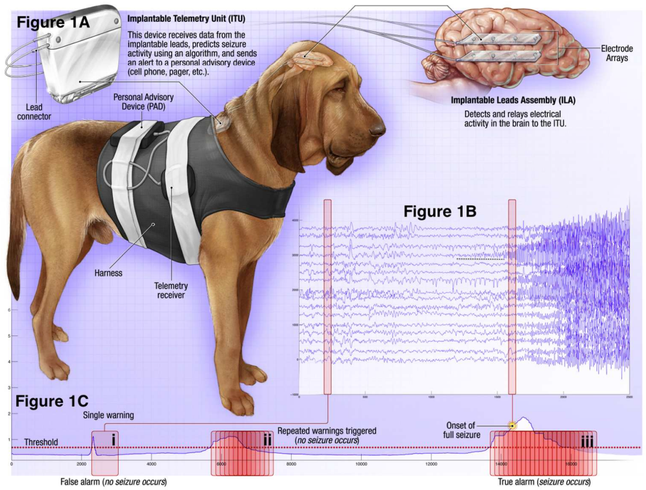
Time-series

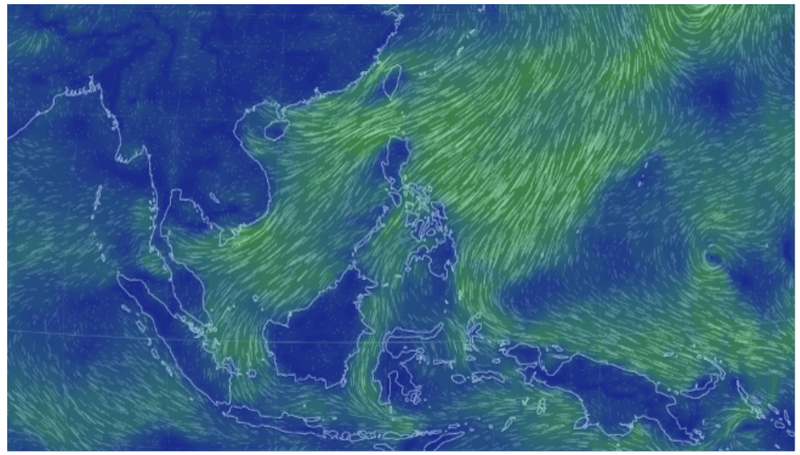
- “Series of data points indexed in time order”
- “Collection of observations of well-defined data items obtained through repeated measurements over time”

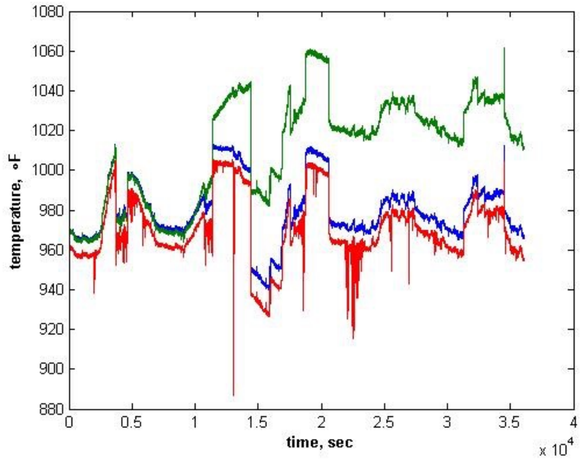
Date	UserId	Sum	Consumed (ml)	Places
2016-12-01	10780	800.00	68	“Outside”
2016-12-05	10780	120.00	60	“Inside”
2016-12-09	10780	145.00	62	“Inside”
2016-12-14	10780	1600.00	145	“Both”
2016-12-19	10780	900.00	71	“Outside”
2016-12-24	10780	600.00	65	“Inside”
2016-12-25	10780	666.00	NA	“Inside”
2016-12-26	10780	1200.00	NA	“Outside”
2016-12-27	10780	1400.00	125	“Outside”
2016-12-28	10780	2300.00	NA	“Both”

Time-series examples







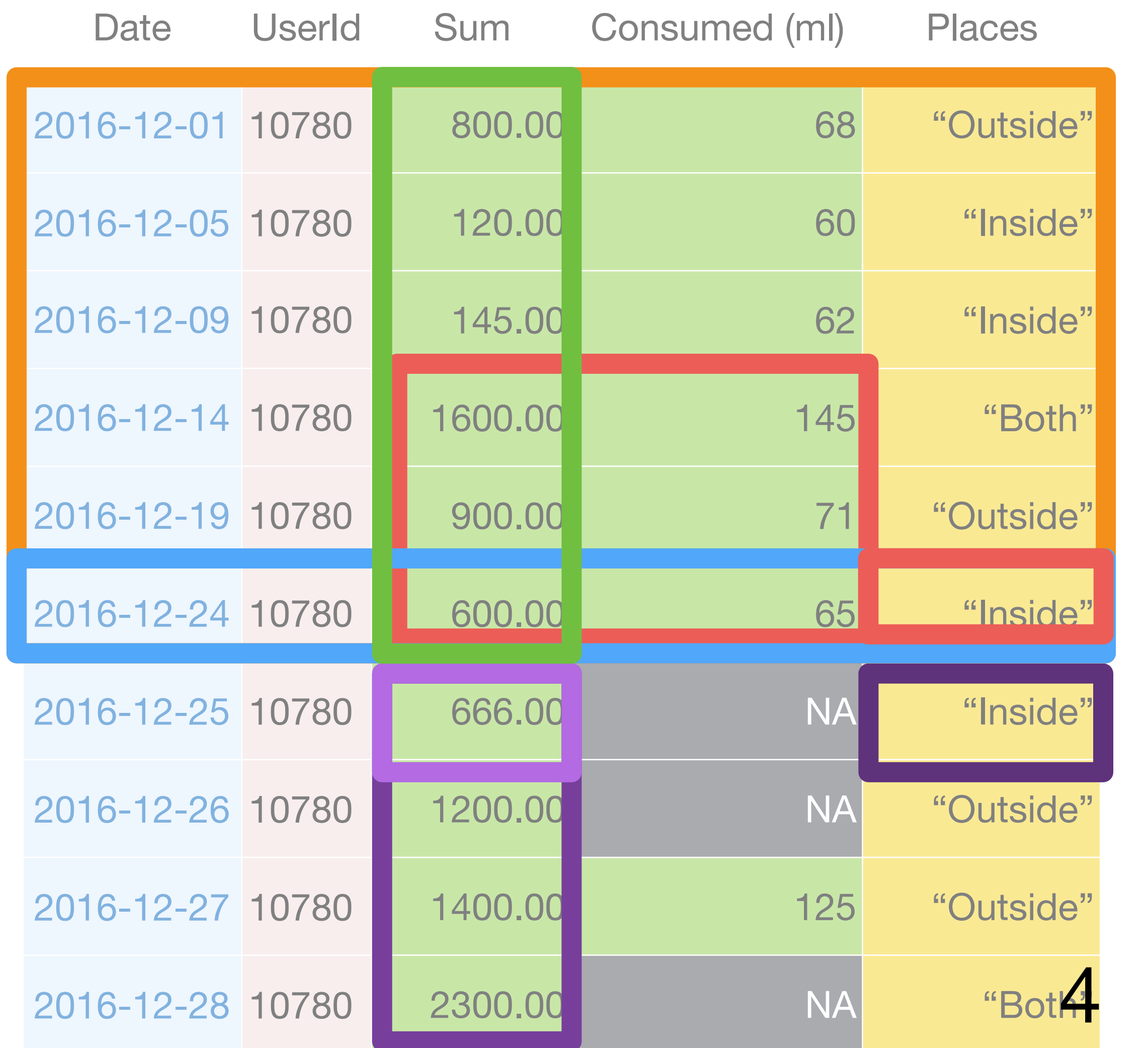


Much prior work

- ARIMA, GARCH, others from statistical school
- Wavelets and other signal processing approaches
- RNN/LSTM get hype
- Casual Kaggle-style ML, …

Forecasting

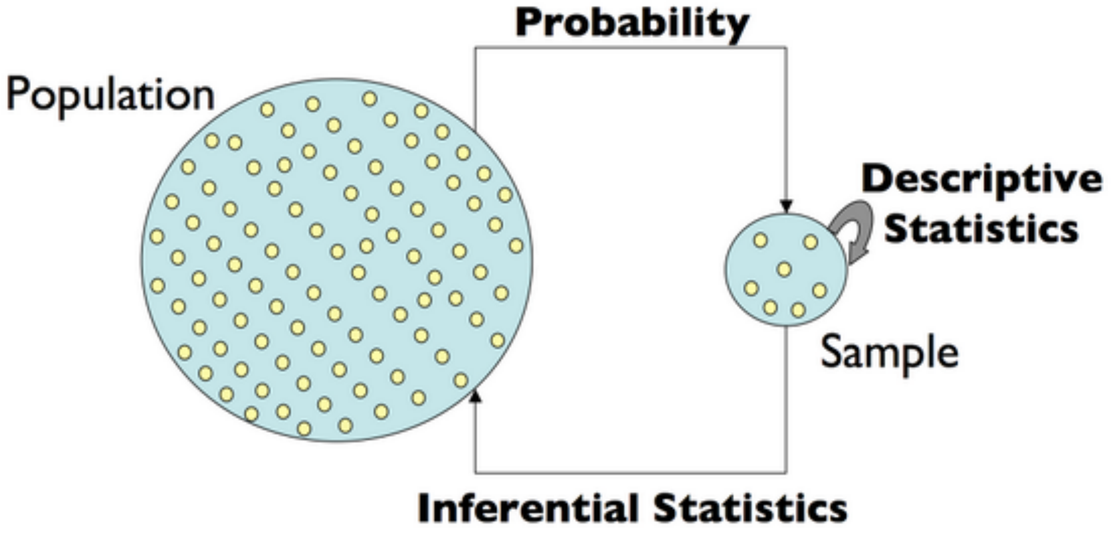
- Choose what to predict:
  value/aggregate/class
- Guess the dependency:
  univariate/naive/all/mixed
- Proceed with setup of other ML problem components



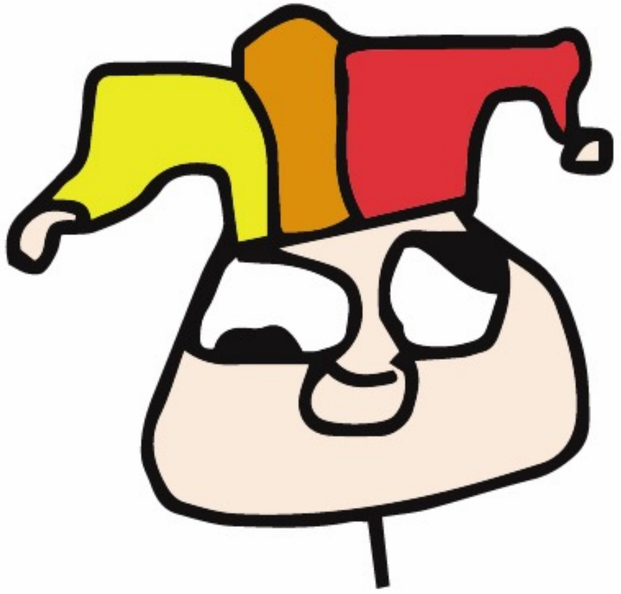
Date	UserId	Sum	Consumed (ml)	Places
2016-12-01	10780	800.00	68	“Outside”
2016-12-05	10780	120.00	60	“Inside”
2016-12-09	10780	145.00	62	“Inside”
2016-12-14	10780	1600.00	145	“Both”
2016-12-19	10780	900.00	71	“Outside”
2016-12-24	10780	600.00	65	“Inside”
2016-12-25	10780	666.00	NA	“Inside”
2016-12-26	10780	1200.00	NA	“Outside”
2016-12-27	10780	1400.00	125	“Outside”
2016-12-28	10780	2300.00	NA	“Both”

ML models for time-series

Central dogma of Statistics:



Central dogma of Machine Learning:



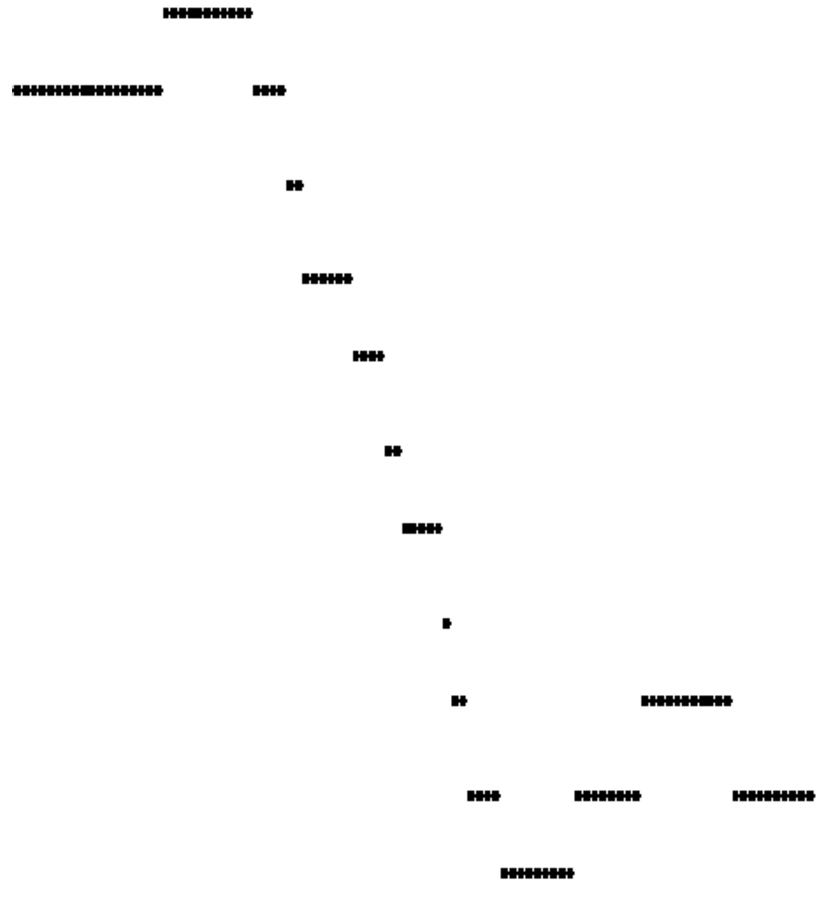
Everything is a feature

Time-series ML pipeline

Data & problem	Data preparation	Feature engineering	Model	Evaluation
Value vs diff	Consistent spacing	**Stationarised series**	Linear	**Time-CV**
Regression/ classification /anomalies	Impute missing values	AR, MA and aggregates	Casual ML & boosting	Test-policy (rolling or not)
Univariate sensors vs panel data	Aggregates/ pooling	Dense represent-ns	LSTM & RNN	Simulations (small data)
		Kaggle-style	Markov-style	
			Kaggle-style	

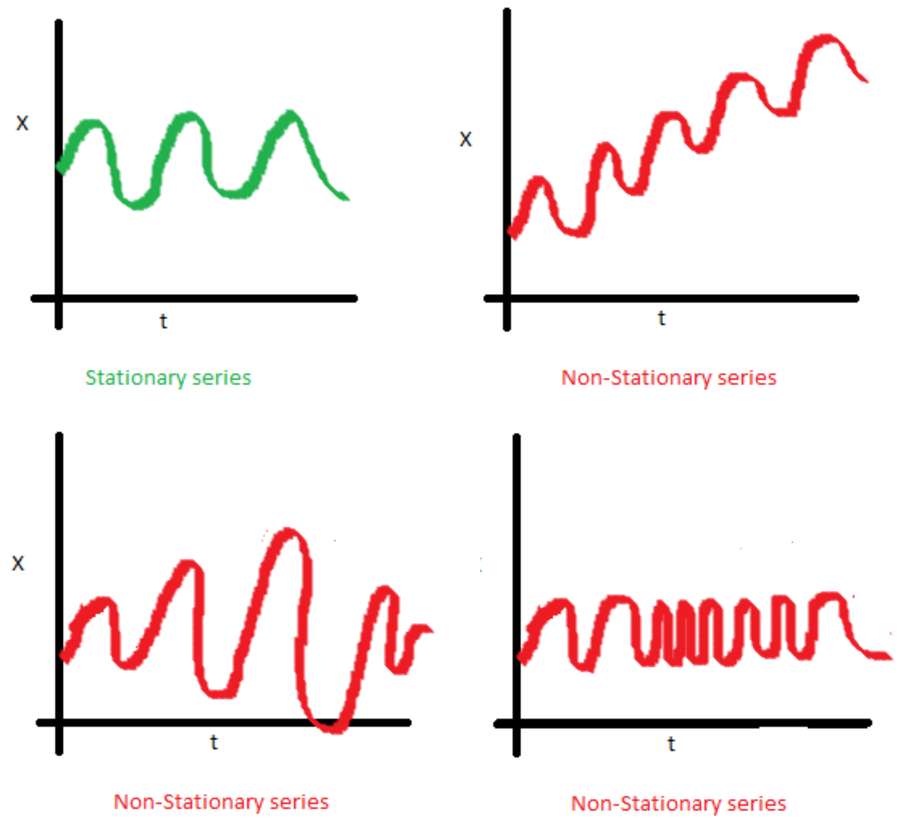
TS data preparation

- Check spacing. If non-uniform, choose time grid (and fill missing)
- Missing values: zero vs LOCF (last observation carried forward) vs interpolation
- Downsample: use minute-level aggregates instead of seconds …



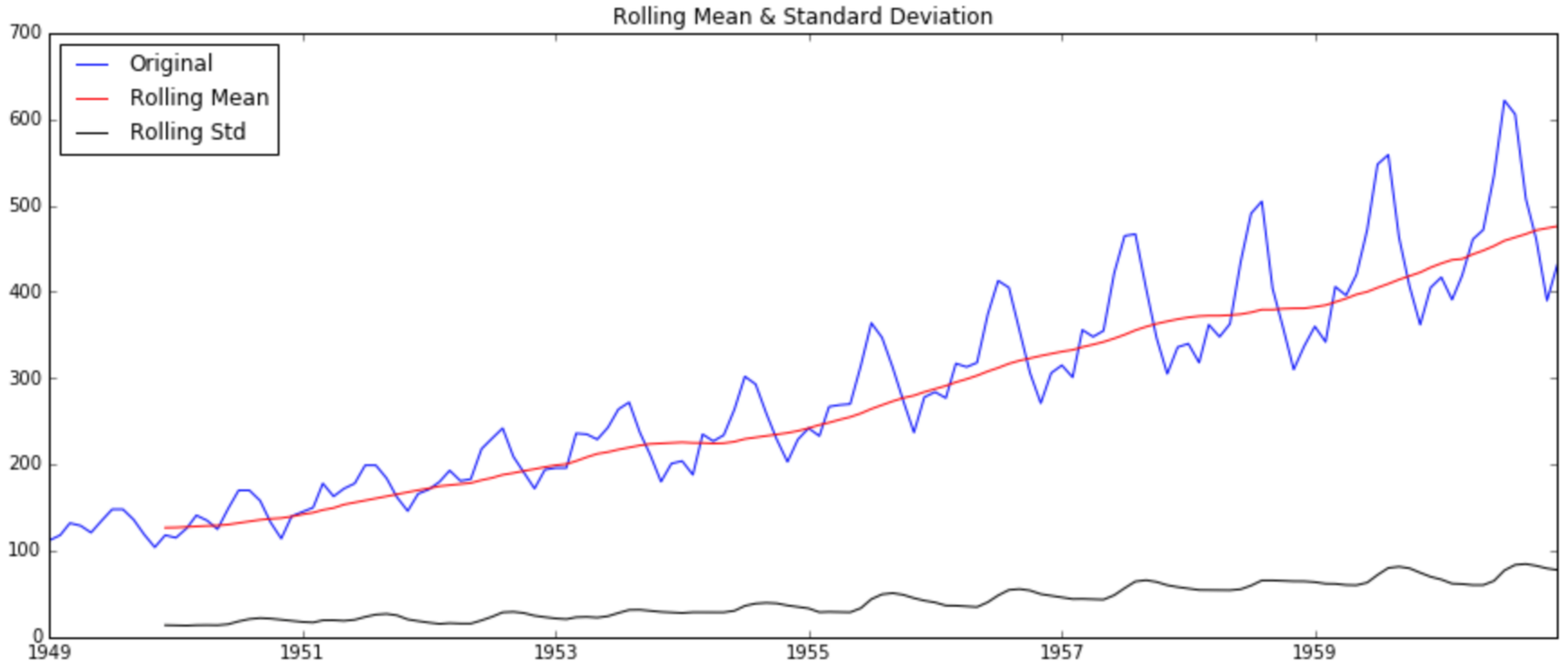
TS feature engineering (1/5)

- Stationarization: detrending & differencing (can be repeated)
- Log-transform
- **NB:** you can keep both original and diff ones for models



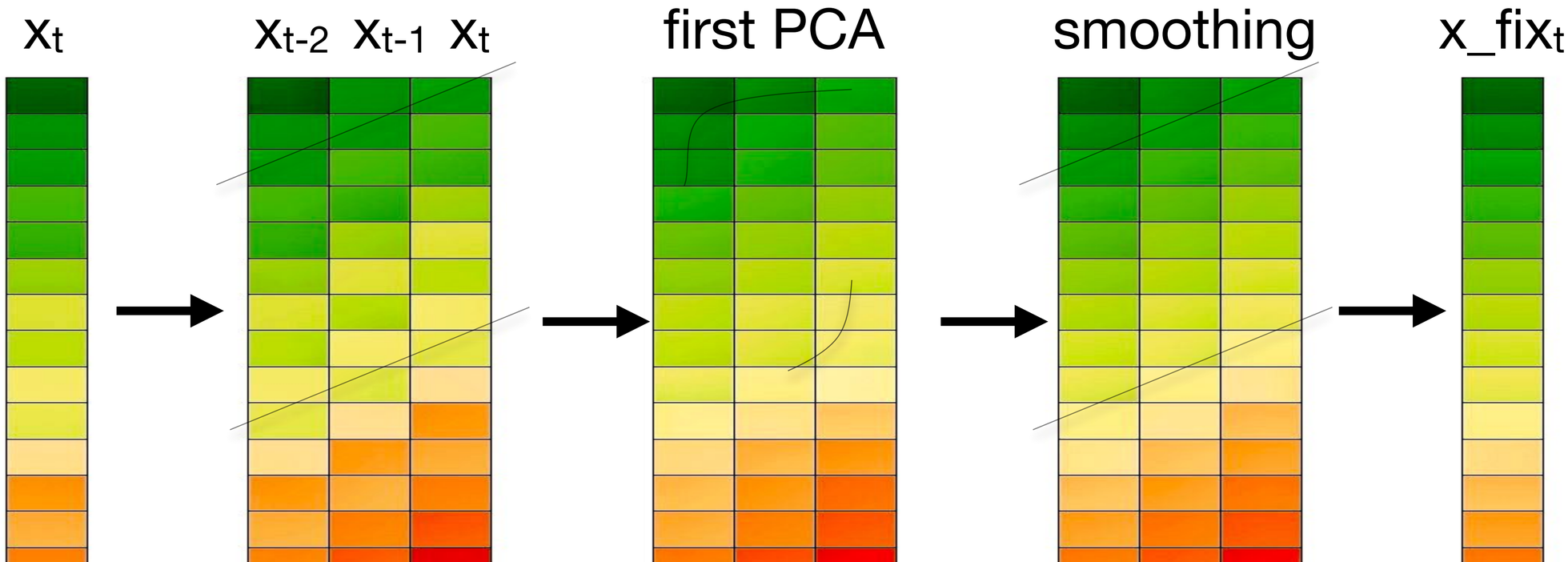
TS feature engineering (2/5)

- Rolling statistics (mean, sd, min, max, …)
- Rolling aggregates (counts, incl. other features)



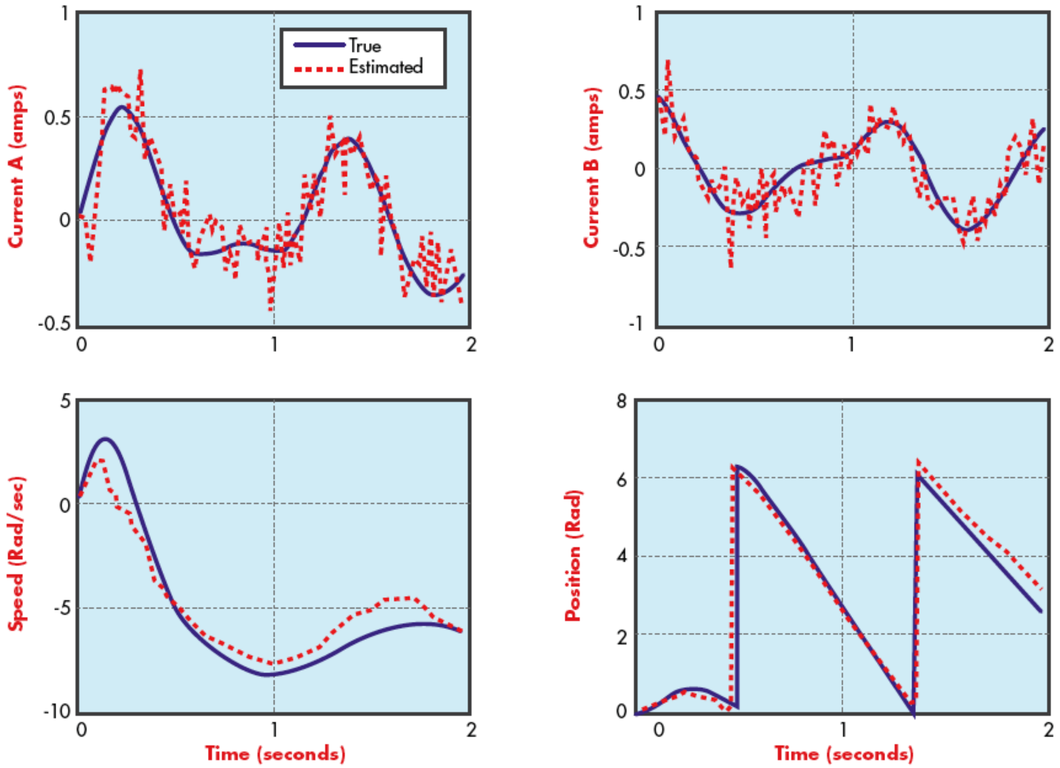
TS feature engineering (3/5)

- AR: lagged values for “auto-regression”
- SSA and respective goodies (trends, periodics, …)



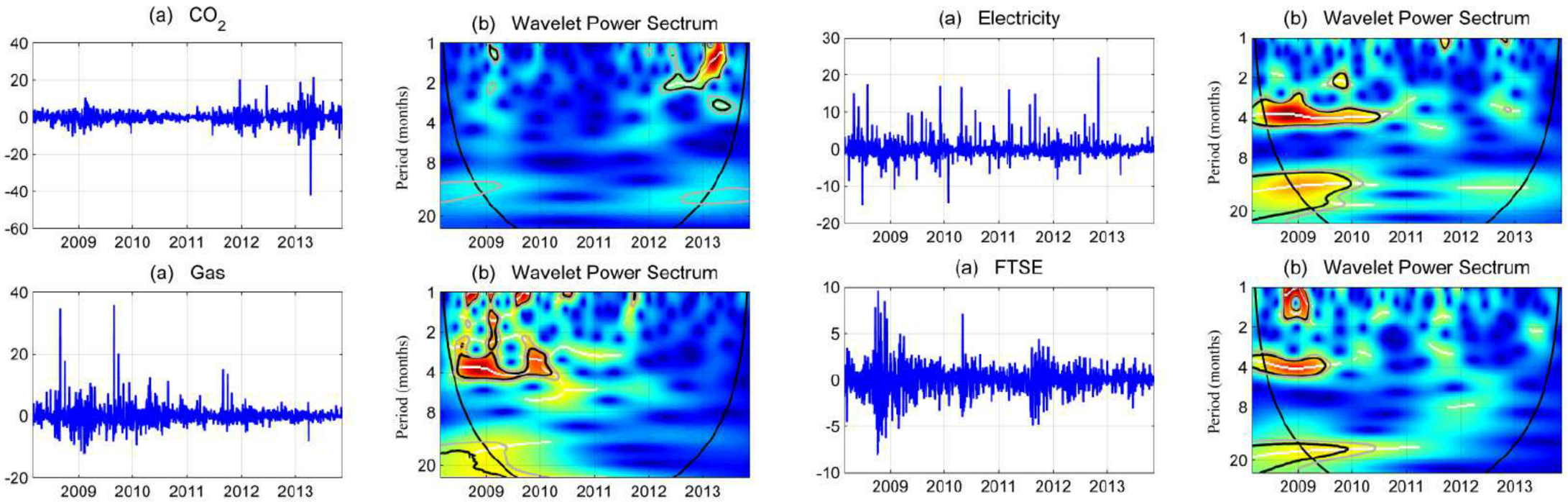
TS feature engineering (4/5)

- Exponential smoothing and other designed AR-functions
- Kalman filter family
- Robust regression filters

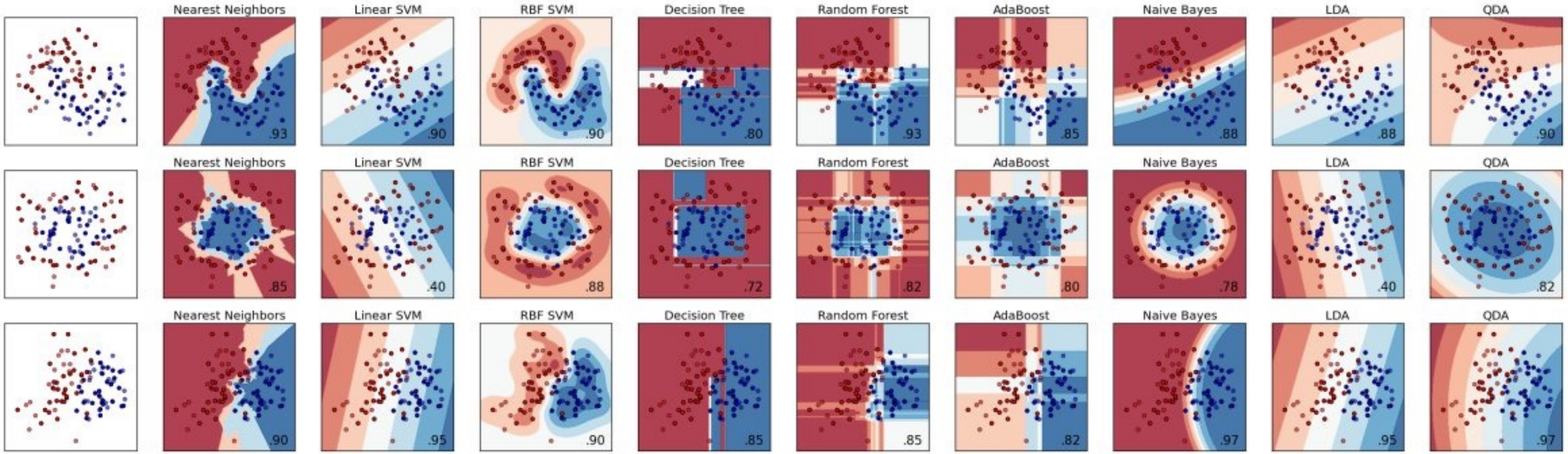


TS feature engineering (5/5)

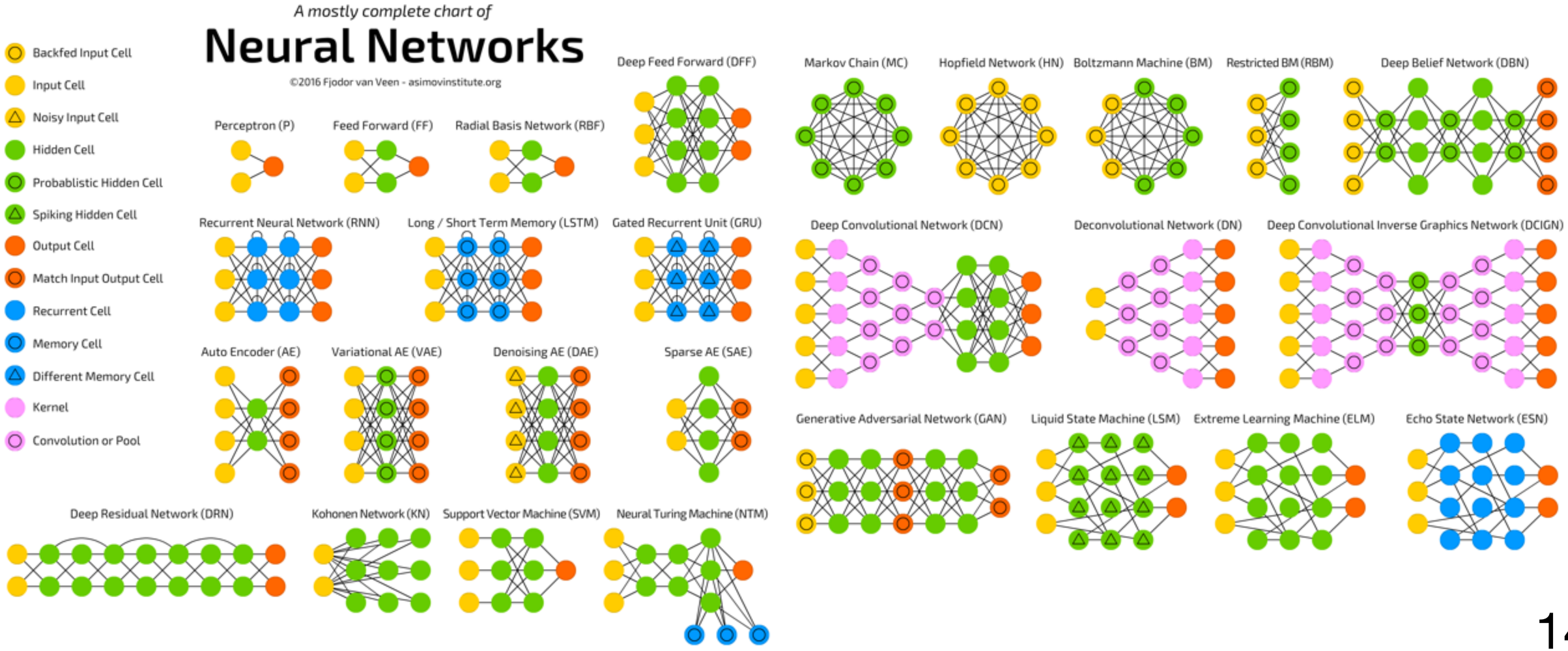
- Fourier and wavelet power spectrum

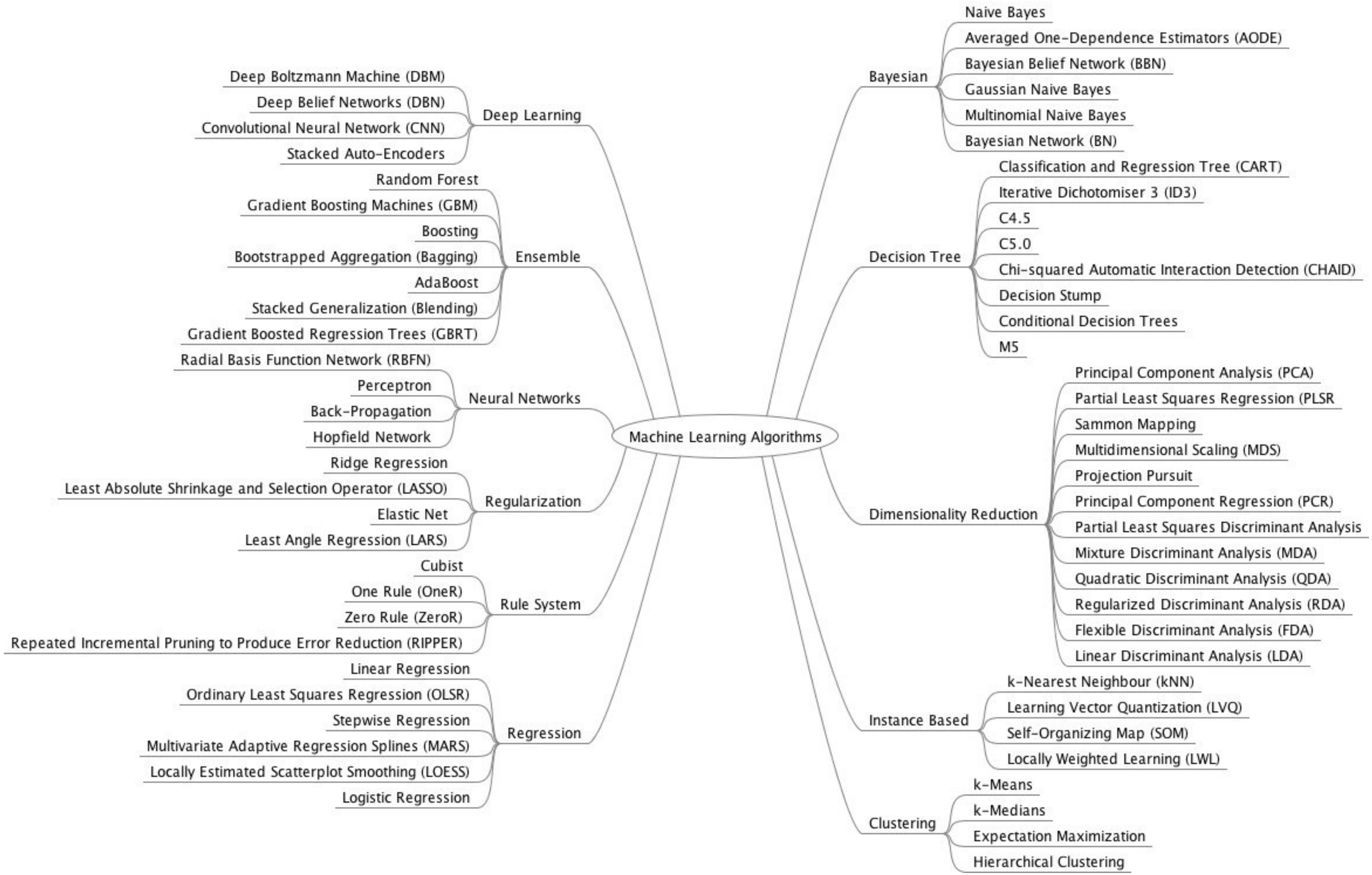


ML models vary a lot, there is no free lunch

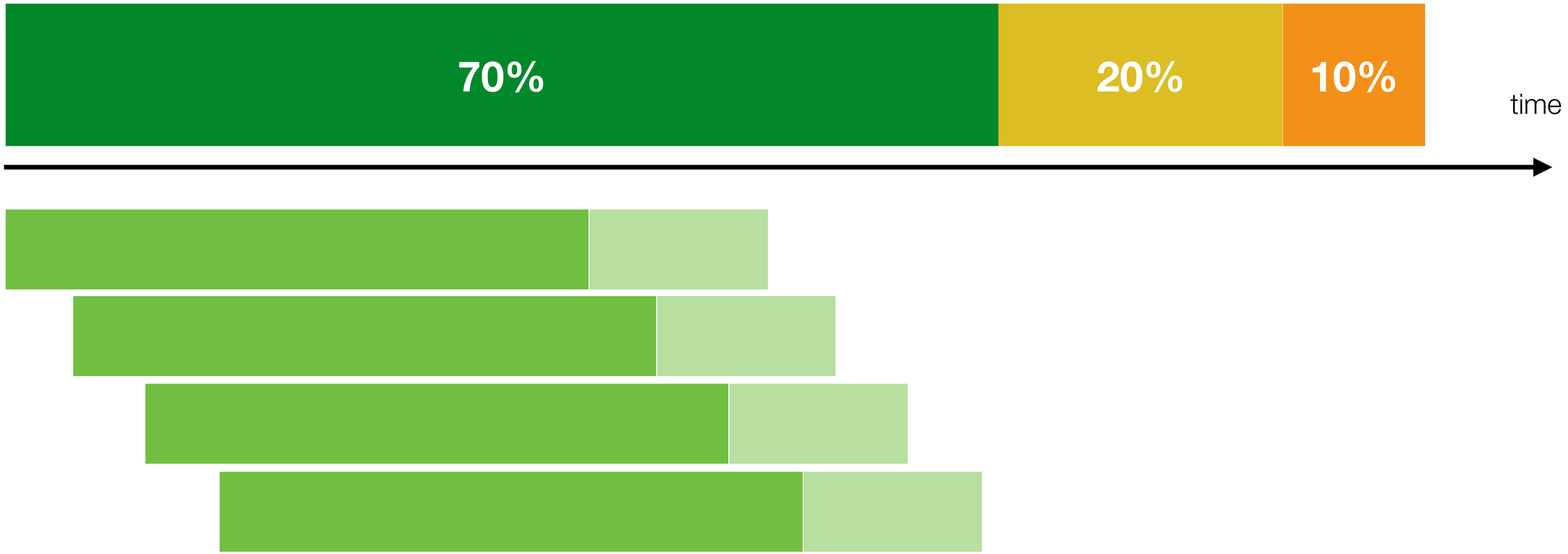


NN - one of many TS options:

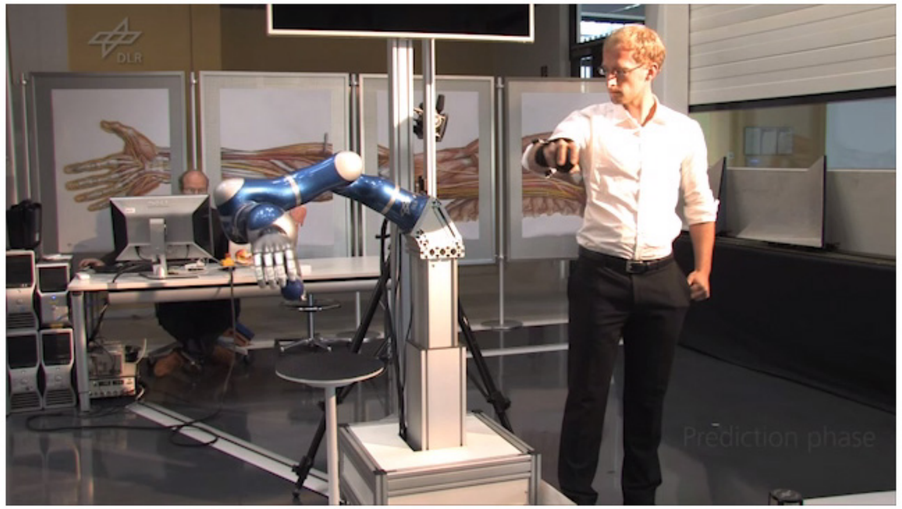


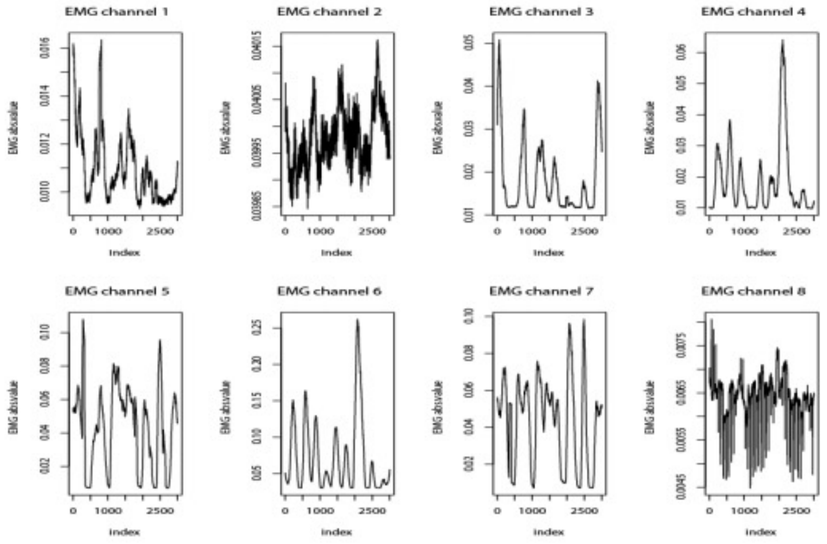

Machine Learning Algorithms
Bayesian
Naive Bayes
Averaged One-Dependence Estimators (AODE)
Bayesian Belief Network (BBN)
Gaussian Naive Bayes
Multinomial Naive Bayes
Bayesian Network (BN)
Decision Tree
Classification and Regression Tree (CART)
Iterative Dichotomiser 3 (ID3)
C4.5
C5.0
Chi-squared Automatic Interaction Detection (CHAID)
Decision Stump
Conditional Decision Trees
M5
Dimensionality Reduction
Principal Component Analysis (PCA)
Partial Least Squares Regression (PLSR
Sammon Mapping
Multidimensional Scaling (MDS)
Projection Pursuit
Principal Component Regression (PCR)
Partial Least Squares Discriminant Analysis
Mixture Discriminant Analysis (MDA)
Quadratic Discriminant Analysis (QDA)
Regularized Discriminant Analysis (RDA)
Flexible Discriminant Analysis (FDA)
Linear Discriminant Analysis (LDA)
Instance Based
k-Nearest Neighbour (kNN)
Learning Vector Quantization (LVQ)
Self-Organizing Map (SOM)
Locally Weighted Learning (LWL)
Clustering
k-Means
k-Medians
Expectation Maximization
Hierarchical Clustering
Deep Learning
Deep Boltzmann Machine (DBM)
Deep Belief Networks (DBN)
Convolutional Neural Network (CNN)
Stacked Auto-Encoders
Ensemble
Random Forest
Gradient Boosting Machines (GBM)
Boosting
Bootstrapped Aggregation (Bagging)
AdaBoost
Stacked Generalization (Blending)
Gradient Boosted Regression Trees (GBRT)
Neural Networks
Radial Basis Function Network (RBFN)
Perceptron
Back-Propagation
Hopfield Network
Regularization
Ridge Regression
Least Absolute Shrinkage and Selection Operator (LASSO)
Elastic Net
Least Angle Regression (LARS)
Rule System
Cubist
One Rule (OneR)
Zero Rule (ZeroR)
Repeated Incremental Pruning to Produce Error Reduction (RIPPER)
Regression
Linear Regression
Ordinary Least Squares Regression (OLSR)
Stepwise Regression
Multivariate Adaptive Regression Splines (MARS)
Locally Estimated Scatterplot Smoothing (LOESS)
Logistic Regression

Time-series require time-CV. K-fold leads to leaks and fails

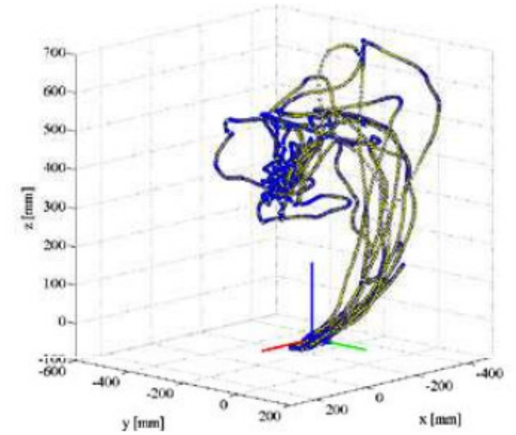


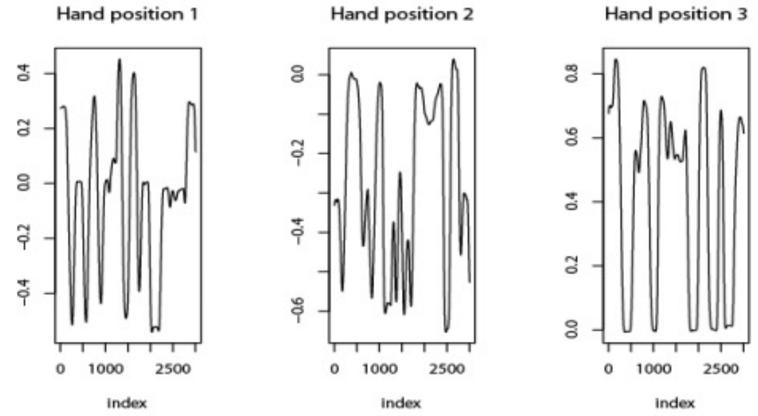
ML cases for time-series:





$$x \xrightarrow{f} y$$

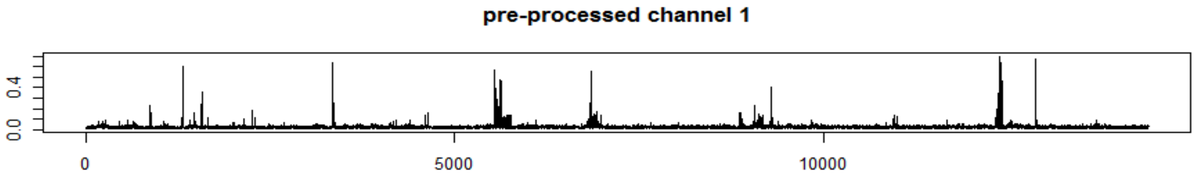


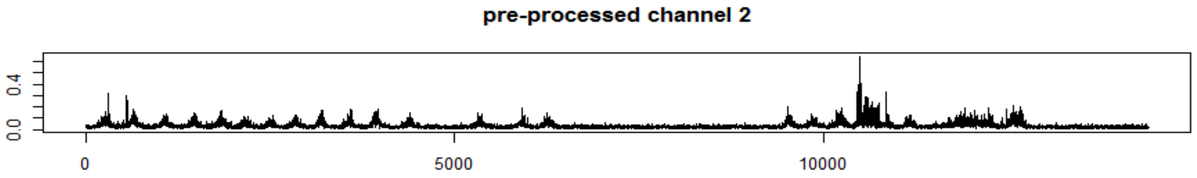


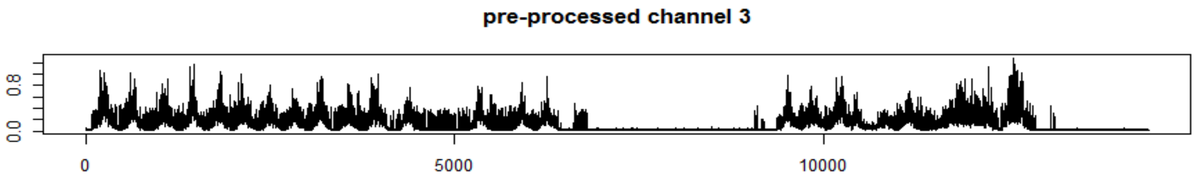
Case 1: ML pipeline

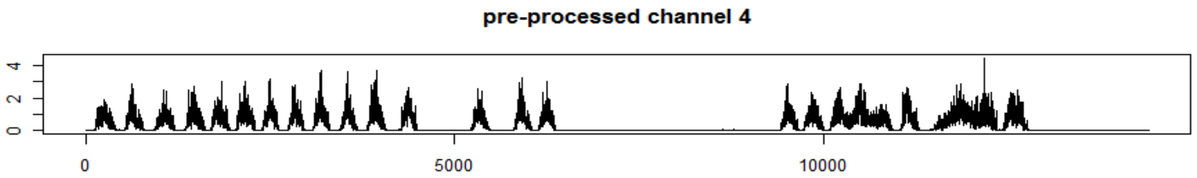
Data & problem	Data preparation	Feature engineering	Model	Evaluation
▸ 8 noisy EMG sensors ▸ Predict 3D position of the hand	▸ Target and input have different frequencies ▸ Choose to aggregate inputs by max in 10	▸ Stationary signals, but take diffs too ▸ Work on low frequencies, rolling stats are great	▸ Classic ML does great: even linear regression ▸ Cook GBM ▸ No need for RNN	▸ Split folds by sequences ▸ Add post-processing (smooth)

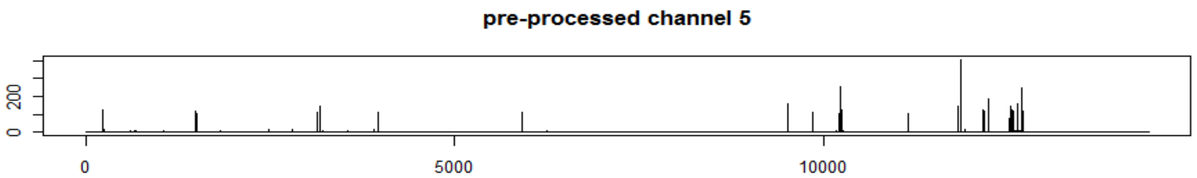
Case 1: input features

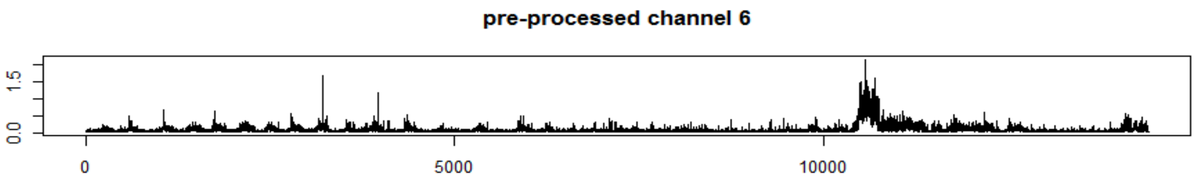


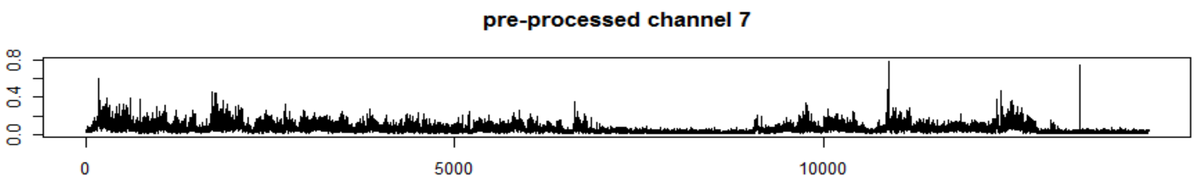


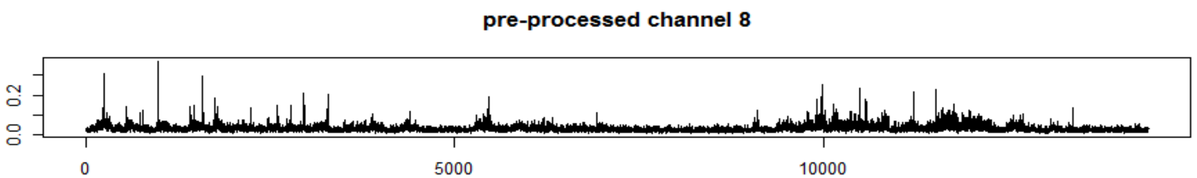






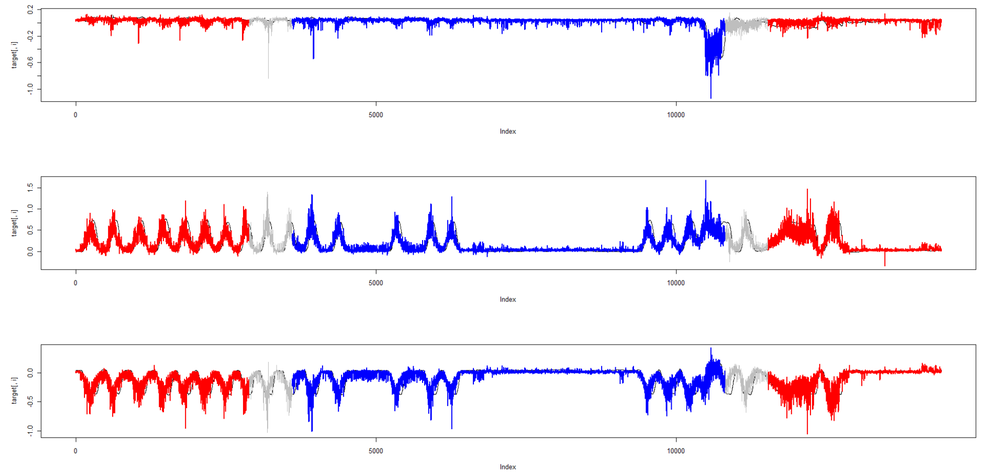




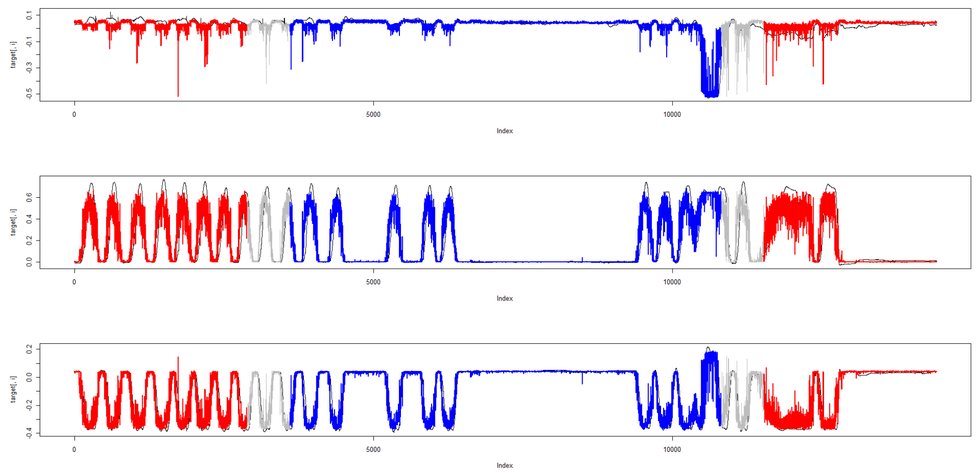


Case 1: naive baseline models

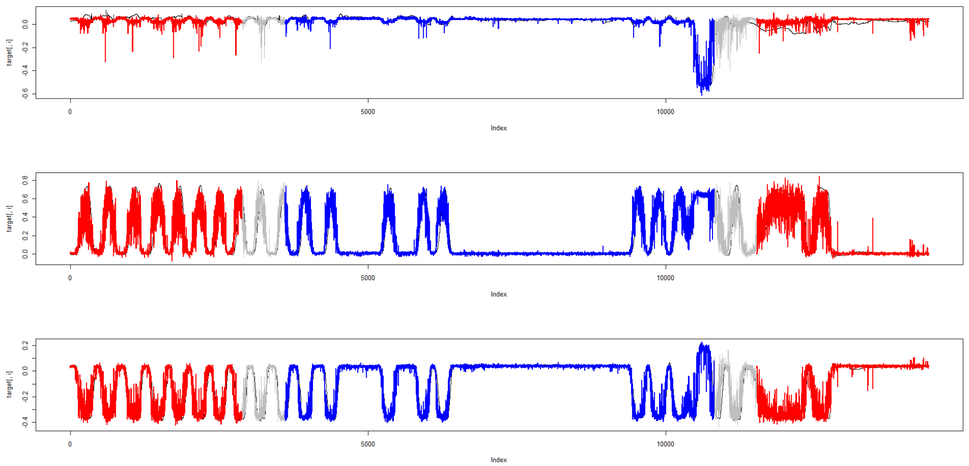
LR:



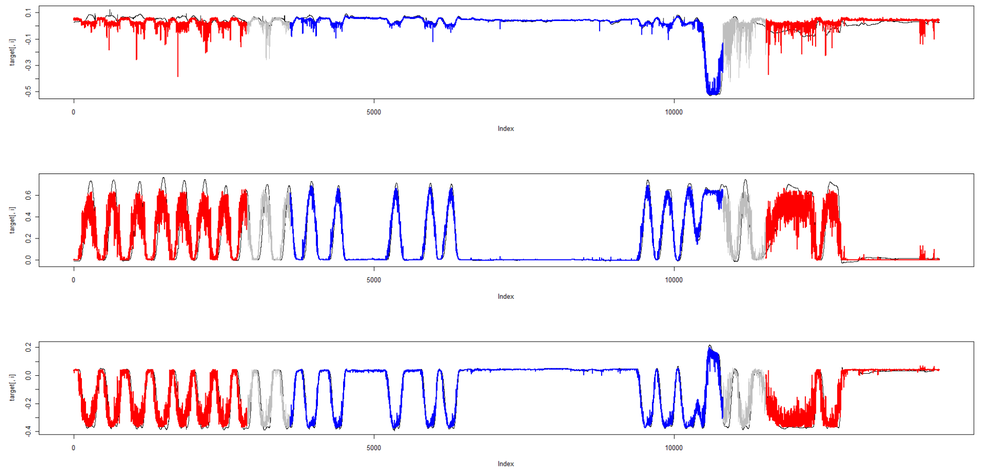
KNN:



SVM:

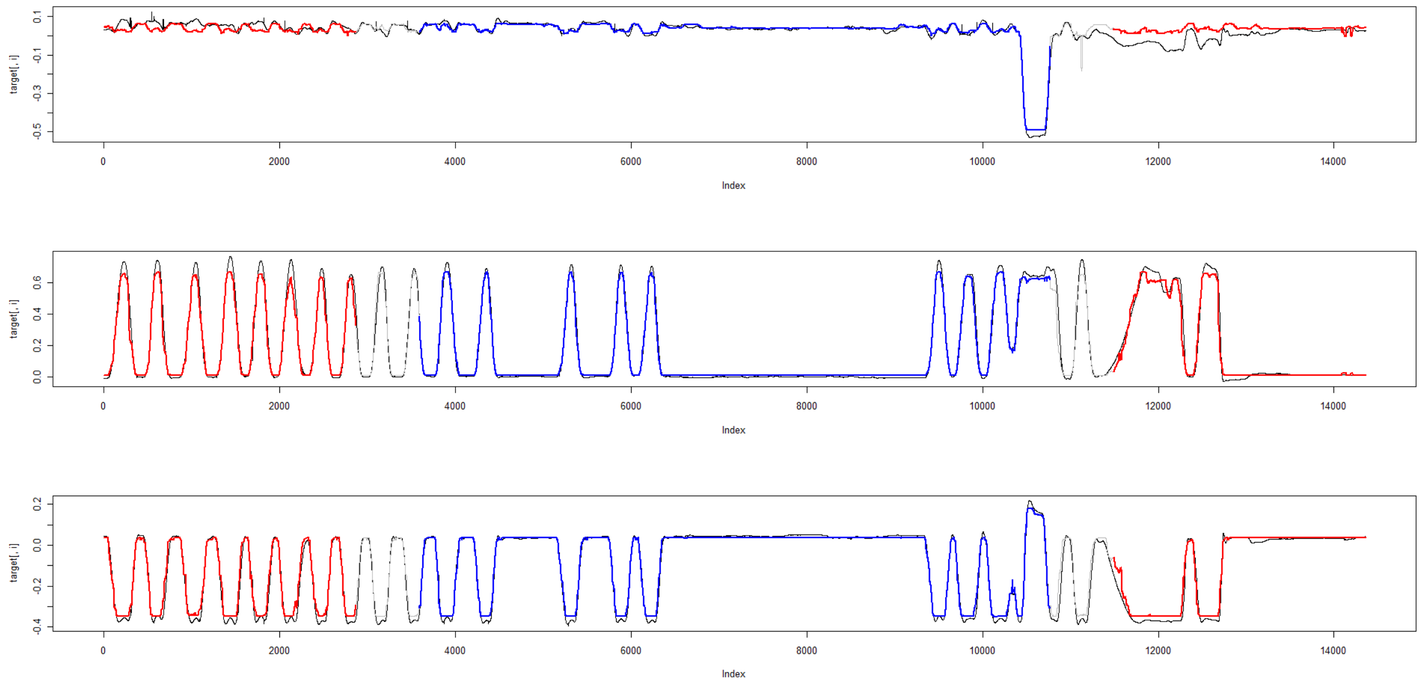


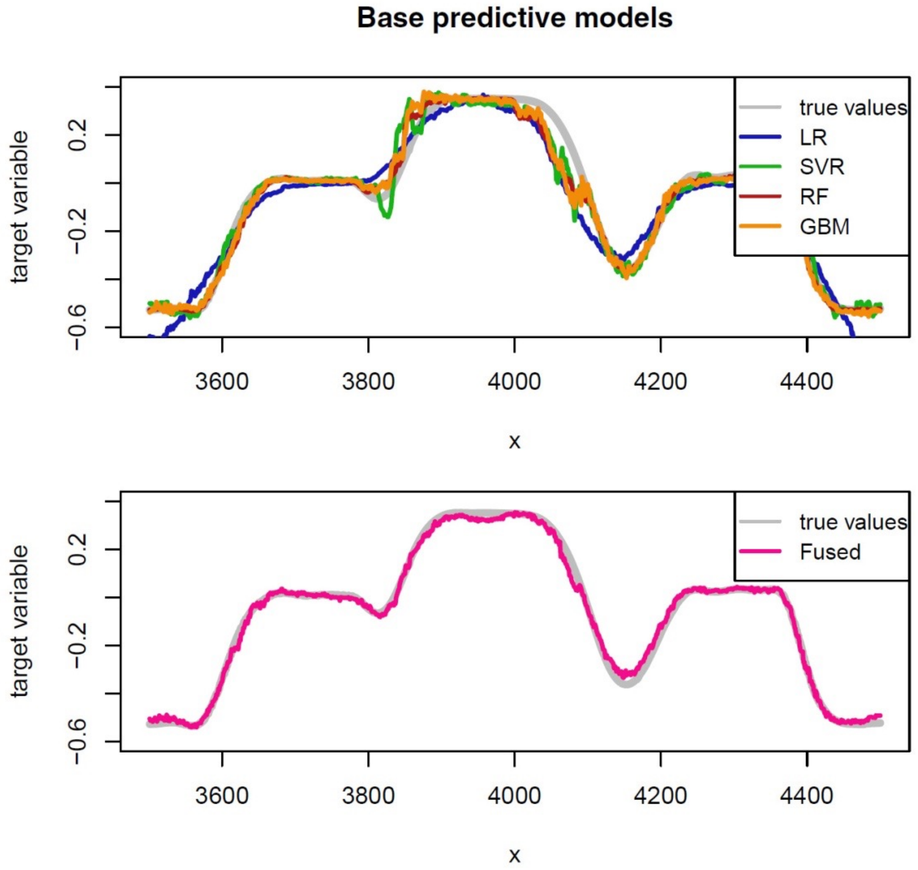
RF:



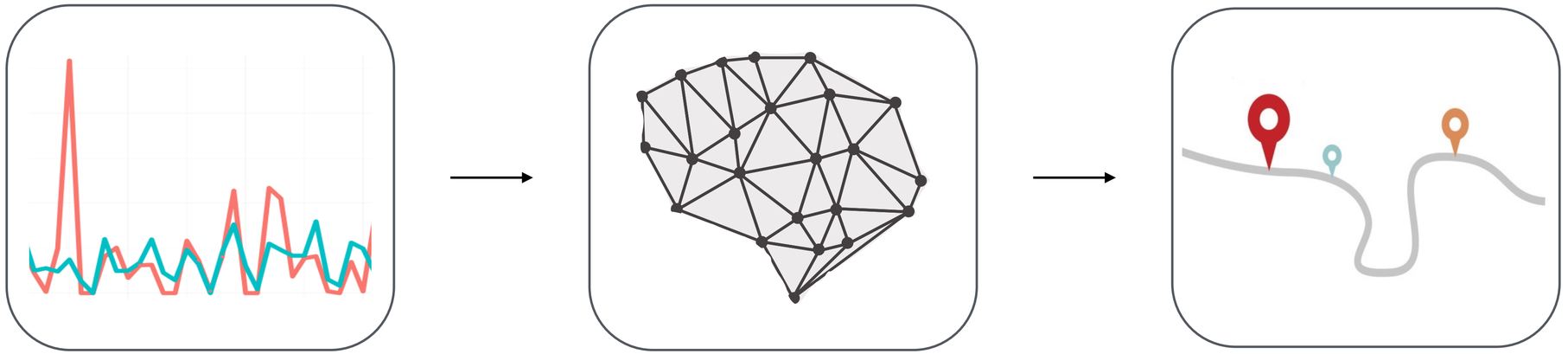
Case 1: result & post-process

- GBM with post-processing





ML cases for time-series:



Cash demand prediction for each ATM device

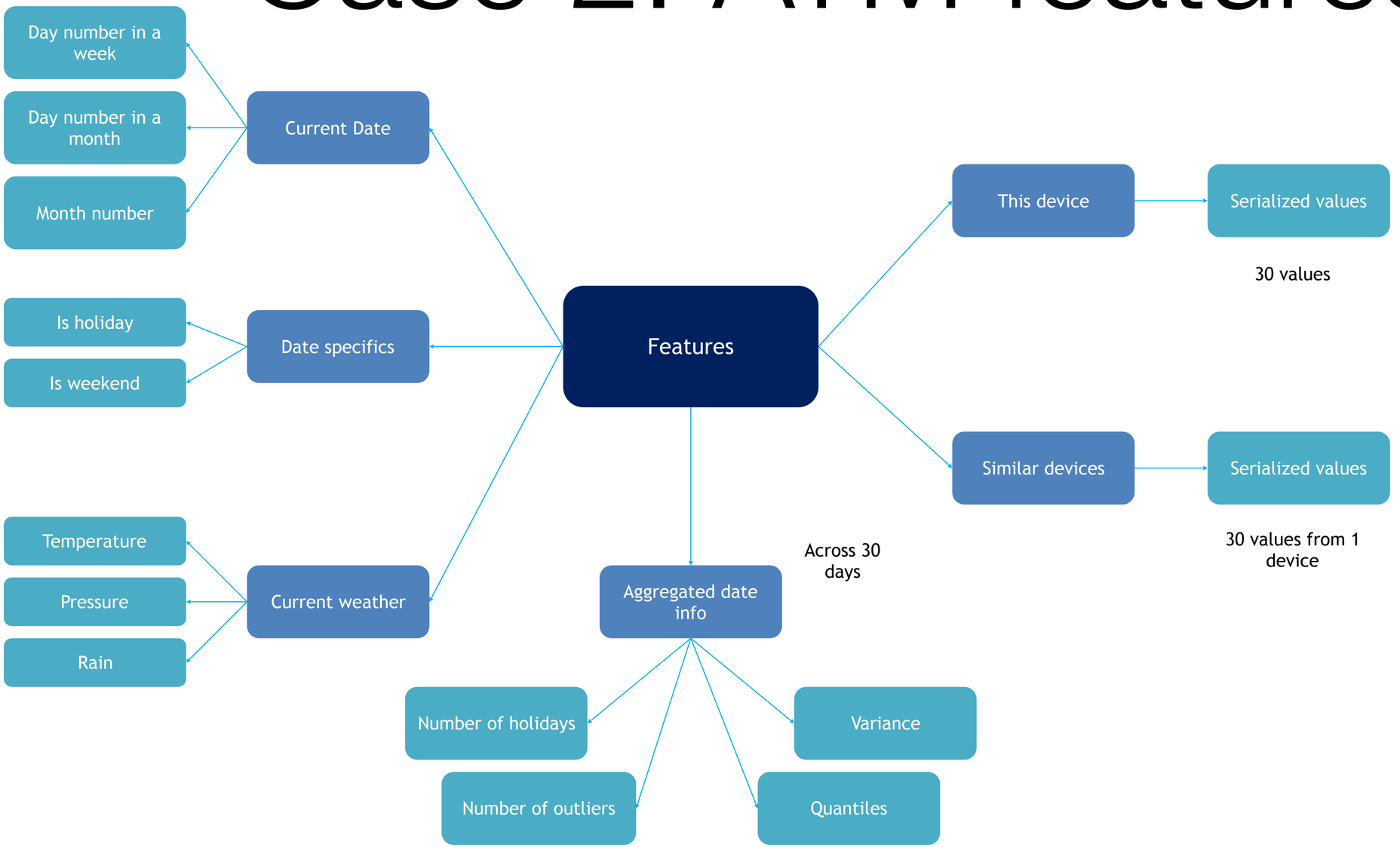
Encashment candidate evaluation and simulations

Optimize logistics for chosen candidates

Case 2: ML pipeline

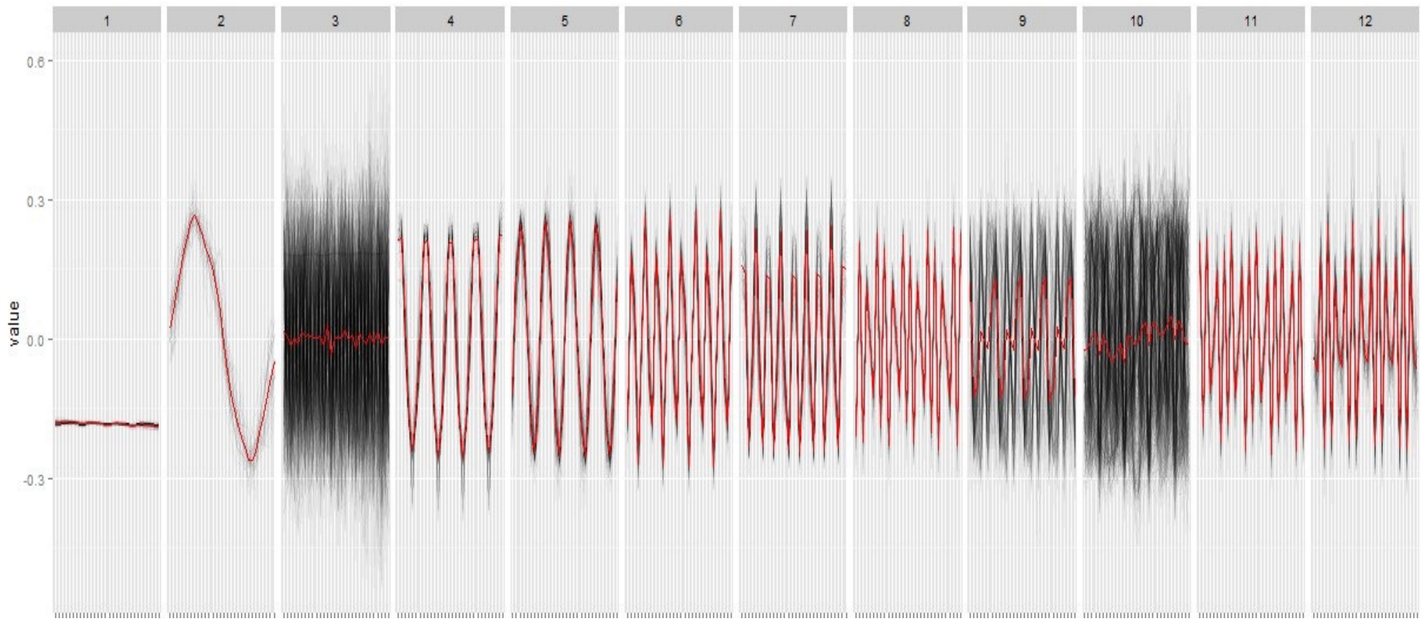
Data & problem	Data preparation	Feature engineering	Model	Evaluation
▸ A lot of external data sources besides cash ▸ Predict diffs and several total sums	▸ Fill NA to zero ▸ Adjust frequencies and time-grids of all external data	▸ Day-to-day ▸ AR and aggregates ▸ Aggregates on top of KNN	▸ Tried a lot ▸ GBM again was good ▸ Had to compete with ARIMA hard	▸ Time-CV ▸ Fair validation and test one year ahead

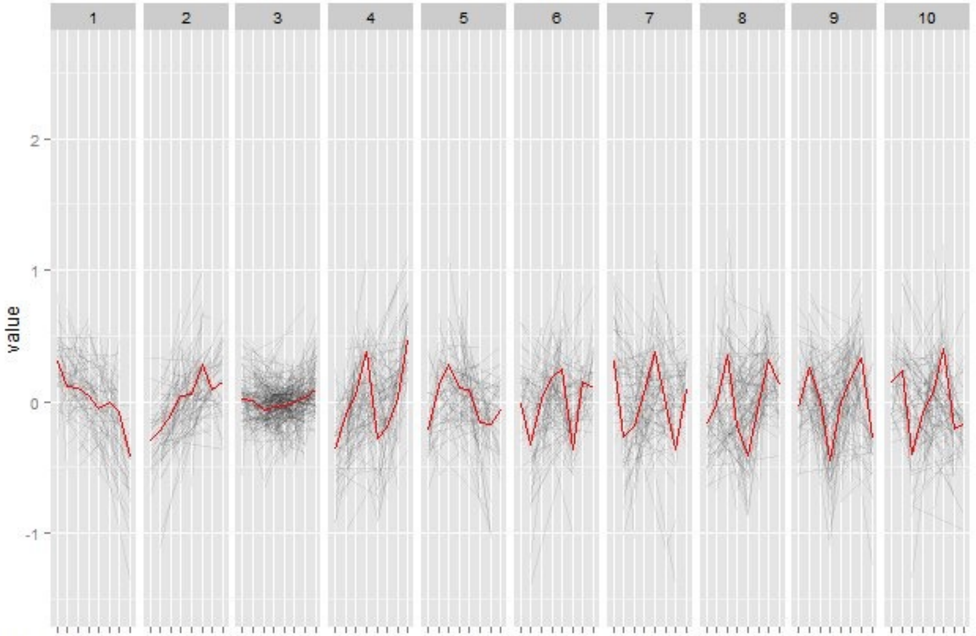
Case 2: ATM features



Case 2: curious stuff

- TS pattern mining was cool, but models already got all the info





Time-series ML pipeline

Data & problem	Data preparation	Feature engineering	Model	Evaluation
Value vs diff	Consistent spacing	**Stationarised series**	Linear	**Time-CV**
Regression/ classification /anomalies	Impute missing values	AR, MA and aggregates	Casual ML & boosting	Test-policy (rolling or not)
Univariate sensors vs panel data	Aggregates/ pooling	Dense represent-ns	LSTM & RNN	Simulations (small data)
		Kaggle-style	Markov-style	
			Kaggle-style	

Can you consistently forecast values of non-stationary time series with a tree-based model like GBM?

A. Sure, watch me xgboost it

B. Maybe, if we work with diffs

C. No, trees can’t extrapolate

D. No, non-stationarity hurts

Can you consistently forecast values of non-stationary time series with a tree-based model like GBM?

A. Sure, watch me xgboost it

B. Maybe, if we work with diffs

C. No, trees can’t extrapolate

D. No, non-stationarity hurts

Summary

- Key time-series differences from classic ML: proper time-CV, specific preparation issues
- The big ML picture for time-series makes it easier to solve practical problems
- On the other hand the magic disappears and you forget names of all those methods

Thank you!

ods.ai

@natekin

alex.natekin@dmlabs.org