tabular

Supervised machine learning: basic theory and practical experience

Václav Šmídl[1,2], Matěj Zorek[1], Josef Justa[2], et. al.

[1] AI Center, FEL, CTU, Prague,
[2] RICE, FEL, UWB, Pilsen,

February 1, 2023

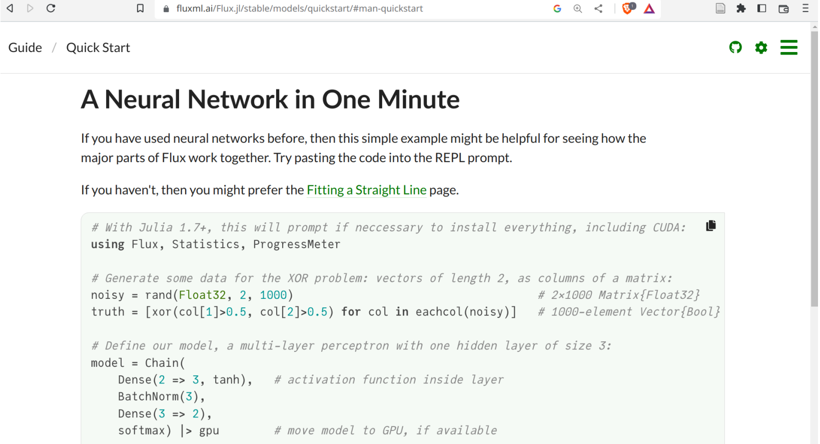
Motivation: students and their interests

Deep learning and AI are hot topics:

- NLP – ChatGPT
- Images
  - Midjourney,
  - Dall-E 2

The community is very open

- open data
- open source code (PyTorch, TensorFlow, Flux.jl)
- tutorials, discourse



A Neural Network in One Minute

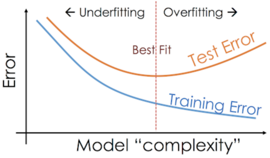
If you have used neural networks before, then this simple example might be helpful for seeing how the major parts of Flux work together. Try pasting the code into the REPL prompt.

If you haven't, then you might prefer the Fitting a Straight Line page.

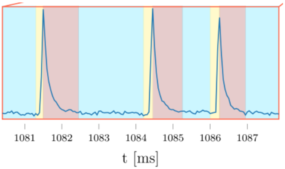
```julia
# With Julia 1.7+, this will prompt if neccessary to install everything, including CUDA:
using Flux, Statistics, ProgressMeter

# Generate some data for the XOR problem: vectors of length 2, as columns of a matrix:
noisy = rand(Float32, 2, 1000)                                    # 2×1000 Matrix{Float32}
truth = [xor(col[1]>0.5, col[2]>0.5) for col in eachcol(noisy)]   # 1000-element Vector{Bool}

# Define our model, a multi-layer perceptron with one hidden layer of size 3:
model = Chain(
    Dense(2 => 3, tanh),   # activation function inside layer
    BatchNorm(3),
    Dense(3 => 2),
    softmax) |> gpu        # move model to GPU, if available
```

Roadmap

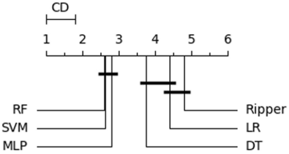
Supervised Learning Theory



Practical Examples[1][2]



Evaluation Protocol[3]



[1] Justa, J., Šmídl, V. and Hamáček, A., 2022. Deep Learning Methods for Speed Estimation of Bipedal Motion from Wearable IMU Sensors. Sensors, 22(10), p.3865.

[2] Zorek, M., Škvára, V., Šmídl, V., Pevný, T., Seidl, J., Grover, O. and Compass Team, 2022. Semi-supervised deep networks for plasma state identification. Plasma Physics and Controlled Fusion, 64(12), p.125004.

[3] Škvára, V., Francu, J., Zorek, M., Pevný, T. and Šmídl, V., 2021. Comparison of anomaly detectors: context matters. IEEE Transactions on Neural Networks and Learning Systems, 33(6), pp.2494-2507.

Supervised learning:

Is actually an input-output function learning:

$$y = f(x),$$

where x is the input, and y is the output, with training samples $\{x_i, y_i\}_{i=1}^{n}$

Supervised learning:

Is actually an input-output function learning:

$$y = f(x),$$

where x is the input, and y is the output, with training samples $\{x_i, y_i\}_{i=1}^{n}$

Regression, output is an infinite number of possibilities, $y \in \mathbb{R}^d$

$$\text{salary} \quad = \quad f(\text{curriculum\_vitae.txt})$$

Classification output is a finite number of possibilities, $y \in \{1, 2, \ldots C\}$

$$\left.\begin{array}{c}\text{engineer}\\ \text{manager}\\ \vdots\end{array}\right\} \quad = \quad f(\text{curriculum\_vitae.txt})$$

Difference

$$\text{error}_{\text{rgr}} = \sum_i ||y_i - f(x_i)||_2^2, \qquad \text{error}_{\text{cls}} = \sum_{i,j} (y_j \log f(x)_j) \text{ or } \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}}$$

What are the functions, f!? How to find the right one?

Black-box models

Universal Approximation Models

Theorem (Cybenko 1989, Hornik 1991) MLP with growing number of neurons can approximate $y = f(x)$ on a compact set can be approximated arbitrarily accurately iff σ is a non-polynomial function.

- Applies even for 2 layer networks
- Deep Network requires exponentially fewer units than shallow for the same accuracy (Mhaskar et. al. 2017).
- hold for many models: kernel methods, probabilistic circuits, ...
- arbitrary accuracy is both blessing and curse

Models with Inductive Bias

The model has **an information bottleneck** and cannot represent any data and can not achieve zero error of noiseless data

- great if we have reasons to believe the model more than the data
- simplicity and explainability
- poor if we have no clue about the true underlying model

Models: Features or deep?

How to represent something relatively complex like "curriculum_vitae.txt"?

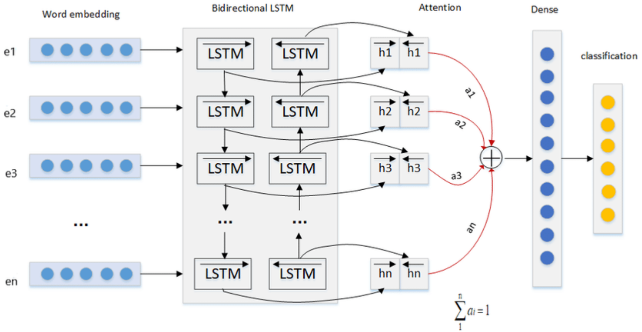
Features: a vector of numbers.

- CV -> bag of words = histogram of a selected dictionary.

school	university	Harvard	manager	...	
3	1	0	0		

- typically designed manually using engineering insight

Deep neural networks: networks that were designed to learn the features themselves, end-to-end



What is the right architecture? Are engineers not needed?

Choosing the model for polynomial curve fitting

Linear regression:

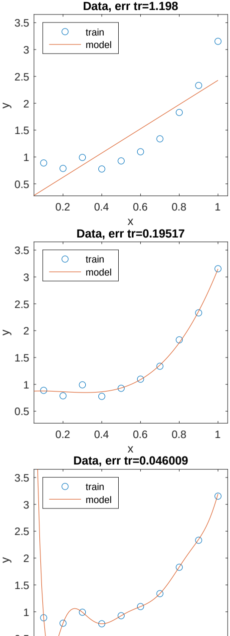
$$y_i = a_1 + a_2 x + a_3 x^2 + \cdots a_{p+1} x^p + e_i$$

has solution

$$\boldsymbol{a}_{LS} = (X^T X)^{-1} X^T y$$

where $X_i = [1, x_i, x_i^2, \ldots x_i^p]$.

- p has to be known! **Hyper-parameter**.
- how to choose p?



Choosing the model for polynomial curve fitting

Linear regression:

$$y_i = a_1 + a_2 x + a_3 x^2 + \cdots a_{p+1} x^p + e_i$$

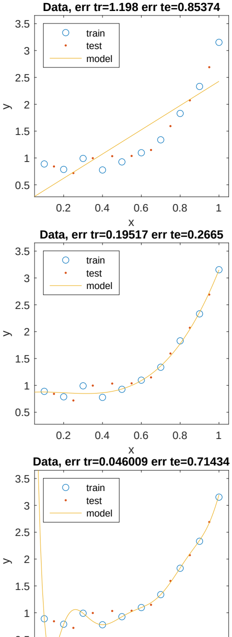
has solution

$$\mathbf{a}_{LS} = (X^T X)^{-1} X^T y$$

where $X_i = [1, x_i, x_i^2, \ldots x_i^p]$.

- p has to be known! **Hyper-parameter**.
- how to choose p?

Two sets of data train & test



Choosing the model for polynomial curve fitting

Linear regression:

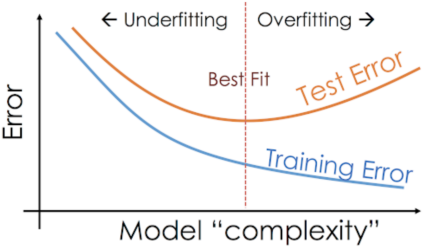
$$y_i = a_1 + a_2 x + a_3 x^2 + \cdots a_{p+1} x^p + e_i$$

has solution

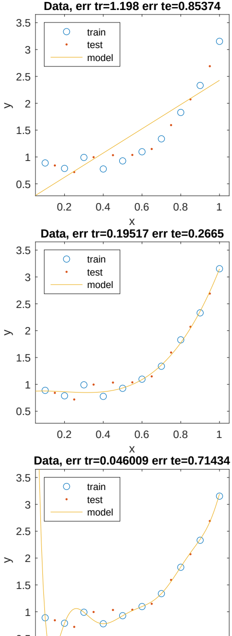
$$\boldsymbol{a}_{LS} = (X^T X)^{-1} X^T y$$

where $X_i = [1, x_i, x_i^2, \ldots x_i^p]$.

- p has to be known! **Hyper-parameter**.
- how to choose p?



The right p is that with minimum test error.



Cross-validation

How to decide what belongs to test and train data?

K-fold cross-validation:

All data records X, Y			
fold1	fold2	$\dots$	foldK
$\{x_i\}_{i\in I_1}$	$\{x_i\}_{i\in I_2}$		$\{x_i\}_{i\in I_K}$
$\{y_i\}_{i\in I_1}$	$\{y_i\}_{i\in I_2}$		$\{y_i\}_{i\in I_K}$

- such that every data point is in one fold only.(shuffle)
- K is usually low 5,10

Theory

nice properties for random (i.i.d) splits

- Consistency and generalization bounds (Vapnik, 1998)

```
for k=1:K
  ▶ Fit model for collection
    of all folds except the
    Fold k
  ▶ Evaluate error on Fold k
end
report average error, or its
distribution
```

Hidden and dangerous assumption

- test and train data are generated from the same distribution

In practice:

- we want to apply our model to an "unseen" phenomena.
- most obvious in time-dependent data
  - train model on historical data

$$\begin{aligned} \text{salary} &= f(\text{curriculum\_vitae.txt}) \\ \text{metrics:} &\quad \textstyle\sum_i ||y_i - f(x_i)||_2^2 \end{aligned}$$

  - apply it for offering new recruits in the company

Hidden and dangerous assumption

- test and train data are generated from the same distribution

In practice:

- we want to apply our model to an “unseen” phenomena.
- most obvious in time-dependent data
  - train model on historical data

$$\begin{aligned} \text{salary} &= f(\text{curriculum\_vitae.txt}) \\ \text{metrics:} &\quad \textstyle\sum_i ||y_i - f(x_i)||_2^2 \end{aligned}$$

  - apply it for offering new recruits in the company

More realistic approach:

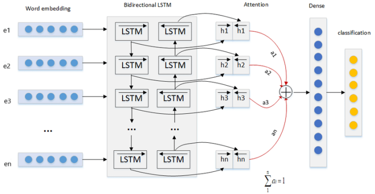
Split the data to three sets

train	use to fit model parameters (for all hyperparameters)
validation	select hyperparameters (order p) ▶ potentially an outer optimization loop!
test	report testing error on “vault” data

Expert knowledge on what is the “test” – see applications.

Neural Networks: hyper-parameters

Everything we need to fix for the model training is a hyper-parameter



Hyper-parameters:

- number of neurons (in layers)
- activation functions
- **Optimization setting(!)**
  - learning-rate,
  - dropout,
  - momentum

For more complex architectures:

- filter sizes
- no of channels
- pooling
- ...

The number of degrees of freedom is relatively high.

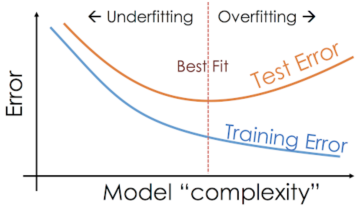
Typical scenario:

1. prepare your data, split into test/validation/test
2. prepare all your methods and their hyperparameters
   2.1 fixed grid search (may be costly)
   2.2 random grid search
3. Run **all versions** of the model
4. Select the best model on validation
5. (If the best hyper parameters on edge, increase edge, GOTO 2)
6. Report results on test

Early stopping:

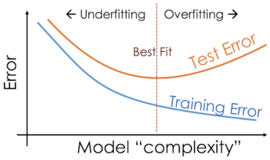
- check validation error during the fitting procedure
- stop training if it starts to steadily increase (e.g. 30 times in a row)

Very useful for diverging models (wrong lr).

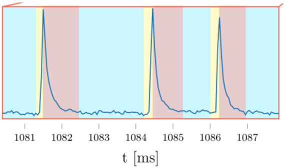


Roadmap

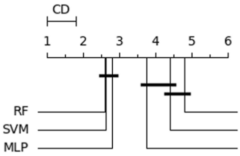
1
Supervised Learning Theory



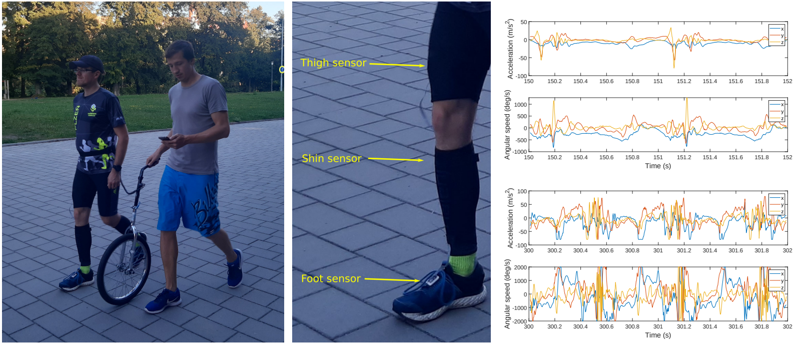
2
Practical Examples



3
Evaluation Protocol



Application #1: Human Motion Speed Estimation



- Measurements of 16 time series (8 different people in 2 different experiments)[4]

Supervised problem:

- x is 1s window of $6n_s$ dimensional signal.
- y is the walking/running speed in km/h

[4] Justa, J., Šmídl, V. and Hamáček, A., 2022. Deep Learning Methods for Speed Estimation of Bipedal Motion from Wearable IMU Sensors. Sensors, 22(10), p.3865.

Following the protocol

1. prepare your data, split into test/validation/test
2. prepare all your methods and their hyperparameters
   2.1 fixed grid search (may be costly)
   2.2 random grid search
3. Run all versions of the model
4. Select the best model on validation
5. (If the best hyper parameters on edge, increase edge, GOTO 2)
6. Report results on test

How to split the data?

- time windows (overlap?)
- what is the test?

Following the protocol

1. prepare your data, split into test/validation/test
2. prepare all your methods and their hyperparameters
   2.1 fixed grid search (may be costly)
   2.2 random grid search
3. Run all versions of the model
4. Select the best model on validation
5. (If the best hyper parameters on edge, increase edge, GOTO 2)
6. Report results on test

How to split the data?

- time windows (overlap?)
- what is the test?

Fold is a single person

Models (simple do not work well):

1. HVAE-LSTM-CNN
2. HVAE-Sine (ours)
3. Perceiver (Jaegle, 2021)
4. InceptionTime (Fawaz, 2022)

Following the protocol

1. prepare your data, split into test/validation/test
2. prepare all your methods and their hyperparameters
   2.1 fixed grid search (may be costly)
   2.2 random grid search
3. Run all versions of the model
4. Select the best model on validation
5. (If the best hyper parameters on edge, increase edge, GOTO 2)
6. Report results on test

How to split the data?

- time windows (overlap?)
- what is the test?

Fold is a single person

Models (simple do not work well):

1. HVAE-LSTM-CNN
2. HVAE-Sine (ours)
3. Perceiver (Jaegle, 2021)
4. InceptionTime (Fawaz, 2022)

Table 4. Hyper-parameters of the semisupervised VAE approach.

Encoder		Decoder	
Hyper-Parameter	**Range**	**Hyper-Parameter**	**Range**
Convolution channels	[1, 2, 4, 8, 16]	Sine: size of hidden layer	[10, 50, 100]
Size of hidden layer	[128, 256, 512]	LSTM-CNN: same as encoder	
Depth of hidden layer	[1, 2]		
Length of latent z	[64, 128, 256]		
Predictor weight α	[0.1, 0.01, 0.001, 0.0001]		
KL weight β	$[1\times10^{-4}, 1\times10^{-5}, 1\times10^{-6}, 1\times10^{-7}]$		

Results – hyper-parameter selection top 3 methods

Table A1. Summary of the best hyper-parameters of the Autoencoder: CONV-LSTM-CONV.

Conv_Channels	Hidden_Size	Hidden_Layer_Depth	Latent_Length	α	β	Error km/h
8	128	1	64	0.1	1×10^{-7}	0.3552
8	128	1	128	0.01	1×10^{-5}	0.3814
8	128	1	64	0.01	0.0001	0.3844

Table A2. Summary of the best hyper-parameters of the Autoencoder: CONV-LSTM-SineNet.

Conv_Channels	Hidden_Size	Hidden_Layer_Depth	Latent_Length	α	β	Sin_Depth	Error km/h
16	256	1	128	0.01	1×10^{-7}	100	0.3238
8	256	1	128	0.01	1×10^{-7}	50	0.3640
8	256	1	64	0.1	1×10^{-4}	10	0.3693

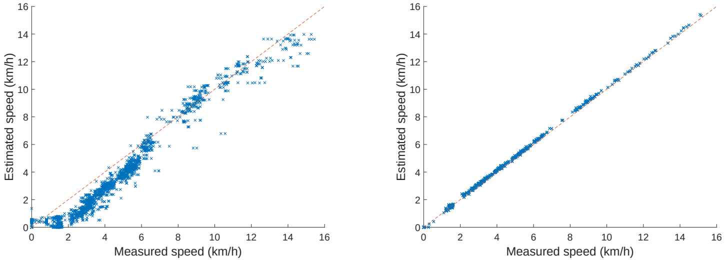
Table A3. Summary of the best hyper-parameters of the InceptionTime.

n_Filters	Kernel_Sizes	Bottleneck_Channels	Error km/h
16	[21, 41, 81]	8	0.3630
16	[11, 21, 41]	8	0.3662
8	[21, 41, 81]	4	0.3799

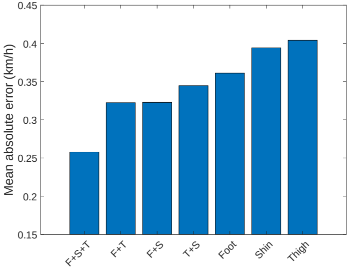
Table A4. Summary of the best hyper-parameters of the Perceiver.

Num_Freq_ Bands	Max_ Freq	Depth	Num_ Latents	Latent_ Dim	Cross_ Dim	Cross_ Dim_Head	Latent_ Dim_Head	Error km/h
6	10.0	6	256	128	256	32	64	0.4339
6	15.0	6	256	128	512	32	16	0.4691
12	15.0	12	512	256	128	64	64	0.4956

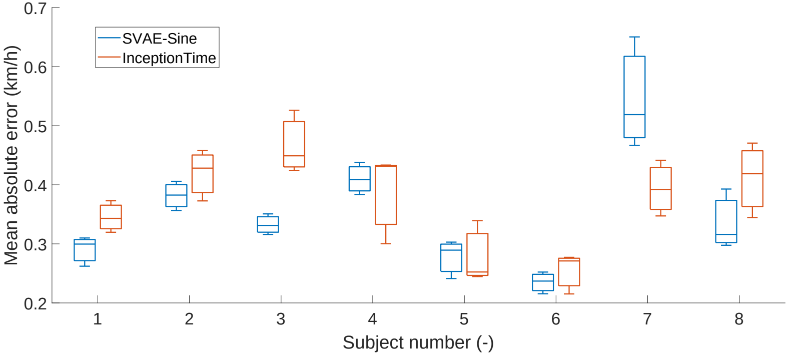
Lessons learned: Deep methods rules



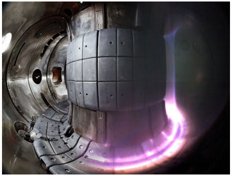
and improve with more data (sensor fusion)

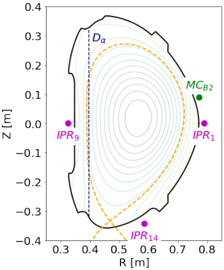


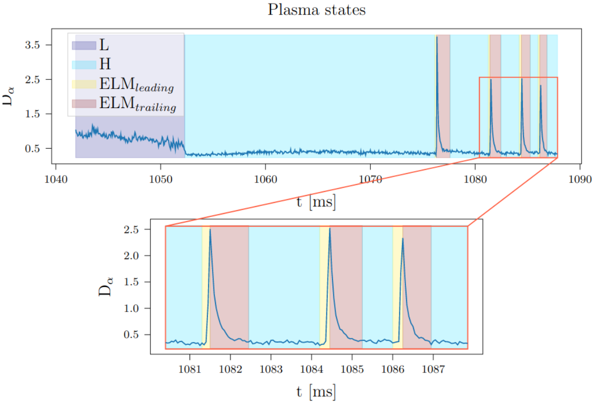
Lessons learned: Humans are tricky



Application #2: plasma state classification







Supervised learning:

- input is a window of 5 signals of 160 consecutive samples
- output, 4 possible states of plasma
- 31 labeled discharges from the physicists

First attempt: Diploma thesis

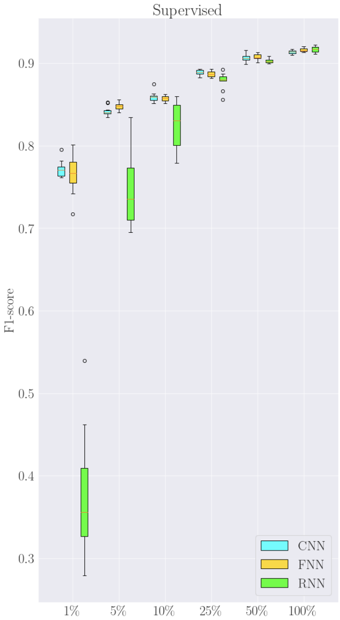
1. prepare your data, split into test/validation/test
2. prepare all your methods and their hyperparameters
3. Run all versions of the model
4. Select the best model on validation
5. Report results on test

With choices

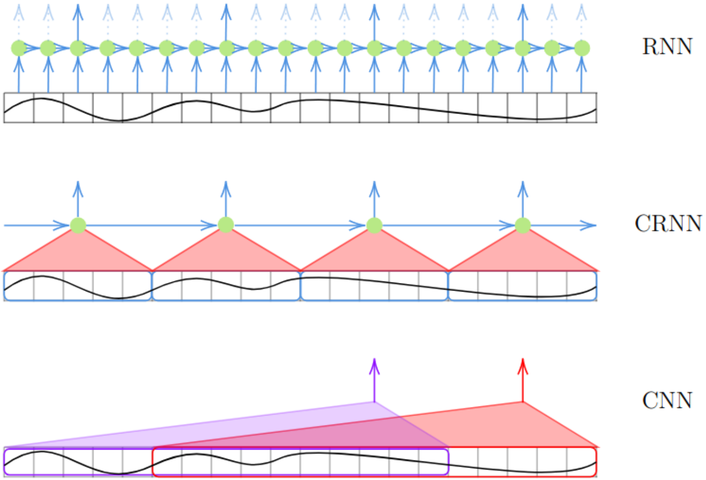
- i.i.d. test/validation/train split
- Models:
  - Convolutional NN
  - Recurrent NN (LSTM)
  - Fully connected NN

Note that:

- with full data, difference in architecture does not matter



Architectures



Feedback from physicists

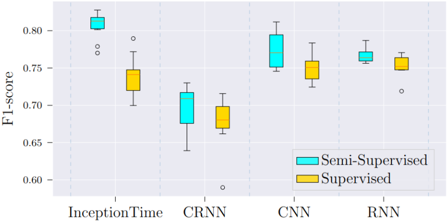
1. Every shot is unique
2. CRNN are standard known in the community
3. F1 metric is not interesting
   - what is the delay of classifications?
   - can we use unlabeled samples
   - where are the mistakes?

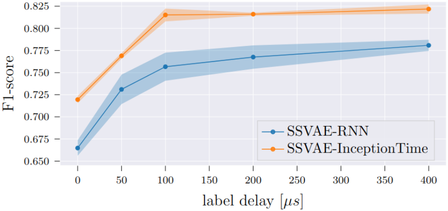
Feedback from physicists

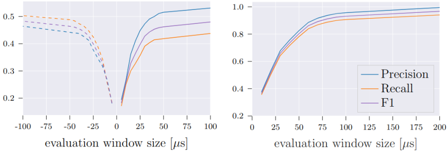
1. Every shot is unique
2. CRNN are standard known in the community
3. F1 metric is not interesting
   - what is the delay of classifications?
   - can we use unlabeled samples
   - where are the mistakes?

Our "Fixes" [a]:

1. Only 20 shots for learning, 11 for testing
2. Modern architectures:
   - 2.1 Semi-supervised Variational AE
   - 2.2 Include InceptionTime Classifier
3. Evaluate transition-sensitive metrics
4. Sensitivity study to label delay





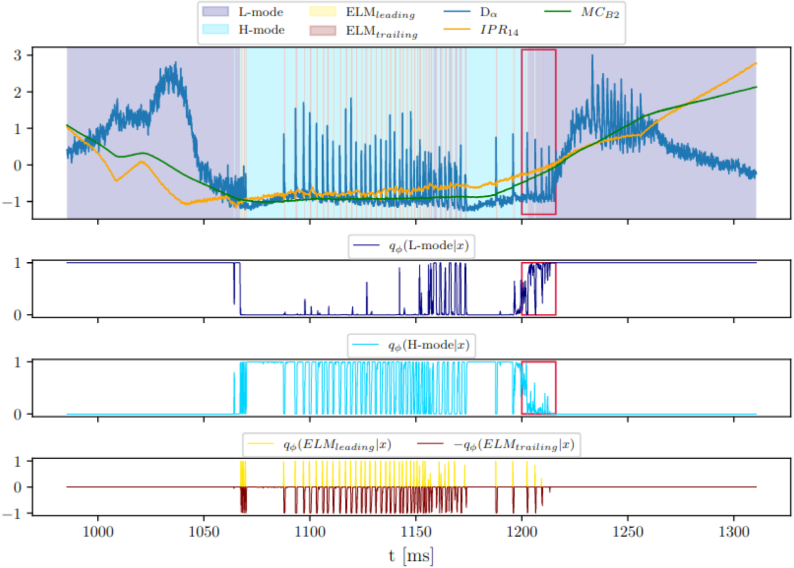


(a) one-sided (b) two-sided

---

[a] Zorek, M., Škvára, V., Šmídl, V., Pevný, T., Seidl, J., Grover, O. and Compass Team, 2022. Semi-supervised deep networks for plasma state identification. Plasma Physics and Controlled Fusion, 64(12), p.125004.

Error analysis

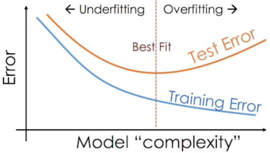


Lessons learned

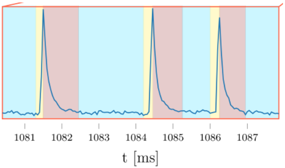
1. Choice of model architecture may not matter
   - models are data-interpolators, with enough data they are equal
   - becomes more relevant with less data
2. Key issues
   - Evaluation protocol – know your data
   - Label quality – may be iterated (feed back to practitioners)
   - Clarify the evaluation metric (accuracy or recall?)

Roadmap

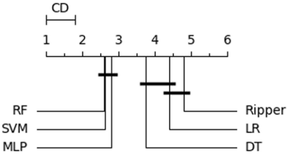
1
Supervised Learning Theory



2
Practical Examples



3
Evaluation Protocol



Which method is the best?

- Each specific problem may have its own "best" architecture
- With small differences in performance a method that is good on "average" may do a good job
- Methods are being evaluated on very large dataset bases
  - may reveal patterns in data
  - some class of methods may be suitable for some type of data
- Can we trust the results?

[5] Škvára, V., Francu, J., Zorek, M., Pevný, T. and Šmídl, V., 2021. Comparison of anomaly detectors: context matters. IEEE Transactions on Neural Networks and Learning Systems, 33(6), pp.2494-2507.

Which method is the best?

- Each specific problem may have its own “best” architecture
- With small differences in performance a method that is good on “average” may do a good job
- Methods are being evaluated on very large dataset bases
  - may reveal patterns in data
  - some class of methods may be suitable for some type of data
- Can we trust the results?

Our experience with benchmarking of anomaly detectors[5]

Anomaly detection

Considers data of two types: normal and anomalies.

- Trains only on normal (unsupervised)
- Evaluates on both normal/anomalies using supervised metrics

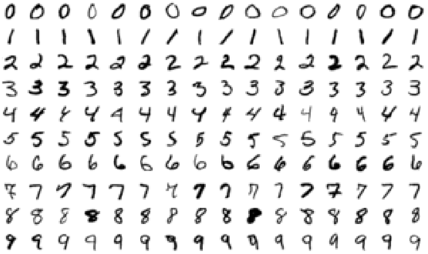
[5] Škvára, V., Francu, J., Zorek, M., Pevný, T. and Šmídl, V., 2021. Comparison of anomaly detectors: context matters. IEEE Transactions on Neural Networks and Learning Systems, 33(6), pp.2494-2507.

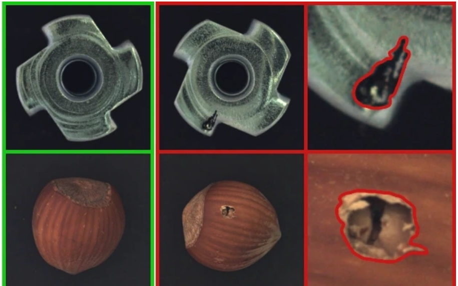
Research question: what to reseach?

- Anomaly detectors are here for ages: KNN (Fix, Hodges, 1951)
- Recent publications focus on Deep model – claiming superiority

What is better?

- Data set selection
  - feature-based data
  - image data
    - multiclass (Mnist)
    - anomaly
- Method types and selection
- Computational time vs. accuracy





Datasets

BASIC STATISTICS OF IMAGE DATASETS DESIGNED FOR ANOMALY DETECTION (ABOVE SPLIT) AND MULTICLASS DATASETS (BELOW SPLIT)

dataset	alias	dim	anom	normal
MNIST-C	mnistc	28x28x1	70000	70000
MVTec-AD - wood	wood	1024x1024x3	60	266
MVTec-AD - grid	grid	1024x1024x3	57	285
MVTec-AD - transistor	transistor	1024x1024x3	40	273
CIFAR10	cifar10	32x32x3	54000	6000
FashionMNIST	fmnist	28x28x1	63000	7000
MNIST	mnist	28x28x1	63686	6312
SVHN2	svhn2	32x32x3	80327	18960

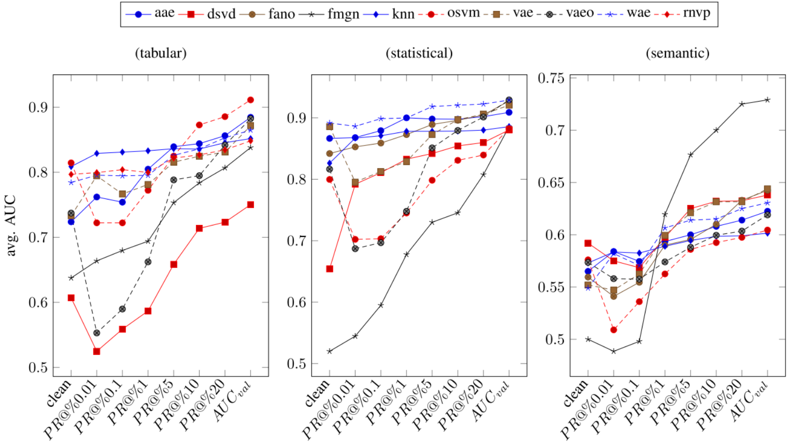
OVERVIEW OF THE MAIN CLASSES OF COMPARED METHODS AND THE ACRONYMS USED IN THE TEXT

class	model	acronym	class	model	acronym
flows	MAF	maf	two-stage	DAGMM	dgmm
	RealNVP	rnvp		DeepSVDD	dsvd
	SPTN	sptn		REPEN	rpn
				VAE-kNN	vaek
autoencoders	AAE	aae		VAE-OC-SVM	vaeo
	adVAE	avae			
	GANomaly	gano	classical	ABOD	abod
	skipGANomaly	skip		HBOS	hbos
	VAE	vae		IsolationForest	if
	WAE	wae		kNN	knn
				LODA	loda
gans	fAnoGAN	fano		LOF	lof
	fmGAN	fmgn		OC-SVM	osvm
	GAN	gan		PidForest	pidf
	MOGAAL	mgal			

dataset	alias	dim	anom	normal
ANNthyroid	ann	21	534	6665
Arrhythmia	arr	275	206	245
HAR	har	561	1944	8355
HTRU2	htr	8	1638	16257
KDD99 (10%)	kdd	118	396742	97276
Mammography	mam	6	260	10921
Seismic	sei	24	170	2412
Spambase	spm	57	1812	2786
Abalone	aba	10	50	2151
Blood Transfusion	blt	4	16	382
Breast Cancer Wisconsin	bcw	30	206	356
Breast Tissue	bts	9	22	65
Cardiotocography	crd	27	228	1830
Ecoli	eco	7	108	205
Glass	gls	10	94	112
Haberman	hab	3	14	225
Ionosphere	ion	33	122	225
Iris	irs	4	46	100
Isolet	iso	617	3300	4496
Letter Recognition	ltr	617	3600	4196
Libras	lbr	90	142	215
Magic Telescope	mgc	10	3882	12331
Miniboone	mnb	50	23922	93565
Multiple Features	mlt	649	800	1200
PageBlocks	pgb	10	384	4911
Parkinsons	prk	22	44	146
Pendigits	pen	16	5384	5537
Pima Indians	pim	8	176	500
Sonar	snr	60	96	110
Spect Heart	sph	44	52	211
Statlog Satimage	sat	36	2630	3592
Statlog Segment	seg	18	938	1320
Statlog Shuttle	sht	8	28	57767
Statlog Vehicle	vhc	18	132	627
Synthetic Control Chart	scc	60	200	400

Evaluation protocol is a game-changer

- The number of anomalies is assumed to be small – what is “small”?
- Sensitivity to the number of considered anomalies



Is it significant?

Statistical comparisons of classifiers over multiple data sets

Seminal publication: Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. The Journal of Machine learning research, 7, pp.1-30.

1. For each dataset
   1.1 sort the performace of the method
   1.2 asign ranks to the methods: 1,2,...
2. Compute average rank
3. Evaluate statistical test
   - Wilcox
   - Nemenyi
   - Friedman
4. Display critical diagram

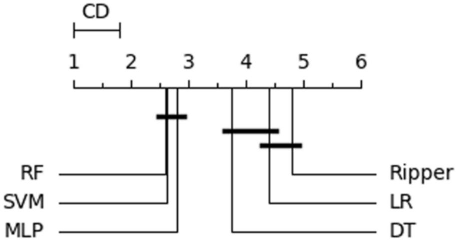
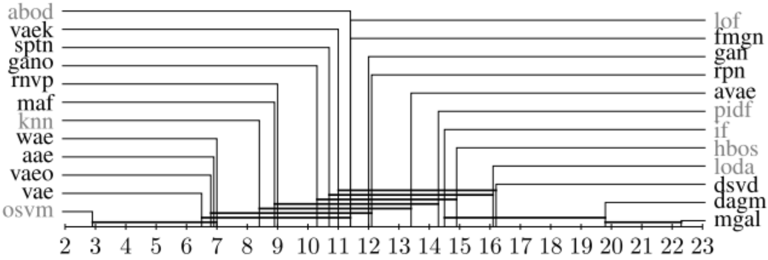


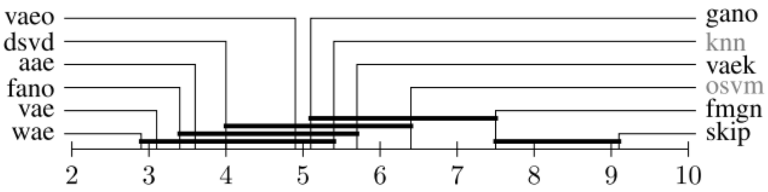
Illustration of the idea from [a]

[a] Goethals, S., Martens, D. and Evgeniou, T., 2022. The non-linear nature of the cost of comprehensibility. Journal of Big Data, 9(1), p.30.

Results


abod
vaek
sptn
gano
rnvp
maf
knn
wae
aae
vaeo
vae
osvm
lof
fmgn
gan
rpn
avae
pidf
if
hbos
loda
dsvd
dagm
mgal
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23

(a)


vaeo
dsvd
aae
fano
vae
wae
gano
knn
vaek
osvm
fmgn
skip
2 3 4 5 6 7 8 9 10

Lessons Learned: Hyper-parameters

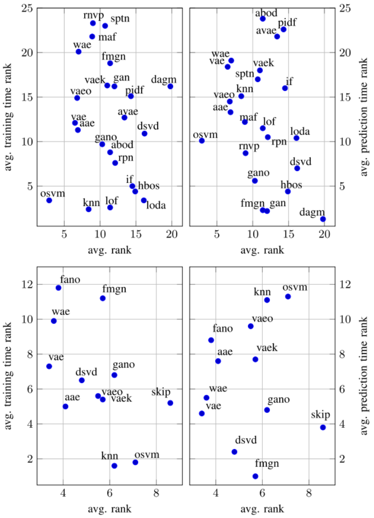
1. Use Hyper-parameters
   - OCSVM was outperformed previously, wins in our case
   - Previous studies used rbf kernel. We searched for it.
   - Anything is a hyper-parameter (loss, architecture, score)
   - Default hyper-parameters harm the method!!!

Lessons Learned: Hyper-parameters

1. Use Hyper-parameters
   - OCSVM was outperformed previously, wins in our case
   - Previous studies used rbf kernel. We searched for it.
   - Anything is a hyper-parameter (loss, architecture, score)
   - Default hyper-parameters harm the method!!!
2. Hyper-parameter optimization
   - We have used 100 random samples of all hyperparaneters
   - Bayesian optimization: 50 initial, 50 iteratively added
     - systematic improvement of all methods
     - negligible in performance, no influence on ranks of best models

Lessons Learned: Hyper-parameters

1. Use Hyper-parameters
   - OCSVM was outperformed previously, wins in our case
   - Previous studies used rbf kernel. We searched for it.
   - Anything is a hyper-parameter (loss, architecture, score)
   - Default hyper-parameters harm the method!!!
2. Hyper-parameter optimization
   - We have used 100 random samples of all hyperparaneters
   - Bayesian optimization: 50 initial, 50 iteratively added
     - systematic improvement of all methods
     - negligible in performance, no influence on ranks of best models
3. Economic issue
   - complex models are costly to train with little benefit
   - classical methods are not dead



Conclusion

1. You want to try “build deep NN in 1min”?
    1.1 Go for it! – Deep methods **are** a commodity technology.
    1.2 Will be useful only with interesting data
    1.3 Data are much more important than NN architectures
2. Think hard about dependencies in the data
    2.1 are testing data same as training?
    2.2 what is the metric of success?
    2.3 concept drift, grouped data?
3. Make sure to do very good state of the art analysis
    3.1 new methods appear frequently
    3.2 their application as well
    3.3 carefully check their protocol: test/validation/test, hyper-params...
    3.4 reproducibility
        3.4.1 rerun the previous experiments
        3.4.2 publish your code and data

AI tea initiative

- Latest AI/ML progress is possible due to collaboration
  - academia/industry
  - cross-domain

AI tea initiative

- Latest AI/ML progress is possible due to collaboration
  - academia/industry
  - cross-domain
- Realized even by many politicians
  - Toronto (vector institute), Singapore AI,
  - Prg.ai has support of the city council
  - Prg.ai minor (major in preparation)
    - CVUT: FEL, FIT,
    - UK: MFF, Social
- ZCU?

AI tea initiative

- Latest AI/ML progress is possible due to collaboration
  - academia/industry
  - cross-domain
- Realized even by many politicians
  - Toronto (vector institute), Singapore AI,
  - Prg.ai has support of the city council
  - Prg.ai minor (major in preparation)
    - CVUT: FEL, FIT,
    - UK: MFF, Social
- ZCU?
  - https://pyvo.cz/plzen-pyvo/
  - AI tea: informal meetings