Software Engineering for Machine Learning Applications

SEMLA:

The Road Ahead

G. Antoniol and F. Khomh — Polytechnique Montreal

SEMLA

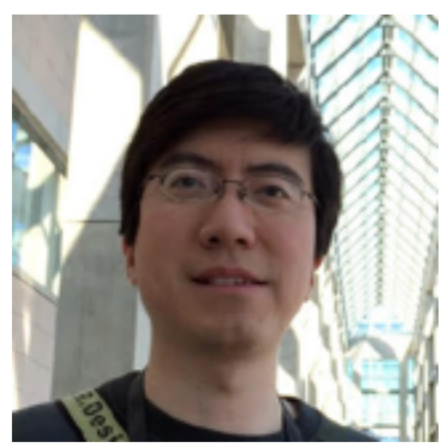
- The idea
- Organizers, speakers and topics
- Moving forward … what next

ORGANIZERS

Giuliano (Giulio) Antoniol

Foutse Khomh

Marios Fokaefs



Jinghui Cheng

Bram Adams

ML/AI - SEMLA

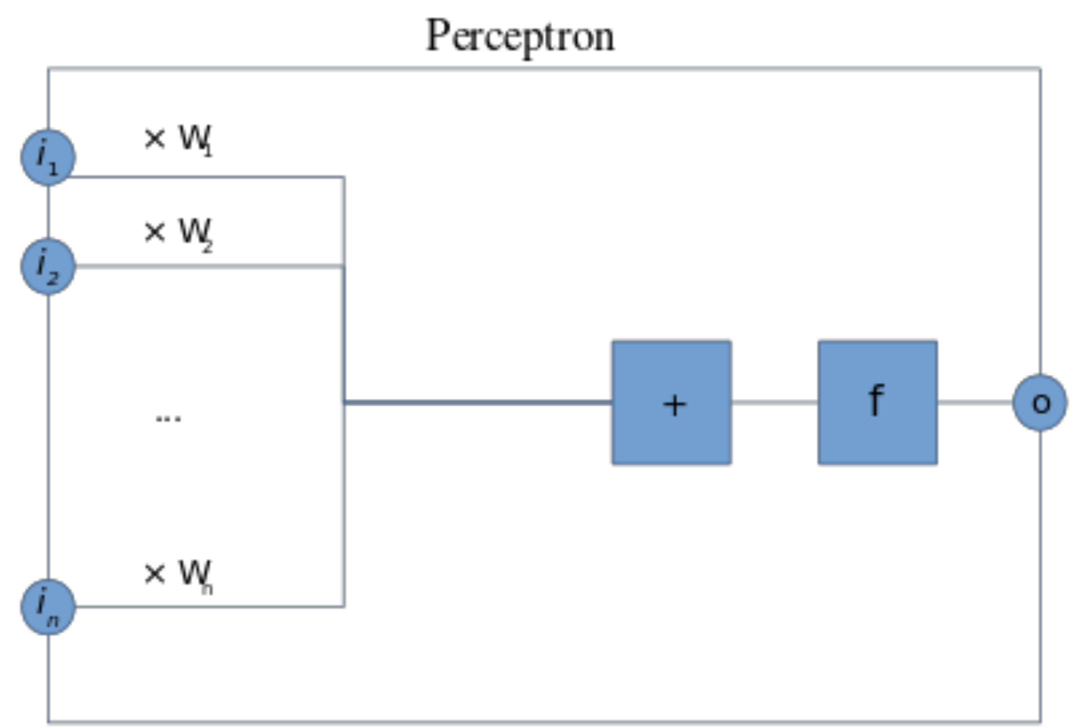
- Eliza (J Weizenbaum 1966) demonstrates we can be easily fooled believing an intelligent behaviour even if it is just pattern matching and pattern substitutions

- Fast forward to early 80's first attempts to integrate pattern recognition, machine learning, vision, spoken and natural language processing into "intelligent" platforms

- The dream is still valid create systems that learn

EARLY ATTEMPTS — SEMLA

- The perceptron was invented late 50's early 60's — Just one layer
- Neural networks have been around since 60's - 70's
- 2010 New hardware architectures — GPU
- More recently better software framework, better models, algorithms and hardware



$$o = f\left(\sum_{k=1}^{n} i_k \cdot W_k\right)$$

Image from Wikipedia

DEEP LEARNING — SEMLA

- Countless possibilities but:
  - How do we cope with robustness
  - How do we deploy in mission critical systems
  - The social and ethical impact

SEMLA GOAL

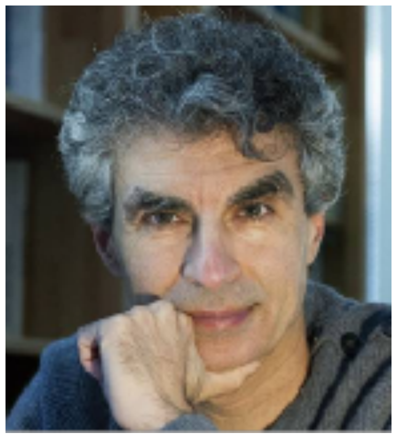
- Bridge the gap between software engineers and machine learning experts — topics:
  - Architecture and software design
  - Model/data verification and validation
  - Change management
  - User experience evaluation and adjustment
  - Privacy, safety, and security issues
  - Ethical concerns

SEMLA SPEAKERS

Jin L.C. Guo

Alessandro Petroni

Andrian Marcus

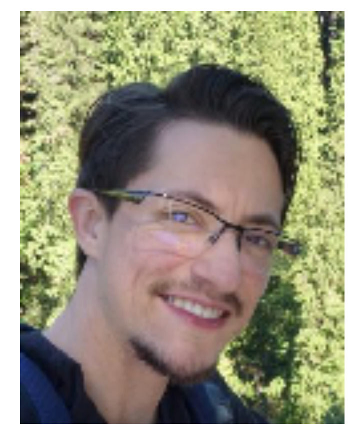


Yoshua Bengio

David Lorge Parnas

Bart van Merriënboer

Bernd Lehnert

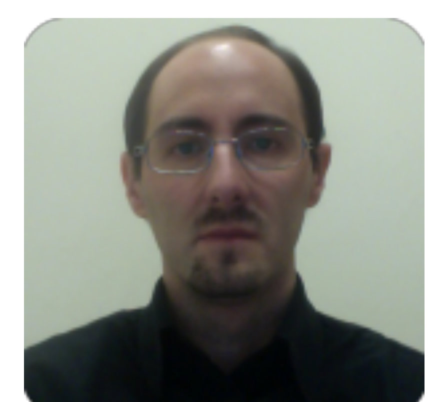


Leandro L. Minku

Lionel C. Briand

Chris Pal

Jason Schlessman



Massimiliano Di Penta

Denys Poshyvanyk

Paolo Tonella

SEMLA PROGRAM

- Talks, panels and posters
- Panels:
  - Computing the world to change the world: risks and opportunities
  - Are we ready for AI
  - The industry's take on SEMLA
  - Discussion session on education
- All material is available at: http://semla.polymtl.ca/

SEMLA MOVING FORWARD

Why worry

WHY WORRY ?

- Software runs the world we need to build more and more applications BUT we need to trust software: we depend on it
- Quality assurance and testing need complete, precise, non ambiguous, non vague specifications
- If specifications are not complete or non ambiguous how can we define an oracle

ML/AI SHOULD IT HELP US TO:

- Imitate human behaviour ?
- Play game well ?
- Build programs that use the same methods that human use?

NON TECHNICAL ISSUES

- Is the ML/AI application adapting to the user or vice-versa?
- If we trust too much the system behaviour we may overlook risky situations:
  - how do we keep the human in the loop?
- The human remains the final judge but there are sociological, ethical and political ramifications

IS ML A PANACEA

- Not all task are well suited for ML
- We can often solve the same or similar problem with traditional coding
- If we have physical laws and mathematical models why should we learn from data ?
- Find the right problem for the right tool is "a huge challenge"

CONTRADICTION

- If we write a program to compute an answer it implies we have not such an answer

- If we do not know what the answer is, how can we write an oracle and test the program?

E. J. Weyuker, "On testing non-testable programs," The Computer Journal, vol. 25, no. 4, pp. 465–470, 1982

NON TESTABLE PROGRAMS

- Since we invented the first programming language we had to deal with non-testable programs (think to an assembler or a compiler !)
- Notable examples:
  - programs that compute an answer
  - programs that produce too much data
  - programs for which the tester has a misconception

E. J. Weyuker, "On testing non-testable programs," The Computer Journal, vol. 25, no. 4, pp. 465–470, 1982

NON TESTABLE PROGRAMS

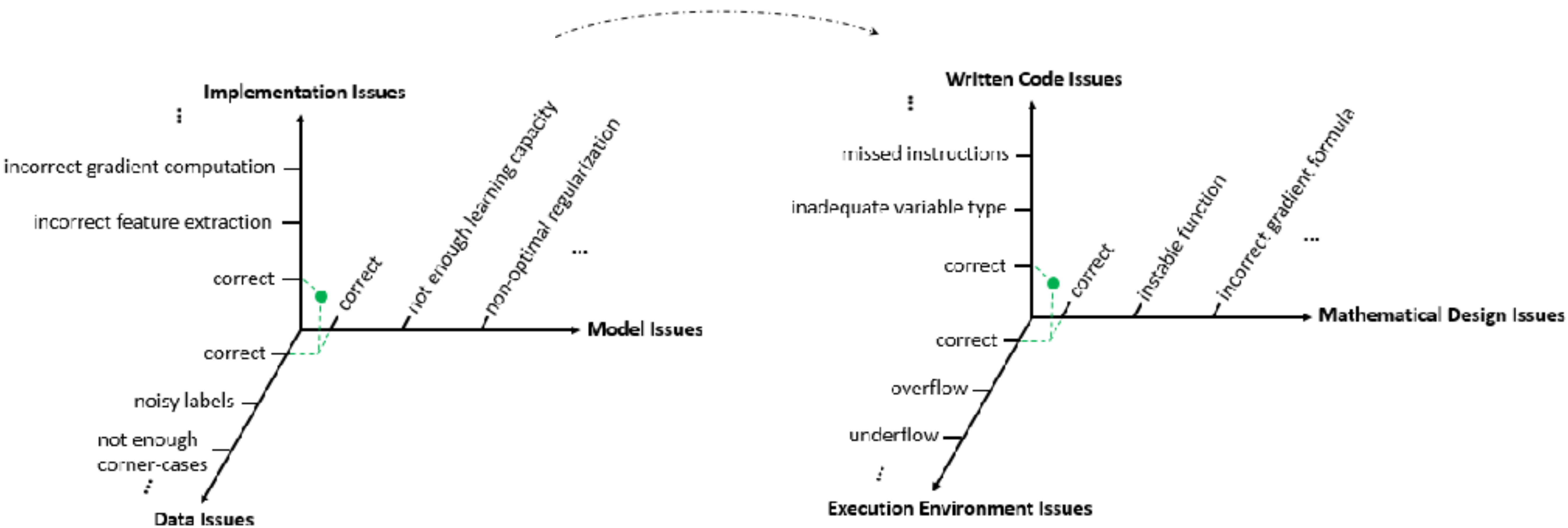
- Pseudo-oracles
  - If we cannot hope to have a full, non vague, precise specification
  - If we cannot reasonably check the output
  - If we do not have the “answer”

PSEUDO-ORACLE PROBLEMS

- Simulation programs
- Compilers
- Combinatorial optimizations
- NLP

Z. Q. Zhou, D. H. Huang, T. H. Tse, Z. Yang, H. Huang, and T. Y. Chen. Metamorphic testing and its applications. In Proc. of the 8th International Symposium on Future Soft- ware Technology (ISFST 2004), 2004.

ERROR SOURCES



Houssem Ben Braiek, Foutse Khomh On Testing Machine Learning Programs; arXiv:1812.02257

CONTROVERSIAL STATEMENT

- The ML/AI QA problem is not new at all
  - The Pseudo-oracle problem was there long before ML and AI
- Untestable programs are just more common
- Today what matter the most are data
- Without the data it may ba hard or impossible to interpret, explain, introspect or validate results

THE NEW ARISTOCRACY

- Have access to (labelled) training data
- Can define architecture and model
- Have enough resources to materialize the model

- Rely on 3-d party components
- Do not have access to (labelled) training data
- Resources may not be there

SW PRODUCTION AND ML/AI

The double speeds contradiction

DEEP LEARNING CONTRADICTION

- Training a DNN requires special hardware to accelerate computations

- To train on source code our SATD model required multiple GPUs and a couple of weeks just for one architecture configuration

- Finding the best configuration may be impractical

SOFTWARE 2.0

- Will traditional software disappear?
  - Likely not
- There are domains where we have plenty of labelled data for example a switch or light controllers, car engines
- Simply learn the desired behaviour
- If you have understanding of the problem and physical laws but the coding task is difficult while data are abundant software 2.0 can be the answer

DATA-OPS

- Data are the key for ML/AI we need new skills and expertises
- There are traditional ML algorithm that can (almost) fit right now in the DevOps cycle
- DNN is another story even with Google resurces

DATA-DRIVEN SW ENG

- The desired behaviour can be learn if we have enough data
  - and computational resources
- We need data engineers working with the traditional software engineers
- Curate collected data, ensure consistency, reliability and trustworthiness

DATA-DEV/OPS

- We need better and less expensive hardware
- We need better and faster training/adaptation algorithm
- We need better and faster testing approaches
- We need better visualization/introspection tools

END OF FIRST PART

ML AND MODELS

The imperfect reality

Code Complete: A Practical Handbook of Software Construction

a) Industry Average: "about 15 - 50 errors per 1000 lines of delivered code."

(b) Microsoft Applications: "about 10 - 20 defects per 1000 lines of code during in-house testing, and 0.5 defect per KLOC (KLOC IS CALLED AS 1000 lines of code) in released product (Moore 1992).''

(c) "Harlan Mills pioneered 'cleanroom development', a technique that has been able to achieve rates as low as 3 defects per 1000 lines of code during in-house testing and 0.1 defect per 1000 lines of code in released product (Cobb and Mills 1990). A few projects - for example, the space-shuttle software - have achieved a level of 0 defects in 500,000 lines of code using a system of formal development methods, peer reviews, and statistical testing."

SOFTWARE DEFECTS PREDICTION

- Multivariate logistic regression models

$$P[Y|X] = \frac{e^{\beta+\alpha_1 x_1 \ldots \alpha_n x_n}}{1 + e^{\beta+\alpha_1 x_1 \ldots \alpha_n x_n}}$$

- Poisson models

$$P[Y = y|X] = \frac{\lambda^y e^{-y}}{y!}$$

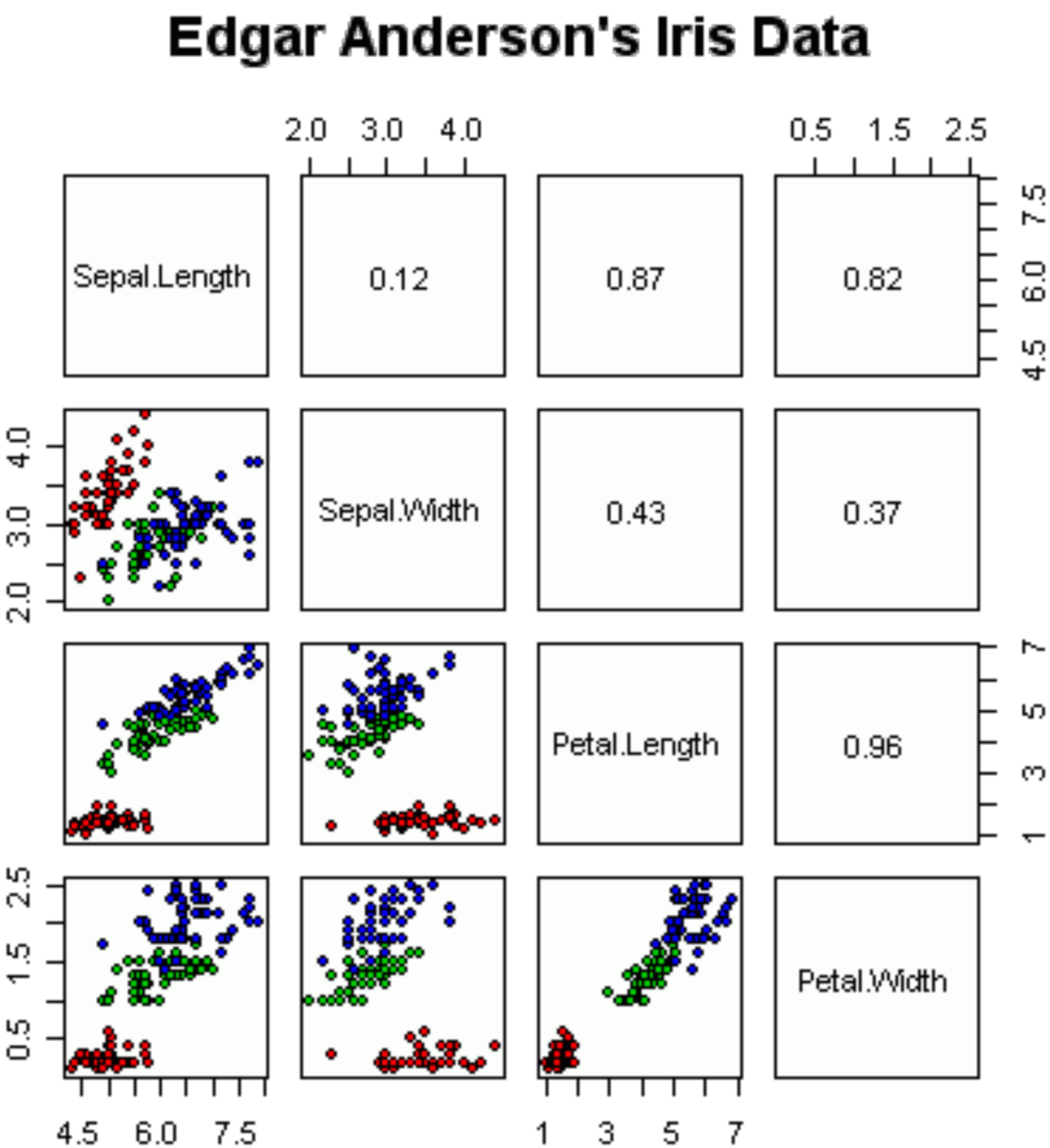
$$log(\lambda(X = x)) = \beta + \alpha_1 x_1 \ldots \alpha_n x_n$$

- Classification and regression trees

G. Canfora, A. De Lucia, M. Di Penta, R. Oliveto, A. Panichella, and S. Panichella. 2013. Multi-objective Cross-Project Defect Prediction. In *Proceedings of the 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation* (ICST '13). I

S. Kpodjedo, F. Ricca, P. Galinier, Y. Guéhéneuc, and G. Antoniol, "Design evolution metrics for defect prediction in object oriented systems," *Empirical software engineering*, vol. 16, iss. 1, pp. 141-175, 2011.

CLUSTERING AND MODELING




```
split = sample.split(dataset$Species, SplitRatio = .8)
training_set = subset(dataset, split == TRUE)
test_set = subset(dataset, split == FALSE)

training_set[,1:4] = scale(training_set[,1:4])
test_set[,1:4] = scale(test_set[,1:4])

classifier = svm(formula = Species~Petal.Width
            + Petal.Length,
              data = training_set,
                type = 'C-classification', kernel = 'radial')

test_pred  = predict(classifier1, type = 'response',
                     newdata = test_set[-5])

table(test_set[,5], test_pred1)

##                 setosa versicolor virginica
##   setosa          10          0         0
##   versicolor       0         10         0
##   virginica        0          2         8
```

R data exploration and visualization plus SVM classifier

DOES IT HOLD TRUE?

Cross-validation: estimate what will happen in the wild

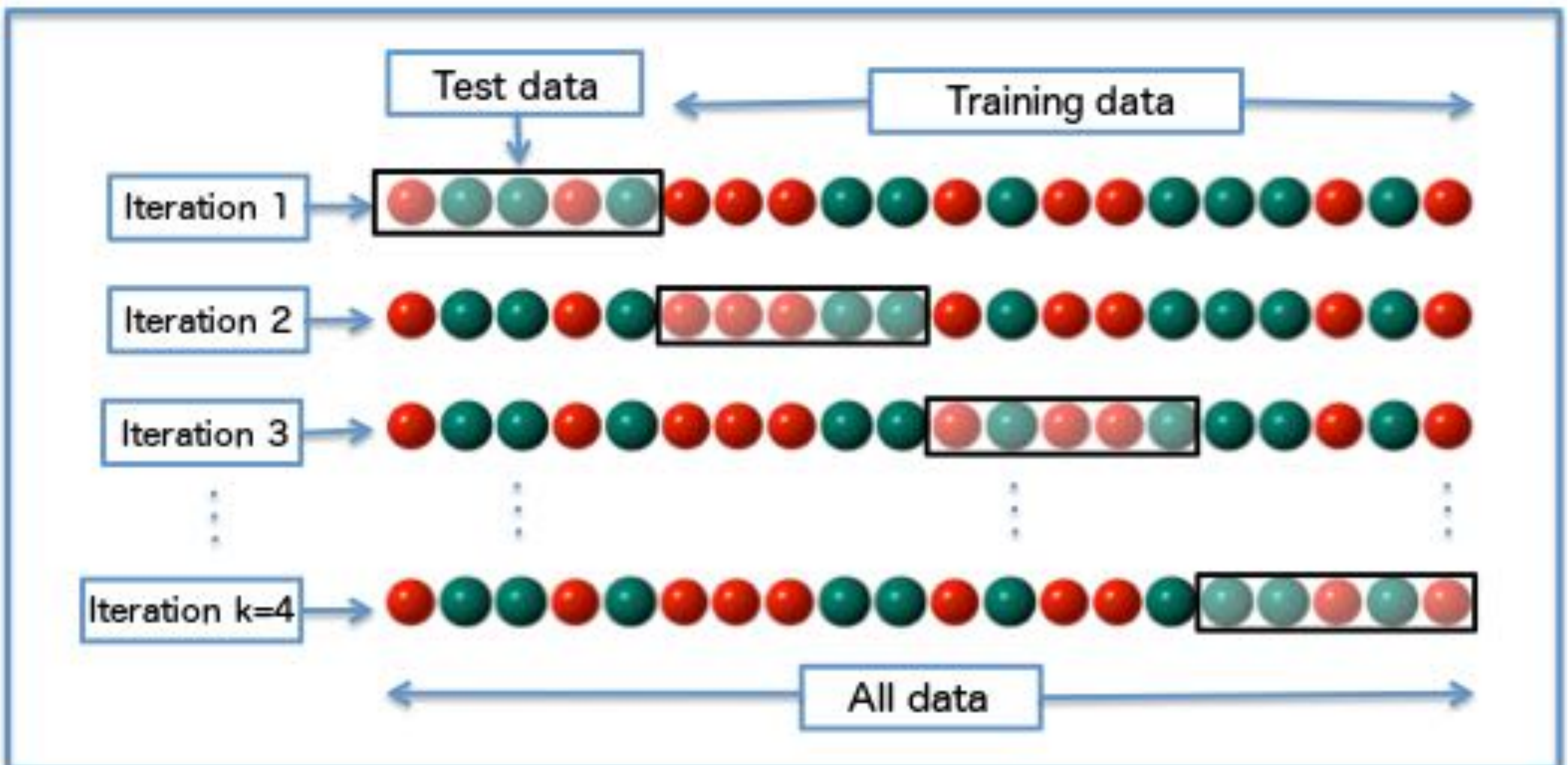
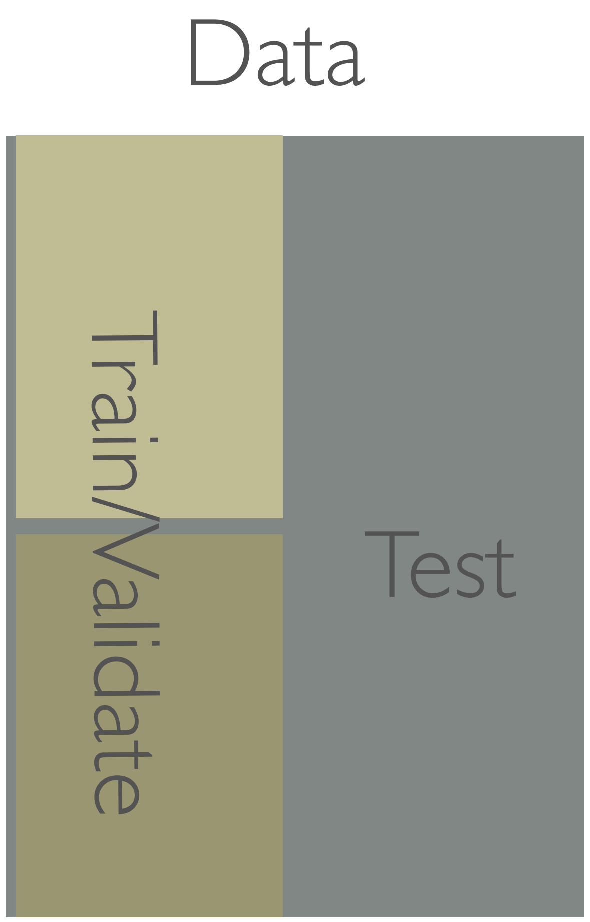


Image from Wikipedia

THREE IS BETTER THEN TWO

- Cross validation may be source of bias
- We need three (or more) sets
- Is it really what ML/AI is doing see Tensorflow training
  - It goes back to MIT media lab and Tomaso Poggio ideas



Cesare Furlanello, Maria Serafini, Stefano Merler, Giuseppe Jurman:
Entropy-based gene ranking without selection bias for the predictive classification of microarray data. BMC Bioinformatics 4: 54 (2003)

ML/AI METHODS EVALUATION

- We base our evaluation on well known and accepted metrics derived from the confusion matrix
- Hardly ever an approach is 100% correct
- Human also are often wrong why should we ask a machine be always correct ?

THE SOCIAL RISK

- We are somehow used to human errors
- A program failure may have catastrophic effects
- The user should be aware of what is under the hood and the associated risks or at least be warned

THE ULTIMATE CHALLENGE: TEST NON TESTABLE PROGRAMS

see E Weyuker 80s papers

UNDERSTANDING HUMAN SPEECH — THE PROTO AI ALGORITHM

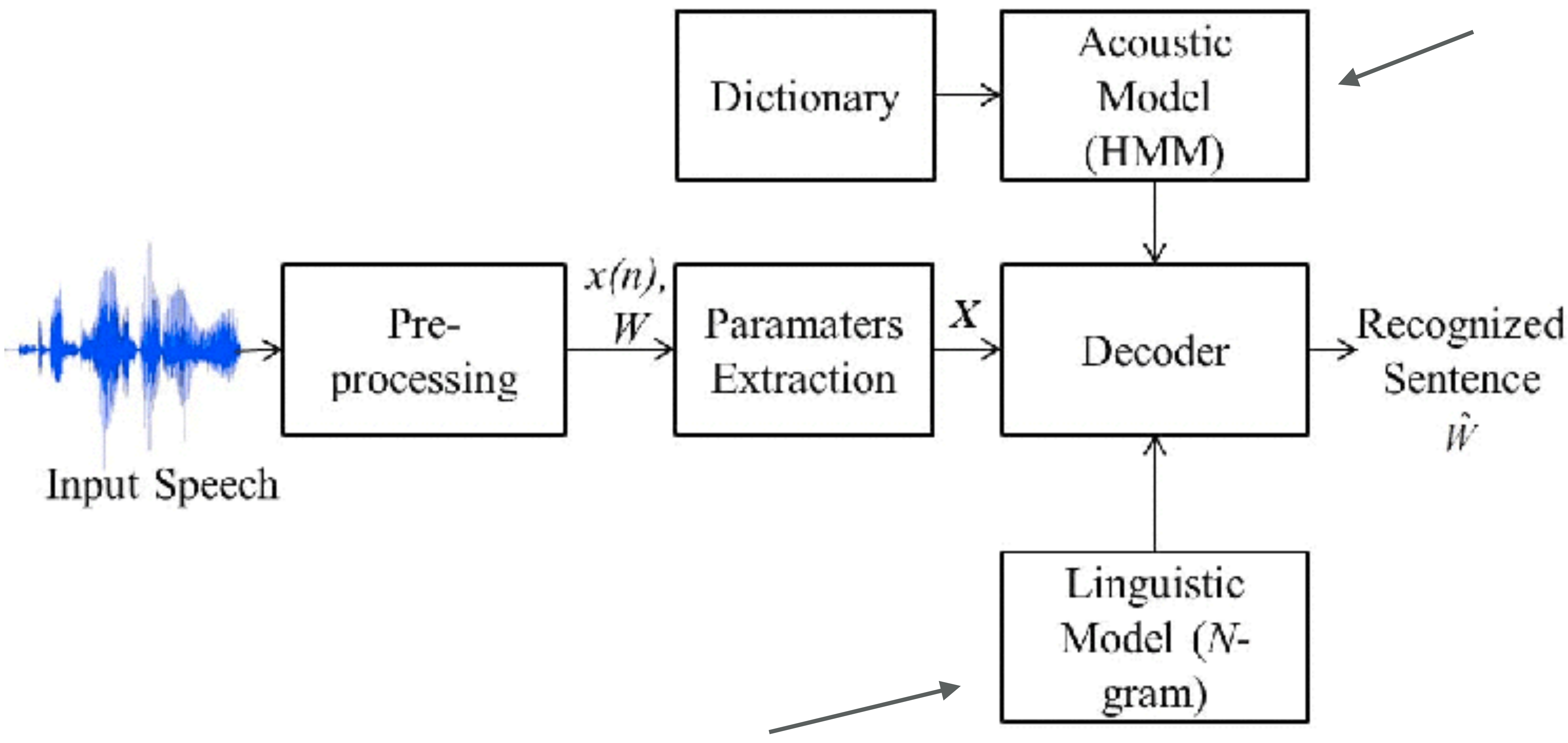
```
public static double YYY(double[] s, double[] t){
        double[][] matr = new double[s.length + 1][t.length + 1];

        matr[0][0] = 0;
        for (int i = 1; i <s.length + 1; i++) {
            matr[i][0] = inf;
        }
        for (int i = 1; i <t.length + 1; i++) {
            matr[0][i] = inf;
        }

        for (int i = 1; i < s.length + 1; i++) {
            for (int j = 1; j < t.length + 1; j++) {
                double cost = distanceD(s[i - 1], t[j - 1]);
                matr[i][j] = cost + minimum(matr[i - 1][j],   matr[i][j - 1],  matr[i - 1][j - 1]);
              }
        }
        return matr[s.length][t.length];
    }
```

Hermann Ney. 1990. The use of a one-stage dynamic programming algorithm for connected word recognition. In *Readings in speech recognition*, Alex Waibel and Kai-Fu Lee (Eds.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA 188-196.

THE ASR SYSTEM

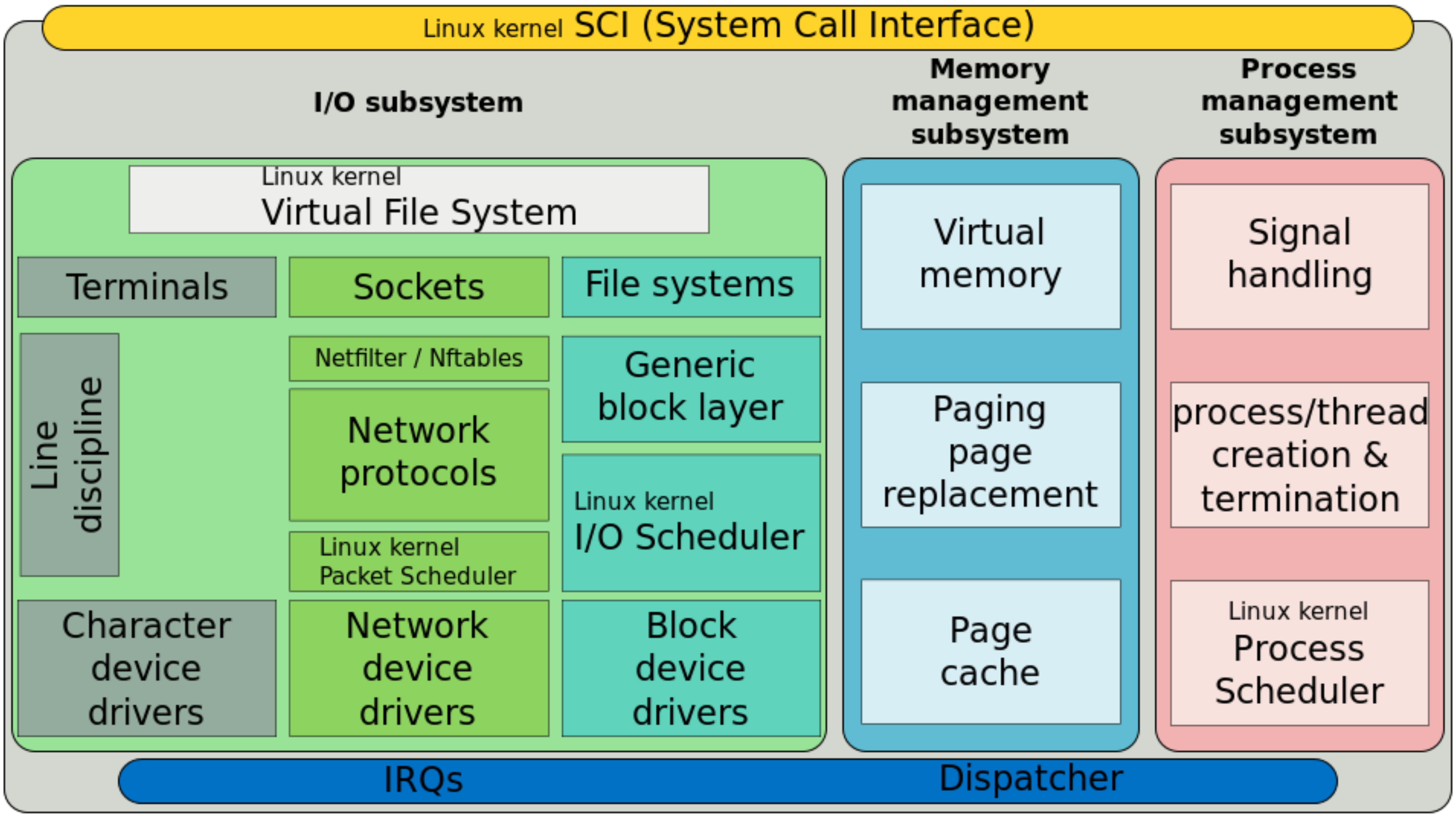


we are missing the entire "semantic" action part !

Lawrence R. Rabiner and Ronald W. Schafer. 2007. Introduction to digital speech processing. Found. Trends Signal Process. 1, 1 (January 2007),

G. Antoniol; Roldano Cattoni; Mauro Cettolo; Marcello Federico, Robust Speech Understanding for Robot Telecontrol, ICAR 1993 , pp. 205- 209, Tokyo, Japan

LINUX KERNEL SIMPLIFIES VIEW



To large output space!

Image from Wikipedia

THE CONUNDRUM

- Size does matter: 1000 LOC is easier to deal with than 10 MLOC
- Complexity and architecture matter too:
  - The Linux kernel is more complex than simple spoken language applications
- It is not black versus white box
  - not many people patch the Linux kernel (white box users)

MATTER OF FACT

- Large, complex long lived systems often do not have complete, precise, non vague specification
- ML is often used when the answer is not know
- The problem is exacerbated by the fact we do not know the “right” ML tool to use!

MAKING THINGS WORSE

- ML models debugging
- ML models introspection
- ML models are often not compositional
- ML models are the result of numerical approximation

Our experience with DNN:

Technical Debt refers to

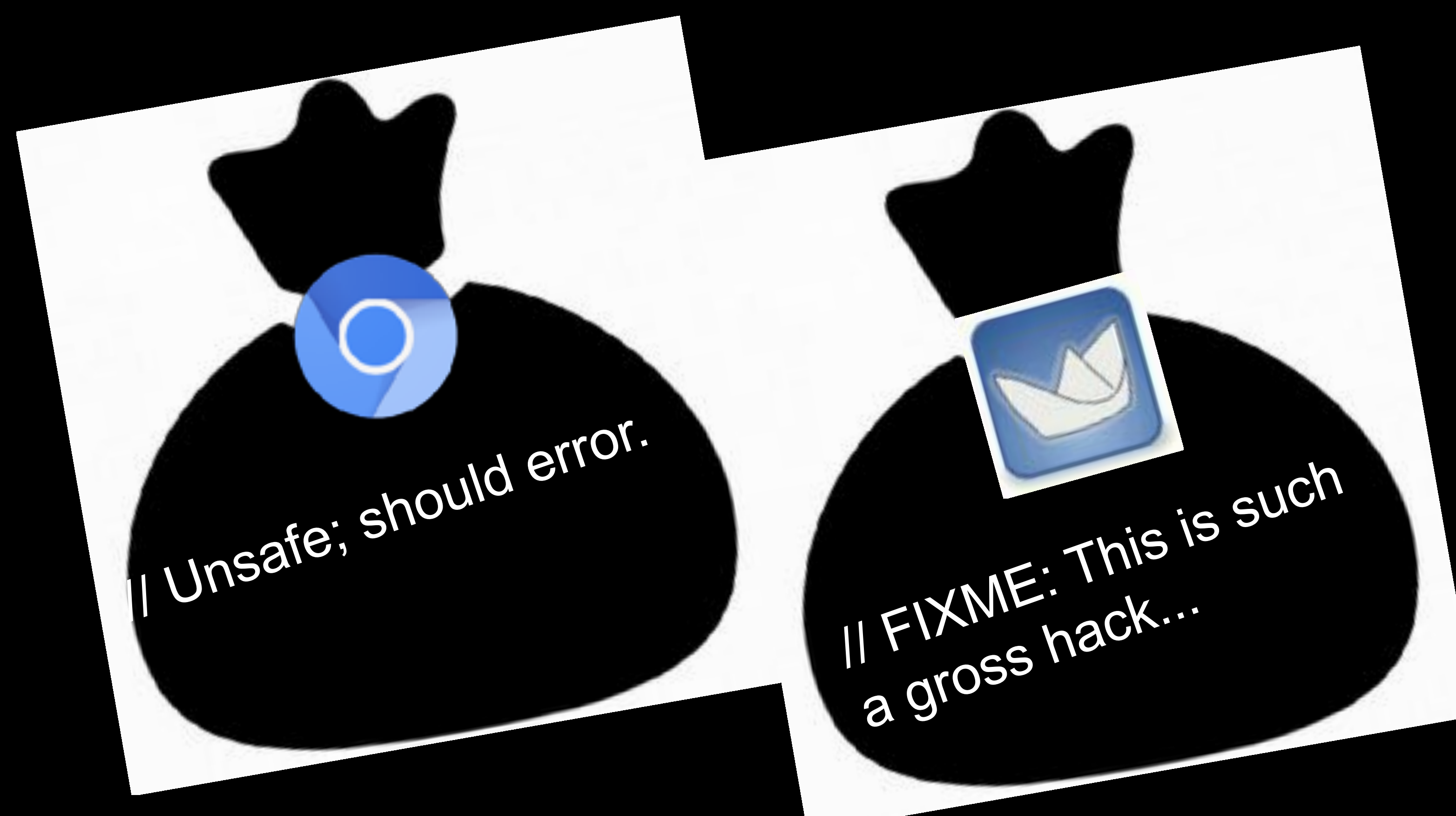
" not quite right code which we postpone making it right."
[Ward Cunningham]

Developers "self-admit" technical debt…


An Exploratory Study on
Self-Admitted Technical Debt

"… at the file level, between 2.4 - 31.0% of the files contained one or more instances of self-admitted technical debt."

Use self admitted debt as an oracle and build a deep network to recognize them



Deep Learning on Sentence Classification Problems

- Kim[1] explores CNNs and their performance compared to previous work using a variety of classifiers (RNN, Naïve Bayes, SVM)

➢In some cases, CNN is out performed by a manually tuned SVM

- Fu and Menzies[2] conduct a study on linked and duplicate posts from Stack Overflow comparing CNN performance to a tuned SVM

➢Compared to CNN, tuning SVM is about 84X faster

- Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," pp. 253–263, 2017.

[1]Y. Kim, "Convolutional neural networks for sentence classification," CoRR, 2014.

[2]W. Fu and T. Menzies, "Easy over hard: A case study on deep learning," in Proc. of the Joint Meeting on Foundations of Software Engineering (ESE), 2017, pp. 49–60.

SUBJECT SYSTEMS

Project	Release	Number of Files	Number of Classes	Number of Methods	Number of Comments	Number of Comments ∈ Methods	Number of Design SATD ∉ Methods	Number of Design SATD ∈ Methods	% of Methods with design SATD
Ant	1.7.0	1,113	1,575	11,052	20,325	13,359	1	57	0.5%
ArgoUML	0.34	1,922	2,579	14,346	64,393	17,722	203	425	2%
Columba	1.4	1,549	1,884	7,035	33,415	10,305	8	418	5%
Hibernate	3.3.2 GA	2,129	2,529	17,405	15,901	9,073	21	377	2%
jEdit	4.2	394	889	4,785	15,468	10,894	6	77	2%
jFreeChart	1.0.19	1,017	1,091	10,343	22,827	15,412	4	1,881	18%
jMeter	2.1	1,048	1,328	8,680	19,721	12,672	95	424	5%
jRuby	1.4.0	970	2,063	14,163	10,599	7,809	16	275	2%
Squirrel	3.0.3	2,325	4,123	16,648	25,216	15,574	35	173	1%

Traditional machine learning classifiers within project

Without Balancing

ML	Pr	Rc	F_1	Acc	MCC	AUC
Random Forests	49.97	52.19	47.15	93.32	0.47	0.92
Bagging	51.91	48.45	45.97	93.35	0.45	0.92
Bayesian	24.29	78.77	34.18	89.01	0.38	0.93
j48	34.86	54.42	39.54	94.18	0.39	0.82
Random Trees	23.09	52.49	29.96	90.35	0.30	0.73

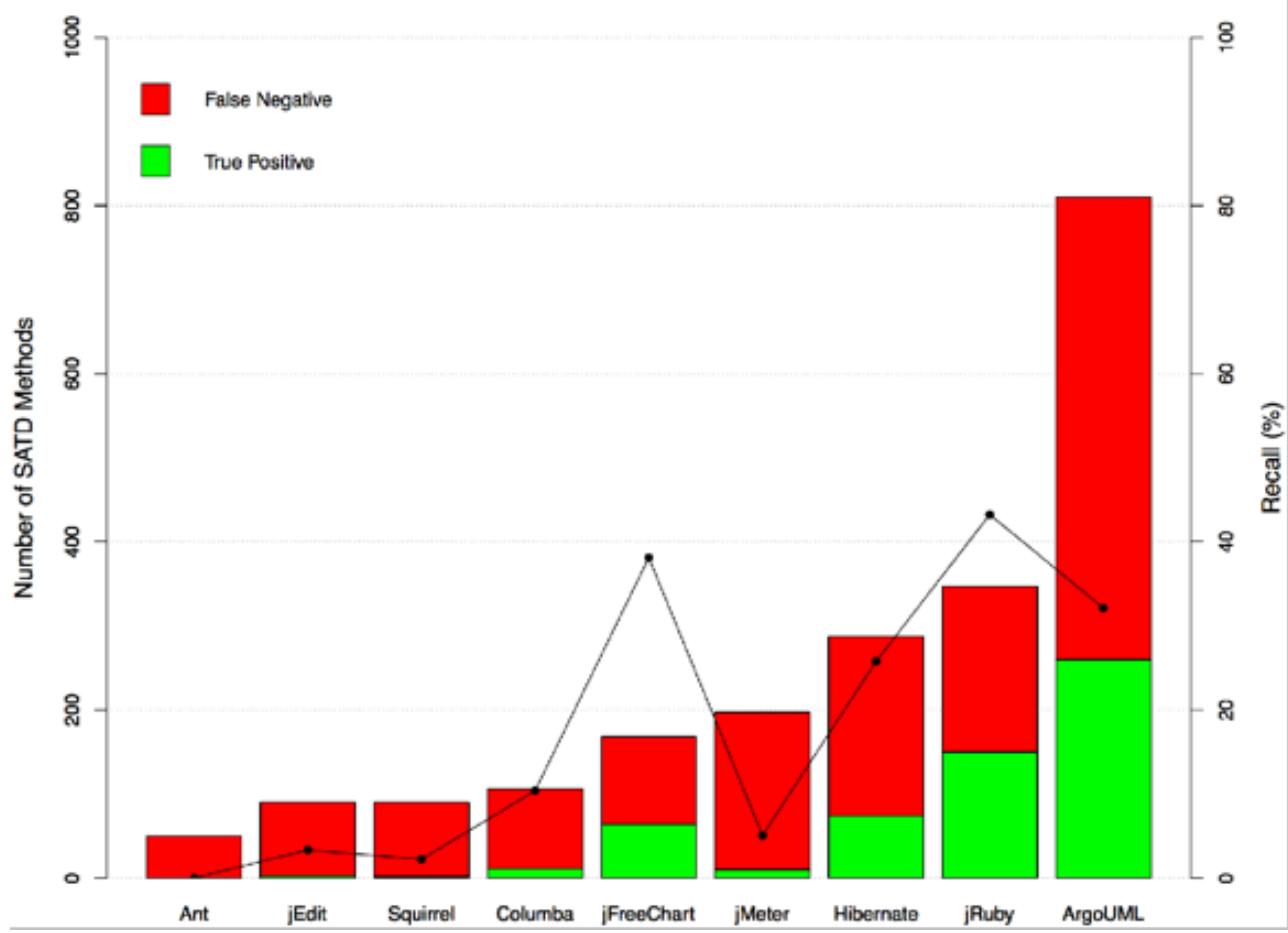
With Balancing

ML	Pr	Rc	F_1	Acc	MCC	AUC
Random Forests	26.56	68.26	36.04	90.45	0.37	0.92
Bagging	18.4	75.12	28.24	85.58	0.31	0.90
Bayesian	4.00	94.07	7.55	15.66	0.04	0.72
j48	16.95	77.76	26.45	84.04	0.30	0.85
Random Trees	16.03	63.22	24.49	85.34	0.26	0.75

Source Code Without Comments

System	Pr	Rc	F_1	Acc
Ant	0	0	0	99.52
ArgoUML	78.31	32.10	45.53	92.72
Columba	55.00	10.38	17.46	98.78
Hibernate	49.01	25.78	33.79	97.04
jEdit	37.50	3.33	6.12	98.29
jFreeChart	75.29	38.10	50.59	97.81
jMeter	31.25	5.08	8.73	97.87
jRuby	75.00	43.23	54.84	96.86
Squirrel	33.33	2.22	4.17	99.00
Total	**68.17**	**26.76**	**38.43**	**97.61**

CNN 128 filters3,4,5
Embeddings 150



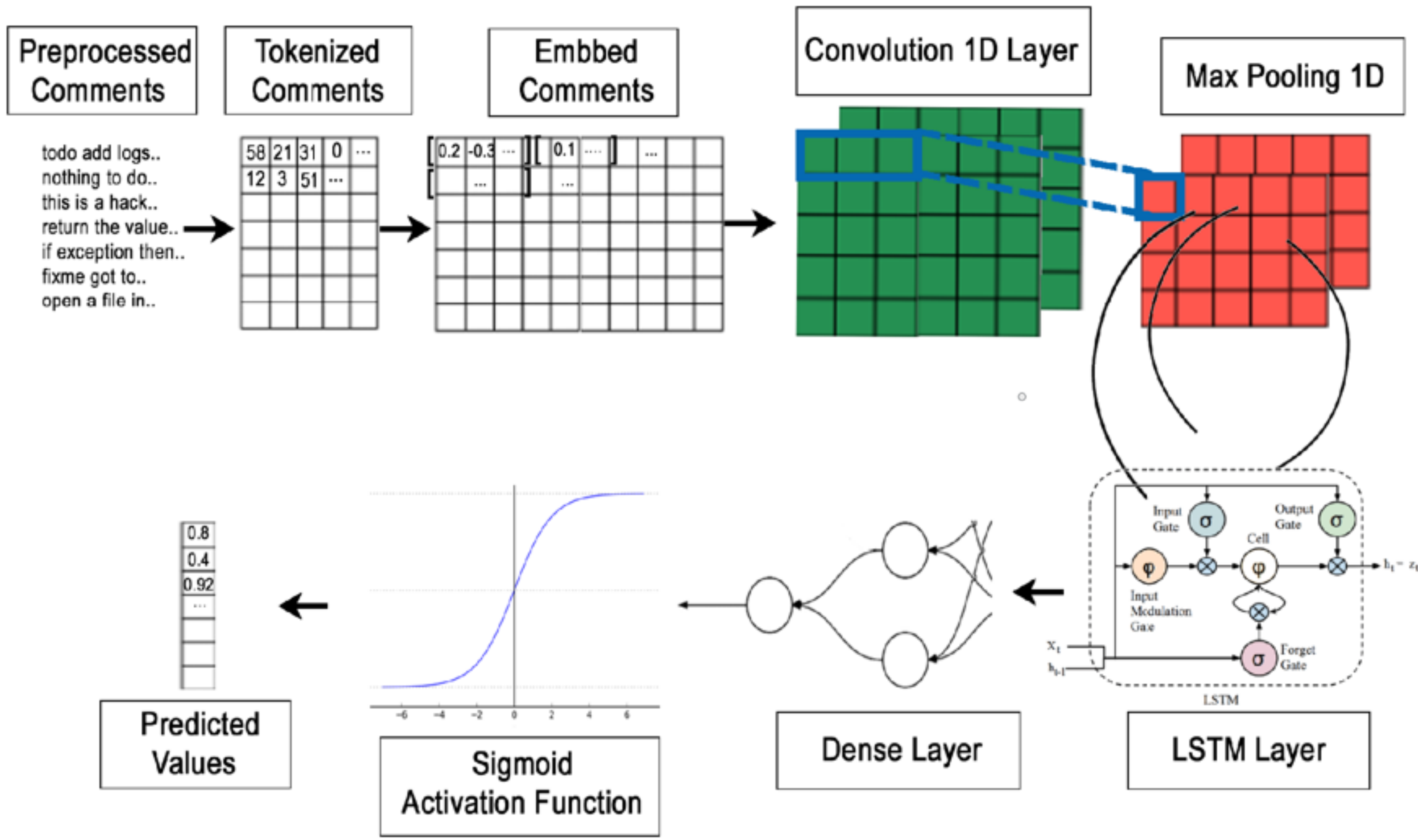
TIME AND MEMORY

TABLE V
TIME AND MEMORY OF CNN AND TRADITIONAL ML CLASSIFIERS

Configuration	Time (s)	Memory (GB)
16-2-3-4-5	570.50	89.42
64-3-3-3	387.26	87.18
16-3-3-3	281.93	86.48
32-4-4-4	342.43	96.12
128-5-5-5	647.48	94.10
Random Forest	4.36	1.80
J48 Pruned	0.35	1.80
J48 UnPruned	0.09	1.80
SMO RBF	2.79	1.80
SMO PUK	3.62	1.80
SMO Poly Kernel	16.77	1.80
SVM RBF	0.48	1.80
SVM Poly Kernel	0.29	1.80
Naive Bayes	0.19	1.80
Tuned SMO Normalized Poly kernel	5.02	1.02

WHAT IF WE ADD MEMORY

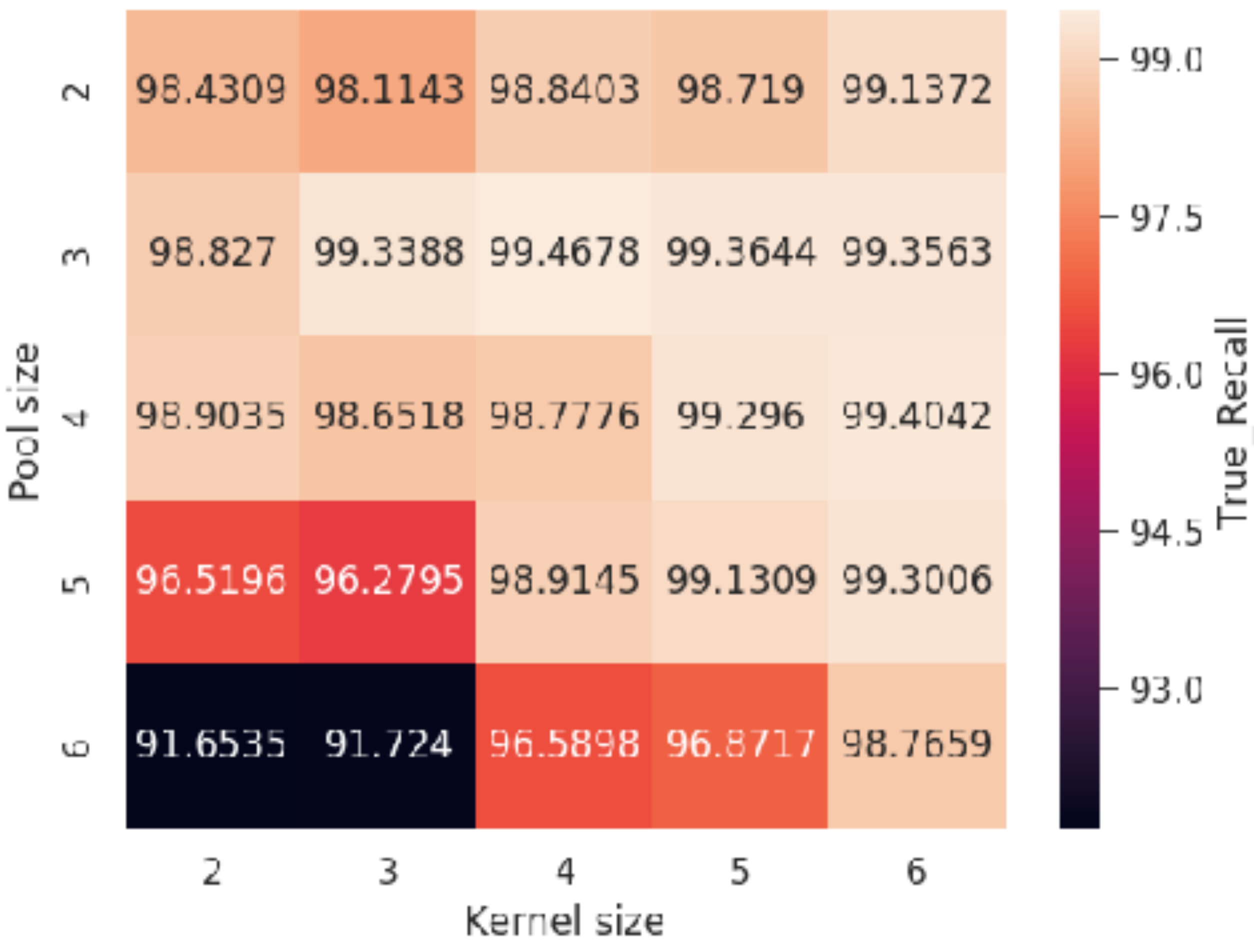
CNN - LSTM .. and cleanup the data

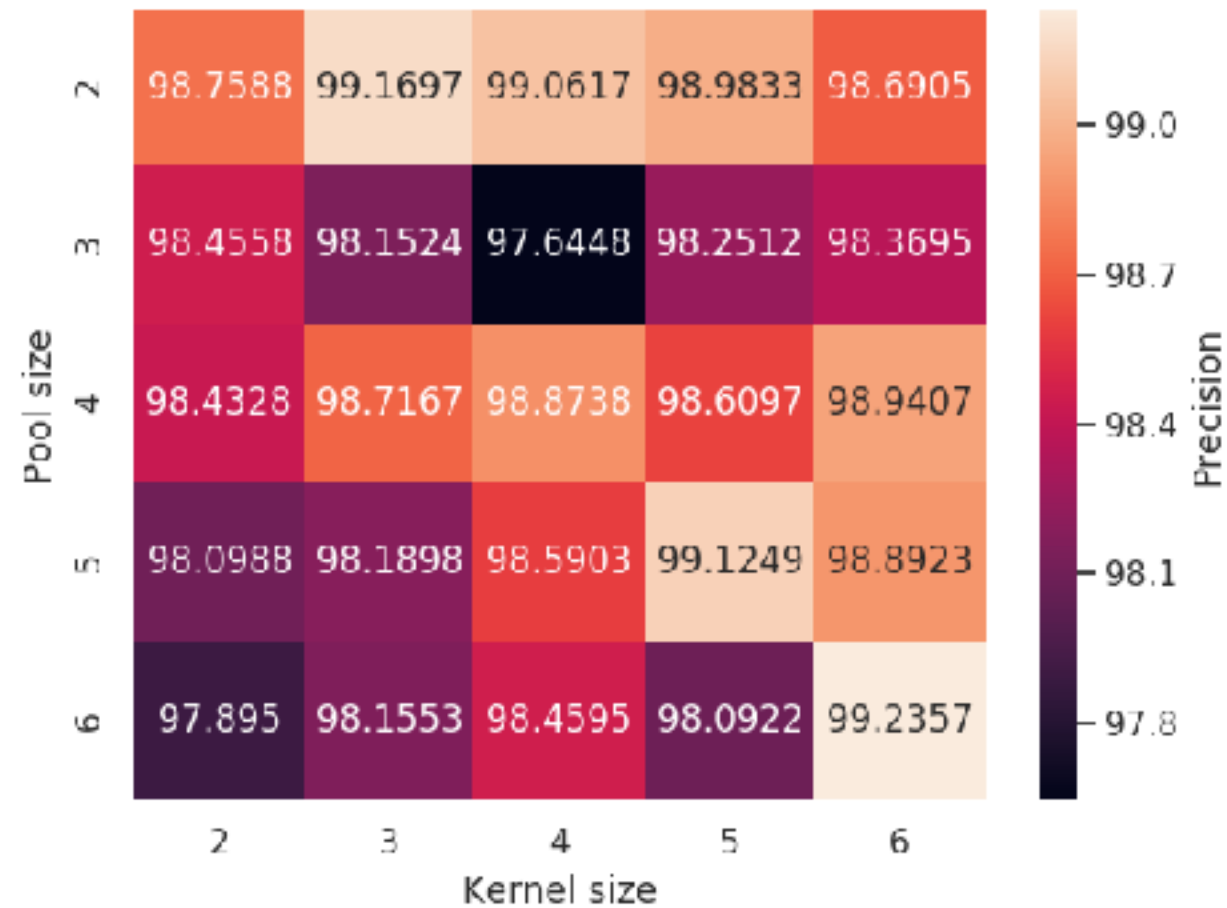


Preprocessed Comments
Tokenized Comments
Embbed Comments
Convolution 1D Layer
Max Pooling 1D
todo add logs..
nothing to do..
this is a hack..
return the value..
if exception then..
fixme got to..
open a file in..
58 21 31 0 ...
12 3 51 ...
0.2 -0.3 ... 0.1 ...
Input Gate
Output Gate
Cell
Input Modulation Gate
Forget Gate
LSTM
0.8
0.4
0.92
Predicted Values
Sigmoid Activation Function
Dense Layer
LSTM Layer

CONFIGURATION MATTERS

COMMENTS





10% left out	**RMSE**	**False Recall**	**True Recall**	**Precision**	**FMeasure**
Pool_size: 3 & Kernel_size: 4	0.158	99.696	63.928	93.294	87.144

INCREMENTAL TRANSFER

One project

	RMSE	False Recall	True Recall	Precision	FMeasure
ArgoUML : 0% of lines in train set	0.521	89.116	17.244	31.587	39.634
ArgoUML : 25% of lines in train set	0.575	78.495	30.591	31.059	39.580
ArgoUML : 50% of lines in train set	0.586	83.112	23.604	36.725	43.506
ArgoUML : 75% of lines in train set	0.648	87.814	11.747	38.235	43.251
ArgoUML : 90% of lines in train set	0.557	87.139	38.961	64.748	63.888

All projects

10% left out	RMSE	False Recall	True Recall	Precision	FMeasure
Pool_size: 3 & Kernel_size: 4	0.158	99.696	63.928	93.294	87.144

RANDOM EXTERNAL GITHUB PROJECT

- Train on the 9 projects and predict satd comments of a completely new project

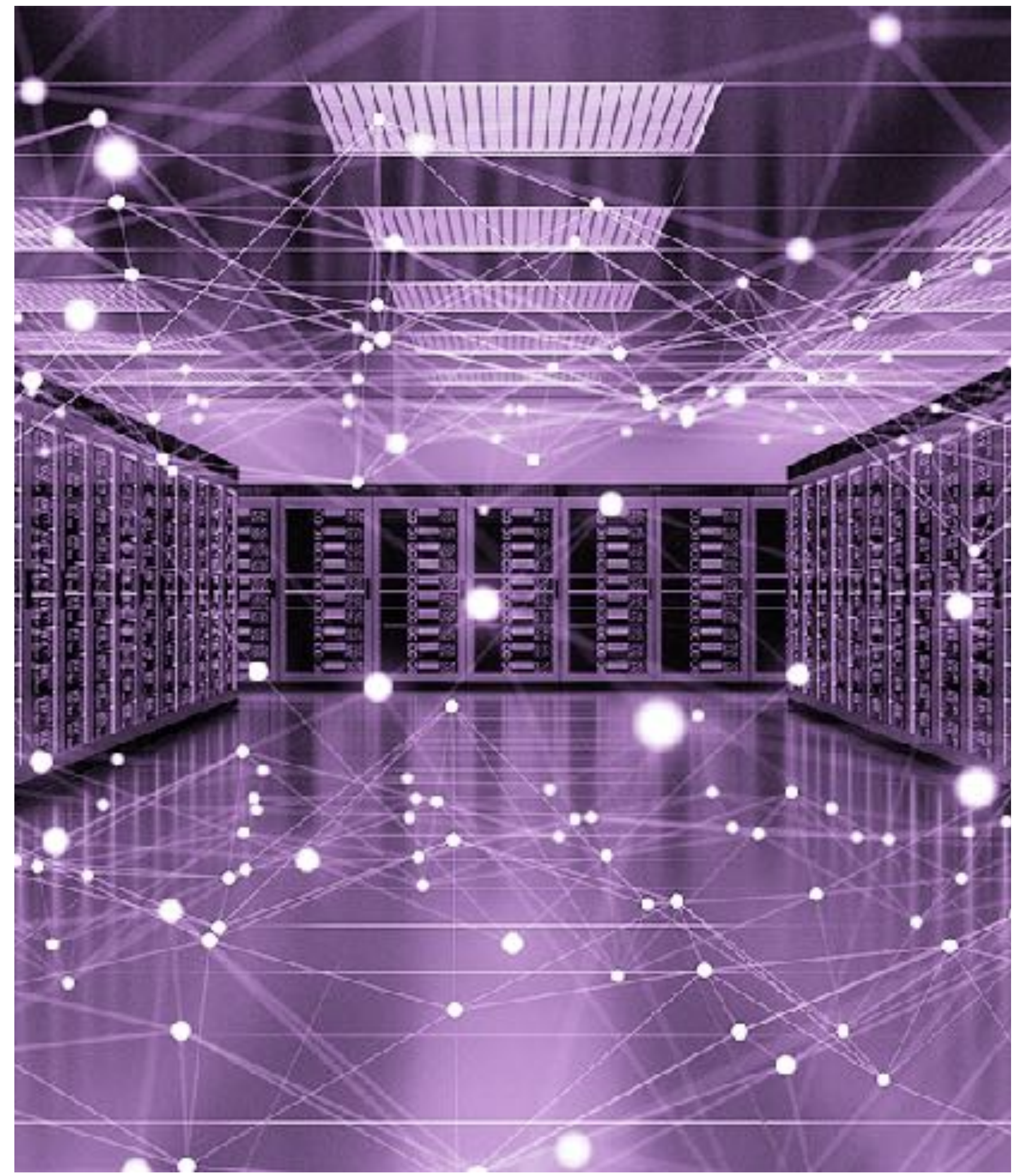
- Precision : 100.0, Recall : 58.33, F1 : 73.68

- But "should never be here" is it an SATD ? OK remove it
  Precision : 100.0, Recall : 87.5, F1 : 93.33

  - the model never encountered "should never be here" in the training material!

ARCHITECTURE AND DATA

- The architecture matters
- The data processing matters
- The data set quality is vital
- Different ML approaches requires substantially different resources



SOFTWARE ENGINEERING CRUX

- Traditional focus processes
- Root causes analysis focus on code or processes
- Data have seldom if ever been the focus
- COTS have been part of hour culture but
    - most COTS do not fall in the pseudo-oracle category
        - think to an ASR or implementing a chat boot

EDA SCALABILITY

Non testable program with too large output space or input domains are simply not suitable for manual validation

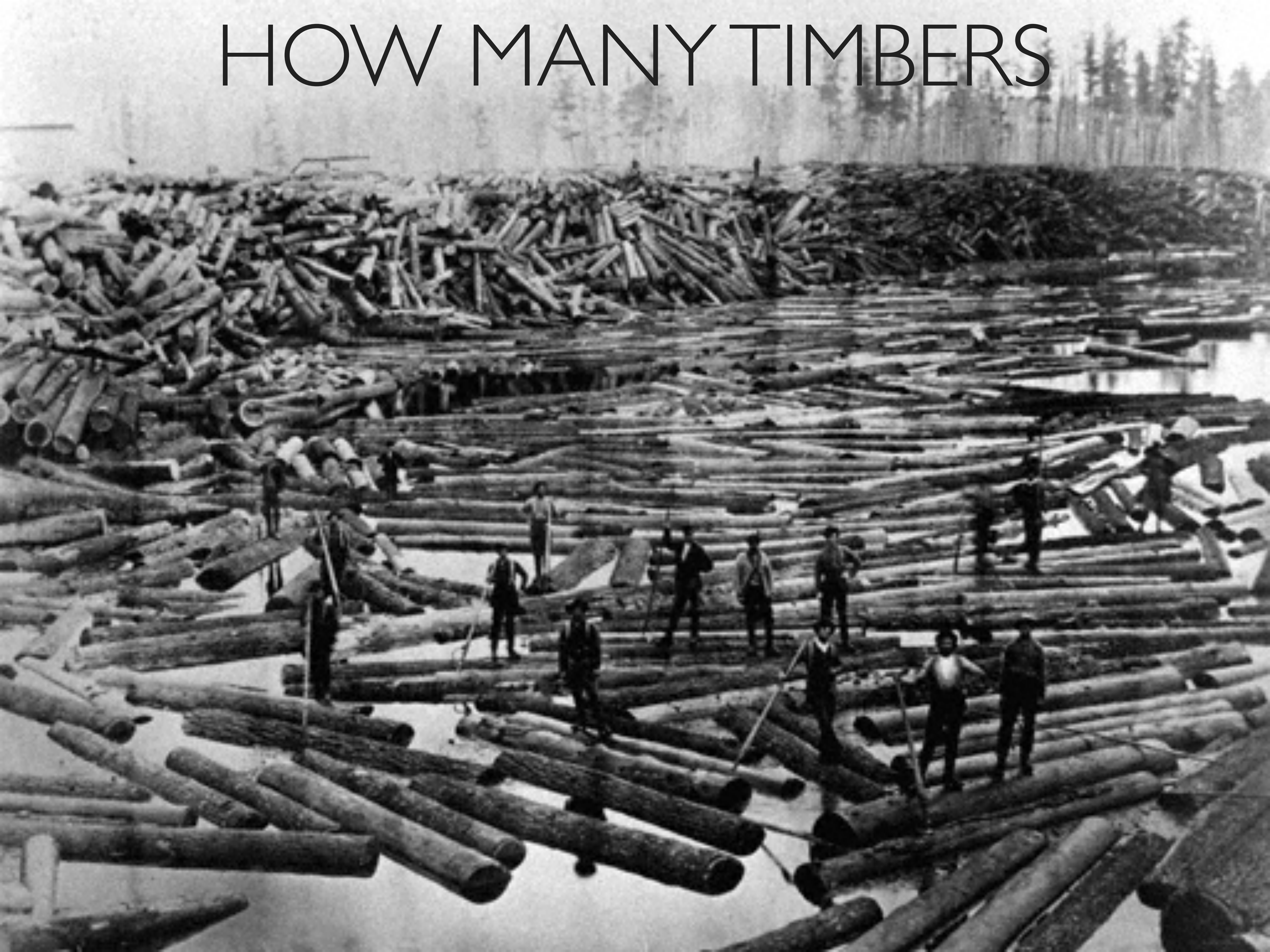
DATA: THE NEW GOLD

- Code is no longer relevant
- Data are the key
- A ML/AI component will be integrated into an environment
- Training data must reflect the deployment environment
- If training data do not represent context X we cannot expect the "right" behaviour

HOW MANY ROSES?

Miller, G. A. (1956). "The magical number seven, plus or minus two: Some limits on our capacity for processing information". Psychological Review. 63 (2): 81–97. doi: 10.1037/h0043158

HOW MANY TIMBERS



FACTS

- The IRIS dataset has 150 observations — four measured variables
- The SATD problem has between 80,000 and 100,000 methods and we have no clue what features DNN extracts
- EDA does not scale up

ML/AI CURSE OF DIMENSIONALITY

- What if we have a lot of observations but with few data — SATD and most common source code metrics

- What if we have a lot of observation and a highly dimensional space — SATD and we model code aka variable, identifiers, code structure
  - this dataset is likely to be sparse

PROPERTY INVARIANT BASED VALIDATION

- If the problem has a clear understanding and we are able to define invariants and property that should be always valid let's use this knowledge

- The speed of a mechanical controller must be always in a safe zone

- The approximation of a function may have a known boundary e.g. $x \in [0,2\pi]$ cos(x) in [0,1]

EXPLOIT WEAKNESSES

- To build more robust ML/AI components exploit known/ understood weaknesses
- We can adapt/transform input data to search for corner cases
  - What kind of transformation operator perform the best ?
  - Does in vitro results represent in vivo results?
- We need a deep understanding of the problem and weaknesses (e.g., effect of snow, rain or fog on images)

DEEPEXPLORE

- A clever application of Weyuker way to solve the pseudo-oracle problem
  - use two or more program instances for the same problem, then check the output for differences
- In essence very similar to back to back testing
- Two (or more) ML/DNN-components are tested together "forcing" one (or more) to behave differently
- Automatic transformation of the input to jointly maximize neurons coverage and decision difference
  - generate new inout test data where components disagree

LATE 90s - METAMORPHIC TESTING

- If we use supervised ML the pseudo-oracle problem can be lessened
- If we have labelled data it imply we know the answer for a subset of the data
- Why do not leveraging such knowledge ?

LATE 90s - METAMORPHIC TESTING - CONT

- Very promising approach
- Circumvent the oracle/pseudo-oracle problem
- We have some labelled data for which we know the answer
- Define metamorphic relations that holds true between input and output:
    - if we multiply a dataset by two the variance of the new dataset is also multiplied by 2

SHIFTING THE FOCUS

- We no longer need the oracle
- We need the metamorphic relations
- It may not ensure "corner" cases aka catastrophic events will never happen

DEEPTEST

- Clever use of a set of "reasonable" image transformation:
    - add rain, fog, lens distortion, blur
- Greedy combination of transformation to increase neurons coverage
- Enforce metamorphic relations
    - "recycle" the labels but change the data
        - rain or snow the road stretch is the same output should be the same but different people drive differently thus impose output are just very close (!)

DEEPROAD

- Improve over DeepTest use more realistic image transformation via a generative adversarial network and autoencoders:
  - add rain, fog, lens distortion, blur, …
- Enforce relaxed metamorphic relations
  - "recycle" the labels but change the data
    - rain or snow the road stretch is the same output should be the same but different people drive differently thus impose output are just very close (!)

CONCLUSIONS

- Although the horizon is changing fast the problem was know long ago
- We have initial and promising QA tools but more efficient and cost effective approaches/tools are needed
- We are shifting in direction of data driven sw eng

CONCLUSIONS - CONT

- There is a urgent need to address the data quality and data management issues
- SW eng and data eng should work together with data scientists
- We need to bridge the rift between domains

CONCLUSIONS - CONT

- Be aware of the risk and the need to make the user aware of the risks
- Investigate the sociological impact of the new types of systems where ML/AI play a major role

CONCLUSIONS - CONT

- How to avoid the gap between those that have knowledge and data and those that have not
- How to:
  - have better hardware and software training tools
  - better tool to introspect and understand the ML/AI output to feed the loop
  - move from DevOps to DataOps

CONCLUSIONS - END

...
geometrica ideo demonstramus, quia facimus, physica si demonstrare possemus, faceremus… G. Vico 1708. Lib. Methaph. Chap III

… Wir müssen wissen — wir werden wissen! … Hilbert 1930

They were wrong: the system cannot demonstrate its own consistency … Goedel 1931

- *Please read Parnas paper:*
    - *The Real Risks of Artificial Intelligence*

THANKS FOR YOUR ATTENTION