Data mining for Software Engineering
-- Some Final Remarks

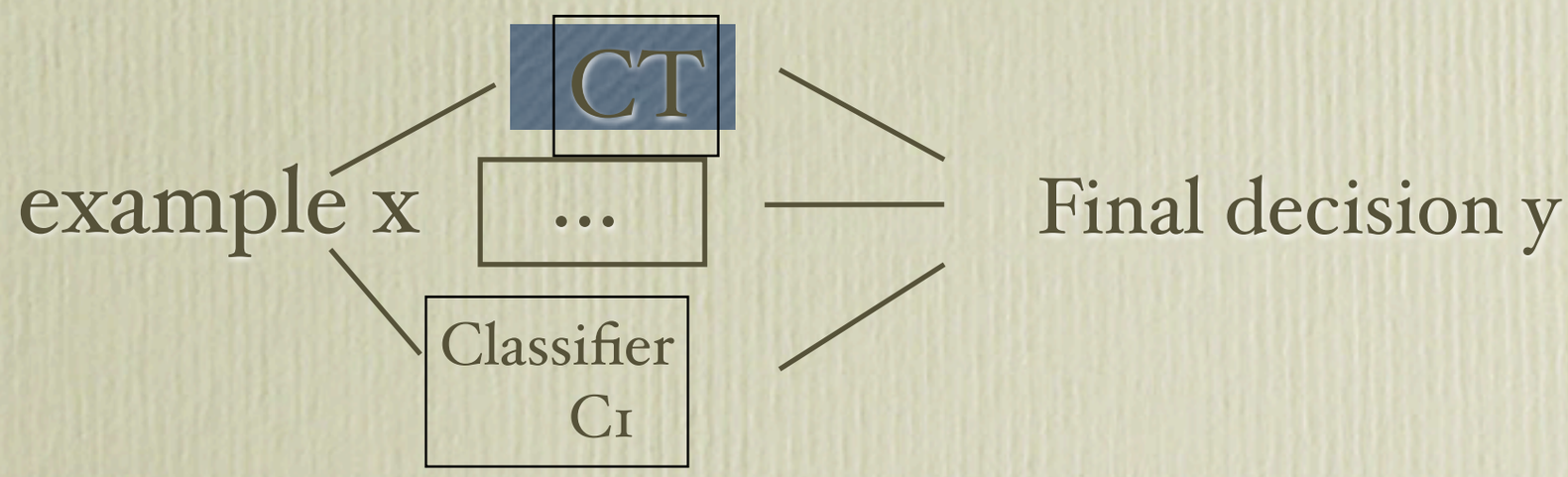
Jerzy Stefanowski
Poznań University of Technology
notes ver. 2010

Multiple Classifiers

- Typical research → create and evaluate a single learning algorithm; compare performance of some algorithms.
- Empirical observations or applications → a given algorithm may outperform all others for a specific subset of problems
- There is no one algorithm achieving the best accuracy for all situations! [No free lunch]
- Growing research interest in combining a set of learning algorithms / classifiers into one system
- „Multiple learning systems try to exploit the local different behavior of the base learners to enhance the accuracy of the

Multiple classifiers - definitions

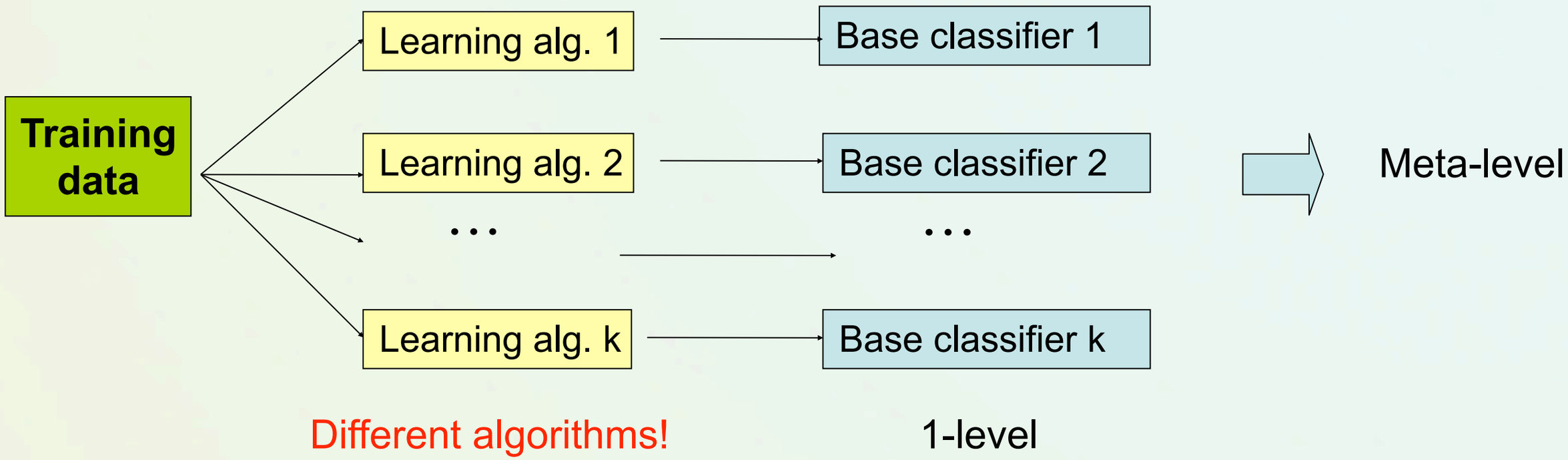
- Multiple classifier – a set of classifiers whose individual predictions are combined in some way to classify new examples.
- Various names: ensemble methods, committee, classifier fusion, combination, aggregation,...
- Integration should improve predictive accuracy!
- Diversity of component classifiers – if they make errors, then they should not correlated!



Approaches to create multiple systems

- Homogeneous classifiers – use of the same algorithm over diversified data sets
  - Bagging (Breiman)
  - Boosting (Freund, Schapire)
  - Multiple partitioned data
  - Multi-class specialized systems, (e.g. ECOC pairwise classification)
- Heterogeneous classifiers – different learning algorithms over the same data
  - Voting or rule-fixed aggregation
  - Stacked generalization or meta-learning

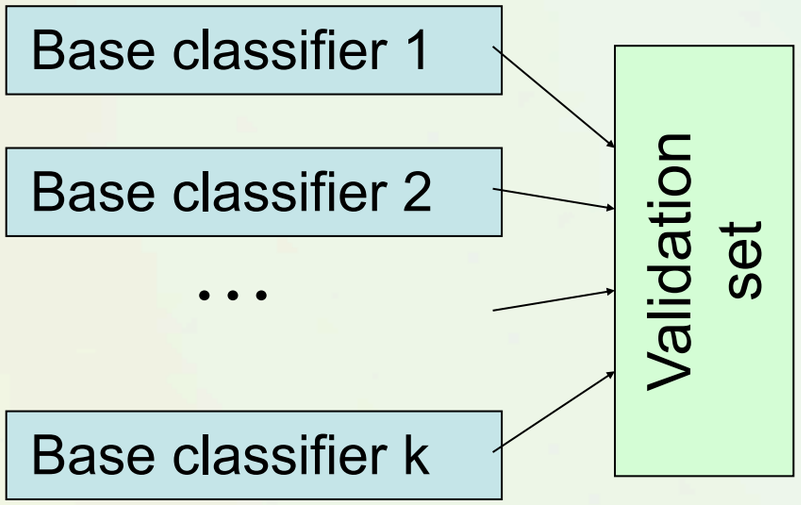
The Combiner Classifier - 1

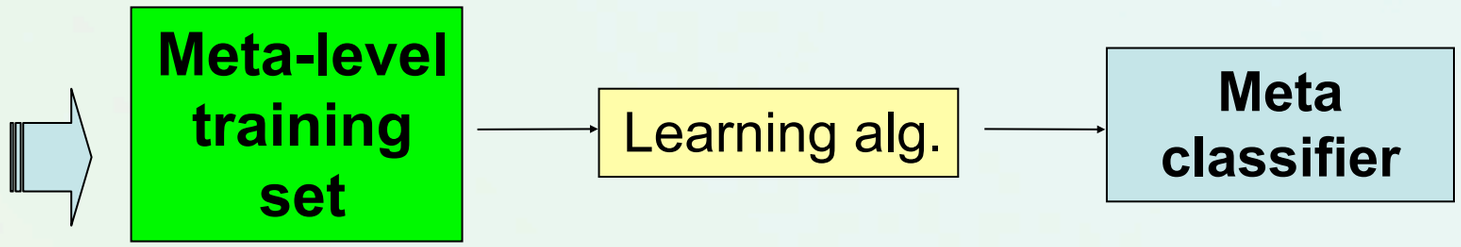


Chan & Stolfo : *Meta-learning*.

- Two-layered architecture:
  - 1-level – base classifiers.
  - 2-level – meta-classifier.
- Base classifiers created by applying the different learning algorithms to the same data.

Learning the meta-classifier

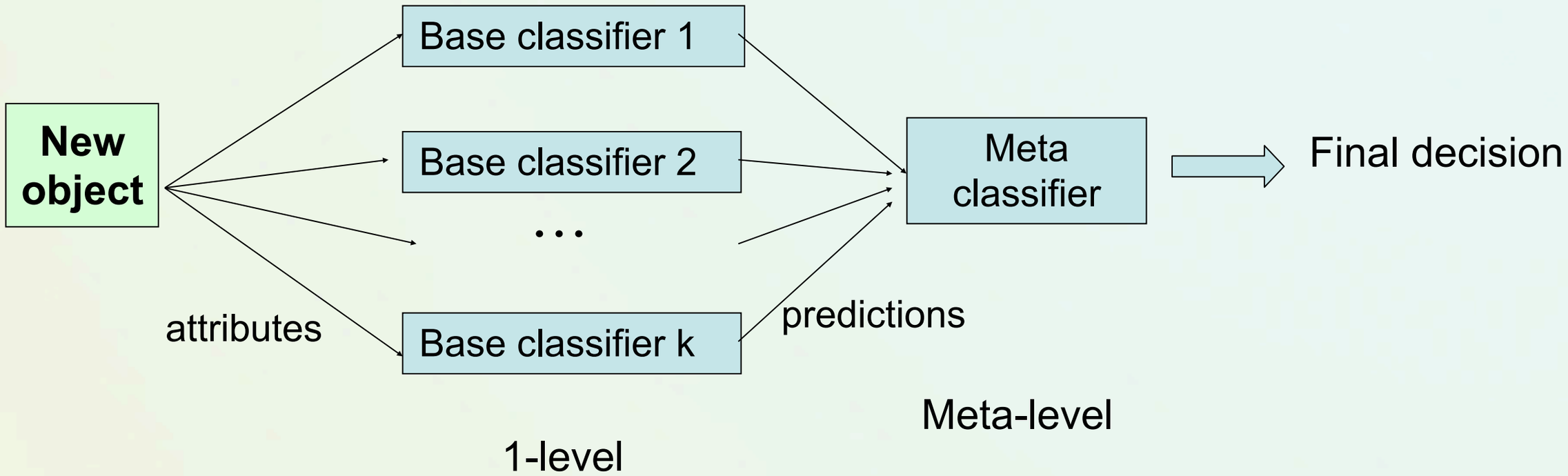




Predictions				Dec. class
Cl.1	Cl.2	…	Cl.K	
A	A	…	B	**A**
A	B	…	C	**B**

- Predictions of base classifiers on an extra validation set (not directly training set – apply „internal" cross validation) with correct class decisions → a meta-level training set.
- An extra learning algorithm is used to construct a meta-classifiers.
- The idea → a meta-classifier attempts to learn relationships between predictions and the final decision;
  It may correct some mistakes of the base classifiers.

The Combiner - 2



Classification of a new instance by the combiner

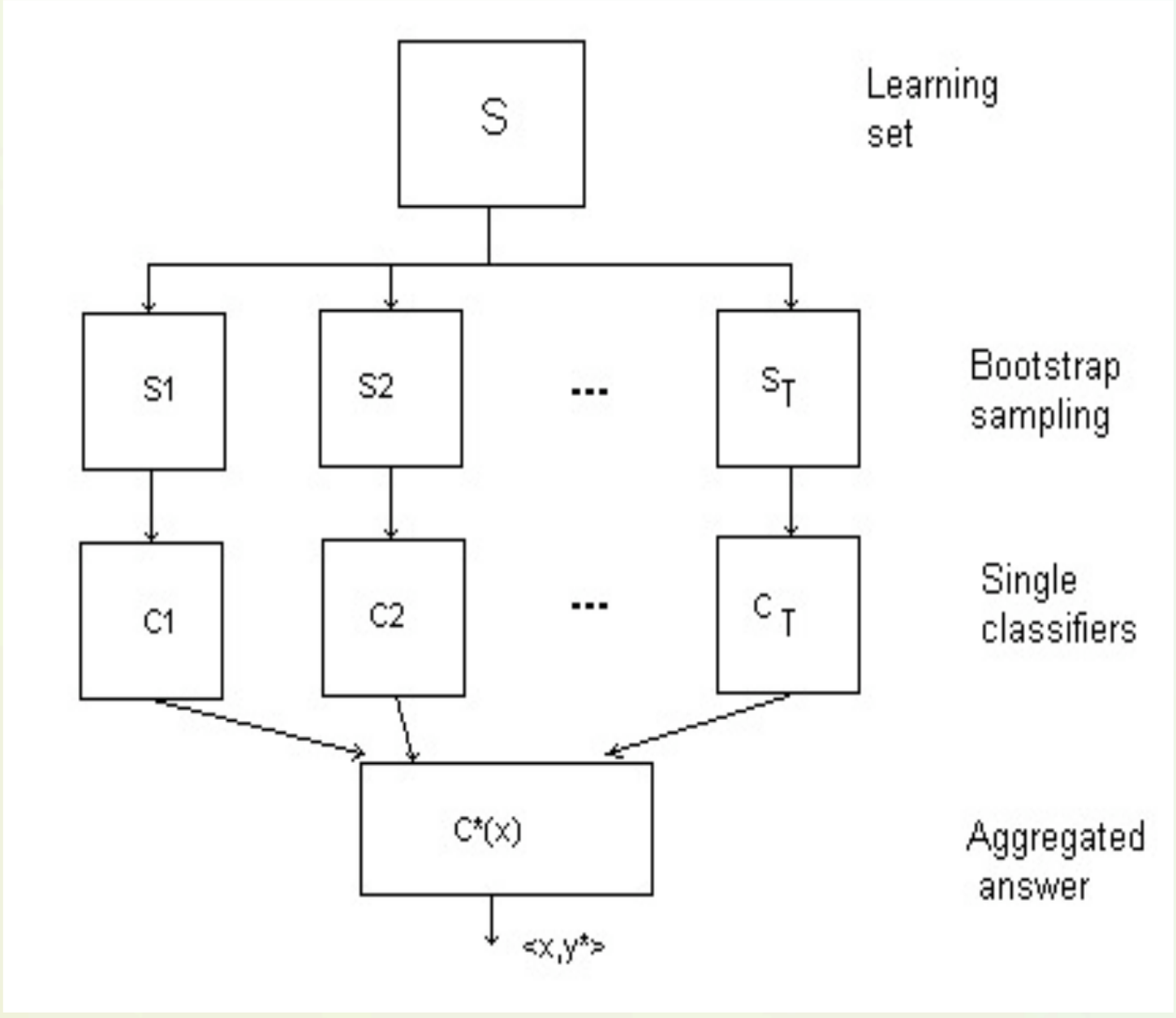
- Chan & Stolfo [95/97] : experiments that their combiner ({CART,ID3,K-NN}→NBayes) is better than equal voting.

Bagging [L.Breiman, 1996]

- Bagging = Bootstrap aggregation
- Generates individual classifiers on bootstrap samples of the training set
- As a result of the sampling-with-replacement procedure, each classifier is trained on the average of 63.2% of the training examples.
- For a dataset with N examples, each example has a probability of 1-(1-1/N)N of being selected at least once in the N samples. For N→∞, this number converges to (1-1/e) or 0.632 [Bauer and Kohavi, 1999]
- Bagging traditionally uses component classifiers of the same type (e.g., decision trees), and combines prediction by a simple majority voting across.

More about „Bagging”

- Bootstrap aggregating – L.Breiman [1996]



input S – learning set, T – no. of bootstrap samples, LA – learning algorithm

output C* - multiple classifier

for i=1 **to** T **do**

begin

S_i:=bootstrap sample from S;

C_i:=$LA(S_i)$;

end;

$$C^*(x) = \text{argmax}_y \sum_{i=1}^{T} (C_i(x) = y)$$

Boosting [Freund & Schapire]

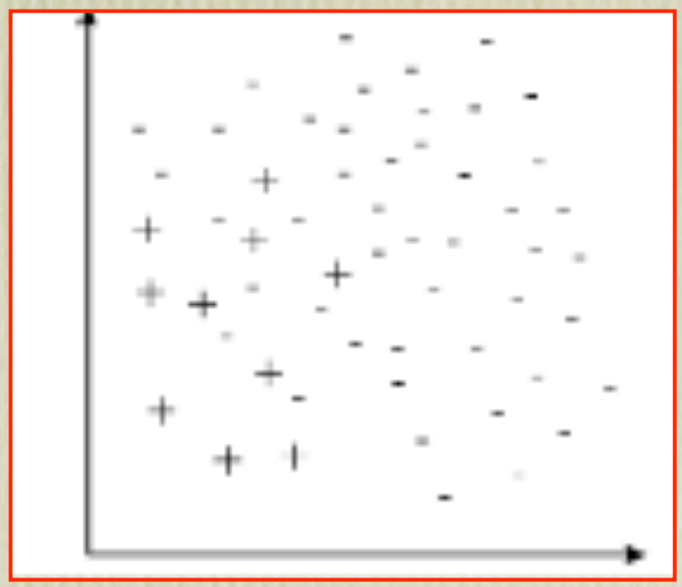
- In general takes a different weighting schema of resampling than bagging.
- Iterative procedure:
  - The component classifiers are built sequentially, and examples that are misclassified by previous components are chosen more often than those that are correctly classified!
  - So, new classifiers are influenced by performance of previously built ones. New classifier is encouraged to become expert for instances classified incorrectly by earlier classifier.
- There are several variants of this algorithm – AdaBoost the most popular (see also arcing).

Random forests [Breiman]

- Feature selection within bagging framework.
- At every level, choose a random subset of the attributes (not examples) and choose the best split among those attributes.
- Combined with selecting examples like basic bagging.
- Doesn't overfit.

Class imbalance

- Data set is said to present a class imbalance if it contains many more examples of one class than the other.
  - There exist many domains that do not have a balanced data set.
  - There are a lot of problems where the most important knowledge usually resides in the minority class.

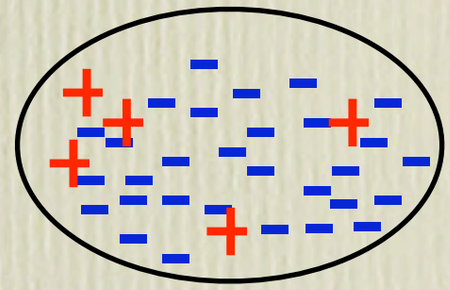


- Some real-problems: Fraudulent credit card transactions, Learning word pronunciation, Prediction of telecommunications equipment failures, Detection oil spills from satellite images, Medical diagnosis, Intrusion detection, Insurance risk modeling, Hardware fault detection

Imbalance → Difficulties

- Standard approach to learn classifiers such as decision tree induction are designed under assumption of partly balanced classes and to optimize overall accuracy without taking into account the relative distribution of each class.

- As a result, these classifiers tend to ignore small classes while concentrating on classifying the large ones accurately



Introduction to Imbalanced Data Sets

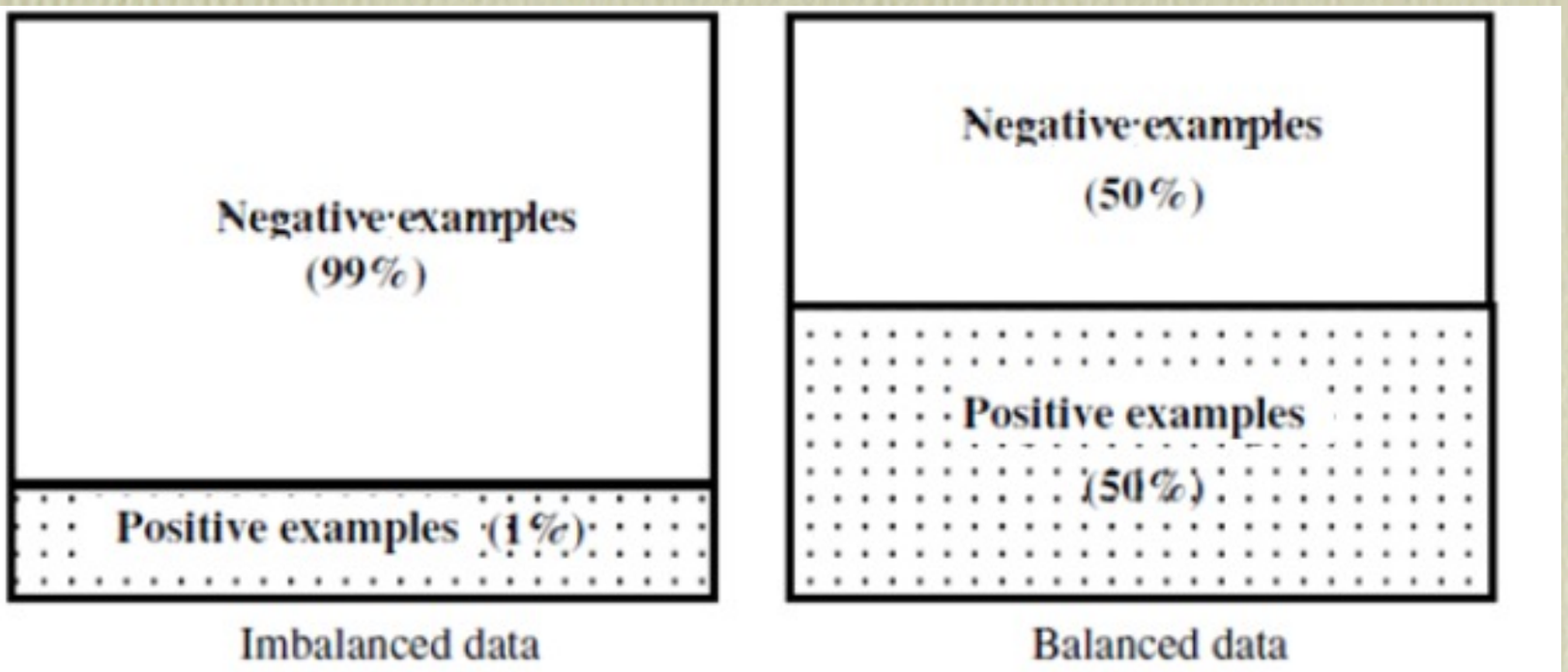


Fig. 1. Imbalanced and balanced data sets.

biased towards the majority class

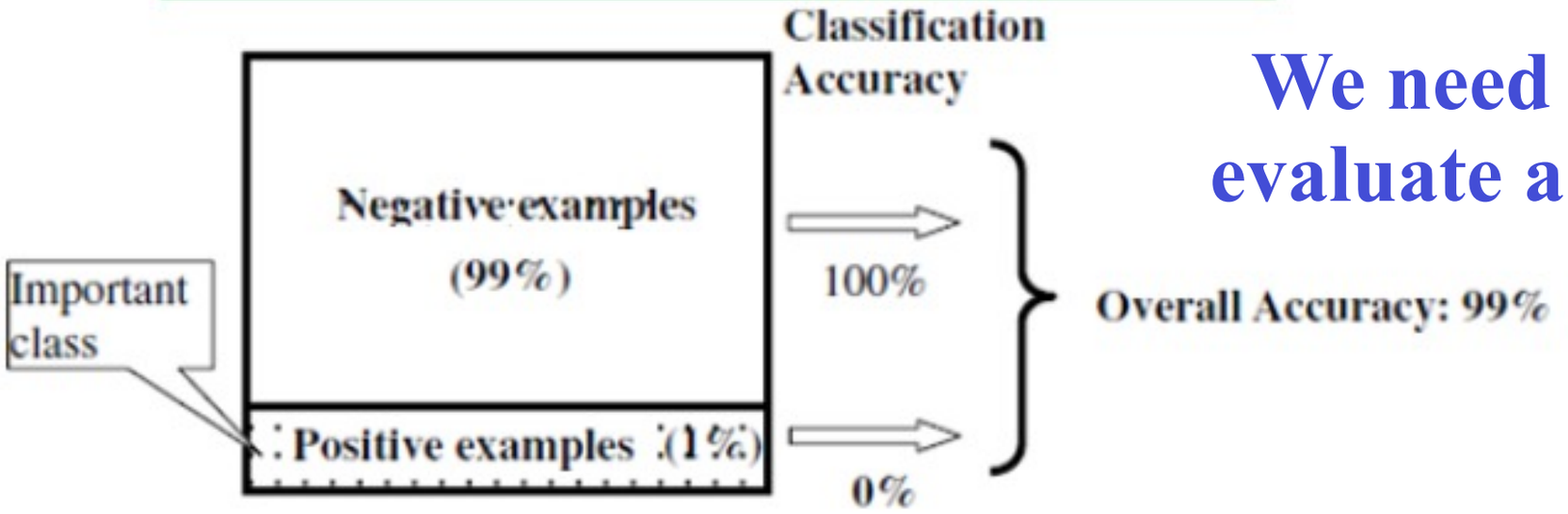


Fig. 2. The illustration of class imbalance problems.

We need to change the way to evaluate a model performance!

After Francisco Herrera lecture

Data level vs Algorithm Level

Strategies to deal with imbalanced data sets

Motivation

Over-Sampling
- **Random**
- **Focused**

→ **Retain influyent examples**
Balance the training set

Under-Sampling
- **Random**
- **Focused**

Text

→ **Remove noisy instances in the decision boundaries**
Reduce the training set

Cost Modifying (cost-sensitive)

Algorithm-level approaches: A commont strategy to deal with the class imbalance is to choose an appropriate inductive bias.

Boosting approaches: ensemble learning, AdaBoost, …

After Francisco Herrera lecture

Data level vs Algorithm Level

Strategies to deal with imbalanced data sets

Motivation

Over-Sampling

Random

Focused

Under-Sampling

Random

Focused

Text

Retain influyent examples
Balance the training set

Remove noisy instances in the decision boundaries

Reduce the training set

Cost Modifying (cost-sensitive)

Algorithm-level approaches: A commont strategy to deal with the class imbalance is to choose an appropriate inductive bias.

Boosting approaches: ensemble learning, AdaBoost, …

After Francisco Herrera lecture

Data level vs Algorithm Level

Strategies to deal with imbalanced data sets

Motivation

Over-Sampling
- **Random**
- **Focused**

Under-Sampling
- **Random**
- **Focused**

Text

Retain influyent examples
Balance the training set

Remove noisy instances in the decision boundaries
Reduce the training set

Cost Modifying (cost-sensitive)

Algorithm-level approaches: A commont strategy to deal with the class imbalance is to choose an appropriate inductive bias.

Boosting approaches: ensemble learning, AdaBoost, …

After Francisco Herrera lecture

Data level vs Algorithm Level

Strategies to deal with imbalanced data sets

Motivation

Over-Sampling
- **Random**
- **Focused**

Under-Sampling
- **Random**
- **Focused**

Text

Retain influyent examples
Balance the training set

Remove noisy instances in the decision boundaries
Reduce the training set

Cost Modifying (cost-sensitive)

Algorithm-level approaches: A commont strategy to deal with the class imbalance is to choose an appropriate inductive bias.

Boosting approaches: ensemble learning, AdaBoost, …

After Francisco Herrera lecture

Data level vs Algorithm Level

Strategies to deal with imbalanced data sets

Motivation

Over-Sampling
- **Random**
- **Focused**

Under-Sampling
- **Random**
- **Focused**

Text

Retain influyent examples
Balance the training set

Remove noisy instances in the decision boundaries
Reduce the training set

Cost Modifying (cost-sensitive)

Algorithm-level approaches: A commont strategy to deal with the class imbalance is to choose an appropriate inductive bias.

Boosting approaches: ensemble learning, AdaBoost, …

After Francisco Herrera lecture

Data level vs Algorithm Level

Strategies to deal with imbalanced data sets

Motivation

Over-Sampling
- **Random**
- **Focused**

Under-Sampling
- **Random**
- **Focused**

Text

Retain influyent examples
Balance the training set

Remove noisy instances in the decision boundaries
Reduce the training set

Cost Modifying (cost-sensitive)

Algorithm-level approaches: A commont strategy to deal with the class imbalance is to choose an appropriate inductive bias.

Boosting approaches: ensemble learning, AdaBoost, …

HANDLING IMBALANCED DATA

Difficulty factors in clinical data

- Challenges for data mining from clinical data
  - Missing and imprecise values, inconsistent examples
  - Uneven distribution of patients across decision classes

Minority class (usually critical) vs. **majority** classes → **class imbalance**

- Class imbalance deteriorates performance of classifiers learned from data (especially for the minority class!)
- Three groups of approaches to address this problem
  - *Data-level* methods – preprocessing before learning (more prevalent)
  - *Algorithm-level* methods – specialized learning algorithms
  - *Cost-based* methods – methods that consider costs of misclassifications (at different times)

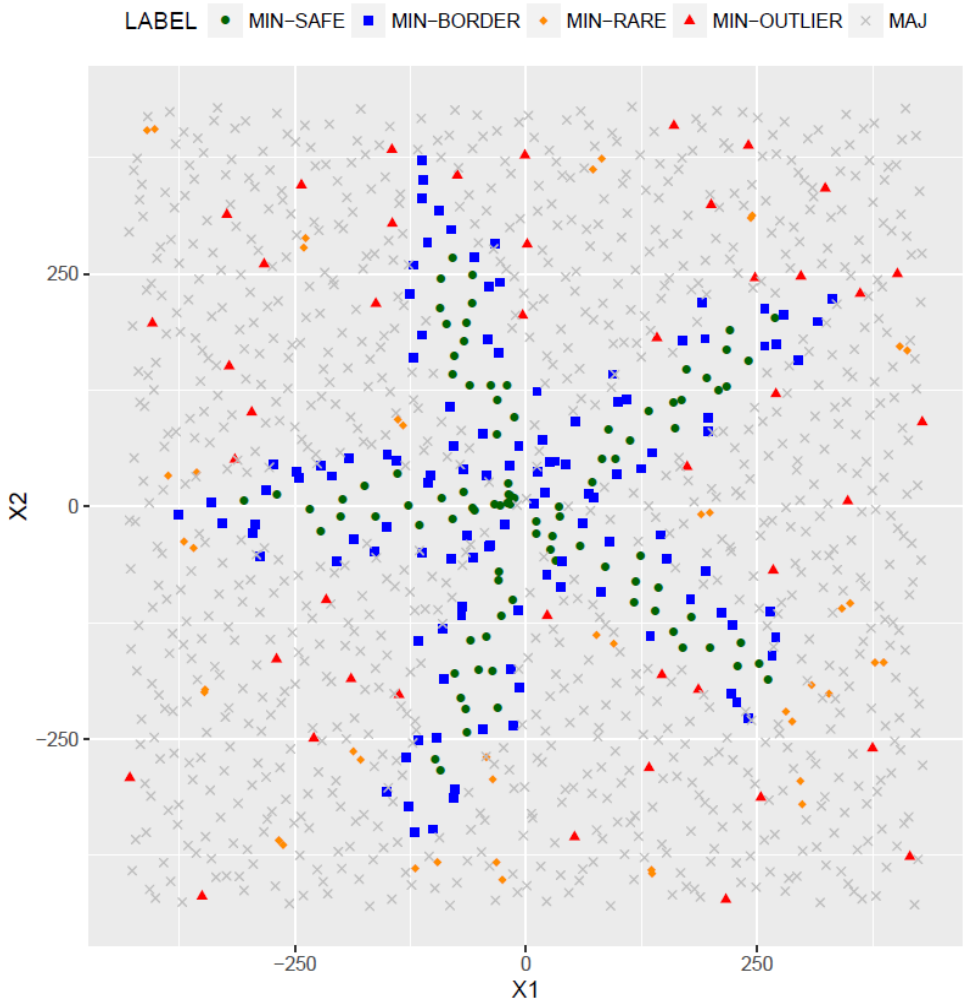
Dealing with imbalanced data

Class imbalance is not the only or main problem…

- Other data difficulty factors (affecting the minority class)
  - Overlapping regions between classes
  - Rare sub-concepts (→ *small disjuncts*) in the minority class and "outliers" thrown into the majority classes
- Identification of difficulty factors – tagging examples based on their local neighborhood [Napierała and Stefanowski, 2015]

Safe vs. *unsafe* (→ borderline, rare and outlier)

Types of examples capturing difficulty factors



Goal and research questions

Goal: evaluate and compare combinations of preprocessing methods and classifiers on clinical data

1. What are the data difficulty factors encountered in the analyzed clinical data sets?
2. How do the preprocessing methods improve the performance of obtained classifiers?
3. What are the best combinations of preprocessing methods and classifiers?

Special focus on the minority class

- Real-life clinical data sets collected in the ED at CHEO
- Common or relevant pediatric presentations
- Minority class indicates patients requiring quick care and significant resources

Wilk S., Stefanowski J., Wojciechowski S., Farion K.J., Michalowski W. (2016) Application of Preprocessing Methods to Imbalanced Clinical Data: An Experimental Study. In: Information Technologies in Medicine. ITiB 2016. Advances in Intelligent Systems and Computing, vol 471. Springer, Cham

Considered data sets

Data set	Clinical problem	# examples (minority)	Imbalance ratio	# attributes (numeric)
AP	Abdominal pain	457 (48)	0.11	13 (3)
HP	Hip pain	412 (46)	0.11	20 (4)
SP	Scrotal pain	409 (56)	0.14	14 (3)
AE1	Asthma exacerbations (2004)	362 (59)	0.16	32 (11)
AE2	Asthma exacerbations (2007)	240 (21)	0.09	42 (9)

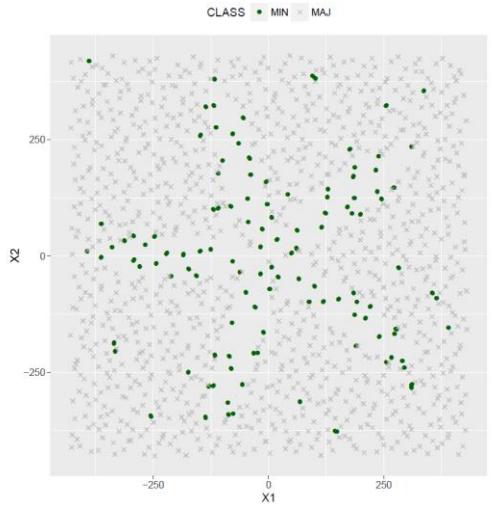
- Data collected retrospectively (HP, SP and AE1) and prospectively (AP and AE2)
- Removal of attributes with ≥ 50% of missing values (15 in SP, 10 in AE1)
- All non-critical classes combined into a single majority class

Experimental design

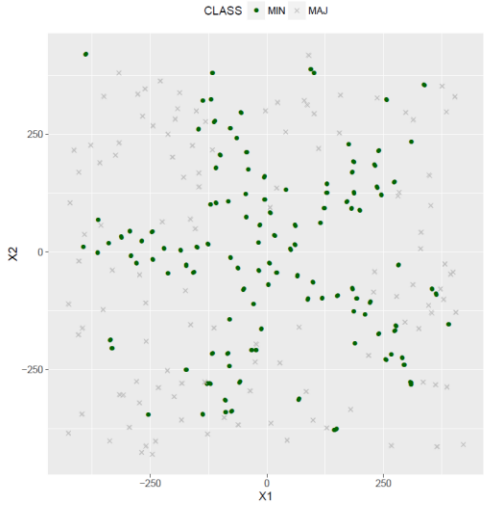
1. Identifying data difficulty factors in the data sets (by tagging examples with their types)
2. Evaluating the performance of selected combinations of preprocessing methods and classifiers on the data sets
   - Sensitivity, specificity and their geometric mean (G-mean, GM)
   - Stratified 10-fold cross validation repeated 10 times for reduced variance
   - Friedman test ($\alpha = 0.05$) to compare the performance of multiple combinations of preprocessing methods and classifiers over multiple data sets

Illustration of Preprocessing Methods

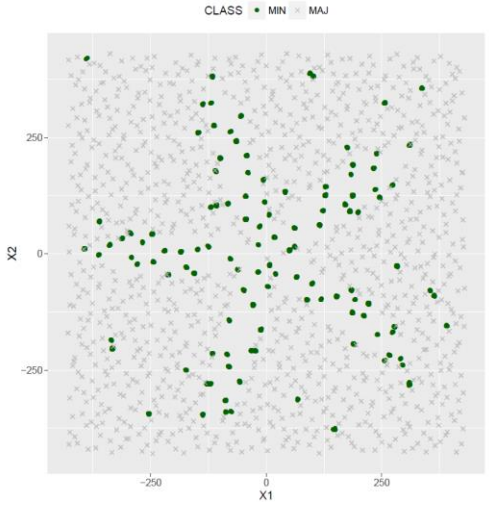
None



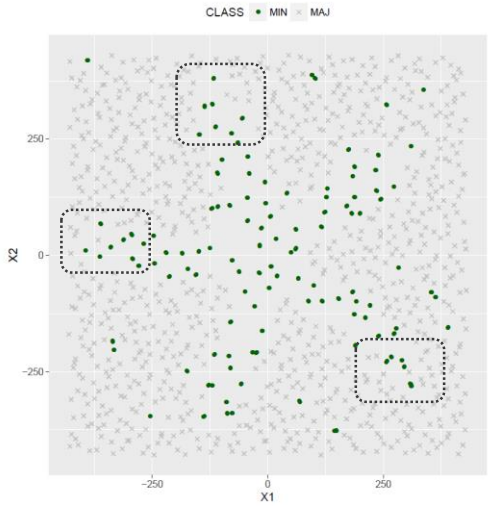
RU



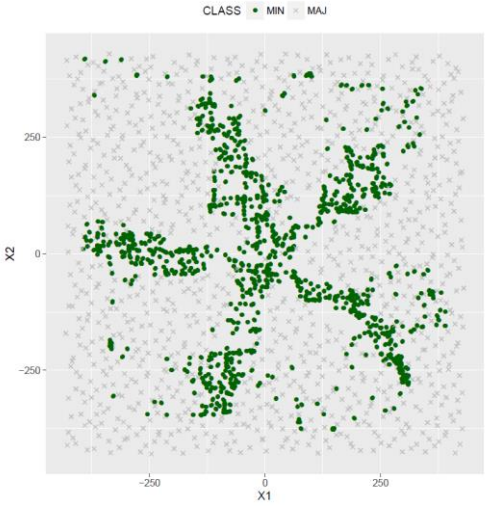
RO



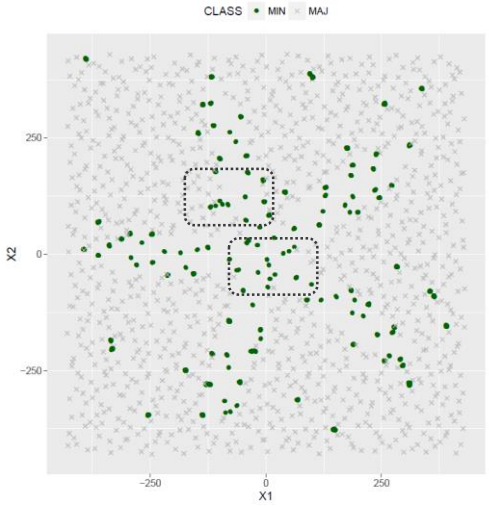
NCR



SM



SP2



Encountered Data Difficulty Factors

Data set	% Safe	% Borderline	% Rare	% Outlier
AP	29	38	8	25
HP	7	28	15	50
SP	4	53	11	32
AE1	2	63	10	25
AE2	14	24	10	52

- Large portion of unsafe (esp. borderline and outlier) examples
- Very small portion of safe examples

Observed sensitivity

AP	None	RU	RO	SM	NCR	SP2
1NN	0.4300	0.7500	0.4300	0.5220	0.5635	0.5005
3NN	0.4385	0.7390	0.6495	0.5365	0.5330	0.6230
C45	0.3680	0.7610	0.5140	0.5005	0.5455	0.5710
PART	0.4375	0.7595	0.5170	0.5255	0.5340	0.5325
NB	0.7160	0.7990	0.7875	0.6770	0.7490	0.8135
RBF	0.5130	0.7860	0.7645	0.6535	0.6685	0.7405
SVM	0.5020	0.7935	0.7880	0.6150	0.5770	0.7640

SP	None	RU	RO	SM	NCR	SP2
1NN	0.2743	0.6307	0.2743	0.3950	0.4743	0.2793
3NN	0.2440	0.6590	0.5553	0.5240	0.4617	0.5513
C45	0.3990	0.6203	0.5523	0.3950	0.4550	0.5883
PART	0.3893	0.6637	0.5487	0.3597	0.4683	0.5760
NB	0.4343	0.7797	0.7203	0.4077	0.5187	0.7220
RBF	0.3913	0.6977	0.4920	0.4070	0.4743	0.5220
SVM	0.3293	0.6597	0.3813	0.3350	0.4163	0.3947

GM – consistent with sensitivity (RU + NB)
Specificity – deteriorated (worst for RU)

HP	None	RU	RO	SM	NCR	SP2
1NN	0.2035	0.6035	0.2035	0.3315	0.3040	0.2035
3NN	0.1205	0.6025	0.4300	0.3630	0.2095	0.4280
C45	0.2690	0.7170	0.4965	0.3865	0.3365	0.4780
PART	0.2875	0.6955	0.5115	0.3585	0.3370	0.4840
NB	0.7535	0.8480	0.8510	0.5645	0.7660	0.8615
RBF	0.5475	0.7920	0.7145	0.4245	0.5865	0.6840
SVM	0.5100	0.7210	0.4985	0.4445	0.5340	0.4970

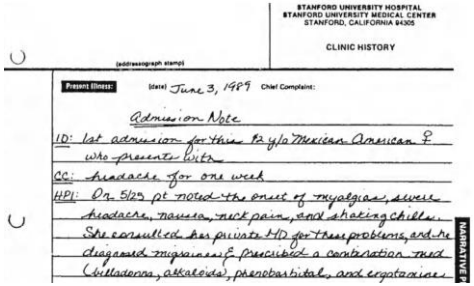
AE1	None	RU	RO	SM	NCR	SP2
1NN	0.2743	0.5903	0.2743	0.4570	0.3957	0.2760
3NN	0.1623	0.6327	0.5097	0.5277	0.3163	0.4860
C45	0.1847	0.6080	0.3910	0.2913	0.3097	0.3617
PART	0.2553	0.6330	0.3723	0.2823	0.3497	0.3953
NB	0.4897	0.7143	0.6833	0.4680	0.5803	0.7167
RBF	0.4343	0.6940	0.6683	0.4763	0.5203	0.7080
SVM	0.3217	0.6170	0.3147	0.3583	0.4080	0.3720

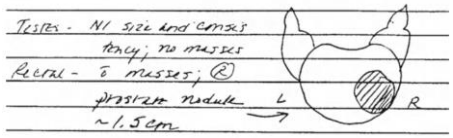
AE2	None	RU	RO	SM	NCR	SP2
1NN	0.1000	0.5867	0.1000	0.3217	0.1317	0.1000
3NN	0.0900	0.7133	0.4200	0.4417	0.1500	0.3750
C45	0.1733	0.6733	0.3933	0.1500	0.2683	0.3300
PART	0.2617	0.6767	0.3767	0.2817	0.3483	0.3400
NB	0.7117	0.7967	0.7267	0.2400	0.7467	0.7533
RBF	0.5317	0.7917	0.7367	0.2500	0.6800	0.7533
SVM	0.4117	0.5950	0.3200	0.3433	0.3533	0.2900

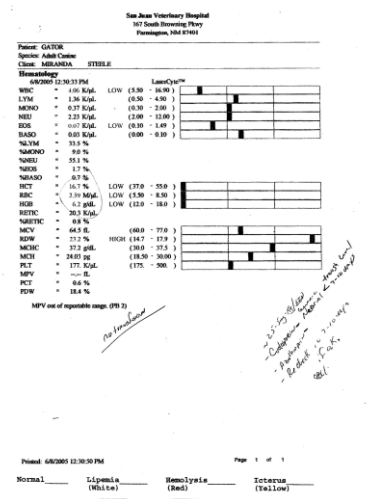
DATA FUSION

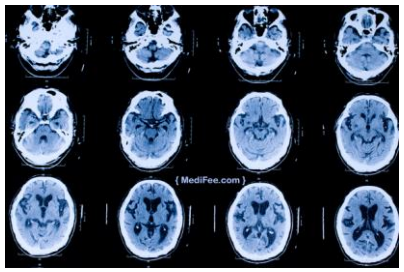
Heterogeneity of clinical data

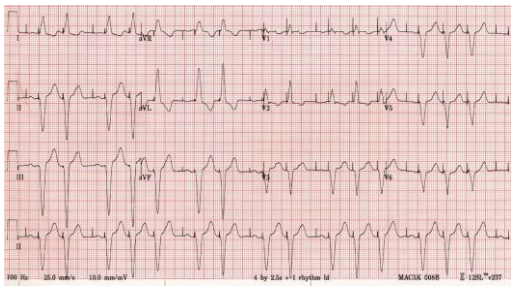
- Text data – “free text” with unformal codes and expressions
- Numerical data
- Omics data (various representations)
- Drawings – hand-made sketches, markings on diagrams (dentistry)
- Signals (numerical time series)
- Images and videos











```
3658435    145    CHROMOSOME_I    1    0    100M    CHROMOSOME_II    2716898    0
GCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCT
AAGCCT
@CCC?:CCCCC@CCCEC>AFDFDBEGHEAHCIGIHHGIGEGJGGIIIHFHIHGF@HGGIGJJJJJIJJJJJJJJJJJJJJJJJJJJHHHHHFF
FFFCCC    RG:Z:1    NH:i:1    NM:i:0

5482659    65    CHROMOSOME_I    1    0    100M    CHROMOSOME_II    11954696    0
GCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCT
AAGCCT
CCCFFFFFHHHGHGJJGIHIJIJJJJJIJJJJJIJJGIJJJJJIIJIIJFJJJJJFIJJJJIIIIGIIJHHHHDEEFFFEEEEEDDDDCDCCCA
AA?CC:    RG:Z:1    NH:i:1    NM:i:0
```

Problem statement

Focus on a single data modality may be insufficient to construct a comprehensive and accurate clinical decision model

- Most of the developed clinical decision model rely on a single data modality (e.g., “traditional” data or image data)
- **Data fusion** may be used address the above limitation

Data fusion

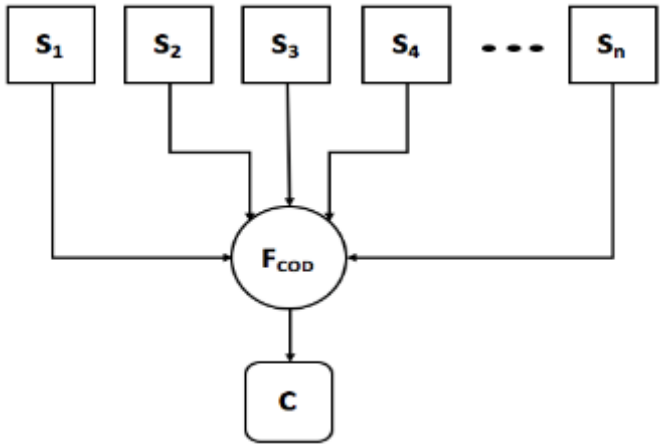
Integration of data and knowledge from multiple sources of diversified format and structure

- Human perception system → extended angular vision is obtained by the combination of percepts from each eye
- Human brain → fusion on information collected through all the senses and previous memory to generate orderly action
- Other application areas – multi-sensor networks, surveillance systems, imaging studies

Data fusion techniques

Combination of data (COD)

- Aggregation of data from various sources into a single space
- Construction of a decision model using aggregated space

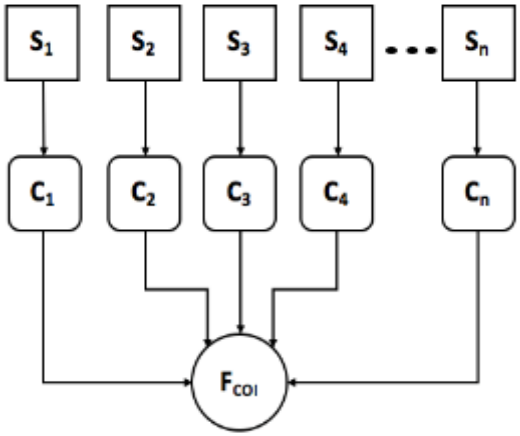


- Drawback: course of dimensionality

Data fusion techniques

Combination of interpretations (COI)

- Construction of decision models from each data source
- Combination of outcomes of obtained models by a combiner to produce a single decision (→ stacking)

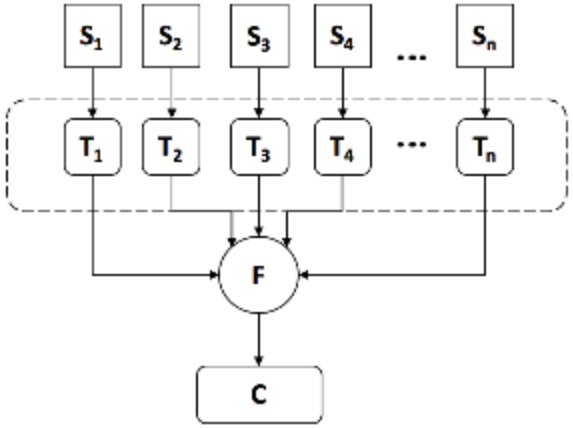


- Drawback: Inability to handle inter-source dependencies

Data fusion techniques

General fusion framework (GFF)

- Brining data into a homogeneous space through a series of simple and complex transformations
  - Simple: data pre-processing (feature selection, transformation)
  - Complex: construction of "intermediary" classifiers
- Construction of the final classifier from the homogeneous space



- Drawback: selecting transformations and their sequence

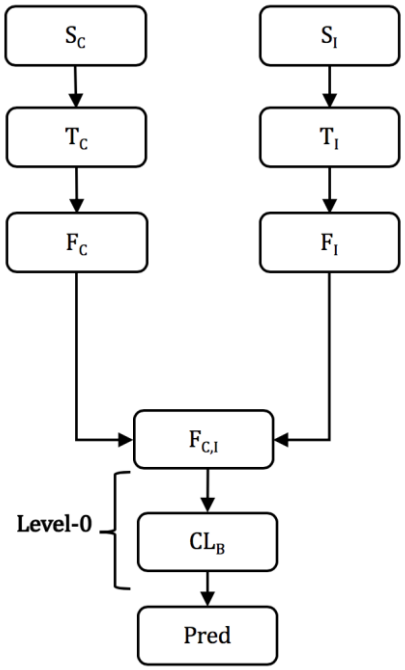
G. Lee, A. Madabhushi: A knowledge representation framework for integration, classification of multi-scale imaging and non-imaging data: Preliminary results in predicting prostate cancer recurrence by fusing mass spectrometry and histology. *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2009.

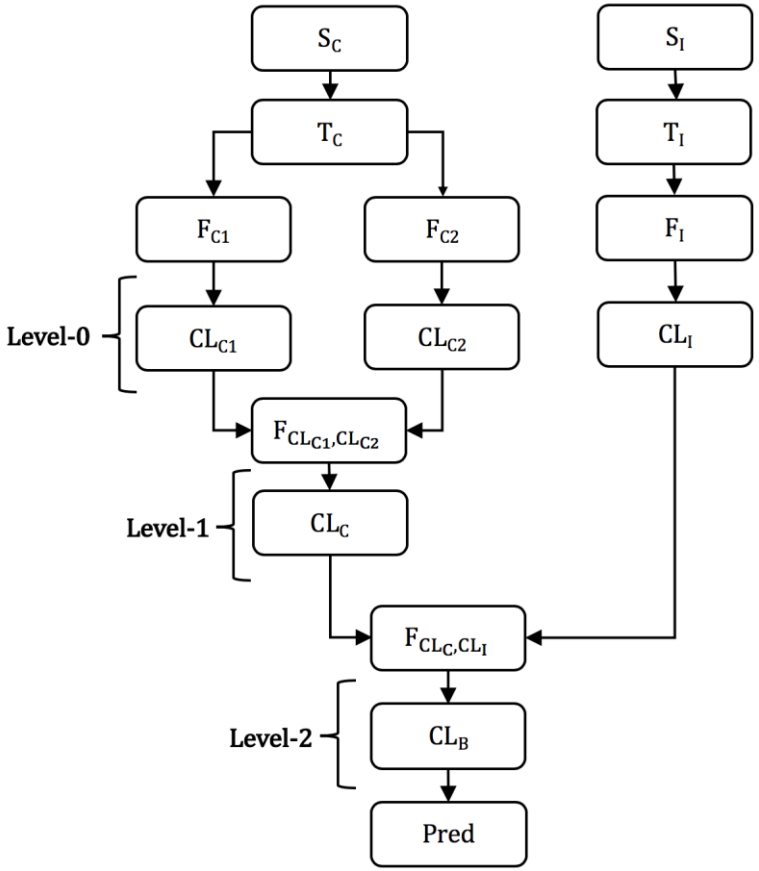
Predicting treatment for fractures

- Prediction of the type of treatment in patients with fractures – surgical vs. non-surgical
- Non-image data (demographics, results of examinations and lab tests) and image data (X-ray)
- 210 patients extracted from a repository of educational cases hosted by the WCT telemedical platform
- Comparison of COD and COI approaches (of varying complexity)

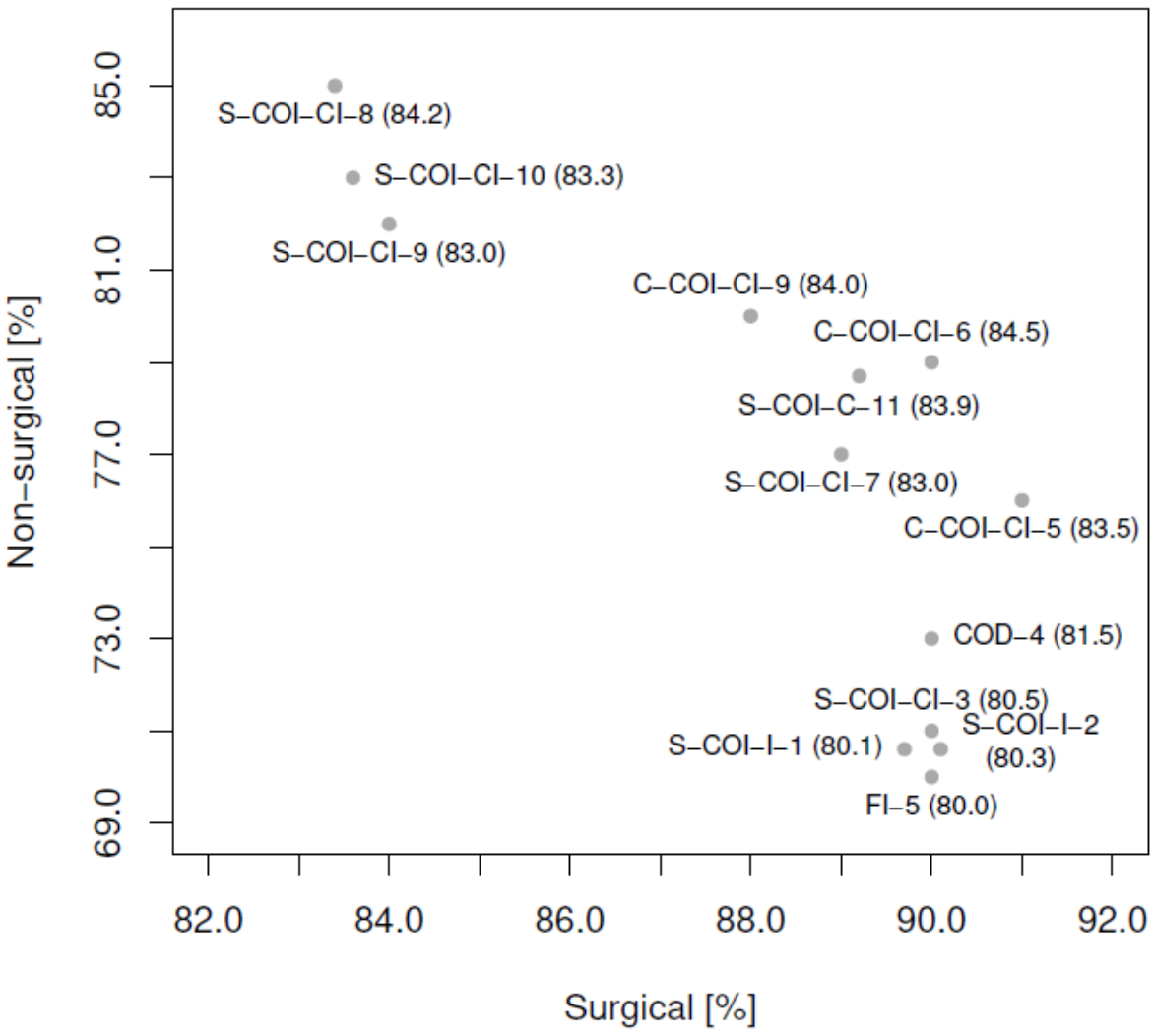
A. Haq, Sz. Wilk, A. Abelló: Fusion of clinical data: A case study to predict the type of treatment of bone fractures. AMC 29 (1), 2019.

Example fusion models





Results


S–COI–CI–8 (84.2)
S–COI–CI–10 (83.3)
S–COI–CI–9 (83.0)
C–COI–CI–9 (84.0)
C–COI–CI–6 (84.5)
S–COI–C–11 (83.9)
S–COI–CI–7 (83.0)
C–COI–CI–5 (83.5)
COD–4 (81.5)
S–COI–CI–3 (80.5)
S–COI–I–2
(80.3)
S–COI–I–1 (80.1)
FI–5 (80.0)
85.0
81.0
77.0
73.0
69.0
Non–surgical [%]
82.0
84.0
86.0
88.0
90.0
92.0
Surgical [%]

Application of deep learning

npj | Digital Medicine

www.nature.com/npjdigitalmed

ARTICLE OPEN

Scalable and accurate deep learning with electronic health records

Alvin Rajkomar[1,2], Eyal Oren[1], Kai Chen[1], Andrew M. Dai[1], Nissan Hajaj[1], Michaela Hardt[1], Peter J. Liu[1], Xiaobing Liu[1], Jake Marcus[1], Mimi Sun[1], Patrik Sundberg[1], Hector Yee[1], Kun Zhang[1], Yi Zhang[1], Gerardo Flores[1], Gavin E. Duggan[1], Jamie Irv… Kurt Litsch[1], Alexander Mossin[1], Justin Tansuwan[1], De Wang[1], James Wexler[1], Jimbo Wilson[1], Dana Ludwig[2], Samu… Katherine Chou[1], Michael Pearson[1], Srinivasan Madabushi[1], Nigam H. Shah[4], Atul J. Butte[2], Michael D. Howell[1], Greg S. Corrado[1] and Jeffrey Dean[1]

Predictive modeling with electronic health record (EHR) data is anticipated to drive personalized medicine and in… quality. Constructing predictive statistical models typically requires extraction of curated predictor variables from … data, a labor-intensive process that discards the vast majority of information in each patient's record. We propose a represe… of patients' entire raw EHR records based on the Fast Healthcare Interoperability Resources (FHIR) format. We demonstra… deep learning methods using this representation are capable of accurately predicting multiple medical events from multiple centers without site-specific data harmonization. We validated our approach using de-identified EHR data from two US academic medical centers with 216,221 adult patients hospitalized for at least 24 h. In the sequential format we propose, this volume of EHR data unrolled into a total of 46,864,534,945 data points, including clinical notes. Deep learning models achieved high accuracy for tasks such as predicting: in-hospital mortality (area under the receiver operator curve [AUROC] across sites 0.93–0.94), 30-day unplanned readmission (AUROC 0.75–0.76), prolonged length of stay (AUROC 0.85–0.86), and all of a patient's final discharge diagnoses (frequency-weighted AUROC 0.90). These models outperformed traditional, clinically-used predictive models in all cases. We believe that this approach can be used to create accurate and scalable predictions for a variety of clinical scenarios. In a case study of a particular prediction, we demonstrate that neural networks can be used to identify relevant information from the patient's chart.

npj Digital Medicine (2018)1:18 ; doi:10.1038/s41746-018-0029-1

216,221 patients,
46,864,534,945 data points
(tokens)

Application of deep learning

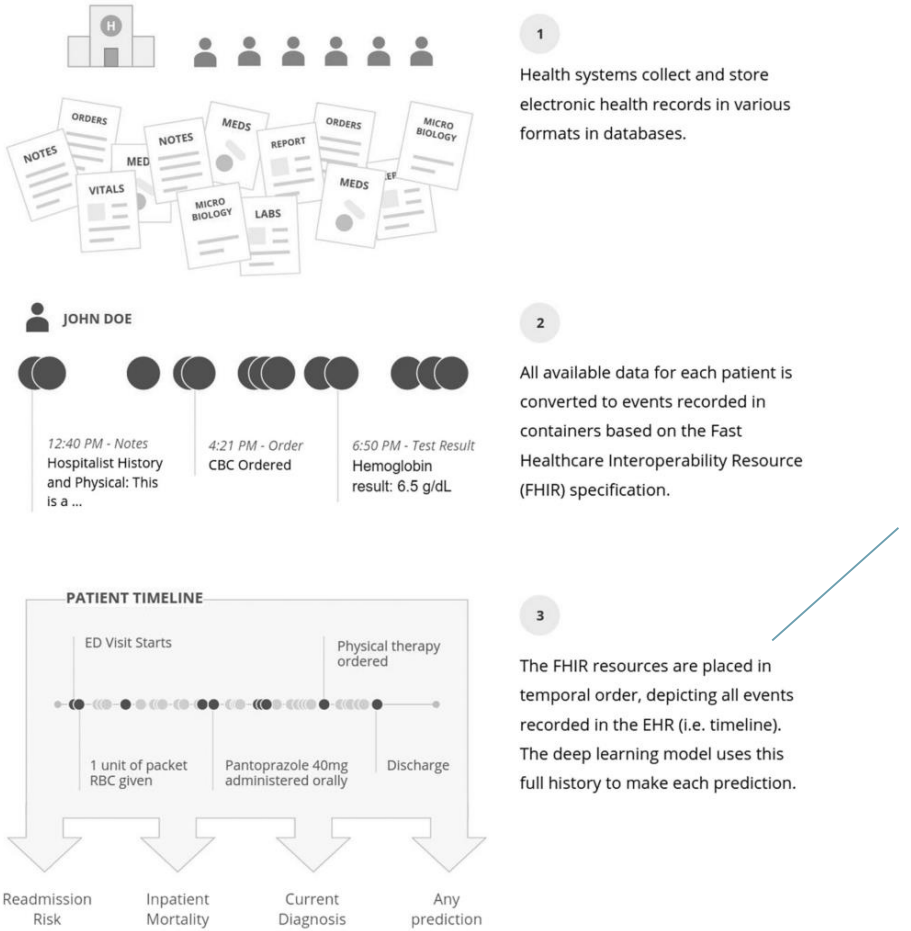


Fig. 4 Data from each health system were mapped to an appropriate FHIR (Fast Healthcare Interoperability Resources) resource and placed in temporal order. This conversion did not harmonize or standardize the data from each health system other than map them to the appropriate resource. The deep learning model could use all data available prior to the point when the prediction was made. Therefore, each prediction, regardless of the task, used the same data

An ensemble of 3 "time-aware" deep neural networks

Rajkomar, A., Oren, E., Chen, K., Dai, A. M., Hajaj, N., Liu, P. J., … Dean, J. (2018). Scalable and accurate deep learning for electronic health records. *Npj Digital Medicine*, (March), 1–10.

Integration of textual and non-textual data

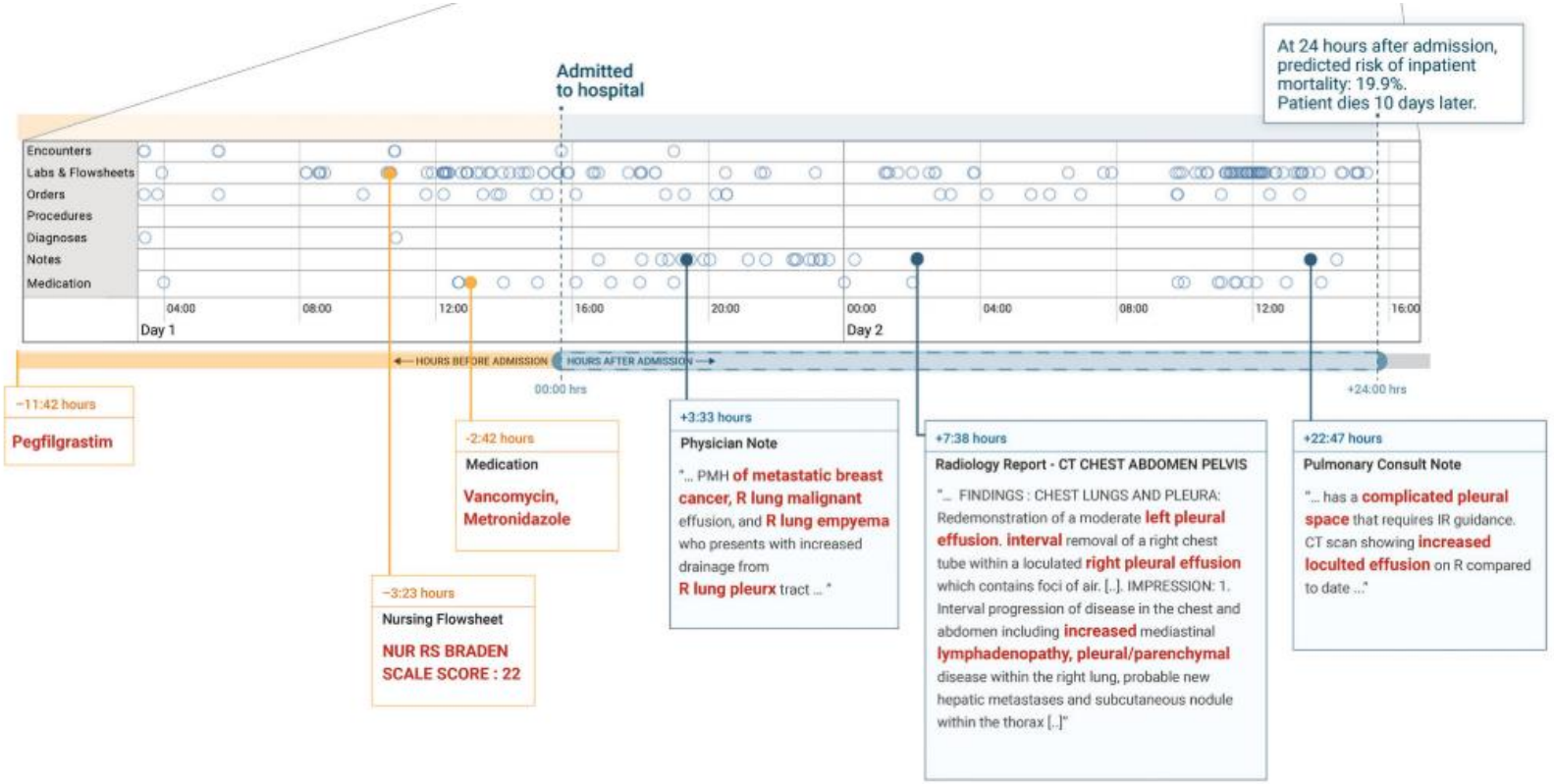


Fig. 3 The patient record shows a woman with metastatic breast cancer with malignant pleural effusions and empyema. The patient timeline at the top of the figure contains circles for every time-step for which at least a single token exists for the patient, and the horizontal lines show the data type. There is a close-up view of the most recent data points immediately preceding a prediction made 24 h after admission. We trained models for each data type and highlighted in red the tokens which the models attended to—the non-highlighted text was not attended to but is shown for context. The models pick up features in the medications, nursing flowsheets, and clinical notes relevant to the prediction