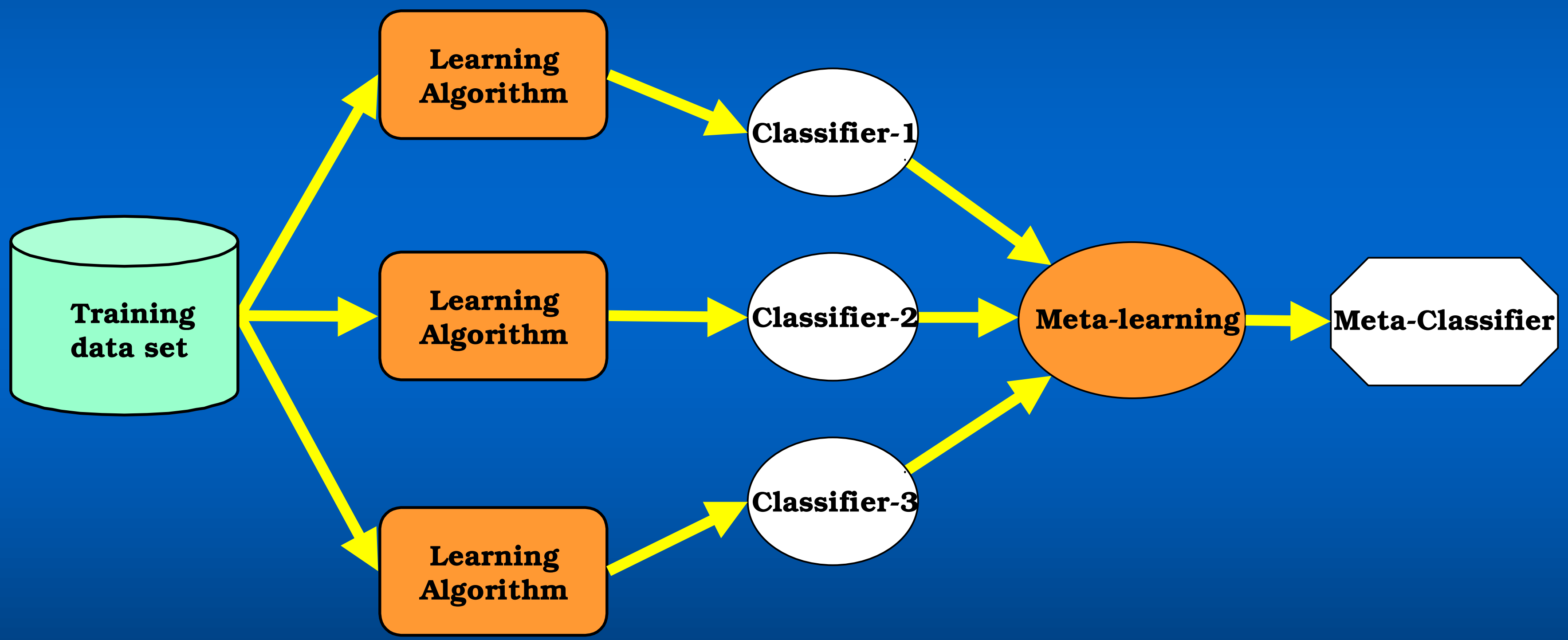
Minimal Cost Complexity Pruning of Meta-Classifiers

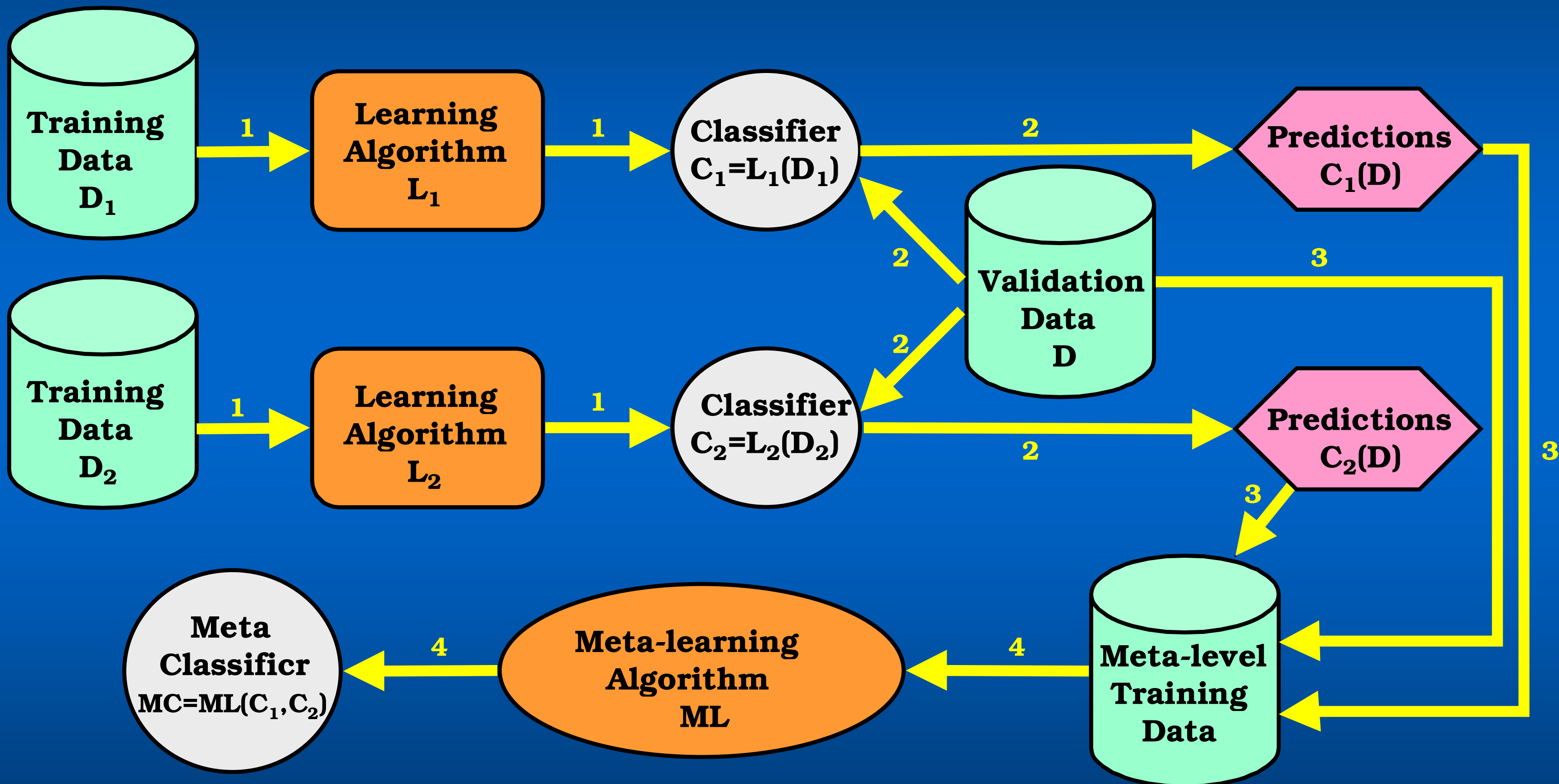
Andreas L. Prodromidis
Salvatore J. Stolfo

Department of Computer Science
Columbia University

Combining multiple models



Meta-learning



A meta-learning training set example

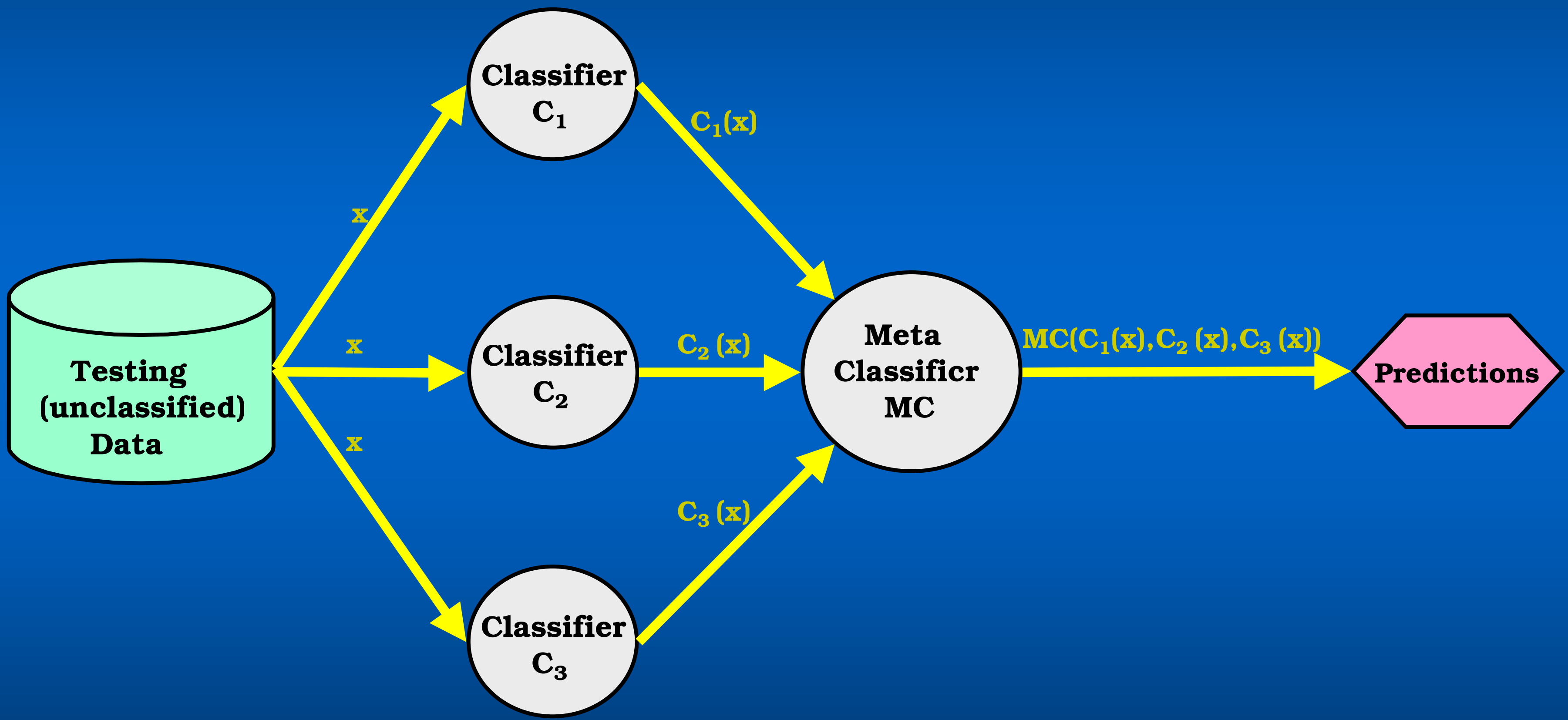
Validation set

CID	*InputType*	*Amt*	*…*	*True Class*
54341	Swipe	19.72	…	Legitimate
54432	KeyIn	88.19	…	Fraudulent
54101	Phone	11.99	…	Legitimate
…	…	…	…	…

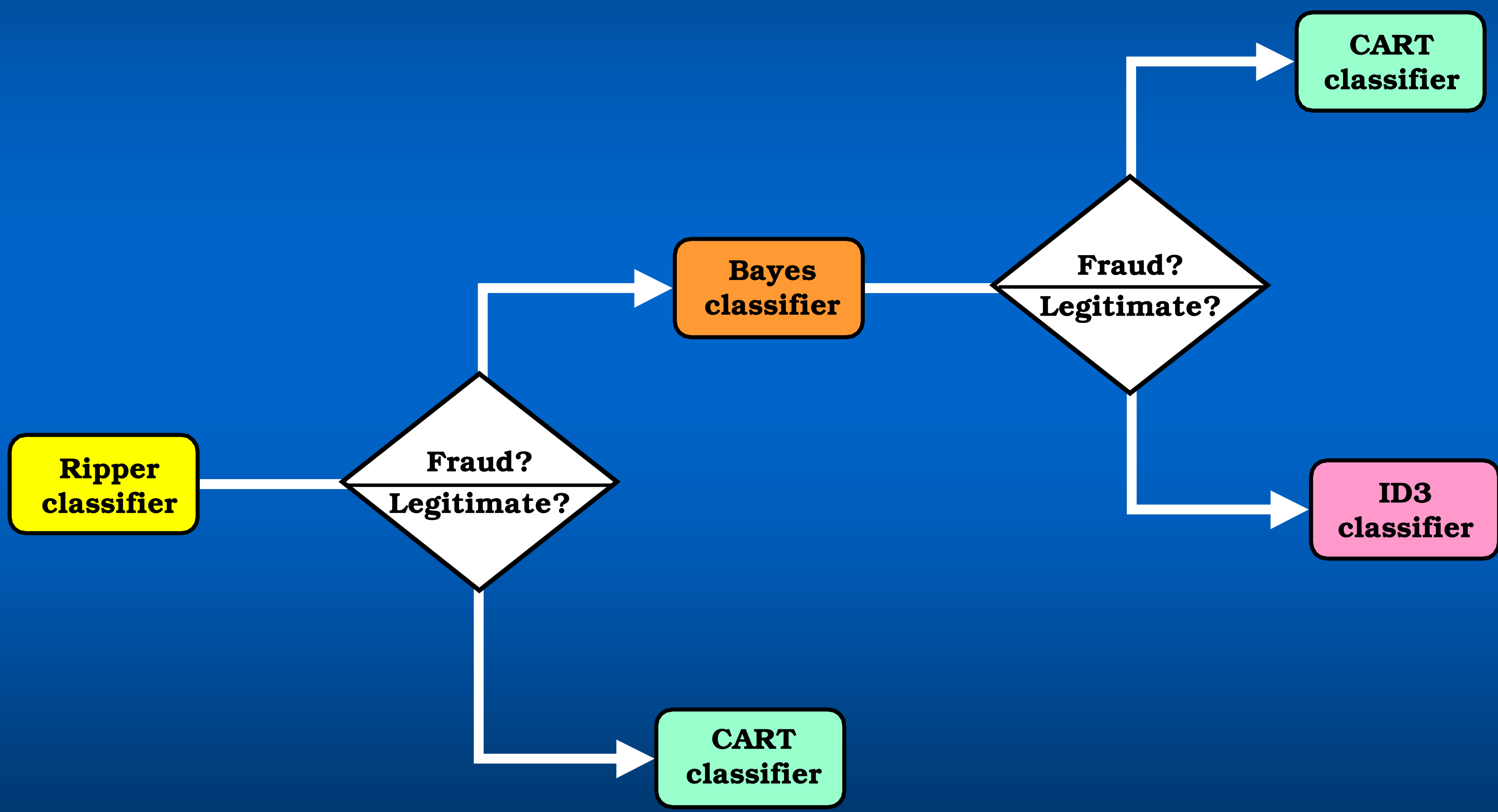
Meta-level training set - Stacking (Wolpert-92)

Classifier-1	*Classifier-2*	*Classifier-3*	*…*	*True Class*
Legitimate	Legitimate	Legitimate	…	Legitimate
Legitimate	Fraudulent	Legitimate	…	Fraudulent
Fraudulent	Fraudulent	Legitimate	…	Legitimate
…	…	…	…	…

Meta-Classifying



Decision tree meta-classifier



Efficiency

- Compute base classifiers in parallel
- Compute “small” meta-classifiers
  - to reduce memory requirements
  - to produce fast classifications
- Pre-training pruning
  - filter before meta-learning (NIT’98, KDD’98-DDM)
- Post-training pruning
  - discard after meta-learning (Prodromidis-et-al-98)

A graphical description

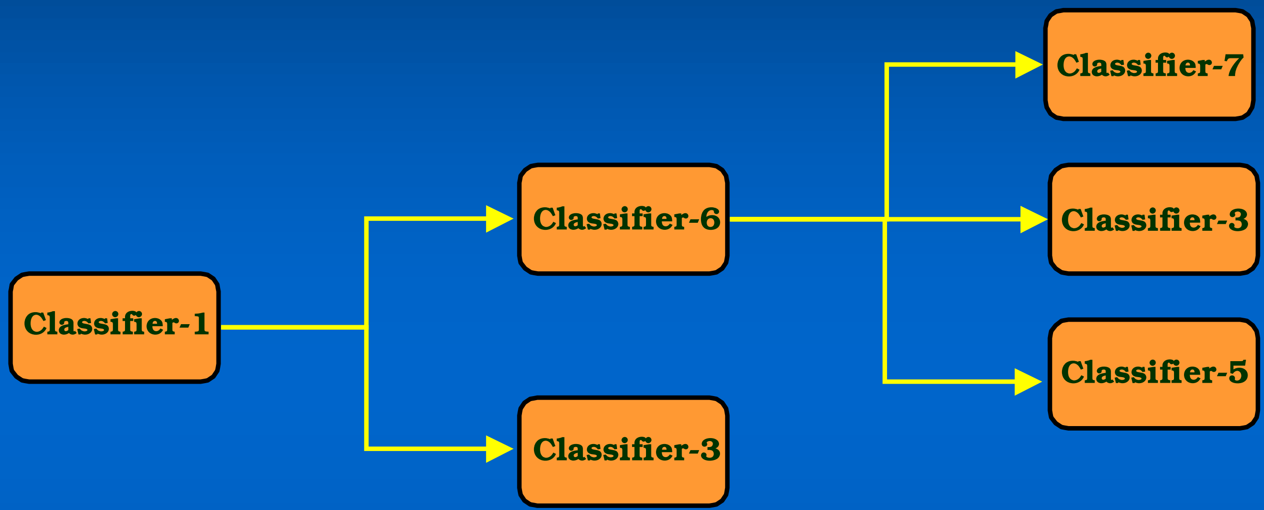
Map arbitrary meta-classifier to a decision tree representation

Prune the decision tree model

Map pruned decision tree to original meta-classifier representation

(Mapping via modeling of the meta-classifier's behavior)

Post-training pruning



- Minimal cost complexity pruning (Breiman-et-al-84)
  - R(T): misclassification cost of a decision tree T
  - C(T): complexity of tree (= number of terminal nodes)
  - α: complexity parameter
- Seek to minimize $R_\alpha(T)$, $R_\alpha(T) = R(T) + \alpha \cdot C(T)$

Decision tree model (unpruned)

Complexity=0.5

Complexity=0.92

Complexity=1.7

Complexity=2.8

Complexity=3.52

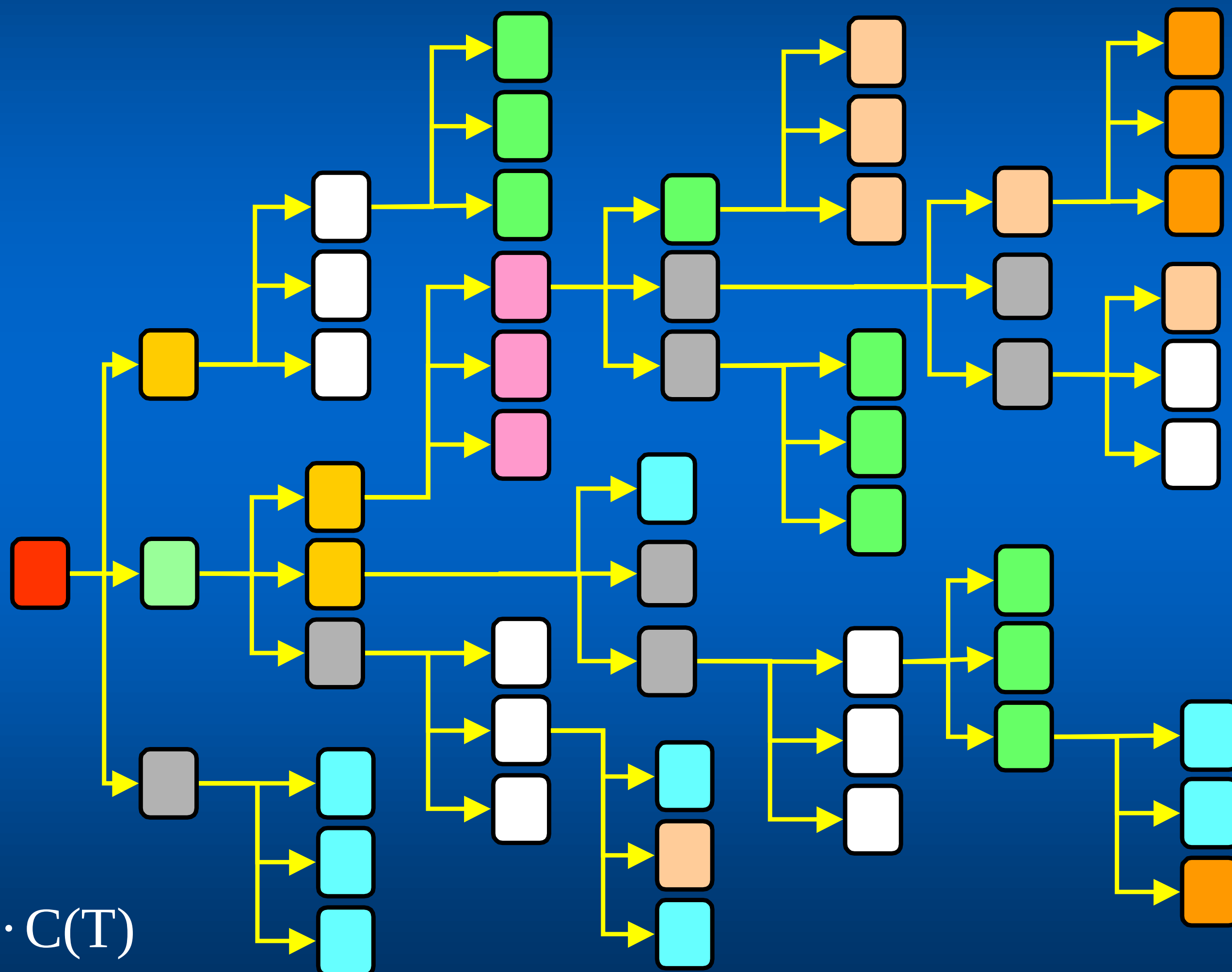
Complexity=3.61

Complexity=3.99

Complexity=5.0

Complexity=7.84

Complexity=10.5



$$R_\alpha(T) = R(T) + \alpha \cdot C(T)$$

Decision tree model (pruned)

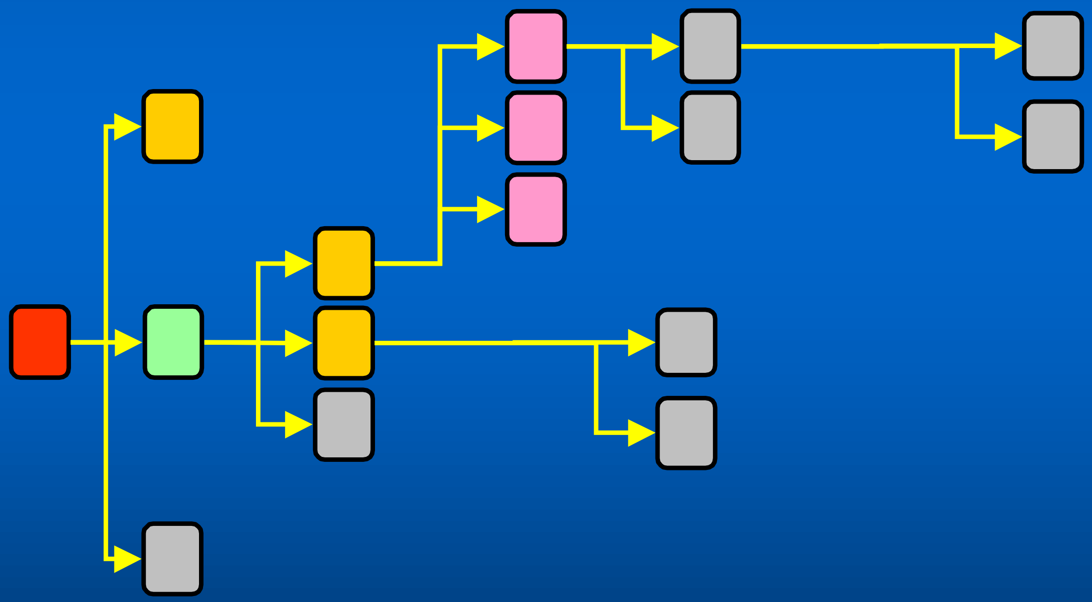
Complexity=3.61

Complexity=3.99

Complexity=5.0

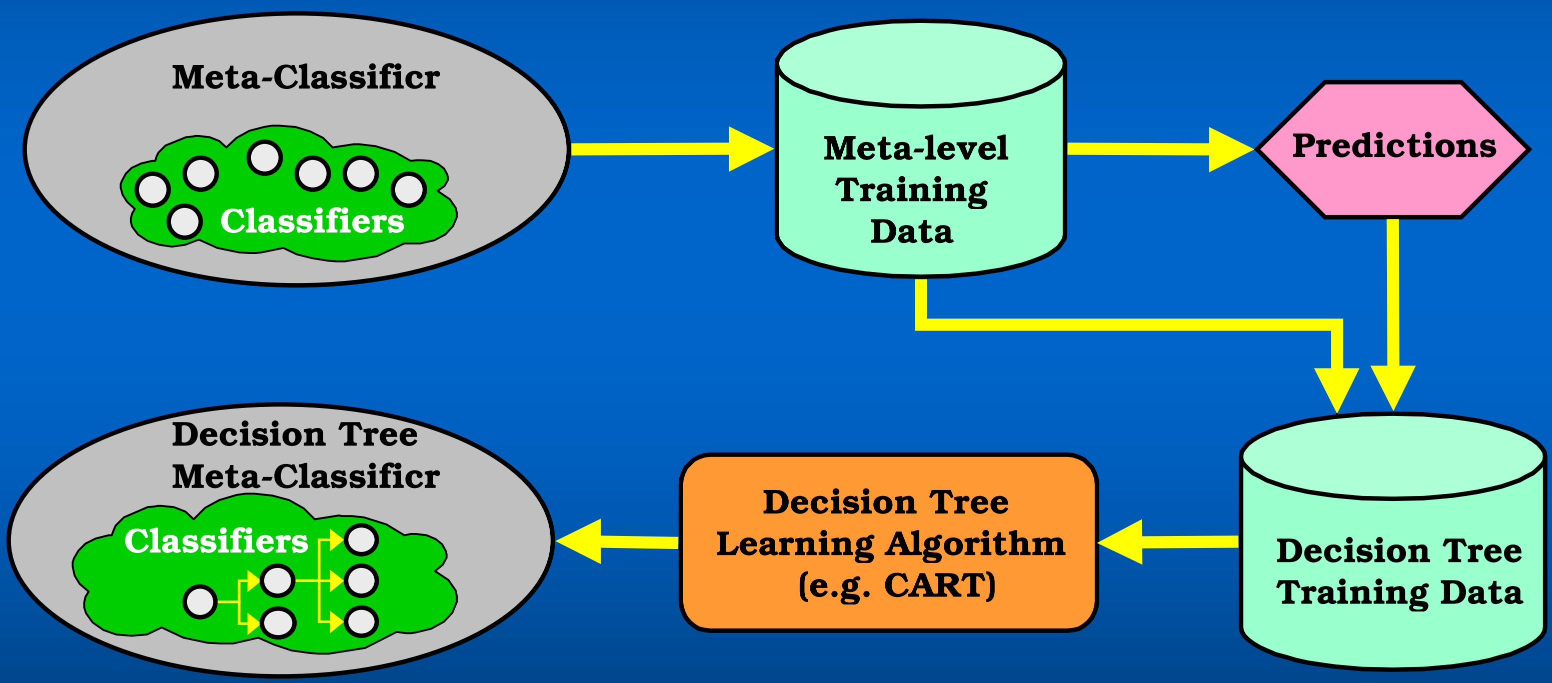
Complexity=7.84

Complexity=10.5

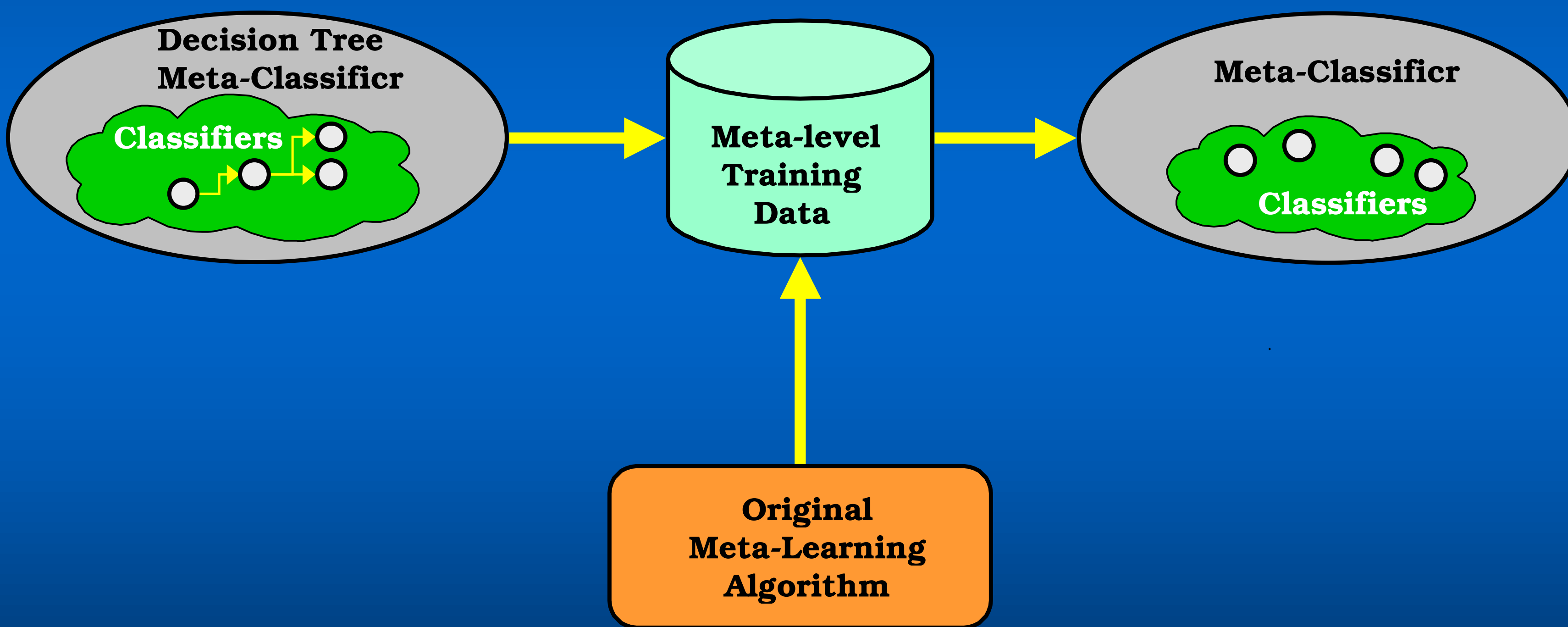


$$R_\alpha(T) = R(T) + \alpha \cdot C(T)$$

Decision tree modeling of meta-classifiers



Final pruned meta-classifier



Credit Card Fraud detection

- Chase Credit Card data
  - 500,000 transaction records
  - 30 attributes (numerical, categorical) in 137 bytes per record
  - 20% fraud, 80% non fraud
- First Union Credit Card data
  - 500,000 transaction records
  - 28 attributes (numerical categorical) in 137 bytes per record
  - 15% fraud, 85% non fraud
- Attributes
  - Hashed credit card account number, date, time, type of entry of transaction, type of merchant, amount, validity codes, past payment information, account information, confidential fields, etc.
  - The fraud label

Experimental setting

- Divide data sets in 12 subsets at 6 sites
- Five learning algorithms
  - Naïve Bayes, C4.5, CART, ID3, Ripper
- Exchange classifiers
  - 10 local, 50 remote per site
- Meta-learn only the remote classifiers

Meta-learning results

Type of Classification model	*Size*	*Accuracy*	*TP-FP*	*Savings*
Best over a single subset	1	88.5%	0.551	$ 812K
Best over largest possible subset	1	88.8%	0.568	$ 840K
Meta-classifier	50	89.6%	0.621	$ 818K
Chase's COTS system	--	85.7%	0.523	$ 682K

Chase data

Maximum savings:

$1,470K

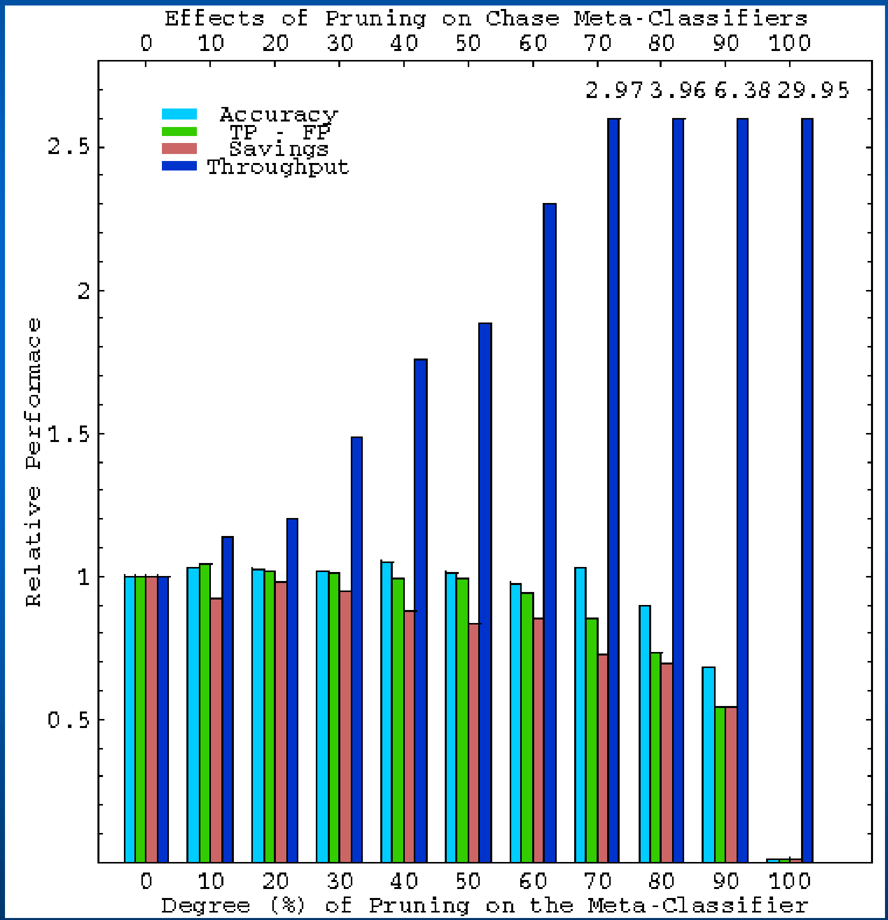
Type of Classification model	*Size*	*Accuracy*	*TP-FP*	*Savings*
Best over a single subset	1	95.2%	0.749	$ 806K
Best over largest possible subset	1	95.3%	0.787	$ 828K
Meta-classifier	50	96.5%	0.831	$ 944K

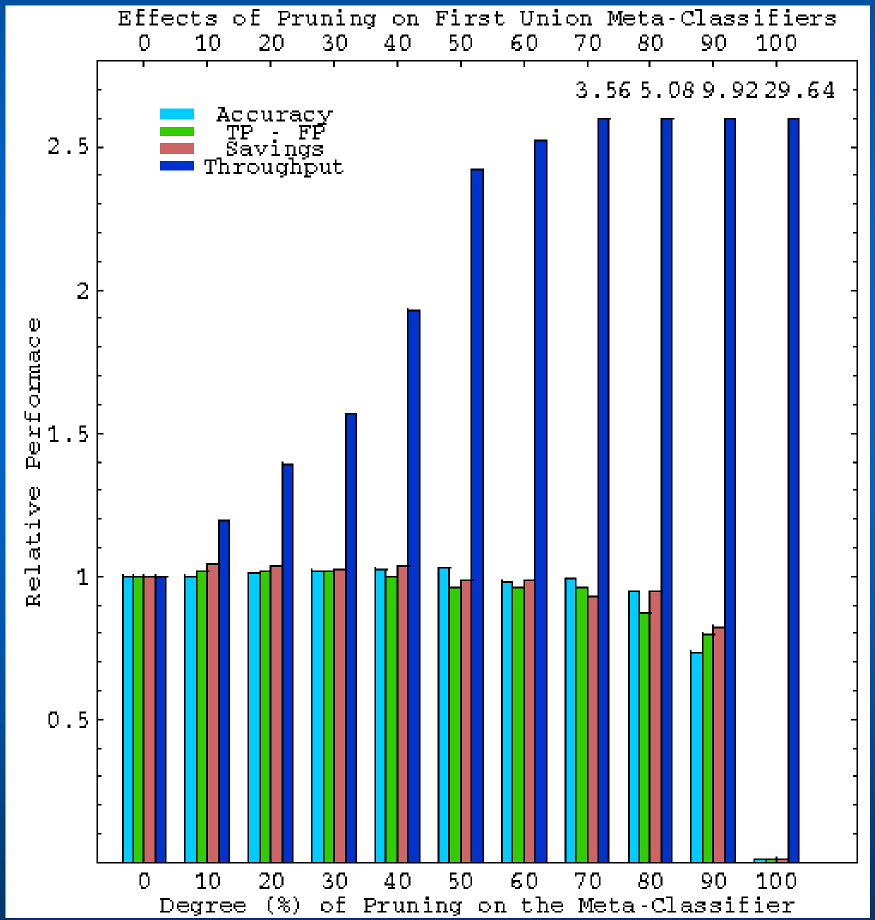
First Union data

Maximum savings:

$1,085K

Pruning results





More information

- About the paper
  - http://www.cs.columbia.edu/~andreas
- About the JAM project
  - http://www.cs.columbia.edu/~sal/JAM/PROJECT
- E-mail contact
  - andreas@cs.columbia.edu