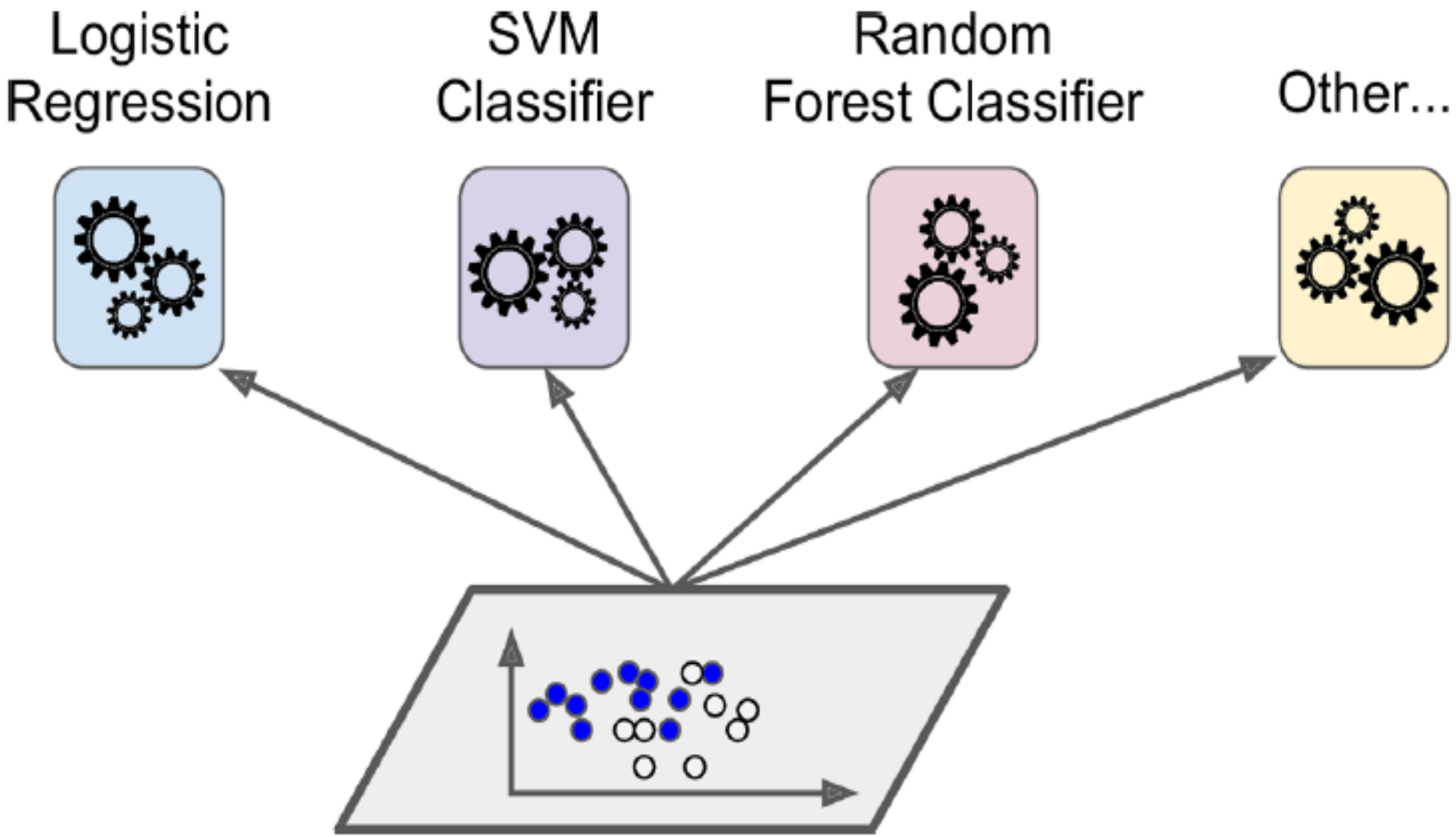
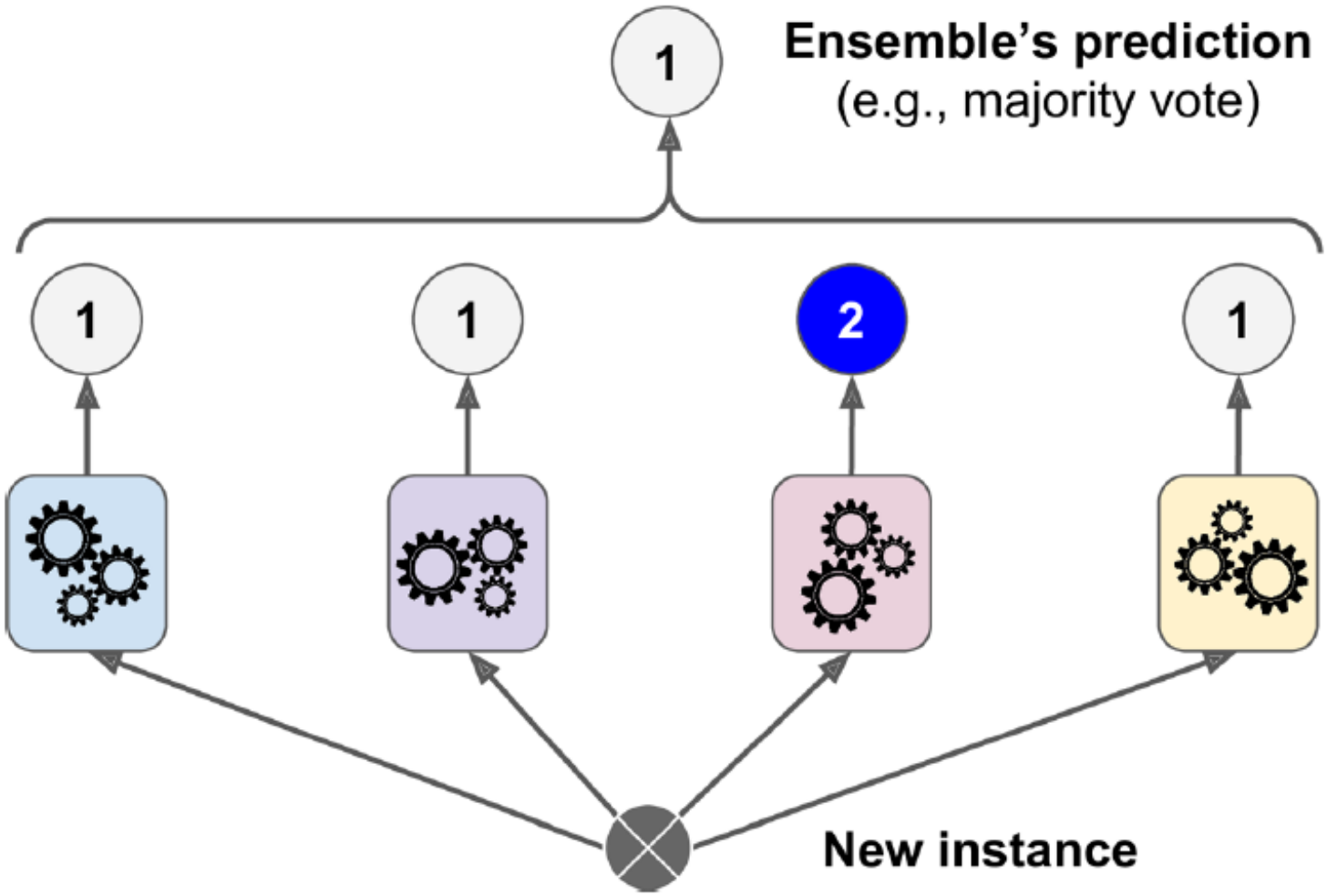
Ensemble learning

Saeed Sharifian

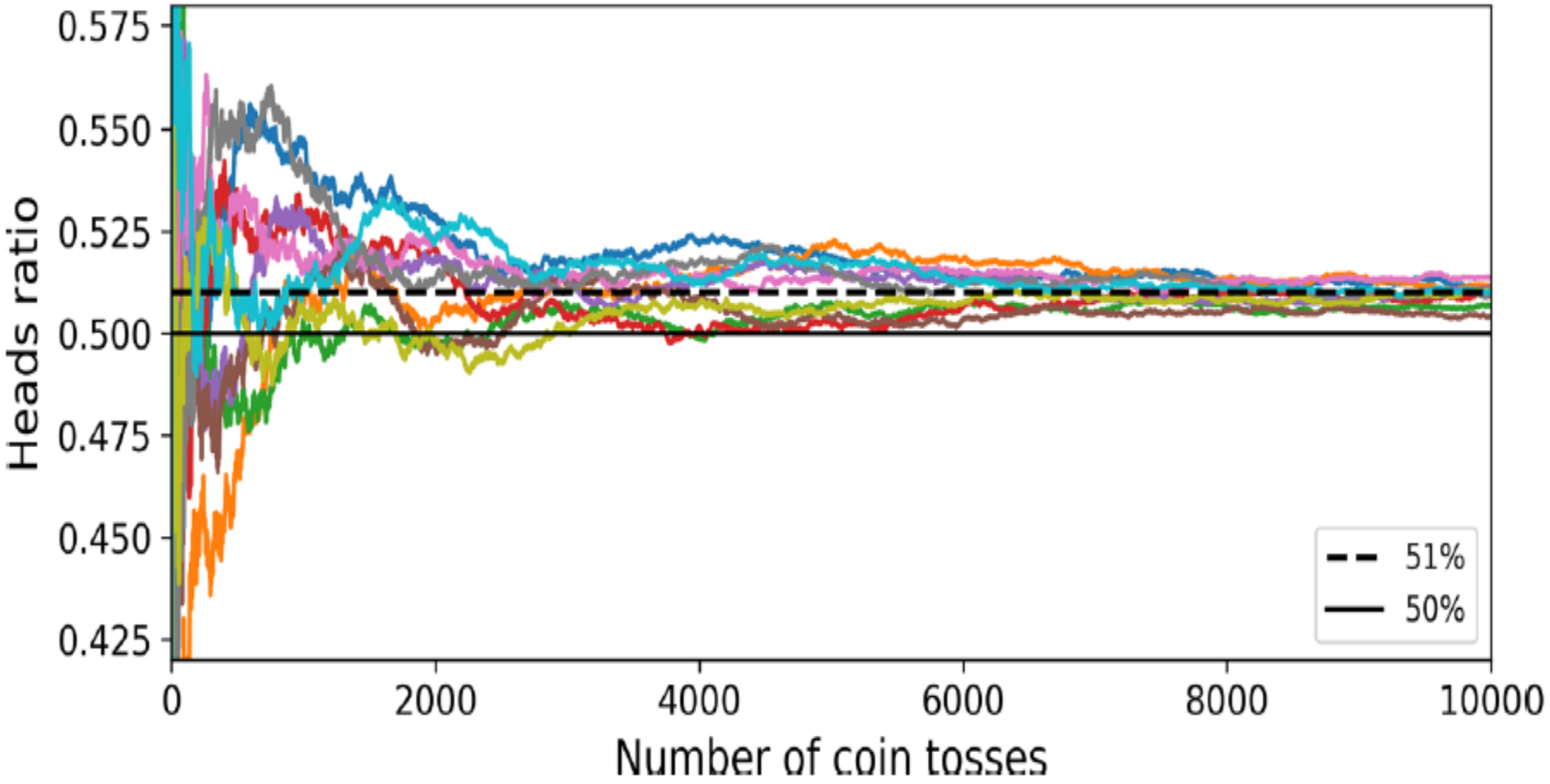
Training diverse classifiers



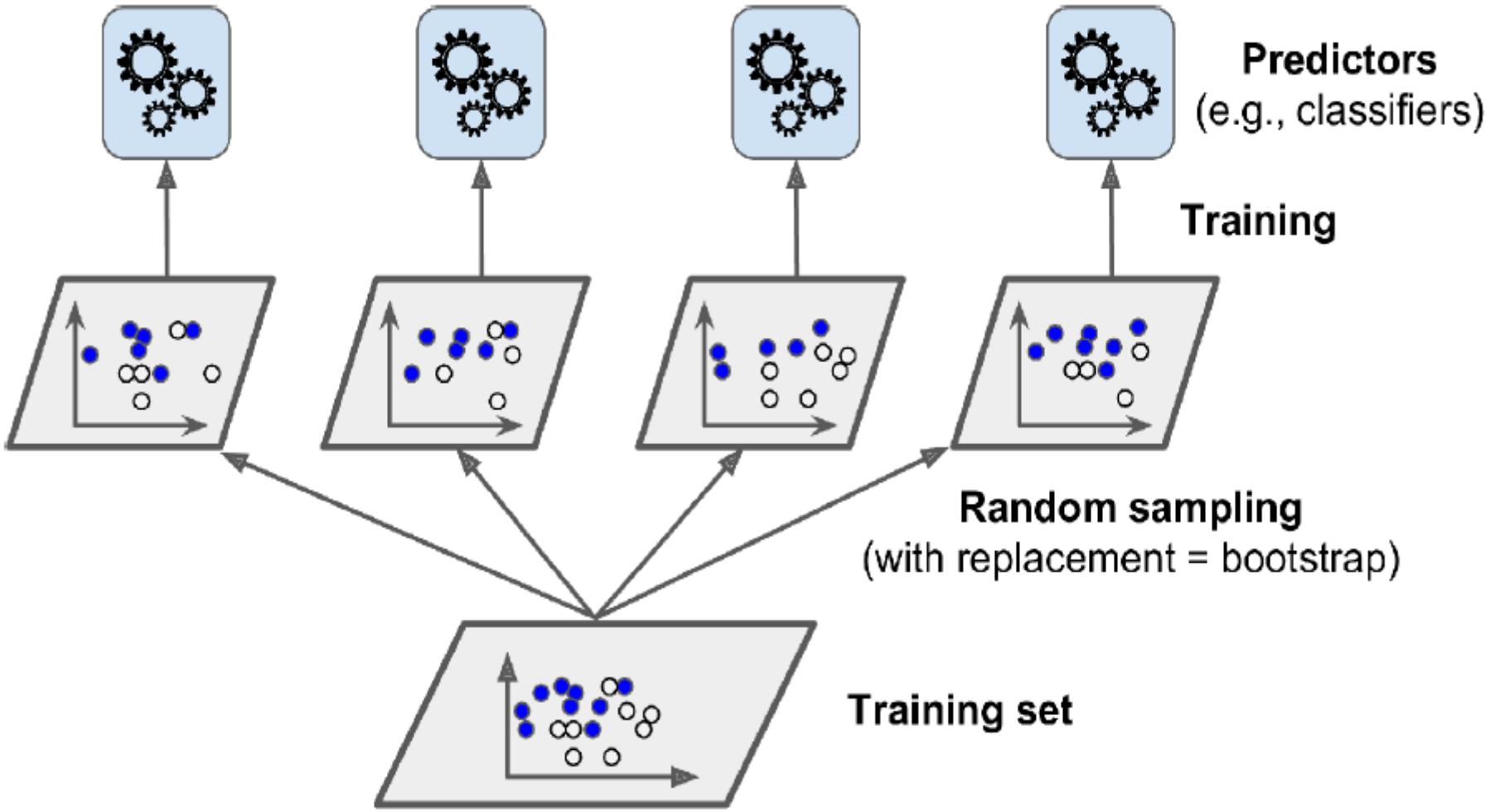
Hard vs Soft voting



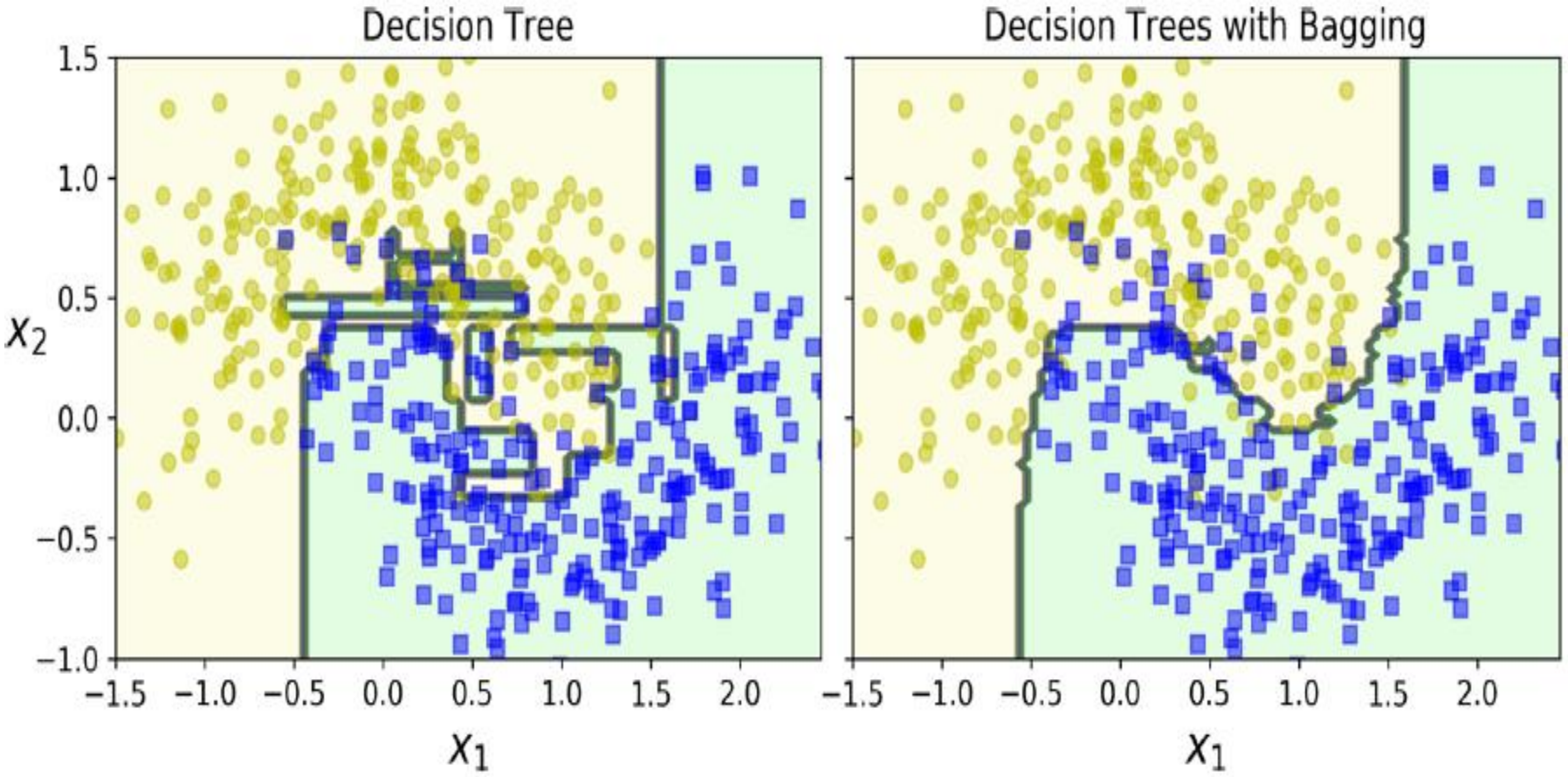
The law of large numbers



Bagging and pasting



Result of 500 Trees



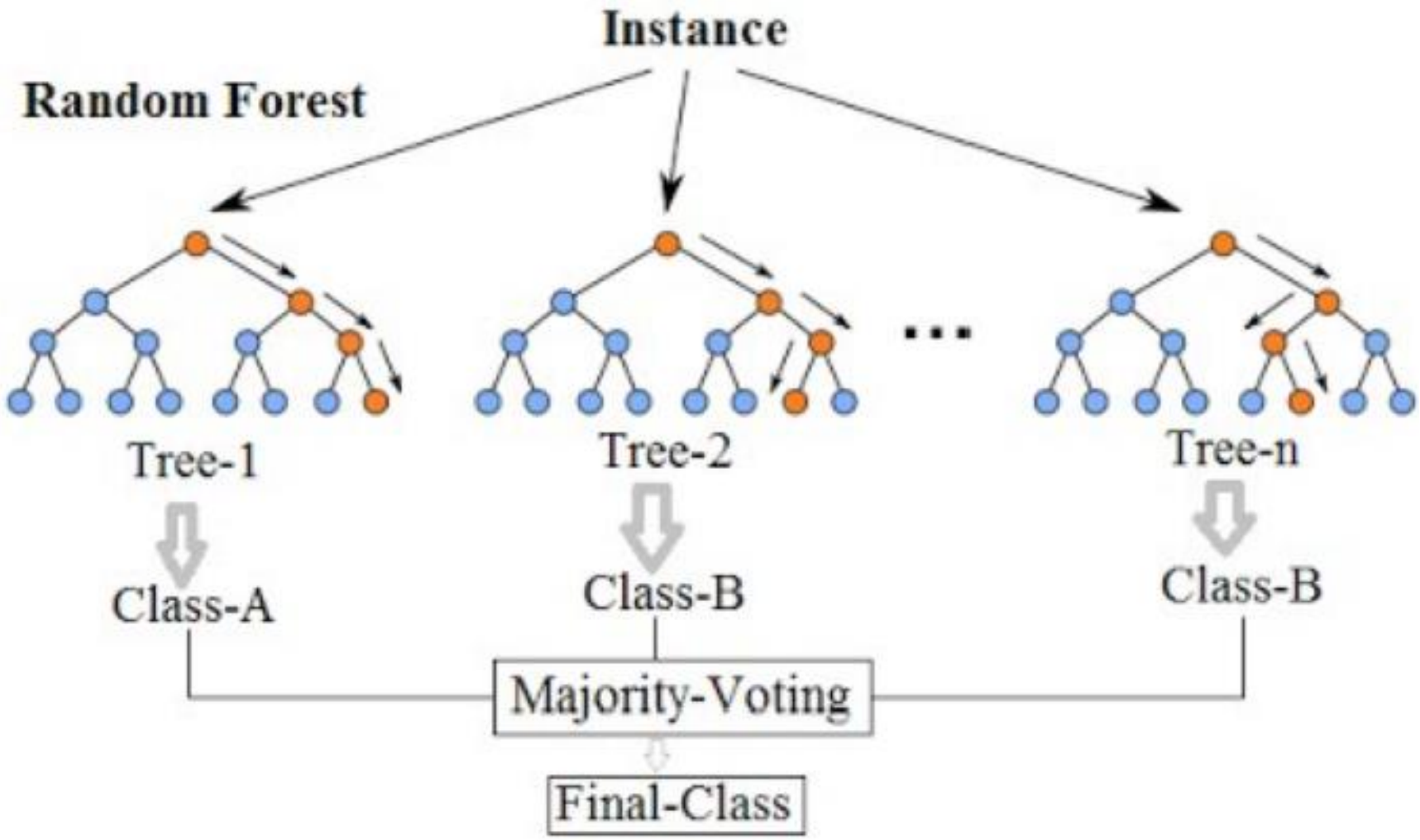
Out-of-Bag Evaluation

- ✓ Only about 63% of the training instances are sampled on average for each predictor
- ✓ The remaining 37% of the training instances that are not sampled are called out-of-bag (oob) instances
- ✓ Note that they are not the same 37% for all predictors.

Random Patches and Subspaces

- ✓ Sampling both training instances and features is called the Random Patches method
- ✓ Keeping all training instances but sampling features is called the Random Subspaces method
- ✓ Sampling features results in even more predictor diversity, trading a bit more bias for a lower variance.

Random Forests



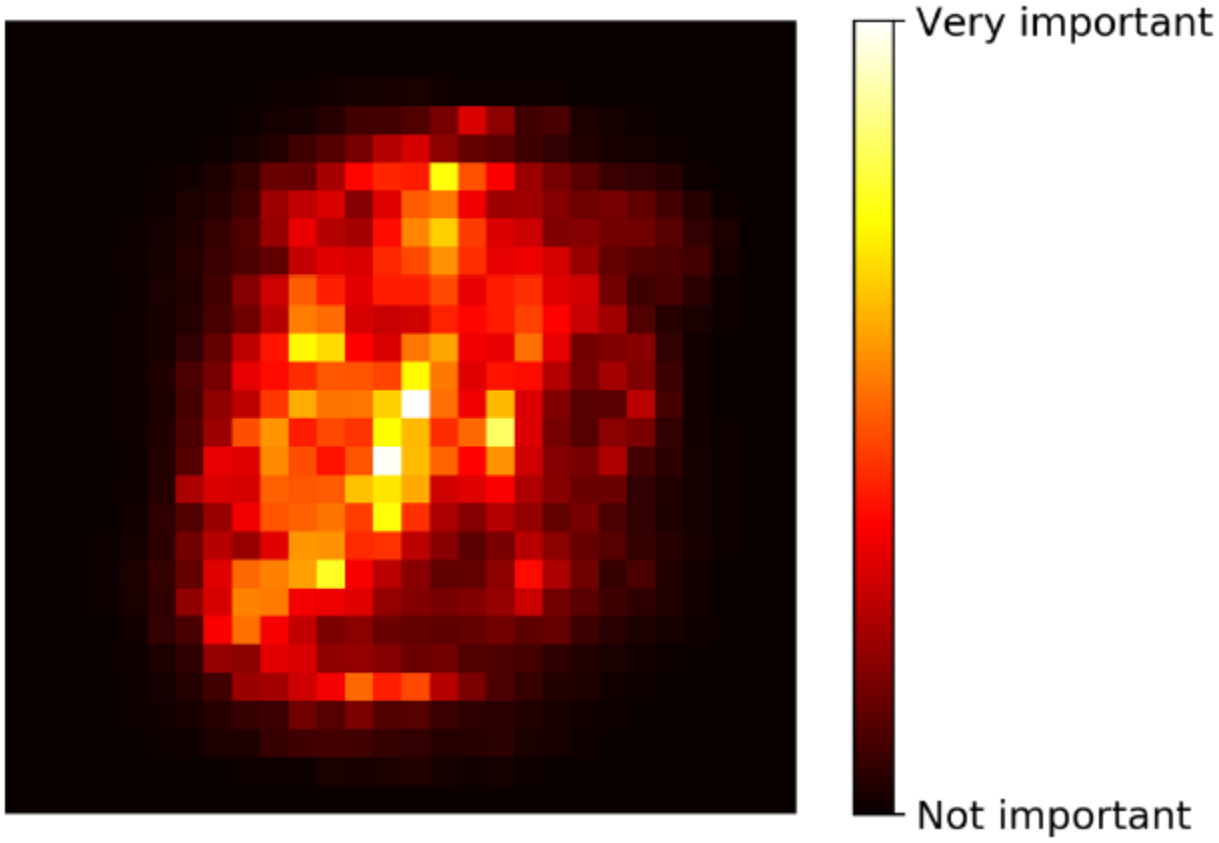
Random Forests

- ✓ Random Forest is an ensemble of Decision Trees, generally trained via the bagging method
- ✓ searches for the best feature among a random subset of features
- ✓ Extra-Trees: using random thresholds for each feature rather than searching for the best possible thresholds

Relative importance of each feature

- ✓ How much the tree nodes that use that feature reduce impurity on average

MNIST pixel importance



Hypothesis Boosting

- ✓ Any Ensemble method that can combine several weak learners into a strong learner

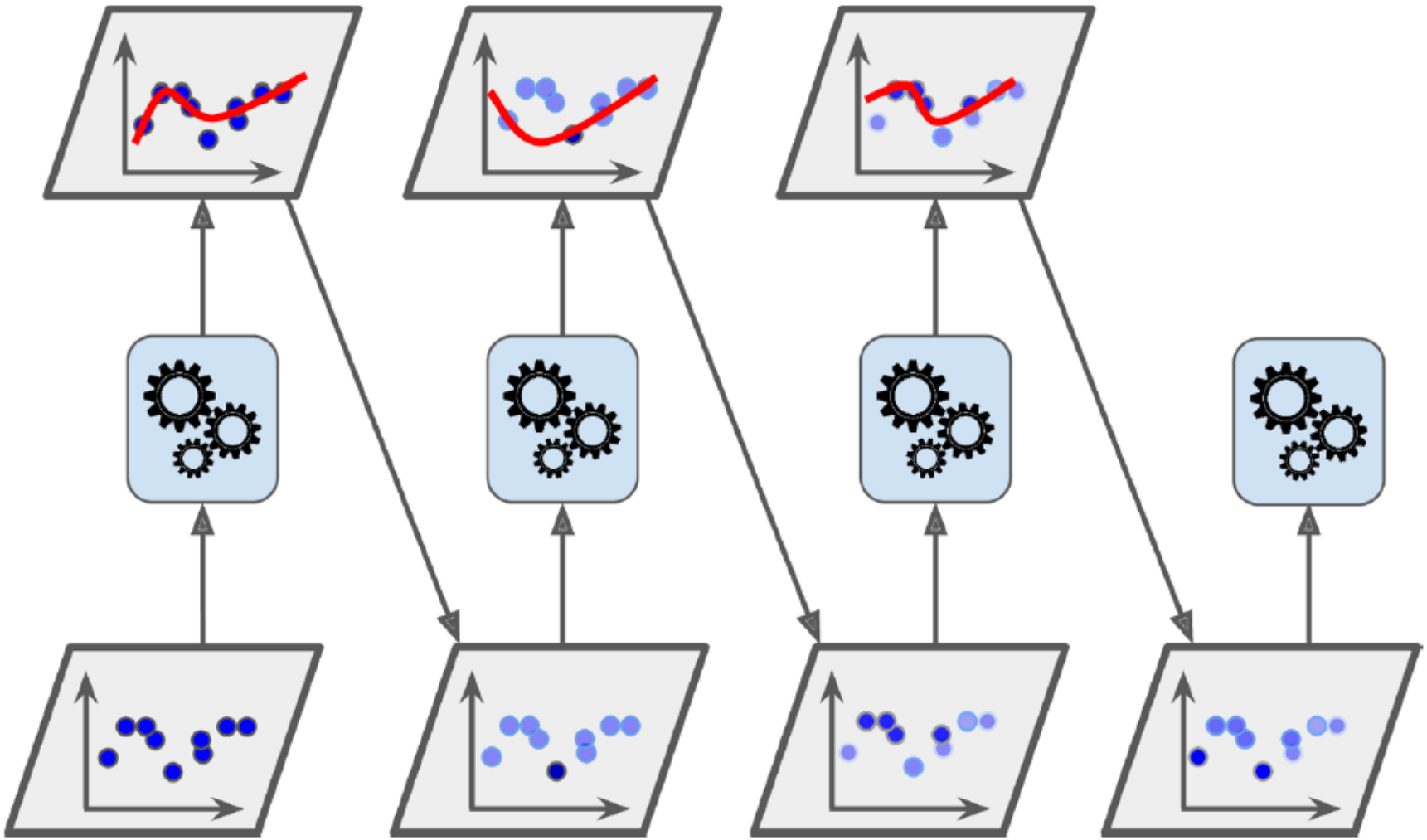
- ✓ train predictors sequentially, each trying to correct its predecessor.

- ✓ **Adaptive Boosting and Gradient Boosting.**

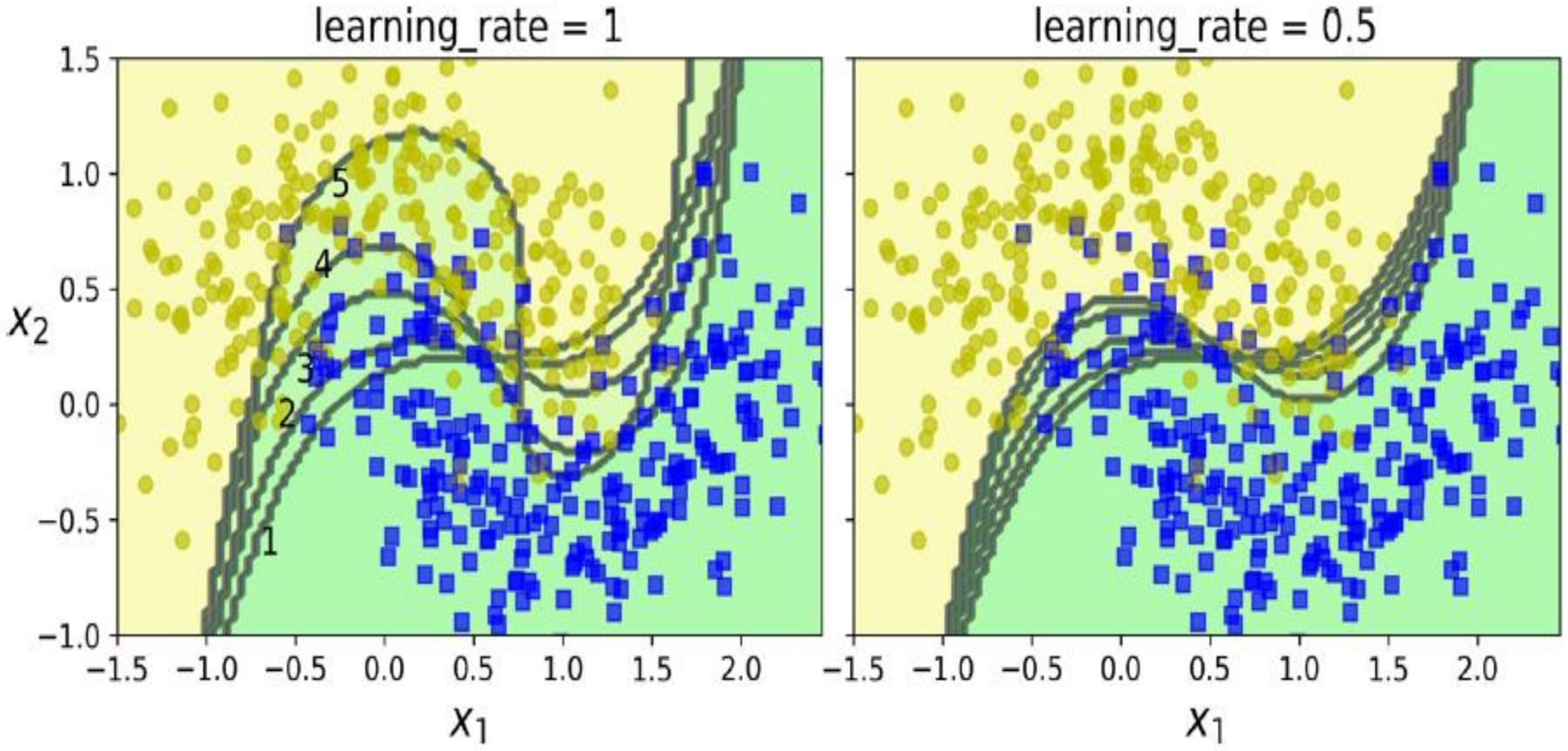
Adaptive Boosting

- ✓ Pay a bit more attention to the training instances that the predecessor underfitted
- ✓ sequential training
- ✓ increases the relative weight of misclassified training instances
- ✓ it cannot be parallelized (or only partially)

AdaBoost with instance weight updates



ecision boundaries of consecutive predicto



AdaBoost algorithm

Each instance weight $w^{(i)}$ is initially set to $1/m$.

Weighted error rate of the j^{th} predictor

$$r_j = \frac{\sum_{i=1}^{m} w^{(i)}}{\sum_{i=1}^{m} w^{(i)}} \quad \text{where } \hat{y}_j^{(i)} \neq y^{(i)}$$

where $\hat{y}_j^{(i)}$ is the j^{th} predictor's prediction for the i^{th} instance.

Predictor weight

$$\alpha_j = \eta \log \frac{1 - r_j}{r_j}$$

Weight update rule

for $i = 1, 2, \cdots, m$

$$w^{(i)} \leftarrow \begin{cases} w^{(i)} & \text{if } \widehat{y_j}^{(i)} = y^{(i)} \\ w^{(i)} \exp\left(\alpha_j\right) & \text{if } \widehat{y_j}^{(i)} \neq y^{(i)} \end{cases}$$

Then all the instance weights are normalized (divided by $\sum_{i=1}^{m} w^{(i)}$).

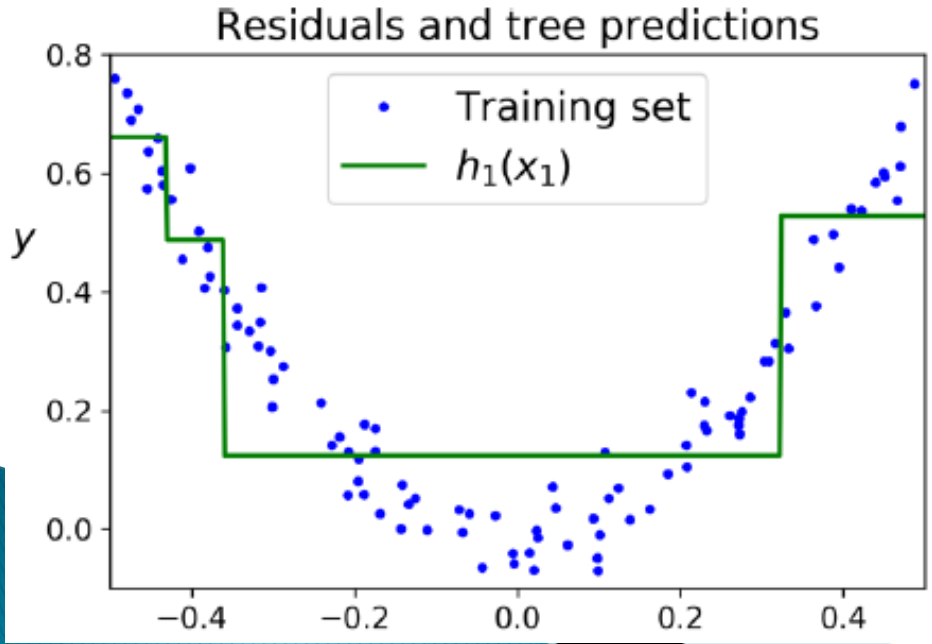
AdaBoost predictions

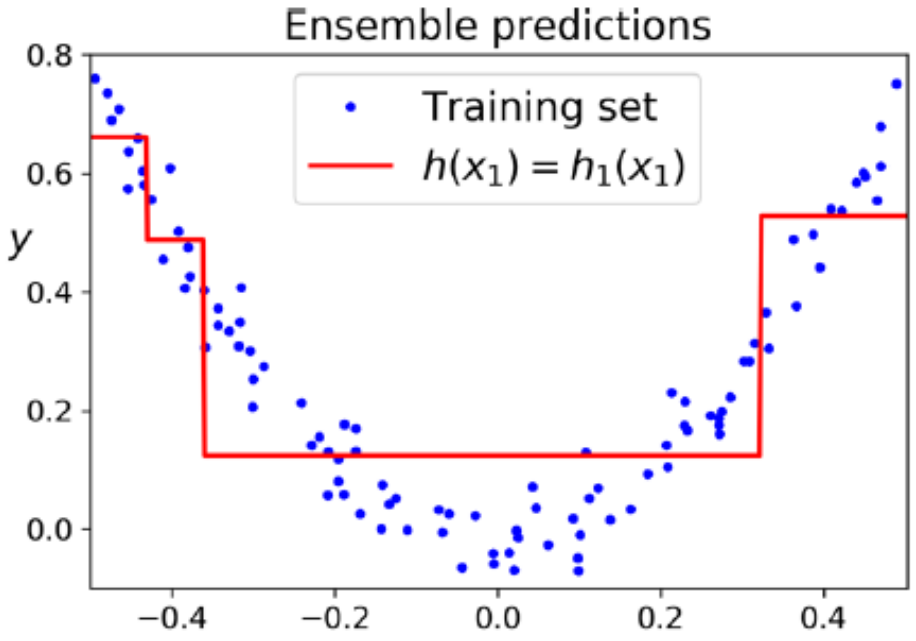
- ✓ AdaBoost simply computes the predictions of all the predictors and weighs them using the predictor weights αj.
- ✓ The predicted class is the one that receives the majority of weighted votes

$$\hat{y}(\mathbf{x}) = \underset{k}{\operatorname{argmax}} \sum_{\substack{j=1 \\ \hat{y}_j(\mathbf{x}) = k}}^{N} \alpha_j$$ where N is the number of predictors.

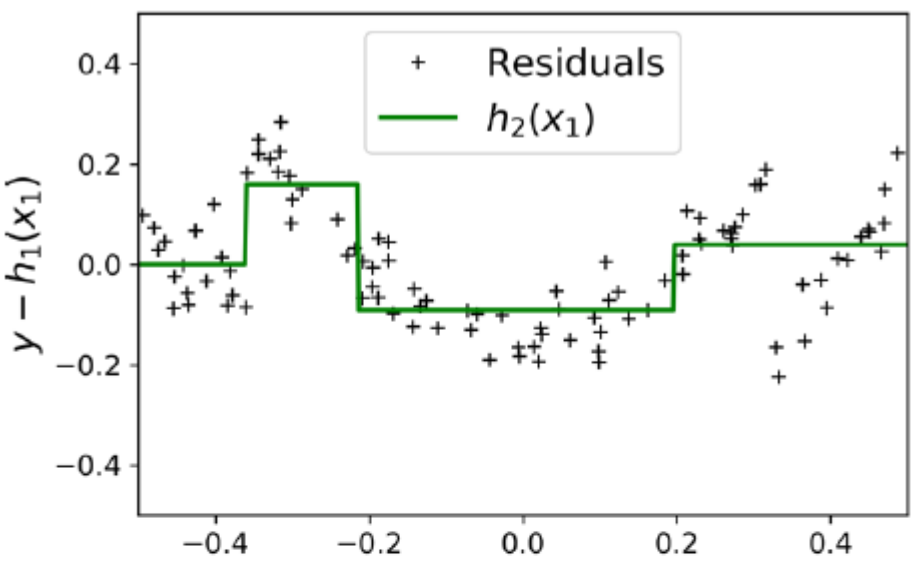
Gradient Boosting

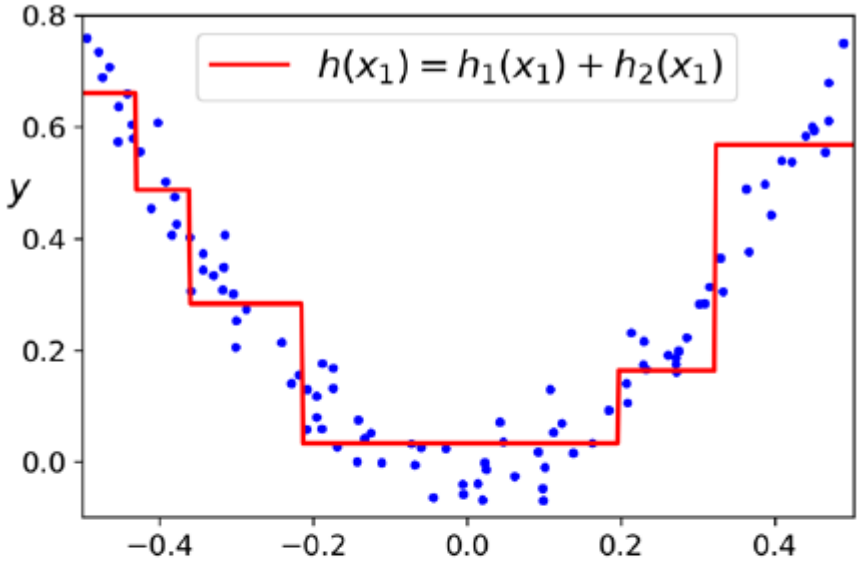
- ✓ tries to fit the new predictor to the residual errors made by the previous predictor.
- ✓ Gradient Tree Boosting, or Gradient Boosted Regression Trees (GBRT)

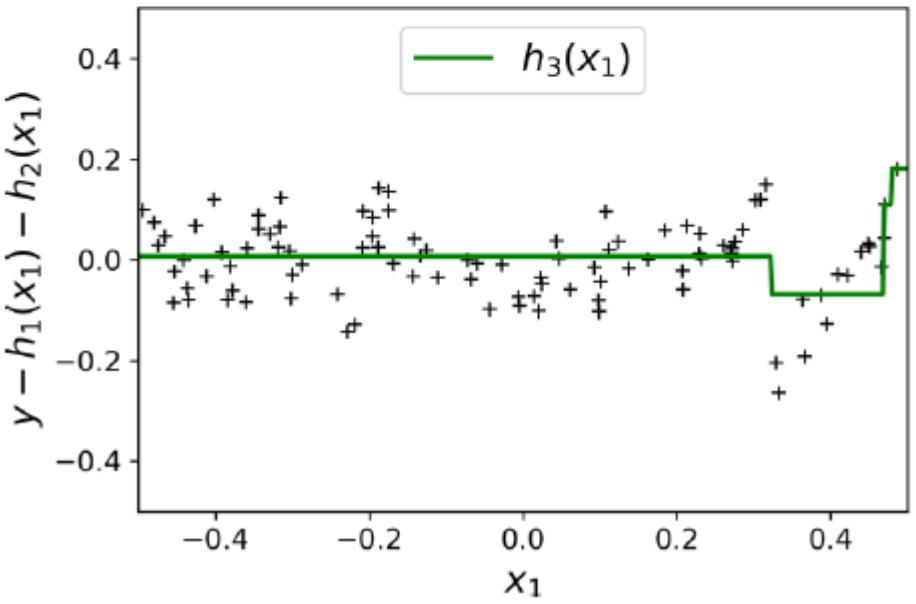


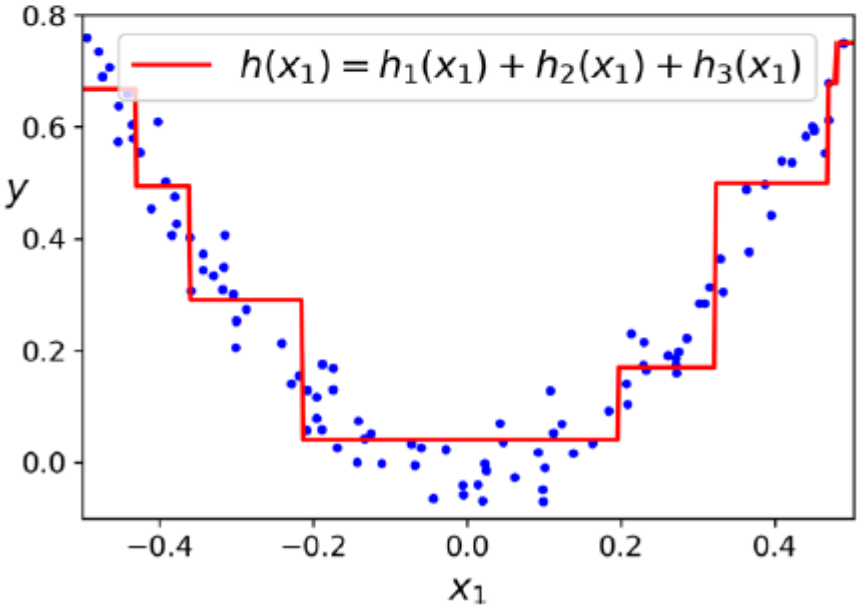


Gradient Boosting



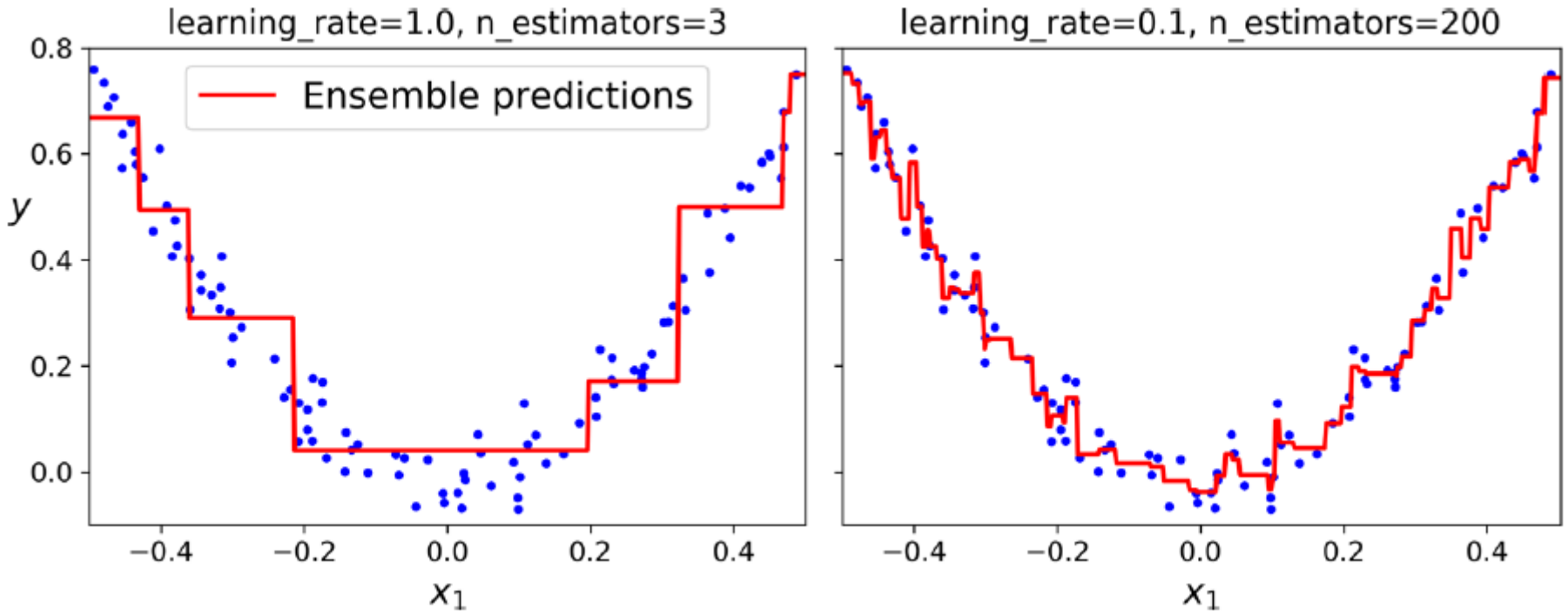




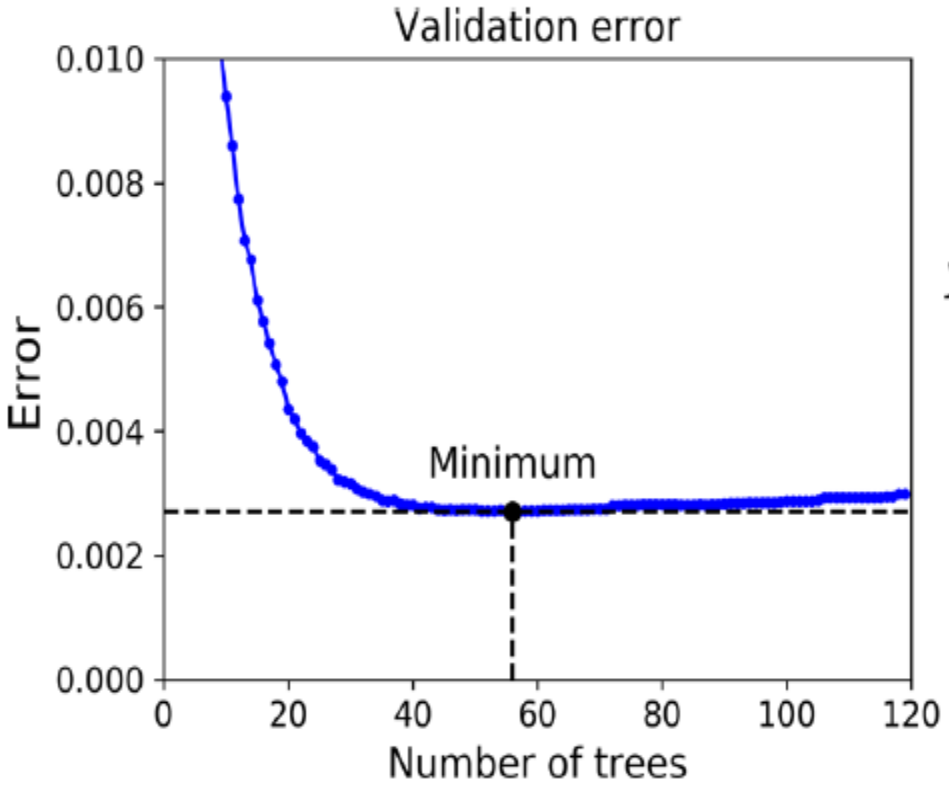


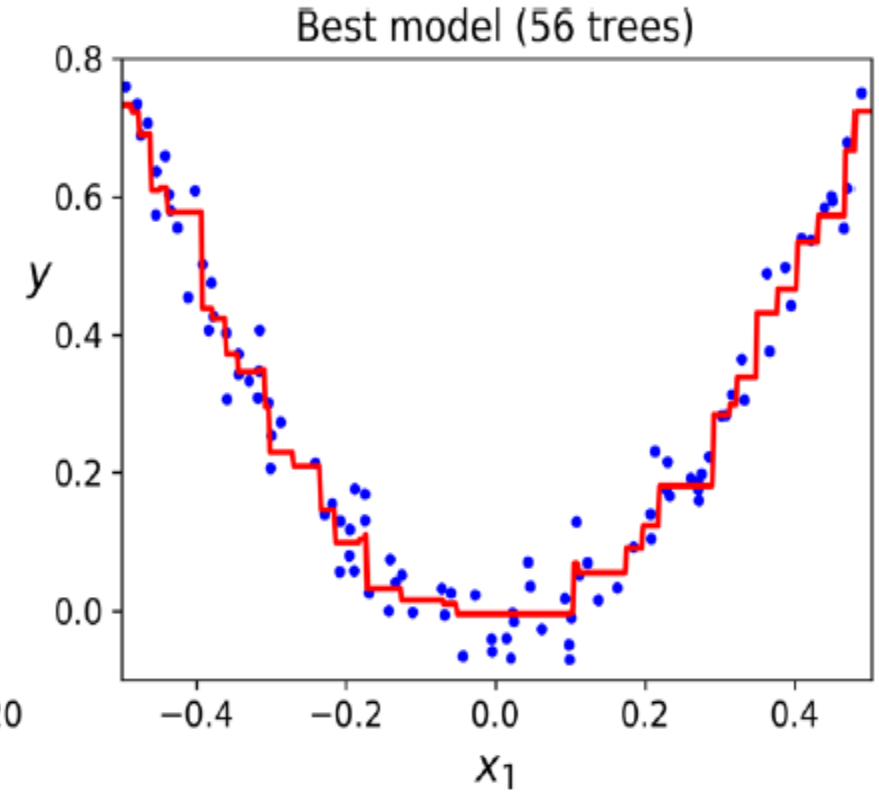
Gradient Boosting learning rate

- ✓ low value, such as 0.1, you will need more trees
- ✓ But generalize better. This is a regularization technique called shrinkage



Tuning the number of trees





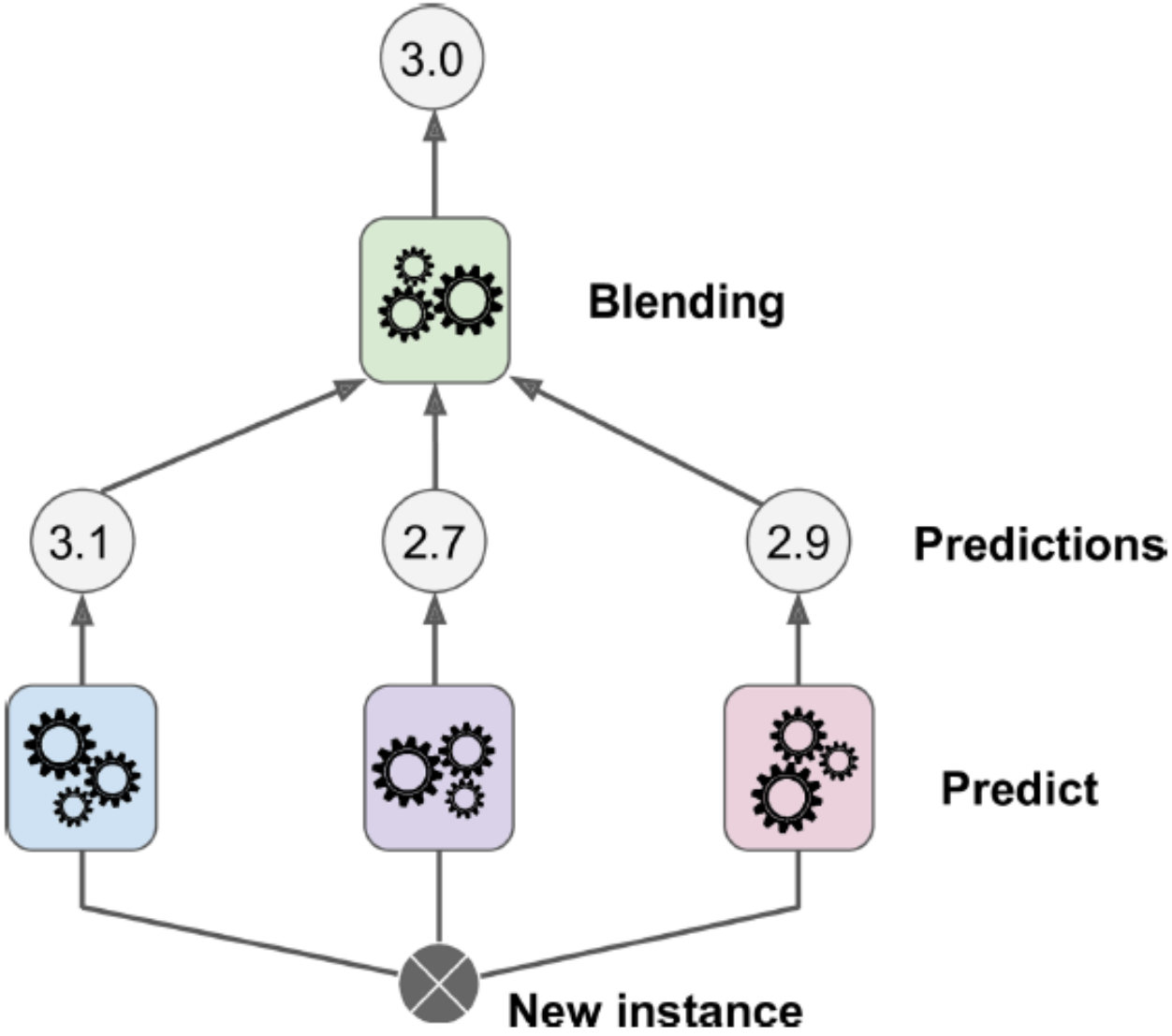
Stochastic Gradient Boosting

- ✓ A fraction of training instances are used for training each tree
- ✓ For example 25% of the training instances, selected randomly
- ✓ higher bias for a lower variance
- ✓ speed up training considerably

Stacked generalization

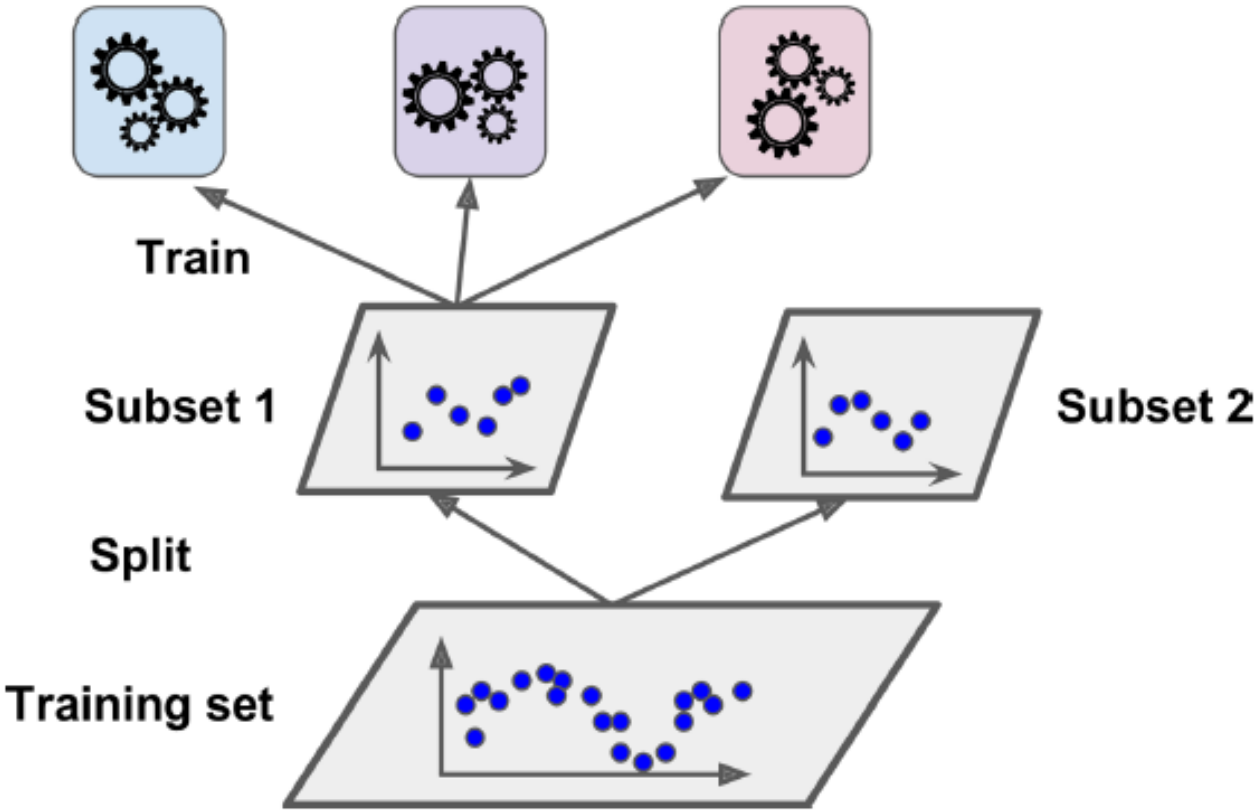
- ✓ Instead of using trivial functions (such as hard voting) to aggregate the predictions; train a model to perform this aggregation
- ✓ final predictor (called a blender, or a meta learner) takes output predictions as inputs and makes the final prediction

Stacking



Training the blender by hold-out set

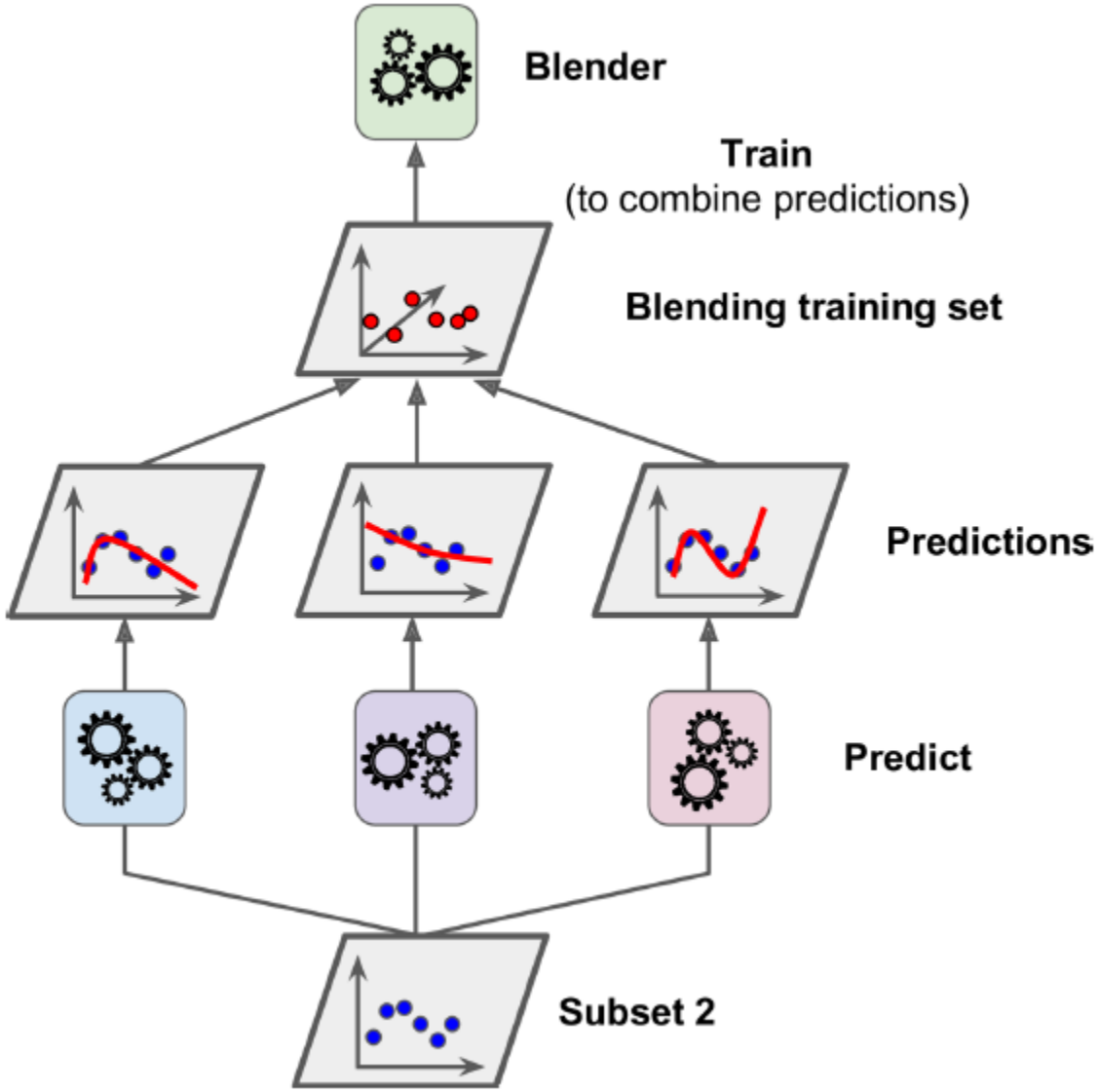
- ✓ training set is split into two subsets. The first subset is used to train the predictors in the first layer



Training the blender

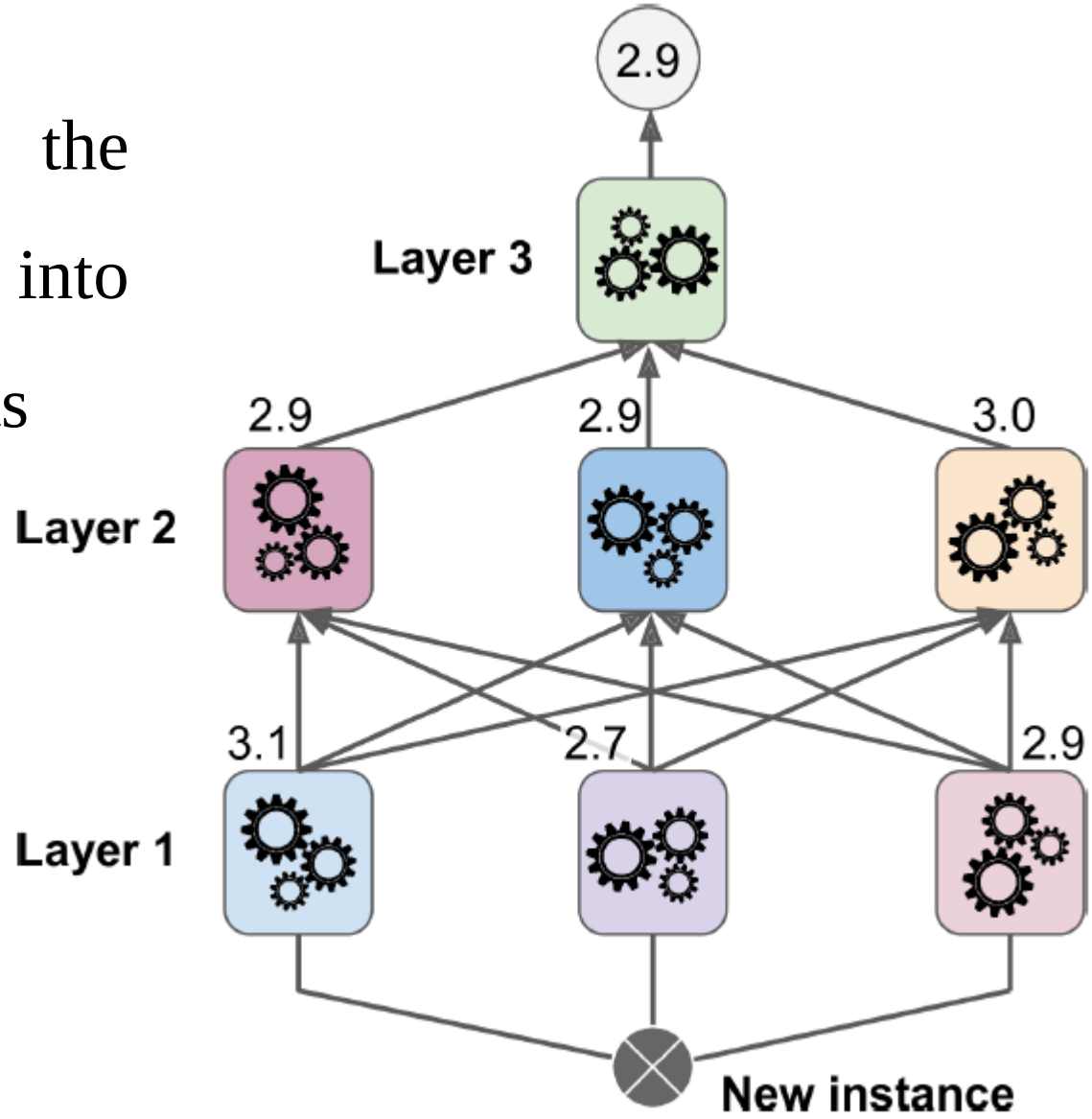
- ✓ first *layer's* predictors are used to make predictions on the second (held out) set
- ✓ create a new training set using these predicted values as input features
- ✓ The blender is trained on this new training set

Training the blender

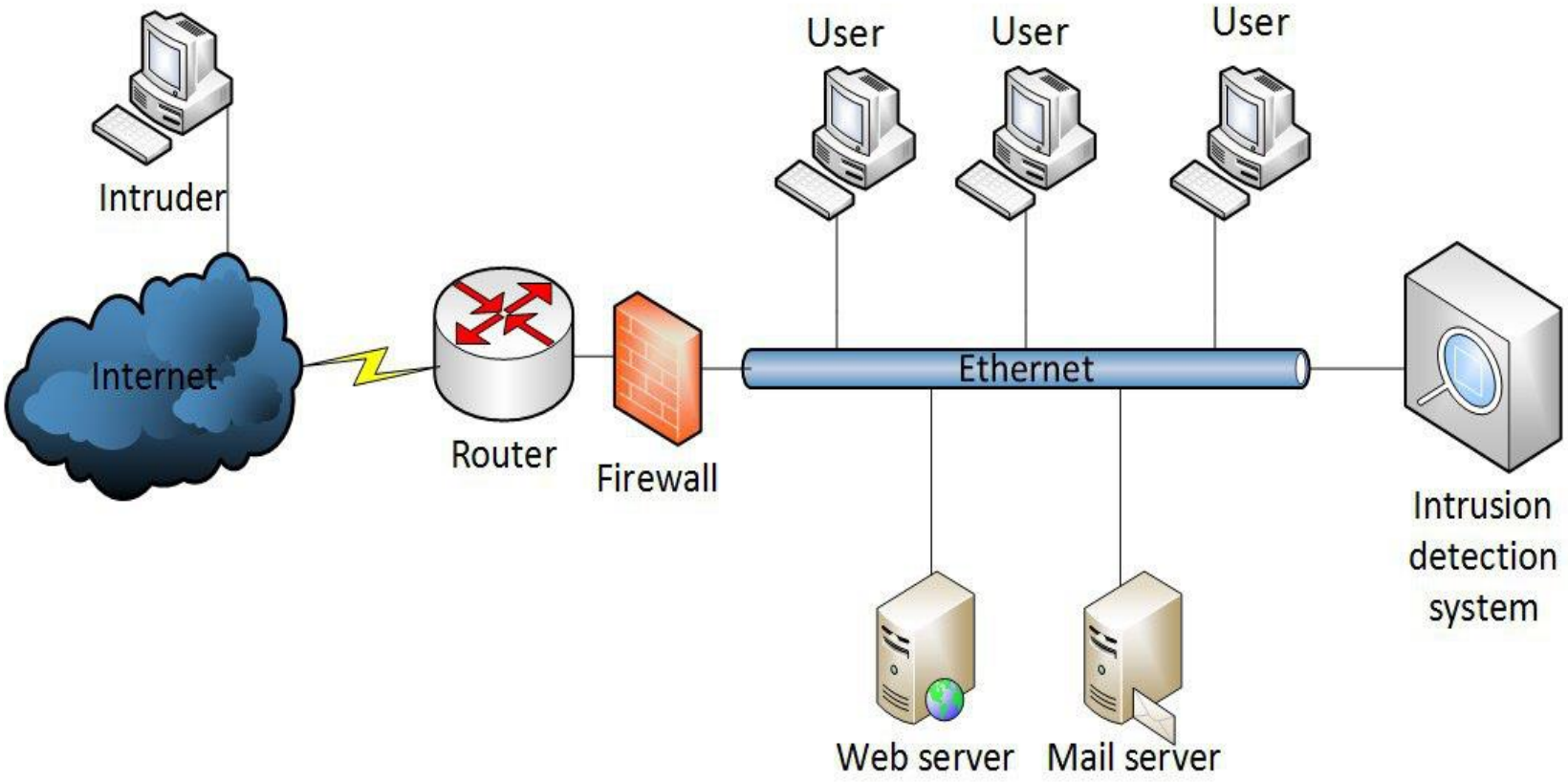


multilayer stacking ensemble

- ✓ split the training set into three subsets



IDS (Intrusion Detection System)



KDD CUP 1999

- ✓ distinct dataset for both train-set and test-set and has 41 features

No.	*Feature name*	*Description*	*Type*
1	Duration	Length of the connection (second)	Continuous
2	Protocol _type	Type of protocol ,eg tcp,udp,etc	symbolic
3	Service	Network service on destination	symbolic
4	Flag	Normal or error status of the connection	symbolic
5	Src _bytes	Number of data bytes from source to destination	Continuous
6	Dst _bytes	Number of data bytes from destination to source	Continuous
7	Land	1 if connection is from /to the same host/port;0 otherwise	Discrete
8	Wrong _fragment	Number of "wrong" fragments	Continuous
9	Urgent	Number of urgent packets	Discrete
10	Hot	Number of "hot" indicators	Continuous
11	Num _failed _logins	Number of failed login attempts	Continuous
12	Logged _in	1 if successfully logged in ; 0 otherwise	Continuous

KDD CUP 1999

No.	*Feature name*	*Description*	*Type*
13	Num _ compromised	Number of compromised condition	Continuous
14	Root_ shell	1 if root shell is obtained ; 0 otherwise	Continuous
15	Su_ attempt	1 if "su root" command attempted ; 0 otherwise	Continuous
16	Num_root	Number of "root" accesses	Continuous
17	Num_file_ceations	Number of file creation operations	Continuous
18	Num_shells	Number of shell prompts	Continuous
19	Num_access_files	Number of operations on access control files	Continuous
20	Num_outbound_cmds	Number of outbound commands in an ftp session	Continuous
21	Is_host_login	*1 if the login belongs to the "hot" list; 0 otherwise*	Continuous
22	Is_guest_login	*1 if the login is a "guest "login; 0 otherwise*	Continuous
23	Count	number of connections to the same host as the current connection in the past two seconds	Continuous
24	Srv_count	Number of connections to the same service as the current connection in the past two seconds	Continuous
25	Serror_rate	*% of connections that have "SYN" errors*	Continuous
26	Srv_serror_rate	*% of connections that have "SYN" errors*	Continuous
27	Rerror_rate	*% of connections that have "REJ" errors*	Continuous

KDD CUP 1999

No.	*Feature name*	*Description*	*Type*
28	Srv_rerror_rate	*% of connections that have "REJ" errors*	Continuous
29	Same_srv_rate	% of connections to the same services	Continuous
30	Diff_srv_rate	% of connections to different services	Continuous
31	Srv_diff_host_rate	% of connections to different hosts	Continuous
32	Dst_host_count	Count for destination host	Continuous
33	Dst_host_srv_count	Srv_count for destination host	Continuous
34	Dst_host_same_srv_rate	Same_srv_rate for destination host	Continuous
35	Dst_host_diff_srv_rate	Diff_srv_rate for destination host	Continuous
36	Dst_host_same_src_port_rate	Same_src_port_rate for destination host	Continuous
37	Dst_host_srv_diff_host_rate	Diff_host_rate for destination host	Continuous
38	Dst_host_serror_rate	Serror_rate for destination host	Continuous
39	Dst_host_srv_serror_rate	Srv_serror_rate for destination host	Continuous
40	Dst_host_rerror_rate	Rerror_rate for destination host	Continuous
41	Dst_host_srv_rerror_rate	Srv_serror_rate for destination host	Continuous

KDD CUP 1999

Type	*Sub-attack*	*Number of records*
PROBE	ipsweep,portsweep,satan,nmap	41102
DOS	neptune,smurf,pod,teatdrop,land,back	3883370
U2R	buffer_overflow,loadmodule,perl,rootkit	52
R2L	guss_passwd,ftp_write,imap,phf,multihop,warezmaster, warezclient,spy	1126
NORMAL	Normal	972781

MNIST dataset