Bipartite Majority Learning with Tensors

Author：Chia-Lun Lee

Advisor: Professor Fang Yu

Software Security Lab, Dept. Management Information Systems

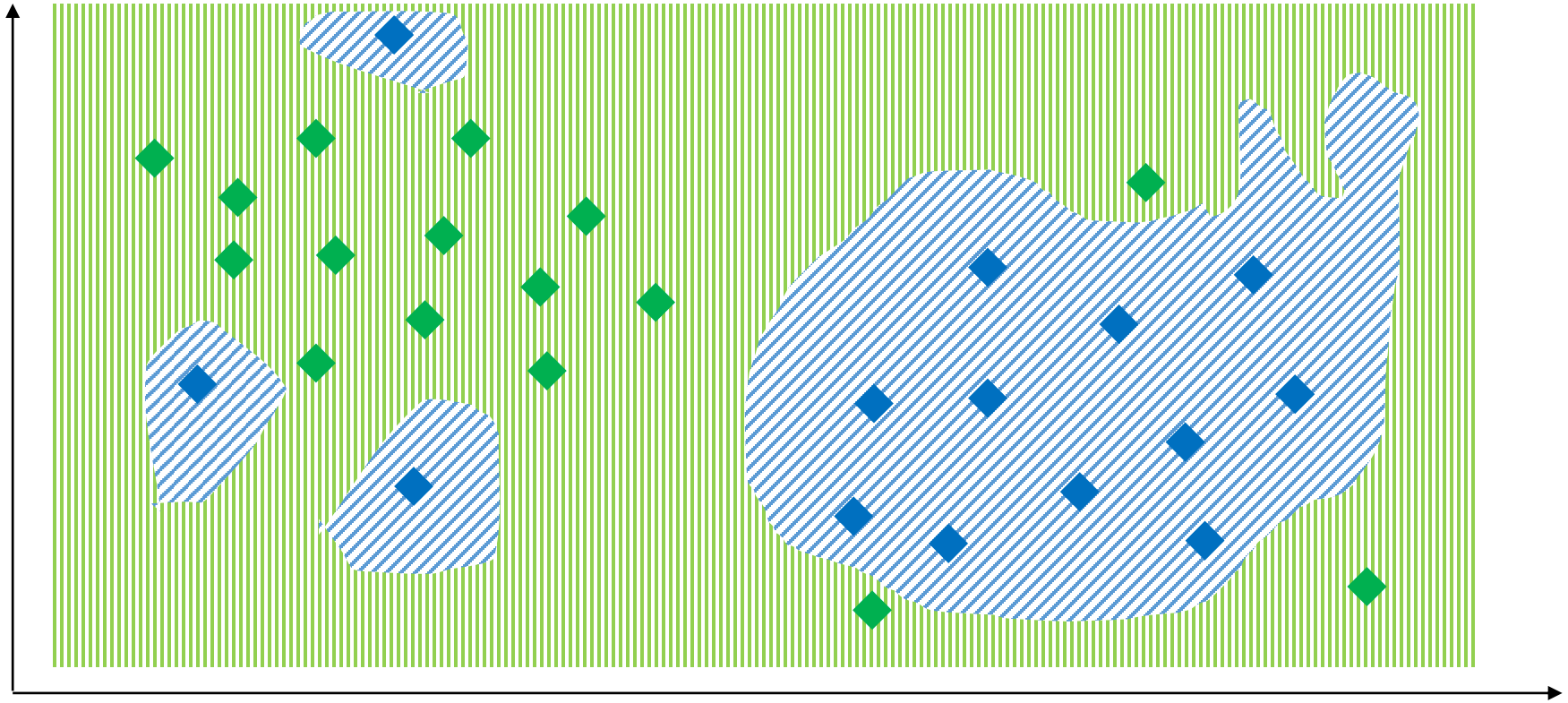
National Chengchi University (NCCU), Taiwan (ROC)

Overview

- Propose a new bipartite majority learning algorithm (BML) for effective resistant learning on two party classification.
- Efficiently train an ANN model to separate two parties in majority without knowing sample distributions and avoid the impacts of outliers.
- Realize the idea with Tensors to leverage GPU computation.
- Reduce training time significantly with competitive accuracy on malware classification of real world data.

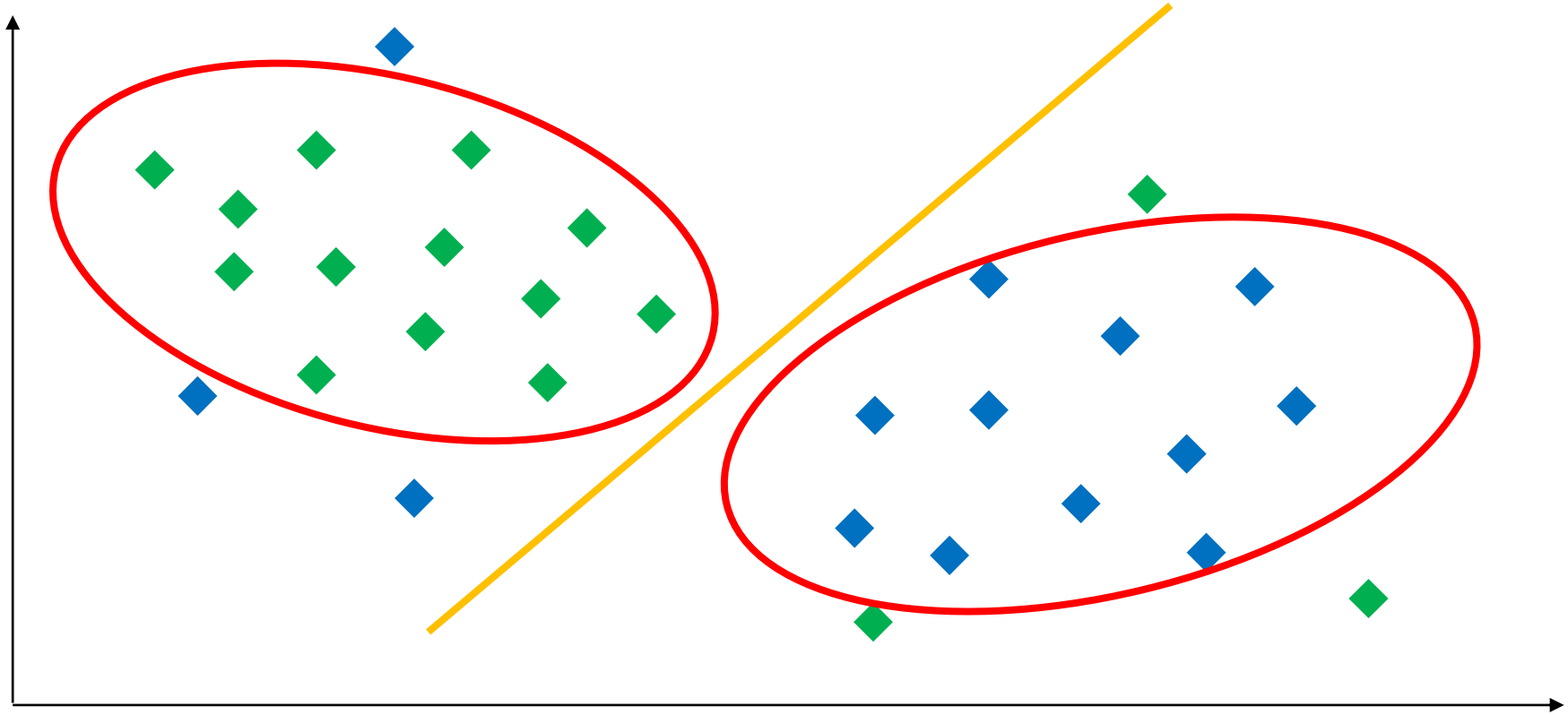
Bipartite Classification Problem

- Perfect bipartite classification.



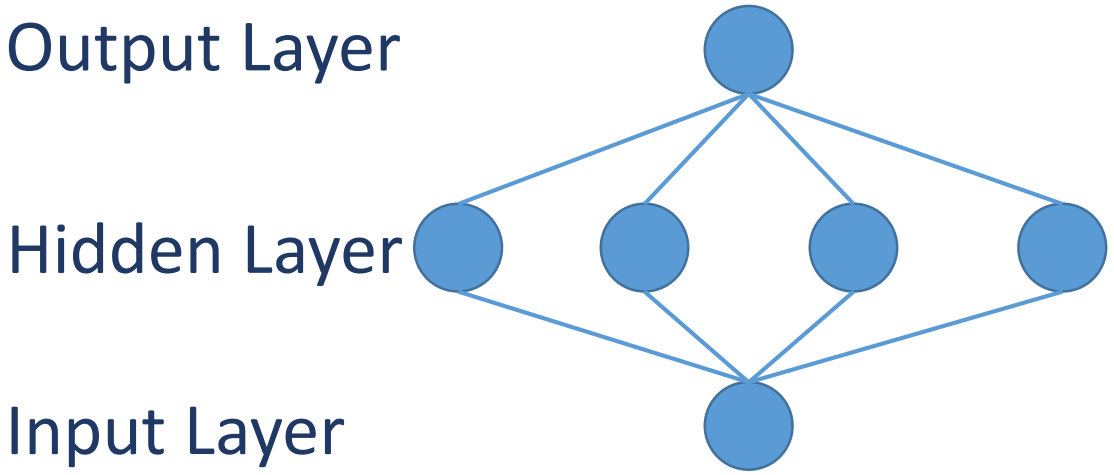
Bipartite Classification Problem

- Bipartite classification for majority.

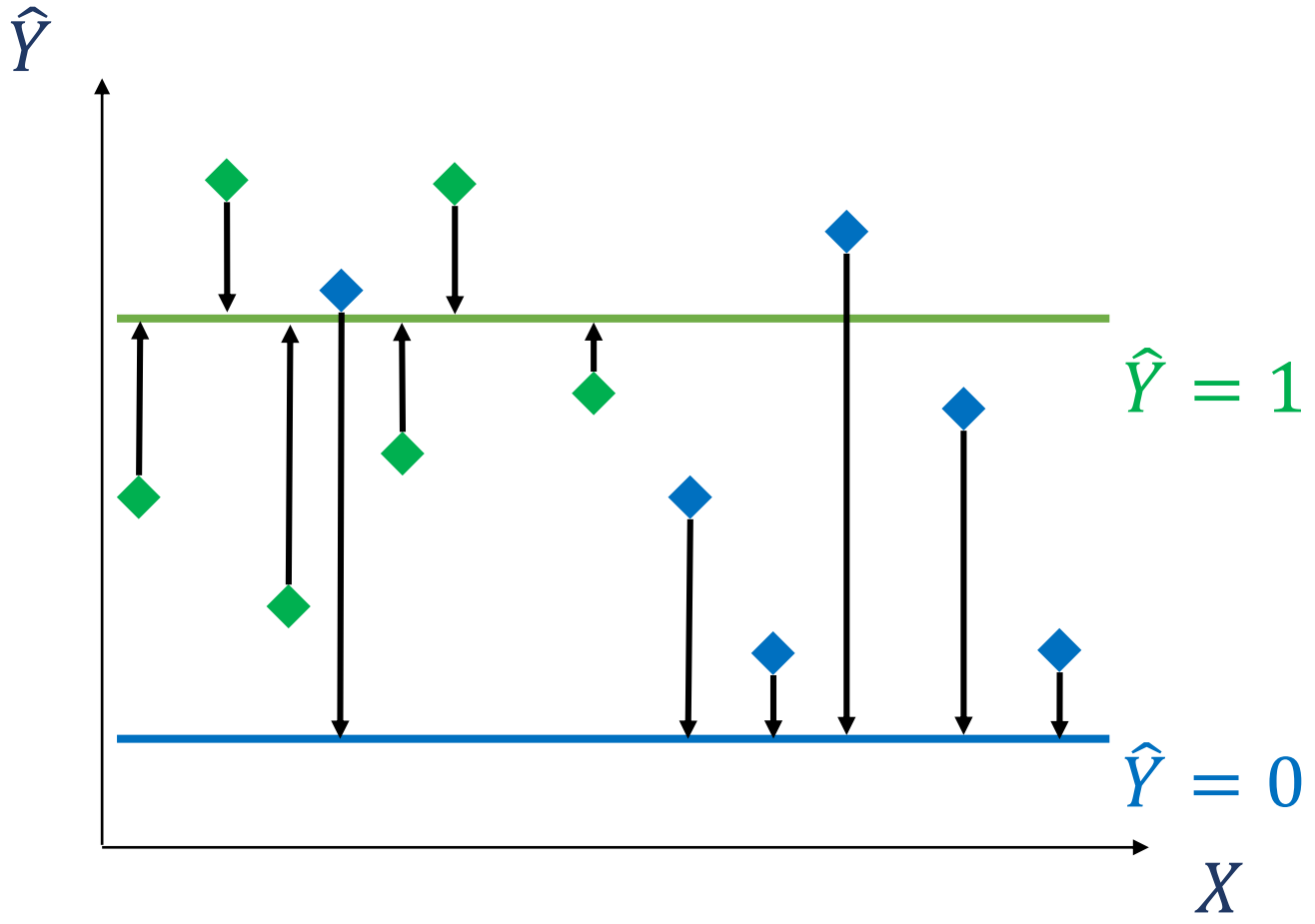


Artificial Neural Network

- Gradient descent may stuck in local optimal.

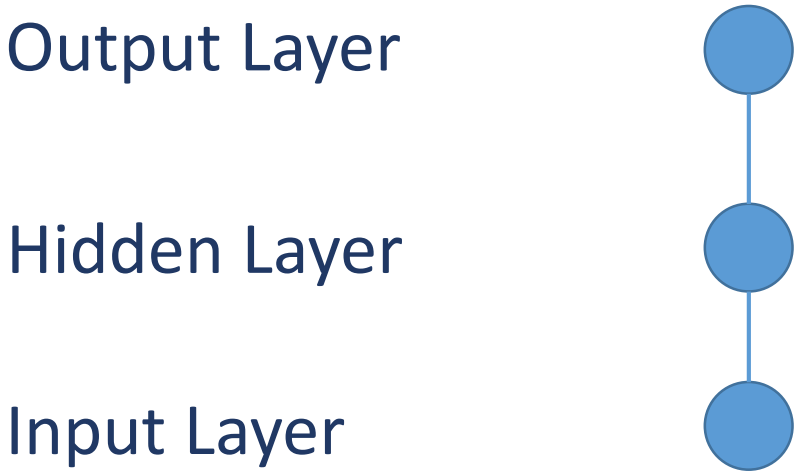


Artificial Neural network (ANN)

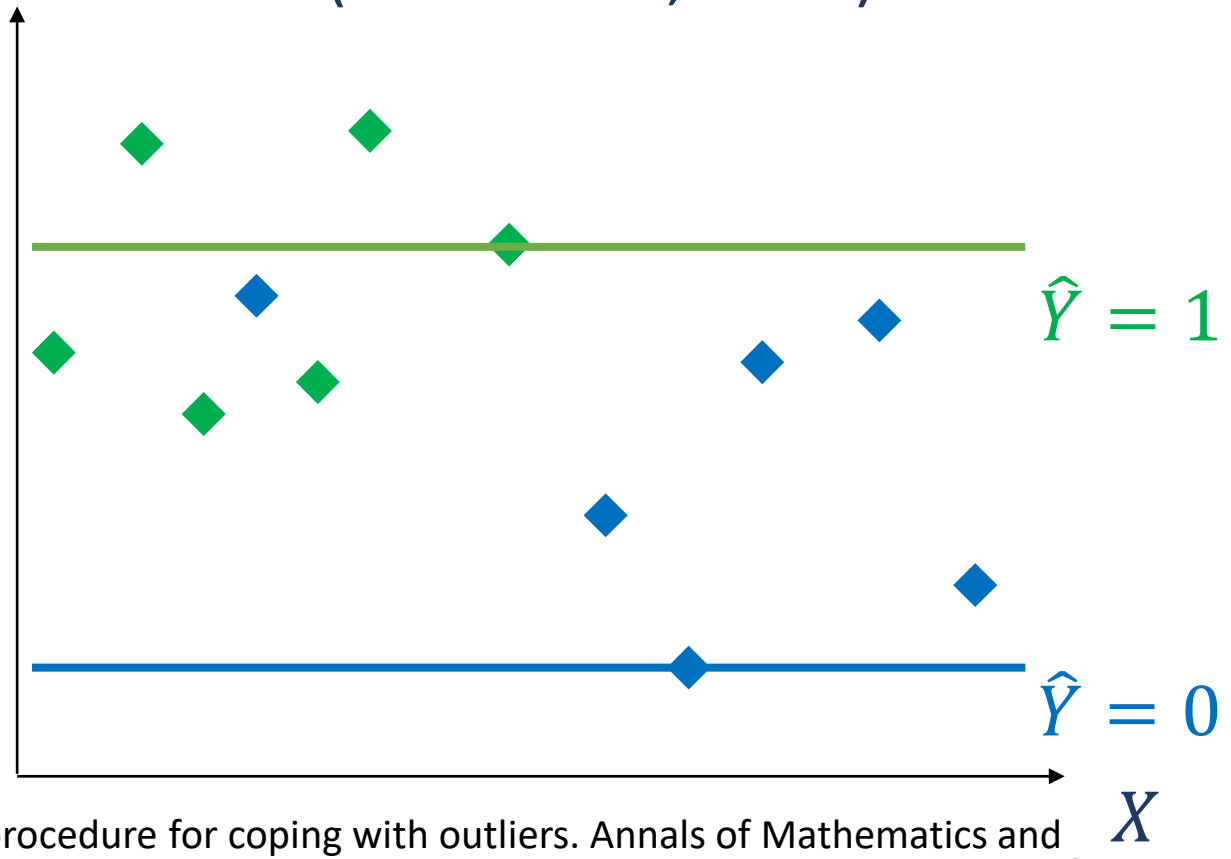


Resistant Learning with SLFN

- Learning pattern of majority without knowing sample distributions.
- Iteratively select training data to train SLFN. (Tsaih et al., 2009)



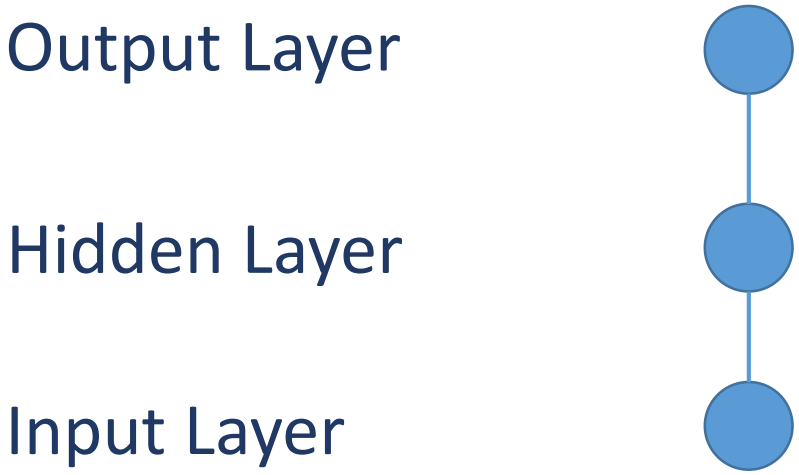
Single-hidden Layer Feed-forward Neural network (SLFN)



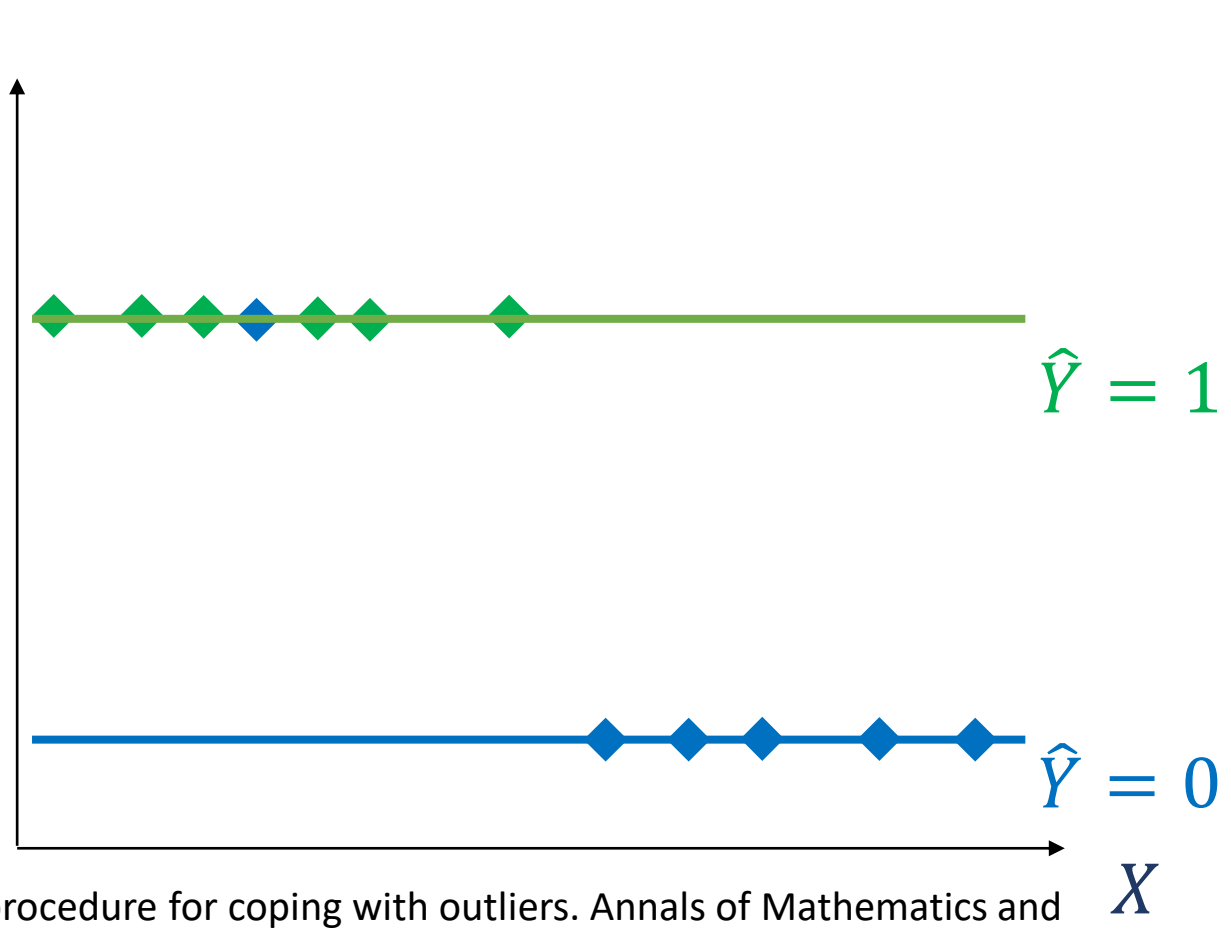
Reference: Tsaih, R. H., & Cheng, T. C. (2009). A resistant learning procedure for coping with outliers. Annals of Mathematics and Artificial Intelligence, 57(2), 161-180.

Resistant Learning with SLFN

- Train a neural network to predict the output value of a specific data point



Single-hidden Layer Feed-forward Neural network (SLFN)



Reference: Tsaih, R. H., & Cheng, T. C. (2009). A resistant learning procedure for coping with outliers. Annals of Mathematics and Artificial Intelligence, 57(2), 161-180.

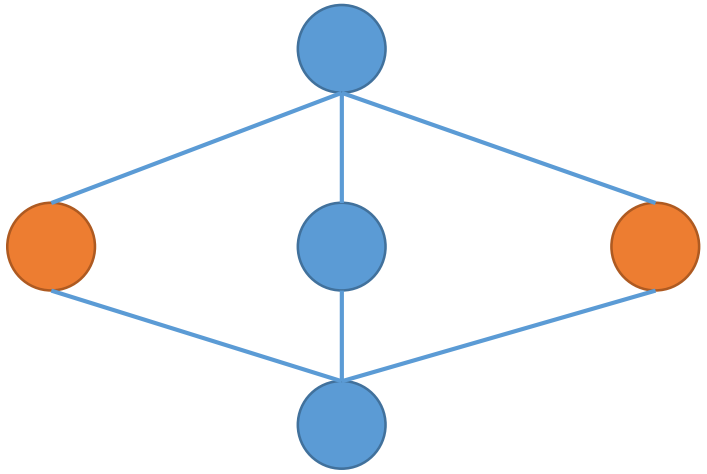
Resistant Learning with SLFN

- Add extra hidden nodes and calculate the weights of new nodes

Output Layer

Hidden Layer

Input Layer



$$w_{p-1}^{O} = w_{p}^{O} = \frac{\left| y^{k'} - w_0^O - \sum_{i=1}^{q} (w_i^O w_i^k) \right|}{2\tanh(\zeta)}$$

$$w_{p,0}^{H} = \zeta + \lambda \alpha^T x^k \qquad w_{p-1,0}^{H} = \zeta - \lambda \alpha^T x^k$$

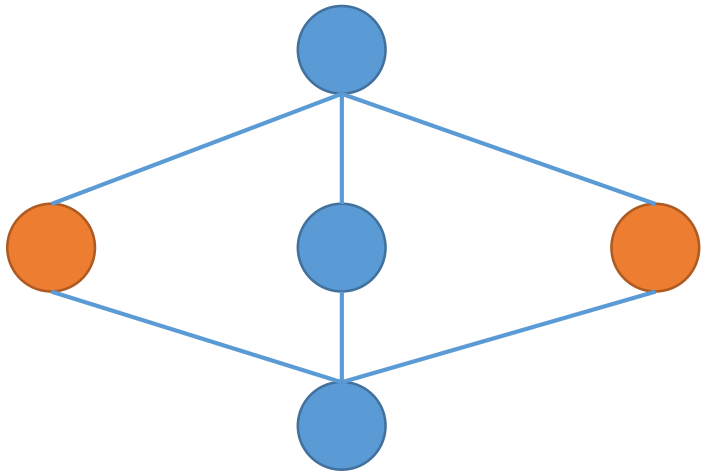
$$w_{p}^{H} = -\lambda \alpha^T \qquad w_{p-1}^{H} = \lambda \alpha^T$$

Single-hidden Layer Feed-forward Neural network (SLFN)

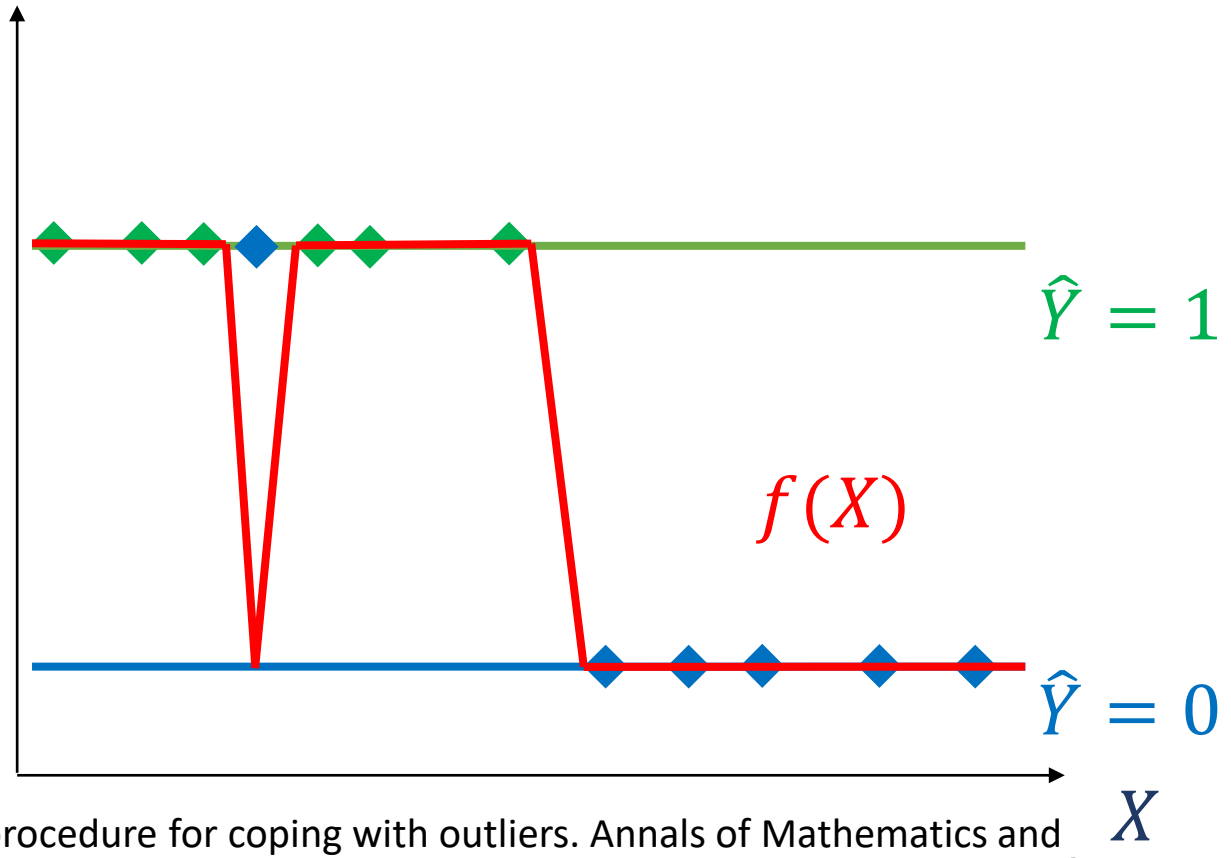
Reference: Tsaih, R. H., & Cheng, T. C. (2009). A resistant learning procedure for coping with outliers. Annals of Mathematics and Artificial Intelligence, 57(2), 161-180.

Resistant Learning with SLFN

- Add extra hidden nodes and calculate the weights of new nodes
- Find near-perfect fitting function
- Consume a lot of time



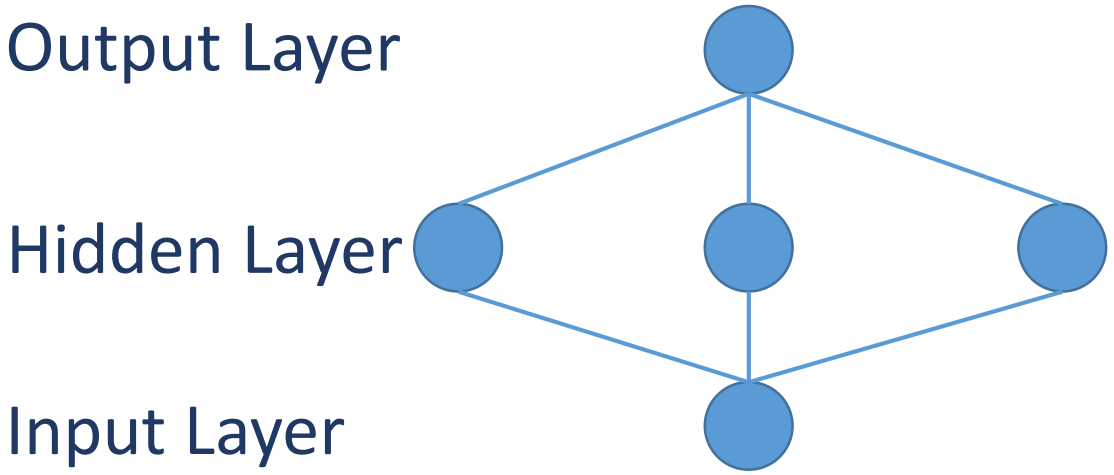
Single-hidden Layer Feed-forward Neural network (SLFN)

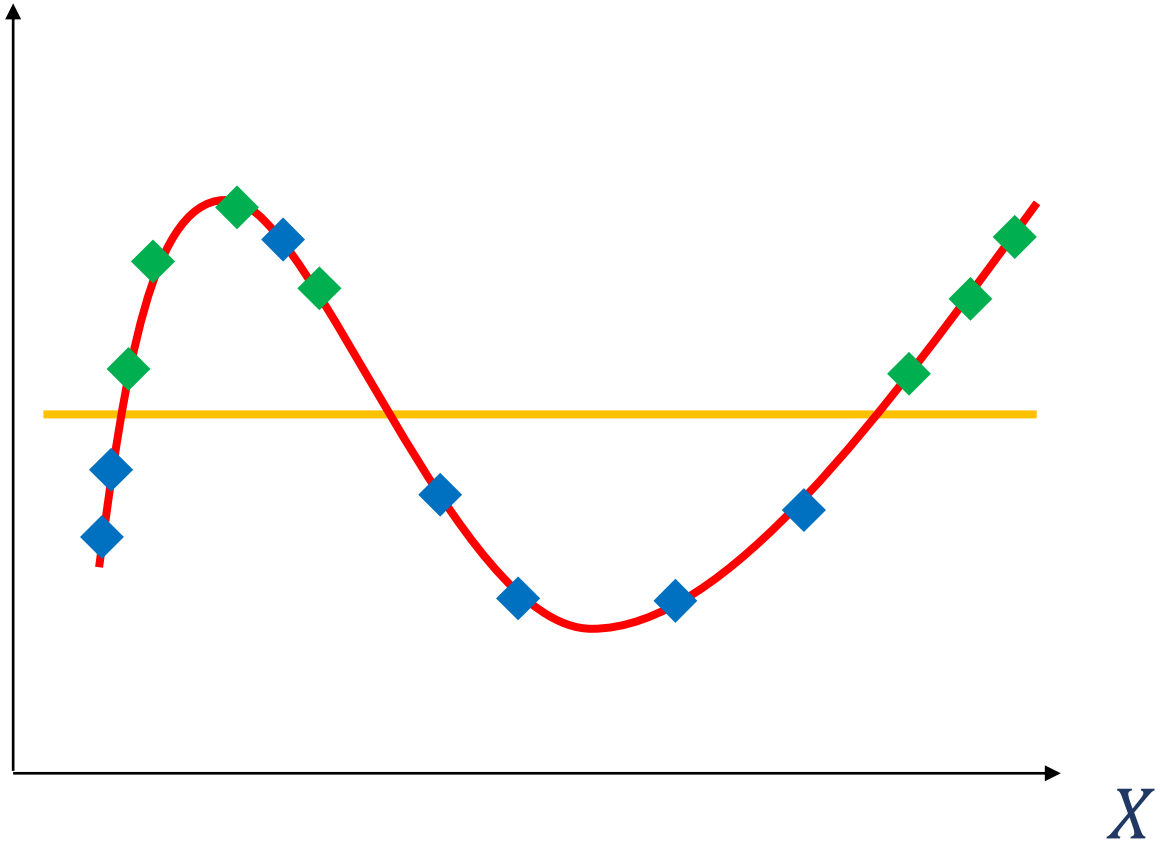


Reference: Tsaih, R. H., & Cheng, T. C. (2009). A resistant learning procedure for coping with outliers. Annals of Mathematics and Artificial Intelligence, 57(2), 161-180.

Bipartite Majority Learning (BML)

- Classify data by a threshold, fitting function is relatively less strict
- Find classification for majority $\hat{Y}$





BML: Three Concepts

- Sort and Select – Sort all training data by a principle, then gradually increase the amount of selected data for model training.

- Train – Apply gradient descent to find an acceptable SLFN with selected data.

- Resistant Procedure – Modify the model architecture (i.e., add additional hidden nodes) to find an acceptable SLFN when the selected training data does not fit the model after training.

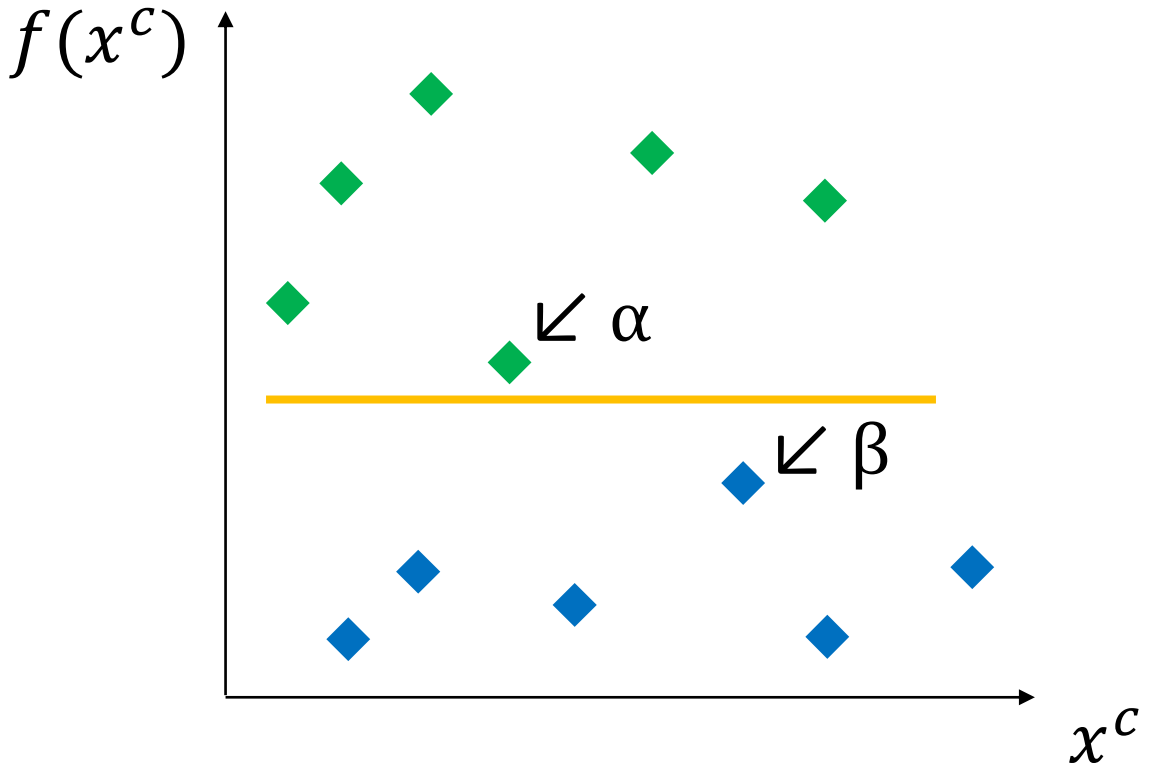
Condition on Linear Separation

- The linear separated condition (the condition L, Tsaih, 1993) is suitable for ANN dichotomy.

$$\alpha = min_{y^c \epsilon \{C_1\}} f(x^c)$$

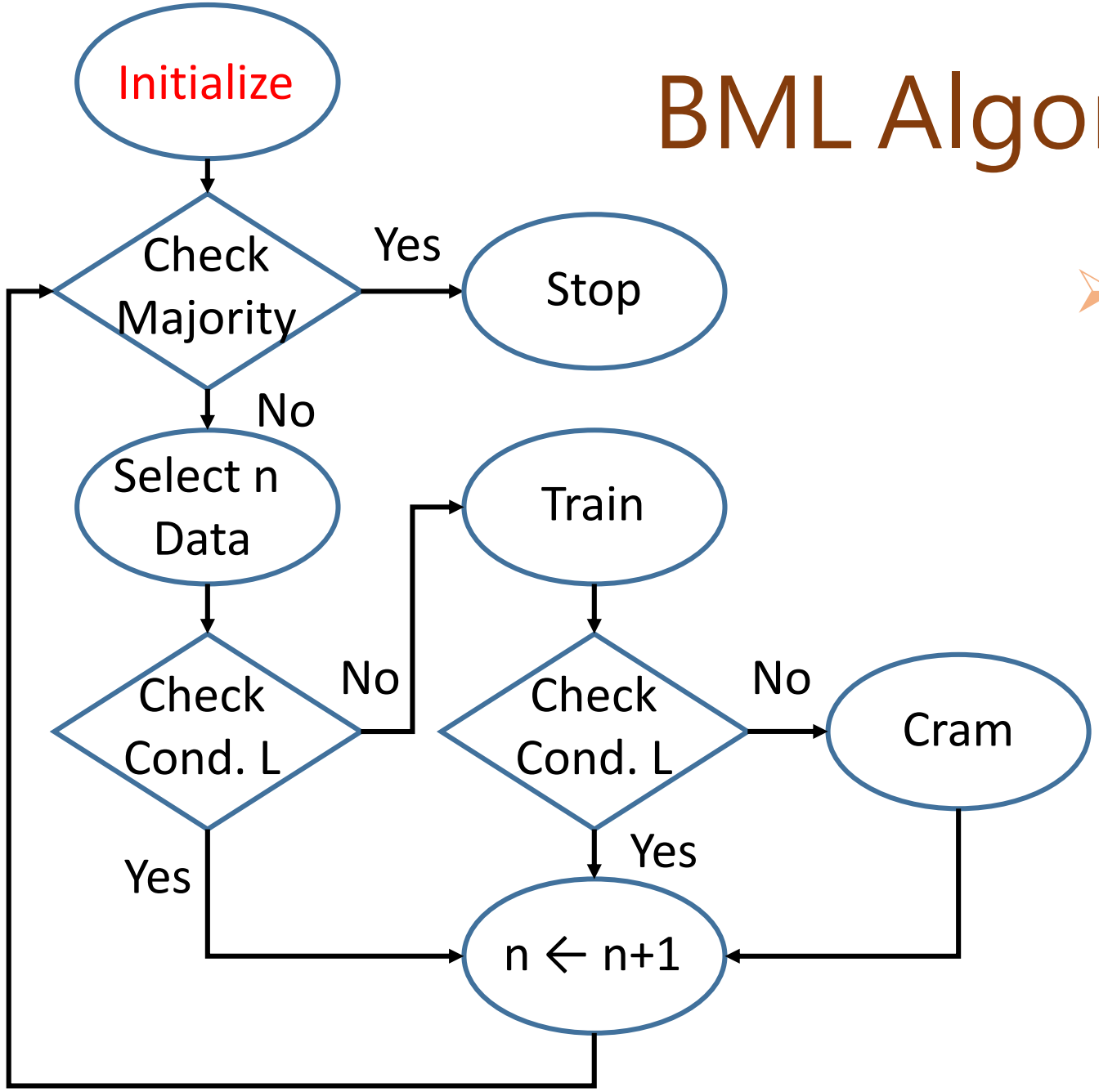
$$\beta = max_{y^c \epsilon \{C_2\}} f(x^c)$$

- If $\alpha > \beta$, the model can classify bipartite data by a threshold, the condition L is satisfied.



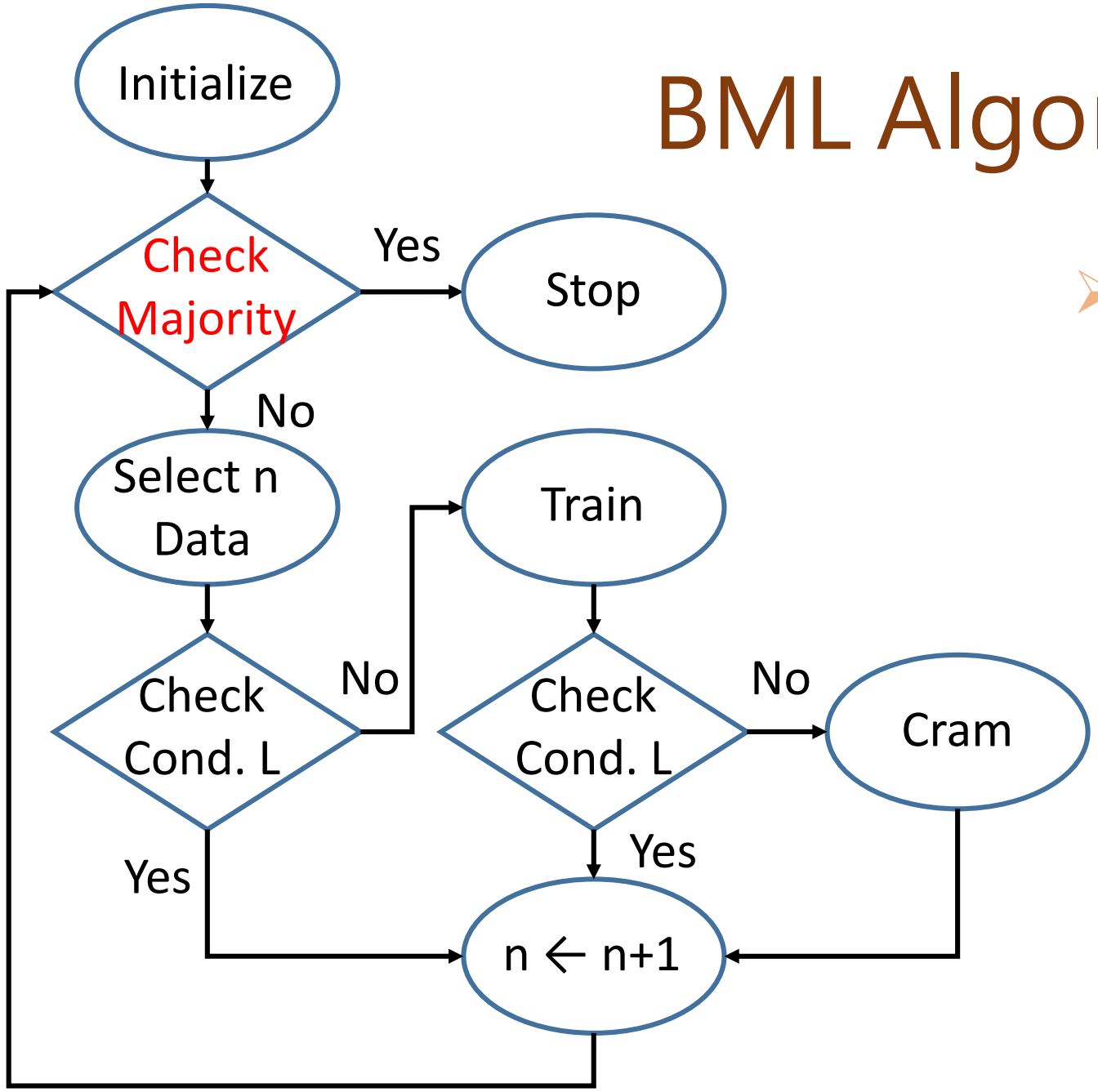
Reference: Tsaih, R. R. (1993). The softening learning procedure. Mathematical and computer modelling, 18(8), 61-64.

BML Algorithm



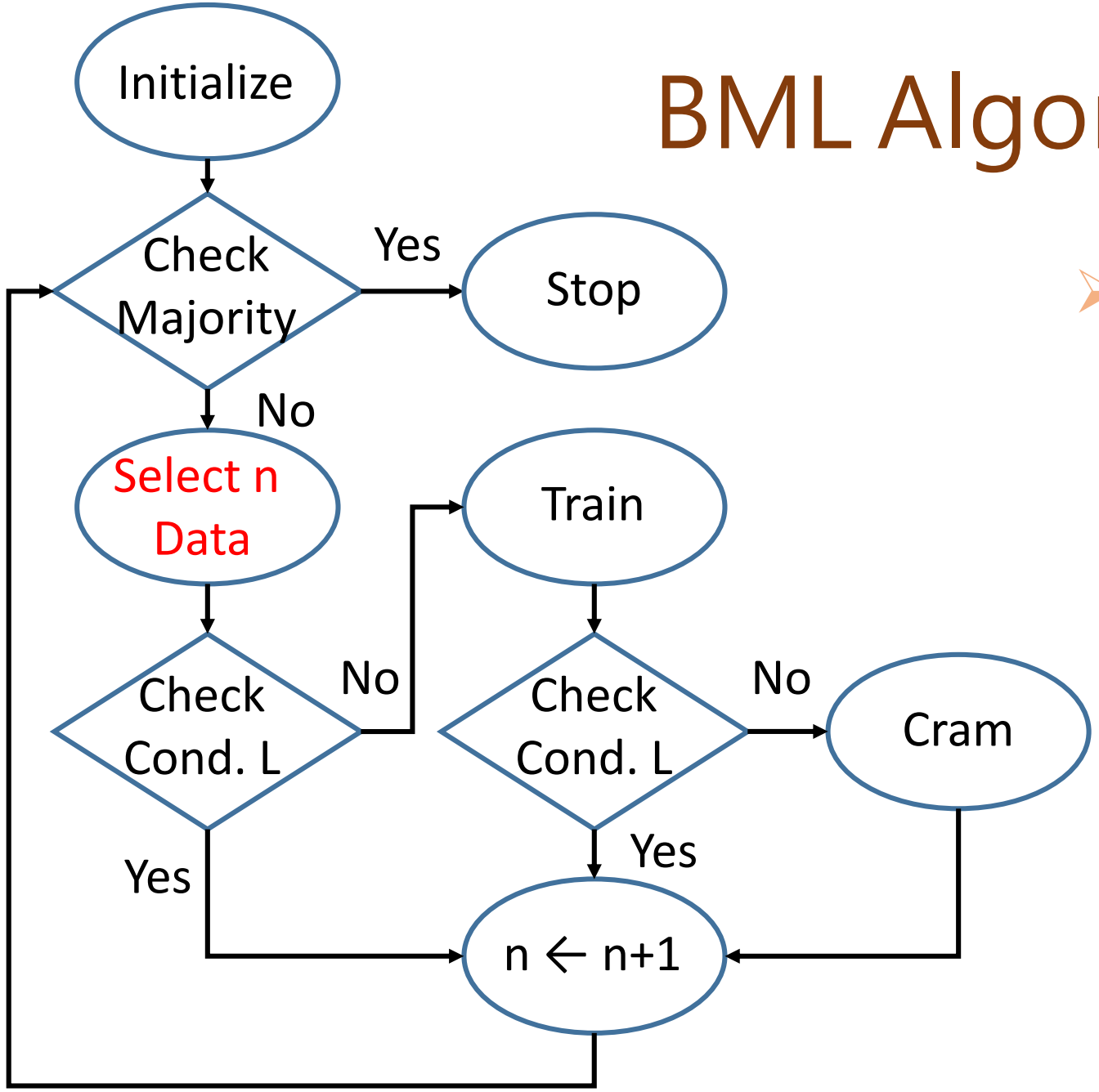
- Step 1: Initialize
  - Randomly obtain the initial m+1 data.
    - Two classes of data each account for half of the m+1 data.
  - Set up an acceptable SLFN with one hidden node.
    - By using simultaneous equations with these m+1 data.
  - Set n = m+2.

BML Algorithm (cont.)



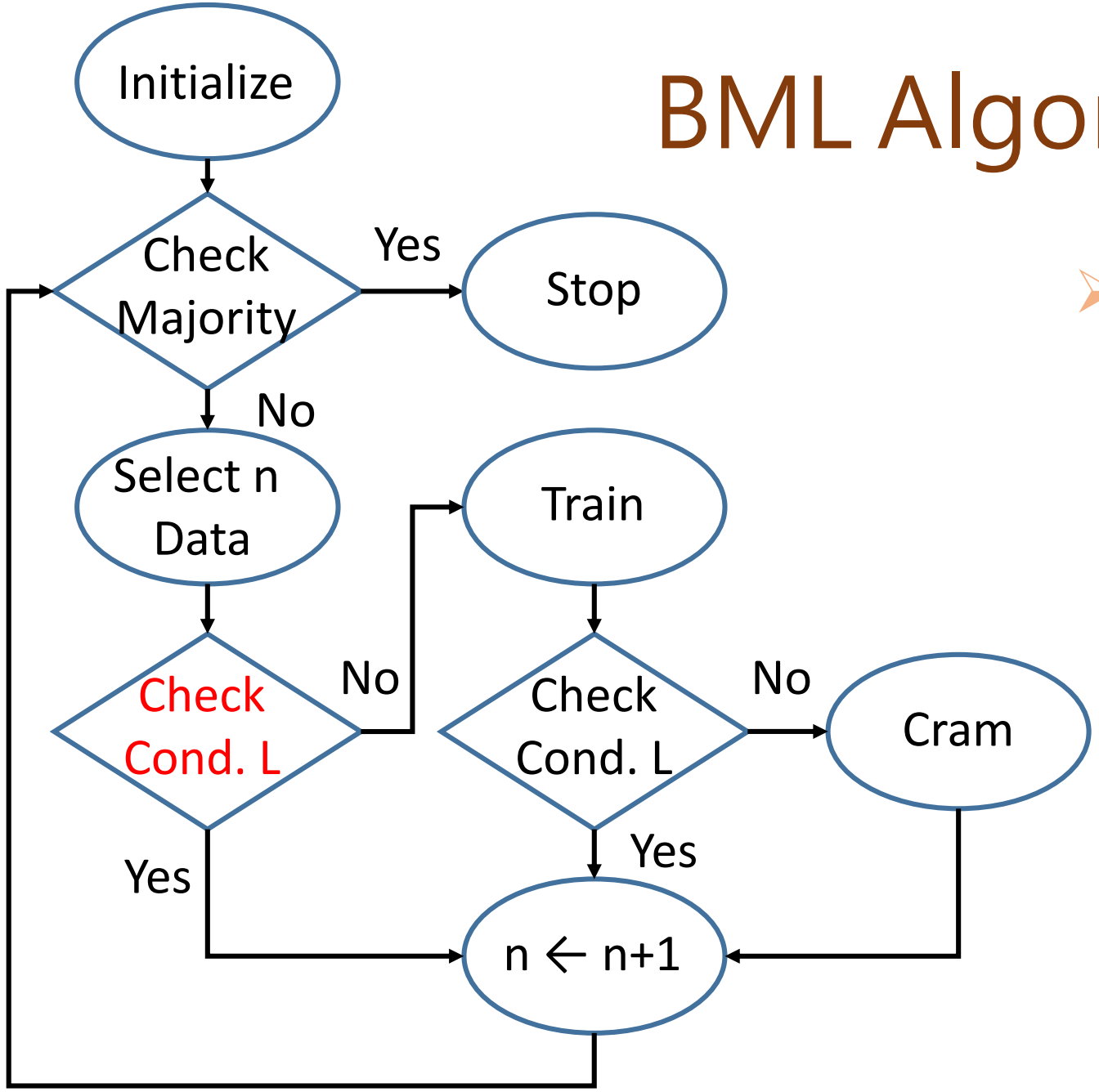
- Step 2: Check Majority
  - Check if n > γN.
  - If yes, the majority of data can fit an acceptable SLFN, stop learning process.
  - If no, go on training process.

BML Algorithm (cont.)



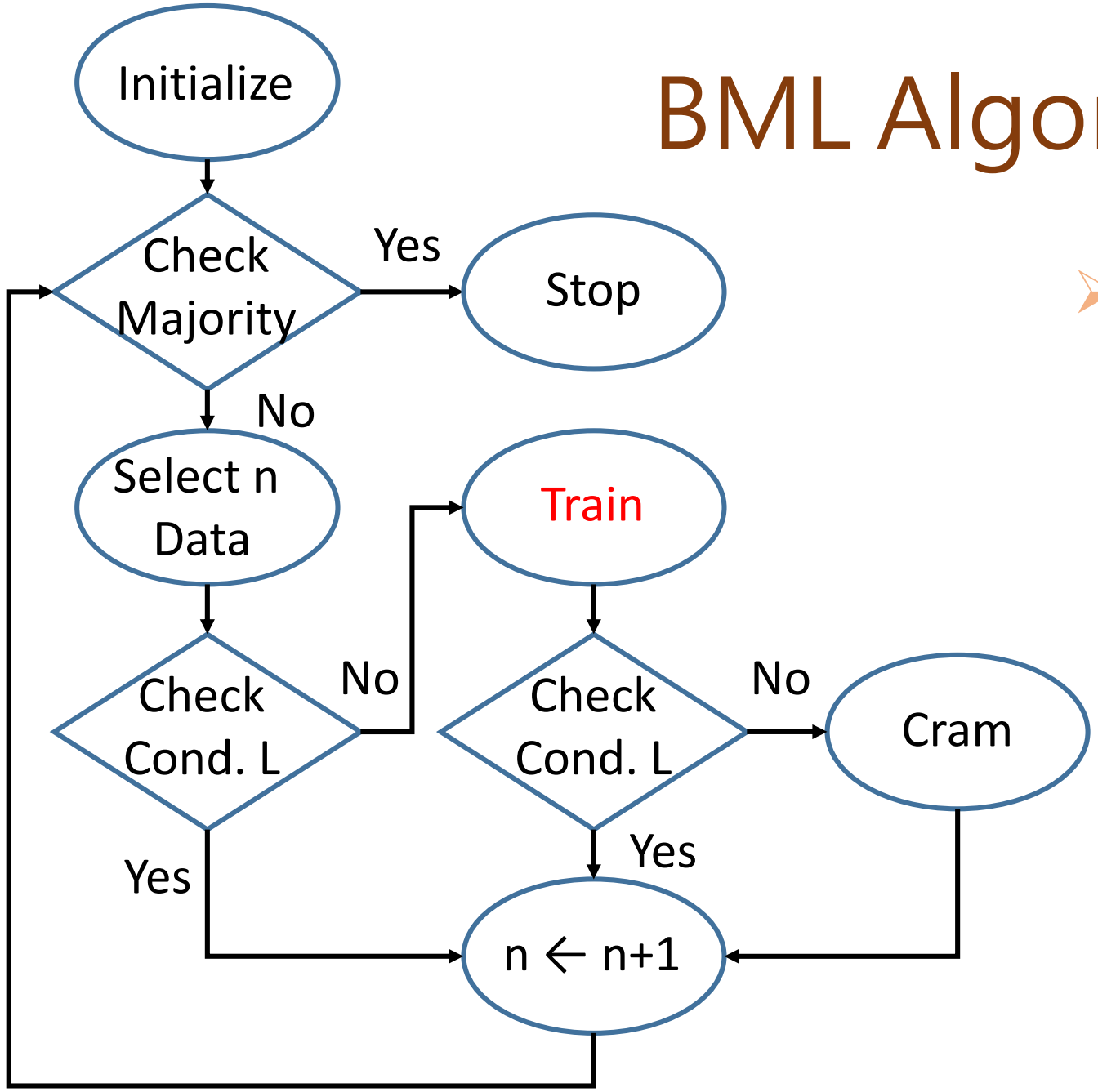
- Step 3: Select n Data
  - Base on the current SLFN, select n data, (x^c, y^c), that have the largest distances between C_1 and C_2.

BML Algorithm (cont.)



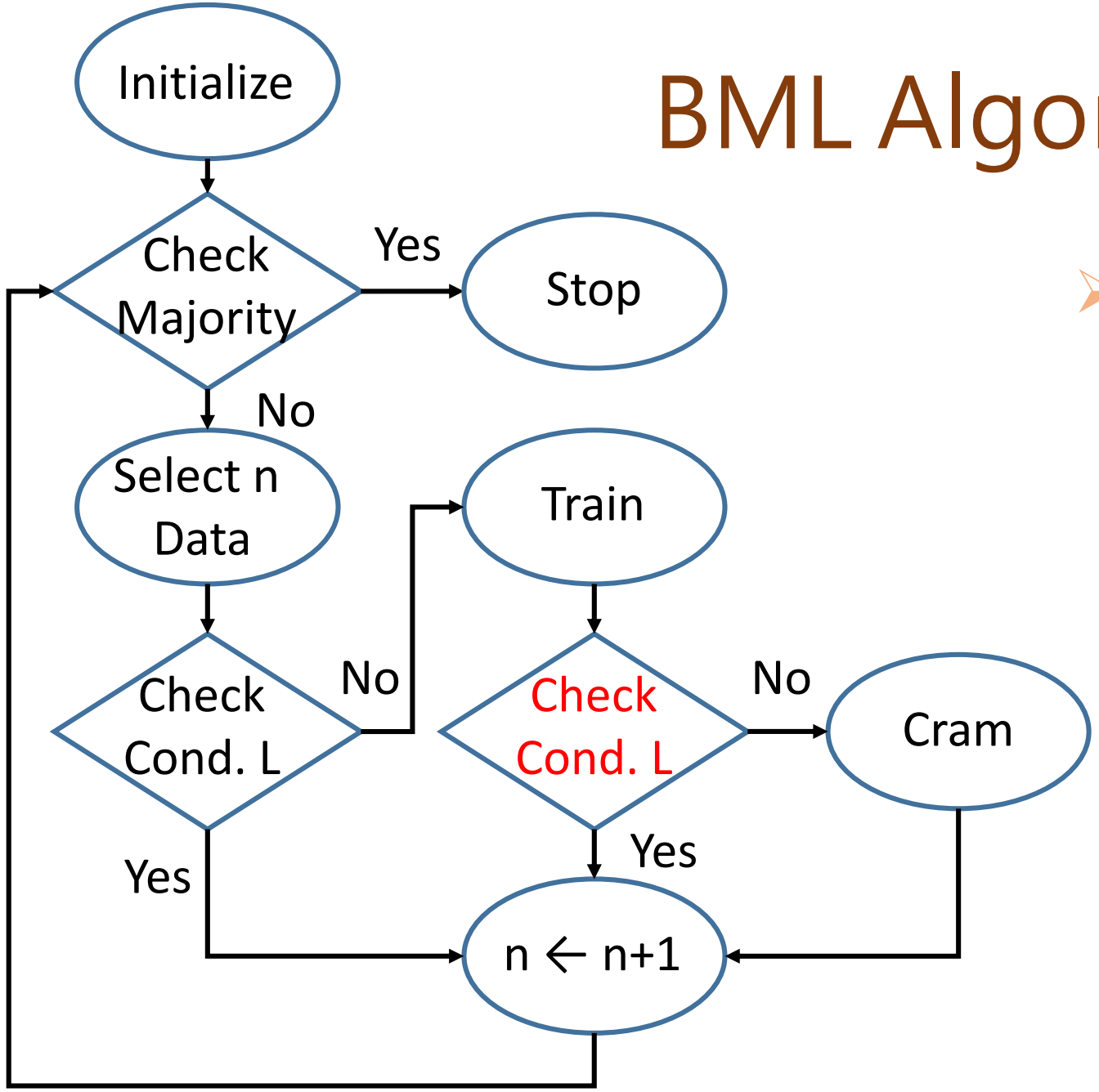
- Step 4: Check Condition L
  - Check if current n selected data satisfy the condition L under current SLFN.
  - If yes, n ← n+1, go to next stage.
  - If no, retrain the SLFN by n selected data.

BML Algorithm (cont.)



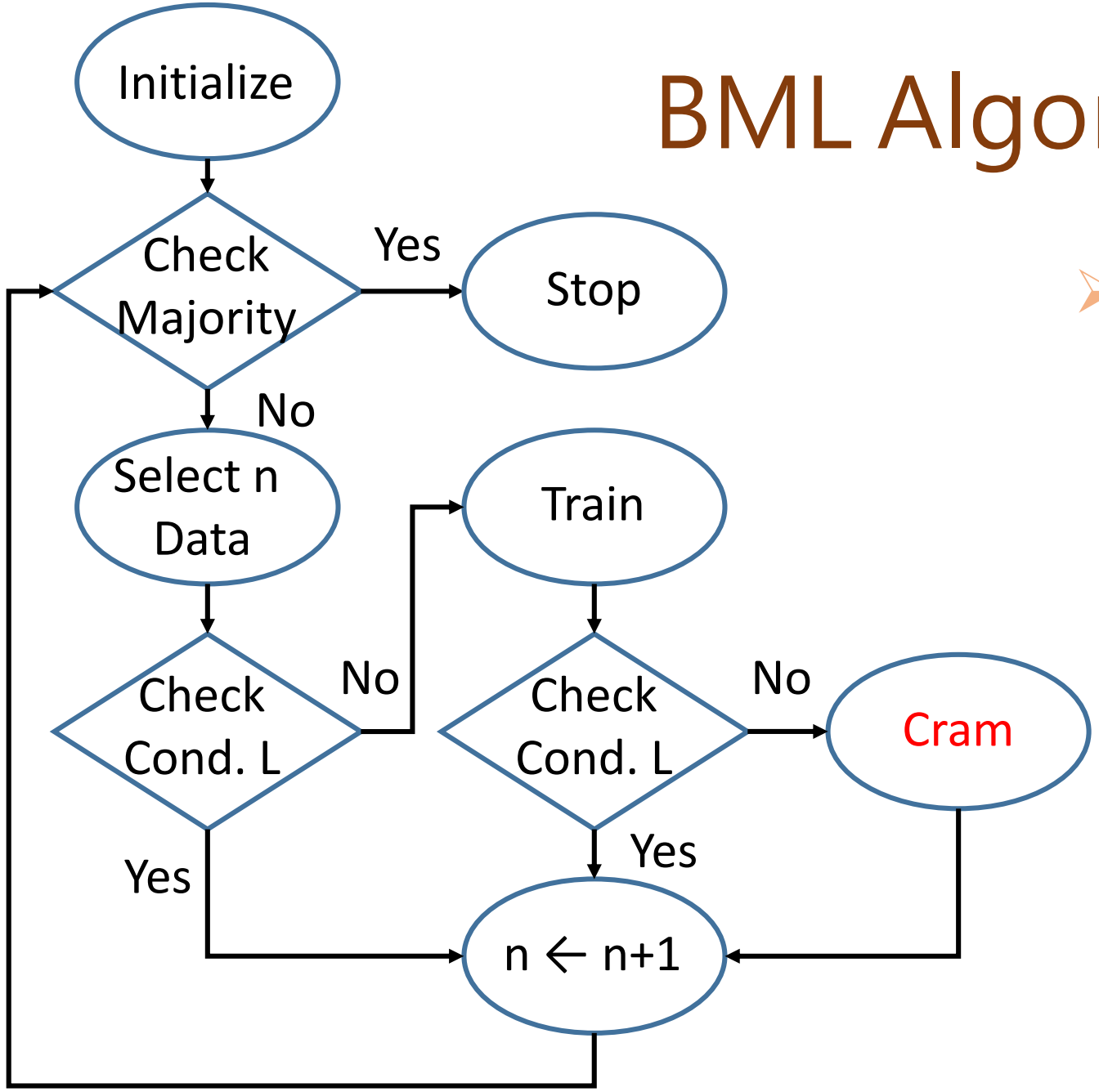
- Step 5: Train
  - Apply the gradient descent algorithm to adjust weights.

BML Algorithm (cont.)



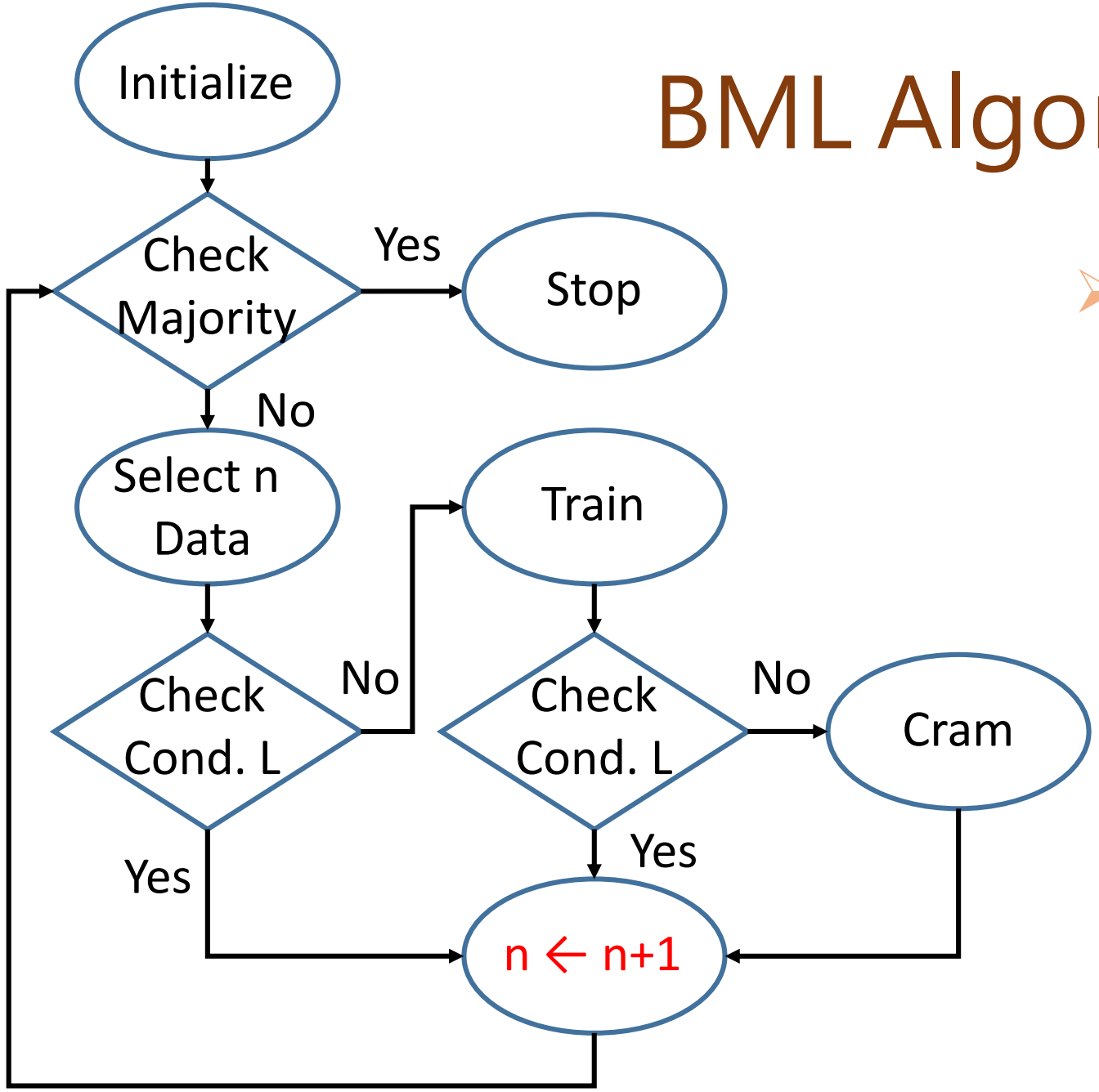
- Step 6: Check Condition L
  - After several times of tuning weights, check if current n data satisfy the condition L under current SLFN.
  - If yes, n ← n+1, go to next stage.
  - If no, apply resistant approach.

BML Algorithm (cont.)



- Step 7: Cram
  - Apply the cramming mechanism in resistant learning by adding extra hidden nodes to obtain an acceptable SLFN.

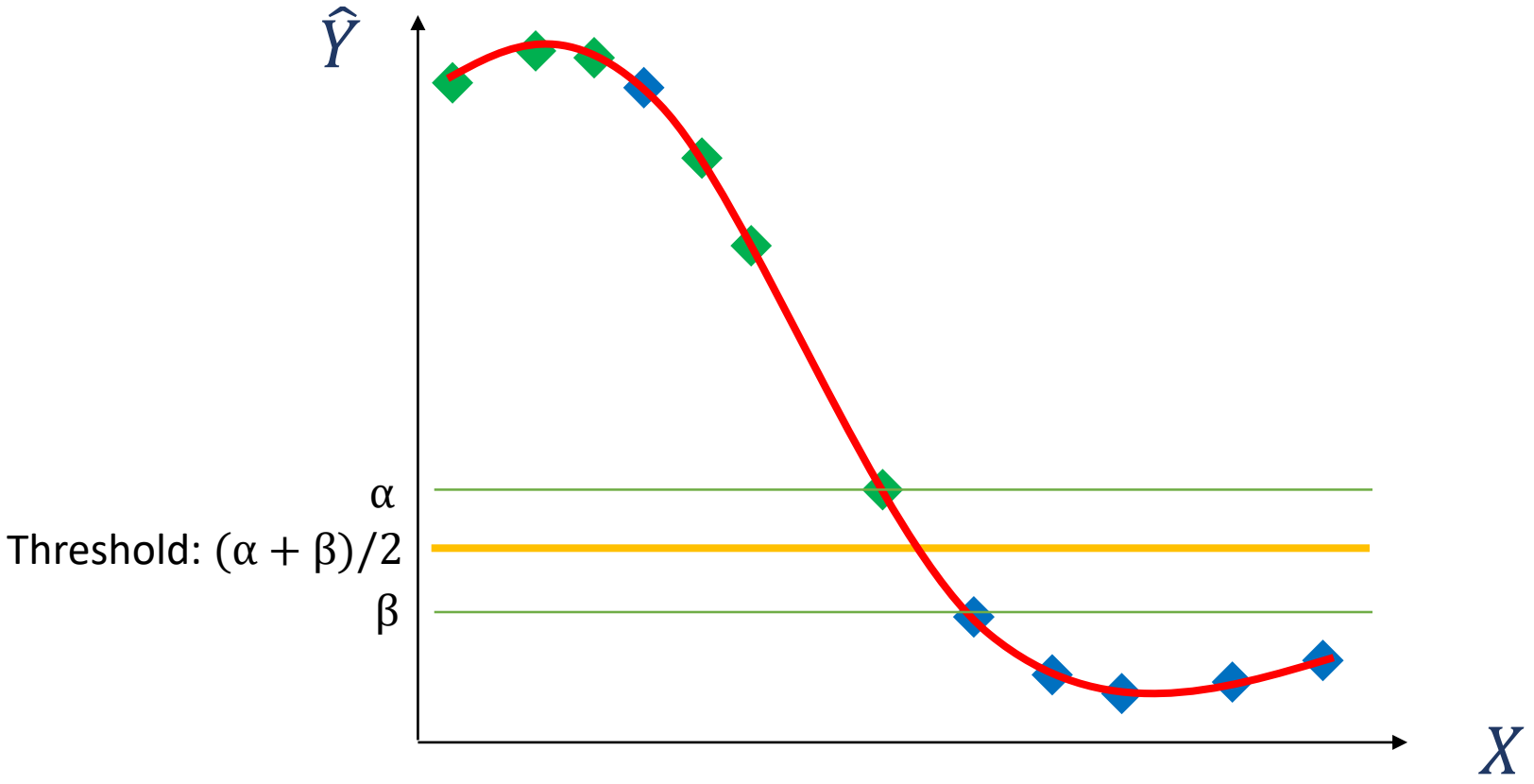
BML Algorithm (cont.)



- Step 8: n ← n+1
  - Gradually increase the amount of selected data by 1.

Bipartite Majority Learning

- Find a fitting function for majority.
- Classify data by threshold.

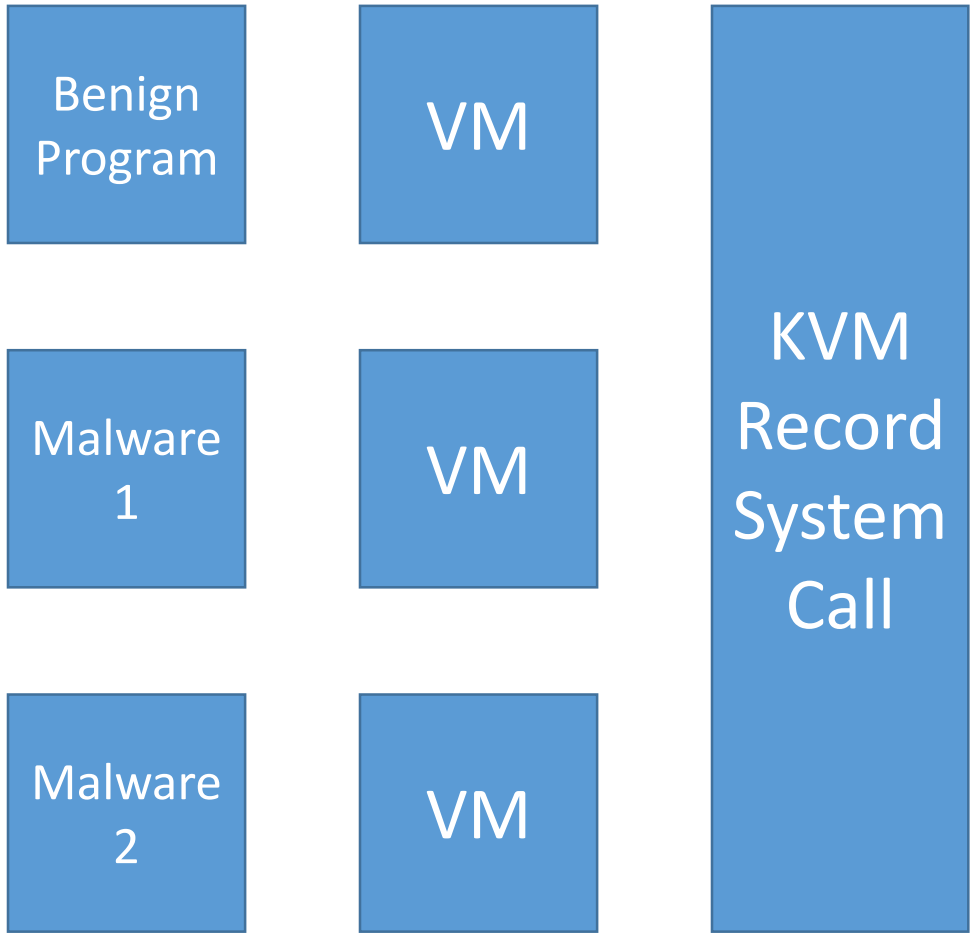


Evaluation

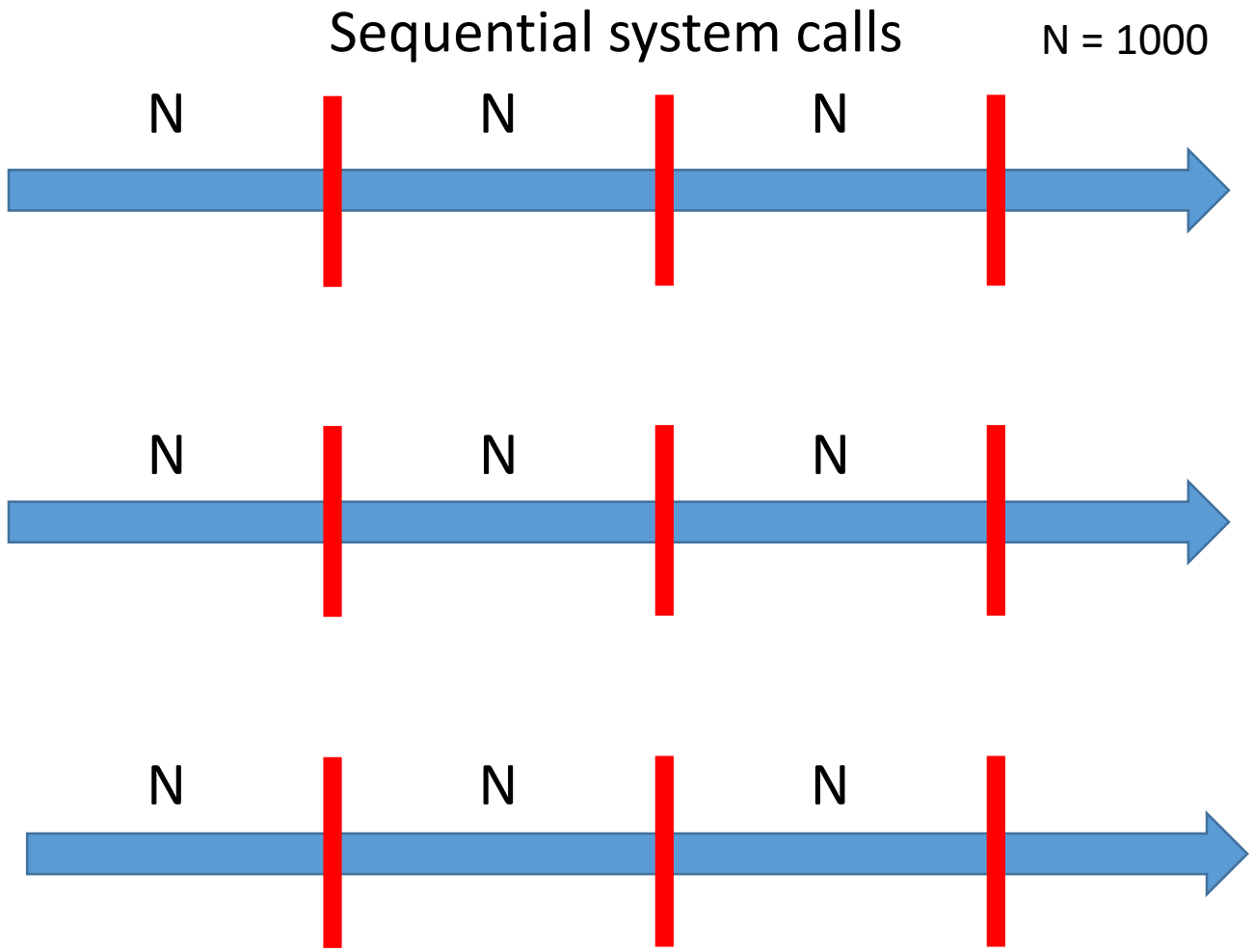
- Problem: Classify malware and benign execution behavior pattern.
- Dataset: Labeled system call sequences.
- 52 types of system calls.

Sample	Read	Write	Ioctl	…
Malware1_1	5	3	7	…
Malware1_2	2	7	6	…
Malware1_3	4	6	5	…
⋮	⋮	⋮	⋮	…

Generate System Call Sequences

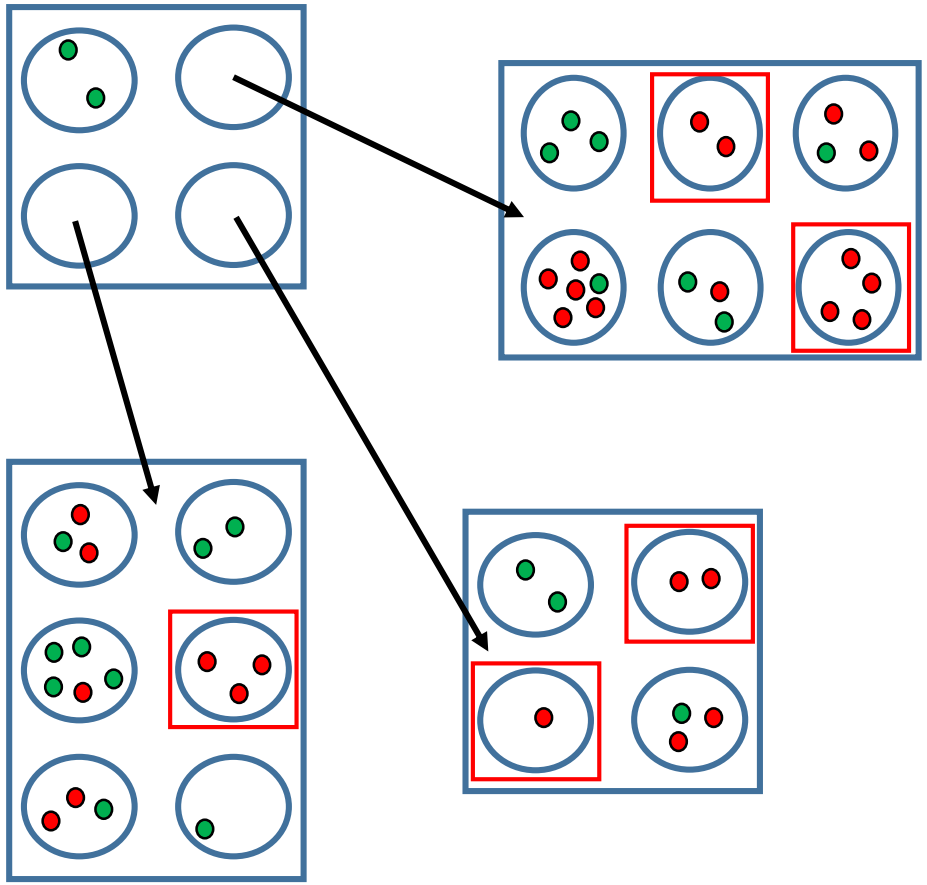


MalID_Period	Read	Write	Ioctl
Malware1_1	5	3	7
Malware1_2	2	7	6



Reference: Chiu, C. H., Chen, J. J., & Yu, F. (2017, June). An Effective Distributed GHSOM Algorithm for Unsupervised Clustering on Big Data. In Big Data (BigData Congress), 2017 IEEE International Congress on (pp. 297-304). IEEE.

Clustered by GHSOM and Label Data



Clusters only contain malware samples
→Unique characteristics of malware
→Detection rules

Reference: Chiu, C. H., Chen, J. J., & Yu, F. (2017, June). An Effective Distributed GHSOM Algorithm for Unsupervised Clustering on Big Data. In Big Data (BigData Congress), 2017 IEEE International Congress on (pp. 297-304). IEEE.

Data Preprocessing

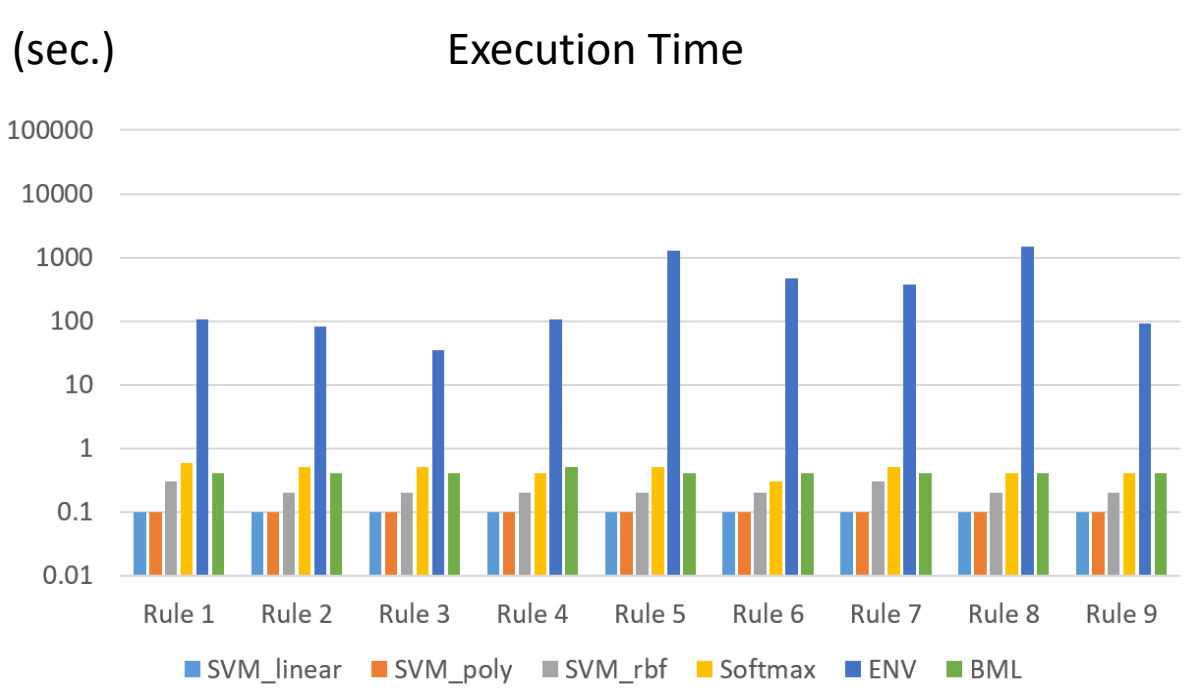
- Labeled by the Growing Hierarchical Self-Organizing Map(GHSOM, Chiu, 2017) clustering result, 19 types of malware behavior was identified.
- Filter out duplicate behavior characteristics.
- Filter out clusters with few data (under 1,000 samples).
- 9 detection rules are preserved.

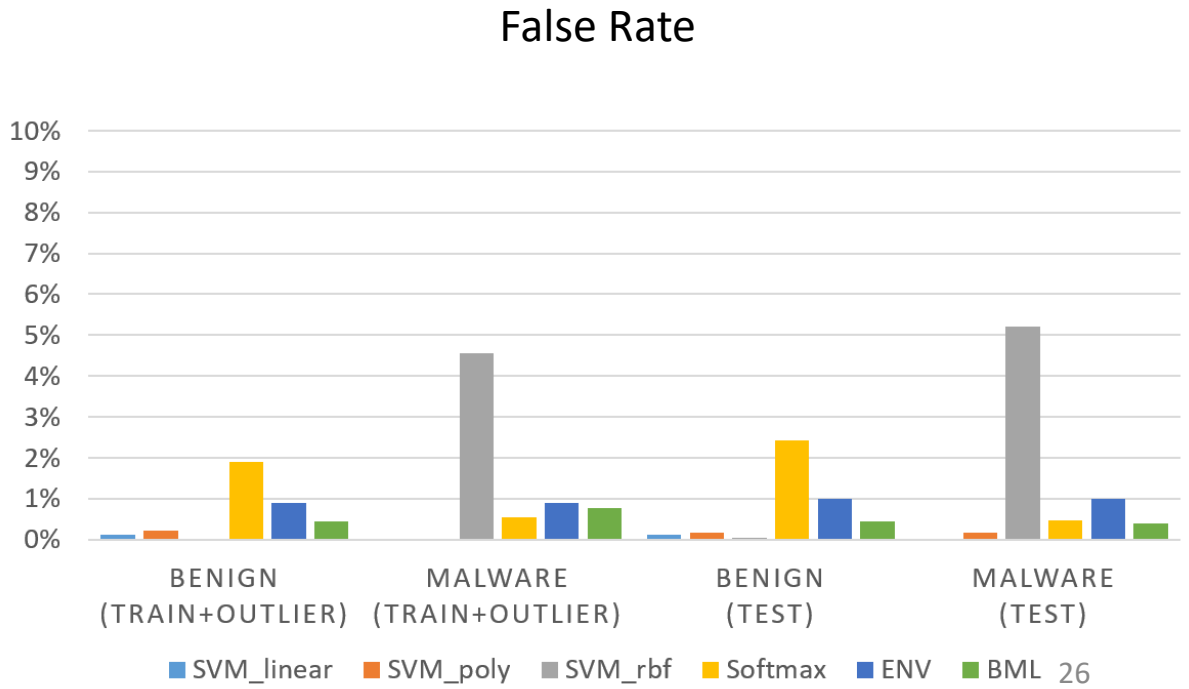
Rule 1.	3175	Rule 2.	1331	Rule 3.	2025	Rule 4.	1838	Rule 5.	2451
Rule 6.	4356	Rule 7.	1208	Rule 8.	1220	Rule 9.	1787	Benign	19391

Reference: Chiu, C. H., Chen, J. J., & Yu, F. (2017, June). An Effective Distributed GHSOM Algorithm for Unsupervised Clustering on Big Data. In Big Data (BigData Congress), 2017 IEEE International Congress on (pp. 297-304). IEEE.

RQ1: How much do we improve training efficiency on small size dataset?

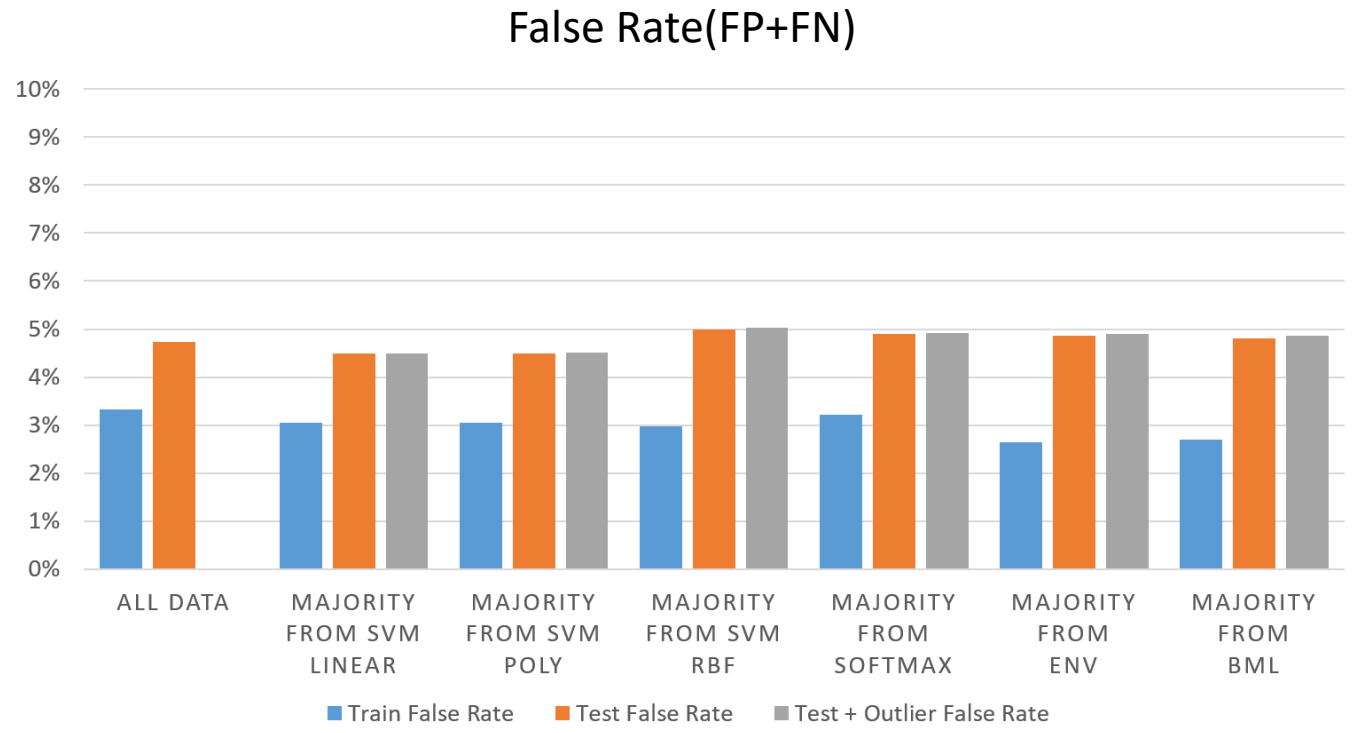
- Randomly select 100 samples from a rule cluster and randomly select 100 samples from 19,391 benign samples.
- Combine the 200 samples to be the training data.





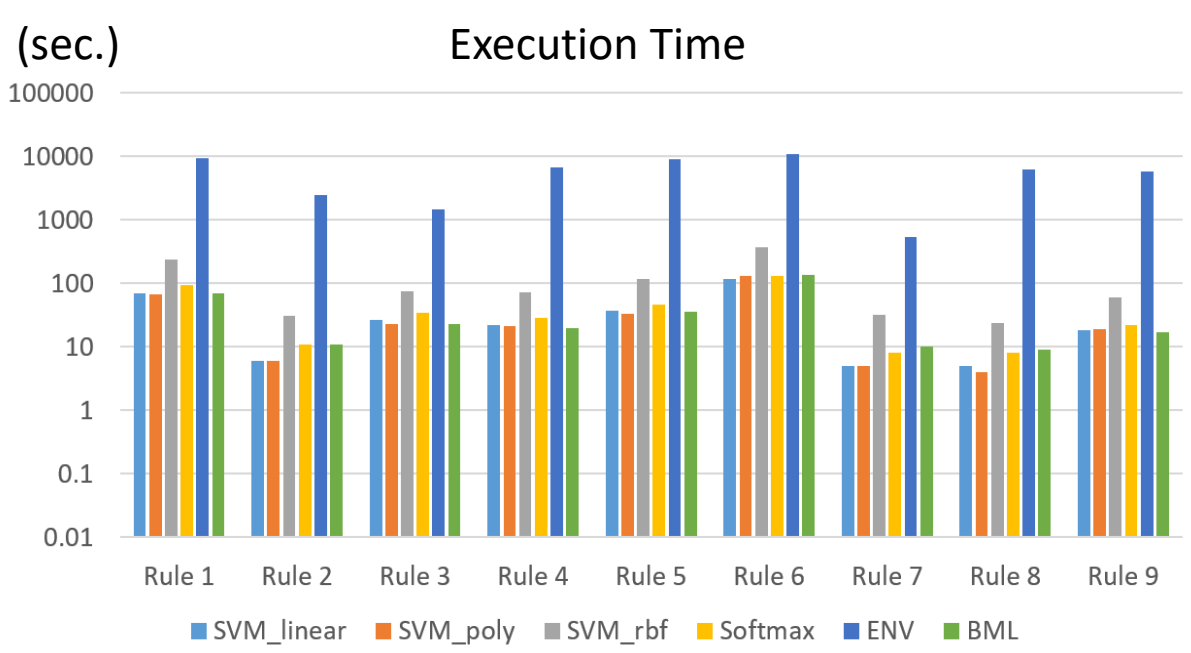
RQ2: Do we lose accuracy with majority selected by BML on small size dataset?

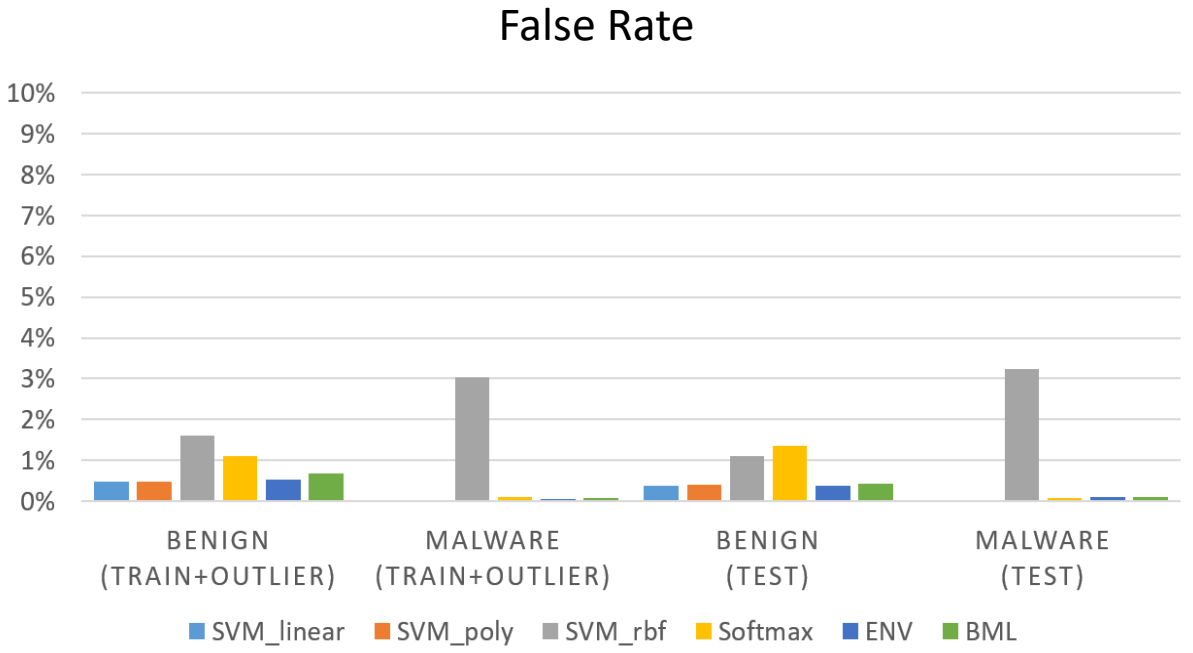
- Obtain the majority data selected by the majority learning methods in RQ1 to be training data.
- Use softmax neural network to conduct multi-class learning.



RQ3: How much do we improve training efficiency on large size dataset?

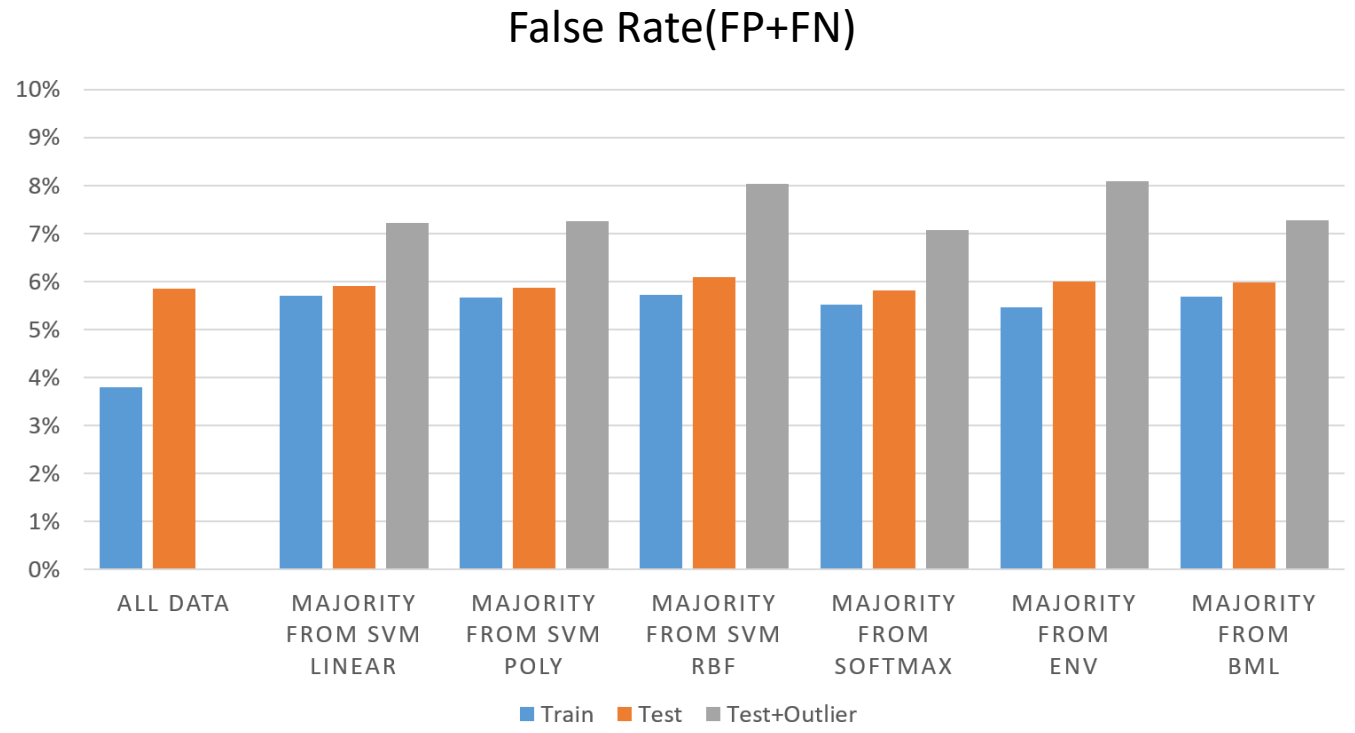
- Randomly select 80% samples from a rule cluster and randomly select equal amount of samples from 19,391 benign samples.
- Combine the selected samples to be the training data.





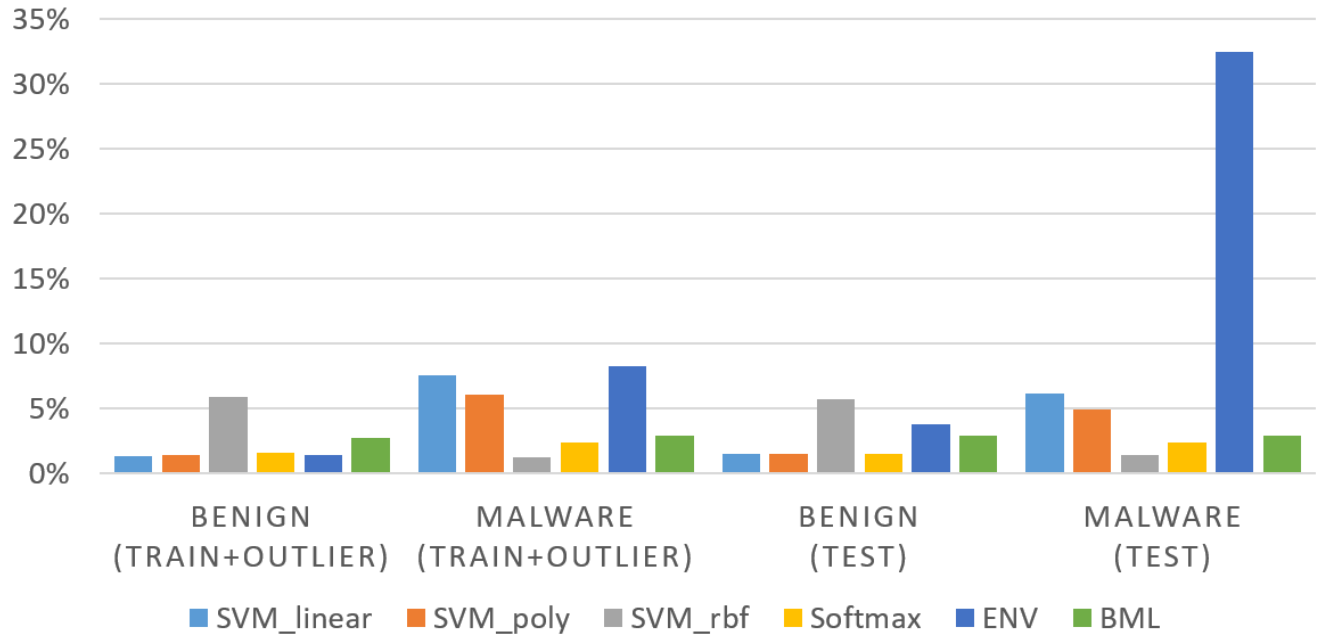
RQ4: Do we lose accuracy with majority selected by BML on large size dataset?

- Obtain the majority data selected by the majority learning methods in RQ3 to be training data.
- Use softmax neural network to conduct multi-class learning.



RQ5: How BML performs on high-variation datasets?

- Regard 9 rules as malware class to increase the variety of malware samples. Randomly select equal number of malware and benign samples. Combine the selected samples to be the training data.
- Three different sampling method: 200, 10%, 80%



Conclusion

- We introduce a novel nominal resistant learning procedure BML to avoid anomalies affecting the effectiveness of learning.
- Through the proposed majority selecting method, we are able to spot anomalies in a global view when the features of anomalies are unknown.
- The experiments on real-world datasets show that BML still possesses high detection accuracy and has significantly improved performance on malware classification.

Thank you

Hyper Parameters

- In machine learning, hyper parameters is a parameter whose value is set before the learning process begins.
- Fixed learning rate Ɛ: 0.01
  - Ɛ set as 0.01 is a proper step size for weight tuning.
- Fixed hidden layer: 1 layer
  - BML algorism only need one layer for complex input-output mappings.
- Max weight tuning times per stage: 10,000
  - Loss value usually stop decreasing after optimizing hundreds times.
- Majority rate γ: 0.95
  - Our experiment data set is easier for classifying into two classes, so we only need to avoid few outliers.