

# Supervised Learning

## Meeting 8/15

Bo Han, Chen

National Yang Ming Chiao Tung University, Taiwan

*bhchen312551074.cs12@nycu.edu.tw*

August 15, 2023

# Presentation Overview

- ① Environment
- ② Public Dataset
  - Oxford 102 Flower
  - NSL-KDD
- ③ Self-Made Dataset
  - MLB
- ④ Conclusion
- ⑤ References
- ⑥ Q & A

# Environment

- Python 3.10.12
  - PyTorch 2.0.1
  - Scikit-learn 1.2.2
- platform: Google Colab
  - GPU: Nvidia Tesla T4

# Public Dataset

## Oxford 102 Flower

- preprocessing
  - split the train, valid, test data (1020, 1020, 6149)
  - resize the image data (224\*224)
  - normalization
- hyperparameters
  - batch size: 100
  - epochs: 50
- classifier
  - VGG 19
  - ResNet
    - residual-block

# Oxford 102 Flower

## Result

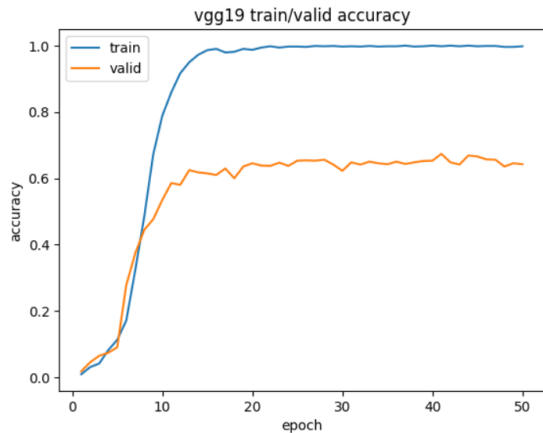


Figure 1: VGG training progress-accuracy

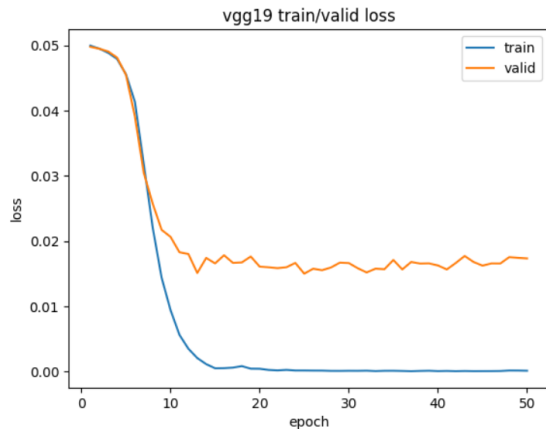


Figure 2: VGG training progress-loss

# Oxford 102 Flower

## Result

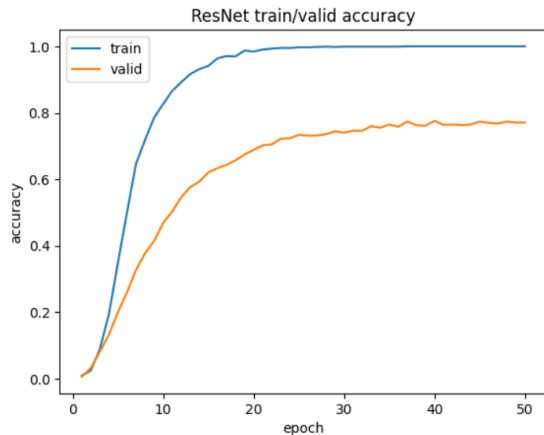


Figure 3: ResNet training progress-accuracy

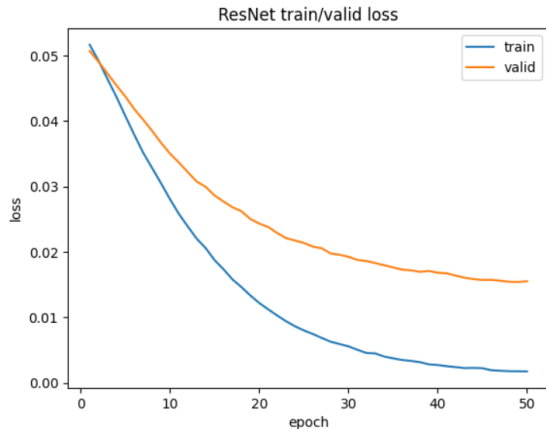


Figure 4: ResNet training progress-loss

# Oxford 102 Flower

## Result

| Model         | Accuracy |
|---------------|----------|
| myVGG         | 59.55%   |
| myResNet      | 75.34%   |
| ResNet50 [1]  | 90.00%   |
| EffNet-L2 [2] | 99.65%   |

- preprocessing
  - one-hot encoding
  - min-max normalization
  - mapping the attack class
    - normal
    - Dos
    - Probe
    - R2L
    - U2R
  - data split (125973, 22544)
- classifier
  - KNN
  - SVM



# NSL-KDD

## Result

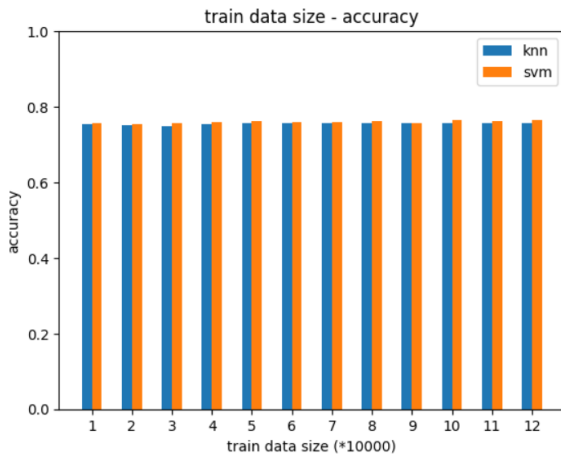


Figure 5: train data size-accuracy on NSL-KDD

# NSL-KDD

## Result

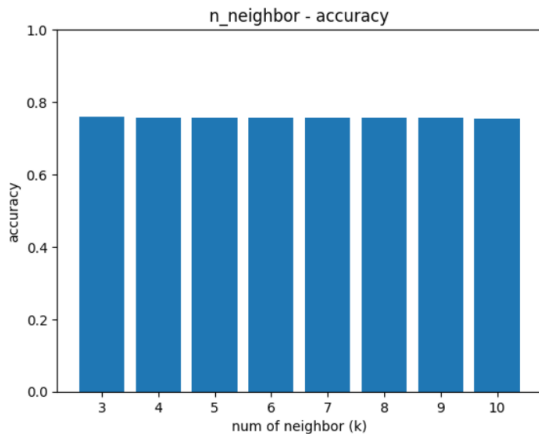


Figure 6: KNN neighbor-accuracy

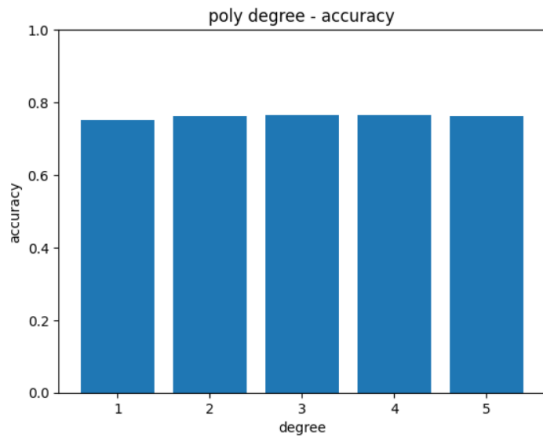


Figure 7: SVM degree-accuracy

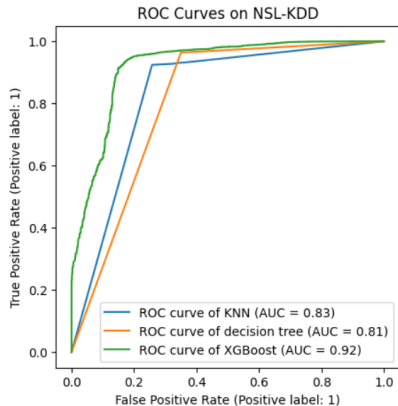


Figure 8: ROC Curves on NSL-KDD

| Model   | Accuracy |
|---------|----------|
| myKNN   | 75.71%   |
| mySVM   | 76.44%   |
| ANN [3] | 79.90%   |
| CNN [4] | 80.13%   |

# Self-Made Dataset

MLB

- preprocessing
  - min-max normalization
  - reorganize the class label
- classifier
  - Decision Tree
  - XGBoost
- cross-validation
  - Stratified K Fold with  $K=5$
  - 60 instances a fold

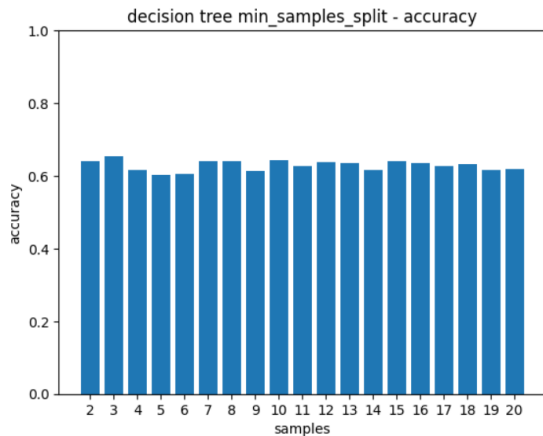


Figure 9: Decision tree split sample-accuracy

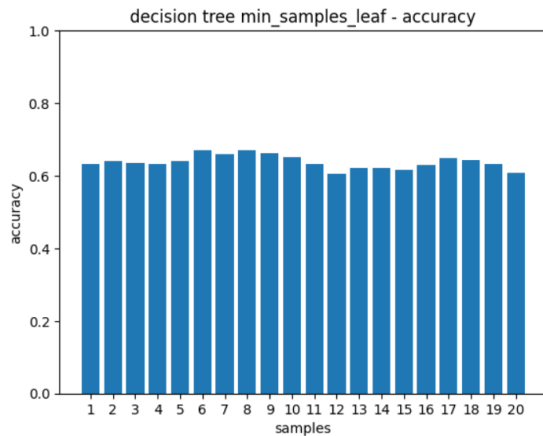


Figure 10: Decision tree leaf sample-accuracy

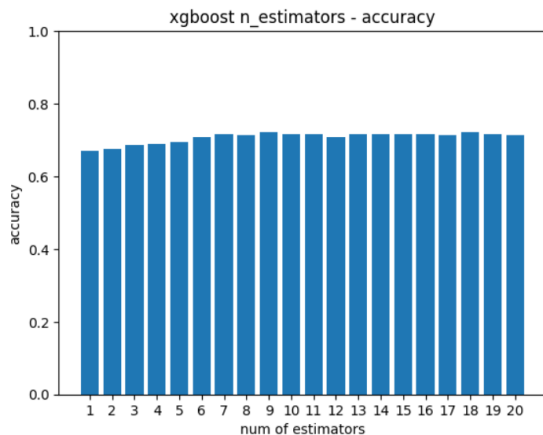


Figure 11: XGBoost estimator-accuracy

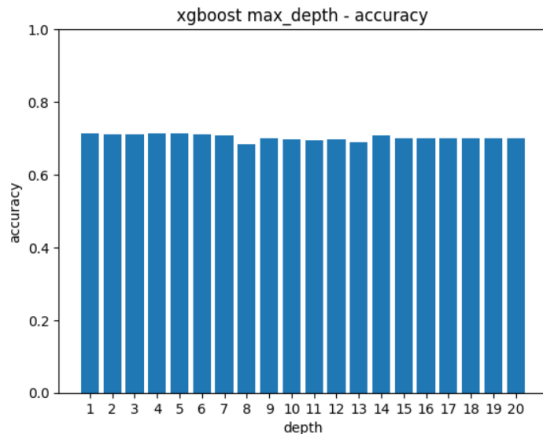


Figure 12: XGBoost max depth-accuracy

# Conclusion

- more efficient way for hyperparameter tuning
  - ex: evolutionary algorithm
- try different preprocessing method
  - ex: dimensionality reduction



- [1] Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. “When vision transformers outperform resnets without pre-training or strong data augmentations”. In: *arXiv preprint arXiv:2106.01548* (2021).
- [2] Pierre Foret et al. “Sharpness-aware minimization for efficiently improving generalization”. In: *arXiv preprint arXiv:2010.01412* (2020).
- [3] Bhupendra Ingre and Anamika Yadav. “Performance analysis of NSL-KDD dataset using ANN”. In: *2015 international conference on signal processing and communication engineering systems*. IEEE. 2015, pp. 92–96.
- [4] Yalei Ding and Yuqing Zhai. “Intrusion detection system for NSL-KDD dataset using convolutional neural networks”. In: *Proceedings of the 2018 2nd International conference on computer science and artificial intelligence*. 2018, pp. 81–85.

Thanks for Listening

Q & A