Weekly Report

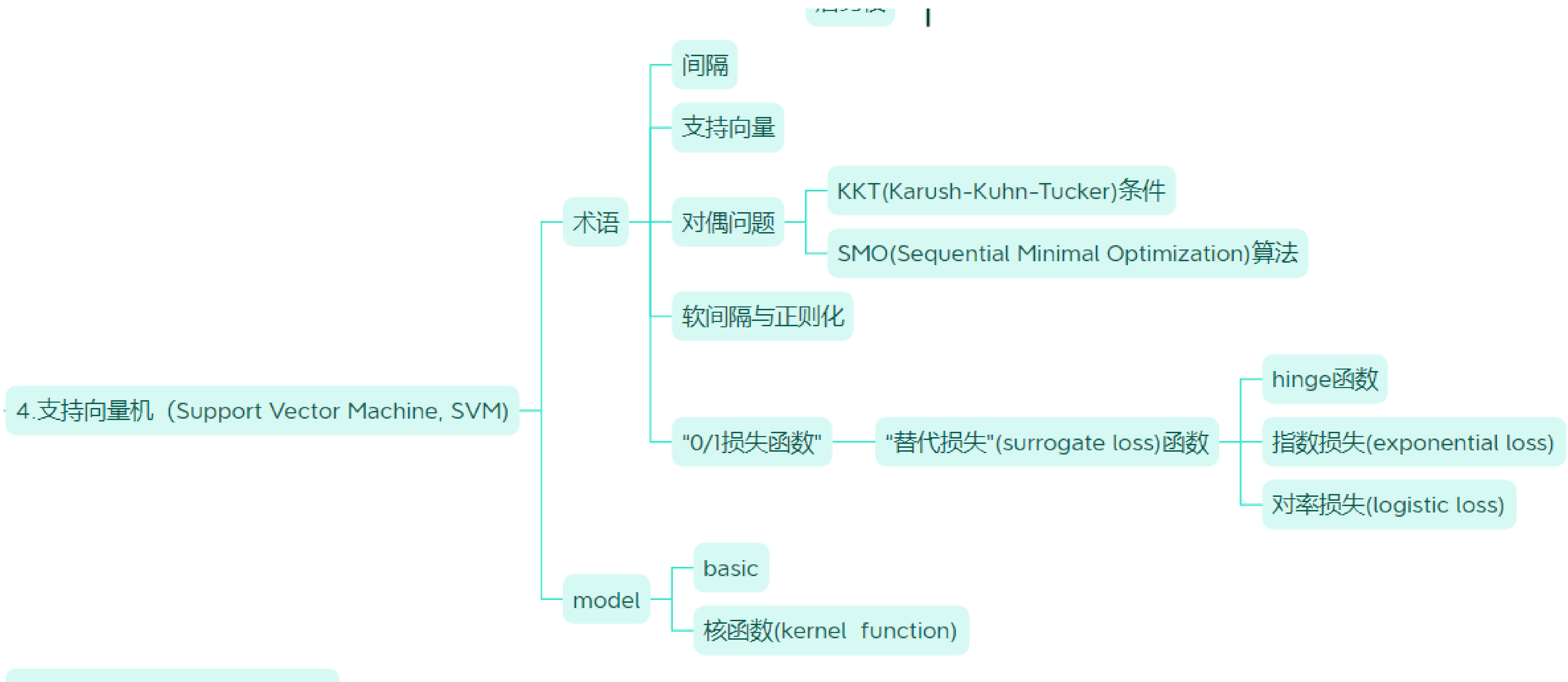
Name: Xiaodan Li

Time: 2024/8/19 – 2024/8/24

What work I have done this week (lists and details)

- Learn and practice SVM
- Paper reading

- SVM's theory



间隔
支持向量
KKT(Karush-Kuhn-Tucker)条件
术语
对偶问题
SMO(Sequential Minimal Optimization)算法
软间隔与正则化
hinge函数
4.支持向量机（Support Vector Machine, SVM)
"0/1损失函数"
"替代损失"(surrogate loss)函数
指数损失(exponential loss)
对率损失(logistic loss)
basic
model
核函数(kernel function)

- Practice SVM (iris_data)

```python
# SpotCheck Algorithms
models = []
models.append(('SVM_linear', SVC(kernel='linear', gamma='auto')))
models.append(('SVM_poly', SVC(kernel='poly', gamma='auto')))
models.append(('SVM_rbf', SVC(kernel='rbf', gamma='auto')))
models.append(('SVM_sigmoid', SVC(kernel='sigmoid', gamma='auto')))
```

```
G:\Pragram_Trainning\machine_learning>python iris_svm.py
SVM_linear: 0.975000 (0.038188)
SVM_poly: 0.958333 (0.055902)
SVM_rbf: 0.983333 (0.033333)
SVM_sigmoid: 0.366667 (0.040825)

G:\Pragram_Trainning\machine_learning>python iris_svm.py
validation_size = 0.25, seed = 1
SVM_linear: 0.973485 (0.040550)
SVM_poly: 0.955303 (0.044748)
SVM_rbf: 0.964394 (0.043658)
SVM_sigmoid: 0.366667 (0.031637)

G:\Pragram_Trainning\machine_learning>python iris_svm.py
validation_size = 0.2, seed = 1

SVM_linear: 0.975000 (0.038188)
SVM_poly: 0.958333 (0.055902)
SVM_rbf: 0.983333 (0.033333)
SVM_sigmoid: 0.366667 (0.040825)

G:\Pragram_Trainning\machine_learning>python iris_svm.py
validation_size = 0.5, seed = 1

SVM_linear: 0.975000 (0.050000)
SVM_poly: 0.923214 (0.101031)
SVM_rbf: 0.975000 (0.050000)
SVM_sigmoid: 0.158929 (0.128633)

G:\Pragram_Trainning\machine_learning>python iris_svm.py
validation_size = 0.1, seed = 1

SVM_linear: 0.978022 (0.033602)
SVM_poly: 0.963187 (0.048777)
SVM_rbf: 0.970879 (0.047903)

G:\Pragram_Trainning\machine_learning>python iris_svm.py
validation_size = 0.2, seed = 7

SVM_linear: 0.991667 (0.025000)
SVM_poly: 0.966667 (0.055277)
SVM_rbf: 0.991667 (0.025000)
```

- CICIDS2017(DT algorithm result)

命令提示符

```
G:\20240708_small_tasks>python flow_data_ddos.py
(225745, 79)
 Label
BENIGN     97718
DDoS      128027
dtype: int64
newdataset.shape = (225741, 79)
newdataset.shape = (225711, 79)
validation_size = 0.2, seed = 7, n_splits = 10

CART: 0.999767 (0.000135)
\CART Prediction
0.9996234189132313
[[19674     5]
 [   12 25452]]
              precision    recall  f1-score   support

      BENIGN       1.00      1.00      1.00     19679
        DDoS       1.00      1.00      1.00     25464

    accuracy                           1.00     45143
   macro avg       1.00      1.00      1.00     45143
weighted avg       1.00      1.00      1.00     45143


G:\20240708_small_tasks>
```

Paper Reading

Name: Xiaodan Li

Time: 2024/8/23 – 2024/8/24

Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset

Authors: Ziadoon Kamil Maseer; Robiah Yusof; Nazrulazhar Bahaman; Salama A. Mostafa;Cik Feresa Mohd Foozy

Affiliation: Malaysia Melaka University & Malaysia Tun Hussein University

From: 2021 IEEE ACCESS

https://ieeexplore.ieee.org/document/9345704

Research Question (What is the problem)

- 1.Which machine learning algorithm is best for an anomaly-based intrusion detection system(AIDS);
- 2.How to evaluate and compare the performance of different AIDS models;
- 3. How to build a standardized benchmarking method to test and evaluate ML and DL-based AIDS models?

Motivation (With the existing work, why the author do this work)

- 1.Address the gaps in existing research: Some issues in related work, including the randomness of the selected algorithms, parameters, and testing criteria, the application of old datasets, or shallow analyses and validation of the results.
- 2.Comprehensively evaluate model performance
- 3.Optimize model efficiency
- 4.Establish a standardized assessment methodology

Challenge (What is the difficulty of this work)

- The limitations of datasets: the datasets are extremely imbalanced in terms of cybersecurity, with the majority (98%) of these datasets being classied as normal, whereas the rest (2%) are classied as attacks.

Methodology (Architecture, solutions or methods, how to do this work)

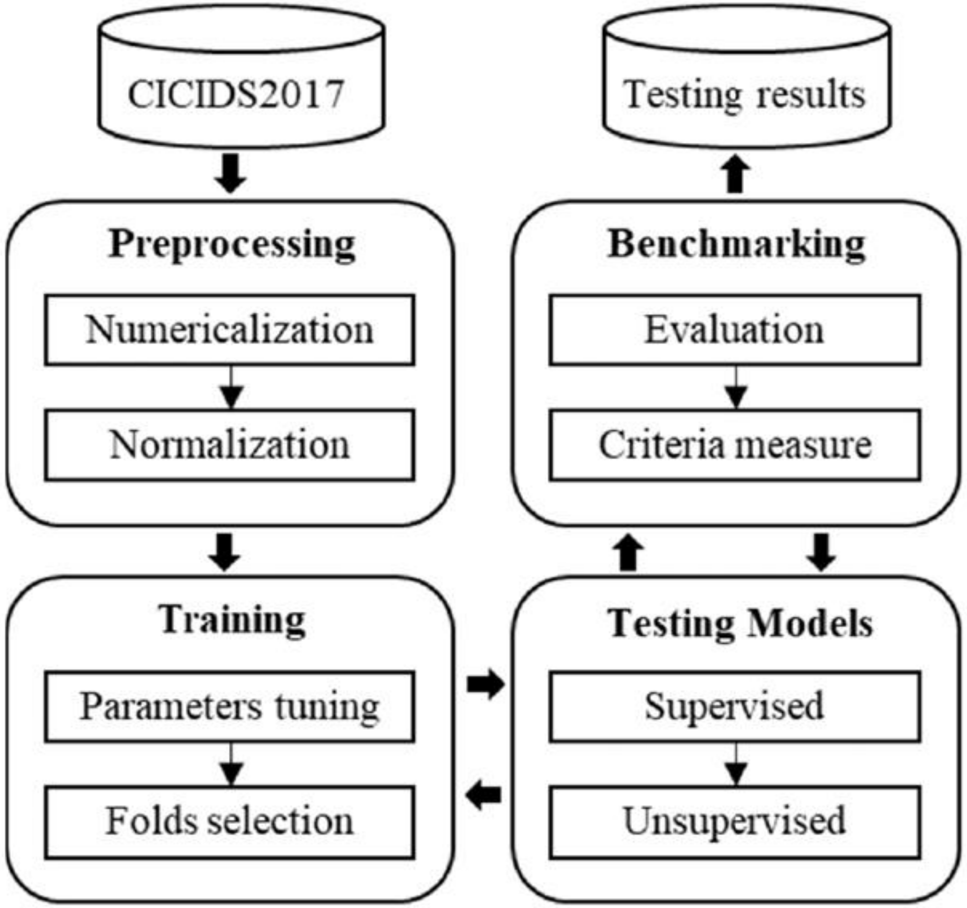


FIGURE 8. The benchmarking methodology of ML-AIDS.

Results (Data & charts or other results, effective and value of the work)

TABLE 7. Performance evaluation results for the DT algorithm.

Model Setting	Class label	Accuracy	Precision	Recall	F1-Score	T1 (s)	T2 (s)
criterion = 'gini', max depth=4	C1	1.00	1.00	1.00	1.00	1.13	0.73
	C2	0.86	0.67	0.86	0.76		
	C3	0.03	0.88	0.03	0.06		
	C4	0.00	0.00	0.00	0.00		
criterion = 'gini', max depth = 6	C1	1.00	1.00	1.00	1.00	3.33	1.68
	C2	0.78	0.72	0.78	0.75		
	C3	0.32	0.45	0.32	0.38		
	C4	0.00	0.00	0.00	0.00		
criterion = gini, max depth = None, class weight = balanced	C1	1.00	1.00	1.00	1.00	1.85	1.33
	C2	0.73	0.73	0.72	0.72		
	C3	0.44	0.41	0.43	0.42		
	C4	0.44	0.57	0.44	0.50		
criterion = entropy, max depth=4	C1	1.00	1.00	1.00	1.00	0.90	0.75
	C2	0.90	0.64	0.90	0.75		
	C3	0.00	0.00	0.00	0.00		
	C4	0.00	0.00	0.00	0.00		
criterion = entropy, max depth = 6	C1	1.00	1.00	1.00	1.00	1.85	0.79
	C2	0.87	0.70	0.88	0.78		
	C3	0.17	0.47	0.17	0.25		
	C4	0.11	0.40	0.22	0.29		
criterion = entropy, max depth = None, class weight = balanced	C1	1.00	1.00	1.00	1.00	1.23	1.12
	C2	0.74	0.73	0.74	0.73		
	C3	0.37	0.38	0.37	0.38		
	C4	0.67	0.67	0.67	0.67		

TABLE 16. Overall performance of the ML-AIDS algorithms.

Algorithm	Accuracy	Precision	Recall	F1-Score	Accuracy SD	T1(s)	T2(s)
ANN	0.9928	0.9937	0.9928	0.9917	0.1233	53.78	48.03
DT	0.9949	0.9943	0.9949	0.9942	0.1363	1.23	1.12
k-NN	0.9952	0.9949	0.9952	0.9949	0.1473	11.13	7.92
NB	0.9886	0.9901	0.9886	0.9885	0.2324	1.07	0.15
RF	0.9930	0.9909	0.9930	0.9912	0.1110	9.38	6.76
SVM	0.7521	0.9916	0.7521	0.7660	0.3084	343.56	33.17
CNN	0.9947	0.9943	0.9946	0.9944	0.4936	261.80	1.73
k-means	0.2559	0.9747	0.2559	0.3996	1.0127	3.12	2.99
EM	0.6006	0.8688	0.6006	0.7411	1.0968	11.19	9.69
SOM	0.5906	0.8588	0.6000	0.7411	1.1096	120.27	0.05
[59]	*	0.96	0.96	0.96	*	*	*
[60]	0.84	*	*	*	*	*	*
[61]	*	0.95	0.98	0.96	*	*	*
[62]	0.98	99.52	98.68	92.76	*	*	*
[63]	*	0.34	0.50	0.74	*	*	*

Evaluation (What do you think about this work? Make some challenge)

TABLE 2. Details of the CICIDS2017 dataset.

Name of Files	Class Found
Monday-Hours.pcap_ISCX.csv	Benign (Normal human activities)
Tuesday-Hours.pcap_ISCX.csv	Benign, FTP-Patator,SSH Patator
Wednesday-.pcap_ISCX.csv	Benign, DoS GoldenEye, DoSHulk, DoS lowhttptest, DoS slow loris, Heartbleed
Thursday-WebAttacks.pcap_ ISCX.csv	Benign, Brute Force, SQL Injection, XSS.
Thursday-Infilteration.pcap_.csv	Benign, Infiltration
Friday-pcap_ISCX.csv	Benign, Bot
Friday-PortScan.pcap_ISCX.csv	Benign, PortScan
Friday- DDos.pcap_ISCX. csv	Benign, DDoS

TABLE 5. Classes in the CICIDS2017 testing dataset.

#	Class name	Class label	Support
1	BENIGN	C1	53518
2	Brute Force	C2	482
3	XSS	C3	210
4	SQL Injection	C4	9

Plan for next (week)

- Learn and practice the algorithms referred in the paper, including KNN, ANN