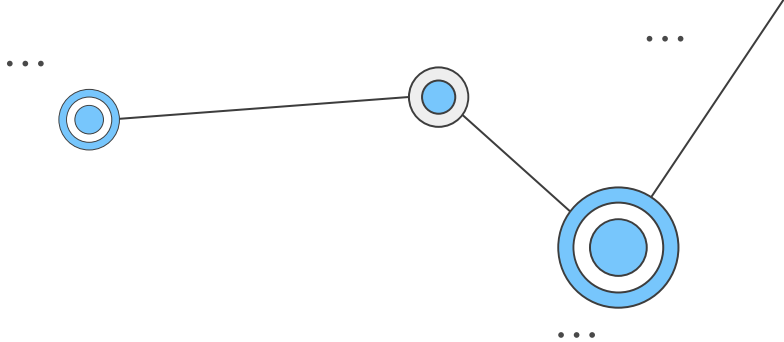


Análise Exploratória de Ataques por meio de Instruções em Código de Máquina em Automóveis Inteligentes

Bruno H. Labres
Ovidio J S Junior

Contexto

01

IoT em veículos inteligentes

CAN (Controller Area Network)

02

Crescimento

Área está em ascensão

03

Fomento da pesquisa de segurança na área

Geração de dados na área

Instruções CAN (Controller Area Network)

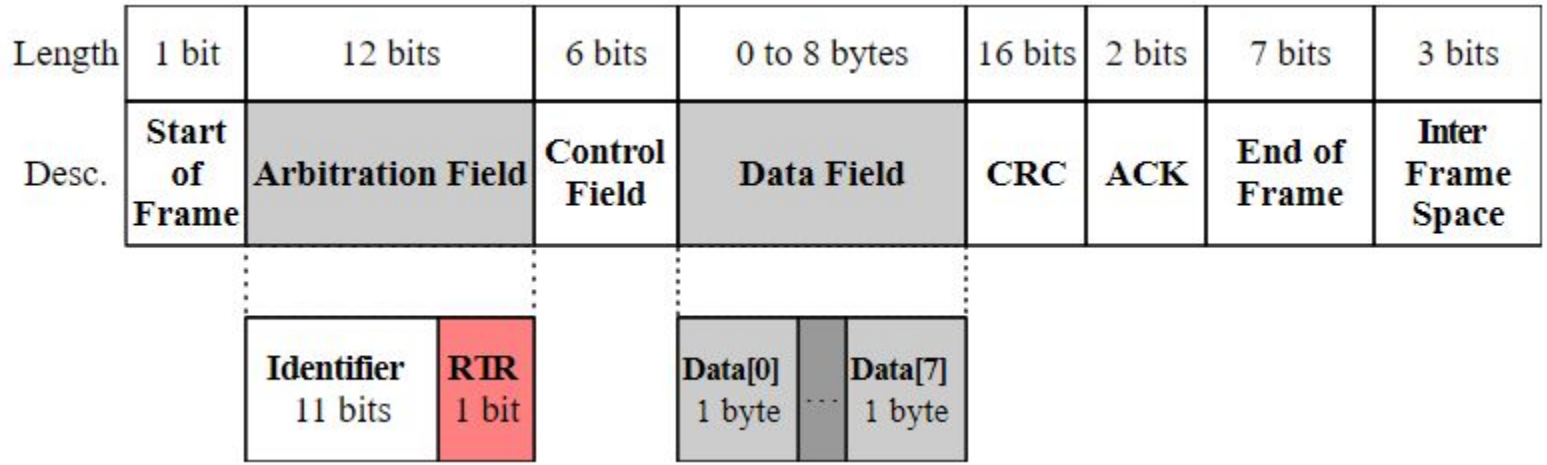


Figura: Instrução Padrão CAN utilizado como exemplo no experimento.

Instruções CAN (Controller Area Network)

	timestamp	can_id	dlc	data0	data1	data2	data3	data4	data5	data6	data7	type
1	1.478196e+09	0545	8	d8	00	00	8a	00	00	00	00	0
2	1.478196e+09	0002	8	00	00	00	00	00	01	07	15	0
3	1.478196e+09	0153	8	00	21	10	ff	00	ff	00	00	0
4	1.478196e+09	0130	8	19	80	00	ff	fe	7f	07	60	0
5	1.478196e+09	0131	8	17	80	00	00	65	7f	07	9f	0
6	1.478196e+09	0140	8	00	00	00	00	02	20	27	a8	0
7	1.478196e+09	0350	8	05	20	14	68	78	00	00	21	0
8	1.478196e+09	02c0	8	15	00	00	00	00	00	00	00	0
9	1.478196e+09	0370	8	00	20	00	00	00	00	00	00	0
10	1.478196e+09	043f	8	10	40	60	ff	7d	8c	09	00	0

	timestamp	can_id	dlc	data0	data1	data2	data3	data4	data5	data6	data7	type
1	1.478196e+09	01cd	8	d0	21	ec	f7	6a	d2	da	6e	1
2	1.478196e+09	0378	8	45	de	26	09	f8	48	11	51	1
3	1.478196e+09	01e2	8	48	20	7d	e8	62	34	61	7b	1
4	1.478196e+09	034e	8	73	9e	b9	77	13	e0	e5	23	1
5	1.478196e+09	0108	8	b9	48	4b	24	a0	35	8f	27	1
6	1.478196e+09	04e8	8	e7	23	3a	fa	6d	34	f8	8b	1
7	1.478196e+09	05e1	8	f0	51	41	f2	69	c2	ac	a5	1
8	1.478196e+09	0102	8	2b	ca	a4	da	e3	42	40	f0	1
9	1.478196e+09	007c	8	2e	19	7b	ea	ed	46	ae	cf	1
10	1.478196e+09	0051	8	6f	5e	40	04	e7	ae	d7	b3	1

Figura: Amostras de instruções do dataset com exemplos benignos e maliciosos, respectivamente.

Instruções CAN (Controller Area Network)

Round	Type	Description	# Normal	# Attack	# Rows (Total)
Preliminary	Training	Normal and four types of attacks dataset with class	3,372,743	299,408	3,672,151
	Submission	Normal and four types of attacks dataset with class (during the competition, without class)	3,358,210	393,836	3,752,046
Final	Submission	Normal and five attacks (4 spoofings, 1 fuzzing) dataset with class (during the competition, without class)	1,090,312	179,998	1,270,310

Figura: Tabela de divisão de instruções normais e ataques.

Metodologia

01 Análise dos dados

02 Pré-processamento

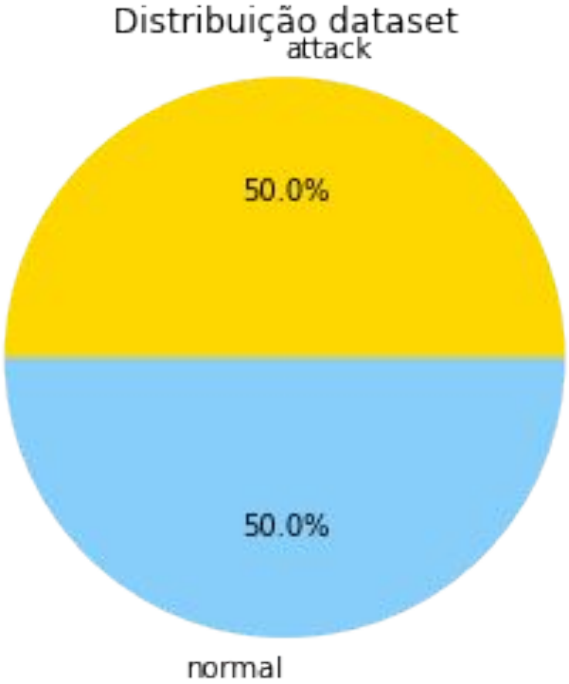
03 Extração de características

04 Treinamento e teste

Metodologia

- Coleta e pré-processamento: CSV -> *dataframes*; Amostragem de ataques; Transformações de dados.
- Extração de características: Campos de *data* da instrução CAN.
- Treinamento e testes: *random forest*, *multilayer perceptron*, *K-Nearest Neighbors*. Uso de GridSearch para testes de parâmetros. Validação cruzada com cinco pastas.

Distribuição de classes



Treinamento e Testes

Random Forest

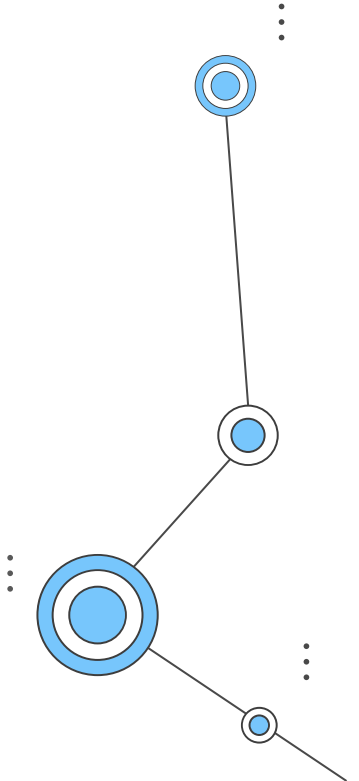
Criação de árvores de decisão

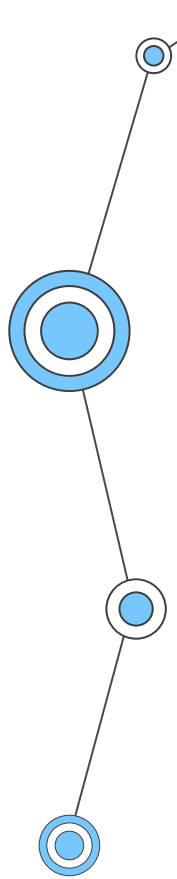
K-Nearest Neighbors

Vizinhos mais próximos

Multilayer perceptron

Neuronios com pesos





Parâmetros

```
param_grid_rf = {
    "max_depth": [1,2],
    "min_samples_split": [2,3,4],
    "min_samples_leaf": [2,3,4]
}
```

A lista completa de parâmetros utilizados (escolhidos manualmente ou com a ajuda do GridSearch) é:

- n_estimators: 100
- criterion: gini
- max_depth: 2
- min_samples_split: 2
- min_samples_leaf: 2
- min_weight_fraction_leaf: 0.0
- max_features: auto
- max_leaf_nodes: None
- min_impurity_decrease: 0.0
- bootstrap: True
- oob_score: True
- n_jobs: None
- random_state: None
- verbose: 0
- warm_start: None
- class_weight: None
- ccp_alpha: 0.0
- max_samples=None

```
param_grid_knn = {
    "n_neighbors": [2,3,5,7,9],
    "weights": ['uniform','distance'],
    "algorithm": ['ball_tree', 'kd_tree', 'brute']
}
```

A lista completa de parâmetros utilizados (escolhidos manualmente ou com a ajuda do GridSearch) é:

- n_neighbors: 9
- weights: uniform
- algorithm: ball_tree
- leaf_size: 30
- p: 2
- metric: minkowski
- metric_params: None
- n_jobs: None

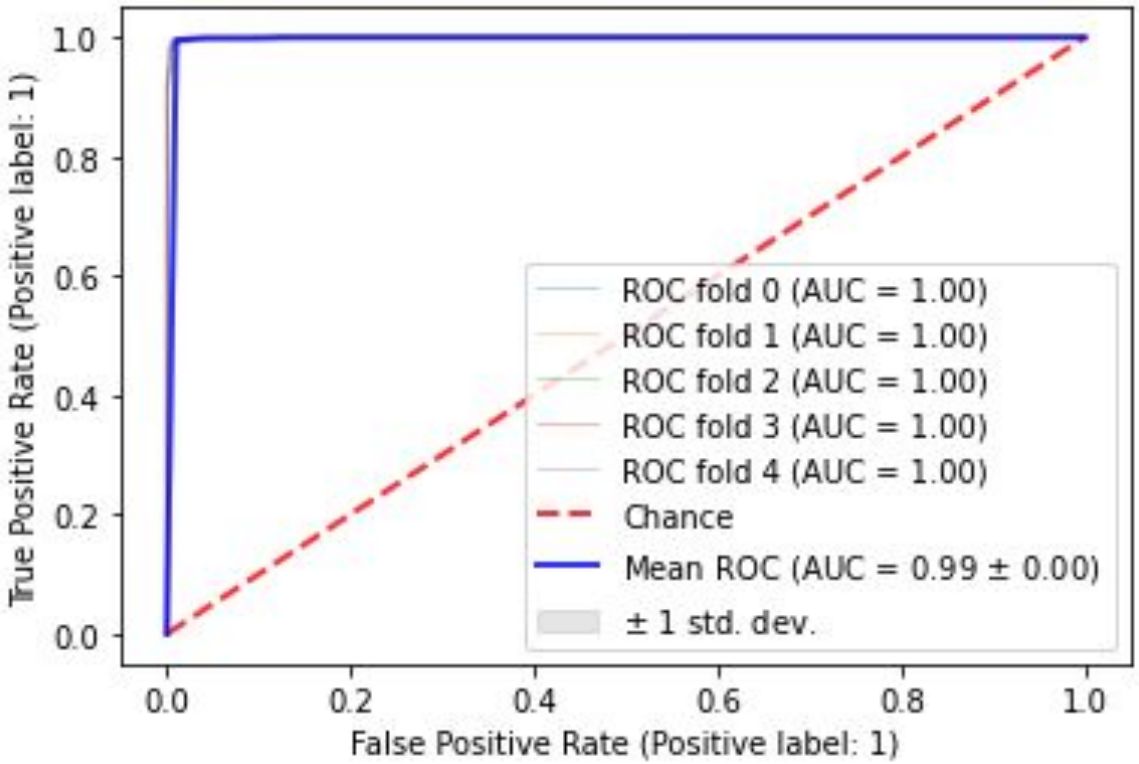
Parâmetros

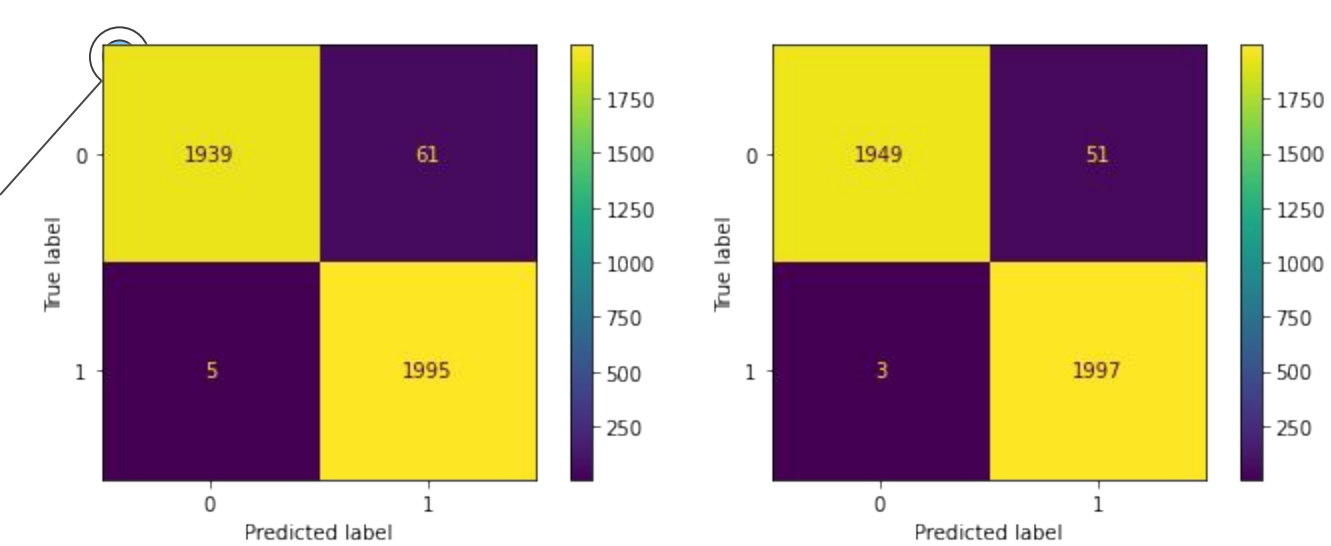
```
param_grid_mlp = {
    'hidden_layer_sizes': [(10,10,10), (10,10), (10,)],
    'activation': ['logistic', 'tanh'],
}{'activation': 'logistic', 'hidden_layer_sizes': (10,)}
```

A lista completa de parâmetros utilizados (escolhidos manualmente ou com a ajuda do GridSearch) é:

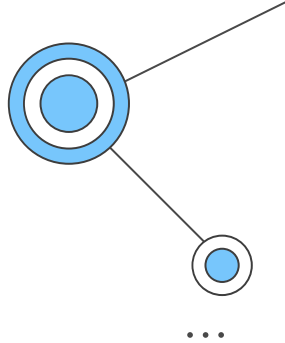
- hidden_layer_sizes: (10,)
- activation: logistic
- solver: adam
- alpha: 0.0001
- batch_size: auto
- learning_rate: constant
- learning_rate_init: 0.001
- power_t: 0.5
- max_it: 200
- shuffle: True
- random_state: None
- tol: 1e-4
- verbose: False
- warm_start: False
- momentum: 0.9
- nesterovs_momentum: True
- early_stopping: False
- validation_fraction: 0.1
- beta_1: 0.9
- beta_2: 0.999
- epsilon: 1e-8
- n_iter_no_change: 10
- max_fun: 15000

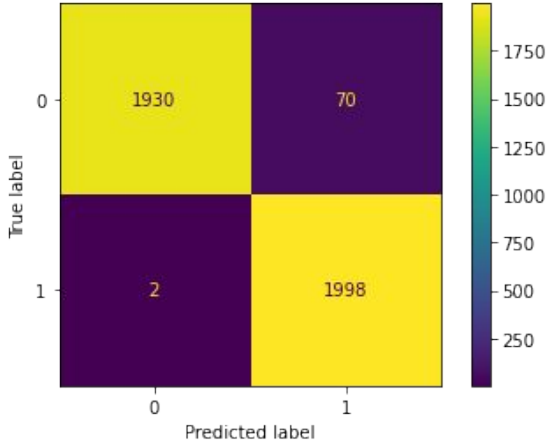
Random forest

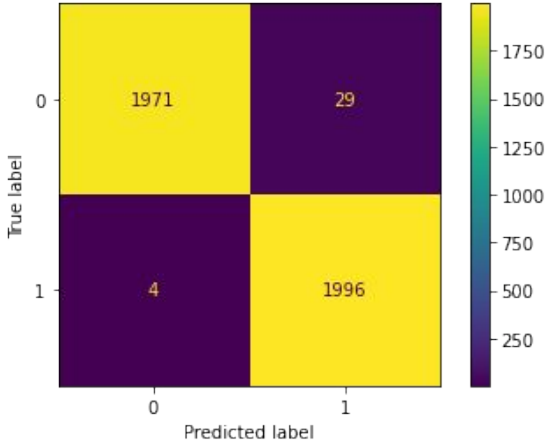


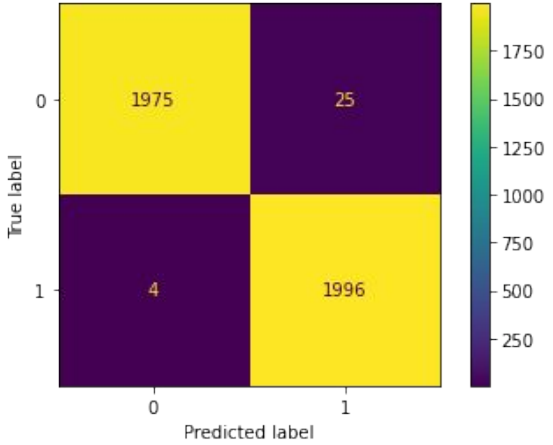

1939
61
5
1995
0
1
True label
Predicted label
1750
1500
1250
1000
750
500
250

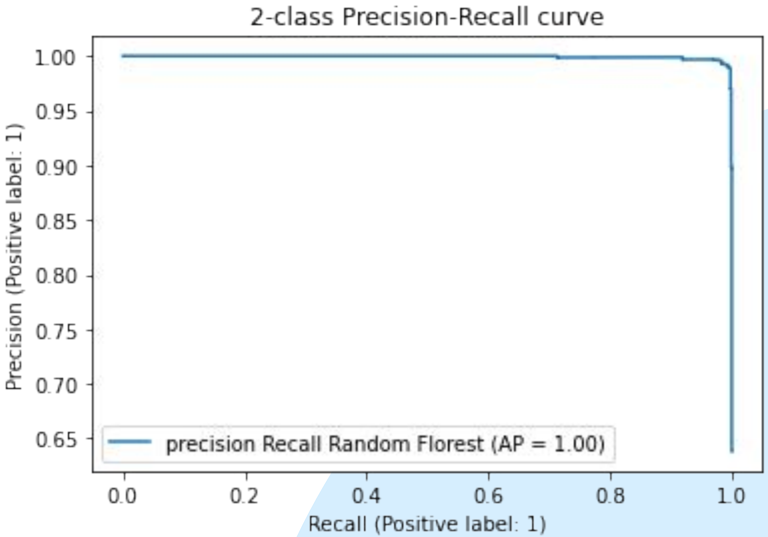
1949
51
3
1997
0
1
True label
Predicted label
1750
1500
1250
1000
750
500
250

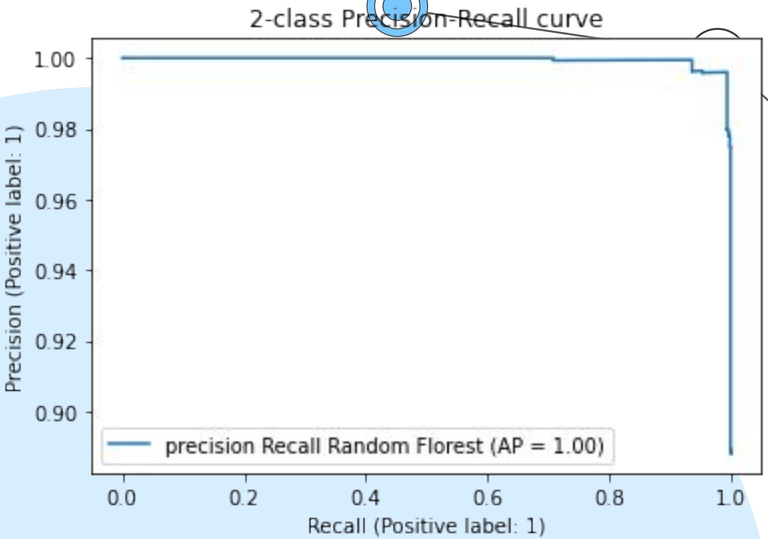


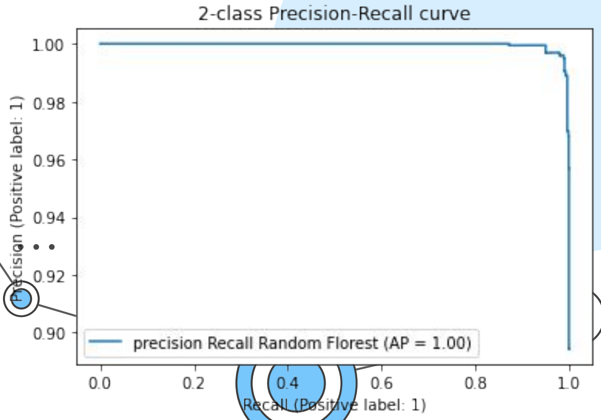

1930
70
2
1998
0
1
True label
Predicted label
1750
1500
1250
1000
750
500
250

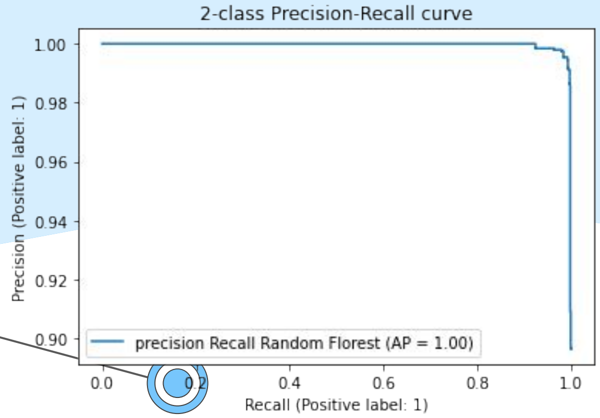

1971
29
4
1996
0
1
True label
Predicted label
1750
1500
1250
1000
750
500
250

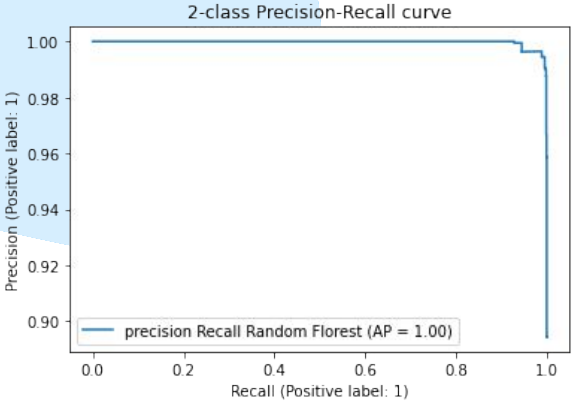

1975
25
4
1996
0
1
True label
Predicted label
1750
1500
1250
1000
750
500
250


2-class Precision-Recall curve
Precision (Positive label: 1)
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
precision Recall Random Florest (AP = 1.00)
0.0
0.2
0.4
0.6
0.8
1.0
Recall (Positive label: 1)

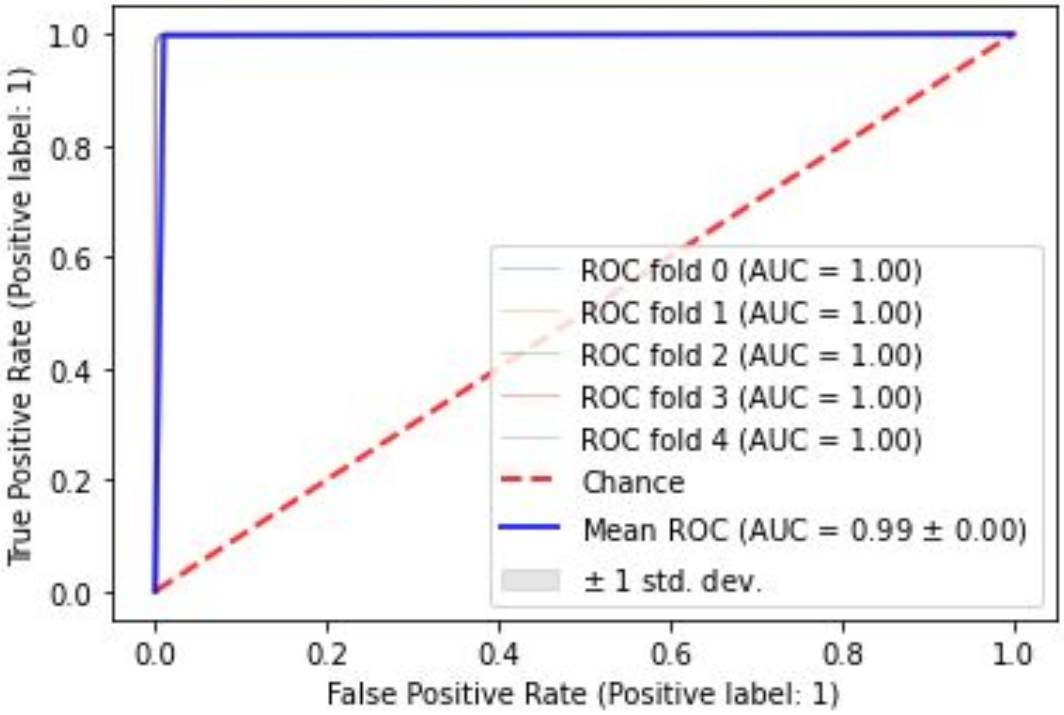

2-class Precision-Recall curve
Precision (Positive label: 1)
1.00
0.98
0.96
0.94
0.92
0.90
precision Recall Random Florest (AP = 1.00)
0.0
0.2
0.4
0.6
0.8
1.0
Recall (Positive label: 1)

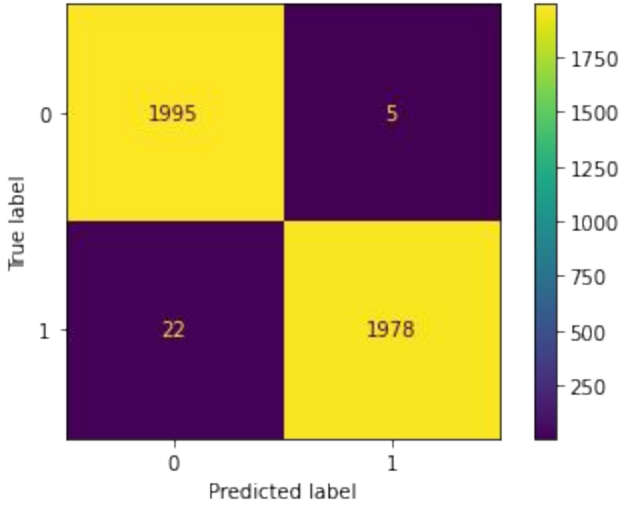

2-class Precision-Recall curve
Precision (Positive label: 1)
1.00
0.98
0.96
0.94
0.92
0.90
precision Recall Random Florest (AP = 1.00)
0.0
0.2
0.4
0.6
0.8
1.0
Recall (Positive label: 1)

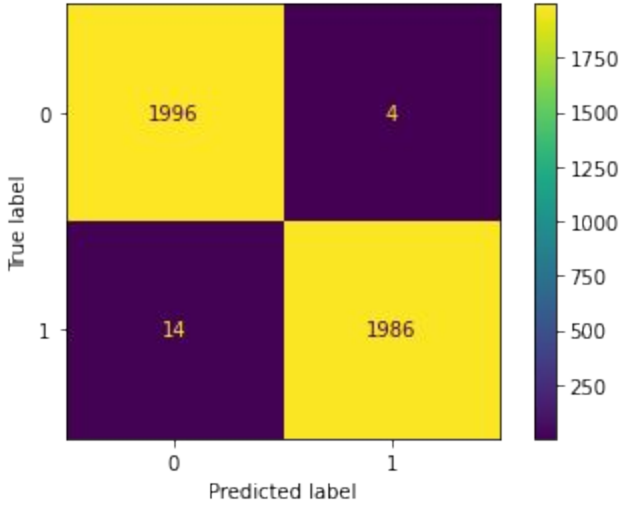

2-class Precision-Recall curve
Precision (Positive label: 1)
1.00
0.98
0.96
0.94
0.92
0.90
precision Recall Random Florest (AP = 1.00)
0.0
0.2
0.4
0.6
0.8
1.0
Recall (Positive label: 1)

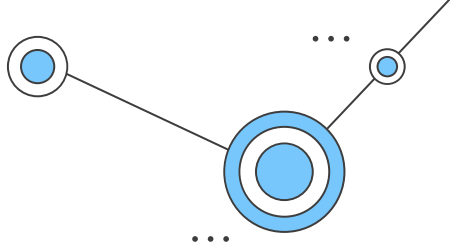

2-class Precision-Recall curve
Precision (Positive label: 1)
1.00
0.98
0.96
0.94
0.92
0.90
precision Recall Random Florest (AP = 1.00)
0.0
0.2
0.4
0.6
0.8
1.0
Recall (Positive label: 1)

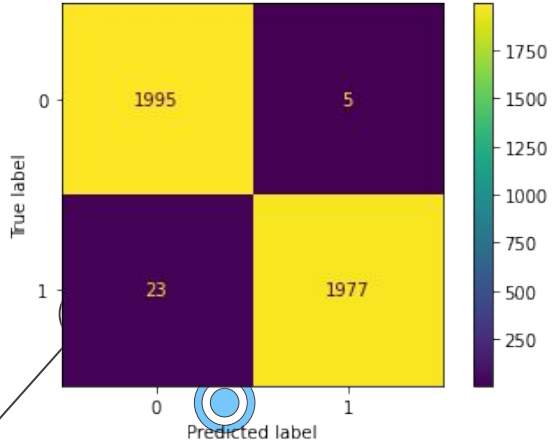
KNN

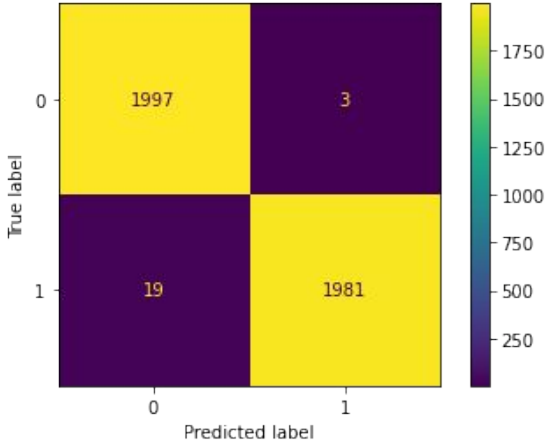

True Positive Rate (Positive label: 1)
False Positive Rate (Positive label: 1)
0.0
0.2
0.4
0.6
0.8
1.0
ROC fold 0 (AUC = 1.00)
ROC fold 1 (AUC = 1.00)
ROC fold 2 (AUC = 1.00)
ROC fold 3 (AUC = 1.00)
ROC fold 4 (AUC = 1.00)
Chance
Mean ROC (AUC = 0.99 ± 0.00)
± 1 std. dev.

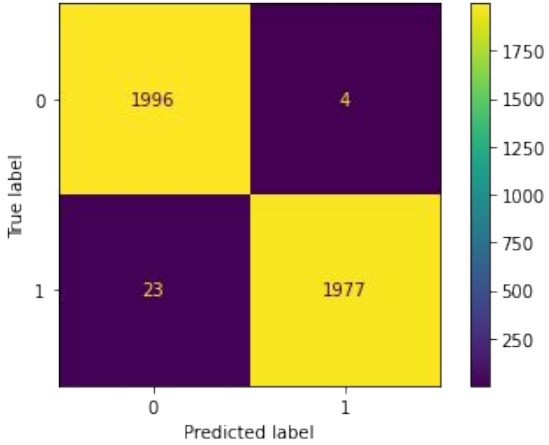

1995
5
22
1978
0
1
True label
Predicted label
1750
1500
1250
1000
750
500
250

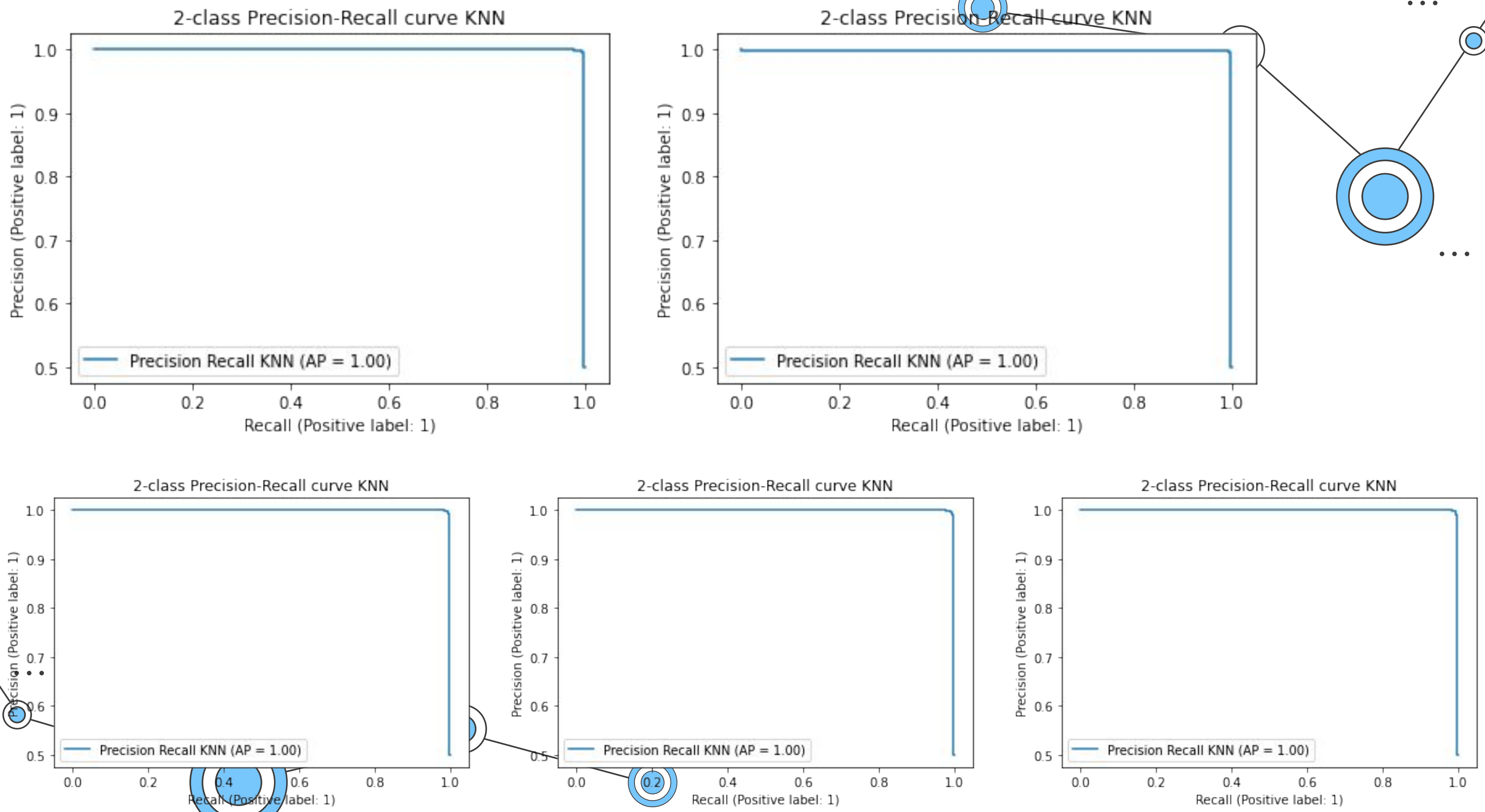

1996
4
14
1986
0
1
True label
Predicted label
1750
1500
1250
1000
750
500
250



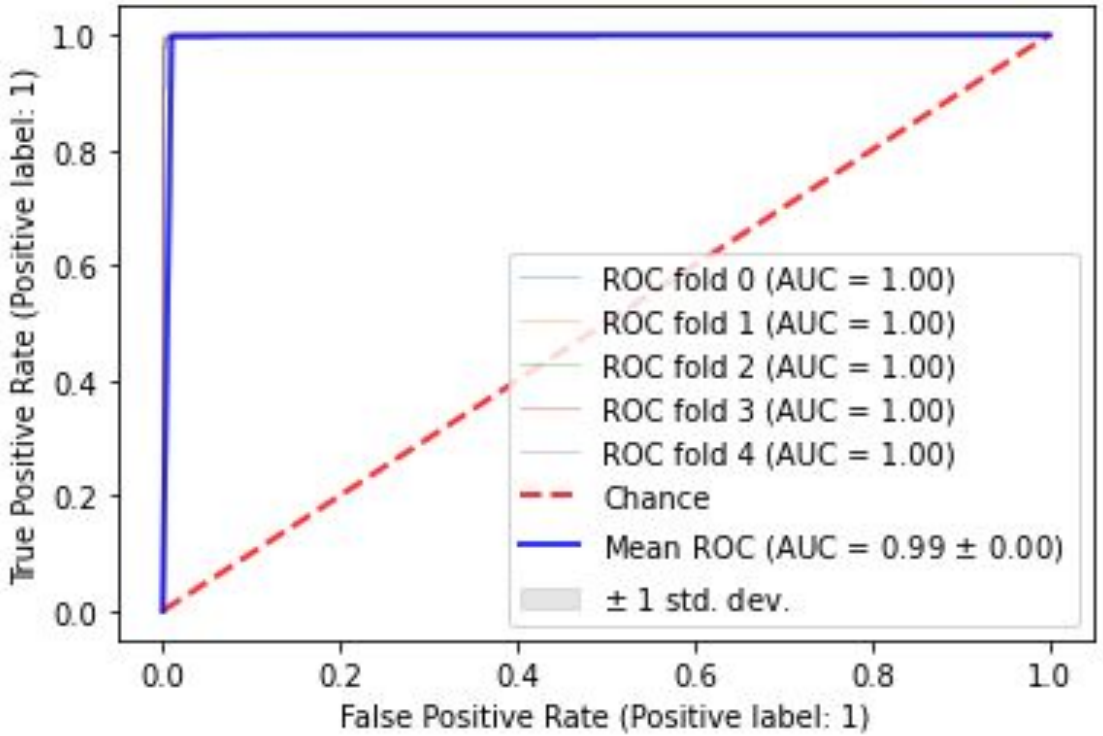

1995
5
23
1977
0
1
True label
Predicted label
1750
1500
1250
1000
750
500
250

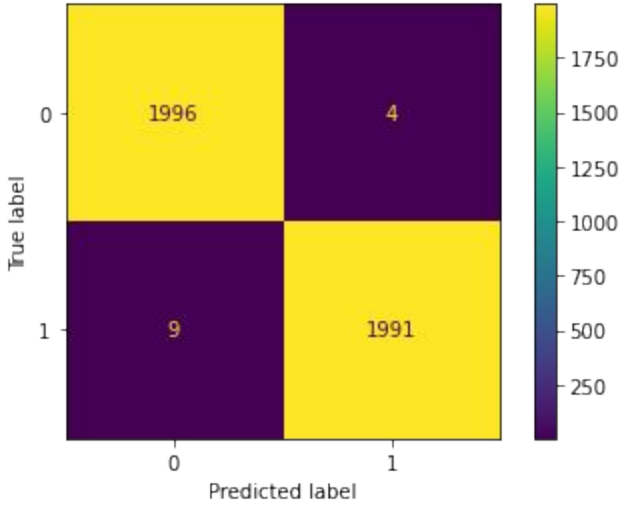

1997
3
19
1981
0
1
True label
Predicted label
1750
1500
1250
1000
750
500
250

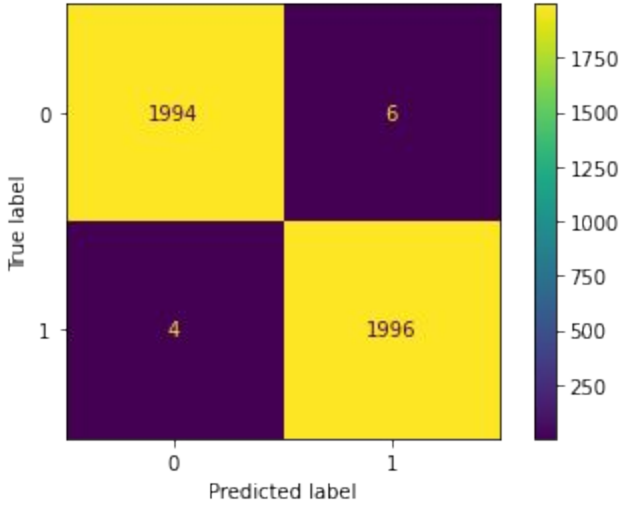

1996
4
23
1977
0
1
True label
Predicted label
1750
1500
1250
1000
750
500
250

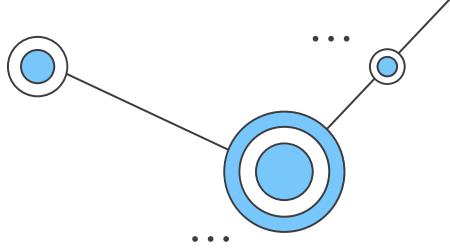

2-class Precision-Recall curve KNN
Precision (Positive label: 1)
Recall (Positive label: 1)
Precision Recall KNN (AP = 1.00)
2-class Precision-Recall curve KNN
Precision (Positive label: 1)
Recall (Positive label: 1)
Precision Recall KNN (AP = 1.00)
2-class Precision-Recall curve KNN
Precision (Positive label: 1)
Recall (Positive label: 1)
Precision Recall KNN (AP = 1.00)
2-class Precision-Recall curve KNN
Precision (Positive label: 1)
Recall (Positive label: 1)
Precision Recall KNN (AP = 1.00)
2-class Precision-Recall curve KNN
Precision (Positive label: 1)
Recall (Positive label: 1)
Precision Recall KNN (AP = 1.00)

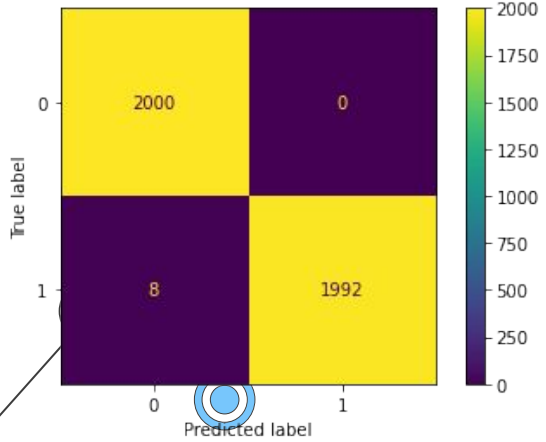
MLP

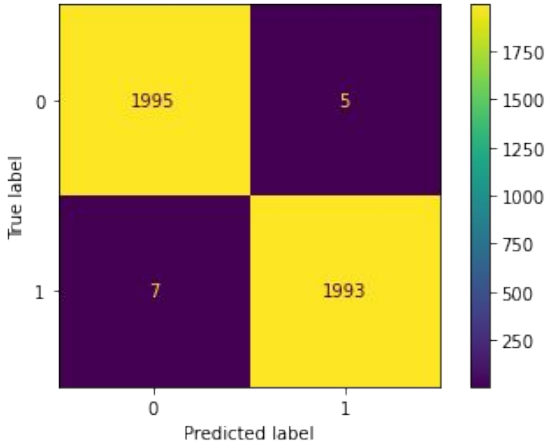

True Positive Rate (Positive label: 1)
False Positive Rate (Positive label: 1)
0.0
0.2
0.4
0.6
0.8
1.0
ROC fold 0 (AUC = 1.00)
ROC fold 1 (AUC = 1.00)
ROC fold 2 (AUC = 1.00)
ROC fold 3 (AUC = 1.00)
ROC fold 4 (AUC = 1.00)
Chance
Mean ROC (AUC = 0.99 ± 0.00)
± 1 std. dev.

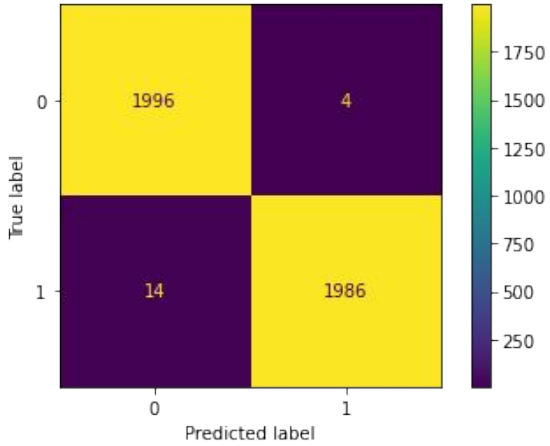

1996
4
9
1991
0
1
True label
Predicted label
1750
1500
1250
1000
750
500
250

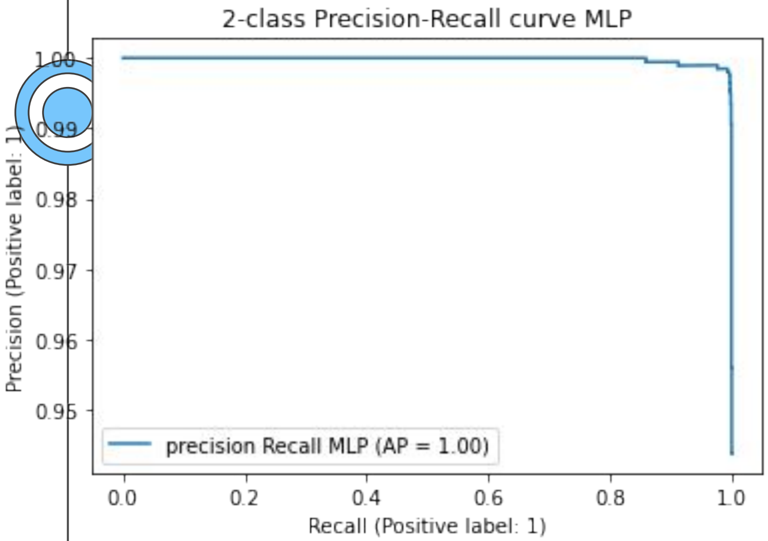

1994
6
4
1996
0
1
True label
Predicted label
1750
1500
1250
1000
750
500
250

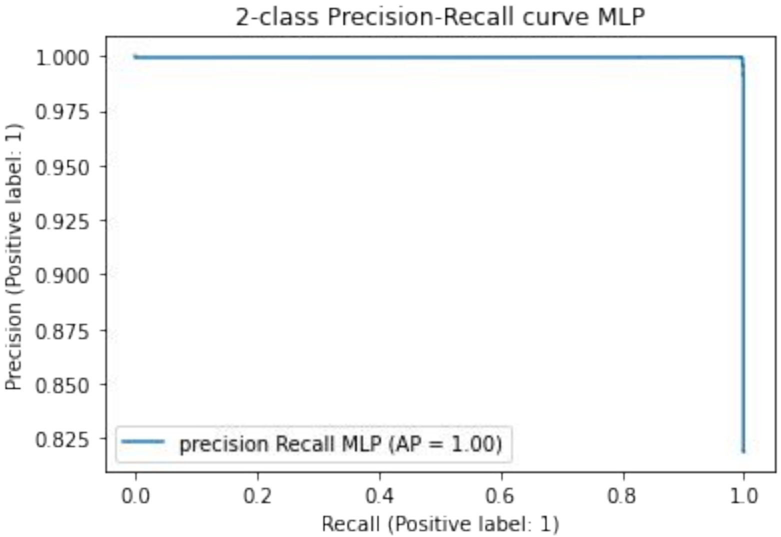


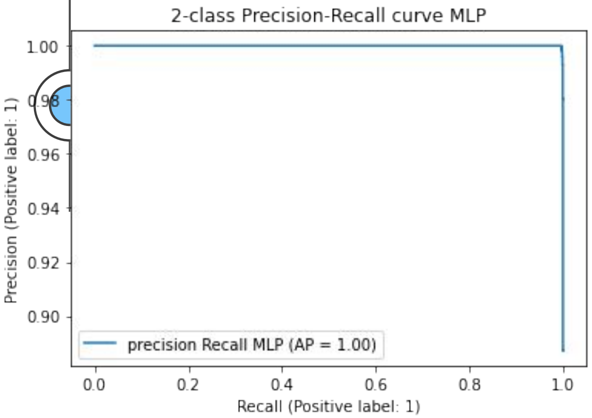

2000
0
8
1992
0
1
True label
Predicted label
2000
1750
1500
1250
1000
750
500
250
0

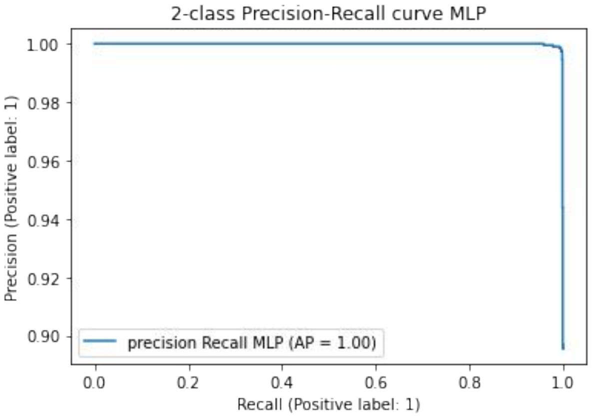

1995
5
7
1993
0
1
True label
Predicted label
1750
1500
1250
1000
750
500
250

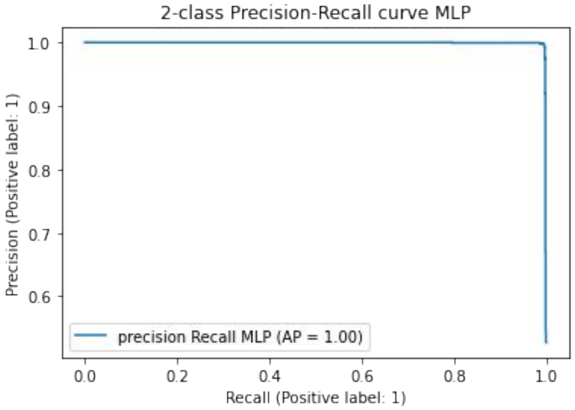

1996
4
14
1986
0
1
True label
Predicted label
1750
1500
1250
1000
750
500
250


2-class Precision-Recall curve MLP
Precision (Positive label: 1)
Recall (Positive label: 1)
precision Recall MLP (AP = 1.00)


2-class Precision-Recall curve MLP
Precision (Positive label: 1)
Recall (Positive label: 1)
precision Recall MLP (AP = 1.00)


2-class Precision-Recall curve MLP
Precision (Positive label: 1)
Recall (Positive label: 1)
precision Recall MLP (AP = 1.00)


2-class Precision-Recall curve MLP
Precision (Positive label: 1)
Recall (Positive label: 1)
precision Recall MLP (AP = 1.00)


2-class Precision-Recall curve MLP
Precision (Positive label: 1)
Recall (Positive label: 1)
precision Recall MLP (AP = 1.00)

Discussão

Alta taxa de acurácia

Ótimos resultados nos três algoritmos

Tipos de Ataque

Cada tipo possui especificidades

Overfitting ou outros métodos

Referencias

Hyunjae Kang, Byung Il Kwak, Young Hun Lee, Haneol Lee, Hwejae Lee and Huy Kang Kim. "Car Hacking and Defense Competition on In-Vehicle Network." Third International Workshop on Automotive and Autonomous Vehicle Security, 2021.

Hyunjae Kang, Byung Il Kwak, Young Hun Lee, Haneol Lee, Hwejae Lee, Huy Kang Kim, February 3, 2021, "Car Hacking: Attack & Defense Challenge 2020 Dataset", IEEE Dataport, doi: https://dx.doi.org/10.21227/qvr7-n418

Hyunsung Lee, Seong Hoon Jeong and Huy Kang Kim, "OTIDS: A Novel Intrusion Detection System for In-vehicle Network by using Remote Frame", PST (Privacy, Security and Trust) 2017