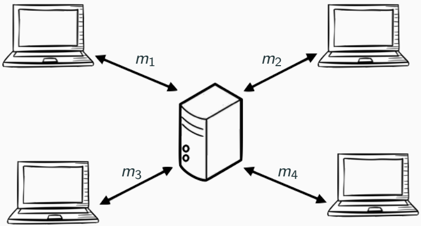
REDOCS

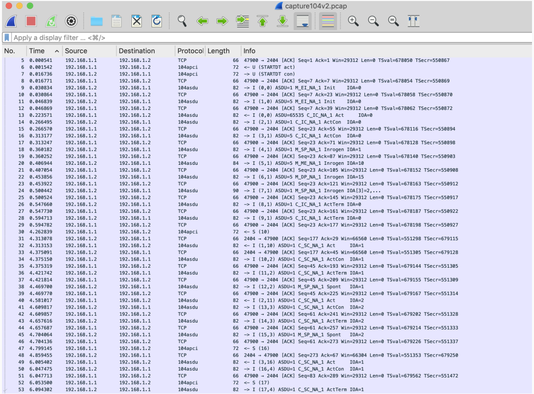
Machine learning for IDS log analysis

Joseph Azar (FEMTO-ST)
Malcolm Bourdon (EDF R&D, LAAS-CNRS)
Alexandre Debant (Univ Rennes, CNRS, IRISA)
Julien Girard-Satabin (CEA, INRIA)
Yuxiao Mao (LAAS-CNRS)

October 25, 2019

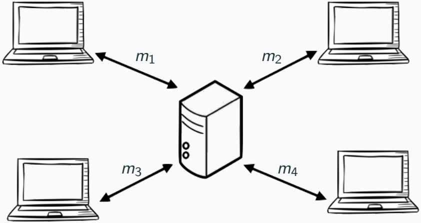
IEC 104 protocol

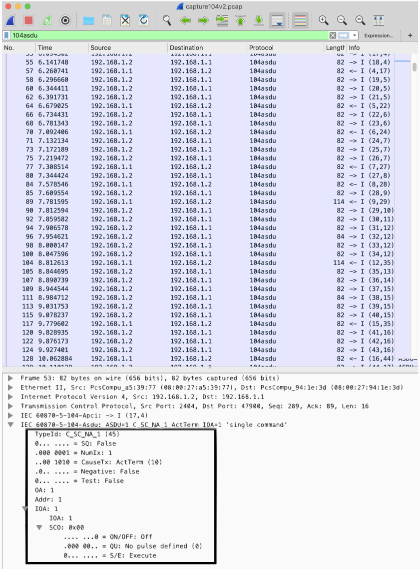




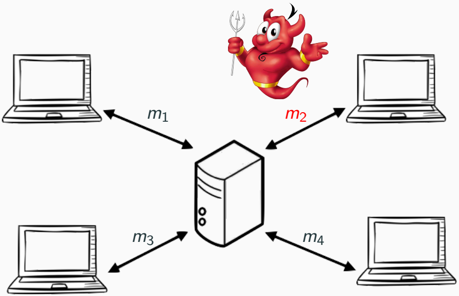
No.	Time	Source	Destination	Protocol	Length	Info
5	0.000541	192.168.1.1	192.168.1.2	TCP	66	47900 → 2404 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=678050 TSecr=550867
6	0.001542	192.168.1.1	192.168.1.2	104apci	72	<- U (STARTDT act)
7	0.016736	192.168.1.2	192.168.1.1	104apci	72	-> U (STARTDT con)
8	0.016771	192.168.1.1	192.168.1.2	TCP	66	47900 → 2404 [ACK] Seq=7 Ack=7 Win=29312 Len=0 TSval=678054 TSecr=550869
9	0.030834	192.168.1.2	192.168.1.1	104asdu	82	-> I (0,0) ASDU=1 M_EI_NA_1 Init IOA=0
10	0.030864	192.168.1.1	192.168.1.2	TCP	66	47900 → 2404 [ACK] Seq=7 Ack=23 Win=29312 Len=0 TSval=678058 TSecr=550870
11	0.046839	192.168.1.2	192.168.1.1	104asdu	82	-> I (1,0) ASDU=5 M_EI_NA_1 Init IOA=0
12	0.046869	192.168.1.1	192.168.1.2	TCP	66	47900 → 2404 [ACK] Seq=7 Ack=39 Win=29312 Len=0 TSval=678062 TSecr=550872
13	0.223571	192.168.1.1	192.168.1.2	104asdu	82	<- I (0,0) ASDU=65535 C_IC_NA_1 Act IOA=0
14	0.266495	192.168.1.2	192.168.1.1	104asdu	82	-> I (2,1) ASDU=1 C_IC_NA_1 ActCon IOA=0
15	0.266570	192.168.1.1	192.168.1.2	TCP	66	47900 → 2404 [ACK] Seq=23 Ack=55 Win=29312 Len=0 TSval=678116 TSecr=550894
16	0.313177	192.168.1.2	192.168.1.1	104asdu	82	-> I (3,1) ASDU=5 C_IC_NA_1 ActCon IOA=0
17	0.313247	192.168.1.1	192.168.1.2	TCP	66	47900 → 2404 [ACK] Seq=23 Ack=71 Win=29312 Len=0 TSval=678128 TSecr=550898
18	0.360182	192.168.1.2	192.168.1.1	104asdu	82	-> I (4,1) ASDU=1 M_SP_NA_1 Inrogen IOA=1
19	0.360252	192.168.1.1	192.168.1.2	TCP	66	47900 → 2404 [ACK] Seq=23 Ack=87 Win=29312 Len=0 TSval=678140 TSecr=550903
20	0.406944	192.168.1.2	192.168.1.1	104asdu	84	-> I (5,1) ASDU=5 M_ME_NA_1 Inrogen IOA=10
21	0.407054	192.168.1.1	192.168.1.2	TCP	66	47900 → 2404 [ACK] Seq=23 Ack=105 Win=29312 Len=0 TSval=678152 TSecr=550908
22	0.453856	192.168.1.2	192.168.1.1	104asdu	82	-> I (6,1) ASDU=5 M_DP_NA_1 Inrogen IOA=15
23	0.453922	192.168.1.1	192.168.1.2	TCP	66	47900 → 2404 [ACK] Seq=23 Ack=121 Win=29312 Len=0 TSval=678163 TSecr=550912
24	0.500442	192.168.1.2	192.168.1.1	104asdu	90	-> I (7,1) ASDU=1 M_SP_NA_1 Inrogen IOA[3]=2,...
25	0.500524	192.168.1.1	192.168.1.2	TCP	66	47900 → 2404 [ACK] Seq=23 Ack=145 Win=29312 Len=0 TSval=678175 TSecr=550917
26	0.547660	192.168.1.2	192.168.1.1	104asdu	82	-> I (8,1) ASDU=1 C_IC_NA_1 ActTerm IOA=0
27	0.547730	192.168.1.1	192.168.1.2	TCP	66	47900 → 2404 [ACK] Seq=23 Ack=161 Win=29312 Len=0 TSval=678187 TSecr=550922
28	0.594713	192.168.1.2	192.168.1.1	104asdu	82	-> I (9,1) ASDU=5 C_IC_NA_1 ActTerm IOA=0
29	0.594782	192.168.1.1	192.168.1.2	TCP	66	47900 → 2404 [ACK] Seq=23 Ack=177 Win=29312 Len=0 TSval=678198 TSecr=550927
30	4.262839	192.168.1.1	192.168.1.2	104apci	72	<- S (10)
31	4.313078	192.168.1.2	192.168.1.1	TCP	66	2404 → 47900 [ACK] Seq=177 Ack=29 Win=66560 Len=0 TSval=551298 TSecr=679115
32	4.313153	192.168.1.1	192.168.1.2	104asdu	82	<- I (1,10) ASDU=1 C_SC_NA_1 Act IOA=1
33	4.375091	192.168.1.2	192.168.1.1	TCP	66	2404 → 47900 [ACK] Seq=177 Ack=45 Win=66560 Len=0 TSval=551305 TSecr=679128
34	4.375150	192.168.1.2	192.168.1.1	104asdu	82	-> I (10,2) ASDU=1 C_SC_NA_1 ActCon IOA=1
35	4.375319	192.168.1.1	192.168.1.2	TCP	66	47900 → 2404 [ACK] Seq=45 Ack=193 Win=29312 Len=0 TSval=679144 TSecr=551305
36	4.421742	192.168.1.2	192.168.1.1	104asdu	82	-> I (11,2) ASDU=1 C_SC_NA_1 ActTerm IOA=1
37	4.421814	192.168.1.1	192.168.1.2	TCP	66	47900 → 2404 [ACK] Seq=45 Ack=209 Win=29312 Len=0 TSval=679155 TSecr=551309
38	4.469700	192.168.1.2	192.168.1.1	104asdu	82	-> I (12,2) ASDU=1 M_SP_NA_1 Spont IOA=1
39	4.469770	192.168.1.1	192.168.1.2	TCP	66	47900 → 2404 [ACK] Seq=45 Ack=225 Win=29312 Len=0 TSval=679167 TSecr=551314
40	4.581017	192.168.1.1	192.168.1.2	104asdu	82	<- I (2,11) ASDU=1 C_SC_NA_1 Act IOA=2
41	4.609817	192.168.1.2	192.168.1.1	104asdu	82	-> I (13,3) ASDU=1 C_SC_NA_1 ActCon IOA=2
42	4.609857	192.168.1.1	192.168.1.2	TCP	66	47900 → 2404 [ACK] Seq=61 Ack=241 Win=29312 Len=0 TSval=679202 TSecr=551328
43	4.657616	192.168.1.2	192.168.1.1	104asdu	82	-> I (14,3) ASDU=1 C_SC_NA_1 ActTerm IOA=2
44	4.657687	192.168.1.1	192.168.1.2	TCP	66	47900 → 2404 [ACK] Seq=61 Ack=257 Win=29312 Len=0 TSval=679214 TSecr=551333
45	4.704064	192.168.1.2	192.168.1.1	104asdu	82	-> I (15,3) ASDU=1 M_SP_NA_1 Spont IOA=2
46	4.704136	192.168.1.1	192.168.1.2	TCP	66	47900 → 2404 [ACK] Seq=61 Ack=273 Win=29312 Len=0 TSval=679226 TSecr=551337
47	4.799145	192.168.1.1	192.168.1.2	104apci	72	<- S (16)
48	4.859455	192.168.1.2	192.168.1.1	TCP	66	2404 → 47900 [ACK] Seq=273 Ack=67 Win=66304 Len=0 TSval=551353 TSecr=679250
49	6.005402	192.168.1.1	192.168.1.2	104asdu	82	<- I (3,16) ASDU=1 C_SC_NA_1 Act IOA=1
50	6.047475	192.168.1.2	192.168.1.1	104asdu	82	-> I (16,4) ASDU=1 C_SC_NA_1 ActCon IOA=1
51	6.047713	192.168.1.1	192.168.1.2	TCP	66	47900 → 2404 [ACK] Seq=83 Ack=289 Win=29312 Len=0 TSval=679562 TSecr=551472
52	6.053500	192.168.1.1	192.168.1.2	104apci	72	<- S (17)
53	6.094302	192.168.1.2	192.168.1.1	104asdu	82	-> I (17,4) ASDU=1 C_SC_NA_1 ActTerm IOA=1

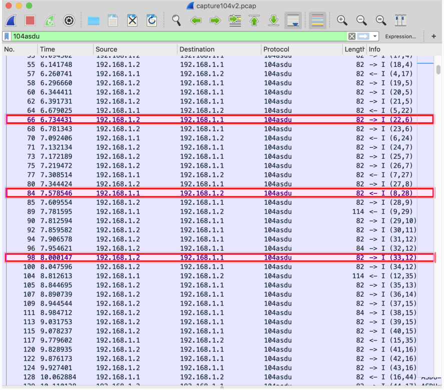
IEC 104 protocol





IEC 104 protocol - attacks





No.	Time	Source	Destination	Protocol	Length	Info
55	6.141748	192.168.1.2	192.168.1.1	104asdu	82	-> I (18,4)
57	6.260741	192.168.1.1	192.168.1.2	104asdu	82	<- I (4,17)
58	6.296660	192.168.1.2	192.168.1.1	104asdu	82	-> I (19,5)
60	6.344411	192.168.1.2	192.168.1.1	104asdu	82	-> I (20,5)
62	6.391731	192.168.1.2	192.168.1.1	104asdu	82	-> I (21,5)
64	6.679025	192.168.1.1	192.168.1.2	104asdu	82	<- I (5,22)
66	6.734431	192.168.1.2	192.168.1.1	104asdu	82	-> I (22,6)
68	6.781343	192.168.1.2	192.168.1.1	104asdu	82	-> I (23,6)
70	7.092406	192.168.1.1	192.168.1.2	104asdu	82	<- I (6,24)
71	7.132134	192.168.1.2	192.168.1.1	104asdu	82	-> I (24,7)
73	7.172189	192.168.1.2	192.168.1.1	104asdu	82	-> I (25,7)
75	7.219472	192.168.1.2	192.168.1.1	104asdu	82	-> I (26,7)
77	7.308514	192.168.1.1	192.168.1.2	104asdu	82	<- I (7,27)
80	7.344424	192.168.1.2	192.168.1.1	104asdu	82	-> I (27,8)
84	7.578546	192.168.1.1	192.168.1.2	104asdu	82	<- I (8,28)
85	7.609554	192.168.1.2	192.168.1.1	104asdu	82	-> I (28,9)
89	7.781595	192.168.1.1	192.168.1.2	104asdu	114	<- I (9,29)
90	7.812594	192.168.1.2	192.168.1.1	104asdu	82	-> I (29,10)
92	7.859582	192.168.1.2	192.168.1.1	104asdu	82	-> I (30,11)
94	7.906578	192.168.1.2	192.168.1.1	104asdu	82	-> I (31,12)
96	7.954621	192.168.1.2	192.168.1.1	104asdu	84	-> I (32,12)
98	8.000147	192.168.1.2	192.168.1.1	104asdu	82	-> I (33,12)
100	8.047596	192.168.1.2	192.168.1.1	104asdu	82	-> I (34,12)
104	8.812613	192.168.1.1	192.168.1.2	104asdu	114	<- I (12,35)
105	8.844695	192.168.1.2	192.168.1.1	104asdu	82	-> I (35,13)
107	8.890739	192.168.1.2	192.168.1.1	104asdu	82	-> I (36,14)
109	8.944544	192.168.1.2	192.168.1.1	104asdu	82	-> I (37,15)
111	8.984712	192.168.1.2	192.168.1.1	104asdu	84	-> I (38,15)
113	9.031753	192.168.1.2	192.168.1.1	104asdu	82	-> I (39,15)
115	9.078237	192.168.1.2	192.168.1.1	104asdu	82	-> I (40,15)
117	9.779602	192.168.1.1	192.168.1.2	104asdu	82	<- I (15,35)
120	9.828935	192.168.1.2	192.168.1.1	104asdu	82	-> I (41,16)
122	9.876173	192.168.1.2	192.168.1.1	104asdu	82	-> I (42,16)
124	9.927401	192.168.1.2	192.168.1.1	104asdu	82	-> I (43,16)
128	10.062884	192.168.1.1	192.168.1.2	104asdu	82	<- I (16,44)

Several classes of attacks

The attacker is able to:

- replay valid packet already sent
- forge and send an invalid packet
- sending arbitrary messages of the protocol
- sending many packets quickly

Problem

How can we detect malicious behaviours using machine learning techniques?

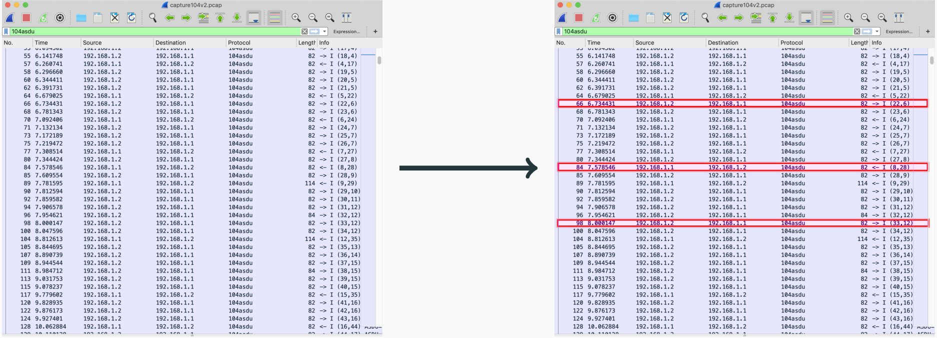


Table of contents

ML blitz

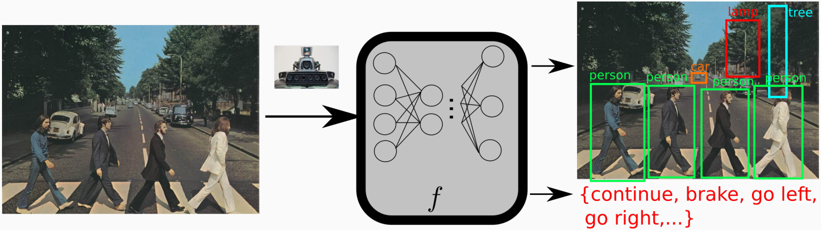
The big question

Why do we use machine learning today?

Drowning in a sea of information

about 10^6 terabytes per day

Hard to specify sometimes



No specification of what is a pedestrian: learn from examples

What is machine learning?

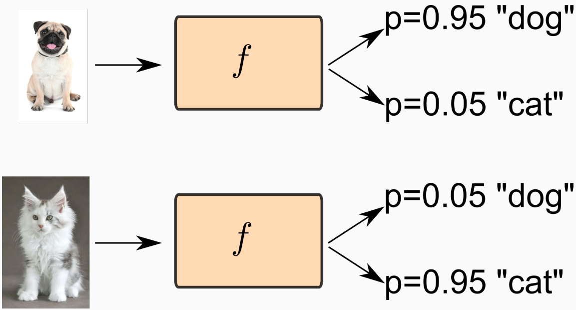
$\tilde{f}$: ideal function
$\tilde{\mathcal{X}}$: ideal representation of data

Goal
learn f approximating $\tilde{f}$, using an approximation of data $\mathcal{X}$

Supervised learning

Dataset $\mathcal{X}$ is *labelled*

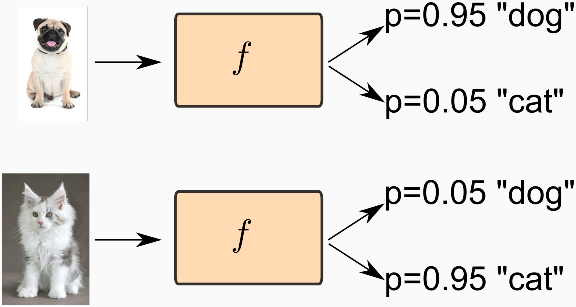
Approximated function: *classifier* between the different labels



Supervised learning

Dataset $\mathcal{X}$ is *labelled*

Approximated function: *classifier* between the different labels



Remark: labelling data is costly!

Some standard algorithms

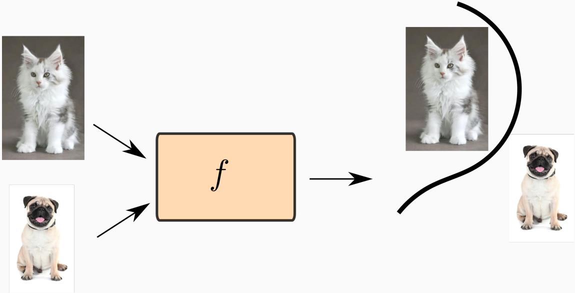
Algorithm	Explainability	Generalization	Learning cost
Decision Tree (DT)	very good	poor	cheap
Support Vector Machine (SVM)	poor	good	cheap
Neural Networks (NN)	poor	very good	expensive

Unsupervised learning

Dataset $\mathcal{X}$ is **not labelled**

Rely on the inherent structure of the data

Approximated function: a representation of the data



Some standard algorithms

Algorithm	Generalization	Learning cost
Clustering (k-nn)	good	cheap
Dimensionality reduction (PCA, t-SNE)	poor	cheap
Neural networks (auto-encoders)	very good	expensive

Data analysis

Data analysis: first analysis

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
1	addr	attack	capturec	causetx	dst_port	frame_	ioa	ip_checks	ip_che	ip_dest	ip_size	ip_src	neg	num	oa	proto_nan	protc	sq	src_port	tcp_size	test	timestam	typeid	qoi	siq	sco	dco
2	100	0	82	4	47900	82	0	65425	2	192.168.1.1	20	192.168.1.2	0	1	0	104asdu	10	0	2404	32	0	1.5E+09	70	-1	-1	-1	-1
3	500	0	82	4	47900	82	0	65424	2	192.168.1.1	20	192.168.1.2	0	1	0	104asdu	10	0	2404	32	0	1.5E+09	70	-1	-1	-1	-1
4	ffff	0	82	6	2404	82	0	7059	2	192.168.1.2	20	192.168.1.1	0	1	1	104asdu	10	0	47900	32	0	1.5E+09	100	20	-1	-1	-1
5	100	0	82	7	47900	82	0	65423	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	100	20	-1	-1	-1
6	500	0	82	7	47900	82	0	65422	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	100	20	-1	-1	-1
7	100	0	82	20	47900	82	65536	65421	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	1	-1	0	-1	-1
8	500	0	84	20	47900	84	655360	65418	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	12	0	2404	32	0	1.5E+09	9	-1	-1	-1	-1
9	500	0	82	20	47900	82	983040	65419	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	3	-1	-1	-1	-1
10	100	0	90	20	47900	90	131072	65410	2	192.168.1.1	20	192.168.1.2	0	3	1	104asdu	18	0	2404	32	0	1.5E+09	1	-1	0	-1	-1
11	100	0	82	10	47900	82	0	65417	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	100	20	-1	-1	-1
12	500	0	82	10	47900	82	0	65416	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	100	20	-1	-1	-1
13	100	0	82	6	2404	82	65536	7049	2	192.168.1.2	20	192.168.1.1	0	1	1	104asdu	10	0	47900	32	0	1.5E+09	45	-1	-1	1	-1
14	100	0	82	7	47900	82	65536	65413	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	1	-1
15	100	0	82	10	47900	82	65536	65412	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	1	-1
16	100	0	82	3	47900	82	65536	65411	2	192.168.1.1	20	192.168.1.2	0	1	0	104asdu	10	0	2404	32	0	1.5E+09	1	-1	1	-1	-1
17	100	0	82	6	2404	82	131072	7045	2	192.168.1.2	20	192.168.1.1	0	1	1	104asdu	10	0	47900	32	0	1.5E+09	45	-1	-1	1	-1
18	100	0	82	7	47900	82	131072	65410	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	1	-1
19	100	0	82	10	47900	82	131072	65409	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	1	-1
20	100	0	82	3	47900	82	131072	65408	2	192.168.1.1	20	192.168.1.2	0	1	0	104asdu	10	0	2404	32	0	1.5E+09	1	-1	1	-1	-1
21	100	0	82	6	2404	82	65536	7040	2	192.168.1.2	20	192.168.1.1	0	1	1	104asdu	10	0	47900	32	0	1.5E+09	45	-1	-1	0	-1
22	100	0	82	7	47900	82	65536	65406	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
23	100	0	82	10	47900	82	65536	65405	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
24	100	0	82	3	47900	82	65536	65404	2	192.168.1.1	20	192.168.1.2	0	1	0	104asdu	10	0	2404	32	0	1.5E+09	1	-1	0	-1	-1
25	100	0	82	6	2404	82	131072	7035	2	192.168.1.2	20	192.168.1.1	0	1	1	104asdu	10	0	47900	32	0	1.5E+09	45	-1	-1	0	-1
26	100	0	82	7	47900	82	131072	65403	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
27	100	0	82	10	47900	82	131072	65402	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
28	100	0	82	3	47900	82	131072	65401	2	192.168.1.1	20	192.168.1.2	0	1	0	104asdu	10	0	2404	32	0	1.5E+09	1	-1	0	-1	-1
29	100	1	82	6	2404	82	131072	7031	2	192.168.1.2	20	192.168.1.1	0	1	1	104asdu	10	0	47900	32	0	1.5E+09	45	-1	-1	0	-1
30	100	1	82	7	47900	82	131072	65399	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
31	100	1	82	10	47900	82	131072	65398	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
32	100	1	82	6	2404	82	131072	7028	2	192.168.1.2	20	192.168.1.1	0	1	1	104asdu	10	0	47900	32	0	1.5E+09	45	-1	-1	0	-1
33	100	1	82	7	47900	82	131072	65397	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
34	100	1	82	10	47900	82	131072	65396	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
35	100	0	82	3	47900	82	131072	65395	2	192.168.1.1	20	192.168.1.2	0	1	0	104asdu	10	0	2404	32	0	1.5E+09	1	-1	0	-1	-1

- data size: 863 row$\times$27 columns, 147 kB
- attack / non attack: 610 / 253

Data analysis: first analysis

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
1	addr	attack	capturec	causetx	dst_port	frame_	ioa	ip_checks	ip_che	ip_dest	ip_size	ip_src	neg	num	oa	proto_nar	prot	sq	src_port	tcp_size	test	timestam	typeid	qoi	siq	sco	dco
2	100	0	82	4	47900	82	0	65425	2	192.168.1.1	20	192.168.1.2	0	1	0	104asdu	10	0	2404	32	0	1.5E+09	70	-1	-1	-1	-1
3	500	0	82	4	47900	82	0	65424	2	192.168.1.1	20	192.168.1.2	0	1	0	104asdu	10	0	2404	32	0	1.5E+09	70	-1	-1	-1	-1
4	ffff	0	82	6	2404	82	0	7059	2	192.168.1.2	20	192.168.1.1	0	1	1	104asdu	10	0	47900	32	0	1.5E+09	100	20	-1	-1	-1
5	100	0	82	7	47900	82	0	65423	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	100	20	-1	-1	-1
6	500	0	82	7	47900	82	0	65422	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	100	20	-1	-1	-1
7	100	0	82	20	47900	82	65536	65421	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	1	-1	0	-1	-1
8	500	0	84	20	47900	84	655360	65418	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	12	0	2404	32	0	1.5E+09	9	-1	-1	-1	-1
9	500	0	82	20	47900	82	983040	65419	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	3	-1	-1	-1	-1
10	100	0	90	20	47900	90	131072	65410	2	192.168.1.1	20	192.168.1.2	0	3	1	104asdu	18	0	2404	32	0	1.5E+09	1	-1	0	-1	-1
11	100	0	82	10	47900	82	0	65417	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	100	20	-1	-1	-1
12	500	0	82	10	47900	82	0	65416	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	100	20	-1	-1	-1
13	100	0	82	6	2404	82	65536	7049	2	192.168.1.2	20	192.168.1.1	0	1	1	104asdu	10	0	47900	32	0	1.5E+09	45	-1	-1	1	-1
14	100	0	82	7	47900	82	65536	65413	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	1	-1
15	100	0	82	10	47900	82	65536	65412	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	1	-1
16	100	0	82	3	47900	82	65536	65411	2	192.168.1.1	20	192.168.1.2	0	1	0	104asdu	10	0	2404	32	0	1.5E+09	1	-1	1	-1	-1
17	100	0	82	6	2404	82	131072	7045	2	192.168.1.2	20	192.168.1.1	0	1	1	104asdu	10	0	47900	32	0	1.5E+09	45	-1	-1	1	-1
18	100	0	82	7	47900	82	131072	65410	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	1	-1
19	100	0	82	10	47900	82	131072	65409	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	1	-1
20	100	0	82	3	47900	82	131072	65408	2	192.168.1.1	20	192.168.1.2	0	1	0	104asdu	10	0	2404	32	0	1.5E+09	1	-1	1	-1	-1
21	100	0	82	6	2404	82	65536	7040	2	192.168.1.2	20	192.168.1.1	0	1	1	104asdu	10	0	47900	32	0	1.5E+09	45	-1	-1	0	-1
22	100	0	82	7	47900	82	65536	65406	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
23	100	0	82	10	47900	82	65536	65405	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
24	100	0	82	3	47900	82	65536	65404	2	192.168.1.1	20	192.168.1.2	0	1	0	104asdu	10	0	2404	32	0	1.5E+09	1	-1	0	-1	-1
25	100	0	82	6	2404	82	131072	7035	2	192.168.1.2	20	192.168.1.1	0	1	1	104asdu	10	0	47900	32	0	1.5E+09	45	-1	-1	0	-1
26	100	0	82	7	47900	82	131072	65403	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
27	100	0	82	10	47900	82	131072	65402	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
28	100	0	82	3	47900	82	131072	65401	2	192.168.1.1	20	192.168.1.2	0	1	0	104asdu	10	0	2404	32	0	1.5E+09	1	-1	0	-1	-1
29	100	1	82	6	2404	82	131072	7031	2	192.168.1.2	20	192.168.1.1	0	1	1	104asdu	10	0	47900	32	0	1.5E+09	45	-1	-1	0	-1
30	100	1	82	7	47900	82	131072	65399	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
31	100	1	82	10	47900	82	131072	65398	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
32	100	1	82	6	2404	82	131072	7028	2	192.168.1.2	20	192.168.1.1	0	1	1	104asdu	10	0	47900	32	0	1.5E+09	45	-1	-1	0	-1
33	100	1	82	7	47900	82	131072	65397	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
34	100	1	82	10	47900	82	131072	65396	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
35	100	0	82	3	47900	82	131072	65395	2	192.168.1.1	20	192.168.1.2	0	1	0	104asdu	10	0	2404	32	0	1.5E+09	1	-1	0	-1	-1

- attack
- captured_length, dst_port, frame_length, ip_checksum, ip_checksum_status, ip_size, ip_dest, ip_src, src_port, tcp_size, timestamp
- addr, causetx, ioa, nega, numix, oa, proto_name, proto_size, sq, typeid, test, qoi, siq, sco, dco

Data analysis: first analysis

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
1	addr	attack	capture	causetx	dst_port	frame_	ioa	ip_checks	ip_che	ip_dest	ip_size	ip_src	neg	num	oa	proto_nan	protc	sq	src_port	tcp_size	test	timestam	typeid	qoi	siq	sco	dco
2	100	0	82	4	47900	82	0	65425	2	192.168.1.1	20	192.168.1.2	0	1	0	104asdu	10	0	2404	32	0	1.5E+09	70	-1	-1	-1	-1
3	500	0	82	4	47900	82	0	65424	2	192.168.1.1	20	192.168.1.2	0	1	0	104asdu	10	0	2404	32	0	1.5E+09	70	-1	-1	-1	-1
4	ffff	0	82	6	2404	82	0	7059	2	192.168.1.2	20	192.168.1.1	0	1	1	104asdu	10	0	47900	32	0	1.5E+09	100	20	-1	-1	-1
5	100	0	82	7	47900	82	0	65423	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	100	20	-1	-1	-1
6	500	0	82	7	47900	82	0	65422	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	100	20	-1	-1	-1
7	100	0	82	20	47900	82	65536	65421	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	1	-1	0	-1	-1
8	500	0	84	20	47900	84	655360	65418	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	12	0	2404	32	0	1.5E+09	9	-1	-1	-1	-1
9	500	0	82	20	47900	82	983040	65419	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	3	-1	-1	-1	-1
10	100	0	90	20	47900	90	131072	65410	2	192.168.1.1	20	192.168.1.2	0	3	1	104asdu	18	0	2404	32	0	1.5E+09	1	-1	0	-1	-1
11	100	0	82	10	47900	82	0	65417	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	100	20	-1	-1	-1
12	500	0	82	10	47900	82	0	65416	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	100	20	-1	-1	-1
13	100	0	82	6	2404	82	65536	7049	2	192.168.1.2	20	192.168.1.1	0	1	1	104asdu	10	0	47900	32	0	1.5E+09	45	-1	-1	1	-1
14	100	0	82	7	47900	82	65536	65413	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	1	-1
15	100	0	82	10	47900	82	65536	65412	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	1	-1
16	100	0	82	3	47900	82	65536	65411	2	192.168.1.1	20	192.168.1.2	0	1	0	104asdu	10	0	2404	32	0	1.5E+09	1	-1	1	-1	-1
17	100	0	82	6	2404	82	131072	7045	2	192.168.1.2	20	192.168.1.1	0	1	1	104asdu	10	0	47900	32	0	1.5E+09	45	-1	-1	1	-1
18	100	0	82	7	47900	82	131072	65410	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	1	-1
19	100	0	82	10	47900	82	131072	65409	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	1	-1
20	100	0	82	3	47900	82	131072	65408	2	192.168.1.1	20	192.168.1.2	0	1	0	104asdu	10	0	2404	32	0	1.5E+09	1	-1	1	-1	-1
21	100	0	82	6	2404	82	65536	7040	2	192.168.1.2	20	192.168.1.1	0	1	1	104asdu	10	0	47900	32	0	1.5E+09	45	-1	-1	0	-1
22	100	0	82	7	47900	82	65536	65406	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
23	100	0	82	10	47900	82	65536	65405	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
24	100	0	82	3	47900	82	65536	65404	2	192.168.1.1	20	192.168.1.2	0	1	0	104asdu	10	0	2404	32	0	1.5E+09	1	-1	0	-1	-1
25	100	0	82	6	2404	82	131072	7035	2	192.168.1.2	20	192.168.1.1	0	1	1	104asdu	10	0	47900	32	0	1.5E+09	45	-1	-1	0	-1
26	100	0	82	7	47900	82	131072	65403	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
27	100	0	82	10	47900	82	131072	65402	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
28	100	0	82	3	47900	82	131072	65401	2	192.168.1.1	20	192.168.1.2	0	1	0	104asdu	10	0	2404	32	0	1.5E+09	1	-1	0	-1	-1
29	100	1	82	6	2404	82	131072	7031	2	192.168.1.2	20	192.168.1.1	0	1	1	104asdu	10	0	47900	32	0	1.5E+09	45	-1	-1	0	-1
30	100	1	82	7	47900	82	131072	65399	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
31	100	1	82	10	47900	82	131072	65398	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
32	100	1	82	6	2404	82	131072	7028	2	192.168.1.2	20	192.168.1.1	0	1	1	104asdu	10	0	47900	32	0	1.5E+09	45	-1	-1	0	-1
33	100	1	82	7	47900	82	131072	65397	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
34	100	1	82	10	47900	82	131072	65396	2	192.168.1.1	20	192.168.1.2	0	1	1	104asdu	10	0	2404	32	0	1.5E+09	45	-1	-1	0	-1
35	100	0	82	3	47900	82	131072	65395	2	192.168.1.1	20	192.168.1.2	0	1	0	104asdu	10	0	2404	32	0	1.5E+09	1	-1	0	-1	-1

- important fields:
  - typeid (type identification)
  - causetx (cause of transmission)
  - ioa (information object address)
  - ip_checksum (IPs, sequence of transmission)

Data analysis: preprocessing

- normalization
  - why? different range
  - what? numeric fields
  - how? mean = 0, standard deviation = [-1,1]

Data analysis: preprocessing

- normalization
  - why? different range
  - what? numeric fields
  - how? mean = 0, standard deviation = [-1,1]

- one-hot encoding
  - why? strings
  - what? non numeric fields (type, address...)
  - how? e.g., IP address

192.168.1.1	[1, 0]
192.168.1.2	[0, 1]

Data analysis: preprocessing

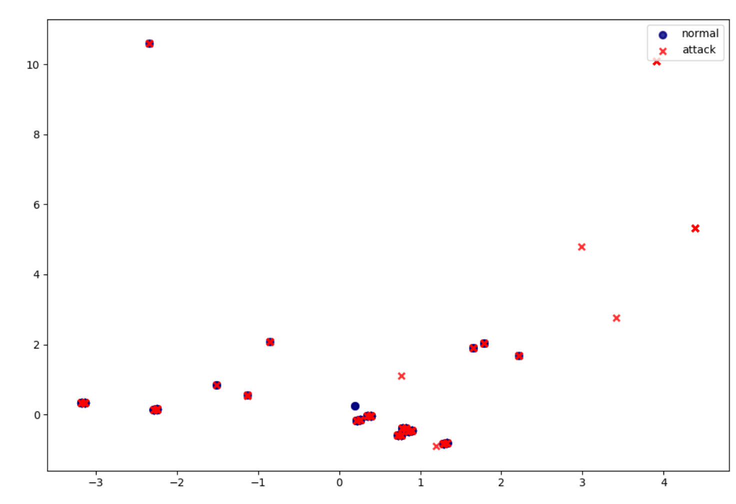
- normalization
  - why? different range
  - what? numeric fields
  - how? mean = 0, standard deviation = [-1,1]

- one-hot encoding
  - why? strings
  - what? non numeric fields (type, address...)
  - how? e.g., IP address

192.168.1.1	[1, 0, 0]
192.168.1.2	[0, 1, 0]
192.168.1.3	[0, 0, 1]

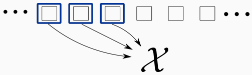
Data analysis: Principal Component Analysis (PCA)

- dimensionality reduction



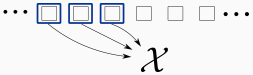
Data analysis: split technics

- sequential split (75% training)
  - training: 398 normals, 249 attacks
  - evaluation: 212 normals, 4 attacks

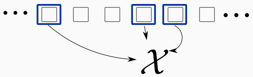


Data analysis: split technics

- sequential split (75% training)
  - training: 398 normals, 249 attacks
  - evaluation: 212 normals, 4 attacks



- random split (75% training)
  - training: 448 normals, 199 attacks
  - evaluation: 162 normals, 54 attacks

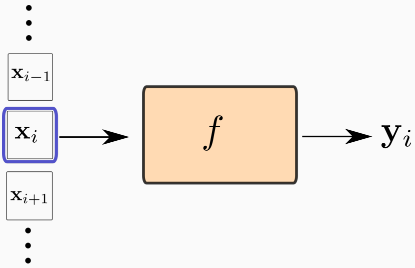


Data analysis: limitations of dataset

- small dataset with only 863 IEC104 packets
- repetitive legitimate behaviours
- unbalanced attacks behaviours
  - many Denial-of-Service (DoS) attack packets
  - few occurences of each attack
  - 2 fields to draw out 1/4 attacks
  - 1 field with sequence to draw out most of DoS attacks

ML without sequence

Problem statement and limitations



- inputs: one packet for one output
- limitation: no context (DoS attacks indistinguishable)

Decision Trees (DT)

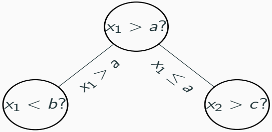
$$\text{Dataset } \mathcal{X}, \text{ classes } \mathcal{Y} = \{y_1, y_2\}, \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \in \mathcal{X}$$

$x_1 > a \rightarrow \mathcal{P}(\mathbf{x} \in y_1)$?

$x_1 \leq a \rightarrow \mathcal{P}(\mathbf{x} \in y_2)$?

Decision Trees (DT)

$$\text{Dataset } \mathcal{X}, \text{ classes } \mathcal{Y} = \{y_1, y_2\}, \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \in \mathcal{X}$$

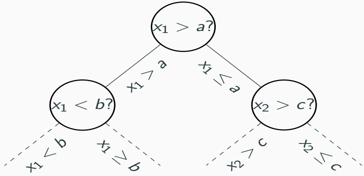


$x_1 > a \rightarrow \mathcal{P}(\mathbf{x} \in y_1)?$
$x_1 \leq a \rightarrow \mathcal{P}(\mathbf{x} \in y_2)?$
Decision tree answer those questions

Decision Trees (DT)

Dataset $\mathcal{X}$, classes $\mathcal{Y} = \{y_1, y_2\}$, $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \in \mathcal{X}$

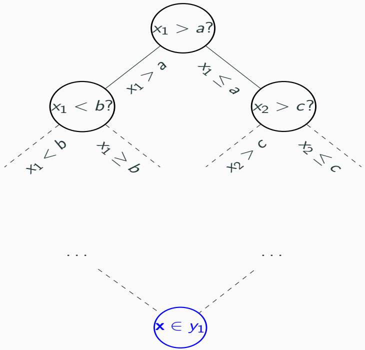
$x_1 > a \rightarrow \mathcal{P}(\mathbf{x} \in y_1)$?
$x_1 \leq a \rightarrow \mathcal{P}(\mathbf{x} \in y_2)$?
Decision tree answer those questions



Decision Trees (DT)

$$\text{Dataset } \mathcal{X}, \text{ classes } \mathcal{Y} = \{y_1, y_2\}, \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \in \mathcal{X}$$

$x_1 > a \rightarrow \mathcal{P}(\mathbf{x} \in y_1)$?
$x_1 \leq a \rightarrow \mathcal{P}(\mathbf{x} \in y_2)$?
Decision tree answer those questions



Decision Trees (DT)

- case split on feature using different criterion (Gini, entropy)
- no parameter tuning, easy to train
- sensitive to data variations, can overfit fast

Decision Trees: k-folding for training

- dataset is small $\Rightarrow$ sensitive to bad data balancing
- mitigation: train **multiple** models on **multiple** splits

Decision Trees: results

$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$: number of correct predictions

$Recall = \frac{TP}{TP+FN}$: number of detected anomalies

- training time is less than 2ms on a Intel I7-8850H
- sequential split: recall is 0%
- random split: recall is 94,3%, accuracy: 96,6%

Decision Trees: results

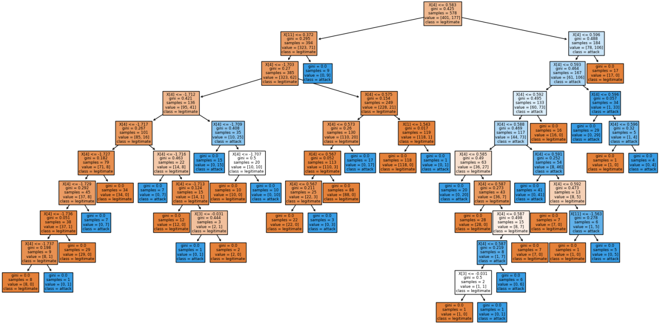
$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$: number of correct predictions

$Recall = \frac{TP}{TP+FN}$: number of detected anomalies

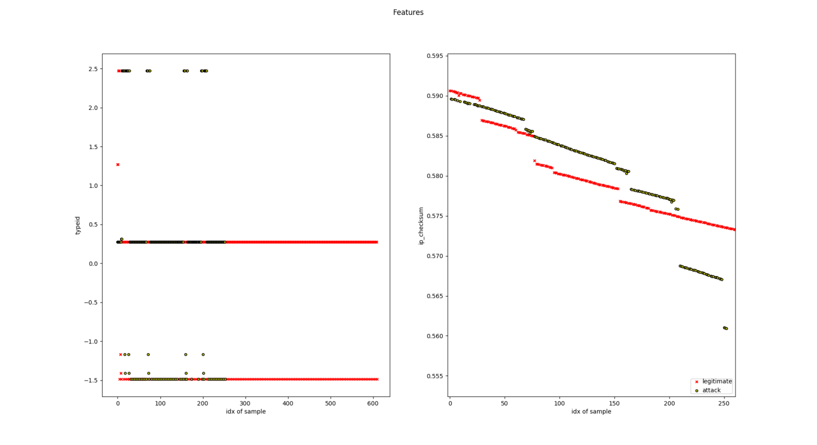
- training time is less than 2ms on a Intel I7-8850H
- sequential split: recall is 0%
- random split: recall is 94,3%, accuracy: 96,6%

Works surprinsingly well. Why?

Our decisions trees are overfitting



The two culprits features



Support Vector Machine (SVM)

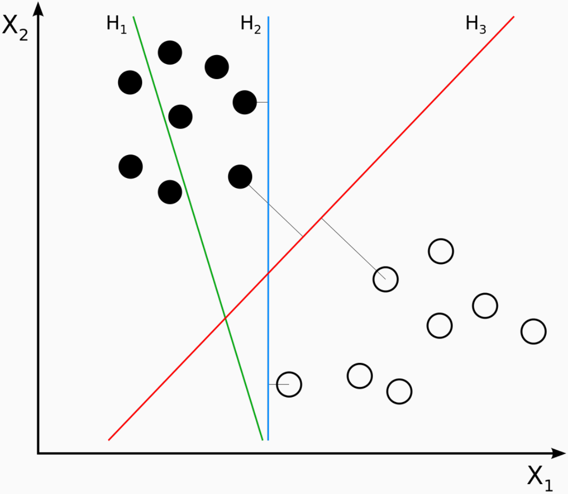
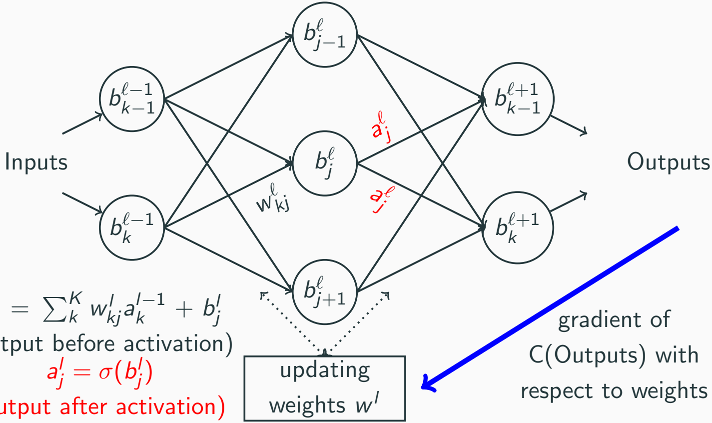


image source: wikipedia

SVM: results

- multiple kernels used
- accuracy: 79,7%, recall: 26,3%

Dense NN



$b_j^l = \sum_k^K w_{kj}^l a_k^{l-1} + b_j^l$

(output before activation)

$a_j^l = \sigma(b_j^l)$

(output after activation)

C (Cost function)

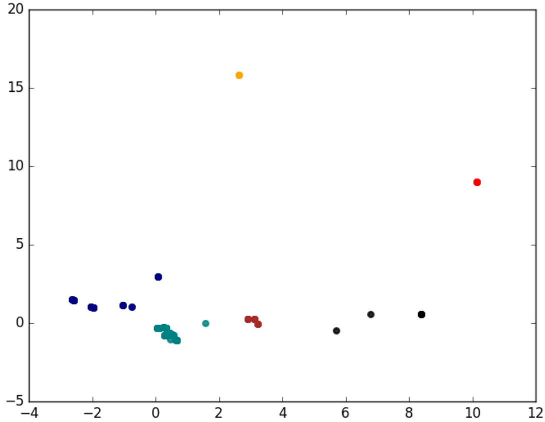
Dense NN: parameters and results

- fully connected network
- 4 layers and 10^6 neurons
- recall : 26,3%, accuracy : 90,9%

Take-home message

Supervised learning works because of over-fitting

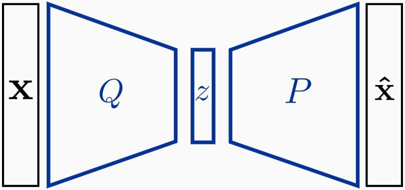
K-means and PCA



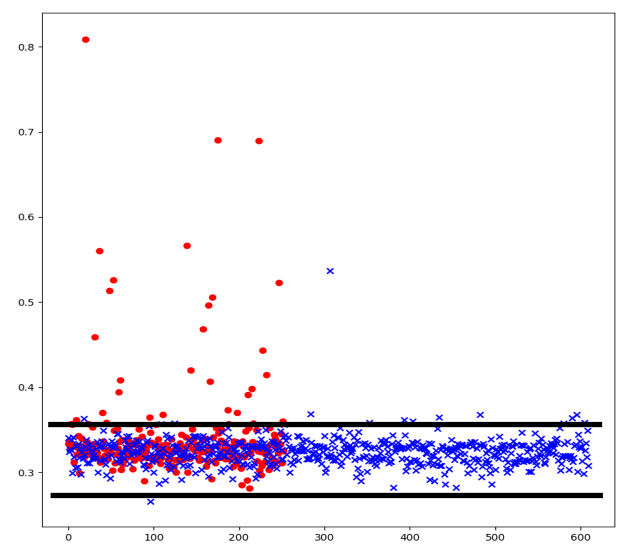
Variational Auto Encoder

Goal: learn the *probability distribution* of the input

Training objective: input **x**, learn a code s, an encoder Q and a decoder P such that $\hat{x}$ is a **good reconstruction**



(Bad) results for VAE



Take-home message (again)

1. Strong similarity between legitimate and attack packets
2. Unsupervised learning cannot separate efficiently

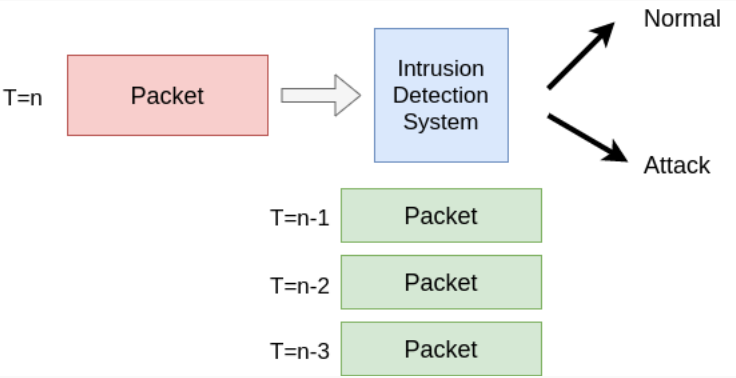
Summary of results

	Supervised				Unsupervised			
	Name	Acc.	Rec.	Time	Name	Acc.	Rec.	Time
No-seq.	SVM	80%	**26%**	<1ms	k-means	N/A	N/A	N/A
	DT	**96%**	**97%**	<1ms	AE	**48%**	80%	5min
	DNN	**91%**	**26%**	2min	VAE	N/A	N/A	N/A

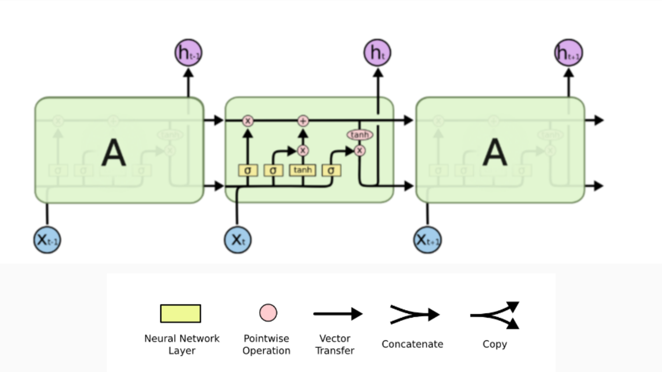
ML with sequences

Sequence classification

- order is important and must be respected
- predicting a class label for a given input sequence
- limitation of classical ML and MLP: Unaware of temporal structure

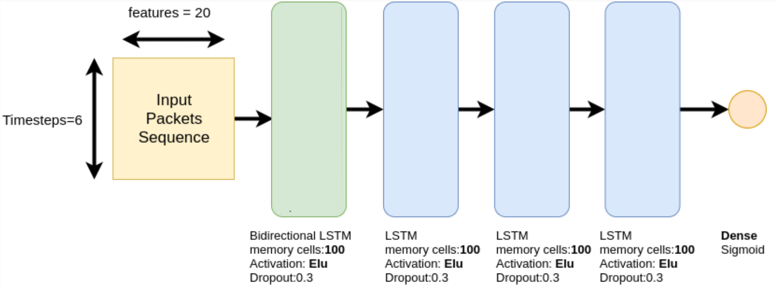


Supervised sequence classification: LSTM



- recurrent connections
- avoid the problems that prevent the training and scaling of other RNN
- memory cells contain weights and gates

IDS using Bidirectional LSTM



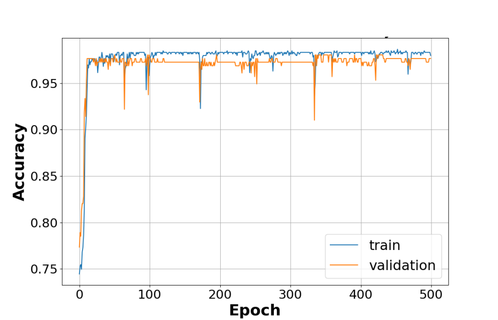
- **loss**:binary cross-entropy, **optimizer**: Adam
- **epoch**: 500, **batch**: 20
- **train**: 595 ($\approx$ 155 anomalies), **test**: 256 ($\approx$ 65 anomalies)
- **training time**: 5min (no GPU)

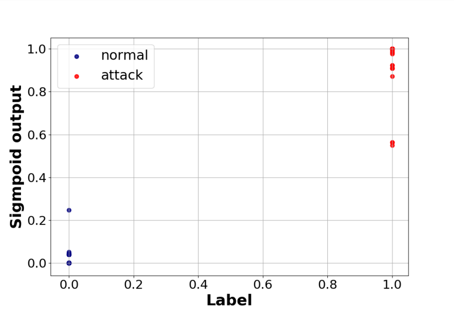
Evaluation: the beginner's mistake

- fit to training; evaluate on test; report skill: Wrong !
- deep learning models are stochastic
- LSTM's use randomness while being fit on a dataset
- same model may give different predictions

```
scores=list()
for i in repeats:
    train, test = random_split(data)
    model.fit(train.X,train.y)
    predicitons=model.predict(test.X)
    skill=compare(test.y,predictions)
    scores.append(skill)
final_skill=mean(scores)
```

IDS using Bidirectional LSTM: results





IDS using Bidirectional LSTM: results

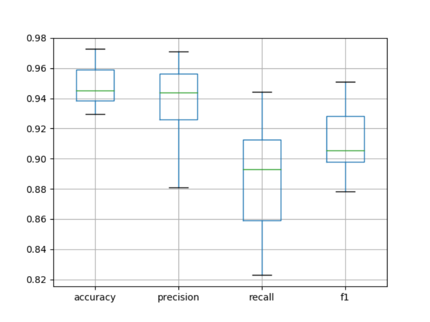
$$accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

$$precision = \frac{TP}{TP+FP}$$

$$recall = \frac{TP}{TP+FN}$$

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}$$

$$\text{Confusion matrix} = \begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}$$



$$\text{Confusion matrix} = \begin{bmatrix} 183 & 4 \\ 3 & 66 \end{bmatrix}$$

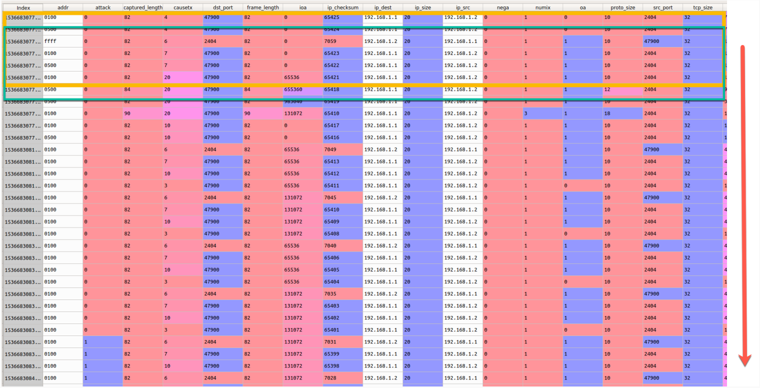
IDS using unsupervised learning

- what if you have no labelled data at all?
- binary analysis requires hours of fingerprinting and study per sample
- incident investigation requires huge resources and bureaucratic layers to triage
- infers hidden latent structure from unlabelled training data
- objective: learn from unlabelled data while respecting the temporal order

IDS using unsupervised learning: strategy

- data preparation
- build an auto-encoder on the normal (negatively labelled) data
- use it to reconstruct a new sample
- if the reconstruction error is high, we label it as an attack

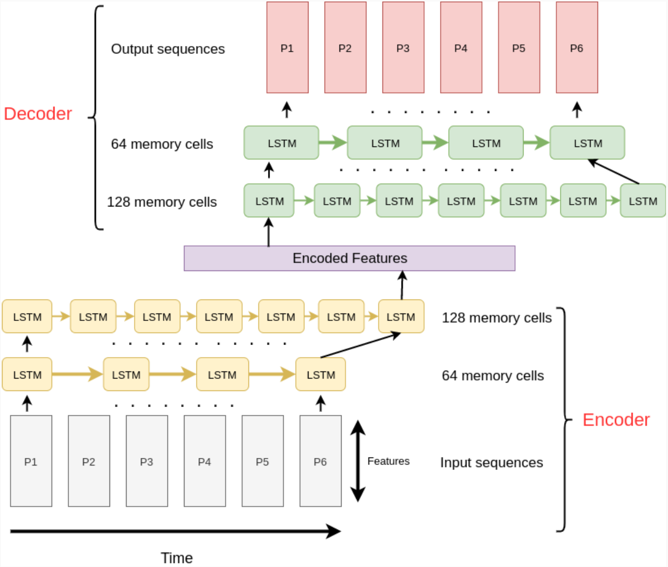
IDS using unsupervised learning: data preparation



Index	addr	attack	captured_length	causetx	dst_port	frame_length	ioa	ip_checksum	ip_dest	ip_size	ip_src	nega	numix	oa	proto_size	src_port	tcp_size
1536683077...	0100	0	82	4	47900	82	0	65425	192.168.1.1	20	192.168.1.2	0	1	0	10	2404	32
1536683077...	0500	0	82	4	47900	82	0	65424	192.168.1.1	20	192.168.1.2	0	1	0	10	2404	32
1536683077...	ffff	0	82	6	2404	82	0	7059	192.168.1.2	20	192.168.1.1	0	1	1	10	47900	32
1536683077...	0100	0	82	7	47900	82	0	65423	192.168.1.1	20	192.168.1.2	0	1	1	10	2404	32
1536683077...	0500	0	82	7	47900	82	0	65422	192.168.1.1	20	192.168.1.2	0	1	1	10	2404	32
1536683077...	0100	0	82	20	47900	82	65536	65421	192.168.1.1	20	192.168.1.2	0	1	1	10	2404	32
1536683077...	0500	0	84	20	47900	84	655360	65418	192.168.1.1	20	192.168.1.2	0	1	1	12	2404	32
1536683077...	0500	0	82	20	47900	82	983040	65419	192.168.1.1	20	192.168.1.2	0	1	1	10	2404	32
1536683077...	0100	0	90	20	47900	90	131072	65410	192.168.1.1	20	192.168.1.2	0	3	1	18	2404	32
1536683077...	0100	0	82	10	47900	82	0	65417	192.168.1.1	20	192.168.1.2	0	1	1	10	2404	32
1536683077...	0500	0	82	10	47900	82	0	65416	192.168.1.1	20	192.168.1.2	0	1	1	10	2404	32
1536683081...	0100	0	82	6	2404	82	65536	7049	192.168.1.2	20	192.168.1.1	0	1	1	10	47900	32
1536683081...	0100	0	82	7	47900	82	65536	65413	192.168.1.1	20	192.168.1.2	0	1	1	10	2404	32
1536683081...	0100	0	82	10	47900	82	65536	65412	192.168.1.1	20	192.168.1.2	0	1	1	10	2404	32
1536683081...	0100	0	82	3	47900	82	65536	65411	192.168.1.1	20	192.168.1.2	0	1	0	10	2404	32
1536683081...	0100	0	82	6	2404	82	131072	7045	192.168.1.2	20	192.168.1.1	0	1	1	10	47900	32
1536683081...	0100	0	82	7	47900	82	131072	65410	192.168.1.1	20	192.168.1.2	0	1	1	10	2404	32
1536683081...	0100	0	82	10	47900	82	131072	65409	192.168.1.1	20	192.168.1.2	0	1	1	10	2404	32
1536683081...	0100	0	82	3	47900	82	131072	65408	192.168.1.1	20	192.168.1.2	0	1	0	10	2404	32
1536683083...	0100	0	82	6	2404	82	65536	7040	192.168.1.2	20	192.168.1.1	0	1	1	10	47900	32
1536683083...	0100	0	82	7	47900	82	65536	65406	192.168.1.1	20	192.168.1.2	0	1	1	10	2404	32
1536683083...	0100	0	82	10	47900	82	65536	65405	192.168.1.1	20	192.168.1.2	0	1	1	10	2404	32
1536683083...	0100	0	82	3	47900	82	65536	65404	192.168.1.1	20	192.168.1.2	0	1	0	10	2404	32
1536683083...	0100	0	82	6	2404	82	131072	7035	192.168.1.2	20	192.168.1.1	0	1	1	10	47900	32
1536683083...	0100	0	82	7	47900	82	131072	65403	192.168.1.1	20	192.168.1.2	0	1	1	10	2404	32
1536683083...	0100	0	82	10	47900	82	131072	65402	192.168.1.1	20	192.168.1.2	0	1	1	10	2404	32
1536683083...	0100	0	82	3	47900	82	131072	65401	192.168.1.1	20	192.168.1.2	0	1	0	10	2404	32
1536683083...	0100	1	82	6	2404	82	131072	7031	192.168.1.2	20	192.168.1.1	0	1	1	10	47900	32
1536683083...	0100	1	82	7	47900	82	131072	65399	192.168.1.1	20	192.168.1.2	0	1	1	10	2404	32
1536683083...	0100	1	82	10	47900	82	131072	65398	192.168.1.1	20	192.168.1.2	0	1	1	10	2404	32
1536683084...	0100	1	82	6	2404	82	131072	7028	192.168.1.2	20	192.168.1.1	0	1	1	10	47900	32

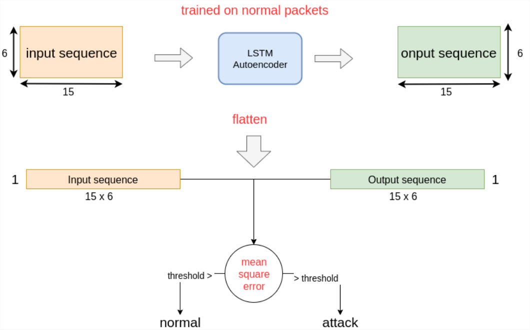
- the input to LSTMs are 3-dimensional arrays
- sliding window of size 6 and step = 1

IDS using unsupervised learning: LSTM auto-encoder

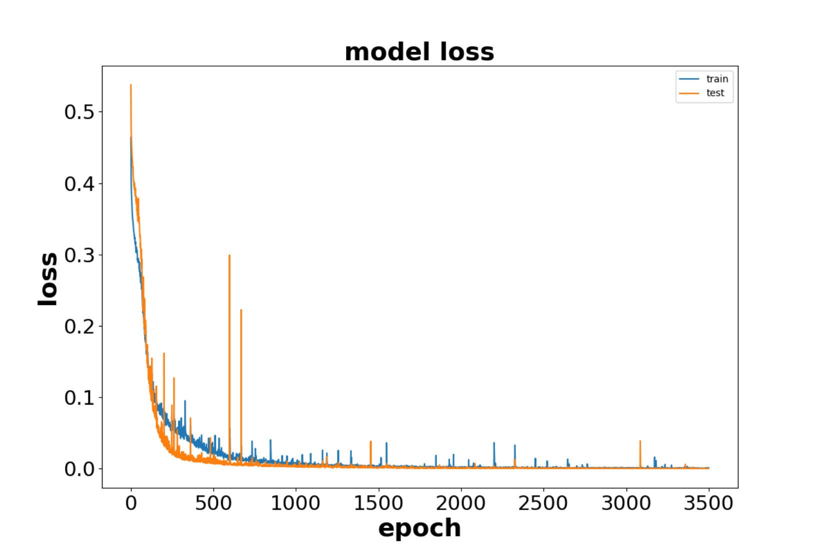


IDS using unsupervised learning: LSTM auto-encoder

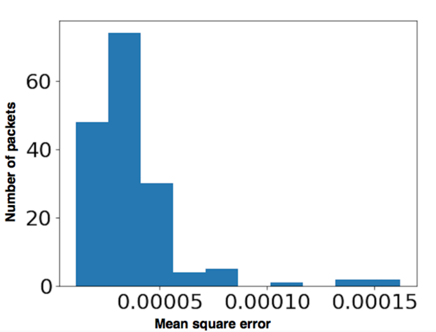
- trained on legitimate packets
- tested on legitimate and attack packets
- **epoch**: 3500, **batch**: 10
- **training time**: ≈30 min (no GPU)



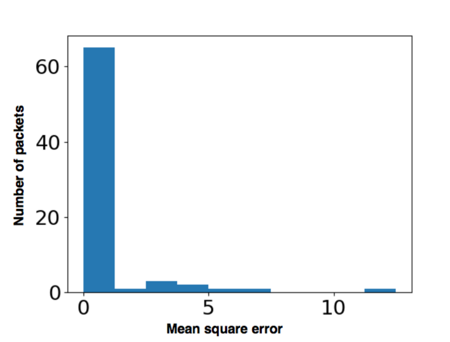
IDS using unsupervised learning: LSTM auto-encoder results



IDS using unsupervised learning: LSTM auto-encoder results



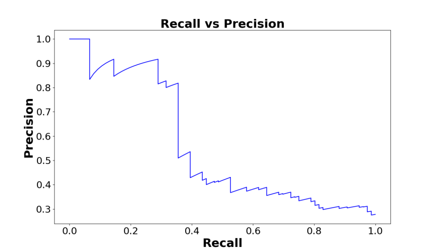
reconstruction error
of **normal** packets



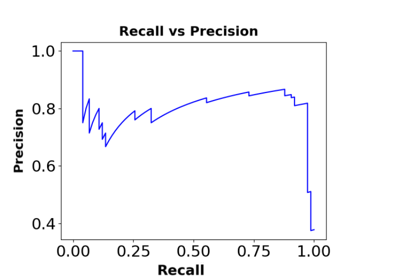
reconstruction error
of **attack** packets

IDS using unsupervised learning: LSTM auto-encoder results

IDS using unsupervised learning: LSTM auto-encoder results

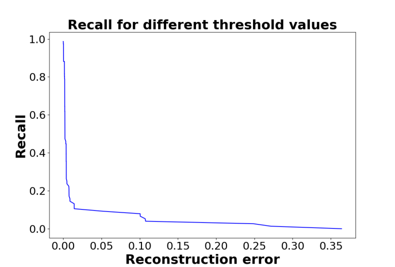


auto-encoder
(no sequence)

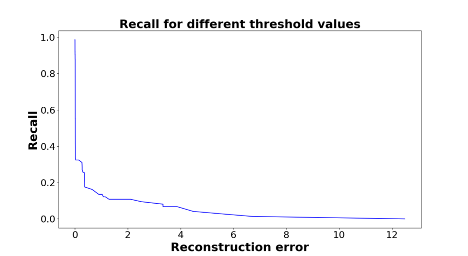


LSTM auto-encoder
(sequence)

IDS using unsupervised learning: LSTM auto-encoder results

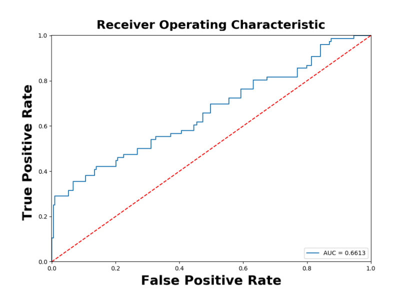


auto-encoder
(no sequence)

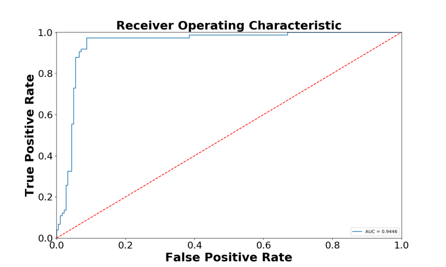


LSTM auto-encoder
(sequence)

IDS using unsupervised learning: LSTM auto-encoder results

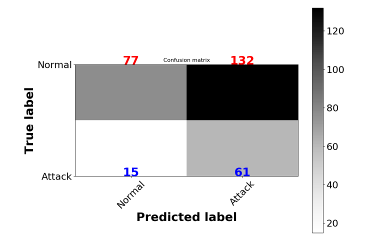


auto-encoder
(no sequence)

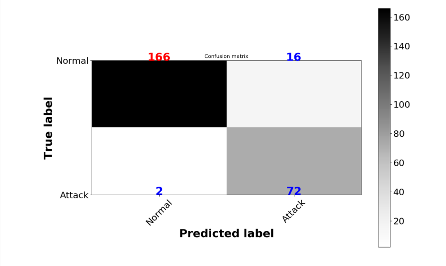


LSTM auto-encoder
(sequence)

IDS using unsupervised learning: LSTM auto-encoder results



auto-encoder
(no sequence)



LSTM auto-encoder
(sequence)

IDS using unsupervised learning: What can be done better on huge data?

- CNN LSTM Autoencoder
- LSTM Dropout (Dropout_U and Dropout_W)
- Gaussian-dropout layer
- SELU activation
- alpha-dropout with SELU activation

Conclusion

How did we tackle the problem using ML?

1. **Preliminary work**
   - understand the protocol specification and the attacker model
   - being able to identify (non-)legitimate packets

How did we tackle the problem using ML?

1. **Preliminary work**
   - understand the protocol specification and the attacker model
   - being able to identify (non-)legitimate packets
2. **Data analysis**
   - identify relevant fields (non-constant fields, principal component analysis...)
   - verify that legitimate/attack packets are balanced

How did we tackle the problem using ML?

1. **Preliminary work**
   - understand the protocol specification and the attacker model
   - being able to identify (non-)legitimate packets
2. **Data analysis**
   - identify relevant fields (non-constant fields, principal component analysis...)
   - verify that legitimate/attack packets are balanced
3. **Apply ML techniques with single or sequence of packets**
   - first, the simplest algorithms (SVM, decision trees, k-means)
   - then the more complex ones (DNN, LSTM, auto-encoders)

How did we tackle the problem using ML?

1. **Preliminary work**
   - understand the protocol specification and the attacker model
   - being able to identify (non-)legitimate packets
2. **Data analysis**
   - identify relevant fields (non-constant fields, principal component analysis...)
   - verify that legitimate/attack packets are balanced
3. **Apply ML techniques with single or sequence of packets**
   - first, the simplest algorithms (SVM, decision trees, k-means)
   - then the more complex ones (DNN, LSTM, auto-encoders)
4. **Evaluation of the results**
   - presentation of results
   - explanation of success/failures (e.g., identify over-fitting)

The different algorithm used

	Supervised				Unsupervised			
	Name	Acc.	Rec.	Time	Name	Acc.	Rec.	Time
No-seq.	SVM	80%	**26%**	<1ms	k-means	N/A	N/A	N/A
	DT	**96%**	**97%**	<1ms	AE	**48%**	80%	5min
	DNN	**91%**	**26%**	2min	VAE	N/A	N/A	N/A
Seq.	LSTM	**94%**	**89%**	5min	LSTM AE	**91%**	**97%**	30min

Results and advices

Results in a nutshell:

- considering sequences is mandatory
- similar results between unsupervised and supervised ML

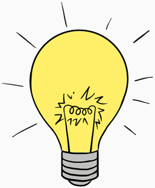
Few advices for re-using our approach:

- generate an adapted dataset
- consider a more realistic network
- test the simplest algorithms first

Results and advices

Results in a nutshell:

- considering sequences is mandatory
- similar results between unsupervised and supervised ML



Few advices for re-using our approach:

- generate an adapted dataset
- consider a more realistic network
- test the simplest algorithms first

Thank you for your attention