
STRING
SIFTER
DATASCIENCE
PRESENTED BY
FIREEYE
FLARE

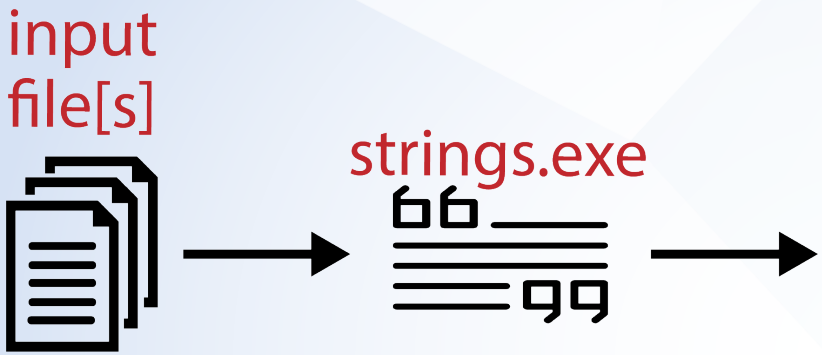
What is a String?

- Persists in compilation
- ASCII/Narrow
  - N characters + NULL
  - No file format, context
  - `0x31 0x33 0x33 0x37 0x00`
    - ‘1337’, right?
  - Not necessarily:
    - Memory addresses
    - CPU instructions
    - Data used by the program
- Unicode/Wide
  - 2 bytes, double-NULL terminated

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	|
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

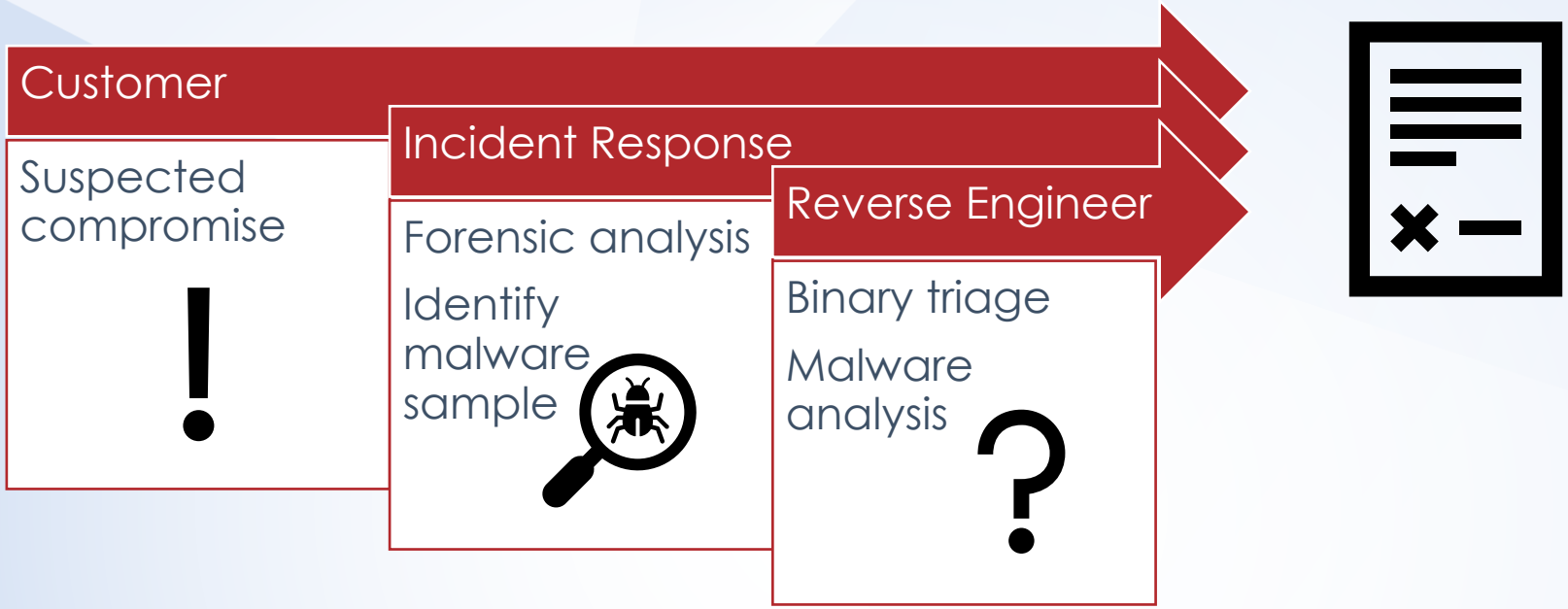
```
         0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
0000h: 48 00 65 00 6C 00 6C 00 6F 00 20 00 57 00 6F 00  H.e.l.l.o. .W.o.
0010h: 72 00 6C 00 64 00 21 00 00 00                    r.l.d.!...
```

The *Strings* Program




```
!This program cannot be run in DOS mode.
??3@YAXPAX@Z
??2@YAPAXI@Z
__CxxFrameHandler
_except_handler3
WSAStartup() error: %d
User-Agent: Mozilla/4.0 (compatible; MSIE 6.00; Windows
NT 5.1)
GetLastInputInfo
SeShutdownPrivilege
%s\IEXPLORE.EXE
SOFTWARE\Microsoft\Windows\CurrentVersion\App
Paths\IEXPLORE.EXE
[Machine IdleTime:] %d days + %.2d:%.2d:%.2d
[Machine UpTime:] %-.2d Days %-.2d Hours %-.2d Minutes
%-.2d Seconds
ServiceDll
SYSTEM\CurrentControlSet\Services\%s\Parameters\
if exist "%s" goto selfkill
del "%s"
attrib -a -r -s -h "%s"
Inject '%s' to  PID '%d' Successfully!
\cmd.exe /c
Hi,Master [%d/%d/%d %d:%d:%d]
```

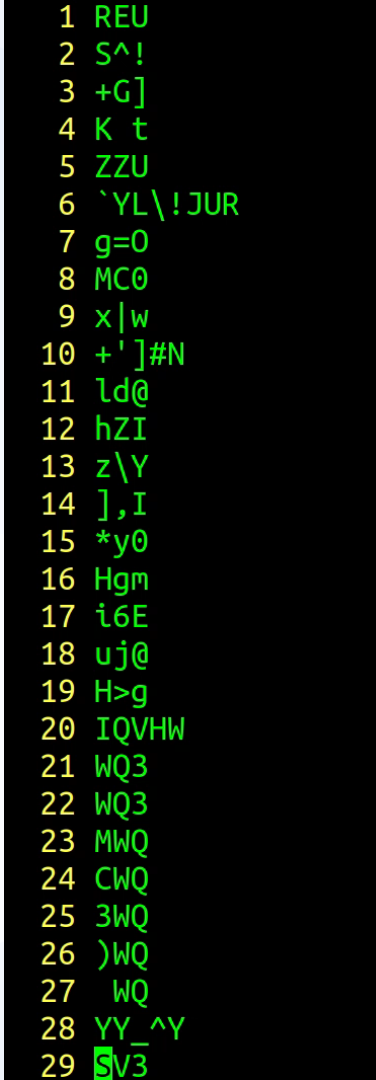
Malware Triage



+SOC analysts, red teamers, malware researchers, n00bs, experts

Strings in Practice – Static Analysis

- Running *Strings* on larger binaries produces **tens of thousands** of strings.
- *Strings* produces a ton of noise mixed in with **important information**
- Knowing which strings are relevant often requires **highly experienced analysts**.
- Relevance is subjective and its definition can **vary significantly** across analysts.

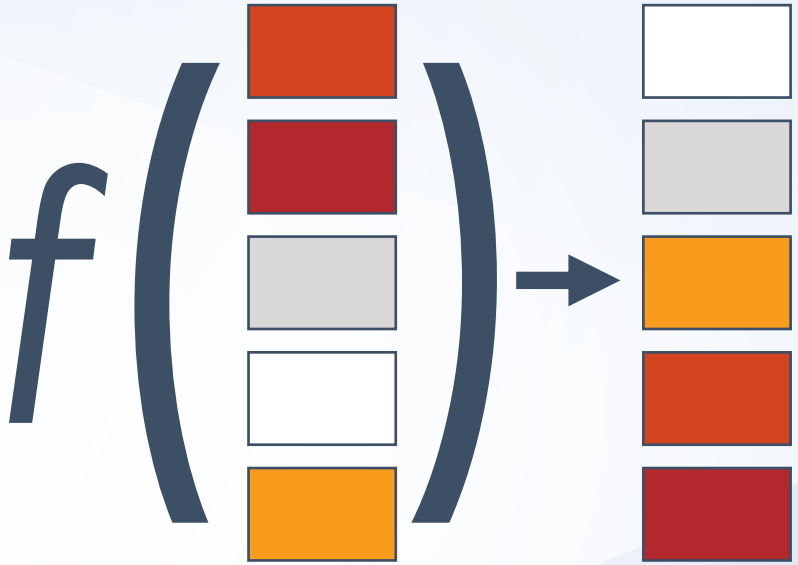


Hypothesis and Goals

- Develop a tool that can:
  - efficiently identify and prioritize strings
  - based on relevance for malware analysis
- StringSifter should:
  - be easy to use
  - generalize across:
    - roles, use cases, downstream apps
  - save time and money
- How does it work?

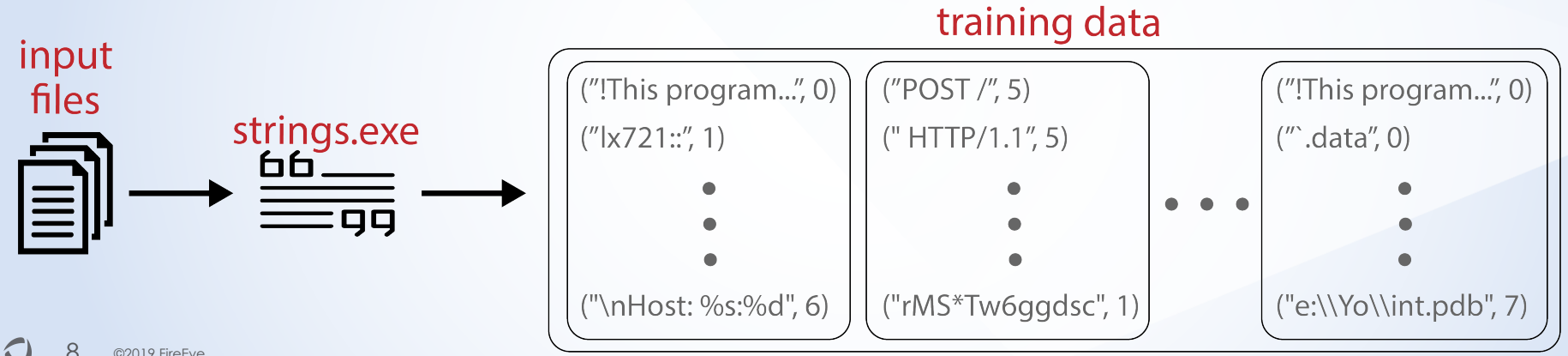
Learning to Rank

- Create optimal ordering of a list of items
- Precise individual item scores less important than their relative ordering
- In classification, regression, clustering we predict a class or single score
- LTR rarely applied in security applications



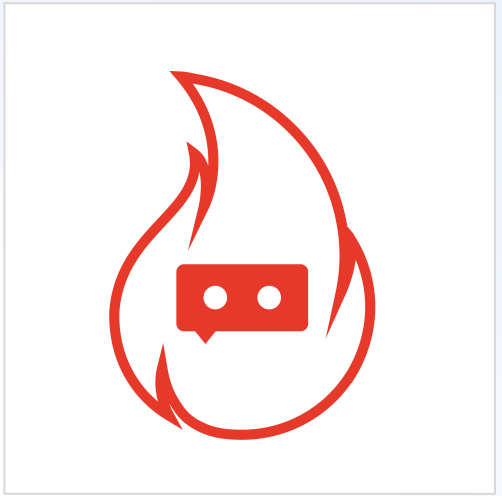
LTR as Supervised Learning

- Rank items within unseen lists in a similar way to rankings within training lists
- Each item associated with a set of features and an ordinal integer label
- Ordinal label is the teaching signal that encodes relevance level



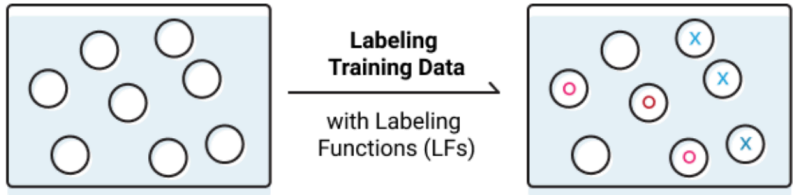
EMBER Dataset

- **E**ndgame **M**alware **BE**nchmark for **R**esearch
  - v1 (1.1 million PE files scanned on or before 2017)
    - https://arxiv.org/abs/1804.04637
    - https://github.com/endgameinc/ember
  - 400k train + test malware binaries from v1
    - malware defined as > 40 VT vendors say malicious
- Ran *Strings* on 400k malware binaries
  - produced 3+ billion ASCII + Unicode strings (24+ GB)
  - performed sampling, stratified by malware family
  - labeled according to weak supervision



Weak Supervision

- Data Labeling Bottleneck
- Ordinal Labeling Functions
  - `ABSTAIN = -1`
  - `VERY_IRRELEVANT = 0`
  - `IRRELEVANT = 1`
  - `SEMI_IRRELEVANT = 2`
  - `NEUTRAL = 3`
  - `SEMI_RELEVANT = 4`
  - `RELEVANT = 5`
  - `VERY_RELEVANT = 6`
- *cardinality* = 7, tie goes `NEUTRAL`
- Apply 70+ LFs over input strings, generate probabilistic labels




```
@snorkel.labeling.labeling_function()
def has_variable_name(string_i):
    has_uppercase_var_name = uppercase_var_name.search(string_i)
    has_period_delimited_var_name = \
        period_delimited_var_name.search(string_i)
    has_no_extension = not _has_extension_helper(string_i)
    return RELEVANT if (has_uppercase_var_name or
                        (has_period_delimited_var_name and
                         has_no_extension)) else ABSTAIN

@snorkel.labeling.labeling_function()
def has_format_specifier(string_i):
    return RELEVANT if _has_format_specifier_helper(string_i) else ABSTAIN

@snorkel.labeling.labeling_function()
def has_padding_string(string_i):
    return VERY_IRRELEVANT if _substring_match_bool(string_i,
                                      ['PADDING']) else ABSTAIN

@snorkel.labeling.labeling_function()
def has_malware_identifier(string_i):
    return RELEVANT if _substring_match_bool(string_i.lower(),
                                     constants['malware identifiers']) else ABSTAIN

@snorkel.labeling.labeling_function()
def is_registry(string_i):
    return VERY_RELEVANT if _substring_match_bool(string_i,
                                     constants['regs']) else ABSTAIN
```

https://github.com/snorkel-team/snorkel

snorkel

LF Examples

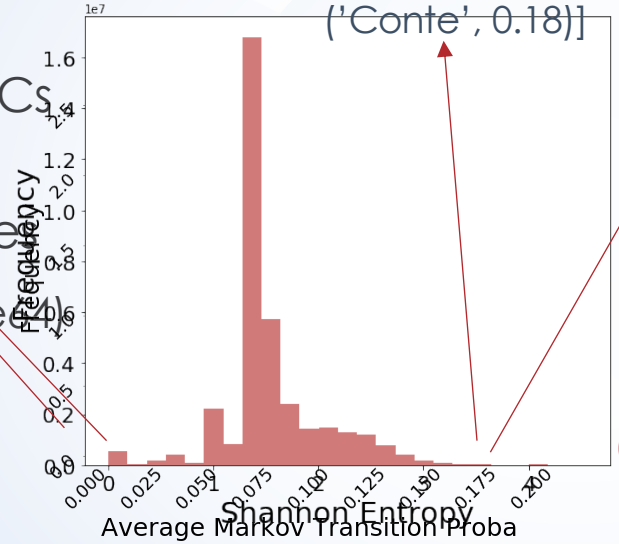
- Natural Language Processing
  - Entropy rate
  - KL Divergence
  - Markov model

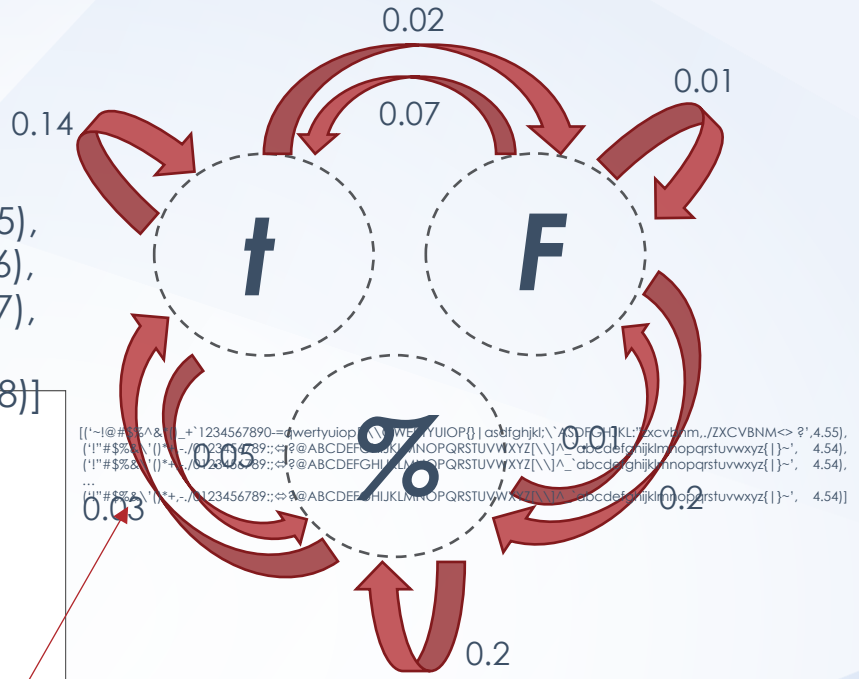
```
[('QQQQ',     0),
 ('QQQQ',     0),
 ('RPPPP',    0),
 ('PRPPP',    0),
 ('WWWWW',    0),
 ('SSSSS',    0),
 ('SSSSS',    0),
 ...
 ('o\ti0{',   0)]
```

- Host, Network IoCs
  - Malware Regexes
  - encodings (base64)
  - format specifiers
  - user agents

```
[('\\\t\\\t\\',  0),
 ('_&\t\\',      0),
 ('\\\\\\',      0),
 ('a\t\\=Q',     0),
 ...
 ('o\ti0{',      0)]
```

```
[('winter',  0.15),
 ('Finger', 0.16),
 ('Inter',   0.17),
 ...
 ('Conte', 0.18)]
```

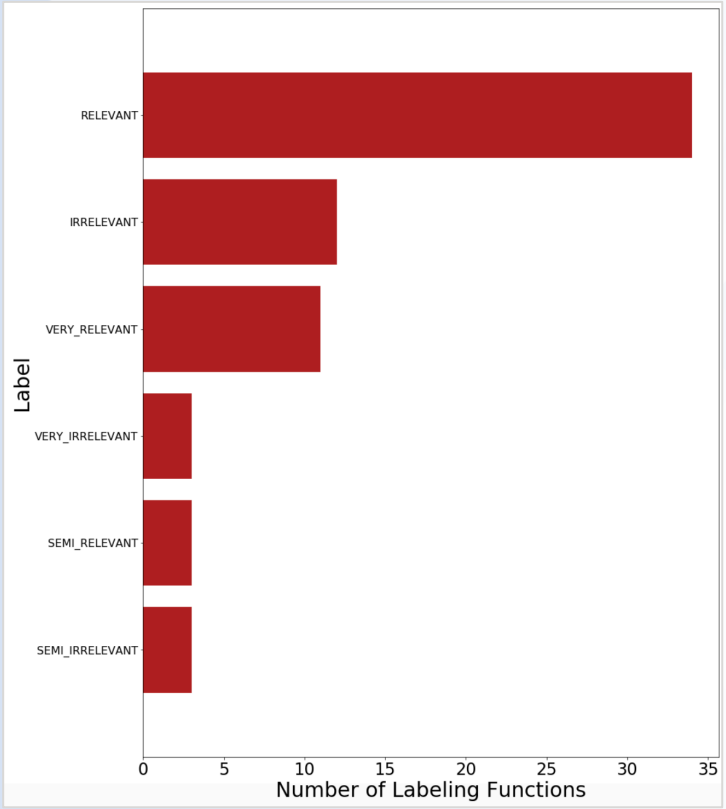


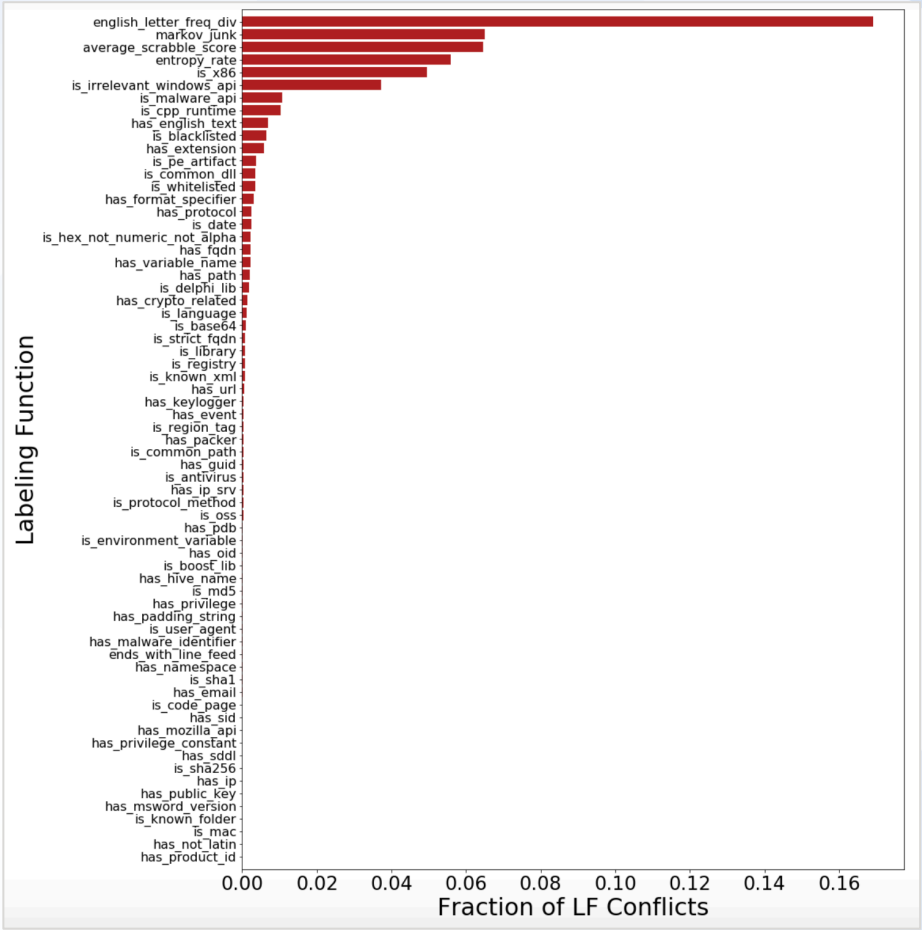



```
[('~!@#$%&\'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\\]^_`abcdefghijklmnopqrstuvwxyz{|}~',  4.55),
 ('!"#$%&\'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\\]^_`abcdefghijklmnopqrstuvwxyz{|}~',  4.54),
 ('!"#$%&\'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\\]^_`abcdefghijklmnopqrstuvwxyz{|}~',  4.54),
 ...
 ('!"#$%&\'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\\]^_`abcdefghijklmnopqrstuvwxyz{|}~',  4.54)]
```

```
http://evil.com                    0.018
\\SOFTWARE\\include\\evil.pdb       0.014
t%Ft                               0.007
Vr}Y                               0.001
```

LFApplier



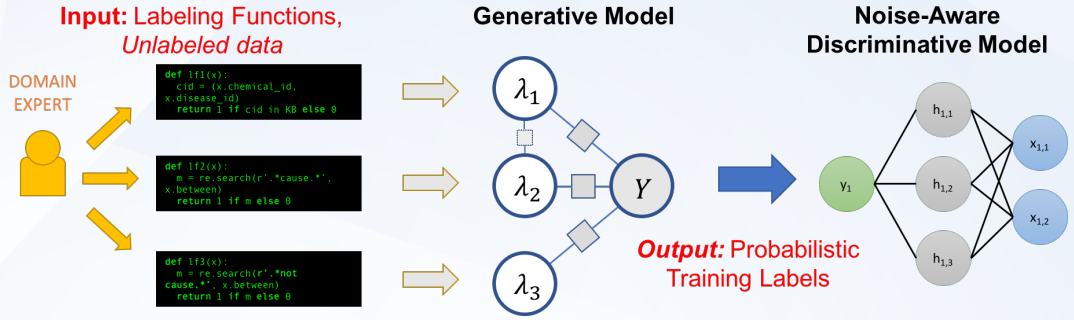


Learning a LabelModel

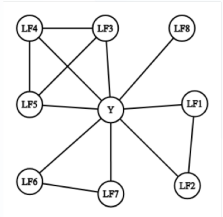
- LFS have different:
  - Accuracies
  - Correlations
  - Certainties

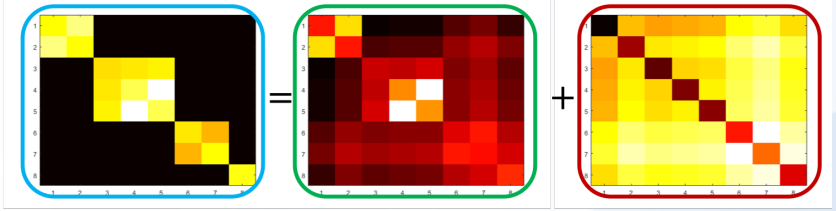
- Learning a LabelModel
  - Inverse generalized covariance matrix of LFs
  - Matrix completion (Robust PCA)

- Snorkel @ *ICML* '19
  - https://arxiv.org/abs/1903.05844



$$(\Sigma^{-1})_O = \Sigma_O^{-1} + ZZ^T$$



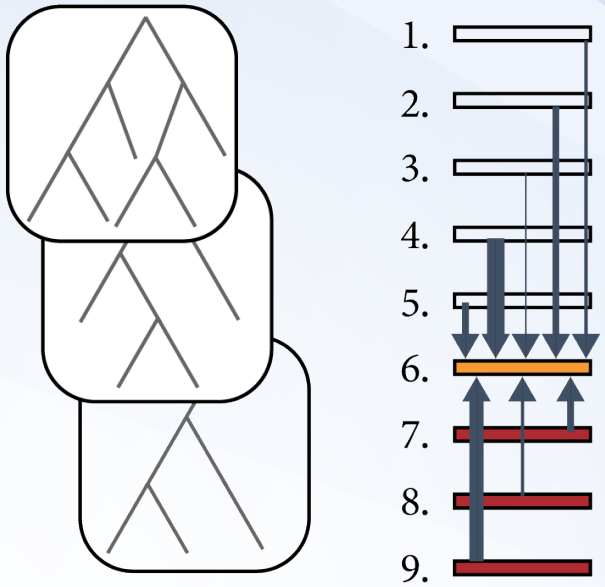


https://github.com/snorkel-team/snorkel

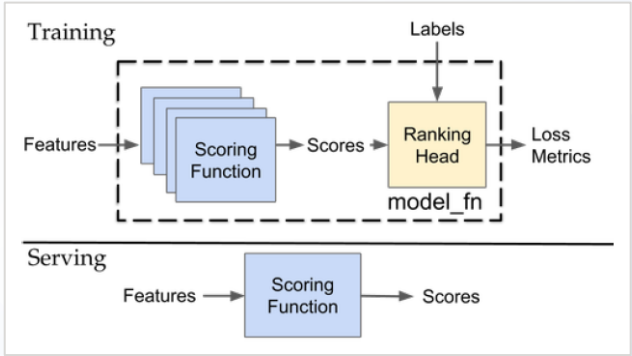
snorkel

LTR Models

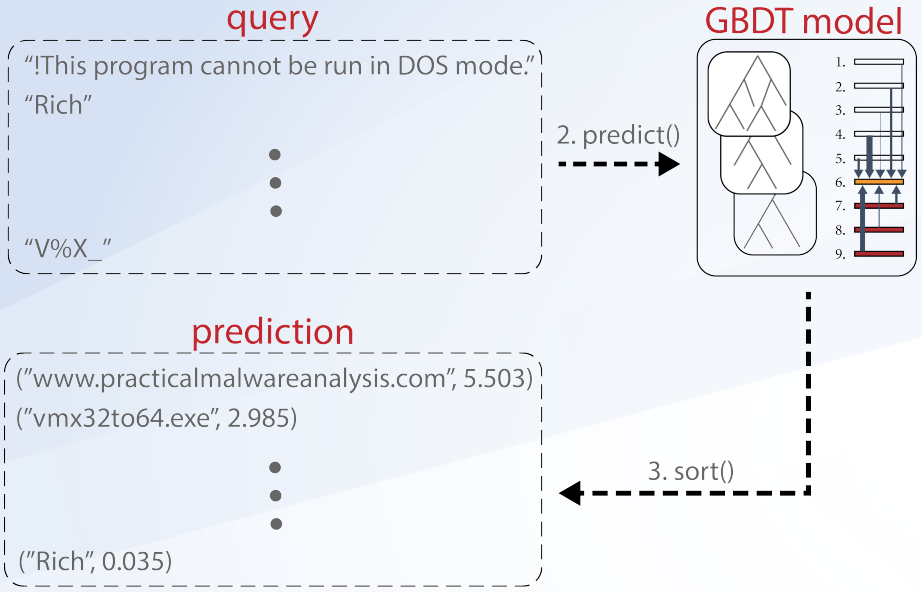
- Gradient Boosted Decision Trees (GBDTs)
  - combine outputs from multiple Decision Trees
  - reduce loss using gradient descent
  - weighted sum of trees' predictions as ensemble
  - LightGBM (https://github.com/microsoft/LightGBM)
    - Histogram-binned GBDTs with LTR obj. function



- Neural networks
  - tf-ranking (https://github.com/tensorflow/ranking)
  - Scoring Function: defines the network
  - Loss (e.g. pairwise logistic), Metrics (e.g. precision)

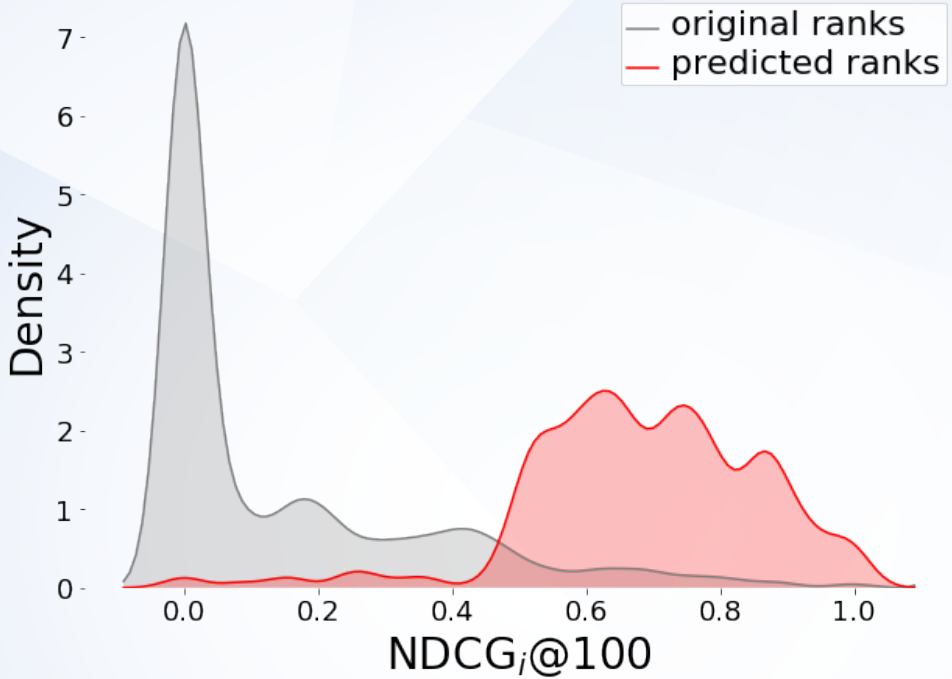


Evaluation



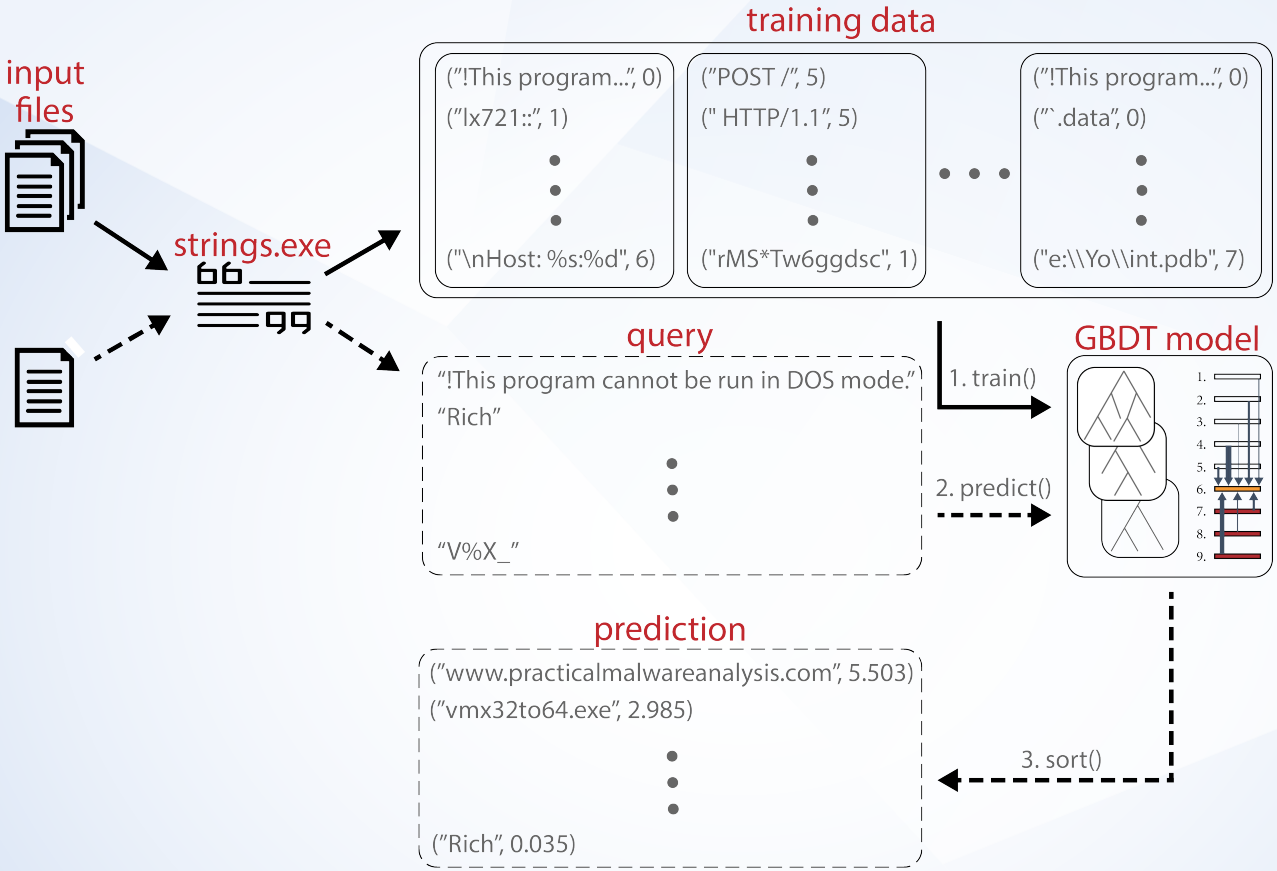
- **N**ormalized **D**iscounted **C**umulative **G**ain
  - **N**ormalized: divide DCG by ideal DCG on a ground truth holdout dataset
  - **D**iscounted: divides each string's predicted relevance by a monotonically increasing function (log of its ranked position)
  - **C**umulative: the cumulative gain or summed total of every string's relevance
  - **G**ain: the magnitude of each string's relevance

Kernel Density of Test NDCG Scores



StringSifter performs well on a holdout set of 7+ years of FLARE malware reports.

Putting it All Together



Demo

Open Sourcing StringSifter

- The tool is now live
  - Command line and Docker tools
  - `flarestrings <my_sample> | rank_strings`
- FLOSS outputs, live memory dumps
- Weak Supervision for Cybersecurity
  - Other label-starved problems?
- In the works
  - more labeling functions, mach-o + ELF files

https://github.com/fireeye/stringsifter

```
pip install stringsifter
```

@phtully