Detection of malware using self-attention mechanism and strings

National Defense Academy of Japan

Satoki Kanno

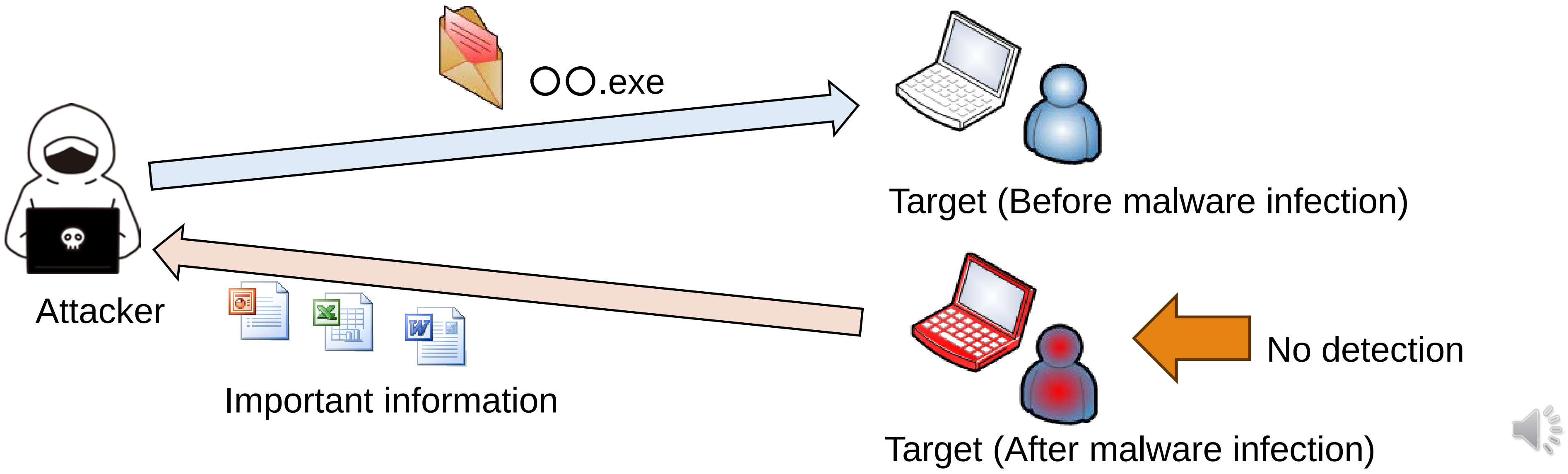
Mamoru Mimura

Contents

1. Background (1 / 4)

Targeted attacks

- This is one of the ways in which organizations and individuals are targeted, for example, to steal important information
- Targeted attacks often contain malware in the form of executable files
- Malware must be analyzed and detected to prevent the attacks.



1. Background (2 / 4)

How to do malware analysis

There are 3 main methods

1. Dynamic analysis

   The method to run malware and analyze it based on its behavior

2. Static analysis

   The method to analyze source code without running malware

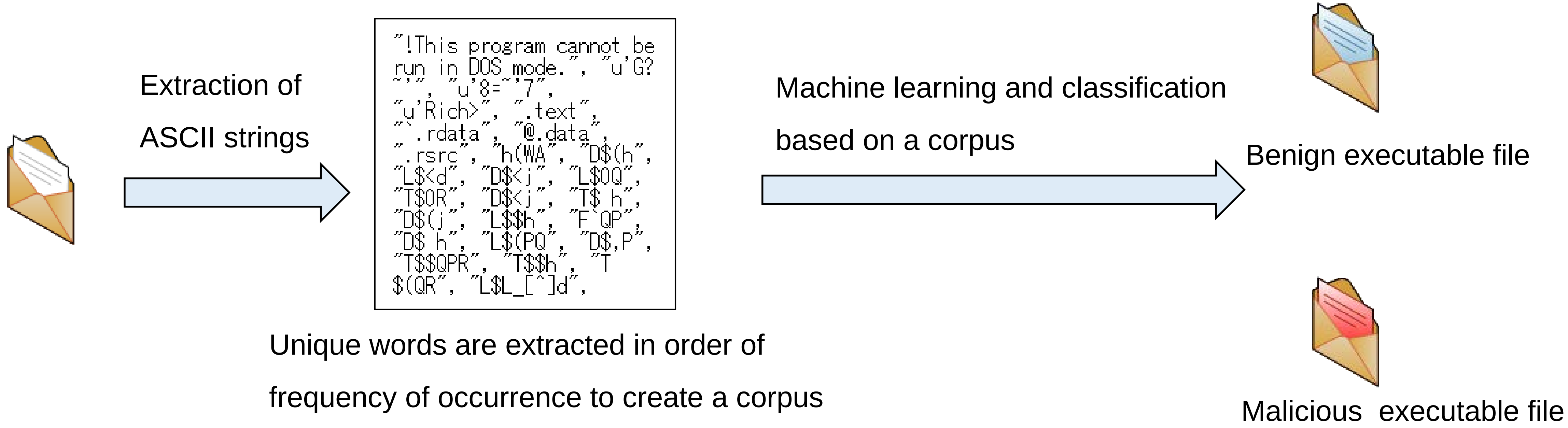
3. Surface analysis

   The method to analyze information (file name, hash, string, etc.) contained in a file without running malware

1. Background (3 / 4)

○Surface analysis

- A method has been proposed to extract features from the results of surface analysis of executable files and classify them using machine learning



Unclear which words contribute to malware detection

1. Background (4 / 4)

Purpose of the study

1. To clarify whether consecutive strings are considered when creating the corpus.
2. To identify the features that contribute to malware detection

Contribution of the study

1. LSTM with self-attention mechanism was used to detect malware, with a maximum F-measure of 0.904
2. We confirmed that removing non-consecutive ASCII strings from the corpus has a certain effect.
3. We have identified the impact of self-attention mechanisms on ASCII strings and confirmed that there are words of high importance that contribute to detection

2. Related Work (1 / 1)

No.	Paper Title	ASCII		NLP	Attention
		All	Some		
1	**Mastjik, F., et al.: Comparison of Pattern Matching Techniques on Identification of Same Family Malware, International Journal of Information Security Science, Vol. 4, No. 3, pp. 104–111 (2015).**		○		
2	**Kolosnjaji, et al.: Empowering convolutional networks for malware classification and analysis, 2017 International Joint Conference on Neural Networks, IJCNN 2017, Anchorage, AK, USA, May 14-19, 2017, pp. 3838–3845 (2017).**		○	○	
3	**Yakura, H., Shinozaki, S., et al.: Neural Malware Analysis with Attention Mechanism, Comput. Secur., Vol. 87, No. C (2019).**	○		○	○
4	**Ye, Y., Chen, et al.: an interpretable string based malware detection system using SVM ensemble with bagging, Journal in Computer Virology, Vol. 5, No. 4, pp. 283–293 (2009).**	○			
5	**Mimura, M. and Ito, R.: Applying NLP techniques to malware detection in a practical environment, Int. J.Inf. Sec., Vol. 21, No. 2, pp. 279–291 (2022).**		○	○	
	This study		○	○	○

3. Related Technique (1 / 3)

Bag-of-Words (BoW)

- A model that counts the number of occurrences of a word in a sentence and represents it as a vector
- This model does not take word order into account

e.g.
Sentence 1 : I have a pen
Sentence 2 : You have an apple

Create unique word dictionaries based on all documents
Corpus = [‘I’ , ’have’ , ’a’ , ’pen’ , ’You’ , ’an’ , ’apple’]

Convert sentences into vectors according to word dictionaries and word frequencies
Sentence 1 : I have a pen → [1, 1, 1, 1, 0, 0, 0]
Sentence 2 : I have an apple → [0, 1, 0, 0, 1, 1, 1]

3. Related Technique (2 / 3)

Words are converted to corresponding IDs

- A model that assign IDs to dictionaries as they are created and represent vectors
- This model takes word order into account

e.g.
Sentence 1 : I have a pen
Sentence 2 : You have an apple

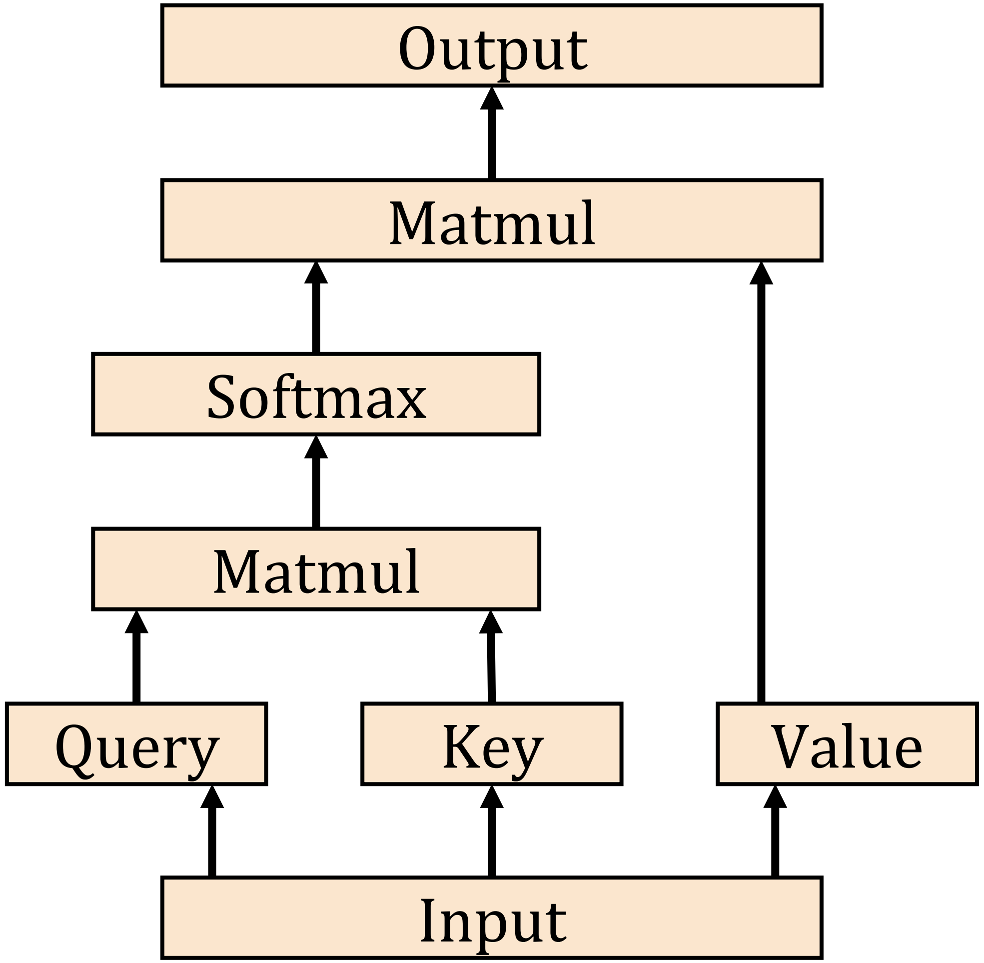
Create unique word dictionaries based on all documents
Corpus = [‘1:I’ , ’2:have’ , ’3:a’ , ’4:pen’ , ’5:You’ , ’6:an’ , ’7:apple’]

Convert sentences into a vector by assigning IDs according to a word dictionary
Sentence 1 : I have a pen → [1, 2, 3, 4]
Sentence 2 : I have an apple → [1, 2, 6, 7]

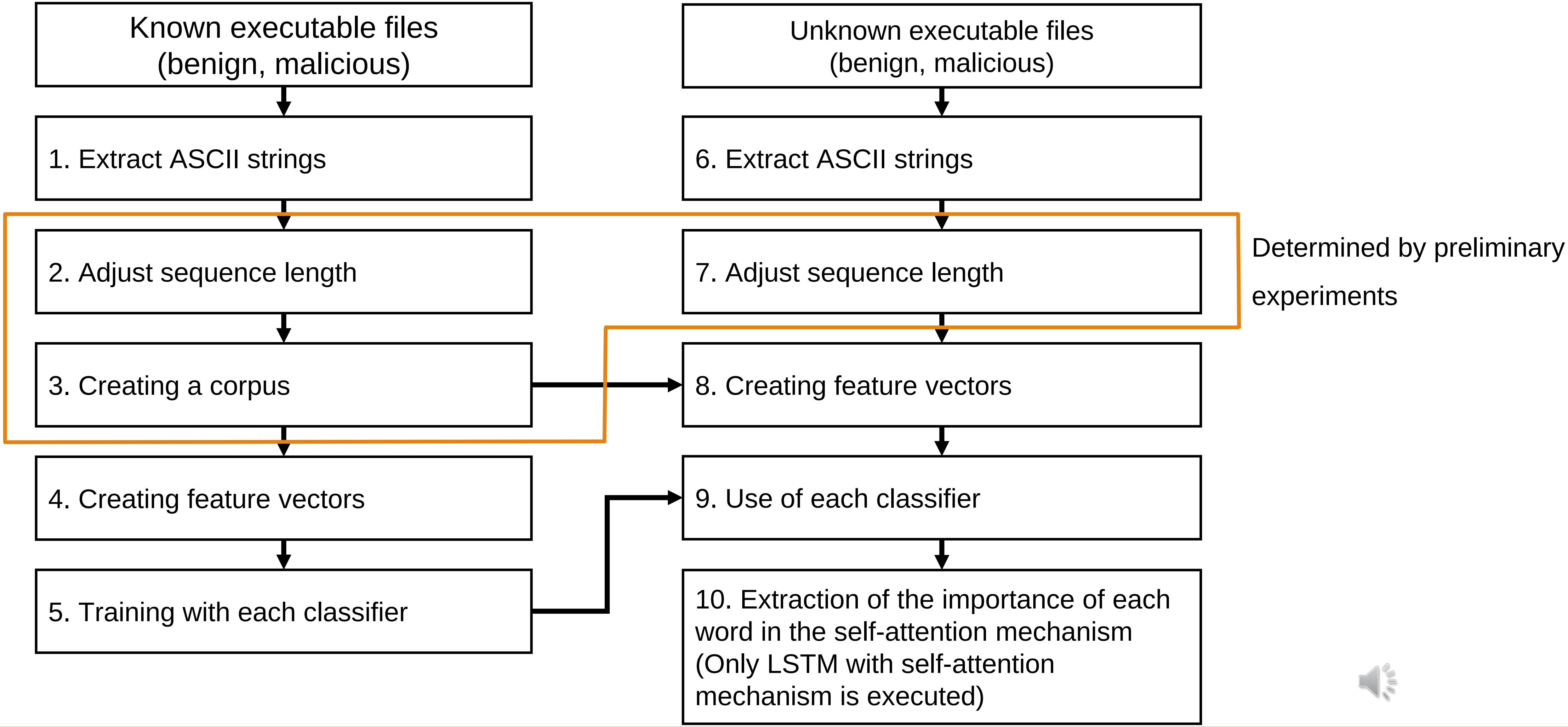
3. Related Technique (3 / 3)

Self-attention mechanism

- Self-attention mechanism is a method of focusing on and expressing the element-by-element relationships of input data
- There are three elements: Query, Key, Value
- Query is the information you want to search for in the input data
- Key is used to calculate the relevance of the Query to the object to be searched
- Value is used to output the appropriate Value based on Key



4. Experimental Method (1 / 3)



4. Experimental Method (2 / 3)

How to create a corpus

1. Extract ASCII strings of n (n≥1) or more consecutive characters from the training data

2. Extract words in order of frequency of occurrence

➢ We experimented with five different corpuses, this time to find words whose meanings we could understand.

e.g.：Case for creating a corpus of words with 2 or more consecutive ASCII strings and the top 3 words

['This:1', 'program:4', 'cannot:2', '@:5', 'DOS:3'] ['Word:Frequency']

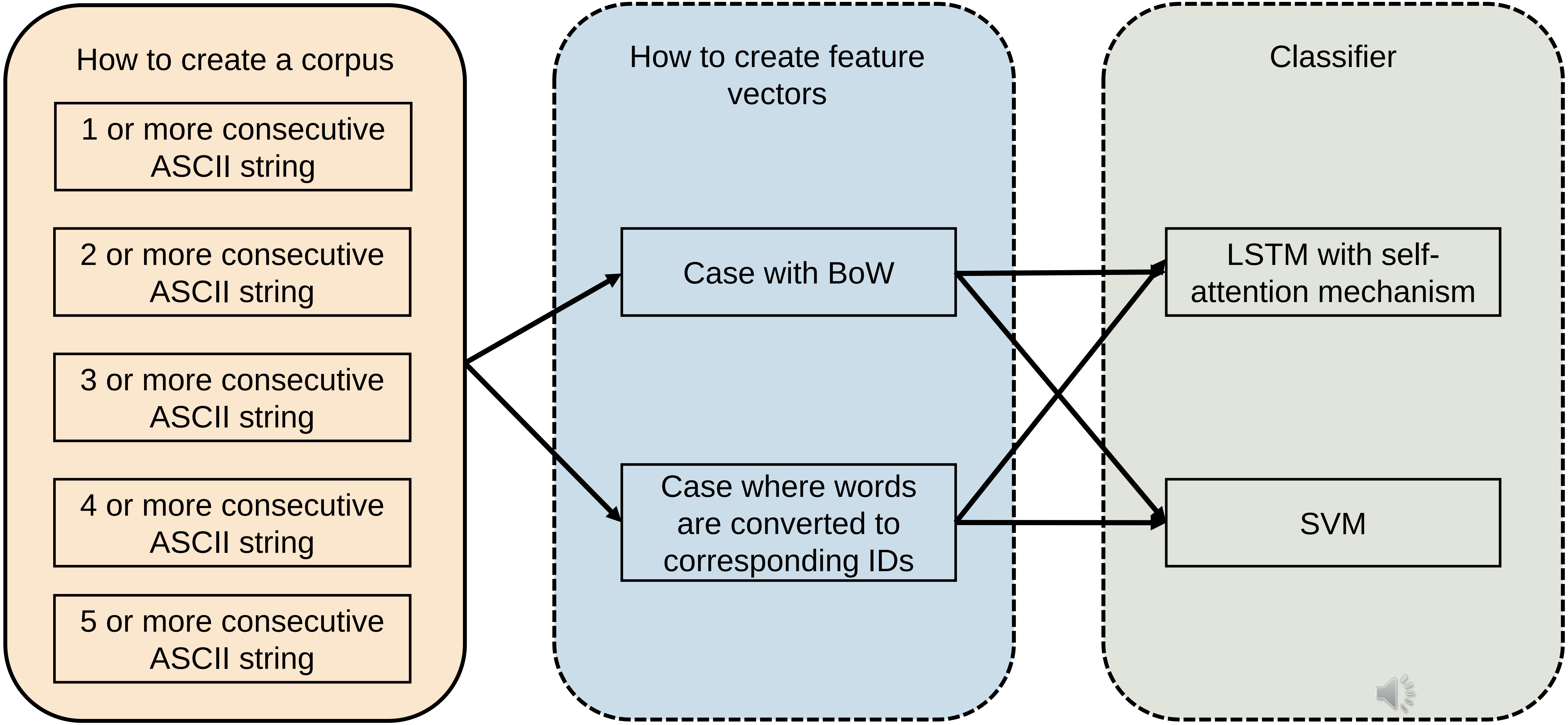
⇩ Extract words with 2 or more consecutive ASCII strings

['This:1', 'program:4', 'cannot:2', 'DOS:3']

⇩ Top 3 words in order of frequency of occurrence

['program:4', 'DOS:3', 'cannot:2']

4. Experimental Method (3 / 3)

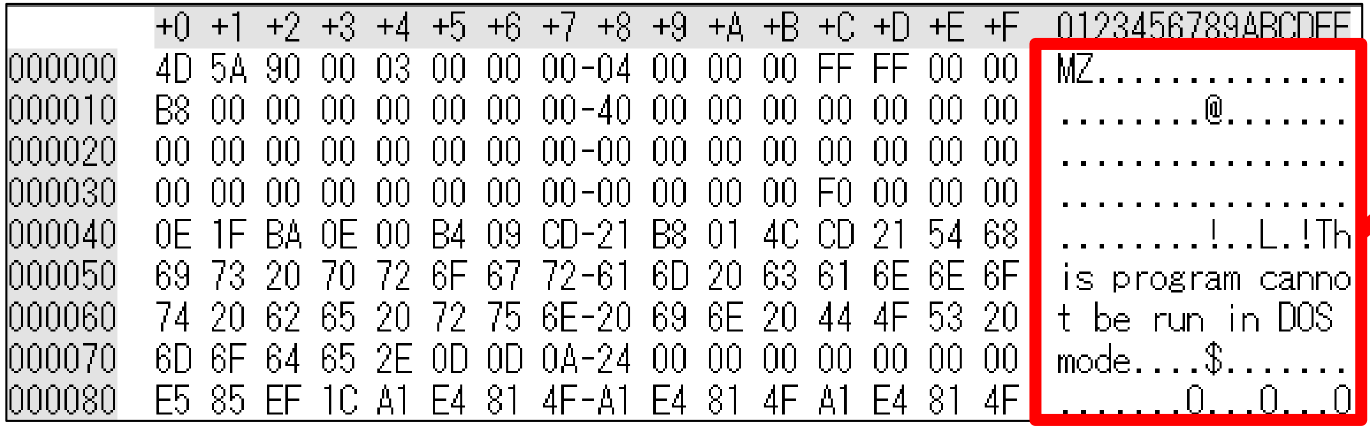


5. Experiment (1 / 8)

About datasets

- FFRI Datasets are datasets of surface analysis
- Distributed in json format
- Cleanware of FFRI Datasets ware collected by AV-TEST
- Malware of FFRI Datasets ware collected by FFRI Security, Inc.

- Use the strings of FFRI Dataset 2020 to 2022



Contents displayed in a binary editor

FFRI Dataset	
Element	**Summary**
id	SHA-256 hash value of samples
file_size	File size
hashes	Various hash values
peid	Output of pypeid
lief	Output of lief
trid	Output of trid
strings	**Output of strings**
die	Output of die
manalyze_plugin_packer	Output of manalyze plugin packer
label	label
date	Date collected
version	Version of a dataset

5. Experiment (2 / 8)

About datasets

Dataset	Classification	Files	Unique words
FFRI Dataset 2020	Cleanware	75,000	967,075,087
	Malware	75,000	162,245,592
FFRI Dataset 2021	Cleanware	75,000	1,001,705,100
	Malware	75,000	15,504,0251
FFRI Dataset 2022	Cleanware	75,000	712,981,765
	Malware	75,000	298,828,720

5. Experiment (3 / 8)

Results of preliminary experiments

- Preliminary experiments were conducted to optimize the parameters of vocab size and sequence length for machine learning.
- The FFRI Dataset 2020 was used for this experiment.

How to create feature vectors	Vocab size	Sequence length
Case with BoW	500	
Case where words are converted to corresponding IDs	100,000	120

5. Experiment (4 / 8)

Combining training and test data in validation experiments

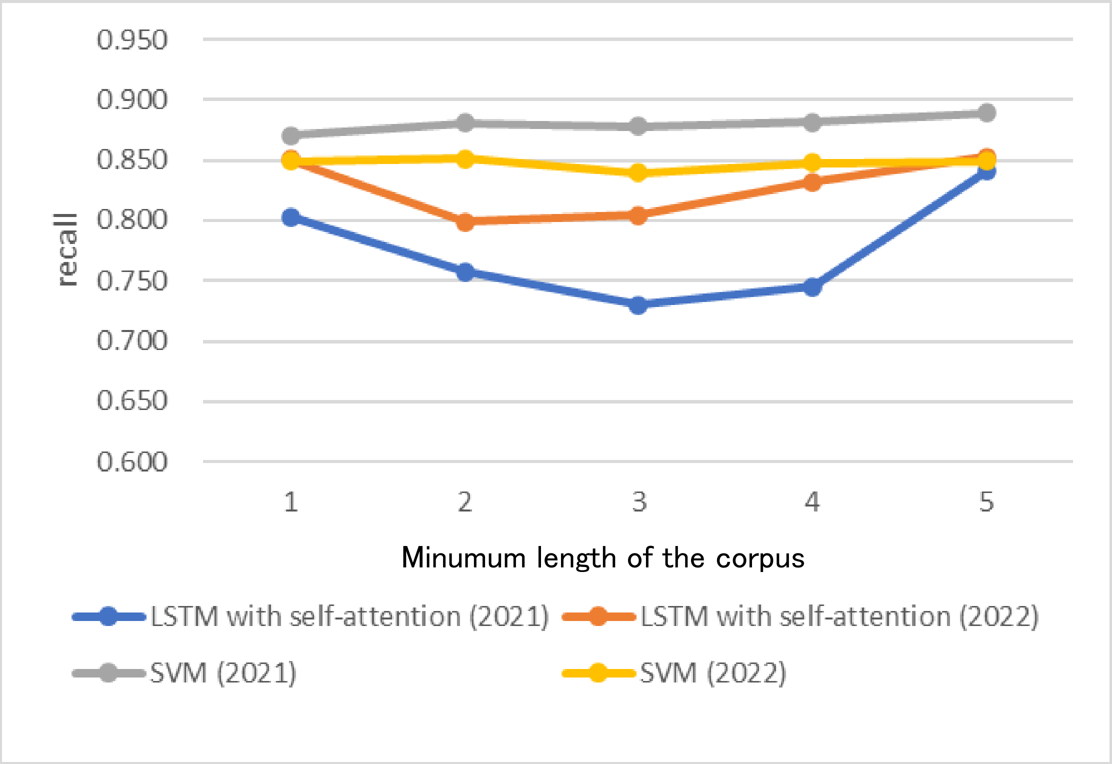
Based on FFRI Dataset 2020, detect FFRI Dataset 2021 and FFRI Dataset 2022

(Time series analysis)

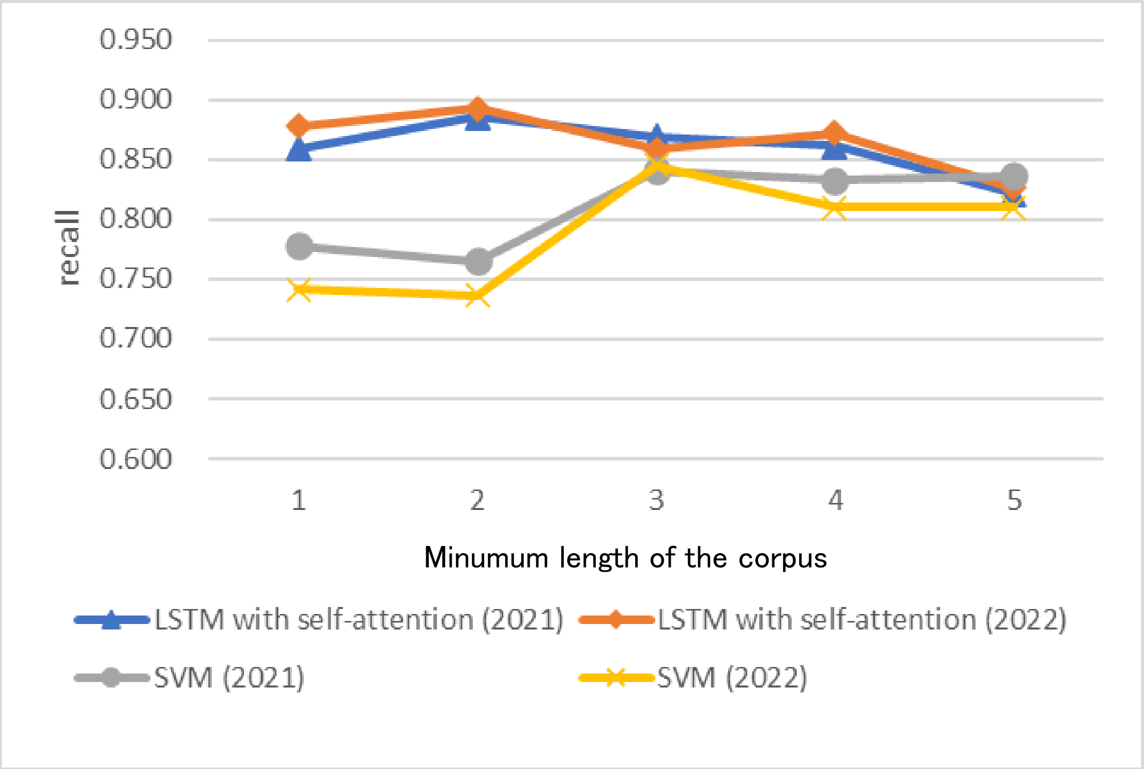
Training data	Test data
FFRI Dataset 2020	FFRI Dataset 2021
	FFRI Dataset 2022

5. Experiment (5 / 8)

Recall results



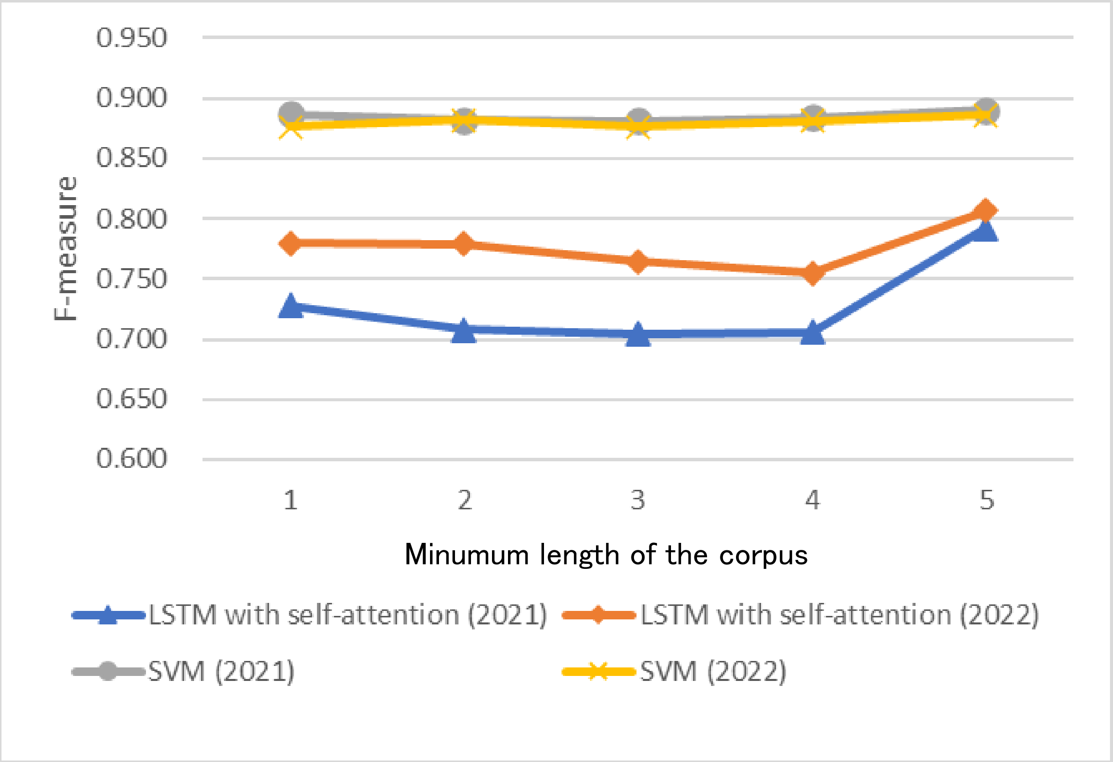
Case with BoW



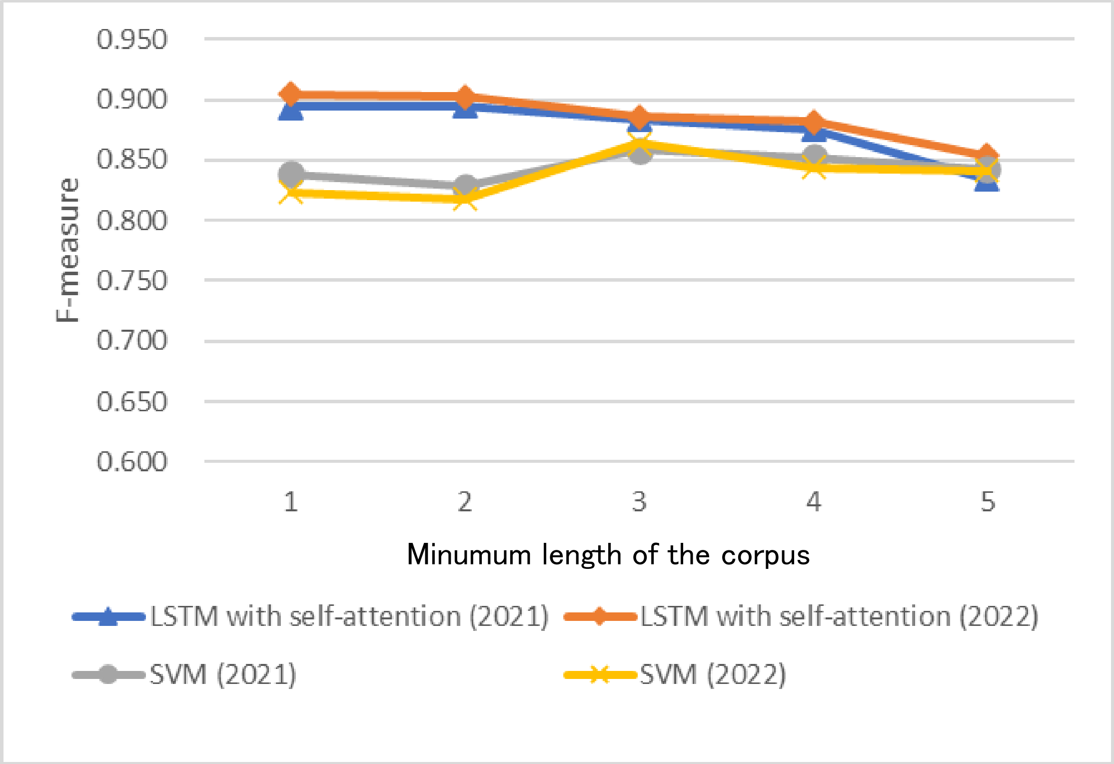
Case where words are converted to corresponding IDs

5. Experiment (6 / 8)

F−measure results



Case with BoW



Case where words are converted to corresponding IDs

5. Experiment (7 / 8)

Aggregated results for words of high importance
(Corpus created with strings of 2 or more consecutive ASCII strings that had the highest Recall value was used)

- Words appearing in all of TN, FP, TP and FN are colored blue
- Words common to three of the four are colored green.

FFRI Dataset 2021				
Rank	TN	FP	TP	FN
1	run	run	in	in
2	program	program	run	run
3	be	be	data	up
4	in	in	rd	rs
5	dos	dos	text	data
6	text	must	rs	text
7	rd	under	rich	rd
8	reloc	win32	id	id
9	data	text	reloc	rich
10	rs	rd	this	dll

FFRI Dataset 2021				
Rank	TN	FP	TP	FN
11	must	data	tls	this
12	bs	up	win32	win32
13	under	rs	under	under
14	win32	rich	boolean	tls
15	id	dll	FALSE	sv
16	tls	as	it	bs
17	xd	wi	TRUE	ad
18	strings	54	integer	as
19	rich	yr	sv	reloc
20	it	reloc	up	4o

5. Experiment (8 / 8)

Aggregated results for words of high importance
(Corpus created with strings of 2 or more consecutive ASCII strings that had the highest Recall value was used)

- Words appearing in all of TN, FP, TP and FN are colored blue
- Words common to three of the four are colored green.

FFRI Dataset 2022				
Rank	TN	FP	TP	FN
1	in	in	cannot	cannot
2	dos	dos	run	run
3	cannot	cannot	rich	rich
4	rd	js	rd	up
5	data	text	data	main
6	bs	data	rs	emu
7	reloc	exe	text	g7
8	rs	rd	be	bs
9	text	z9	under	fv
10	tls	rs	up	dl

FFRI Dataset 2022				
Rank	TN	FP	TP	FN
11	id	dll	sn	petite
12	be	go	id	ein
13	run	ai	gg	sv
14	win32	kt	reloc	rs
15	core	mp	ad	dll
16	pd	zo	as	5t
17	303	cm	ed	text
18	hh	ds	tls	hd
19	uu	qb	bs	uw
20	sv	ni	le	code

6. Discussion (1 / 2)

Need to consider consecutive ASCII string in corpus creation

- In both the BoW case and the case where IDs are assigned corresponding to words, the recall and f-measure values are improved by considering consecutive ASCII strings when creating a corpus
- However, a long ASCII string is not always better

⇩

There are certain benefits to considering consecutive ASCII strings when creating a corpus

6. Discussion (2 / 2)

Effect of self-attention mechanism on ASCII strings

- Of the top 20 words in each of TN, FP, TP, and FN, about 60% of the words in FFRI Dataset2021 and about 30% in FFRI Dataset2022 had 3 or more words in common with each of TN, FP, TP, and FN
- Focusing on the top words in TN, FP, TP, and FN, about 50% of the words in the test data are common to both FFRI Dataset2021 and FFRI Dataset2022

↓

Potential to improve detection rate by creating a corpus of only **words of high importance**

e.g.
Create a corpus of words common only to benign files and words common only to malignant files

7. Conclusion (1 / 1)

Conclusion

1. A new model with a self-attention mechanism was used to detect malware using ASCII strings. The maximum F-measure was 0.904
2. We confirmed that removing non-consecutive ASCII strings from the corpus has a certain effect.
3. The influence of the self-attention mechanism on readable strings was clarified, and it was confirmed that there are words of high importance that contribute to detection.

Future Plans

1. How does accuracy change when combined with features other than readable strings
2. Check the effect on accuracy of using other datasets
3. Check the effect on accuracy of creating a corpus with words of high importance