# Evasion attempt for the malicious PowerShell detector considering feature weights

National Defense Academy
Kou Sugiura
Mamoru Mimura

# Contents

# 1-1 Background

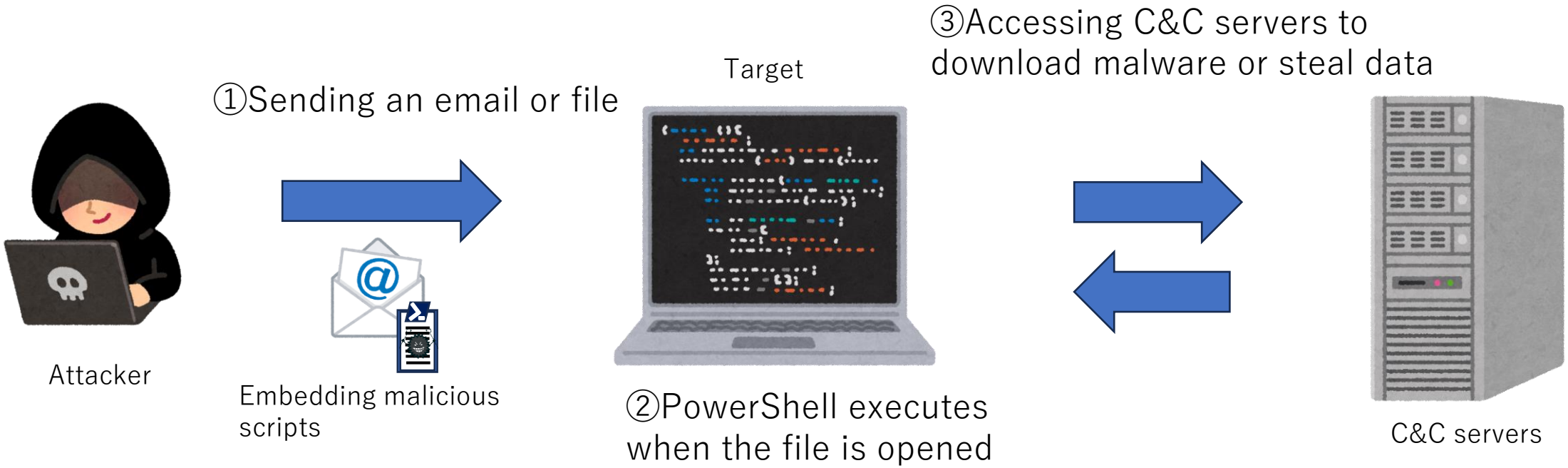
- **The number of attacks on widely used software is increasing**
  - PowerShell, which is highly convenient, is frequently exploited by attackers.

- **A detection method using natural language processing and machine learning has been proposed to detect malicious PowerShell in previous research**
  - It has been reported that evasion attacks against the machine learning models for detecting malicious PowerShell are possible
  - However, evaluation of evasion attacks against malicious PowerShell detectors using neural networks has not been conducted

**We evaluated the possibility of evasion attacks against malicious PowerShell detectors using neural networks.**

# 1-2 Background

- **An example of an attack using PowerShell**



# 1-3 Background

- **Objective**

  - Evaluation of evasion attacks against malicious PowerShell detectors using neural networks

- **Contribution**

  - Demonstrating the possibility of evasion attacks on malicious PowerShell detectors using neural networks

  - Showing that effective evasion attacks are possible by inserting words extracted using attention weights

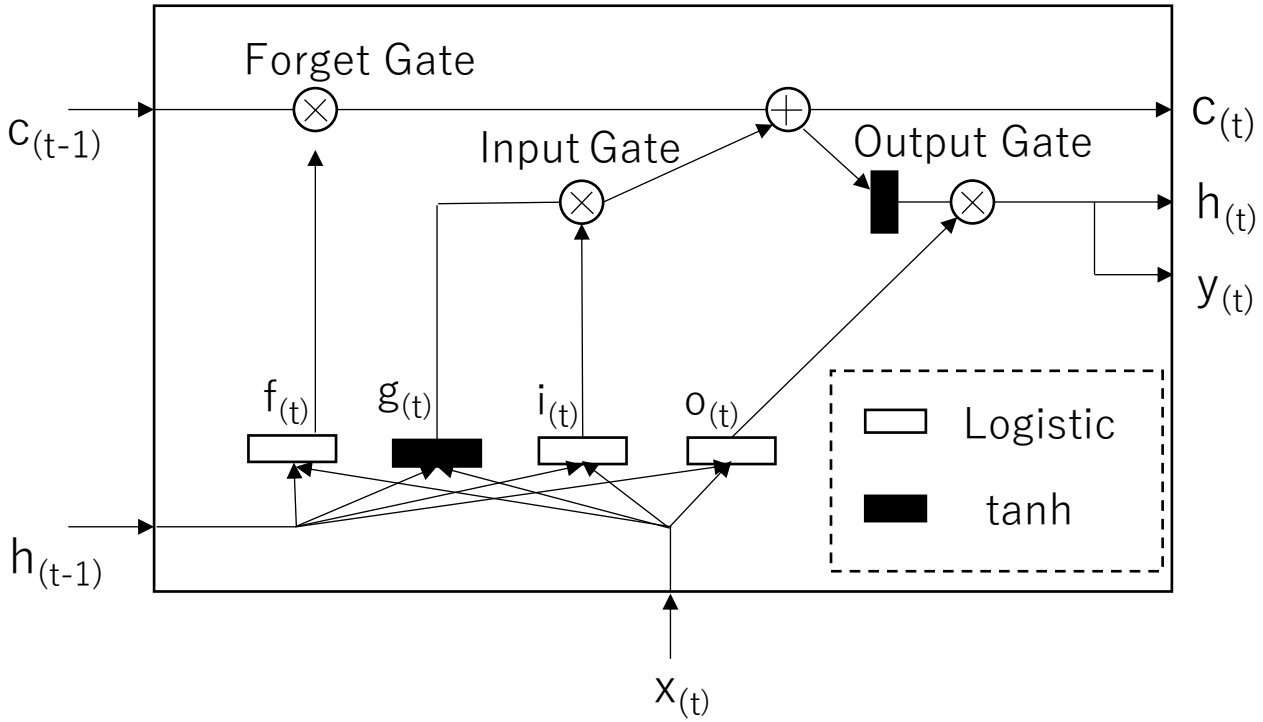
# 2 Related Work

| No | Paper Title | Purpose | Evasion Method | Detection Model |
|---|---|---|---|---|
| 1 | **Detecting Malicious PowerShell Commands using Deep Neural Networks**<br>Danny Hendler, Shay Kels, Amir Rubin<br>ASIACCS '18: Proceedings of the 2018 on Asia Conference on Computer and Communications Security | detection | × | NLP（3-gram、BoW、LSTM）and Deep Learning（9-CNN、4-CNN） |
| 2 | **Detection of Malicious PowerShell Using Word-Level Language Models**<br>Yui Tajiri, Mamoru Mimura<br>15th International Workshop on Security: Advances in Information and Computer Security | detection | × | NLP（BoW、Doc2Vec、LSI）and Machine Learning（SVM、RandomForest、XGBoost） |
| 3 | **Evaluating the Possibility of Evasion Attacks to Machine Learning-Based Models for Malicious PowerShell Detection**<br>Mezawa,Y.Mimura,M<br>The 17th International Conference on Information Security Practice and Experience | evasion | Inserting words that appear only benign PowerShell | NLP（BoW、Doc2Vec、LSI）and Machine Learning（SVM、RandomForest、XGBoost |
| **4** | **This study** | evasion | Inserting words extracted attention weights | Attention+LSTM,LSTM,RNN,DNN,CNN |

# 3-1 Related Techniques

- **Long Short Term Memory（LSTM）**

  LSTM is a type of recurrent neural network (RNN) that can solve the long-term memory problem caused by vanishing and exploding gradients, which are issues faced by RNNs




$$i_{(t)} = \sigma(W_{xi}x_{(t)} + W_{hi}h_{(t-1)} + b_i)$$

$$f_{(t)} = \sigma(W_{xf}x_{(t)} + W_{hf}h_{(t-1)} + b_f)$$

$$o_{(t)} = \sigma(W_{xo}x_{(t)} + W_{ho}h_{(t-1)} + b_o)$$

$$g_{(t)} = \tanh(W_{xg}x_{(t)} + W_{hg}h_{(t-1)} + b_g)$$

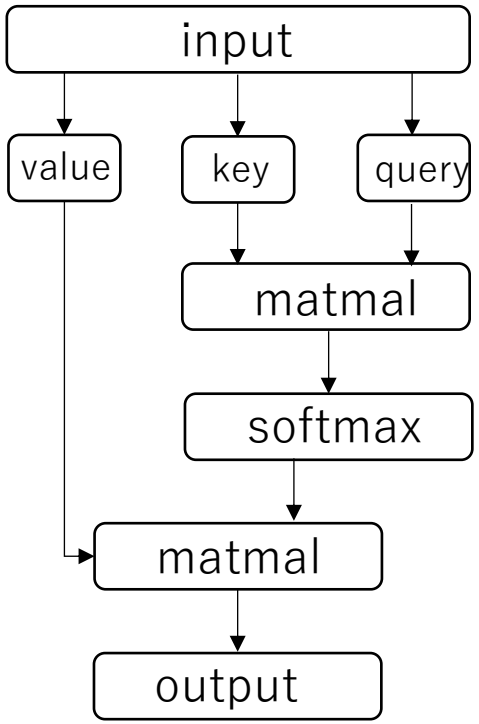
$$c_{(t)} = f_{(t)}c_{(t-1)} + i_{(t)}g_{(t)}$$

$$y_{(t)} = h_{(t)} = o_{(t)}\tanh(c_{(t)})$$

# 3-2 Related Techniques

- **Self-Attention**

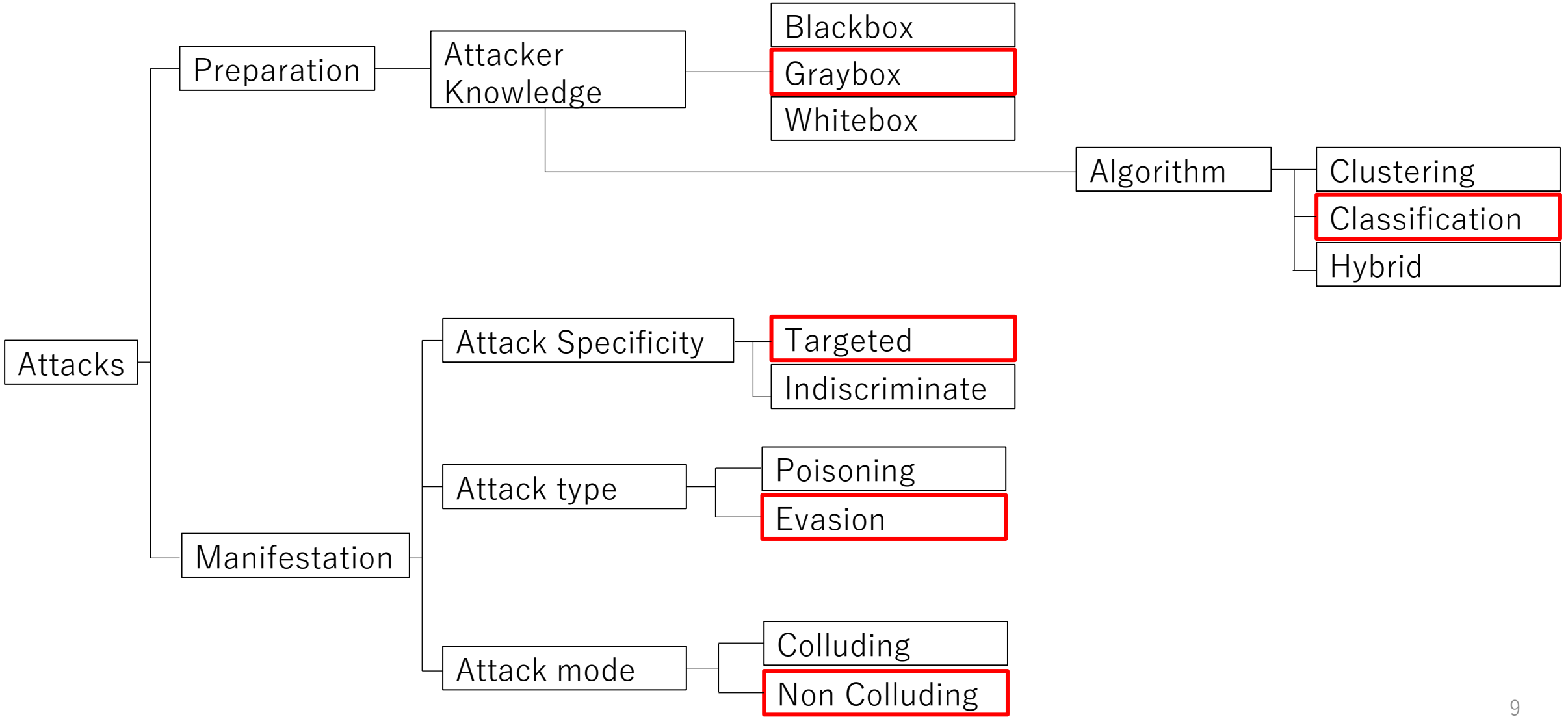
By visualizing the weights of the input data tokens, called attention weights, it is possible to reveal which parts of the input data were focused on for making predictions.




$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

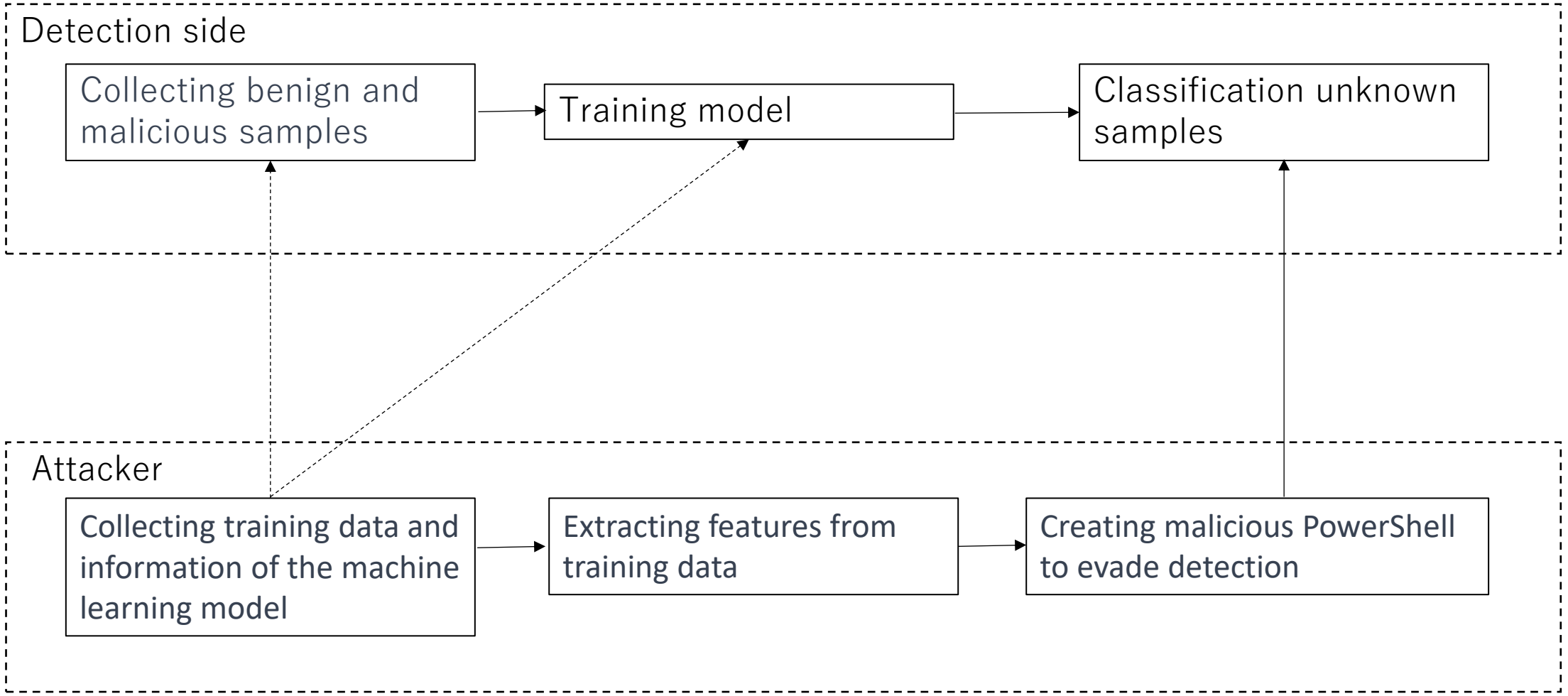
# 3-3 Related Techniques

- **Attack on the machine learning model**

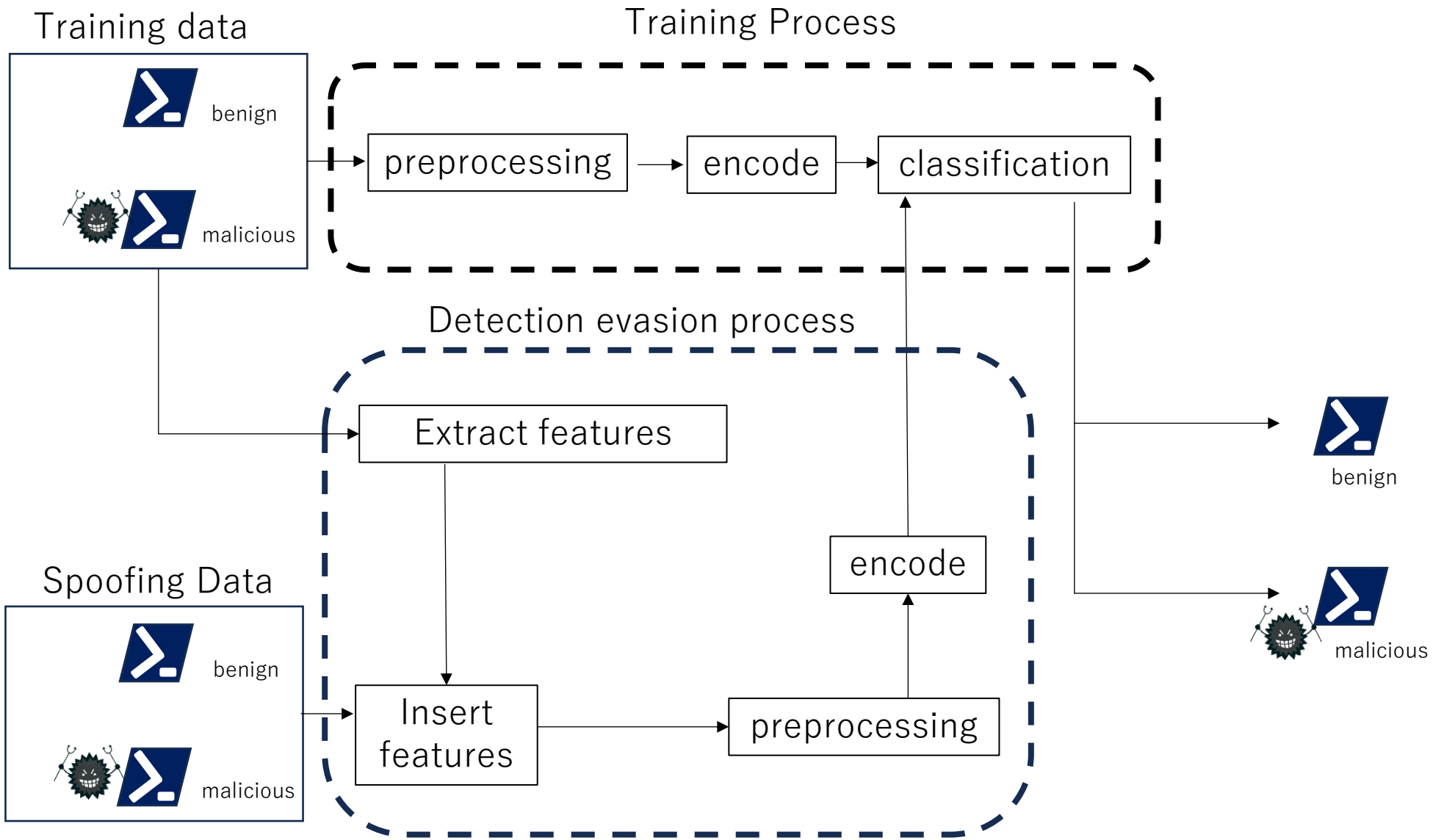


# 4-1 Evaluation Method

- **Condition**



# 4-2 Evaluation Method



# 4-3 Evaluation Method

- **Example of insert process**

C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥
powershell.EXE -nop -ep bypass -e
SQBFAFgAIAAoAE4AZQB3AC0ATwBiAGoAZQBjAH
QAIABOAGUAdAAuAFcAZQBiAEMAbABpAGUAbgB0
ACkALgBkAG8AdwBuAGwAbwBhAGQAcwB0AHIAa
QBuAGcAKAAnAGgAdAB0AHAAOgAvAC8AdgAuAGI
AZABkAHAALgBuAGUAdAAvAHcAbQA/AGgAZABw
ACcAKQA=
"pipeline".ToUpper()
"name".Length

High-weight benign features from the Attention mechanism

clcommentout
pipeline
name
server
clurl
win0
basestring
foreach
function

# 5-1 Evaluation Experiment

- **Dataset**

Data sources： HybridAnalysis,　AnyRun,　GitHub
Collection period： January 2019 to March 2020
Collection method： Manual and Using public APIs

Split by time series as shown in the table below ➡ Test data is unknown to the machine learning model

| | AnyRun,　HybridAnalysis | | | GitHub |
|---|---|---|---|---|
| Dataset type | period | Malicious | benign | benign |
| Train data | Until June 2019 | 309 | 232 | 4901 |
| Spoofing data | From July 2019 | 171 | 92 | |

・Undersampling was applied to the training data to equalize the number of malicious and benign samples.
・Undersampling was not applied to the spoofind data.

# 5-2 Evaluation Experiment

- **Setting**

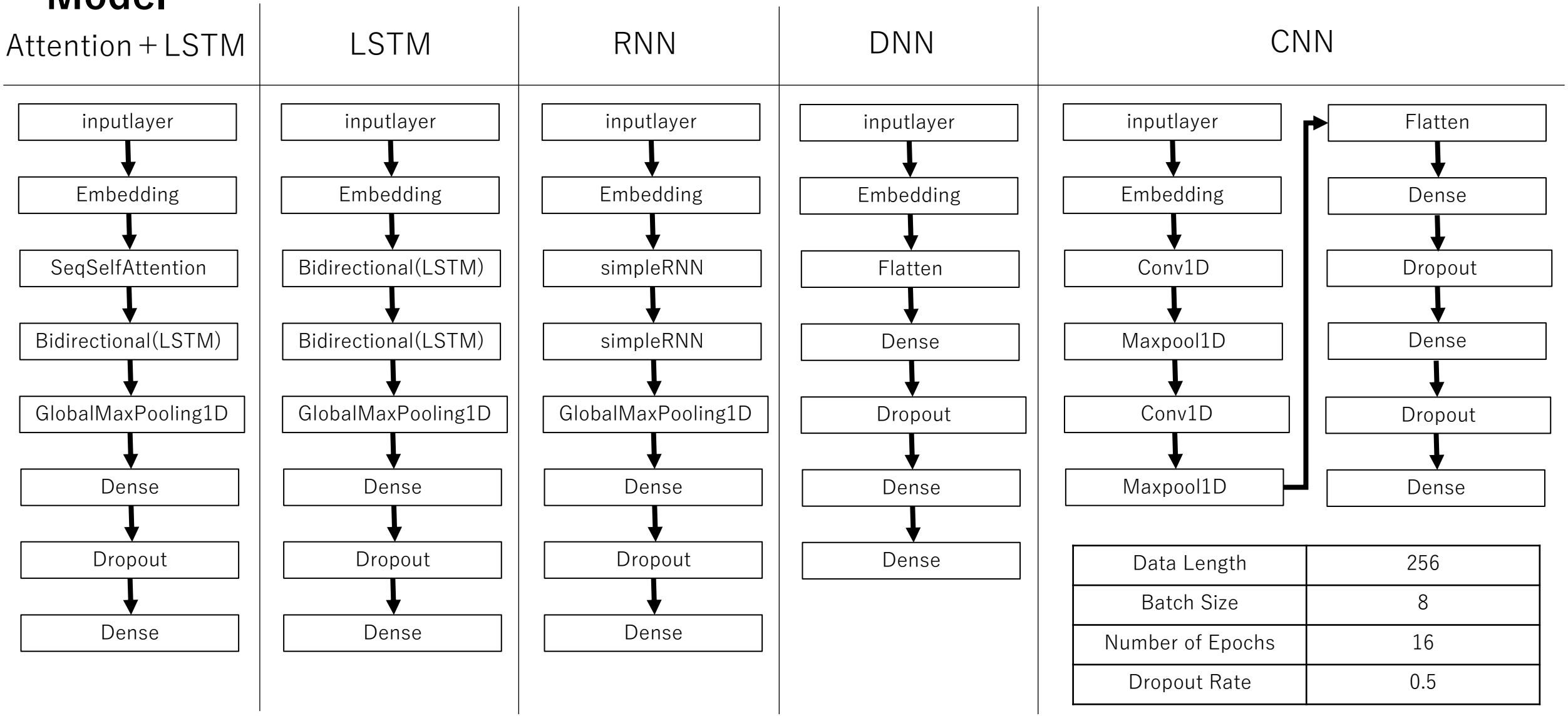
| | |
|---|---|
| CPU | Core i7-9700K 3.60GHz |
| Memory | 64GB |
| OS | Windows10 Home |
| Language | Python3.8.9 |

- **Libraries**

| | |
|---|---|
| Scikit-learn | 1.0.2 |
| Keras | 2.11.0 |
| Keras-self-attention | 0.51.0 |
| Tensorflow-estimater | 2.11.0 |

# 5-3 Evaluation Experiment

- **Model**




| | |
|---|---|
| Data Length | 256 |
| Batch Size | 8 |
| Number of Epochs | 16 |
| Dropout Rate | 0.5 |

# 5-4 Evaluation Experiment

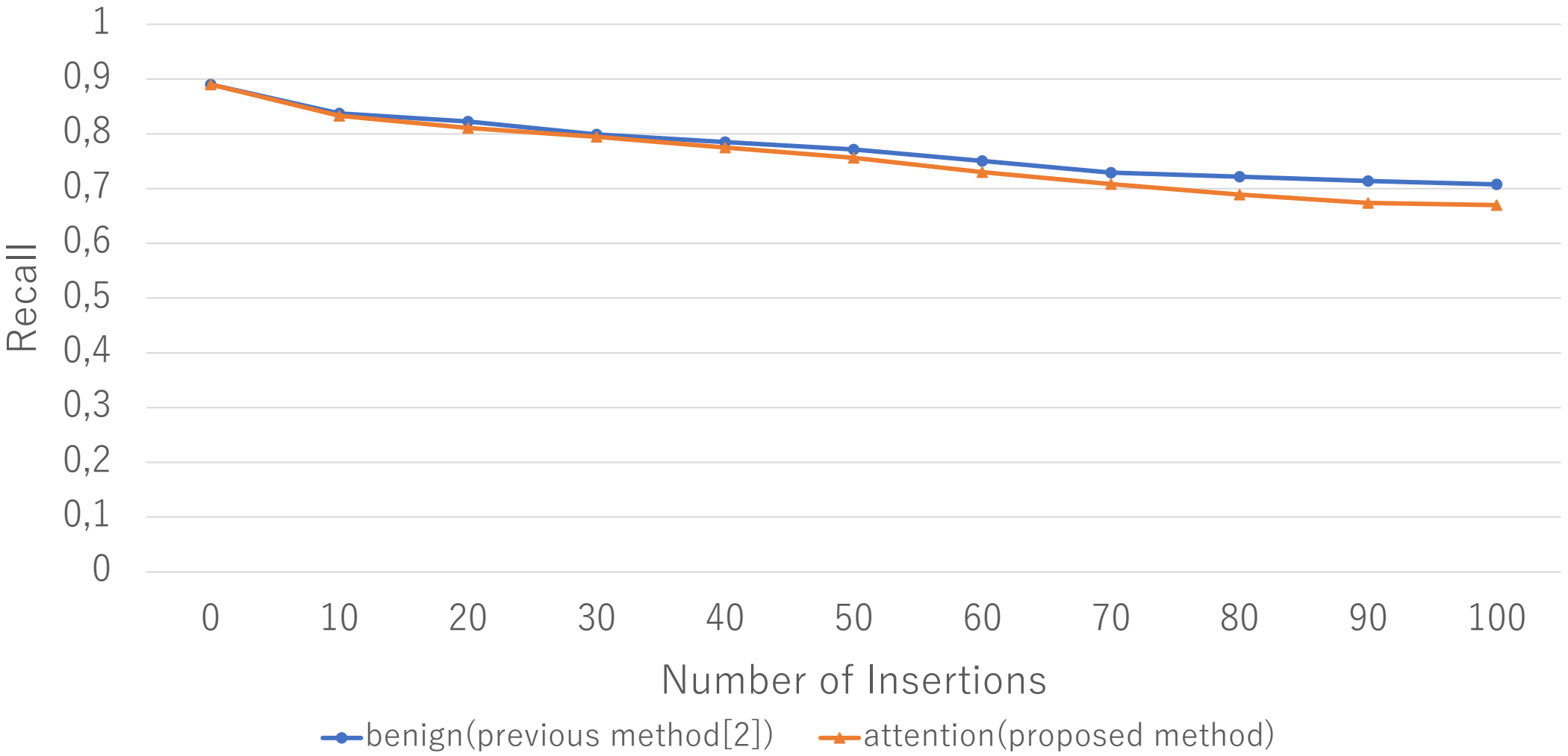
・Overview of the experiment

| Number | Experimental Model | Purpose |
|---|---|---|
| 1 | Attention+LSTM | ・Evaluation of evasion attacks using the same model as in previous research [1]<br>・Comparison of recall rate when words extracted using attention weights and words frequently appearing only in benign samples are inserted |
| 2 | LSTM<br>RNN<br>DNN<br>CNN | Comparison of the recall rate when words extracted using attention weights and words that frequently appear only in benign samples are inserted into each model |

・All experiments are conducted 10 times, and the average value is taken as the result.

[1]Mezawa, Y. and Mimura, M.: An Attention Mechanism for Visualizing Word Weights in Source Code of PowerShell Samples: Experimental Results and Analysis, International Conference on Broadband and Wireless Computing, Communication and Applications, Springer, pp. 114–124 (2022).

# 6 Result




**The Recall Rate of the Model using Attention + LSTM**

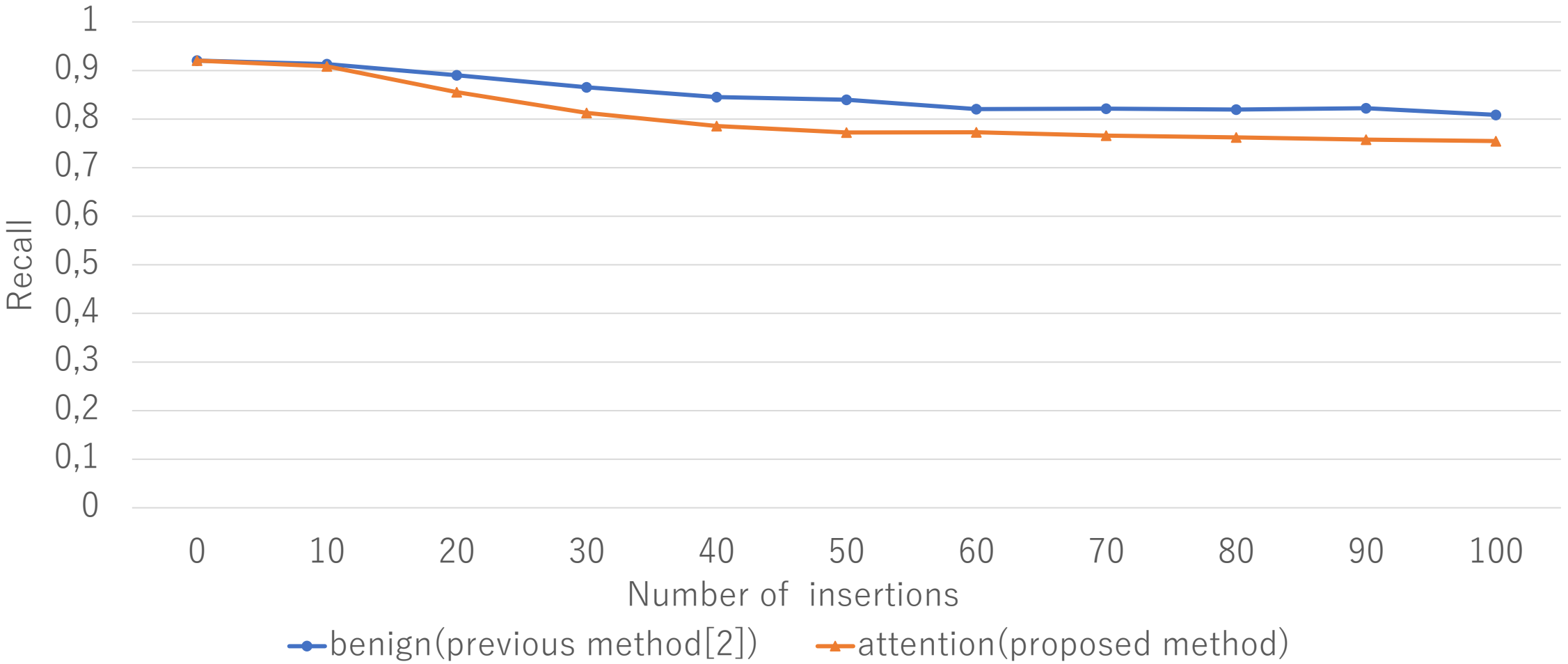
[2]Evaluating the Possibility of Evasion Attacks to Machine Learning-Based Models for Malicious PowerShell Detection
Mezawa,Y.Mimura,M The 17th International Conference on Information Security Practice and Experience

# 6 Result


**The Recall Rate of the Model using LSTM**

[2]Evaluating the Possibility of Evasion Attacks to Machine Learning-Based Models for Malicious PowerShell Detection
Mezawa,Y.Mimura,M The 17th International Conference on Information Security Practice and Experience

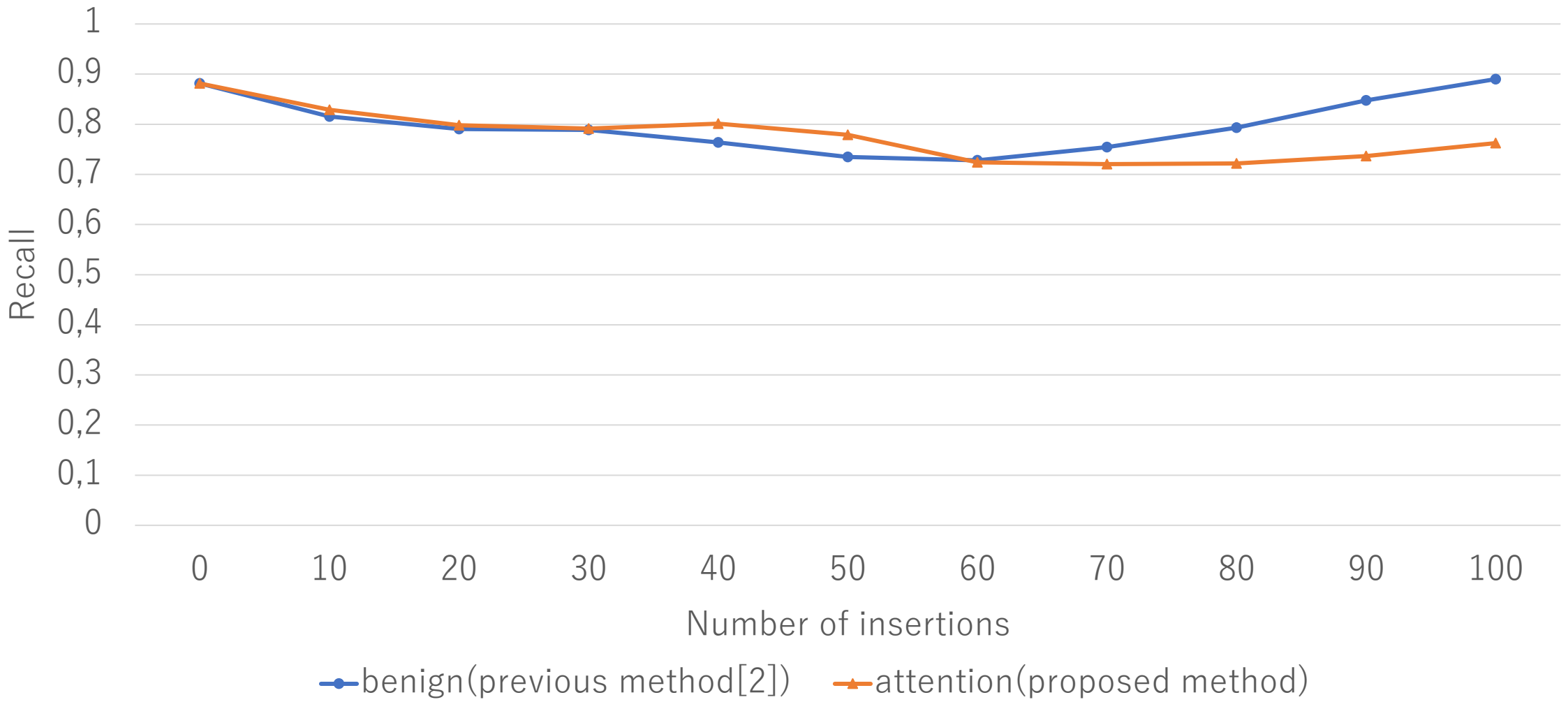
# 6 Result




**The Recall Rate of the Model using RNN**

[2]Evaluating the Possibility of Evasion Attacks to Machine Learning-Based Models for Malicious PowerShell Detection
Mezawa,Y.Mimura,M The 17th International Conference on Information Security Practice and Experience

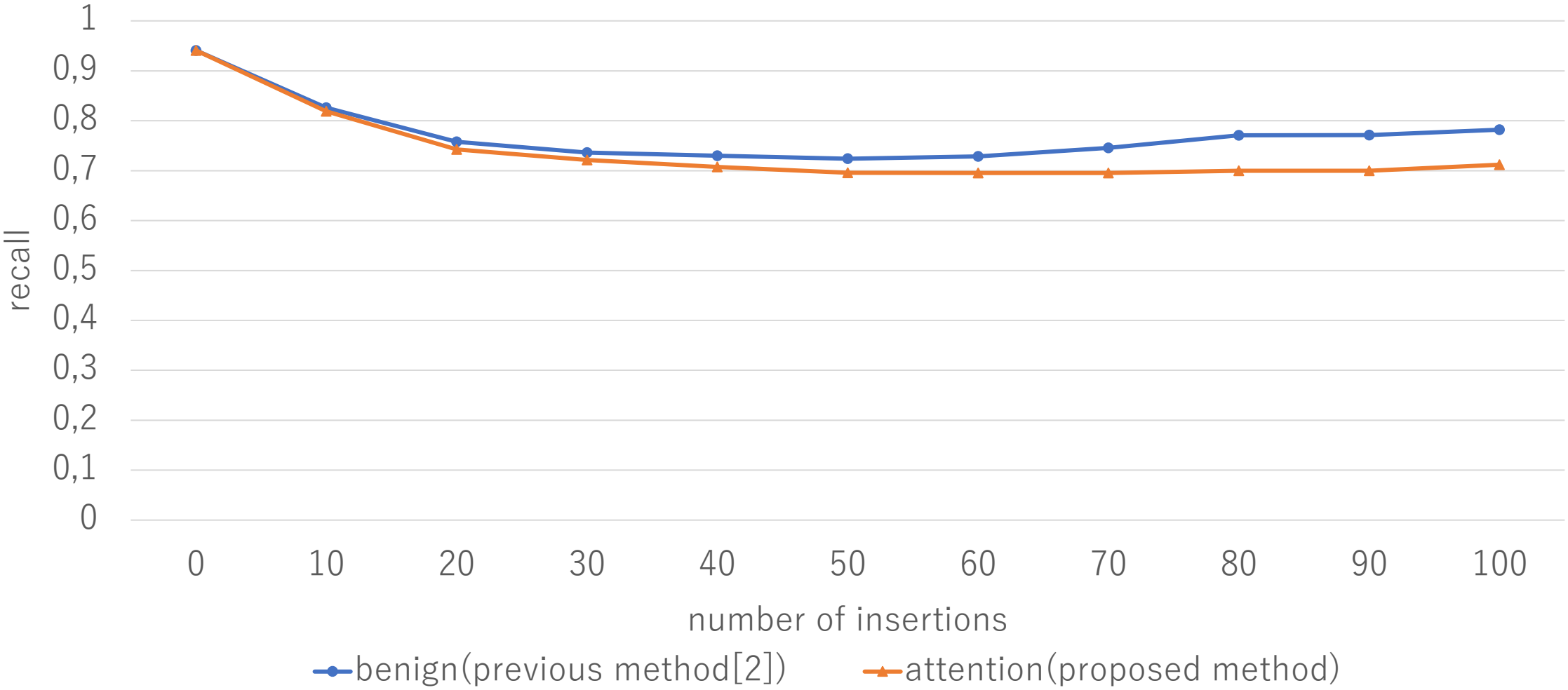
# 6 Result




**The Recall Rate of the Model using DNN**

[2]Evaluating the Possibility of Evasion Attacks to Machine Learning-Based Models for Malicious PowerShell Detection
Mezawa,Y.Mimura,M The 17th International Conference on Information Security Practice and Experience

# 6 Result




**The Recall Rate of the Model using CNN**

[2]Evaluating the Possibility of Evasion Attacks to Machine Learning-Based Models for Malicious PowerShell Detection
Mezawa,Y.Mimura,M The 17th International Conference on Information Security Practice and Experience

# 7-1 Discussion

- **Potential evasion attack on the model combining LSTM and Attention**

➡ The recall rate decreased by 0.23 when inserting words extracted using attention weights.

➡ The recall rate decreased more by inserting words extracted using the attention weights.

▼

**More efficient evasion attacks are possible by using words with high feature weights**

# 7-2 Discussion

- **Potential evasion attacks on other neural network models**

➡ The recall rate decreased for all models

➡ The recall rate decreased more when inserting words extracted using the Attention weight

▼

Models using neural networks are thought to classify based on some common words.

▼

**More efficient evasion attacks are possible by using words with high feature weights.**

# 7-3 Discussion

- **Maintaining Malware behavior**

To maintain the behavior of the original malware, words indicating benign characteristics and functions that do not affect the behavior were inserted at the end of the source code

If there is no change in the insertion position or the inserted words, it may become a new feature of malicious PowerShell

In DNN and CNN, the recall rate increases to some extent as the number of insertions increases.

**The insertion process is recognized as a new feature of malicious PowerShell.**

# 7-4 Discussion

- **Feasibility of the attack**

➡ Under the conditions of this experiment, attackers need to have prior access to the training data and the models used

➡ Commercial detectors capture various features for detection.

▼

**The feasibility of conducting attacks under realistic conditions is considered low.**

# 8 Conclusion

- **Contribution of this study**

➡ It was confirmed that evasion attacks are possible against malicious PowerShell detectors using a neural networks.

➡ It was confirmed that more effective evasion attacks can be carried out by inserting words with high feature weights extracted using the Attention mechanism.

- **Future challenge**

➡ Increasing the number of samples and verifying the behavior of modified samples

Thank you.