

REPORT PROJECT

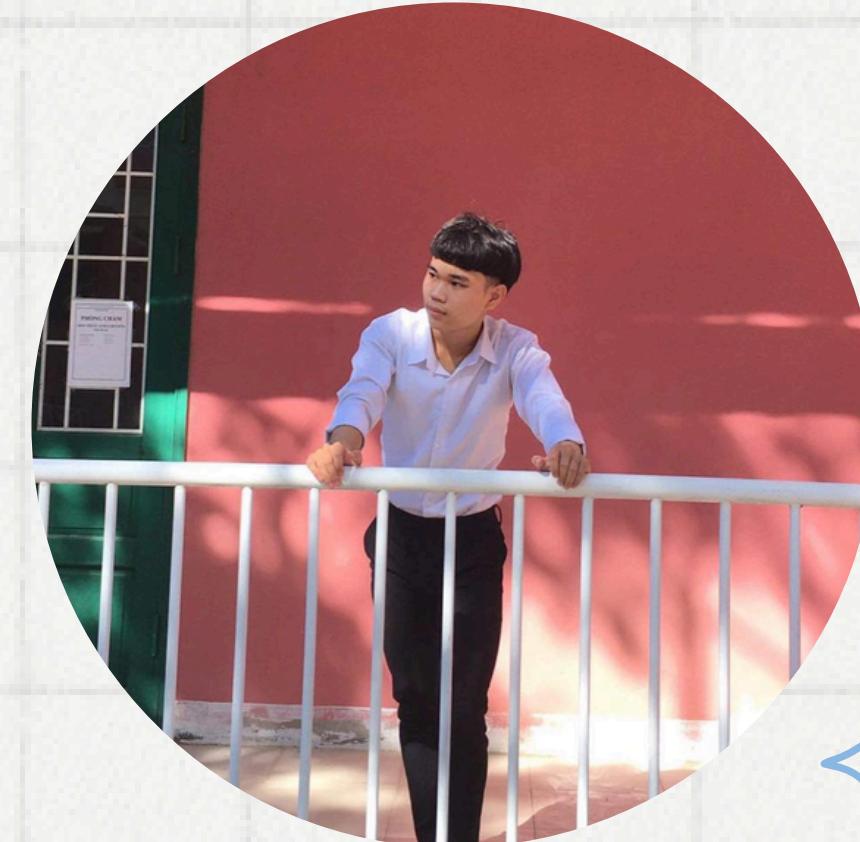
TOPIC

# Detecting Malicious Web Requests Using an Enhanced TextCNN

**NT213.O21.ANTN**  
**Nhóm 11**



**Nguyễn Vũ Anh Duy**



**Nguyễn Văn Khang Kim**

# CONTENT

01. GIỚI THIỆU ĐỀ TÀI

02. QUY TRÌNH THỰC HIỆN

03. DEMO

04. KẾT LUẬN

01

# GIỚI THIỆU ĐỀ TÀI

# Paper

## **Detecting Malicious Web Requests Using an Enhanced TextCNN**

Lian Yua , Lihao Chena , Jingtao Donga , Mengyuan Lia , Lijun Liub , Bei Zhaob , Chen Zhangb a School of Software and Microelectronics, Peking University b Institute of Research and Development, Mobile China Beijing, China lianyu@ss.pku.edu.cn

[link](https://ieeexplore.ieee.org/document/9202428) <https://ieeexplore.ieee.org/document/9202428>

# Ngữ cảnh

- Các cuộc tấn công ứng dụng web đang gia tăng nhanh chóng theo từng năm, đặc biệt các cuộc tấn công này gia tăng 400% từ năm 2015 đến năm 2016.
- Các cuộc tấn công này để lại hậu quả nghiêm trọng, gây tổn thất lớn về tiền bạc và danh tiếng.
- Uniform Resource Locators (URL) còn được gọi là bộ định vị tài nguyên thống nhất là nơi cung cấp các yêu cầu web tới các trang web và đóng vai trò quan trọng giữa máy khách và máy chủ.
- Vì những lý do đó, đề tài này sẽ thực hiện việc ứng dụng một phương pháp dựa trên học sâu cụ thể là TextCNN để thực hiện ngăn chặn các yêu cầu độc hại đến ứng dụng web.

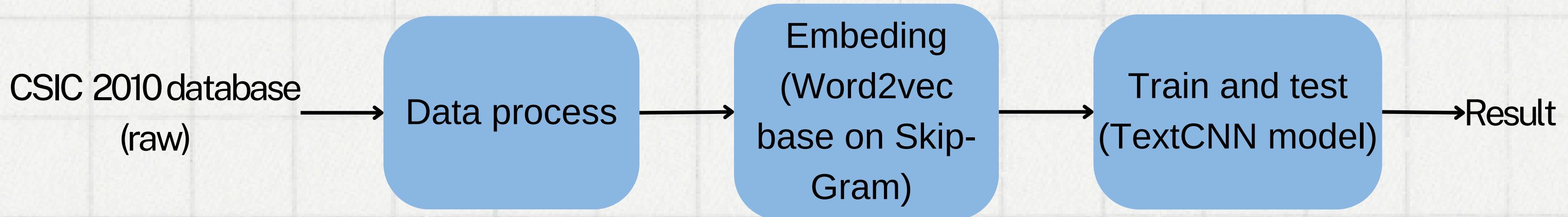
# Mục tiêu

Thực hiện phân loại các yêu cầu web dựa trên mô hình học sâu TextCNN

02

# QUY TRÌNH THỰC HIỆN

# Các bước thực hiện



# Data process

- Tập dữ liệu thô HTTP CSIC 2010 với 36000 yêu cầu hợp lệ và 25065 yêu cầu không hợp lệ.

```
1 Start - Id: 31586
2 class: Valid
3 GET http://localhost:8080/tienda1/publico/productos.jsp HTTP/1.1
4 User-Agent: Mozilla/5.0 (compatible; Konqueror/3.5; Linux) KHTML/3.5.8 (like Gecko)
5 Pragma: no-cache
6 Cache-control: no-cache
7 Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=
8 Accept-Encoding: x-gzip, x-deflate, gzip, deflate
9 Accept-Charset: utf-8, utf-8;q=0.5, *;q=0.5
10 Accept-Language: en
11 Host: localhost:8080
12 Cookie: JSESSIONID=17AC631A905D4B8ABCB255F2B9122C0
13 Connection: close
```

Ví dụ về yêu cầu hợp lệ

# Data process

```
1 Start - Id: 11044
2 class: Attack
3 GET http://localhost:8080/tienda1/publico/caracteristicas.jsp?idA=2 HTTP/1.1
4 User-Agent: Mozilla/5.0 (compatible; Konqueror/3.5; Linux) KHTML/3.5.8 (like Gecko)
5 Pragma: no-cache
6 Cache-control: no-cache
7 Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=
8 Accept-Encoding: x-gzip, x-deflate, gzip, deflate
9 Accept-Charset: utf-8, utf-8;q=0.5, *;q=0.5
10 Accept-Language: en
11 Host: localhost:8080
12 Cookie: JSESSIONID=372DDFCE584244AED285252827076DA6
13 Connection: close
```

Ví dụ về yêu cầu không hợp lệ

# Data process

- Trích xuất url từ yêu cầu và loại bỏ các url trùng lặp

```
9620 http://localhost:8080/tienda1/publico/anadir.jsp?id=2&nombre=Queso+Manch
9621 http://localhost:8080/tienda1/publico/registro.jspmodo=registro&login=darn&password=56
9622 http://localhost:8080/tienda1/miembros/editar.jspmodo=registro&login=fis
9623
```

Normal

Sau khi loại bỏ thì  
ta được 9622 url  
normal

Tương tự ta có  
15699 url attack

```
15697 localhost:8080/tienda1/publico/pagar.jspmodoA=insertar&precio=6858&B1=Confirmar
15698 localhost:8080/tienda1/miembros/editar.jsp?modoA=registro&login=darn&password=56
15699 localhost:8080/tienda1/publico/pagar.jsp?modo=insertar&precio=6714&B1=Pasar+por-
15700
```

Attack

# Data process

- Giải mã: Đôi lúc kẻ tấn công sẽ mã hóa nhiều lần để che giấu ý định tấn công của mình vì vậy ta sẽ thực hiện giải mã nhiều lần.

## URL:

localhost:8080/tienda1/publico/autenticar.jsp?modo=entrar&login=bob%2540%253CSCRipt%253Ealert%2528Paros%2529%253C%252FscrIPT%253E.parosproxy.org&pwd=c69p04e13&remember=off&B1=Entrar

## Decode Once:

localhost:8080/tienda1/publico/autenticar.jsp?modo=entrar&login=bob%40%3CSCRipt%3Ealert%28Paros%29%3C%2FscrIP  
T%3E.parosproxy.org&pwd=c69p04e13&remember=off&B1=En  
trar

## Decode Twice:

localhost:8080/tienda1/publico/autenticar.jsp?modo=entrar&login=bob@<SCRipt>alert(Paros)</scrIPT>.parosproxy.org&pwd=c69p04e13&remember=off&B1=Entrar

# Data process

- Chuyển các ký tự thành chữ thường
- Phân đoạn các ký tự của chuỗi theo các ký tự đặc biệt.

```
localhost : 8080 localhost : 8080 / tienda1 / publico / autenticar .  
jsp ? modo = entrar & login = bob @ < script > alert ( paros ) < /  
script > . parosproxy . org & pwd = c69p04e13 & remember = off &  
b1 = entrar
```

Một đoạn url được chuyển sang chữ thường và phân đoạn

# Data process

- Gán nhãn với 1 là tấn công, 0 là bình thường.

	A1	B	C	D	E	F	G	H	I
1	localhost : 8	1							
2	localhost : 8	1							
3	localhost : 8	1							
4	http : / / lc	0							
5	localhost : 8	1							
6	localhost : 8	1							
7	localhost : 8	1							
8	localhost : 8	1							
9	localhost : 8	1							

Tập dữ liệu sau khi được xử lý hoàn tất

# Embedding

- Trong bài toán này, chúng ta xem một câu (sentences) chính là một chuỗi các từ thu được khi xử lý một url đại diện cho một yêu cầu trang web.
- Một câu được định nghĩa là một tập hợp có thứ tự các từ như sau.

$$\textit{sentence} = \{\textit{word}_1, \dots, \textit{word}_i, \dots, \textit{word}_d\}$$

- Trong đó d là độ dài cố định của mỗi url, và  $\textit{word}_i$  chính là thứ tự của từ thứ i.

# Embedding

- Sau khi nhúng từ bằng Word2vec dựa trên Skip-Gram, ta thu được vector từ có dạng như sau.

```
word_vectors = {  
    'the': [0.1, 0.2, 0.3, 0.4, 0.5],  
    'cat': [0.2, 0.3, 0.4, 0.5, 0.6],  
    'sat': [0.3, 0.4, 0.5, 0.6, 0.7],  
    'on': [0.4, 0.5, 0.6, 0.7, 0.8],  
    'mat': [0.5, 0.6, 0.7, 0.8, 0.9]  
}
```

word vectors

# Embedding

- Giai đoạn cuối của nhúng từ là chúng ta sẽ tạo 1 ma trận nhúng từ vector từ và chỉ số từ.

```
[[0.  0.  0.  0.  0. ]
 [0.1 0.2 0.3 0.4 0.5]
 [0.2 0.3 0.4 0.5 0.6]
 [0.3 0.4 0.5 0.6 0.7]
 [0.4 0.5 0.6 0.7 0.8]
 [0.5 0.6 0.7 0.8 0.9]]
```

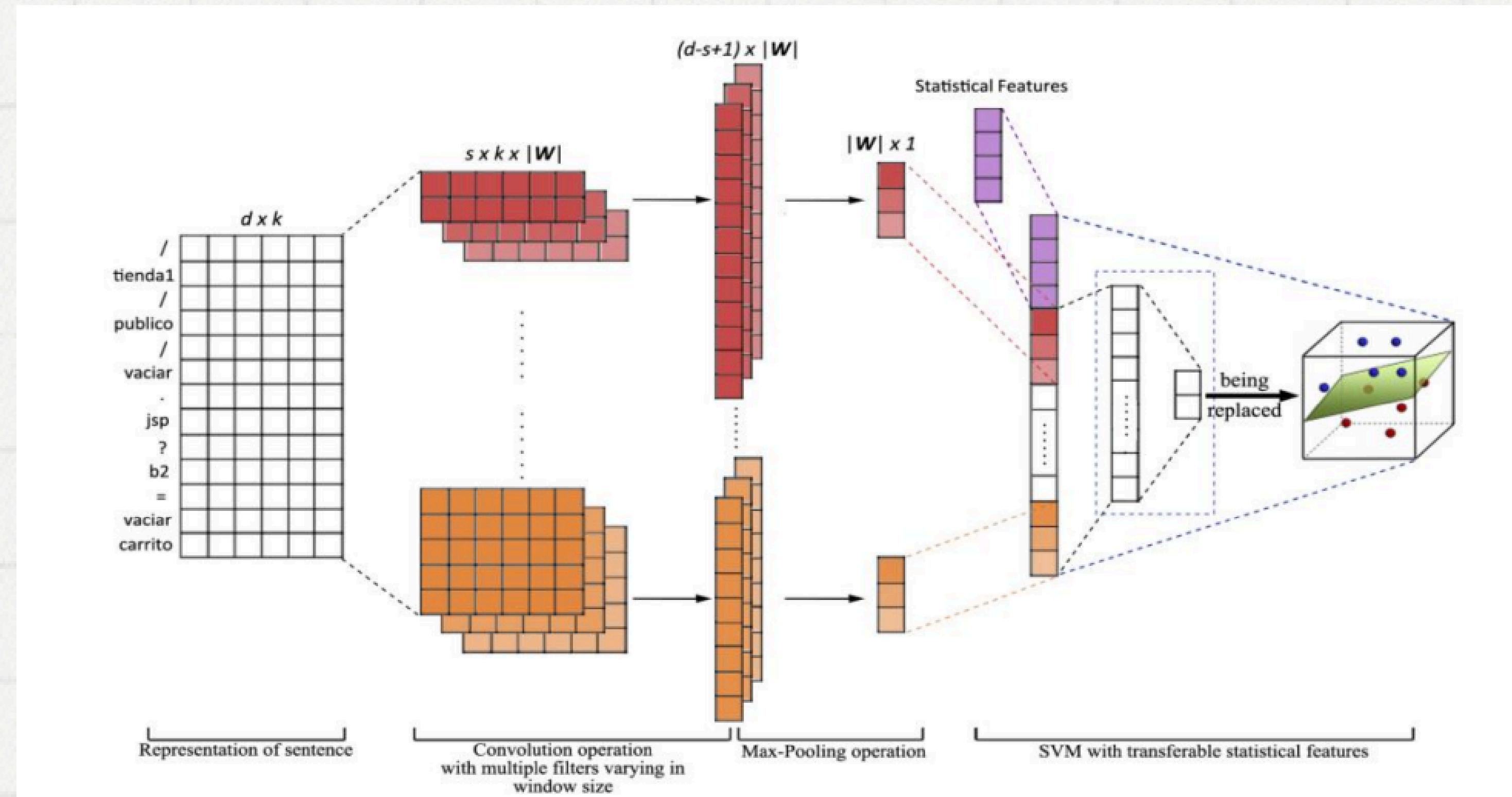
embedding matrix

- Ma trận nhúng từ này chính là lớp đầu tiên trong mô hình TextCNN.

word index

```
word_index = {'the': 1, 'cat': 2, 'sat': 3, 'on': 4, 'mat': 5}
```

# Architecture of TextCNN model



03

# DEMO

### 3. Demo

- Dựa theo kết quả đạt được ta có thể thấy độ chính xác (accuracy) đang tăng dần từ 64.45% tại vòng thứ nhất đến 99.42% tại vòng thứ 10.
- Độ chính xác trên tập kiểm tra (val\_accuracy) ban đầu là 0.86 và nhanh chóng đạt 1 tại vòng thứ 6.

```
(kali㉿ kali)-[~/Desktop/bao_mat_web] python3 textcnn.py
2024-06-05 16:14:02.368782: I external/local_tsl/tsl/cuda/cudart_stub.cc:32] Could not find cuda drivers on your machine, GPU will not be used.
2024-06-05 16:14:02.384513: I external/local_tsl/tsl/cuda/cudart_stub.cc:32] Could not find cuda drivers on your machine, GPU will not be used.
2024-06-05 16:14:02.641108: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-c
To enable the following instructions: FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2024-06-05 16:14:06.734774: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
/home/kali/.local/lib/python3.11/site-packages/keras/src/layers/core/embedding.py:90: UserWarning: Argument `input_length` is deprecated. Just remove it.
    warnings.warn(
Epoch 1/10 # Threshold for binary classification
633/633 31s 42ms/step - accuracy: 0.6445 - loss: 0.6138 - val_accuracy: 0.8608 - val_loss: 0.3764
Epoch 2/10
633/633 25s 40ms/step - accuracy: 0.8837 - loss: 0.2833 - val_accuracy: 0.9988 - val_loss: 0.0486
Epoch 3/10
633/633 25s 39ms/step - accuracy: 0.9802 - loss: 0.0804 - val_accuracy: 0.9996 - val_loss: 0.0165
Epoch 4/10
633/633 25s 40ms/step - accuracy: 0.9895 - loss: 0.0481 - val_accuracy: 0.9996 - val_loss: 0.0144
Epoch 5/10
633/633 26s 40ms/step - accuracy: 0.9936 - loss: 0.0366 - val_accuracy: 0.9998 - val_loss: 0.0071
Epoch 6/10
633/633 25s 39ms/step - accuracy: 0.9927 - loss: 0.0392 - val_accuracy: 1.0000 - val_loss: 0.0054
Epoch 7/10
633/633 25s 40ms/step - accuracy: 0.9923 - loss: 0.0397 - val_accuracy: 1.0000 - val_loss: 0.0056
Epoch 8/10
633/633 25s 39ms/step - accuracy: 0.9941 - loss: 0.0325 - val_accuracy: 1.0000 - val_loss: 0.0030
Epoch 9/10
633/633 26s 40ms/step - accuracy: 0.9932 - loss: 0.0347 - val_accuracy: 1.0000 - val_loss: 0.0045
Epoch 10/10
633/633 24s 38ms/step - accuracy: 0.9942 - loss: 0.0315 - val_accuracy: 1.0000 - val_loss: 0.0043
159/159 3s 15ms/step
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
F1-score: 1.0 Tokenizer(num_words=maximum_features)

```

### 3. Demo

- Độ mất mát (loss) giảm từ 0.61 đến 0.03.
- Mức độ sai lệch trung bình của các dự đoán trên tập kiểm tra (val\_loss) giảm từ 0.3764 xuống còn 0.0043.

```
(kali㉿kali)-[~/Desktop/bao_mat_web] input_sequences
└─$ python3 textcnn.py
2024-06-05 16:14:02.368782: I external/local_tsl/tsl/cuda/cudart_stub.cc:32] Could not find cuda drivers on your machine, GPU will not be used.
2024-06-05 16:14:02.384513: I external/local_tsl/tsl/cuda/cudart_stub.cc:32] Could not find cuda drivers on your machine, GPU will not be used.
2024-06-05 16:14:02.641108: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-c
To enable the following instructions: FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2024-06-05 16:14:06.734774: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
/home/kali/.local/lib/python3.11/site-packages/keras/src/layers/core/embedding.py:90: UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(
    Epoch 1/10
633/633 ━━━━━━━━━━ 31s 42ms/step - accuracy: 0.6445 - loss: 0.6138 - val_accuracy: 0.8608 - val_loss: 0.3764
Epoch 2/10
633/633 ━━━━━━━━━━ 25s 40ms/step - accuracy: 0.8837 - loss: 0.2833 - val_accuracy: 0.9988 - val_loss: 0.0486
Epoch 3/10
633/633 ━━━━━━━━━━ 25s 39ms/step - accuracy: 0.9802 - loss: 0.0804 - val_accuracy: 0.9996 - val_loss: 0.0165
Epoch 4/10
633/633 ━━━━━━━━━━ 25s 40ms/step - accuracy: 0.9895 - loss: 0.0481 - val_accuracy: 0.9996 - val_loss: 0.0144
Epoch 5/10
633/633 ━━━━━━━━━━ 26s 40ms/step - accuracy: 0.9936 - loss: 0.0366 - val_accuracy: 0.9998 - val_loss: 0.0071
Epoch 6/10
633/633 ━━━━━━━━━━ 25s 39ms/step - accuracy: 0.9927 - loss: 0.0392 - val_accuracy: 1.0000 - val_loss: 0.0054
Epoch 7/10
633/633 ━━━━━━━━━━ 25s 40ms/step - accuracy: 0.9923 - loss: 0.0397 - val_accuracy: 1.0000 - val_loss: 0.0056
Epoch 8/10
633/633 ━━━━━━━━━━ 25s 39ms/step - accuracy: 0.9941 - loss: 0.0325 - val_accuracy: 1.0000 - val_loss: 0.0030
Epoch 9/10
633/633 ━━━━━━━━━━ 26s 40ms/step - accuracy: 0.9932 - loss: 0.0347 - val_accuracy: 1.0000 - val_loss: 0.0045
Epoch 10/10
633/633 ━━━━━━━━━━ 24s 38ms/step - accuracy: 0.9942 - loss: 0.0315 - val_accuracy: 1.0000 - val_loss: 0.0043
159/159 ━━━━━━━━━━ 3s 15ms/step
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
F1-score: 1.0 Tokenizer(num_words=maximum_features)
```

- Các chỉ tiêu để đánh giá kết quả như accuracy, precision, recall, F1-score đều đạt giá trị cao nhất là 1.

```
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
F1-score: 1.0
```

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$F1 - score = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

- trong đó TP (True Positive) là số lượng yêu cầu bất thường được phát hiện, TN (True Negative) là số lượng yêu cầu bình thường được phát hiện, FP (False Positive) là số lượng phân loại sai yêu cầu bình thường và FN (False Negative) là số lượng phân loại sai các yêu cầu bất thường.

Link video demo

<https://youtu.be/RVh4g-p7RKk>

04

# KẾT LUẬN

- Đồ án này thực hiện việc ứng dụng một phương pháp học sâu là TextCNN để phát hiện các yêu cầu web độc hại. Trong đó word2vec dựa trên skipgam được dùng để tạo nên ma trận nhúng của các từ, đây là lớp đầu tiên của model. TextCNN có chứng năng trích xuất các tính năng trừu tượng và dùng để phân loại các yêu cầu web độc hại.
- Cơ sở dữ liệu để thực hiện đồ án là HTTP CSIC 2010 với tính lắp lại cao và khá đơn giản có thể gây sai lệch về độ chính xác.

**Thank you  
very much!**