Attention機構によるXSS攻撃検出の貢献度分析

中川 勇輝[1,a)]　三村 守[1,b)]

概要：XSS (Cross-Site Scripting) 攻撃は Web アプリケーションの脆弱性を突いた攻撃であり，様々な被害が報告されている．これに対し先行研究では，自然言語処理技術と機械学習モデルを組み合わせ，XSS 攻撃を検知する手法が提案されている．しかしながら，先行研究では攻撃の検知に貢献した特徴や，データセットの妥当性の議論は不十分である．そこで本研究では，Attention 機構を用いて重みを分析し，攻撃の検知に貢献した特徴の解明を試みた．分類モデルは，LSTM (Long Short Term Memory) と Attention 機構を組み合わせて構築した．検証実験では，複数のサイトから収集した 63,477 件の良性サンプル，および 47,012 件の悪性サンプルを用いて不均衡なデータセットを構成した．その結果，分類モデルはスクリプト言語に用いる記号や，文字コードをエンコードした際の 16 進数等，典型的な XSS 攻撃の特徴に着目しており，これらが攻撃の検知に貢献していることを確認した．さらに，データセットによって着目する特徴が異なったことから，分類モデルの汎用性および実用性については検証の余地があることが判明した．

キーワード：XSS，LSTM，Attention 機構，自然言語処理

Contribution analysis of XSS attack detection with attention mechanism

Yuki Nakagawa[1,a)]　Mamoru Mimura[1,b)]

Abstract: XSS (Cross-Site Scripting) attacks exploit vulnerabilities in web applications, and many victims have been reported. As a countermeasure for this, existing studies propose methods to detect XSS attacks by combining natural language processing techniques and machine learning models. Few studies reveal the features that contribute to the classification and the validity of the dataset. In this study, we analyzed the weights of attention mechanism and attempted to identify the features that contributed to the classification. Our models are combinations of LSTM (Long Short Term Memory) and attention mechanism. In the experiment, we used imbalanced datasets with 63,477 benign samples and 47,012 malicious samples obtained from multiple sites. The experimental result shows that typical features such as script tags and text encoded hex string contribute to the classification. Since the focused features depend on the dataset, the generality and practicality of the classification model are still to be evaluated.

Keywords: XSS，LSTM，Attention mechanism，NLP

1. はじめに

近年のサイバー攻撃において，Web アプリケーションにおける CWE-79 [1] と呼ばれる XSS (Cross Site Scripting) の脆弱性を利用した攻撃は，依然として多く発生しており，様々な被害が報告されている．日本国内においても，2022 年の 4 月から 6 月の間に XSS で 285 件の脆弱性対策情報の通知があり [2]，過去 2 年の累計で全体の脆弱性情報の通知の内 55%が Web アプリケーションの脆弱性に関するものである [3]．これに対して運用分野では，入力値の制限，スクリプトの無効化等による対策が講じられている．一方で，研究分野では，自然言語処理技術と機械学習を用いた XSS 攻撃の検知に関する様々な手法が提案されている．し

[1] 防衛大学校情報工学科
National Defense Academy of Japan
[a)] em61009@nda.ac.jp
[b)] mim@nda.ac.jp

かしながら，機械学習モデルが XSS 攻撃をどのように検知しているのか検知原理に言及した研究は少なく，データセットの実用性についても疑念があることから十分な検証がされているとは言い難い．

例えば，文献 [4]，[5]，[6]，[7]，[8] などでは様々な機械学習モデルによる XSS 攻撃の検知が試みられている．しかしながら，その一方で文献 [9] によると，XSS 攻撃の検知に関する研究報告の多くは研究に使用したデータセットが公開されておらず，データセットの実用性やサンプル数が不十分であることから結果の信用度に関して懸念があると報告されている．

それに対して先行研究 [10] では，Attention 機構の注目箇所を分析し，機械学習モデルの検知原理の解明を試みている．しかしながら，注目箇所の一例を示し，それに基づく考察が示されているのみであり，典型的な XSS 攻撃にみられる特徴に対して包括的に着目した結果であるのか疑念が残る．また，正常入力のデータセットに対するパラメータおよびディレクトリの有無といった入力の長さによる分類の容易さについても触れ，より分類が困難なデータセットを作成し検証している．しかしながら，データセットの間の比較を分類モデルの検出精度結果による比較のみで行っていることから，データセットの実用性についても疑問が残る．

このように，NN (Neural Network）を用いた XSS 攻撃の検知および分類に関する研究は盛んに行われているものの，使用されるデータセットの汎用性および実用性が不明確である．そのため，結果が導出される経緯または原因については分析されることが少なく，十分な検証には至っていない．

そこで，本研究では各先行研究で用いられたデータセットに対し，Attention 機構を用いた機械学習モデルによる分類を行い，機械学習モデルが XSS 攻撃の特徴的な要因に基づき攻撃を検知していることを貢献度分析により検証することで，検知原理の解明を試みる．これにより，NN を用いた機械学習モデルによる分類が，XSS 攻撃の典型的な特徴に正しく着目し分類していることを確認できれば，より実用的な検知精度を評価できると考えられる．また，データセットの違いによる検出精度の差がデータセットの汎用性および実用性の差と考えられることから，使用するデータセットの妥当性についても考察できるものと考える．

本論文の貢献は次のとおりである．

(1) Attention 機構を用いて，分類モデルが典型的な XSS 攻撃で使用される「<>」「%」などの単語に着目し，XSS 攻撃を検知していることを確認した．
(2) 分類モデルが着目した単語がデータセット毎に異なることから，分類モデルがデータセットに依存することを確認した．
(3) 分類モデルはデータセットに依存し，かつ分類精度結果の差が小さいことから，データセットおよび分類モデルには，汎用性および実用性に検証の余地があることが判明した．

2. 関連研究

XSS 攻撃を検知する研究として，様々な特徴に着目した手法が提案されている．

文献 [4] では，URL，HTML，JavaScript それぞれに基づく特徴を抽出した後に，分類に有用な情報量の大きい特徴のみを使用する SBS（Sequential Backward Selection）と呼ばれる逐次後退選択手法を用いて特徴量を選別し，XGBoost Model を用いて攻撃を検知している．その結果，Accuracy，Precision，Recall，F 値が全て 99%以上と高い精度となっている．しかしながら，XSS 攻撃の検知に高い精度を出すために 41 種類の特徴量を使用しており，またデータセットについても収集元とサンプル数しか示しておらず，分類モデルの実用性および汎用性には懸念がある．

文献 [6] では，URL 形式の通常の通信データおよび XSS 攻撃を含むデータに対して，前処理としてエンコード処理の後，数字および URL のホスト，スキーム部，String 文字を置換する処置を行い，可読可能な形式にデコードした上で，Word2vec および CNN-LSTM モデルを用いて検知を行っている．その結果，検出精度が Accuracy，Precision，Recall，F 値全てが 99%以上の結果となっている．しかしながら，XSS 攻撃の特徴には言及しておらず，また，通常通信のデータセットの収集元が不明であり，デコード処理などのデータセットの変形が多いことから分類モデルの実用性および汎用性に懸念がある．

文献 [8] では，まず XSS 攻撃のデータセットおよび DMOZ と呼ばれるオープンディレクトリプロジェクトのデータベースから収集した正常通信のデータセットに対し，前処理として HTML エンコードおよび URL エンコードされた部分をデコードし，数字，URL のホスト，スキーム部を置換した後，XSS 攻撃で頻出するスクリプトを 6 種類のカテゴリに分類を行う．その後，Word2vec および LSTM ベースの分類モデルによる検知処理を行っており，Precision，Recall，F 値がそれぞれ 99.5%，97.9%，98.7% の精度で検知している．また，文献 [7] では，同様の手法であるものの，データセットの件数を増やし，分類モデルを LSTM-Attention ベースの分類モデルに変更し検知を行った．その結果，Precision，Recall，F 値がそれぞれ 99.3%，98.2%，98.5%の精度で検知している．しかしながら，データセットは文献 [8] のみしか公開しておらず，データセットに対してエンコードおよびデコード処理が行われることから同じく分類モデルの実用性および汎用性に疑義が残る．

そこで，本研究では分類モデルの実用性および汎用性を検証するため，文字特徴に着目した貢献度分析による検知原理の解明を試みる．まず，先行研究 [10] と同様に，

Attention 機構を用いた文字特徴に着目した XSS 攻撃の検知を行う．その後，XSS 攻撃の検知に対する貢献度の高い文字特徴を解明することで，XSS 攻撃の典型的な特徴を包括的に着目しているかを貢献度分析により確認する．これにより，分類モデルが XSS 攻撃を検知する根拠を明らかにし，実用的な XSS 攻撃の検知を行っているかを分析する．

3. 関連技術

3.1 LSTM

LSTM（Long Short Term Memory）[11] は RNN (Recurrent Neural Network) の 1 種で，RNN の持つ勾配消失問題を解決したモデルの一つである．RNN は隠れ層を用いて，前の出力を次の出力として用いる NN であるが，長い時間軸では情報が取り扱えない問題があった．これを忘却ゲート (1)，入力ゲート (2)，記憶候補セル (3)，記憶セル (4)，出力ゲート (5)，隠れ層 (6) を使用して解決したものが LSTM である．モデルを図 1 に示し，図中の各ゲートにおける式を以下に示す [12].

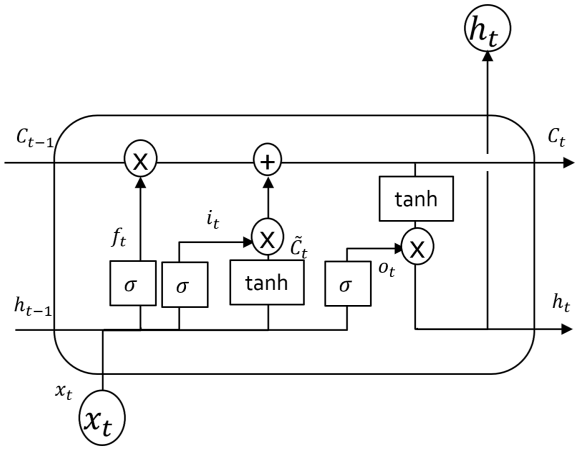

図 1 lstm モデル

忘却ゲート f_t は入力 x_t と前の状態の隠れ層 h_{t-1} からどの程度情報を廃棄するかを決定しており，以下の式で定義される．W は重みベクトル、b はバイアスを表す.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t]) + b_f \tag{1}$$

入力ゲート i_t は、どの情報をセルに記憶するかを決定しており，以下の式で定義される.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t]) + b_i \tag{2}$$

記憶候補セル $\tilde{C}_t$ は長期記憶として保存される記憶セルに加えられる候補値を示しており，以下の式で定義される.

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t]) + b_C \tag{3}$$

記憶セル C_t は忘却ゲート f_t，前の状態の記憶セル C_{t-1}，入力ゲート i_t および記憶候補セル $\tilde{C}_t$ によって更新され，以下の式で定義される.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

出力ゲート o_t は記憶セル C_t と組み合わせ出力値を決定するのに使用され，以下の式で定義される.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t]) + b_o \tag{5}$$

隠れ層 h_t は入力 x_t に対する出力値でもあり，出力ゲート o_t と記憶セル C_t によって，以下の式で定義される.

$$h_t = o_t * tanh(C_t) \tag{6}$$

3.2 Attention 機構

Attention 機構は，Encoder-Decoder モデルに導入される要素の関係性や注意箇所を学習する機構である．Transforer と呼ばれる Encoder-Decoder モデルを実装している．このうち，本研究では Self-Attention と呼ばれる機構を活用した [13]．Self-Attention は入力データ内の照応関係を同じ入力データを用いて獲得する手法である．入力データから，Query，Key，Value の各ベクトルを算出し，Query，Key の類似度を Softmax 関数によって重みとして計算する．この重みから Query と Value の照応関係を獲得し出力する．Query を Q，Key を K，Value を V とすると，類似度は以下の式で定義される.

$$Attention(Q, K, V) = Softmax(\frac{Q \cdot K^T}{\sqrt{d_k}}) \cdot V \tag{7}$$

4. 検証手法

4.1 概要

本研究では LSTM に Attention 機構を組み合わせた分類モデルにより XSS 攻撃を検知し，Attention 機構により着目した注意箇所への重みに基づく XSS 攻撃に対する注意特徴の解明を行う．検証の手順を図 2 に示す.

まず，正常通信データからの良性サンプルおよび XSS 攻撃時の通信データからの悪性サンプルを，テキストデータとして訓練データおよびテストデータに分割する．次に，訓練データおよびテストデータの前処理として，データクレンジングおよびトークナイズ処理を行う．なお，トークナイズ処理とは，テキストデータをトークンと呼ばれる最小単位に区切り，各トークンに ID を振ることを指す．その後，訓練データによる分類モデルの訓練を行い，訓練済みの分類モデルを用いたテストデータの 2 値分類による検出精度を確認する．その際，分類時に Attention 機構が着目した注意箇所の重み情報から，重みの強い文字および単語をトークンとして抽出，集計し，文字列として注意箇所の傾向を確認する．この傾向から，Attention 機構による分類に有用であったトークンを確認し，分類に XSS 攻撃の特徴的な語句が用いられているかを分析する.

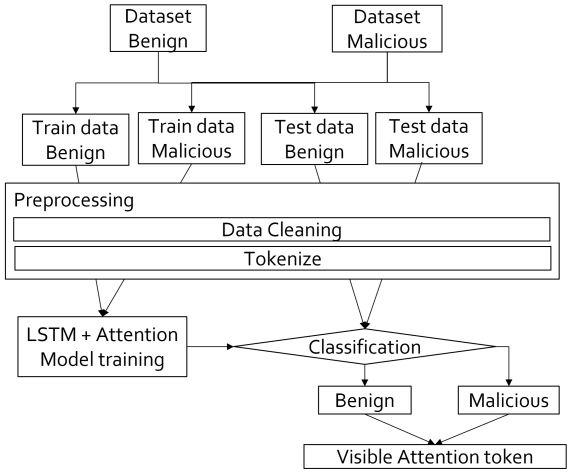

図 2 検証の手順

4.2 前処理

前処理では，データに対するデータクレンジングおよびトークナイズ処理を行う．トークナイズ処理の前準備として，トークナイズ処理に用いる辞書の作成を行う．辞書は，Kaggle より収集した Benign，Malicious の URL を集めた 450,176 件の URL リストのテキストデータから，Keras ライブラリの Tokenizer モジュールを用いてユニークな単語に分割した 452,020 語に，テキストの開始位置などを示す特殊文字を 1,000 語追加した 453,020 語を使用した．また，「. 」「%」「/」などの記号も 1 単語として辞書に設定している．

4.2.1 データクレンジング

入力データに対するデータクレンジングとしてテキストに対する処置を行う．URL のホスト部，スキーム部のみを除去し，その他のパラメータやディレクトリが記載されるクエリ部分は URL に入力されているテキストを直接使用する．また，XSS 攻撃でよく行われる他の文字コードを用いた難読化に関しても，復号化の処理を行わず，通信時の HTTP 通信のログのまま使用する．これは，XSS 攻撃が発生する実環境に近いデータセットを構成するためである．

4.2.2 トークナイズ処理

入力データのトークナイズ処理として，作成した辞書に基づき Keras ライブラリの Tokenizer モジュールを用いて入力データのテキストを最小単位に分割し，ユニークな単語にごと固有の番号を付与する．これによりテキストデータを分類モデルが訓練可能な数値データに変換する．

4.3 実装およびパラメータ設定

本実験で使用した機械学習モデルは Tensorflow 2.4 [14]，Keras 2.4.3 [15]，Python-3.8.9，scikit-learn 1.0.2 [16] を用いて実装した．

パラメータチューニングは手動で複数例を行い，そのうち最も良い値を使用した．隠れ層の次元を 128，epoch を 16，dropout 値を 0.2 とした．

4.4 データセット

2 種類のデータセットを用いて比較を行う．データセットは妥当性および再現性を担保するため，URL のテキストデータを使用しており，かつ XSS 攻撃検知に関する文献において公開されているものを使用した．

1 つ目は，文献 [8] で使用されたデータセットであり，XSSed.com よりクローリングして収集した 33,426 件の攻撃データおよび DMOZ データから作成した 31,407 件の正常データからなる．しかしながら，先行研究 [10] では，正常データの方にディレクトリやパラメータが含まれているものが少なく，データ長による分類が容易だとする意見が述べられている．

2 つ目が，先行研究である文献 [10] で使用されたデータセットであり，XSSed.com よりクローリングして収集した 13,586 件の攻撃データおよび CIC-IDS2017 データセットより抽出した 32,070 件の正常データからなる．このデータセットは，正常データにもディレクトリやパラメータが含まれており，1 つ目のデータセットに比べ分類は困難であると考えられる．

使用したデータセットのサンプル数の内訳は，表 1 のとおりである．

表 1 使用データセットのサンプル数内訳

データセットの種別	Benign	Malicious
文献 [8]	31407	33426
文献 [10]	32070	13586

4.5 分類モデル

文献 [7] を参考に，LSTM と Attention 機構を組み合わせた分類モデルを使用し分類を行う．モデルの構成については，図 3 のように Keras ライブラリを使用し構成した．

まず Input レイヤに入力された単語を ID 化したベクトルを入力する．次に，Input レイヤを Embedding レイヤに入力し，整数インデックスをベクトルにマッピングする．本研究においては 128 次元ベクトルとした．続いて，SeqSelfAttention レイヤをによって重み処理をできるようにする．その後，Bidirectional レイヤでの LSTM の訓練および GlobalMaxPooling レイヤでのダウンサイジングを行う．ダウンサイジングしたベクトルを Dense レイヤで全結合し，Dropout レイヤによる過学習を抑制した後，最後の Dense レイヤで結果を出力している．なお，活性化関数にはシグモイド関数を使用した．

4.6 Attention 機構による注意箇所の集計

分類モデルにより，トークナイズされた入力データの各

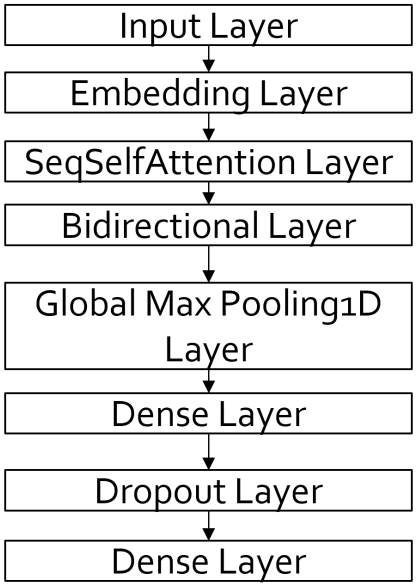

図 3 分類モデル構造

トークンには分類に用いられる重みがつく．この重みに準じ，1 サンプルごと重みの高いトークンから上位 10 番目までを抽出する．抽出されたトークンを集計し出現回数の多い順に並び替え，上位にあるトークンほど，データセットの分類に対して重要なトークンであるとみなし，分類結果に応じて考察を行う．なお，この際にスペシャルトークンが集計された場合は除外する．

5. 検証実験

5.1 実験内容

検証実験として 2 種類のデータセットに対して同一の分類モデルを用いて分類を行い検出精度を確認したのち，分類結果ごと分類のために注意したトークンを抽出し集計を行う．

本実験では，正常な URL を Benign，XSS 攻撃で用いられる悪性スクリプトが含まれる URL を Malicious とし，それぞれを正しく予測しているかを検証している．それぞれの関係は表 2 のとおり．

表 2 Benign と Malicious の入力および予測関係

	予測結果	
	正常入力	攻撃入力
実際の入力（Benign）	TN	FP
実際の入力（Malicious）	FN	TP

分類モデルの訓練には，訓練サンプル数を合わせるため，各データセットから Benign および Malicious を同数の 10,000 件ずつランダムに抽出し訓練データとして使用した．テストデータに関しては，実用性を考慮し Malicious の割合が少ない不均衡なデータセット構成とした．また，両データセットで同一サンプル数とするため，両データセットとも Benign を 12,000 件，Malicious を 1,000 件ランダムに抽出しテストデータとして使用した．

評価指標として，Accuracy，Precision，Recall，f1 のそれぞれを用いて検証した．それぞれの評価指標は以下の式で表される．

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{8}$$

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

$$f1 = \frac{2Recall \times Precision}{Recall + Presicion} \tag{11}$$

また，各データセットごと分類に貢献した特徴に対する分析を行うため，1 サンプルごと Self-Attention による重みの高い単語を抽出し，出現回数を集計した．

これをすべてのサンプルに対して行い，抽出したすべてのトークンをサンプルが分類された TP，FP，TN，FN の 4 象限ごとに集約する．その後，出現回数の多い順に並び替え，上位のトークンの比較を行い，分類された 4 象限ごと，分類時に注意されたトークンの傾向について確認し，考察を行った．

5.2 実験環境

実験に使用した環境を表 3 に示す．

表 3 実験環境

CPU	Core i7-8700K 3.70GHz
GPU	NVIDIA GeForce RTX 2080 Ti
Memory	64GB
OS	Windows10 Home
使用言語	Python3.8.9

5.3 実験結果

分類結果と検出精度を表 4〜7 に示す．検出精度は両データセットともに高い数値であり，表 5 の Precision 結果からやや見逃しがあるものの，最低でも 98%以上の検知精度があることを確認した．

また，表 6 より文献 [10] のデータセットにおいては，XSS 攻撃入力の見逃しがなく高い精度で攻撃を検知できていることがわかった．

TP，FP，TN，FN の 4 象限に分類し，それぞれ分類されたサンプルごとにトークンを抽出し集計した．集計結果は，表 8，表 9 に示す．なお，表 9 に示す出力結果にあるように，分類結果の件数が 0 の場合は抽出されるトークン

表 4　文献 [8] データセットの分類サンプル数

		Predict		Total
		Benign	Malicious	
Actual	Benign	11920	16	11936
	Malicious	13	979	992
Total		11933	995	12928

表 5　文献 [8] データセットの分類精度

	Precision	Recall	f1-score	support
Benign	1.00	1.00	1.00	11936
Malicious	0.98	0.99	0.99	992
Accuracy			1.00	12928
macro avg	0.99	0.99	0.99	12928
weighted avg	1.00	1.00	1.00	12928

表 6　文献 [10] データセットの分類サンプル数

		Predict		Total
		Benign	Malicious	
Actual	Benign	11909	25	11934
	Malicious	0	994	994
Total		11909	109	12928

表 7　文献 [10] データセットの分類精度

	Precision	Recall	f1-score	support
Benign	1.00	1.00	1.00	11934
Malicious	0.98	1.00	0.99	994
Accuracy			1.00	12928
macro avg	0.99	1.00	0.99	12928
weighted avg	1.00	1.00	1.00	12928

表 8　文献 [8] データセットの分類貢献トークン

rank	TN	FP	FN	TP
1	=	2f	=	/
2	/	/	id	+
3	id	+	&	2f
4	;	=	.	-
5	amp	-	_	&
6	_	html	-	20
7	-	3a	20	_
8	cid	uk	/	id
9	ring	.	d	3e
10	&	asp	pne	;
11	+	support	telia	3a
12	aid	microscopy	com	title
13	record	micropolitan	l	20s
14	reviews	fresh	en	y
15	article	algae	php	3c
16	1023	frame	5c	p
17	name	amm	message	type
18	search	field	100	2fm
19	page	sub	submit	]
20	p	band	go	<

はないため空欄としている．

結果として，TN および TP の両側に共通して URL に含まれる記号に多く着目していることを確認した．特に，Malicious として TP に分類されたサンプルにおいては，XSS 攻撃で用いられる HTML や JavaScript のタグを表す記号<>や，それらの記号の文字コードをエンコードして示した際の 3e，3c といった 16 進数表示のトークンに着目していることが分かった．また，表 8，表 9 を比較したところ，同一の記号に着目している一方で表 8 では Benign の上位にあるトークンである「amp」が表 9 では Malicious の上位に含まれており，データセットによって着目する特徴の文字が分類結果によらないことがわかった．

表 9　文献 [10] データセットの分類貢献トークン

rank	TN	FP	FN	TP
1	-	french		/
2	d8	cee		%
3	/			-
4	_			(
5	board			com
6	83			<
7	84			3c
8	b8			-
9	b5			amp
10	tob			+
11	b4			1
12	82			73
13	b0			74
14	a8			alert
15	d9			gov
16	81			id
17	jp			2f
18	recom			register
19	8f			phmsa
20	05			?

6. 考察

6.1 検出精度

実験の検出精度および関連研究から引用した検出精度を表 10 に，実験サンプル数を表 11 に示す．

検出精度は，十分なパラメータの最適化を行っていないが，一定以上の検出精度があることを確認した．これは，データセットへの依存度によるものと考えられ，Benign と Malicious のデータセット構成が分類に容易な構成であることが原因のひとつとして考えられる．十分なパラメータチューニングによる検出精度の検証については今後の課題である．

6.2 貢献度分析

表 8，表 9 より 2 種類のデータセットから抽出したトークンを確認したところ，各クラスとも，URL に用いられる「.」，「/」，「=」記号や URL エンコードされた際の「%」記号に多くの注意が向けられていることがわかった．特に，

表 10　データセットごとの精度比較

文献等	Accuracy	Precision	Recall	f1
実験 (文献 [8])	0.998	0.984	0.987	0.985
実験 (文献 [10])	0.998	0.976	1.0	0.988
文献 [4]	0.996	0.995	0.990	0.993
文献 [5]	0.993	0.992	0.984	0.988
文献 [6]	0.991	0.999	0.995	0.993
文献 [7]	-	0.993	0.982	0.985
文献 [8]	-	0.995	0.979	0.987
文献 [10]	-	1.0	0.97	0.98

表 11　データセットごとのサンプル数比較

文献等	通常サンプル数	攻撃サンプル数
実験 (文献 [8])	31,407	33,426
実験 (文献 [10])	32,070	13,586
文献 [4]	100,000	38,569
文献 [5]	100,000	38,569
文献 [6]	74,000	40,000
文献 [7]	78,652	32,168
文献 [8]	31,407	33,426
文献 [10]	32,063	13,593

濃い色で着色したトークンは，4 つの区分のうち複数のクラスで上位に含まれることものの，分類に寄与する特異な差が確認できなかった．すなわち，分類に重要なトークンであるものの，単独で分類に大きな影響をあたえるものではなく，他の固有なトークンと同時に分類に用いられることで重要な要素になるものと推察される．

また, 表 8 の TP 列および表 9 の TP 列では，薄い色で着色した部分で示した典型的な XSS 攻撃で確認される<>といったタグを表す記号や 2f，3e，3c といった URL エンコードされた文字をデコードする前の 16 進数表記部分と考えられるトークンに注意している傾向がわかる．URL エンコードにおいて「%2f」は「/」,「%3e」は「 >」,「%3c」は「<」であることから，XSS 攻撃で使用されるスクリプト部分に注意していると考えられる．

抽出されるトークンはデータセットに依存することから，入力データに頻出の「.」,「/」,「=」,「-」といった記号が各分類クラスの上位に共通して存在するものの，それ以外の各クラスの上位に含まれるトークンは使用したデータセットにおいては分類の影響が大きいトークンであると考えられる．すなわち，分類モデルは訓練によって XSS 攻撃で用いられる通信データの言語特徴と通常通信データの言語特徴をとらえ，それに基づき重要と判定した要素が上位に含まれており，「 >」などの典型的な XSS 攻撃の特徴に着目していることが確認できたものと考えられる．

その一方で，データセットによって着目する上位のトークンの種類および分類されるクラスに差があることから分類モデルの汎用性および実用性には検証の余地があると考えられる．

6.3　研究倫理

入手したデータセットは全て公開されたものを使用しており，再現性については確保している．また，XSS 攻撃の検知原理について解明を試みているものの，分類モデルの訓練データに依存する要素が大きく，攻撃の検知を回避することにはつながらないものと考える．

6.4　研究の限界

本研究で使用したデータセットは，攻撃データとして使用したのものが XSSed.com より収集したデータセットであり，実際の攻撃入力で用いられる文字列ではない．また，正常なデータも 2 種類のデータセットからから抽出したデータを使用しているが，データの偏りについては検証する余地があると考えられる．

実際の攻撃入力のデータセットは公開されているものからは見つけられなかったため，実際の攻撃入力データを用いることでより実情に即した注意箇所の状況が確認できる余地があると考える．

7.　おわりに

本研究では，LSTM と Attention 機構を組み合わせた機械学習モデルを用いて，XSS 攻撃を検知する際の Attention 機構の重みを分析することで XSS 攻撃検知に貢献する特徴の解明を試みた．考察の結果，XSS 攻撃に用いられる典型的な特徴に着目して分類していることがわかったものの，着目しているトークンの種類および検出精度はデータセットにより異なることから分類モデルの汎用性および実用性については検証の余地があり，今後の課題である．

謝辞　本研究は JSPS 科研費 21K11898 の助成を受けたものです．

参考文献

[1] Corporation, T. M.: CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') (2006–2022). https://cwe.mitre.org/data/definitions/79.html.

[2] IPA: 脆弱性対策情報データベース JVN iPedia に関する活動報告レポート [2022 年第 2 四半期（4 月〜6 月）], https://www.ipa.go.jp/files/000099946.pdf (2022). (Accessed on 07/27/2022).

[3] IPA: ソフトウェア等の脆弱性関連情報に関する届出状況 [2022 年第 2 四半期（4 月〜6 月）], https://www.ipa.go.jp/files/000099955.pdf (2022). (Accessed on 07/27/2022).

[4] Mokbal, F. M. M., Dan, W., Xiaoxi, W., Wenbin, Z. and Lihua, F.: XGBXSS: An Extreme Gradient Boosting Detection Framework for Cross-Site Scripting Attacks Based on Hybrid Feature Selection Approach and Parameters Optimization, *Journal of Information Security and Applications*, Vol. 58, p. 102813 (online), DOI: https://doi.org/10.1016/j.jisa.2021.102813 (2021).

[5] Mokbal, F. M. M., Dan, W., Imran, A., Jiuchuan, L., Akhtar, F. and Xiaoxi, W.: MLPXSS: An Inte-

grated XSS-Based Attack Detection Scheme in Web Applications Using Multilayer Perceptron Technique, *IEEE Access*, Vol. 7, pp. 100567–100580 (online), DOI: 10.1109/ACCESS.2019.2927417 (2019).

[6] Raed, W. K. and Methaq, T. G.: A hybrid of CNN and LSTM methods for securing web application against cross-site scripting attack, *Indonesian Journal of Electrical Engineering and Computer Science*, Vol. 21, No. 2, pp. 1022–1029 (online), DOI: 10.11591/ijeecs.v21.i2.pp1022-1029 (2021).

[7] Lei, L., Chen, M., He, C. and Li, D.: XSS Detection Technology Based on LSTM-Attention, *2020 5th International Conference on Control, Robotics and Cybernetics (CRC)*, pp. 175–180 (online), DOI: 10.1109/CRC51253.2020.9253484 (2020).

[8] Fang, Y., Li, Y., Liu, L. and Huang, C.: DeepXSS: Cross Site Scripting Detection Based on Deep Learning, *Proceedings of the 2018 International Conference on Computing and Artificial Intelligence*, ICCAI 2018, New York, NY, USA, Association for Computing Machinery, p. 47–51 (online), DOI: 10.1145/3194452.3194469 (2018).

[9] 飯野和真, 宇田隆哉: 不適切なデータセットや処理方法を用いた機械学習による XSS 攻撃検出研究の解説と精度の比較, 技術報告 20, 東京工科大学, 東京工科大学 (2021).

[10] 宮崎裕一郎, 三村　守: ONLSTM と Attention 機構による XSS 攻撃の検知に関する一考察, 技術報告 11, 防衛大学校, 防衛大学校 (2021).

[11] Hochreiter, S. and Schmidhuber, J.: Long Short-Term Memory, *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780 (online), DOI: 10.1162/neco.1997.9.8.1735 (1997).

[12] Olah, C.: Understanding LSTM Networks, `http://colah.github.io/posts/2015-08-Understanding-LSTMs/` (2015). (Accessed on 07/28/2022).

[13] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I.: Attention is all you need, *Advances in neural information processing systems*, Vol. 30 (2017).

[14] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y. and Zheng, X.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems (2015). Software available from tensorflow.org.

[15] Chollet, F. et al.: Keras, `https://keras.io` (2015).

[16] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E.: Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research*, Vol. 12, pp. 2825–2830 (2011).