L-023

サポートベクタマシンを用いた Web アプリケーションの攻撃検知

清水 大貴 † 小高 知宏 † 黒岩 丈介 † 諏訪 いずみ ‡ 白井 治彦 ‡
† 福井大学工学研究科 ‡ 福井大学工学部

1. はじめに

近年, インターネットの普及に伴い,web アプリケーションの利用者が増加している. そのため, データベースに格納されている個人情報を脅威から守るために管理を徹底する必要がある. web アプリケーションを対象とした外部からの攻撃 (XSS,SQL インジェクション等) の対策として WAF(Web Application Firewall) が挙げられ実際に運用されている.

先行研究では外部からの入力データである HTTP リクエストに対して特徴抽出を行い, 生成した特徴ベクトルを用いて攻撃検知を試みている [1][2]. 本研究では,HTTP リクエストの特殊文字に着目し, 攻撃と正常入力に対して特徴抽出を行う. また, 生成された特徴ベクトルを機械学習アルゴリズムを用いて分類を行う.

2. Web アプリケーション

Web アプリケーションとは, ブラウザから利用可能なアプリケーションサービスのことである. クライアントとサーバ間で HTTP 通信を利用してデータの送受信を行っている.

HTTP 通信はステートレスな通信であり, クライアントサーバ間で HTTP メッセージの送受信を行う. クライアントからサーバへの HTTP メッセージは HTTP リクエストとであり, リクエスト内部はヘッダとボディで構成されている. 主に, 利用者からの入力は HTTP リクエストのヘッダ部分に現れる.

Web アプリケーションへの攻撃は様々あり, ここでは代表的な攻撃手法を表 1 に挙げる. 表 1 の攻撃手法はいずれも Web アプリケーションを標的としている. 主な被害として,Cookie 値の漏洩, データベース内の個人情報流出など様々である.

表 1 では, 各攻撃手法に出現する特殊文字の入力を示したものである. これらの攻撃手法は入力スクリプトに特殊文字が含まれている. そのため,HTTP リクエストのヘッダ部分に表 1 の特殊文字が現れる場合, 攻撃を受けている可能性がある. そこで, 本研究では出現する特殊文字に着目した特徴量抽出を 3 章で述べる.

Attack detection of web application using support vector machine
†Daiki Shimizu †Tomohiro Odaka †Jousuke Kuroiwa ‡Izumi Suwa ‡Haruhiko Shirai
†Graduate School of Engineering, University of Fukui
‡Faculty of Engineering, University of Fukui

表 1: 各攻撃手法に出現する特殊文字

攻撃名	特殊文字
XSS(Cross-Site-Scripting)	< > = · ;
SQLI(SQL-Injection)	’ sp +
DT(Directory-Traversal)	/ · \

3. Support Vector Machine

SVM は 2 クラスパターン識別器を構成する手法である. カーネル法による SVM では決定関数 $\hat{f}(x)$ は以下の式で定式化されている.

$$\hat{f}(x) = sgn(\sum_{v} \alpha_i y_i K(x_i, x) + b)$$

ここで,x は入力ベクトル,y は予測値,$K(x_i, x)$ はカーネル関数,α はラグランジュ乗数,b はバイアスパラメータである. また, 本研究では次の二つのカーネルを使用する.

線形カーネル : $x_i^T \cdot x$

ガウスカーネル : $exp(-\gamma||x_i - x||^2)$

SVM は分類境界と最も近いデータとの距離 (マージン) を最大化することで, 汎化誤差が最小になるような分類境界を求める.

3. 特徴量抽出

HTTP リクエストのクエリ文字列における特殊文字列に着目し, 特徴量抽出手法とする. 本手法は,HTTP リクエストのクエリ文字列の有無, 各入力パラメータに含まれている特殊文字 (表 2) の出現回数から生成した特徴量である.

特殊文字の有無は, 入力パラメータ内に, 特殊文字が存在する場合 0, 存在しない場合 1 として表現を行う. 特徴量の大きさは対象 HTTP リクエストの入力パラメータの数により異なる.

4. 自己符号化器の適応

本研究の特徴量抽出手法では, 次元数が入力パラメータの数によって異なり, 冗長的である. よって, 生成した特徴量に自己符号化器を適応させて冗長性の改善を試みる.

Copyright © 2019 by Information Processing Society of Japan and
The Institute of Electronics, Information and Communication Engineers
All rights reserved.

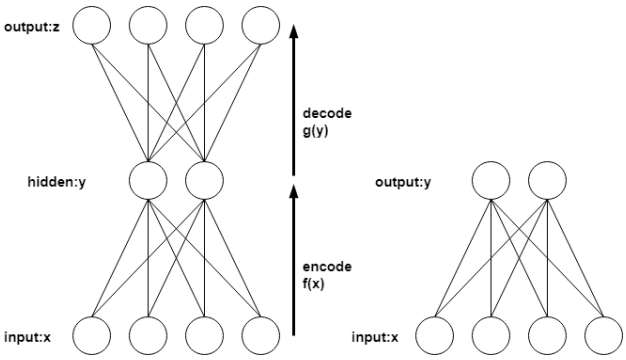

図 1: 自己符号化器のモデル

自己符号化器とは教師無し学習の深層学習の手法のひとつである．ネットワークは encode 部と decode 部で構成されている．図 1 左に示すように, 入力 x から encode によって隠れ層 y を得て,decode によって入力 x の再構築を目指し z を出力する．encode と decode は以下で表される．

$$encode : y = f(x) = act(Wx + b)$$

$$decode : z = g(y) = act(W'y + b')$$

ここで,W は重み行列で b はバイアス項である．act() は活性化関数を表している．また, 誤差関数は平均二乗誤差を用いる．

隠れ層の出力 y は入力 x より少ない次元数で表現することが可能になり, この出力 y を特徴量として用いる (図 1 右).

5. 実験方法

本実験では, 機械学習アルゴリズムとして SVM を用いる．また, 線形カーネルとガウスカーネルの二つに対して分類を行う．実行環境として,Python の機械学習ライブラリである scikit-learn を使用する．また, 自己符号化器の実装は TensrFlow を使用する．

実験を行うにあたり,HTTP リクエストは [3] より入手した．クエリ文字列内の入力パラメータが 5 個の正常入力 1000 個, 攻撃入力 1000 を dataset1, クエリ文字列内の入力パラメータが 13 個の正常入力 1000 個, 攻撃入力 1000 個を dataset2 とし, 本実験のデータセットとする．ここで, 抽出した HTTP リクエストは GET メソッドのみとしている. また, 訓練データとテストデータとして 7:3 で分割を行った．上記データセットから特徴量抽出を行うと dataset1 では 185 次元,dataset2 では 481 次元となっている．

評価を行うにあたって, 使用する評価項目として, 正解率, 適合率, 再現率,F 値を用いる．

6. 結果・考察

二つのデータセットに対して,SVM による分類結果を表 3 に示す. ここでハイパーパラメータは線形カーネルでは $C = 10$, ガウスカーネルでは $C = 100, \gamma = 0.1$ とした．また, 自己符号化器を適応させた結果は当日示す.

表 3 より,dataset1 に関して, 線形カーネルとガウスカーネルの両方とも高い精度であったが,dataset2 に関しては dataset1 と比較すると精度が下がってしまったことがわかる．また, 両カーネルでの精度の差から小さくはあるがガウスカーネルの方が分類できているといえる.

特に入力パラメータが 5 個と少ない dataset1 では両カーネルとも正解率が 99 %程度となり, その他の値も非常に高い良い結果となったことから, 特徴空間上で十分な分類が可能であるといえる.

表 2: 特殊文字一覧

sp	!	“	#	$	%	&	’	(	)	*
+	,	-	·	/	:	;	<	=	>	?
@	[	\	]	^	_	‘	{	|	}	~

表 3: SVM による分類結果

	線形カーネル		ガウスカーネル	
	dataset1	dataset2	dataset1	dataset2
Accuracy	0.995	0.940	0.997	0.943
Precision	1.000	1.000	0.993	1.000
Recall	0.990	0.877	1.000	0.884
F-measure	0.995	0.934	0.997	0.938

参考文献

[1] 清水大貴 小高知宏 黒岩丈介 白井治彦 諏訪いずみ, Web アプリケーションの攻撃検知における機械学習手法の比較, 電気関係学会北陸支部連合大会, E-18, 2018.

[2] 伊波靖 高良富夫, サポートベクタマシンを用いた WAF への異常検知機能の実装と評価, 情報処理学会論文誌コンピューティングシステム, Vol.7(1), pp.1-13, 2014.

[3] HTTP DATA SET CSIC 2010 http://www.isi.csic.es/dataset/

Copyright © 2019 by Information Processing Society of Japan and
The Institute of Electronics, Information and Communication Engineers
All rights reserved.