2N-08

SQLインジェクションに対する機械学習を用いた攻撃検知手法の提案

井村　悠成[†1]　　岸　知二[‡2]

[†1‡2]早稲田大学　創造理工学部　経営システム工学科

1. 序論

Web アプリケーションの開発が盛んであるが、そのセキュリティを高めるための設計やレビューには専門的な知識や技量が必要となり，小さな組織ではそうした技量を持った人材を揃えづらい。そこで本研究では，そうした小さい組織でも比較的容易に導入できる属人性を低くした対策，具体的には異常検知に取り組むことにする。

2017 年 OWASP が発表した最も重大なウェブアプリケーションリスクトップではインジェクション問題がトップに位置している．今回はそれらの中でも特に重要度が高い SQL インジェクションを異常検知の対象にする。

2. 従来研究

2.1. Rawat らの手法[1]

SQL インジェクションに対して機械学習を用いて検知する研究は複数あるが、ここでは SQL インジェクションに対する SVM を用いた攻撃を検知する手法の流れを(1)~(3)に示す。

(1) 学習データを SQL クエリと SQL インジェクションを起こすクエリに分類する。

(2) 各クエリを単語に分解し重複しない単語リストである BoW を生成する。

(3) 単語を特徴ベクトル化し学習させたモデルが SQL クエリか SQL インジェクションを起こすクエリか分類する。

この手法では SQL クエリと SQL インジェクションを起こすクエリを SVM で分類し数百件のデータに対する正答率は 96%に到達する。

2.2. 従来研究の課題

従来研究では、数千件のデータを学習させ精度を調査しているが、現実の課題に適用した際にデータ数の増加による学習時間の変化を検討できていない。データ数が増加すると BoW を用いて作成した辞書の量が増加し学習時間が増えてしまう。BoW を用いる際には、次元の設定が難しくうまく処理しないと必要なデータ数が増大する次元の呪いが発生する可能性が生じる。そこで、本研究では、SQL インジェクションの特徴を踏まえた辞書構築とそれを利用した検知手法を提案する。

「A SQL injection attack Detection method using machine learning」

†1 Yusei Imura ・Waseda University

‡2 Tomoji Kishi ・Waseda University

3. 提案手法と評価実験

3.1.提案手法

佐野らは，表 1 にある文字列及び下記の予約語 select, if, where, set, char, or, insert, alter, create, from, delete, table を SQL インジェクションの攻撃検知に効果が高いと報告している。[2]

そこで、提案手法では、これらの文字を基にしたオリジナルの辞書を用いることで攻撃検知を行う。手法の概要を図 1 に示す。提案手法の特徴は、学習データを特徴ベクトル化する際に SQL インジェクションの検知に効果がある単語に絞った抽出である。そのため、図 1 で定めた出現頻度 80%以下の単語リストと上記の重要単語の組み合わせを用いて特徴ベクトル化を行う。ここで出現頻度とは，各単語の出現数の全単語数に対する割合を意味する．

ここで出現頻度 80%以下の単語を用いたのは予備実験を行なった結果である。予備実験では、辞書内の単語を出現頻度別で層別し各々の頻度で試行した際の処理時間と正答率を比較しもっとも正答率が高い中で処理時間が最小の 80%を選んだ。

表 1. 攻撃の特徴を抽出する文字列一覧[2]

半角スペース	セミコロン	シングルクォーテーション	右側丸括弧	左側丸括弧
	：	`	)	(

注）文字列の中でも攻撃の特徴を抽出する効果が高い 5 つの文字に絞る。

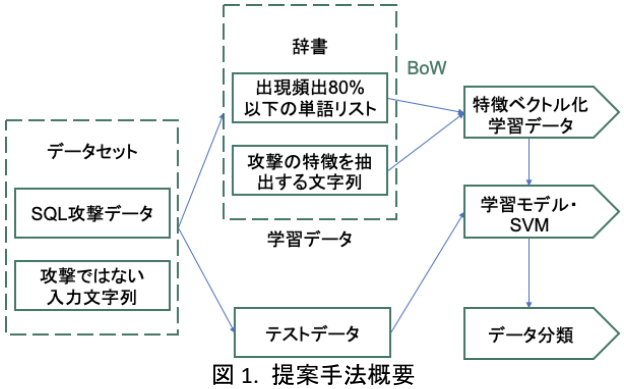

図 1. 提案手法概要

3.2. 評価実験

(1) 実験の目的

提案手法の有効性を示すことを目的とし処理時間及び

Copyright ©2019 Information Processing Society of Japan.
All Rights Reserved.

正答率を既存手法と比較する。また、データ量の変化によって処理時間及び正答率がどのように変化するかを確認する。

(2) 実験の条件や方法

既存手法[1]と提案手法に対して同様条件のもと学習および予測をそれぞれ 100 回行いそれぞれの正答率、処理速度の比較を行う。

今回用意したデータセットは、一般的な Web の入力フォームを想定し、攻撃を行わない入力文字列を 204 個作成した。この文字列の集合は記号を多用した英語のテキスト文、Wiki 文法に基づいたサンプル、顔文字のサンプルで構成している。また、攻撃を行う文字列は、1130 個作成した。この文字列の集合には、OSS で脆弱性を検証することができるアプリケーションである DVWA[4]に対して、SQL インジェクション攻撃を仕掛けることができる SQLMAP[5]を用いて図２のコマンドを実行し攻撃ログを 1000 件収集する。

```
sqlmap.py-o-u
"http://localhost/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit"--
cookie="PHPSESSID=db89cb39bb5dfd58197da6d105f143ab;
security=high"
```

図 2.　SQL インジェクション攻撃するコマンド

このコマンドはブラインド SQL と呼ばれる SQL 文かつデータベースの作成およびデータのダンプを行う SQL 文に限定されてしまうため、これらだけを学習させると過学習が起きる。そのため、SQL Injection Cheat Sheat[6]を用いて SQL インジェクションの他の攻撃のパターンを手動で 130 件生成し合計で 1334 件のデータとする。これらのデータの 6 割を学習用のデータとして利用し 4 割のデータを攻撃検知の正答率及びモデルの評価に使う。

さらに、データ数を 1334 件だけでなく半分のデータ量である 667 件と 2 倍のデータ量である 2668 件として同様な実験を行う。

(3) 実験結果

図 3 は、データ数と処理時間の関係示したものである。データ数が増えるにつれて提案手法と既存手法の処理時間の差が大きくなる。図 3 の右側のラベルにある既存手法の累計処理時間の推移から分かるようにデータ数が増えるほど処理時間は非線形的に増加している。既存手法と提案手法の処理時間の比率は、667 件では 1. 9 倍、1334 件では 3. 6 倍、2668 件では 4. 3 倍になる。

図 4 は、データ数と正答率の関係示したものである。提案手法の正答率はほぼ横ばいに対して、既存手法は、データ数が半減すると 86. 4%に減少している。

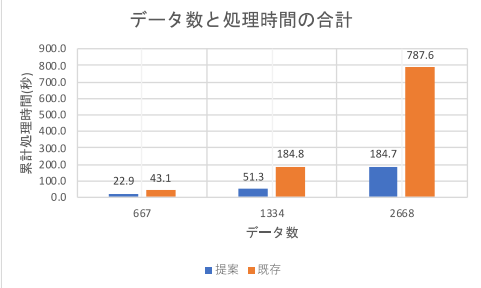

図 3. データ数と処理時間の関係

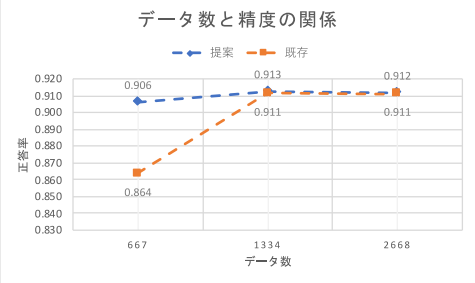

図 4.　データ数と正答数の関係

3.3.考察

上記の結果から辞書を圧縮しても正答率は損なわれず短い処理時間で学習することが可能である。特に、データ数が増えるほど処理時間の差分が顕著に現れる。正答率が 100%にならなかった原因は、今回使用した SQL インジェクションに占める自作の比率が 9.7%と低く全てのデータが異常と学習できなかったと考えられる。

オリジナルの辞書はデータ数が少ないほど正答率への影響が顕著であるが、データが増えるほど辞書に含まれる単語数が増えるため影響が小さくなり正答率の差分は小さくなる。

4. 結論および今後の課題

本研究では、オリジナルの辞書を作成し学習させることで正答率を維持した状態で処理速度を大幅に速めることができる手法を提案した。データ数を増やした際に処理速度も比例して増加するが、提案手法ではより高速に処理が実現できる。課題の１つとして、用意したデータセットによって結果が大きく変わってしまうのが現状である。今後は、他のデータセットでも確認したい。

5. 参考文献

[1] Romil Rawat, SQL injection attack Detection using SVM, International Journal of Computer Applications, No.13, pp.1-4, 2012.

[2] 佐野綾子ら,文字単位特徴量による SQL 攻撃手法の検知手法,情報処理学会研究報告, No.16, pp.1-6, 2015.

[3]合路健人, SQL インジェクション攻撃に含まれる記号の出現頻度とその関連性による攻撃検出手法の提案, 高知工科大学 学士学位論文,pp.1-41, 2017.

[4] Damn Vulnerable Web Application (2019/1/10)
http://www.dvwa.co.uk/

[5] SQLMAP(2019/1/10)
http://sqlmap.org/

[6] SQL Injection Cheat Sheat (2019/1/10)
http://www.byakuya-shobo.co.jp/hj/moh/sqlinjectioncheatsheet.html

Copyright ©2019 Information Processing Society of Japan.
All Rights Reserved.