3K-08

クロスサイトスクリプティング(XSS)攻撃における バッチ学習とオンライン学習の比較実験

梅原 章宏[1] 松田 健[2] 園田 道夫[1] 水野 信也[2] 趙 晋輝[3]

中央大学大学院理工学研究科情報工学専攻[1]

静岡理工科大学総合情報学部コンピュータシステム学科[2]

中央大学理工学部情報工学科[3]

1. 背景

近年インターネットの普及に伴い,Web アプリケーションに対するサイバー攻撃も増加している.その中でもクロスサイトスクリプティング(XSS)攻撃は HTML の入力部分などの脆弱性に対するサイバー攻撃である.従来の対策として構文解析によるフィルタが提案,実装されている[1]が,XSS 攻撃に用いられる入力は正常な入力との区別が容易でなく,機械的な攻撃検出が困難である.

先行研究[2]では,入力中の ASCII 文字に着目した特徴抽出を行い,生成した特徴ベクトルに機械学習アルゴリズムを適用することで,攻撃検知を試みた.その結果,ある程度の攻撃検知が可能であった.しかし,用いる機械学習アルゴリズムによって,学習に対する未知の形式の入力を与えた場合に,検知結果に異なる影響を与えることが考えられた.

そこで,本研究では学習データに対する新たな形式の入力を与えた際に,バッチ学習とオンライン学習の各機械学習アルゴリズムでどのように検知結果が変化するかを調べた.また,それぞれの結果を比較することで,各機械学習アルゴリズムが入力に対してどのようにふるまうのかを考察した.

2. XSS 攻撃

XSS 攻撃による主な被害として,Cookie 値を盗まれることによる成りすまし被害などがあげられる.例としては,会員制サイトのマイページに脆弱性が存在する場合,セッション ID を奪われることにより不正ログインの被害にあうことが考えられる.

既存の検知手法として特定の入力に対して処理を行うブラックリスト,ホワイトリスト方式や,HTML における特殊記号を別の記号に置き換えるエスケープ処理がある.しかし,ブラックリスト,ホワイトリスト方式はあらかじめ処理を行う入力を指定する必要があるため,未知の入力への対応が難しい.またエスケープ処理も,処理を行う記述を忘れてしまった場合に,脆弱性が生じてしまう危険性が存在する.

Comparative Experiments of Batch Learning and Online Learning in Cross-Site Scripting Attacks

Akihiro Umehara[1], Takeshi Matsuda[2], Michio Sonoda[1], Shinya Mizuno[2], Jinhui Chao[3]

1. Departments of Information and System Engineering, Graduate School of Science and Engineering, Chuo University.

2. Department of Computer Science, Faculty of Comprehensive informatics, Shizuoka Institute of Science and Technology.

3. Departments of Information and System Engineering, Faculty of Science and Engineering, Chuo University.

3. 機械学習アルゴリズム

3.1 バッチ学習とオンライン学習

機械学習アルゴリズムの学習方式として,バッチ学習とオンライン学習の 2 つがあげられる.

バッチ式学習は,1 回の学習の際にすべての学習データを用いて学習を行う方式である.そのため,学習後の再学習を行うためには,もう一度学習データをすべて与える必要がある.

オンライン学習は 1 回の学習の際に 1 つの学習データを用いて学習を行う方式である.そのため,テスト中に誤分類を行った場合は,そのデータを用いて再学習を行うことが可能となっている.

3.2 SVM

SVM はバッチ式機械学習アルゴリズムの一つであり,分類は以下の式で定義される.

$$\mathrm{y}(\mathbf{x}) = \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}) + b = \sum_{n=1}^{N} a_n t_n k(\boldsymbol{x}, \boldsymbol{x}_n) + b$$

ここで $\mathbf{x}$ は入力ベクトル,$\mathbf{w}$ は重みベクトル,ϕ は特徴空間変換関数,a_n はラグランジュ乗数,t_n は目標値,$\mathrm{k}(\mathbf{x}, \boldsymbol{x}_n)$ はカーネル関数,b はバイアスパラメータである.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,汎化誤差が最も小さくなるような分類境界を求める.マージンを最適化する解は,以下の目的関数を最小化することで得られる.

$$\mathrm{C}\sum_{n=1}^{N} \xi_n + \frac{1}{2}\|\boldsymbol{w}\|^2$$

$$\mathrm{s.t}\ \ t_n\big(y(\boldsymbol{x}_n)\big) \geq 1 - \xi_n \quad \mathrm{n} = 1, \ldots, \mathrm{N}$$

$$\xi_n \geq 0$$

ここで C は五分類に対するペナルティの大きさを制御するパラメータであり,大きいほど誤分類を許さないような分類境界を求める.また ξ_n はスラック変数であり,$0 \leq \xi_n \leq 1$ となるデータは正しく分類され,$\xi_n > 1$ となるデータは誤分類されている.

3.3 Random Forest

Random Forest は複数の決定木による集団学習を行うバッチ式機械学習アルゴリズムの一つである.全ての学習データからランダムに抽出した要素を用いて決定木を作成し,それらの分類の結果から最終的な出力を決定する.

3.4 SCW [3]

SCW はオンライン式の機械学習アルゴリズムの一つであり,分類は以下の式で定義される.

$$y(\mathbf{x}) = \mathrm{sgn}(\boldsymbol{\mu}_{t-1}\boldsymbol{x}_t)$$
$$\text{if} \quad \boldsymbol{\mu}\mathbf{x} \geq 0 : y(\mathbf{x}) = 1$$
$$\text{else:} \quad y(\mathbf{x}) = -1$$

ここで $\mathbf{x}$ は入力ベクトル, $\boldsymbol{\mu}$ は重みの平均ベクトルであり, バイアスパラメータは存在しない. また, 損失関数 l^{ϕ} は以下の式であらわされる.

$$l^{\phi} = \max(0, \phi\sqrt{\boldsymbol{x}_t^T \boldsymbol{\Sigma} \boldsymbol{x}_t} - y_t \boldsymbol{\mu} \boldsymbol{x}_t)$$

ここで $\boldsymbol{\Sigma}$ は共分散行列, $\phi = \Phi^{-1}(\eta)$ である(Φ は正規分布の累積密度関数, η は誤差を許容するパラメータである).

$\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ の更新式は以下の最適化問題であらわされる.

$$(\boldsymbol{\mu}_{t+1}, \boldsymbol{\Sigma}_{t+1}) = \arg\min_{\boldsymbol{\mu},\boldsymbol{\Sigma}} D_{KL}(\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma}) \,||\mathcal{N}(\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t))$$
$$+ C l^{\phi}(\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma}) : (\boldsymbol{x}_t, y_t))$$

ここで, D_{KL} はカルバック情報量, $\mathcal{N}$ は平均ベクトル $\boldsymbol{\mu}$ と共分散行列 $\boldsymbol{\Sigma}$ の多変量正規分布, C は重みの更新を制御するパラメータである.

SCW は同じオンラインの機械学習アルゴリズムである CW[4]を改良したものであり, 特徴としてデータの信頼度による重み付けを行う. また, SVM のようなマージンの最大化を行うことで, CW の弱点であるノイズに弱い面を克服している.

4. 実験準備

4.1 実験用データの生成

実験を行うにあたり, URL 形式の攻撃入力, スクリプト形式の攻撃入力, URL 形式の正常入力, TeX で用いられる数式形式の正常入力の 4 パターンの入力を収集した. これらの入力からそれぞれ 200 個を抽出し, 攻撃と正常それぞれを混ぜ合わせることで, 一つあたり攻撃入力 200 個, 正常入力 200 個のデータセットを 4 つ生成した.

TeX 形式の数式を正常入力として用いた理由としては, 数式を入力する Web ページのフォームは特殊なものでなく, また数式には不等号が含まれることがあり, HTML のタグと区別することが困難な場合があるためである. また TeX は様々な記号を使用するため, 使用する記号を制限すると誤検知を導く可能性がある.

4.2 特徴抽出

生成した各データセット内の各入力に対して特徴抽出を行い, 128 次元の特徴ベクトルに変換を行った. 機械学習アルゴリズムによる学習, テストにはこの特徴ベクトルを入力として用いた. 特徴抽出は, 入力内の各 ASCII 文字の出現頻度を特徴量として行った.

4.3 評価

検知実験の評価を行うにあたって, 特徴抽出を行った各データセットに対し, 5 分割交差確認を行った. SVM, Random Forest の評価項目として, 以下を用いた.

1. データ全体に対して予測が正しかったものの割合(正解率)
2. 予測が実際に正しいものの割合(精度)
3. 真の結果に対して, その結果であると予測されたもの割合(再現率)
4. 精度と再現率の調和平均(F 値)
5. ROC 曲線の下の面積(AUC)

SCW の評価項目としては, 上記の 1 から 4 を用いた.

5. 結果・考察

今回の実験では学習データ 4 パターンに対しテストデータ 4 パターンの計 16 パターンの実験を行った. その中から, 正常 URL, 攻撃 URL で学習した時に, テストデータとして正常 URL, 攻撃 URL…(1)と正常 URL, 攻撃スクリプト…(2)の 2 パターンを与えた場合の結果を, 表 1 に示す.

表 1 正常 URL-攻撃 URL で学習した時の結果

	(1)			(2)		
	SVM	RF	SCW	SVM	RF	SCW
正解率	97.5	97.5	85.3	82.5	81.0	85.0
攻撃精度	96.5	97.9	95.6	97.2	97.4	91.2
正常精度	98.4	97.0	75.1	75.0	73.7	79.0
攻撃再現率	98.6	97.1	79.2	67.5	64.7	81.5
正常再現率	96.6	98.1	94.5	98.0	98.1	89.7
攻撃 F 値	97.5	97.5	86.5	79.4	77.0	85.9
正常 F 値	97.5	97.5	83.5	84.8	83.8	83.8
AUC	97.6	97.6		82.8	81.4	

単位:%

表 1 の正解率を見ると, SVM, Random Forest は学習データとテストデータにおいて攻撃入力の形式が変化した場合, 正解率が 10%程度下落している. その一方で, SCW の正解率はほとんど変化していない. この結果から, SVM, Random Forest といったバッチ式学習では, 未知の形式の攻撃入力が与えられた際に対応が難しいが, SCW のようなオンライン式学習の場合, 再学習を行うことでこのような未知形式の攻撃に対しても, ある程度の対応が可能であることが考えられる.

他の各学習データとテストデータの組み合わせによる結果についても, SVM, Random Forest は未知の形式のテストデータを与えた場合に, 結果が大きく変動する傾向が見られたが, SCW では結果の変動が少ない傾向となった. このことからも, 学習データに対する全く未知の形式をしたテストデータに対しても, オンライン学習アルゴリズムであれば柔軟な対応が可能となる可能性が考えられる.

しかし, SCW の正解率などの実数値が, SVM や Random Forest と比較して低い値となっていることが多い. SCW に未知形式のデータを与えた場合の結果のばらつきは少ないものの, 実数値で見ると SVM や Random Forest と同等かそれより低くなっているパターンも見受けられた. そのため, SCW の検知性能事態の向上は, 今後解決していくべき課題であると考える.

参考文献

[1] "IE8 Security Part IV: The XSS Filter – IEBlog – SiteHome – MSDN Blogs", http://blogs.msdn.com/b/ie/archie/2008/07/02/ie8-security-part-iv-the-xss-filter.aspx, 最終閲覧日:2015/1/7

[2] 梅原 章宏, 松田 健, 園田 道夫, 水野 信也, 趙 晋輝, "機械学習を用いたクロスサイトスクリプティング(XSS)攻撃の検知に関する考察", 情報処理学会第 71 回 CSEC 研究会研究報告

[3] Jialei Wang, Peilin Zhao, Steven C.H. Hoi, "Exact Soft Confidence-Weighted Learning", ICML(2012)

[4] Mark Dredze,Koby Crammer, Fernando Pereira, "Confidence-weighted Linear Classification", ICML, pp.264-pp.271 (2008)