Vol.2012-MPS-87 No.14
2012/3/2

SQLインジェクション攻撃自動検出支援モデルと予測誤差

松　田　　健[†1]

SQL インジェクション攻撃を代表とする Web アプリケーション攻撃の増加に伴い，パターン認識やブラックリストなどを用いてこれらの攻撃を自動検出する手法が開発されている．しかしながら，リストが肥大化により検出コストが増加するため，多様化する Web アプリケーション攻撃への対処がますます困難になってきている．この問題に対し，本研究では SQL インジェクション攻撃の文字列に含まれる文字に着目した攻撃検出法を提案した．また提案モデルの誤検出の割合いを測るための予測誤差を定義し，2 値判別の問題で広く利用されているシグモイド関数を用いた方法と提案法との予測誤差を比較し，提案法の方がシグモイド関数を用いた方法より予測誤差が小さくなることを示した．

Automatic Detection Model of SQL Injection Attacks and Prediction Error

Takeshi Matsuda[†1]

The method of web application attack has been becoming diversified, and the detection methods, such as pattern matching processing and black list, has been developed. However, it becomes increasingly difficult to detect attacks because new attacks are developed and the black list becomes enlarged.

In this paper, we proposed a detection model of SQL injection attacks, and defined prediction error to measure the performance of the detection models. Then we compared our proposed model with the detection model applying sigmoid function, and showed that the prediction error of our proposed model is less than the sigmoid model.

†1
サイバー大学　IT 総合学部

1. はじめに

SQL インジェクション攻撃は Web ページのフォームを介して行われる比較的単純な攻撃である．攻撃が成功すると Web アプリケーションのデータベースへの不正なアクセスが可能となり，データベースの情報を閲覧したり改ざんしたりすることができる．最近では，自動拡散するワームやボットネットを利用した攻撃が観測されるようになり[2]，SQL インジェクション攻撃から Web アプリケーションを防御する技術的手段として，過去に観測された SQL インジェクション攻撃に対してパターン認識や構文解析を応用した方法や，Web アプリケーションファイアウォール (WAF) と呼ばれる攻撃パターンをブラックリスト化して入力された文字列とリストの照合を行う手法を用いた攻撃検出法が開発・実用化されている[4)–12)]．しかしながら，これらの自動検出手法では新しく開発された攻撃を検出することが出来なくなったり，ブラックリストを更新することで検出コストが増加したりするため，多様化する Web アプリケーション攻撃への対処がますます困難になってきている．この問題に対し，我々は SQL インジェクション攻撃に含まれる記号や数などの文字を利用した攻撃検出を行うアルゴリズムを提案している[13]．本研究では，その方法を応用した SQL インジェクション攻撃を検出するモデル[14] を提案し，2 値判別の問題で広く利用されているシグモイド関数[15)–17)] を応用して SQL インジェクション攻撃を検出するモデルの定義を行った．その上で，SQL インジェクション攻撃を検出するモデルの誤検出の度合いを評価するための予測誤差を定義し，提案モデルではシグモイド関数を応用したモデルより予測誤差を小さできることを示した．本論文の構成は以下の通りである．2 章では SQL インジェクション攻撃の実例を用いて我々の提案アルゴリズムの紹介を行う．3 章では SQL インジェクション攻撃検出のための提案モデルとシグモイド関数を応用したモデルを定義し，4 章で定理の証明を行う．5 章で提案法に関する考察をして，最後の 6 章でまとめを行い，今後の課題を示す．

2. SQL インジェクション攻撃とその検出アルゴリズム

この章では，[13] で提案した SQL インジェクション攻撃検出のアルゴリズムについて紹介する．これ以降，Web ページのフォームに入力された文字列（以下，入力文字列という）が SQL インジェクション攻撃である場合それを攻撃文字列と呼び，SQL インジェクション攻撃でない場合は正常文字列と呼ぶことにする．フォームから入力される入力文字列を l，記号・数字・アルファベットなどの文字を s_a $(a = \mathrm{i}, \mathrm{ii}, \mathrm{iii}, \cdots)$ とする．文献[13] では，表 1 に

示す 20 個の文字を用意して SQL インジェクション攻撃の特徴を捉える文字の選定を行った．用意した 20 個の文字は SQL 文によく利用記号を中心として構成した．

SQL インジェクション攻撃の攻撃特徴となる文字の選定方法は以下の通りである．入力文字列 l に対して，

$$x_{l,a} = \frac{\#s_a}{|l|}$$

を定義し，0 から 1 の有理数に値をとる $x_{l,a}$ を入力として扱う．ここで，$|l|$ は入力文字列長を，$\#s_a$ は入力文字列に含まれる文字 s_a の総数を表すものとする．SQL インジェクション攻撃の検出では，入力文字列が攻撃文字列であるか正常文字列であるかを判別する問題を考える．そのために次のような判別関数を定義する．

$$h(x_{l,a}) = \begin{cases} 1 & (x_{l,a} \geq \alpha) \\ 0 & (x_{l,a} < \alpha) \end{cases}$$

ここで，α は 0 から 1 の値をとる実数で，入力文字列 l が攻撃文字列であるか正常文字列であるかを判別するための閾値である．本論文では $h(x_{l,a}) = 1$ なら l を攻撃文字列と，$h(x_{l,a}) = 0$ なら l を正常文字列と判別することにする．以下に，例 1 で紹介した例に対する判別関数の計算例を示す．

例 1　入力文字列 l_1 を
SELECT * FROM CardInfo WHERE ID= ’name ’ and PASS= ’777 ’ OR ‘ 7=7 ’’
とする．この入力文字列では最後の部分の恒等式 7=7 という常に真となる命題があるため，入力時に無害化や入力制限などの処理が適切にされない場合，データベースは攻撃者に対してすべての情報へのアクセスを許してしまう．この攻撃に対して，表 1 にある文字の s_{i}（半角スペース），s_{ii}（セミコロン），s_{iii}（シングルクォーテーション）を用いて判別関数の値を計算する．

$$x_{l_1,\mathrm{i}} = \frac{9}{62} = 0.15$$
$$x_{l_1,\mathrm{ii}} = \frac{0}{62} = 0$$
$$x_{l_1,\mathrm{iii}} = \frac{4}{62} = 0.06$$

となる．計算結果は小数点第 3 位で四捨五入を行ったものであり，以下同様の計算を行うものとする．この場合，閾値を $\alpha = 0.06$ ととれば s_{i}（半角スペース）または s_{iii}（シングルクォーテーション）であれば攻撃の検出が可能であるが，s_{ii}（セミコロン）を用いた場合は攻撃検出ができない．また，次のような電話番号を想定した入力文字列 l_2

01 2345 6789

が入力された場合，s_{i}（半角スペース）を用いて攻撃検出を行うと，

$$x_{l_2,\mathrm{i}} = \frac{2}{12} = 0.17$$

となり，閾値を $\alpha = 0.06$ とした場合，判別関数は l_2 を攻撃文字列であると誤判別することになる．

表 1　攻撃検出率 P_A と正常誤検出率 P_N

攻撃特徴文字候補	P_A	P_N
s_{i}（スペース）	0.971	0.098
s_{ii}（セミコロン）	0.664	0.000
s_{iii}（シングルクォーテーション）	0.483	0.000
s_{iv}（左側丸括弧）	0.459	0.060
s_{v}（右側丸括弧）	0.414	0.090
s_{vi}（左側中括弧）	0.000	0.060
s_{vii}（右側中括弧）	0.000	0.026
s_{viii}（左側大括弧）	0.000	0.026
s_{ix}（右側代括弧）	0.000	0.026
s_{x}（シャープ）	0.032	0.021
s_{xi}（パーセント）	0.024	0.026
s_{xii}（ダブルクォーテーション）	0.010	0.000
s_{xiii}（アンパサンド）	0.019	0.021
s_{xiv}（バックスラッシュ）	0.032	0.000
s_{xv}（パイプ）	0.040	0.103
s_{xvi}（等号）	0.294	0.060
s_{xvii}（大なり記号）	0.000	0.090
s_{xviii}（小なり記号）	0.000	0.038
s_{xix}（アスタリスク）	0.128	0.094
s_{xx}（スラッシュ）	0.112	0.073

表 1 は，[20)–23)] などから収集した 624 個の攻撃文字列サンプルとフォームに入力される文字を想定して作成した 234 個の正常文字列サンプルを用いて行い，攻撃文字列を攻撃であると判別した割合（以下，攻撃検出率 P_A という）と正常文字列を攻撃であると判別した割合（以下，正常誤検出率 P_N という）を計算してまとめたものである．

表 1 の結果から，攻撃特徴となる文字を 1 文字だけ用いて攻撃検出を行ったときの攻撃

検出率は，半角スペース以外を用いた場合それほど高くならないことがわかる．また半角スペースの攻撃検出率は非常に高い一方，正常誤検出率についても他の文字と比較すると高くなっている．そこで入力文字列 l に対する $x_{l,a}$ の計算式を次のように変更して攻撃検出を行う．

$$x_{l,k} = \frac{\#S_k}{|l|}$$

ここで S_k は攻撃特徴となる複数の文字からなる集合であり，$\#S_k$ を入力文字列 l 中の S_k に属する文字の総数と定義する．いま，$S_1 = \{s_{\mathrm{i}}, s_{\mathrm{ii}}, s_{\mathrm{iii}}\}$（半角スペース，セミコロン，シングルクォーテーション）と定義すると例 3 の入力文字列 l_1 に対して，

$$x_{l_1,1} = \frac{9+0+4}{62} = \frac{13}{62} = 0.21$$

となる．この場合，閾値を $0 \leq \alpha < 0.21$ とすれば入力 l_1 を攻撃であると判別することができる．さらに例 3 の電話番号を想定した入力文字列 l_2 に対しても，

$$x_{l_2,1} = \frac{2}{12} = 0.17 < 0.21$$

となることから，l_2 は攻撃でないと判別されることがわかる．以上の考察により，表 1 にある攻撃検出率 P_A が高い文字を組み合わせて攻撃を検出することで攻撃検出率 P_A が高く，かつ正常誤検出率 P_N を低くするような攻撃検出が出来るものと期待される．論文[13] では，文字集合として $S = \{s_{\mathrm{i}}, s_{\mathrm{ii}}, s_{\mathrm{iv}}\}$（半角スペース，セミコロン，右側丸括弧）を用いて攻撃検出した場合，閾値 $\alpha = 0.08$ で攻撃検出率 P_A と正常検出率 $1 - P_N$ の加重平均 μ に対して，$\mu = \beta P_A + (1-\beta)(1-P_N) > 0.9$ が $b = 0, 0.1, 0.2, \cdots, 1.0$ で成り立つことを実験によって確認した．実験に使用したサンプルは[20)–23)] から収集したものである．

3. 提案モデル

本章では，SQL インジェクション攻撃を自動検出するためのモデルを提案する．[13] の実験では，入力文字列 l に対する $x_{l,k}$（以下，x とする）の値が 0.1 程度の小さな値で l が攻撃文字列である可能性が高くなるという結果を得ている．これを踏まえ，本論文ではベルヌーイ分布を基にした SQL インジェクション攻撃を自動検出するためのモデルとして

$$p_1(y|x,\theta) = \sum_{i=1}^{I} a_i (x^{b_i})^y (1 - x^{b_i})^{1-y}$$

を定義する．ただし，$i = 1, 2, \cdots, I$ は自然数であり，$\theta = \{(a_i, b_i)\}_{i=1}^{I}$，すべての i に対して $a_i \geq 0$ かつ $\sum_{i=1}^{I} a_i = 1$ である．さらに，シグモイド関数

$$g(x) = \frac{x^b}{x^b + (1-x)^b}$$

を用いて SQL インジェクション攻撃を検出するモデルを

$$p_2(y|x,\theta) = \sum_{i=1}^{I} a_i \left(\frac{x^{b_i}}{x^{b_i} + (1-x)^{b_i}}\right)^y \left(1 - \frac{x^{b_i}}{x^{b_i} + (1-x)^{b_i}}\right)^{1-y}$$

と定義する．一般に，シグモイド関数は $s(t) = \frac{1}{1+e^{-bt}}$ のように定義されるが，関数

$$t = \begin{cases} -\infty & \Big(x = 0\Big) \\ \log \frac{x}{1-x} & \Big(x \in (0,1)\Big) \\ \infty & \Big(x = 1\Big) \end{cases}$$

を用い，定義域と値域がそれぞれ $[0,1]$ となるようにシグモイド関数を再定義した．我々の目標は前章で定義したとおり攻撃検出率を高く，正常誤検出率を低くすることである．そこで，

$$e(p) = \int_{\{x|h(x)=1\}} p(x, y=0)dx + \int_{\{x|h(x)=0\}} p(x, y=1)dx$$

を考え，$e(p)$ をモデル $p(y|x,\theta)$ の予測誤差として定義する．また $p(x)$ を以下の性質を満足する関数と定義する．$\{x_0, x_1, x_2, \cdots, x_n\}$ を実数の閉区間 $[0, \frac{1}{2}]$ の $x_0 \leq x_1 \leq \cdots \leq x_n$ を満足する実数列とする．$x \in [0, \frac{1}{2}]$ に対して，

$$p(x) = \sum_{j=0}^{n-1} \tau_j \mathbf{1}(x)_{(x_j, x_{j+1})}$$

$$p(1-x) = \sum_{j=0}^{n-1} \tau'_j \mathbf{1}(x)_{(1-x_{j+1}, 1-x_j)}$$

かつ

$$p(x) \leq p(1-x)$$

が成り立つと仮定する．ただし，$j = 0, 1, \cdots, n-1$ であり

$$\mathbf{1}(x)_{(x_j,x_{j+1})} = \begin{cases} 1, & x \in (x_j, x_{j+1}) \\ 0, & x \text{ それ以外} \end{cases}$$

である．このとき，本論文で提案するモデル $p_1(y|x,\theta)$ とシグモイド関数を応用したモデル $p_2(y|x,\theta)$ との間に次の関係が成り立つ．

定理 1 次の 2 つの条件
(1) $0 < b < 1$
(2) $0 \leq \alpha \leq \frac{1}{2}$
が成り立つとき，
$e(p_2) > e(p_1)$
である．

4. 証　　明

この章では，4.1 で定理の証明に必要な補題について整理を行い，4.2 で前章で紹介した定理 1 の証明を行う

4.1 補　　題

補題 1 すべての $i = 1, 2, \cdots, I$ に対して，$0 \leq b_i \leq 1$ であるとき，

$$f_i(x) = x^{b_i} - \left(\frac{x^{b_i}}{x^{b_i} + (1-x)^{b_i}}\right)$$

とすると，$f_i(x) \geq 0$ である．

[補題 1 の証明] i を固定して考える．

$$f_i(x) = \frac{x^{b_i}}{x^{b_i} + (1-x)^{b_i}}\Big(x^{b_i} + (1-x)^{b_i} - 1\Big)$$

であり $0 \leq x \leq 1$，$0 \leq b_i \leq 1$ で

$$0 \leq \frac{x^{b_i}}{x^{b_i} + (1-x)^{b_i}} \leq 1, \quad (x^{b_i} + (1-x)^{b_i} - 1) \geq 0$$

が成り立つ．したがって，すべての $i = 1, 2, \cdots, I$ に対して $f_i(x) \geq 0$ である．(証明終わり)

補題 2 $0 \leq \epsilon \leq \frac{1}{2}$ を満足する実数 ϵ に対して，

$$f(1-\epsilon) \geq f(\epsilon)$$

が成り立つ．

[補題 2 の証明]

$$f(\epsilon) = \epsilon^{b_i}\Big(1 - \frac{1}{\epsilon^{b_i} + (1-\epsilon)^{b_i}}\Big)$$
$$f(1-\epsilon) = (1-\epsilon)^{b_i}\Big(1 - \frac{1}{\epsilon^{b_i} + (1-\epsilon)^{b_i}}\Big)$$

より，

$$f(1-\epsilon) - f(\epsilon) = \Big\{(1-\epsilon)^{b_i} - \epsilon^{b_i}\Big\}\Big(1 - \frac{1}{\epsilon^{b_i} + (1-\epsilon)^{b_i}}\Big)$$

である．$0 \leq \epsilon \leq \frac{1}{2}$ より $(1-\epsilon)^{b_i} - \epsilon^{b_i} \geq 0$，また補題 1 より $\epsilon^{b_i} + (1-\epsilon)^{b_i} - 1 \geq 0$ であるから，

$$f(1-\epsilon) - f(\epsilon) \geq 0$$

が成り立つ．等号は $\epsilon = \frac{1}{2}$ のときに成立する．(証明終わり)

4.2 定理 1 の証明

$0 \leq b \leq 1$ である b を固定して考える．

$$e(p_2) - e(p_1) = \Big\{\int_{\{x|h(x)=1\}} p_2(y=0|x)p(x)dx + \int_{\{x|h(x)=0\}} p_2(y=1|x)p(x)dx\Big\}$$
$$- \Big\{\int_{\{x|h(x)=1\}} p_1(y=0|x)p(x)dx + \int_{\{x|h(x)=0\}} p_1(y=1|x)p(x)dx\Big\}$$

であり，$i = 1, 2, \cdots, I$ に対して

$$f_i(x) = x^{b_i} - \frac{x^{b_i}}{x^{b_i} + (1-x)^{b_i}}$$

とおくと

$$e(p_2) - e(p_1) = \sum_{i=1}^{I} a_i\Big(\int_{\{x|h(x)=1\}} f_i(x)p(x)dx - \int_{\{x|h(x)=0\}} f_i(x)p(x)dx\Big)$$

と書くことができる．$e(p_2) > e(p_1)$ を示すには，$\alpha = \frac{1}{2}$ の場合の場合を考えれば十分である．$\{x_0, x_1, \cdots, x_n\}$ を $x_0 \leq x_1 \leq \cdots \leq x_n$ を満たす閉区間 $[0, \frac{1}{2}]$ の実数列において，
$\triangle_j = x_{j+1} - x_j$
$\triangle'_j = (1-x_j) - (1-x_{j+1})$
と定義する．このとき，$\triangle_j = \triangle'_j$ である．$f_i(x)$ は閉区間 $[0,1]$ 上の連続関数であり，

$$\int_0^{\frac{1}{2}} f_i(x)p(x)dx = \lim_{n\to\infty}\sum_{j=0}^{n-1} f_i(\xi)p(\xi)\triangle_j,$$

$$\int_{\frac{1}{2}}^{1} f_i(x)p(x)dx = \lim_{n\to\infty}\sum_{j=0}^{n-1} f_i(\xi')p(\xi')\triangle'_j,$$

と書ける．ただし，$\xi \in [x_j, x_{j+1}]$ ，$\xi' = 1-\xi$ である．さらに補題 2 と定理の仮定から，すべての $i = 1, 2, \cdots, I$ に対して

$$f_i(\xi')p(\xi')\triangle'_i > f_i(\xi)p(\xi)\triangle_i$$

が成り立つ．したがって，$\alpha = \frac{1}{2}$ のとき

$$\int_{\{x|h(x)=0\}} f_i(x)p(x)dx < \int_{\{x|h(x)=1\}} f_i(x)p(x)dx$$

であり，$0 \le \alpha \le \frac{1}{2}$ のとき $e(p_2) - e(p_1) > 0$ である．（証明終わり）

5. 考　　察

本論文では，SQL インジェクション攻撃を検出するためのモデル

$$p_1(y|x,\theta) = \sum_{i=1}^{I} a_i (x^{b_i})^y (1-x^{b_i})^{1-y}$$

を提案した．本研究で使用した正常文字列のサンプルは，Web 上のフォームに入力される通常の入力を文字列として

- ID やパスワード，電話番号，日本語も想定した氏名や住所
- ブログで頻繁に使用される Wiki の文法
- ウェブページで使用される顔文字

などに注意してサンプルの生成を行った．提案モデルはベルヌーイモデルの混合モデルであるが，これは正常文字列のサンプルの上述の傾向に配慮して仮定したものである．以下，簡単のため $I = 1, b_1 = b$ として考察を行う．定理 1 で提案モデルの予測誤差がシグモイド関数を応用したモデルの予測誤差より小さくなることを理論的に示したが，実際にどの程度予測誤差が小さくなるかということを実験で示す．図**??**は以下の条件のもとで提案モデルとシグモイド関数を応用した場合のモデルの予測誤差の比較を行ったものである．

条件

- $b = 0.05, 0.1, 0.15$ の 3 つの場合を考える．
- 閾値 α を 0.05 から 0.15 まで 0.01 ずつ増加させた場合で予測誤差の計算を行う．

積分の計算にはニュートン・コーツ法を用い，ステップ長を 0.01 として計算した．図 1 の結果が示すとおり，$b = 0.15$ の場合では，提案モデルの予測誤差はシグモイド関数を応用したモデルの予測誤差より 3.5 以上小さくなることを確認することができた．

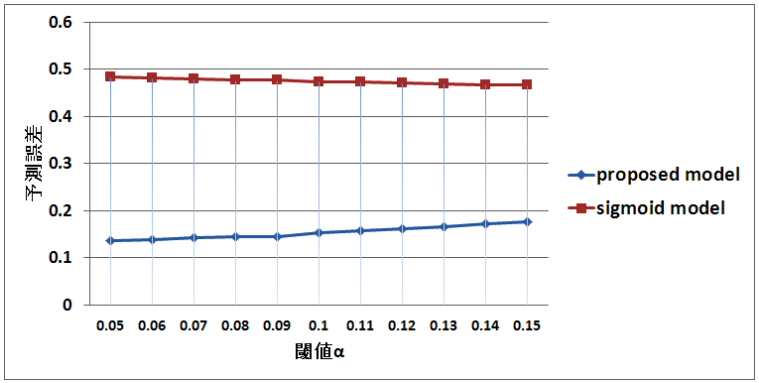

図 1　予測誤差の比較 ($b = 0.15$)

6. ま と め

本論文では，攻撃文字列に含まれる文字を攻撃特徴として SQL インジェクション攻撃を自動検出するモデルを提案し，シグモイド関数を応用したモデルとの予測誤差を比較することでその有効性を示した．本研究で使用したデータは文献[20)–23)] から収集したデータをもとに作成した人工サンプルである．実データの収集を行い，提案モデルの有用性について検証することが今後の課題である．また，提案モデルのパラメータを最適化する方法や他の Web アプリケーション攻撃への適用について検討を行うことも今後の課題である．

参　考　文　献

1) *NT Web Technology Vulnerabilities*, http://www.phrack.com/issues.html?issue=54.
2) 独立行政法人 情報処理推進機構: **ソフトウェア等の脆弱性関連情報に関する届け出状況** *(2010* **年第四半期***)*, http://www.ipa.go.jp/security/vuln/report/vuln2010q4.html.

3) 小菅 祐史, 花岡 美幸, 河野 健二,: *SQL* **インジェクション攻撃の脆弱性の効果的な自動検出手法**, 情報処理学会研究報告 CSEC[コンピュータセキュリティ], (45), pp.103-108, (2008).
4) William Robertson, Giovanni Vigna, Christopher Kruegel, Richard A. Kemmerer,: *Using Generalization and Characterization Techniques in the Anomaly-based Detection of Web Attacks*, In Proceeding of the Network and Distributed System Security (NDSS) Symposium, San Diego, CA, February, (2006).
5) Fredrik Valeur, Darren Mutz, Giovanni Vigna,: *A Learning-Based Approach to the Detection of SQL Attacks*, http://www.cs.ucsb.edu/?vigna/publications/2005 valeur mutz vigna dimva05.pdf, (2005).
6) Engin Kirda, Christopher Kruegel, Giovanni Vigna, Hennaed Jovanovic,: *Noxes:A Client-Side Solution for Mitigating Cross Site Scripting Attacks*, Security Track of the 21st ACM Symposium on Applied Computing (SAC 2006), Dijon, France, April, (2006).
7) Frank S. Rietta,: *Application layer intrusion detection for SQL injection*, ACMSE 44 Proceedings of the 44th annual Southeast regional conference, (2006).
8) Y. Kosuga, M. Hanaoka and K. Kono,: *Effective Automated Testing for Detecting SQL Injection vulnerabilities*, The Special Interest Group Notes of IPSJ, CSEC, pp.103-108, (2008).
9) Yingbo Song, Angelos D. Keromytis, Salvatore J. Stolfo, Spectrogram,: *A Mixture-of-Markov-Chains Model for Anomaly Detection in Web Traffic*, Proceedings of the 16th Annual Network and Distributed System Security Symposium (NDSS), (2009)
10) Davide Ariu, Giorgio Giacinto,: *HMMPayl: an application of HMM to the analysis of the HTTP Payload*, JMLR: Workshop and Conference Proceedings 11, pp.81-87, (2010).
11) A. Tajpour, M. Masrom, M.Z. Heydari and S. Ibrahim,: *SQL injection detection and prevention tools assessment*, 2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT 2010), pp.518-522, (2010)
12) 伊波靖, 高良富夫,: *SVM* **を用いた** *WAF* **の検知機能の提案**, 情報処理学会全国大会講演論文集, pp.445-447, (2011)
13) M. Sonoda, T. Matsuda, D. Koizumi and S. Hirasawa,: *On the Automatic Detection of the SQL Injection Attacks by the Feature Extraction of the Single Character*, Proceeding of 4th International Conference on Security Information and Networks, pp.81-86, ACM, (2011).
14) T. Matsuda, D. Koizumi, M. Sonoda and S. Hirasawa, *On Predictive Errors of SQL Injection Attack Detection by the Feature of the Single Character*, Proceeding in 2011 IEEE International Conference on Systems, Man and Cybernetics, pp.1722-1727, (2011).
15) I.S.Isa, Z.Saad, S.Omar, M.K.Osman,K.A.Ahmad and H.A.Mat Sakim,: *Suitable MLP Network Activation Functions for Breast Cancer and Thyroid disease Detection*, In proceedings of Second International Conference on Computational Intelligence, Modeling and Simulation (2010).
16) T. Takiguchi, A. Sako, T. Yamagata and Y. Ariki,: *System Request Utterance Detection Based on Acoustic and Linguistic Features*, I-Tech Education and Publishing, pp. 539-550, (2008).
17) Jeong Hoon Ko, Jung Suk Joo, and Yong Hoon Lee,: *On the Use of Sigmoid Functions for Multistage Detection in Asynchronous CDMA Systems*, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 48, NO. 2, pp. 522-526, MARCH (1999).
18) W. G. Halfond, Viegas and A. Orso,: *A Classification of SQL Injection Attacks and Countermeasures*, College of Computing Georgia Institute of Technology IEEE, (2006).
19) A. Tajpour, M. Masrom, M.Z. Heydari and S. Ibrahim,: *SQL injection detection and prevention tools assessment*, 2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT 2010), pp.518-522, (2010).
20) Justin Clarke,: *SQL Injection Attacks And Defense*, Syngress Publishing Inc., 2009.
21) *SQL Injection Cheat Sheet*, http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/.
22) *SQL Injection Cheat Sheet*, http://michaeldaw.org/sql-injection-cheat-sheet.
23) *MySQL SQL Injection Cheat Sheet*, http://pentestmonkey.net/blog/mysql-sql-injection-cheat-sheet.