「なりすまし」ユーザ検知システムの性能評価 —隠れマルコフモデルと他の確率モデルの比較*

岡本 剛[†]・渡邊 裕司[‡]・石田 好輝[‡]

Test Statistics for a Masquerader Detection System —A Comparison between Hidden Markov Model and Other Probabilistic Models*

Takeshi OKAMOTO[†], Yuji WATANABE[‡] and Yoshiteru ISHIDA[‡]

The performance analysis is carried out for a masquerader detection system by a hidden Markov model (HMM). A Markov model and a probability distribution have been applied to many intrusion detection systems, however it is not clear to what extent the HMM improves the performance in detecting a masquerader, compared with other two probabilistic models. Statistics of these three models are compared and the advantage of the HMM for a masquerader detection system is clarified. Some implications useful in extending the other two probabilistic models to an HMM are also provided.

1. はじめに

近年のインターネットの拡大と共に、情報セキュリティの重要性が認識され、コンピュータシステムのセキュリティ強化が施策されはじめた。代表的なセキュリティ強化策として外部のネットワークから内部のネットワークを保護するファイアウォールの設置がある。ファイアウォールによって外部からのセキュリティ侵害行為（秘匿性、完全性、可用性の侵害行為）は減少するが、ファイアウォールだけではネットワーク内部のパスワードクラックやネットワークの盗聴、なりすまし行為などのセキュリティ侵害行為に弱い。

内部のセキュリティ侵害行為や「なりすまし」行為の検出には異常検出 (anomaly detection) が有効であるといわれている [1]。異常検出とは、侵入や不正行為を直接検出するのではなく、システムやユーザの振舞いを監視し、通常と異なる振舞いを検出する手法である。異常検

出では、まずユーザのシステム利用状況やネットワークトラフィックの状況などから、正常な振舞いを定義する。その正常な振舞いは、プロファイルとよばれるデータに集約される。プロファイルを作成したあと、観測されたユーザの振舞いが、正常な振舞いと異なるとき、警報などの処理によってシステム管理者に知らせる。プロファイルを作成する手法には、統計的手法 (たとえば, [2])、ニューラルネットワークを用いた手法 (たとえば, [3])、データマイニングの技術を利用した手法 (たとえば, [4]) など、様々な手法がある。

本論文では、統計的手法の一つである隠れマルコフモデル (以下, HMM) によって、不正に入手したパスワードで UNIX システムへログインする「なりすまし」ユーザを検出する手法について検討する。HMM は、状態遷移確率、シンボル出現確率、初期状態確率からなり、モデル化したい対象がその対象の状態によって出力が変化するような時系列データを表現するのに適している。状態遷移確率 (マルコフモデル) やシンボル出現確率 (出現確率分布モデル) は、なりすましユーザの検出に応用されてきた (たとえば, [5]) が、これらの確率モデルによる検出性能と HMM による検出性能の違いが明らかにされていない。

そこで、本論文では、出現確率分布モデル、マルコフモデル、HMM の検出性能を比較評価し、なりすましユーザの検出における HMM の有効性を明らかにする。また今後のなりすましユーザの検出に関する研究におい

* 原稿受付 2002年3月18日

[†] 神奈川工科大学 工学部 情報ネットワーク工学科 Department of Information Network Engineering, Kanagawa Institute of Technology; 1030 Shimo-Ogino, Atsugi, Kanagawa 243-0292, JAPAN

[‡] 豊橋技術科学大学 知識情報工学系 Department of Knowledge-based Information Engineering, Toyohashi University of Technology; Tempaku, Toyohashi, Aichi 441-8580, JAPAN

Key Words: masquerader, intrusion detection system, hidden Markov model, profile, command sequence.

て、従来の確率モデルからHMMに拡張する際に参考となるHMMのパラメータなどの情報を提供することを目的とする。

以下、2.では、まず本論文に関連する研究と本論文の目的について述べる。3.では、本研究のなりすましユーザ検知システムの構成や各部の処理について述べる。4.では、HMMと出現確率分布モデル、マルコフモデルの性能を比較し、HMMの有効性を評価する。5.では、プロフィールの容量や計算コストについて考察する。最後に、本研究で得られた結果についてまとめる。

2. 関連研究

HMMを用いた侵入検出手法は、入力データによって二つに分けられる。その入力データは、システムコールとコマンドである。システムコールを入力データとする手法ではプログラム内部の処理を検査できることから、プログラムの異常やトロイの木馬の検出に適している。システムコールを入力データとする異常検出には、ForrestらとChoiらによるものがある。

Forrestらは、特定のプログラムのシステムコールを入力データとし、その特定プログラムの異常を発見するために、HMMを含め4種類の手法と比較している[6]。HMM以外の手法には、データマイニングの手法の一つであるRIPPER[7]を応用した手法[4]と、Forrestらが提案した免疫的な手法[8]がある。HMMの状態数は、検出の対象となるプログラムが利用するシステムコールの数に応じて設定され(状態数は20から60)、状態遷移モデルは遷移規則に制限がないErgodicモデルであった。検出結果の比較では、HMMが最もよい結果を得ていた。

Choiらは、特定のシステムコールを入力データとし、システムレベル、ユーザレベル、グループレベルの三つの段階に分けて、侵入行為として代表的なu2r(User to Root)¹の検出に関して検出の性能を評価している[9]。また、入力データ長やHMMの状態数について分析し、HMMの適用に有効な情報を提供している。

コマンドを入力データとする侵入検出は、ユーザの操作履歴を検査できることから、異常操作やなりすましユーザの検出に適している。コマンドの利用頻度やコマンドの連鎖を調べる方法は、なりすましユーザの検出に有効であることが示されている(たとえば、[5]など)。また、コマンド列は、UNIXシステムやその互換システムにおいて、シェルのヒストリ機能やアカウンティングシステム(以下、acctシステム)により採取できる。コマンドを入力データとする異常検出には、OrwantやLaneによるものがある。

Orwantはユーザが実行したコマンドを入力として、ユーザの作業内容を推定する手法を提案している[10]。作業内容は、Orwantが作成した13個の状態数からな

る8個のHMMによって推定される。一つのHMMの各状態はコンパイルや編集など詳細な作業状態を表し、各HMMはハッキングや画像処理などマクロな作業を表している。Viterbiアルゴリズムによって最適状態系列を推定し、8個のHMMの中で最尤の最適状態系列を計算したHMMから現在の作業状態を推定する。Orwantによる手法は侵入検出のためのものではないが、HMMによってコマンド列からユーザの作業を推定する点に関して関連がある。

Laneはユーザ単位にプロフィールを作成して、全ユーザのコマンド列と全ユーザのプロファイルの全組み合わせについてコマンド列の生起確率を計算し、ユーザごとに検出の正確さを評価している[2]。その結果、HMMはコマンドに現れるユーザの特徴を認識できることを明らかにした。また状態数の分析では、状態数が大きいHMMはプロフィールされたユーザ(本人)の認識率が高く、状態数が小さいHMMは、プロフィールされていないユーザ(他人)の認識率が高くなることを発見した。HMMにおいて最適な状態数はユーザごとに異なり、コマンド列の複雑さに関係していることを指摘している。

HMMは状態遷移確率とシンボル出現確率分布を有する確率モデルである。状態遷移確率は、確率モデルの一つであるマルコフモデルで表現される。マルコフモデルと出現確率分布モデルは、コマンド列を用いた侵入検出によく用いられる(たとえば、[5])。HMMは、これら二つの確率モデルと等価に置き換えることが可能である。LaneはHMMをなりすましユーザの検出に応用したが、HMMの基本モデルであるマルコフモデルと出現確率分布モデルの性能と比べて、どの程度性能が優れているかを明らかにしていない。そこで、本研究では、HMM、出現確率分布モデル、マルコフモデルの性能を比較評価して、HMMの有効性を明らかにする。

3. なりすましユーザ検知システム

侵入検知システムのフレームワークとして共通侵入検知フレームワーク(以下、CIDF)[11]がある。本節では、CIDFに基づいてシステムの構成を述べる(Fig.1参照)。CIDFは、イベント生成部、イベント分析部、レスポンス部とイベントデータベースからなる。

イベント生成部では、acctシステムから得られたコマンド列を前処理して、そのコマンド列をイベント分析部とイベントデータベースへ伝える。イベントデータベースは、イベント分析部でプロフィールを作成するために、受け取ったコマンド列を保存する。イベント分析部では、プロフィールの作成と、コマンド列が入力される確率(以下、生起確率)の計算を行う。その生起確率は、レスポンス部に送られる。レスポンス部では、入力された生起確率と閾値を比較して、なりすましの判定を行う。コマンド列を入力したユーザがなりすましユーザであった場合は、警報を発してシステム管理者に通知する。

¹一般ユーザがルート権限を不正に取得する行為である。

Fig. 1 Overview of a masquerader detection system

以下では、各部の詳細な処理について述べる。

3.1 イベント生成部

イベント生成部では、acct システムから得られたコマンド列を前処理して、そのコマンド列をイベント分析部とイベントデータベースへ伝える。本節では、コマンド列の採取と、前処理の方法について述べる。

3.1.1 コマンド列の採取

コマンド列を採取する手段には、シェルの履歴機能と acct システムがある。履歴機能では、ユーザがシェル上で入力したすべてのコマンドを採取できるが、プログラム内部で実行されたコマンド（たとえば、シェルスクリプト内で実行されたコマンドや X のウィンドウマネージャ上で実行されたコマンドなど）を採取できない。それに対して、acct システムではユーザの権限で実行されたすべてのコマンドを採取できる。また、シェルの履歴機能による採取はアプリケーションレベルであるのに対して、acct システムはカーネルレベルであることから、acct システムから採取されるデータは信頼性が高い。したがって、本システムでは acct システムからコマンド列を採取する。

3.1.2 コマンド列の前処理

採取されたコマンド列は、イベント分析部に送られる前に、四つの処理が施される。以下に、その処理の詳細と、その処理を行う理由について述べる。

ステップ1 コマンド列をセッション（ログイン～ログアウト）単位に整形する

採取されたコマンド列には、ユーザの権限で実行されたすべてのコマンドが含まれる。そのコマンド列には、ログインしていない期間のコマンドも含まれる。ユーザがログインしていない期間に実行されたコマンドは実際のユーザの振舞いとは関係ないため、それらを取り除いて、セッション単位のコマンド列を作成する。

ステップ2 セッション単位のコマンド列からユーザが直接実行していないコマンドを削除する

セッション単位のコマンド列には、ログイン時に呼び出される初期起動プログラムや、crond によって定期的に行われるコマンドなどユーザが直接実行していないコマンドが含まれる。これらのコマンドはユーザが誰であろうと必ず実行される。そのため、これらのコマンドは実際のユーザの振舞いとは関係ないため、コマンド列から削除する。

ステップ3 冗長なコマンド列を削除する

セッション内のコマンド列には、多数の連続した同じコマンドが含まれる。とくに、スレッドを生成するコマンドでは、スレッドが生成されるたびにスレッドに対応したコマンドがログに記録される。連続したコマンド列はプロファイルの作成や検出において冗長なので、連続した冗長なコマンド列を削除する。

ステップ4 コマンドを一意に識別可能な正整数にマッピングする

イベント分析部にコマンド列を入力するために、それぞれのコマンドを一意に識別できる正整数にマッピングする。マッピングされたコマンド列（正整数列）は、イベント分析部とイベントデータベースに送られる。

3.2 イベント分析部

イベント分析部では、プロファイルの作成と、コマンド列の生起確率の計算を行う。計算された生起確率は、レスポンス部へ送られる。本節では、プロファイルの作成と、生起確率の計算について述べる。

3.2.1 プロファイルの作成

プロファイルの作成は、一つのサブネットにアカウントを有する全ユーザに対して行われる。つまり、プロファイルの個数はコンピュータのユーザ数になる。プロファイルの作成には、イベントデータベースから事前に蓄積しておいたコマンド列を用いる。本論文では、プロファイルの作成に用いるデータと検出の評価に用いるデータをそれぞれ「学習データ」と「テストデータ」とよぶ。これら二つのデータは互いに独立である。

本論文では、三つの確率モデル（HMM、出現確率分布モデル、マルコフモデル）の性能を比較するため、確率モデルごとにプロファイル λ_H , λ_P , λ_M を作成する。

統計的手法によるプロファイルの作成では、学習データを十分採取できないとき、「学習データに含まれないが、テストデータに含まれるコマンド」が存在すると、それらの出現確率や遷移確率は 0 になるため、そのコマンドが含まれるコマンド列の生起確率が 0 になる。正当なユーザが、学習データに含まれなかったデータをたった 1 回入力するだけで、なりすましユーザとして判定されてしまうことを意味する。

そこで本システムでは、学習データに含まれなかったコマンドの出現確率や遷移確率を、0 に設定するのではなく、0 に近い値 $\varepsilon (> 0)$ に設定する (スムージング [12] を行う)。推定したコマンドの出現確率 (または遷移確率) が ε より小さいとき、その確率を ε に置き換える。なお、 ε の大きさは検出精度に影響するので、4.2.3 で、 ε の大きさと検出精度の関係を調べる。

以下では、HMM、出現確率分布モデル、マルコフモデルに関するプロファイルの作成方法について述べる。

[HMM のプロファイル]

HMM のプロファイル λ_H は、HMM のパラメータに対応する。HMM のパラメータには、状態遷移確率行列 $\mathbf{A}$ 、シンボル (コマンド) 出力確率行列 $\mathbf{B}$ 、初期状態確率行列 $\boldsymbol{\pi}$ がある。コマンド列 $\mathbf{O}$ において、第 k セッションの第 t コマンド目のコマンドを $\mathbf{o}_t^{(k)}$ ($t=1, \dots, T_k, k=1, \dots, K$) とし、状態を q_t^k ($t=1, \dots, T_k, k=1, \dots, K$) とすると、それぞれの確率行列は $\mathbf{A} = [a_{ij}] = [P(q_t^k = j | q_{t-1}^k = i)]$ 、 $\mathbf{B} = [b_j(v)] = [P(\mathbf{o}_t^{(k)} = v | q_t^k = j)]$ 、 $\boldsymbol{\pi} = [\pi_i] = [P(q_1^k = i)]$ である。 $i, j (i=1, \dots, N)$ は状態を表し、 $v (v=1, \dots, V)$ はコマンドを表す。本論文では、Baum-Welch アルゴリズム [12] を用いて、これらのパラメータを推定する。

Baum-Welch アルゴリズムでは、まずランダムに λ_H を生成し、次の (1) 式から (7) 式をくり返し適用し、 $|P(\mathbf{O} | \bar{\lambda}_H) - P(\mathbf{O} | \lambda_H)|$ を最小化する (閾値以下に達するまでくり返す)。

$$\bar{a}_{ij} = \frac{\sum_{k=1}^K \frac{1}{p_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) a_{ij} b_j(\mathbf{o}_{t+1}^{(k)}) \beta_{t+1}^k(j)}{\sum_{k=1}^K \frac{1}{p_k} \sum_{t=1}^{T_k-1} \sum_{j=1}^N \alpha_t^k(i) a_{ij} b_j(\mathbf{o}_{t+1}^{(k)}) \beta_{t+1}^k(j)} \quad (1)$$

$$\bar{b}_j(v) = \frac{\sum_{k=1}^K \frac{1}{p_k} \sum_{\substack{t=1, \\ s.t. \mathbf{o}_t^{(k)}=v}}^{T_k} \alpha_t^k(j) \beta_t^k(j)}{\sum_{k=1}^K \frac{1}{p_k} \sum_{t=1}^{T_k} \alpha_t^k(j) \beta_t^k(j)} \quad (2)$$

$$\bar{\pi}_i = \frac{\sum_{k=1}^K \frac{1}{p_k} \sum_{j=1}^N \alpha_1^k(i) a_{ij} b_j(\mathbf{o}_2^{(k)}) \beta_2^k(j)}{\sum_{k=1}^K \frac{1}{p_k} \sum_{i=1}^N \sum_{j=1}^N \alpha_1^k(i) a_{ij} b_j(\mathbf{o}_2^{(k)}) \beta_2^k(j)} \quad (3)$$

ただし、

$$\alpha_1^k(i) = \pi_i b_i(\mathbf{o}_1^{(k)}) \quad (4)$$

$$\alpha_{t+1}^k(j) = \sum_{i=1}^N \alpha_t^k(i) a_{ij} b_j(\mathbf{o}_{t+1}^{(k)}) \quad (5)$$

$$\beta_{T_k}^k(j) = 1 \quad (6)$$

$$\beta_t^k(i) = \sum_{j=1}^N a_{ij} b_j(\mathbf{o}_{t+1}^{(k)}) \beta_{t+1}^k(j) \quad (7)$$

である。そして、閾値以下に達したパラメータ $\bar{\lambda}_H$ を推定パラメータ λ_H とする。

[出現確率分布モデルのプロファイル]

出現確率分布モデルのプロファイル λ_P は、コマンドの出現確率を要素とする出現確率行列 $\mathbf{C}$ に対応する。コマンドの出現確率行列 $\mathbf{C}$ は、 $\mathbf{C} = [c_v] = [P(\mathbf{o}_t^{(k)} = v)]$ であり、コマンドの出現確率 c_v は、

$$c_v = \frac{\text{ユーザが入力したコマンド } v \text{ の総数}}{\text{ユーザが入力したコマンドの総数}} \quad (8)$$

によって求められる。コマンドの出現確率行列 $\mathbf{C}$ は、状態数 1 の HMM におけるシンボル出現確率行列 $\mathbf{B}$ に等しい。

[マルコフモデルのプロファイル]

マルコフモデルに関するプロファイル λ_M は、コマンド列の先頭に現れるコマンドの出現確率を要素とする初期出現確率行列 $\mathbf{D}$ と、コマンド間の遷移を表した遷移確率を要素とするコマンド遷移確率行列 $\mathbf{M}$ から構成される。初期出現確率行列 $\mathbf{D}$ は、 $\mathbf{D} = [d_v] = [P(\mathbf{o}_1^{(k)} = v)]$ であり、コマンドの出現確率 d_v は、

$$d_v = \frac{\text{コマンド } v \text{ で始まるセッションの総数}}{\text{ユーザのセッション総数 } K} \quad (9)$$

によって求められる。コマンド遷移確率行列 $\mathbf{M}$ は、 $\mathbf{M} = [m_{xy}] = [P(\mathbf{o}_t^{(k)} = y | \mathbf{o}_{t-1}^{(k)} = x)]$ である。 $x, y (x, y=1, \dots, V)$ はコマンドを表す。 m_{xy} は、

$$m_{xy} = \frac{\text{コマンド } x \text{ からコマンド } y \text{ に遷移した総数}}{\text{コマンド } x \text{ から遷移した総数}} \quad (10)$$

によって求められる。

3.2.2 コマンド列の生起確率の計算

コマンド列の生起確率の計算は、ユーザがログインしたアカウントのプロファイル λ を用いて、ユーザがログインしたときに入力された、長さ t のコマンド列 $\mathbf{o}_1, \dots, \mathbf{o}_t$ に対して行われる。このコマンド列が検査の対象となる。本論文では、このコマンド列をテストコマンド列とよび、そのコマンド列の長さ t をテストコマンド列長とよぶ。各確率モデルにおけるコマンド列の生起確率は、

$$P(\mathbf{o}_1, \dots, \mathbf{o}_t | \lambda_H) = \sum_{i=1}^N \alpha_t(i) \quad (11)$$

$$P(\mathbf{o}_1, \dots, \mathbf{o}_t | \lambda_P) = P(\mathbf{o}_1 | \lambda_P) \cdot P(\mathbf{o}_2 | \lambda_P) \dots P(\mathbf{o}_t | \lambda_P) \quad (12)$$

$$P(\mathbf{o}_1, \dots, \mathbf{o}_t | \lambda_M) = P(\mathbf{o}_1 | \lambda_M) \cdot P(\mathbf{o}_2 | \mathbf{o}_1, \lambda_M) \dots P(\mathbf{o}_t | \mathbf{o}_{t-1}, \lambda_M) \quad (13)$$

から求める。HMM では (11) 式、出現確率分布モデルは (12) 式、マルコフモデルは (13) 式を用いる。(11) 式の

$\alpha_t(i)$ は, (4), (5) 式から求められる.

検査の対象となるコマンド列は, ログイン直後の固定長コマンド列である. コマンドの利用頻度やコマンドの連鎖を調べる方法は, なりすましユーザの検出に有効であることが示されている (たとえば, [5] など). このことから, ユーザの特徴は, 作業内容 (利用するコマンドの種類) やその作業順序 (コマンドの実行順序) に現れると考えられる. そこで本システムでは, 作業の開始が明確なログイン直後の操作 (コマンド列) に焦点を当てる. なお, コマンド列の長さは, 検出精度に影響すると考えられるので, 4.2.2 でテストコマンド列の長さ と検出精度の関係を調べる.

3.3 レスポンス部

レスポンス部では, イベント分析部で計算されたコマンド列の生起確率からログインしたユーザがなりすましユーザであるかどうかを判定する. なりすましの判定は, 検知システムの管理者が事前に設定した閾値にしたがって行われる (この閾値は, HMM のパラメータ推定に用いる閾値と異なる). 生起確率が閾値以上であれば正当なユーザとして, 閾値より小さければなりすましユーザとして判定する. 判定した結果, ログインしたユーザがなりすましユーザであった場合には, 警報を発する.

4. 実験

4.1 実験データの作成

実験では, 一つのサブネットにアカウントをもつユーザ 8 名からコマンド列を 3 ヶ月にわたって採取した. ユーザが利用した計算機の OS はすべて Vine Linux 2.0 であり, 利用できるコマンドはすべて同じである.

なりすましユーザのコマンド列を採取するのは困難であるので, 本実験では, なりすましユーザのコマンド列の代わりに, 各アカウントのユーザ以外 (他人) のコマンド列を用いる.

本実験用の前処理されたコマンド列は, 評価時のテストコマンド列の長さによって変わる (詳細は, 4.2.2 を参照) が, 260 から 465 セッションで構成されている. 本実験では, 前処理によってセッション単位に分割されたコマンド列を, セッション数が均等になるように二つに分け, 一つを学習データとしてプロファイルの作成に用い, もう一方をテストデータとしてなりすましユーザの検出に用いた.

4.2 性能の評価

侵入検出の性能評価では, 誤報 (false positive) と欠報 (false negative) の頻度について議論される. 誤報とは, 正当なユーザをなりすましユーザとして判定することであり, 欠報とはなりすましユーザであるにもかかわらず正当なユーザとして判定することである. 一般に, 誤報を減らすと欠報が増え, 欠報を減らすと誤報が増える関係にある.

侵入検出の研究では, この関係を視覚化するために ROC (Receiver Operating Characteristic) [13] 曲線が用いられることがある. 本論文でも ROC 曲線を用いて各手法の性能を評価する. ROC 曲線は, 横軸を誤報率, 縦軸を欠報率とし, なりすまし判定の閾値 (3.3 参照) を変化させて, 閾値ごとにプロットしたものである. 誤報率とは, アカウントの正当なユーザから採取された全セッションのうち, なりすましユーザとして誤って判定したセッションの割合である. 欠報率とは, 他人のアカウントから採取された全セッションのうち, 他人であるにもかかわらず他人ではない (なりすましユーザであるにもかかわらずなりすましユーザではない) と誤って判定したセッションの割合である. 本論文では, 各手法の違いを視覚的にわかりやすくするため, ROC 曲線を近似曲線で描く.

ROC 曲線のグラフにおいて性能がよい手法とは, グラフの軸に漸近して最も原点に近づく曲線を描く手法である. つまり, グラフの軸と ROC 曲線で囲まれた面積が小さいとき, その性能は優れていると考えられる. そこで本研究では, ROC 曲線とグラフの軸で囲まれた面積 (以下, ROC 面積) を性能評価尺度 (以下, 検出精度) として用いる.

本節では, 各セッションのテストコマンド列長を 10, ε を各手法で最もよい結果であった値 (4.2.3 参照) に設定して各手法の検出精度を比較した. Fig. 2 に ROC 曲線を, Table 1 に ROC 面積を示す. HMM の状態数と状態遷移モデルは最も検出精度が高かった状態数と状態遷

Fig. 2 ROC curve of a test command sequence with length 10

Table 1 ROC area for each model

Probabilistic model	ROC area
Hidden Markov model	0.100
Markov model	0.115
Probability distribution model	0.151

移モデルを用いた (4.2.1 参照)。

Table 1 から、HMM の検出精度が最も高いことがわかる。マルコフモデルから HMM に拡張することによって、約 10.4% の改善を期待できる。出現確率分布モデルからの拡張では、約 31.8% の改善がみられた。しかし、ROC 面積に基づいた改善率は手法全体の性能を表しているにすぎないので、局所的な部分では HMM より他の手法が優れていることがある。それは、Fig. 2 の誤報率が 10% 付近で確認することができる。誤報率が 10% 付近では HMM より出現確率分布モデルの方が軸に漸近していることから、誤報率を 10% ぐらいに抑えたいときは出現確率分布モデルの方がよい検出手法になる。

4.2.1 HMM の状態遷移モデルと状態数

HMM の検出精度は、状態遷移モデルや状態数によって変化することが知られている [12]。本節では、状態遷移モデルと状態数に関して性能を評価する。

HMM によるプロファイルの作成は、3.2.1 で述べた Baum-Welch アルゴリズムを用いる。推定されたパラメータは、初期パラメータに依存するため、100 個の初期パラメータを用意し、最もよい結果を出力したパラメータをプロファイルに採用した。初期パラメータの状態遷移モデルには、代表的な遷移モデルとして、状態遷移に制約がない Ergodic モデルと、状態 i から状態 j への遷移は $i < j$ のときのみ許される Left-Right モデルの二つを評価する。

HMM の最適な状態数は、コマンド列の複雑さに比例すると考えられる [2] ので、HMM の状態数をユーザが使用したコマンドの数に比例するようにする。たとえば、20 種類のコマンドを使ったことがあるユーザであれば、割合が 50% のとき、状態数は 10 となる。本実験では、ユーザが使用したコマンドの数に対して、五つの割合 10%, 30%, 50%, 80%, 100% に対応する状態数について調べた。テストコマンド列の長さは 10、 ε は 10^{-250} とした。

Table 2 に、各状態遷移モデルと各状態数の全組合せに関する ROC 面積を示す。いずれの状態数においても Left-Right モデルが最もよい結果であった。Left-Right モデルでは、状態数が利用コマンド数の 30% を超えると、ROC 面積はほとんど変わらなかった。また、状態遷移モデルが異なっても ROC 曲線の概形に違いはなかった。

Table 2 ROC area for each state proportion and state transition model

State transition model	State proportion				
	10%	30%	50%	80%	100%
Left-Right model	0.136	0.106	0.103	0.100	0.106
Ergodic model	0.153	0.134	0.112	0.122	0.114

さらに、いくつかの ε とテストコマンド列長を変化させて、性能を評価した結果、状態数が利用コマンド数の 30% を越え、状態遷移モデルが Left-Right モデルのとき

に、最もよい検出精度が得られた。HMM では状態数が増加するにつれて計算コストが増大する¹ので、本節以降では、検出精度と計算コストを考慮して、HMM では状態数をユーザの利用したコマンド数の 50%²とし、状態遷移モデルを Left-Right モデルとする。

4.2.2 テストコマンド列の長さ

テストコマンド列は長くなるにつれてユーザの特徴が多く含まれるようになり、検出の精度が改善されると考えられる。本節では、テストコマンド列の長さと検出精度の関係を明らかにする。 ε は各手法で最もよい検出精度を得た値に設定し、テストコマンド列の長さを 5 から 25 まで変化させて、三つの手法の検出精度を比較する。まず、コマンド列の長さをそろえるために、それぞれの長さに満たないコマンド列を実験データから削除する。本実験データでは、それぞれの長さにそろえられたコマンド列は、長くなるにつれてデータ数 (セッション数) が減少する。長さが 5 のときは、総セッション数は 465 セッションあるが、長さが 25 のときには 260 セッションまで減少する。これは 1 セッションを構成するコマンド列の長さが短いものが多いためである。1 セッションの作業が、おもにメールのチェックやホームページの閲覧であった場合、コマンドを実行することが少ないため、コマンド列は短くなる。

Fig. 3 に、テストコマンド列の長さを変化させて各手法の ROC 面積をプロットした結果を示す。横軸がテストコマンド列の長さ、縦軸が ROC 面積である。検出精度の最もよい手法はテストコマンド列の長さごとに異なるが、おおむね HMM が最もよい手法といえる。全体の傾向として、テストコマンド列が長くなるにつれて検出精度が改善されることが確認された。また長くなるにつれて、三つの手法の性能の差が小さくなることから、十分に長いコマンド列を検査できるときは、いずれの手法でもよい検出精度が期待できると考えられる。

4.2.3 ε の大きさ

ε はコマンドの出現確率や遷移確率の最小値である。学習データであまり含まれなかったコマンドがテストコマンド列に含まれるとき、 ε の大きさはそのコマンド列の生起確率に影響する。本節では、 ε の大きさと検出精度の関係を明らかにする。

テストコマンド列長を 10 に設定し、 ε を 10^{-2} から 10^{-250} まで変化させて、前節と同様に、三つの手法の検出精度を比較した。Fig. 4 に ε と ROC 面積の関係を示す。横軸が ε で、縦軸が ROC 面積である。Fig. 4 より、各手法の最適な ε は異なることがわかる。最も検出精度

¹本実験では、プロファイルの作成にかかる計算時間はおよそ状態数の 3 乗に比例した。

²検出精度は 80% のとき最も高いが計算コストが大きく、30% のときは計算コストが小さいが検出精度が低いいため、その両者の間にある実験データとして 50% の値を適用した。

Fig. 3 Length of a test command sequence vs. ROC area

がよかった手法は ε が 10^{-250} のHMMであるが、マルコフモデルでは ε を 10^{-7} に設定すれば、HMMと同程度の検出精度が得られる。各手法の全体の傾向として、 ε が小さくなるにつれて検出精度が改善されている。 ε を小さくしたとき、ユーザが使用していないコマンドを含むコマンド列の生起確率は小さくなる。つまり、 ε を限りなく0に近づけたとき、ユーザが過去に使用していないコマンドを含むコマンド列を、なりすましユーザが入力したコマンド列として検出する。しかし、ユーザは過去に使用していないコマンドを使う可能性があるため、 ε を0に近づけた場合、誤報が多くなり全体の検出精度は低下すると考えられる。ところが、本結果では ε を小さくするにつれて、出現確率分布モデルとHMMの検出精度が改善されている。出現確率分布モデルとHMMの検出精度が改善される原因は次節で考察する。

Fig. 4 ε vs. ROC area

4.2.4 コマンドに対するユーザの嗜好

本実験では、Fig.2からわかるように、出現確率分布モデルは最も単純なモデルであるにもかかわらず、その検出精度は残り二つの確率モデルの検出精度に近い値であった。出現確率分布モデルは、コマンドの利用頻度に基づいてなりすましユーザを検出することから、本実験

の対象になったユーザには、ユーザ間のコマンドの嗜好に大きな違いがあることを示している。実験に用いたテストコマンド列長10のデータを調べると、本人のテストデータに、学習データには含まれなかったコマンドが含まれる割合が8.7%であったのに対して、他人のデータでは79.1%であった。学習データに含まれていないコマンドがテストコマンド列に含まれたとき、そのコマンド列をなりすましユーザが入力したコマンド列として判定する単純な検出方法によって、誤報率8.7%、欠報率20.9%の検出精度が得られる。この検出精度は、出現確率分布モデルのROC曲線において ε を適切に設定したとき、ROC曲線上で最も原点に近い点に近づく。つまり、コマンドを使用したかしていないか判定するだけでも性能が改善される。これは、 ε を限りなく0に近づけることに等しい。そのため、 ε を小さくすることによって、HMMの検出精度も改善されたと考えられる。

したがって、なりすましユーザが使用したコマンド集合が正当なユーザが過去に使用したコマンド集合にすべて含まれるとき、コマンドの出現確率に基づいた手法(HMMと出現確率分布モデル)は、なりすましユーザを検出することが難しいと考えられる。そこで、本節では、全ユーザが過去に使用したことがあるコマンドだけで構成されるテストコマンド列に対して、それぞれの性能を評価する。テストコマンド列長を10、 ε を各手法において最もよい検出精度であった値に設定して、各手法のROC面積を評価した。その結果、いずれの性能も低下したが、HMMの検出精度が最も高かった。HMMは出現確率分布モデルより48.9%、マルコフモデルより5.50%優れていた。この結果から、HMMの検出性能は、三つの確率モデルの中で、ユーザのコマンド嗜好の影響を最も受けないと考えられる。これは、HMMがコマンドの出現確率だけでなく、隠された状態の遷移確率を用いることによって、残り二つの確率モデルより時系列データの表現力が豊かになっているためと考えられる。

5. 考察

本節では、プロファイルの容量や実環境での運用における計算コストについて考察する。

5.1 プロファイルの容量

HMMのプロファイルは、状態遷移確率行列とシンボル出現確率行列、初期状態確率行列から構成されるので、プロファイルの容量は、これらの行列に含まれる要素の個数で決まる。シンボル数を V 、状態数を K で表すと、状態遷移確率行列には K^2 個、シンボル出現確率行列には VK 個、初期状態確率行列には K 個の要素がある。したがって、HMMのプロファイルの容量は、 $K^2 + VK + K$ と表せる。本実験の状態数は、ユーザが利用したコマンド数の割合に対応した数を適用しているので、その割合を a ($0 < a \leq 1$)とすると、状態数は $K = aV$

である。本システムにおける HMM のプロファイルの容量は、 $(aV)^2 + aV^2 + aV$ となる。一方、マルコフモデルのプロファイルは、コマンドの遷移確率行列と初期出現確率行列から構成されるので、プロファイルの容量は、 $V^2 + V$ となる。出現確率分布モデルでは、コマンドの出現確率行列から構成されるので V である。

これらの容量を比較すると、最も容量が小さいモデルは出現確率分布モデルあり、HMM とマルコフモデルは、状態数 aV とシンボル数 V によって、その大小関係は変化する。HMM とマルコフモデルの容量を比較すると、

$$0 < a < \frac{-(V+1) + \sqrt{5V^2 + 6V + 1}}{2V} \quad (14)$$

を満たすとき、HMM のプロファイルの容量がマルコフモデルより小さくなる。ここで、

$$a_{max} = \frac{-(V+1) + \sqrt{5V^2 + 6V + 1}}{2V} \quad (15)$$

とおくと、シンボル数が $1 \leq V < \infty$ のとき、 $0.618 < a_{max} \leq 0.732$ である。つまり、状態数がユーザの使用したコマンド数の約 61.8% を超えなければ、HMM のプロファイルの容量はマルコフモデルのものより小さい。HMM の性能評価 (4.2.1) では、状態数がユーザの使用したコマンド数の 30% 以上であれば検出精度は他のモデルより優れていたことから、本システムでは、HMM のプロファイルの容量は、マルコフモデルのものより小さくできる。

5.2 検査に要する計算コスト

なりすましユーザの検査では、まず 3.2.1 の (11) 式から (13) 式に基づいて、コマンド列の生起確率を計算する。つぎに、その生起確率が、事前にシステム管理者が設定しておいた閾値以上であれば、正当なユーザのコマンド列とし、閾値より小さければ、なりすましユーザのコマンド列として判定する。この検査に要する計算コストは、(11) 式から (13) 式より、マルコフモデルと出現確率分布モデルの計算量は T のオーダーであるが、HMM では N^2T のオーダーを要する。

実環境では、複数のユーザが同時に、検査に必要な規定のコマンド数に達することが予想される。このとき、複数のユーザのコマンド列を同時に検査しなければならないが、HMM は他のモデルと比べて検査にかかる計算コストが大きいので、検出に遅れが生じる可能性がある。そのため、リアルタイム性が求められる環境では、HMM の計算コストを改善する必要がある。

5.3 プロファイルの更新に要する計算コスト

コマンド列を入力データとするなりすましユーザ検知システムの実用性を追求した場合、ユーザが使用するコマンドやコマンドの実行順序の変化に追従できることが望まれる。そのようなユーザの変化に追従する方法として、プロファイルの更新がある。Schonlau らは、プロ

ファイルを更新することによって、プロファイルを更新しない時よりも良好な結果が得られることを示している [5]。プロファイルを更新する間隔には、セッション単位や、コマンド単位、時間単位などが考えられる。更新の間隔は、プロファイルの計算に要する時間より十分に長くしなければ、システムが過負荷になることやプロファイルの更新が間に合わないなどの問題が生じる。とくに、HMM におけるプロファイルの更新は、他のモデルと比べて多大な計算コストを要する。そのため、HMM を適用する際には、更新間隔を長くするなど、プロファイルの更新コストを考慮したシステムを設計する必要がある。

6. おわりに

本研究では、コマンド列を入力データとするなりすましユーザ検知システムを構築し、三つの確率モデル (HMM, 出現確率分布モデル, マルコフモデル) の検出性能を比較評価した。その結果、次のことを明らかにした。

- HMM の検出精度は、出現確率分布モデルより約 3 割、マルコフモデルより約 1 割優れている。
- ユーザ間にコマンドの嗜好がない場合においても、HMM の検出精度が最も高い。
- HMM の状態遷移モデルは、Left-Right モデルがよく、状態数はユーザが使用したコマンド数の 30% 以上のときに良好な検出精度が得られる。

考察では、プロファイルの容量や計算コストについて議論し、実環境での運用における今後の課題を示した。

謝 辞

本研究は国際コミュニケーション基金、学術振興財団の科学研究費 (C10650431)、立松財団、および 21 世紀 COE プログラム「インテリジェントヒューマンセンシング」の支援を受けて行われた。

参考文献

- [1] 武田, 磯崎: ネットワーク侵入検知, ソフトバンク (2000)
- [2] T. Lane: Hidden Markov models for human / computer interface modeling; *Proc. the IJCAI-99 Workshop on Learning About Users*, pp. 35-44 (1999)
- [3] J. M. Bonifacio Jr, A. M. Cansian, A. C. P. L. F. de Carvalho and E. S. Moreira: Neural network applied in intrusion detection systems; *Proc. the IEEE International joint conference on neural networks*, pp. 205-210 (1998)
- [4] W. Lee and S. J. Stolfo: Data mining approaches for intrusion detection; *Proc. the 7th USENIX Security Symposium* (1998)
- [5] M. Schonlau, W. DuMouchel, W. Ju, A. Karr, M. Theus and Y. Vardi: Computer Intrusion: Detecting Masquerades; *Statistical Science*, Vol. 16, No. 1, pp. 58-74 (2001)

- [6] C. Warrender, S. Forrest and B. Pearlmutter: Detecting intrusions using system calls: Alternative data models; *Proc. the 1999 IEEE Symposium on Security and Privacy*, pp. 133–145 (1999)
- [7] W. W. Cohen: Fast effective rule induction; *Proc. the 12th International Conference on Machine Learning*, pp. 115–123 (1995)
- [8] S. Forrest, S. A. Hofmeyr, A. Somayaji and T. A. Longstaff: A sense of self for unix processes; *Proc. the 1996 IEEE Symposium on Security and Privacy*, pp. 120–128 (1996)
- [9] J. Choi and S. Cho: Hidden Markov model for sequence recognition in intrusion detection system; *Proc. the 4th International Conference on Advances in Pattern Recognition and Digital Techniques* (1999)
- [10] J. Orwant: Heterogeneous learning in the doppelgänger user modeling system; *User Modeling and User-Adapted Interaction*, pp. 107–130 (1995)
- [11] P. Porras, D. Schnackenberg, S. Staniford-chen, M. Stillman and F. Wu: The common intrusion detection framework architecture; <http://www.gidos.org/drafts/architecture.txt>(1999)
- [12] L. Rabiner and B. Juang: *Fundamentals of speech recognition*, PTR Prentice-Hall (1993)
- [13] J. P. Egan: *Signal Detection Theory and ROC Analysis*, Academic Press (1975)