2003－CSEC－21 (14)
2003／5／16

コマンド入力型システムにおけるユーザ判別手法の提案

大坊 徹* 長瀬 智行** 吉岡 良雄**

概要

UNIX 等のコマンドで操作するタイプのシステムにおいては, ユーザの入力系列には本人が持つ知識や経験などからくる特徴がある．それに着目し，新たにシステムに入力されたユーザの入力系列とあらかじめシステム側に記録された入力履歴との類似度を計算し，それを比較することで，なりすましを発見する手法を提案する．入力系列と入力履歴との類似度を計算する方法として，WWW サーバやデータベースへの他人のアクセス履歴を用いて自分の目的の情報を検索する手法を適用している．この手法を用いた結果，本人同士のデータの類似度が他人との類似度に比べ，大きくなることがわかった．この性質を利用することで正規ユーザと不正ユーザを識別し，なりすましを発見することができる．

A new Intrusion Detection technique based on discriminating user's input Data

Toru DAIBOU* Tomoyuki NAGASE** Yoshio YOSHIOKA**

Abstract

In a system based on input command such as UNIX system, illegitimate intentions of a user's intrusion into the system can be detected from a sequence of user's input data. This paper proposes a new detection technique based on measuring the degree of the similarity and analyzing the behavior of user's input commands which include input commands' history and newly inputted commands. Although, the information of a user's behavior on WWW servers and databases is also collected that may provide user's intrusion on the system. Simulation results for interrelating user's input commands using the proposed technique demonstrate that the intrusions could be easily determined and quickly constrained.

１．まえがき

ネットワーク上のコンピュータで個人を識別するときには多くの場合, ユーザ ID とパスワードを使う．実名や顔が表に出ることなく，匿名のままインターネットのサービスを利用する事が可能である．これはある面ではインターネットの自由さを支えるものである．しかし匿名性は，裏を返せば無責任で悪質な行為につながる危険性をはらんでいる．そして他人のユーザ ID とパスワードを入手すれば, 容易に他人になりすますことが可能になる。さらには,「なりすまし」をインターネット上で見破ることは難しいのが現状である．

パスワードによる認証方式では，一度突破されると，そのユーザの行為はシステムが許可した範囲であればすべて許されてしまう．実際に外部からのパスワード破りなどの不正侵入による被害は数多く報告されているが，実際はデータ流出等の被害は内部犯の場合が多い．パスワード情報が比較的もれやすい内部犯に対してはパスワードのみのセキュリティでは不十分といえる．

*弘前大学大学院理学研究科情報科学専攻
Graduate School of Science, Hirosaki University
**弘前大学理工学部電子情報システム工学科
Faculty of Science and Technology, Hirosaki University

ユーザがコンピュータを使用する中で，その入力履歴には何らかの個人の特徴が含まれている．それは，ユーザの作業目的の違い，コンピュータシステムへの慣れ，知識，経験の差から生まれる．また，よく使用するコマンドやファイル，そしてあるコマンドにおいてよく使用するオプションやファイル名，またはファイル名そのものにも個人の特徴が現れる．

コマンド入力タイプのシステムにおいて入力履歴から個人の特徴を抽出し，それをユーザの認証に利用する方法として，あるコマンドからあるコマンドへの遷移確率を個人の特徴として認証に利用する手法が提案されている[1]．また最近では遺伝的アルゴリズムによって入力履歴から個人の特徴を抽出する手法も提案されている[2].

本論文では，入力履歴と現在システムを利用しているユーザの入力系列との類似度を計算することによって現在のシステム利用者が正規のユーザであるか，あるいは不正なユーザであるかを判別する手法を提案する．

２．ユーザの入力系列と入力履歴との類似度

２．１ 類似度計算に使用する特徴ベクトルの生成

まず正規ユーザが今までシステムを利用してきた際に残された入力履歴と，新しく入力されている系列との類似度を計算するために，それぞれの特徴ベ

クトルを定義する．

まず，新しく入力されている文字列で作る特徴ベクトルを入力特徴ベクトルと呼ぶ．これは入力中に存在する文字列，及び，その文字列ごとの入力系列中の出現数である．例えば，ユーザ U の入力系列の中に出現する文字列 A の出現数を $U_{N,A}$ と表す．この値を入力特徴ベクトルの各要素とする.すなわち入力特徴ベクトルを次式とおく．

$$V_{U,N} = (U_{N,A}\, U_{N,B}\, U_{N,C} \cdots \quad U_{N,M}) \qquad (1)$$

次に入力特徴ベクトルに存在する文字列を対象に履歴特徴ベクトルを生成する．すなわち入力特徴ベクトルと同様に，ユーザ U の入力履歴の中に出現する文字列 A の出現数を $U_{R,A}$ と表す．もし履歴中に入力系列に存在する文字列が無い場合はその要素数は０とする．こうして得られた値を履歴特徴ベクトルの各要素とする.この履歴特徴ベクトルを次式とおく．

$$V_{U,R} = (U_{R,A}\, U_{R,B}\, U_{R,C} \cdots \quad U_{R,M}) \qquad (2)$$

２．２　　ベクトル間の類似度

ここでは前節で説明した２つのベクトル間の類似度の計算法について説明する．

T_U を履歴特徴ベクトルを生成する際に対象とした入力履歴のデータ数，$N_{U,R}$ を履歴特徴ベクトルの各要素の値の合計とする．このとき，2 つのベクトルの類似度 S は，次式で表す[3][4].

$$S = \frac{N_{U,R}}{T_U} \qquad (3)$$

３．正規ユーザと不正ユーザでの類似度の差

３．１　実験内容

本人同士のデータを照らし合わせた時と他人のデータと照らし合わせた時に類似度にどのような差が出るかについて調べる為，次のような実験を行った．

実験の被験者は研究室所属の大学院生５名である．この 5 人が所属する研究室にあるシステムに記録されている一定数の入力履歴を用いた．実験では各ユーザの入力履歴のうち，新しい方から一定数のデータを新しくシステムに入力されたデータと見立て，入力特徴ベクトルを生成する．入力特徴ベクトルの生成に使用しなかった残りのデータを各ユーザのシステムに記録された入力履歴とし，履歴特徴ベクトルを生成する．

３．２ 一定数の履歴と入力との類似度の傾向（実験－1）

実験 1 では，まず各ユーザとも履歴特徴ベクトルを作る為に，最新の入力履歴 250 データ中で連続した 1 から 200 データを使用した．また，残りの 201 から 250 データをユーザが入力したデータと仮定し，入力特徴ベクトルを生成するのに用いた．入力データと履歴データを連続したものに設定したのは，正規ユーザが行う作業にはある程度継続性があり，データ間の類似度は高いからである．

表１　各ユーザの履歴と入力の類似度
（履歴データ数 200，入力データ 50）

Log Data	Input data A	B	C	D	E
A	**0.6**	0.405	0.4	0.56	0.4
B	0.33	**0.785**	0.325	0.33	0.325
C	0.12	0.115	**0.365**	0.165	0.115
D	0.165	0.1	0.15	**0.36**	0.12
E	0.16	0.13	0.16	0.16	**0.26**

表２　各ユーザの履歴と入力の類似度
（履歴データ数 200，入力データ 30）

Log Data	Input data A	B	C	D	E
A	**0.56**	0.365	0.365	0.37	0.365
B	0.315	**0.74**	0.31	0.31	0.31
C	0.09	0.115	**0.285**	0.17	0.135
D	0.095	0.095	0.16	**0.305**	0.13
E	0.115	0.13	0.155	0.155	**0.255**

表３　各ユーザの履歴と入力の類似度
（履歴データ数 200，入力データ 10）

Log data	Input data A	B	C	D	E
A	**0.49**	0.335	0.345	0.015	0.345
B	0.305	**0.435**	0.3	0	0.3
C	0.1	0.095	**0.205**	0.07	0.165
D	0.1	0.09	0.185	**0.145**	0.135
E	0.075	0.085	0.11	0.035	**0.205**

表 1 はユーザＡからユーザＥまでの各ユーザ間の履歴と入力系列との類似度を示している．また，入力データ数によって類似度がどのように変化するかについても調べた．表２は各ユーザの最新の入力履歴データ数 230 のうち入力データ数を 30 とした場合の結果である．表３は各ユーザの最新の入力履歴データ数 210 のうち入力データ数を 10 とした場合の結果である．

表１から表３における対角線の類似度を S^R とする．また，対角線以外の類似度を S^F とする．このとき，S^R に関して見ていけば，最も高い S^R を示したのはユーザＢであり，最も小さい S^R を示したのはユーザＥ

である．

表 1 と表 2 で列ごとに見ていくと，S^R がどの S^F より高い値を示したユーザは A と B のみである．表 3 ではユーザ A，B に加え，ユーザ D も S^R がどの S^F よりも高い値を示した．

実験 1 では S^R が S^F を上回る場合が最高で 5 人中 3 人という結果になった．S^F が S^R を上回る結果が見られた原因は各ユーザの入力履歴に共通に存在する文字列があるからである．

この場合，どのユーザの入力にもよく現れる文字列が存在すると考えられる．すべてのユーザの入力履歴に存在する文字列は，システム使用中のユーザが本人であるか，あるいは不正に利用している他人であるかを判別することができない．

3．3 特定の入力に関する情報を除去した際の類似度の傾向（実験－2）

3．3．1　類似度計算対象外の文字列の選択

実験 1 の結果を踏まえ，実験 2 では各ユーザに共通に存在する文字列については類似度の計算対象から外した上で類似度を計算した．

まず，類似度の計算対象から外す文字列として“ ls ”と“ cd ”に着目した．[5]の実験のなかで，セッション中に“ ls ”と“ cd ”の出現頻度が平均 93.3%であったことからこの二つを選択した．

表 4　入力履歴中の“ ls ”と“ cd ”の割合

	Input 50	40	30	20	10	平均
A（ls）	0.395	0.38	0.355	0.335	0.335	0.36
（cd）	0.01	0.01	0.01	0.005	0.005	0.008
B	0.325	0.32	0.31	0.3	0.3	0.311
	0.05	0.05	0.05	0.055	0.06	0.053
C	0.085	0.085	0.085	0.09	0.095	0.088
	0.03	0.03	0.03	0.03	0.03	0.03
D	0.075	0.07	0.07	0.07	0.075	0.072
	0.01	0.01	0.01	0.01	0.01	0.01
E	0.115	0.115	0.115	0.09	0.075	0.102
	0.005	0.005	0.005	0.005	0	0.004

表 4 は実験 1 で使った各ユーザの入力履歴中に存在する“ ls ”と“ cd ”の割合である．各列は入力データ数を 50 から 10 まで 10 ずつ変化させた時の結果である．これを見ると，ユーザ A と B は平均してログの 3 割以上が“ ls ”であることがわかる．また，“ cd ”はすべてのユーザに存在しているが，入力履歴中における割合は最大でもユーザ B の平均 0.053 となっており，どのユーザに関してもその割合は小さいものとなっている．

入力履歴中で“ ls ”の値が最も小さかったのはユーザ D で 0.07 である．ここで“ ls ”と“ cd ”の他に類似度に影響を与える可能性がある文字列が無いかを探す為，入力履歴中から 0.07 以上の割合で存在する文字列を抜き出し，その結果を表 5 にまとめた．表 5 から各ユーザに 0.07 以上の割合で共通して存在する文字列は“ ls ”のみであることがわかる．

表 5　各ユーザのログの中で 0．07 以上の割合で存在する文字列（入力データ数 30）

	Log Data				
	A	B	C	D	E
lm35-1	0	0	0	0	0.09
lamtest3	0	0	0.07	0	0
cd ..	0.14	0	0	0	0
cd jasper	0	0.075	0	0	0
emacs &	0	0.11	0	0	0
emacs lm35-1.c	0	0	0	0	0.1
login	0	0	0	0.08	0
ls	0.355	0.31	0.085	0.07	0.115
ls –a	0	0	0	0.075	0
make	0	0.085	0	0	0

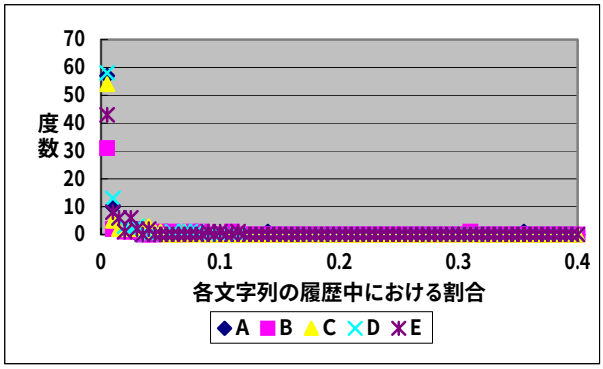

図 1　各ユーザの入力履歴中に存在する文字列それぞれの割合の度数分布（入力データ数　30）

また図 1 は各ユーザの入力履歴に存在する文字列の履歴中における割合を横軸とした時の度数分布であり，入力データ数を 30 とした場合の結果である．入力された文字列は，そのほとんどが入力系列の中での割合が小さい事が分かる．全ユーザを見ても，度数が 2 以上の割合は 0.04 までである．それ以上の割合で入力履歴中に存在する文字列はどのユーザの履歴にも存在する可能性が高い．したがって，各ユーザの特徴を表す文字列は入力履歴中の割合が 0.04 までの中に存在している．

この事から，あるユーザの入力系列に“ ls ”が存在した場合，式（3）では入力履歴中の“ ls ”の個数を類似度に加算されている．すなわち，そのユーザがもつ特徴を表している文字列に関する情報が，“ ls ”の入力履歴中の個数が多いことにより丸め込まれることになる．したがって，類似度を計算する際には“ ls ”のように，全てのユーザの入力履歴中に存在し，かつその中での割合が比較的高い文字列は類似度計算の対象外とする．

３．３．２　計算対象外文字列選択後の類似度

表６　各ユーザの履歴と入力の類似度
（履歴データ 200，入力データ数 50，
“ ls ”を除く）

	Input data				
	A	B	C	D	E
A	**0.339**	0.017	0.008	0.273	0.008
B	0.007	**0.681**	0.000	0.007	0.000
C	0.038	0.033	**0.306**	0.087	0.033
D	0.097	0.027	0.081	**0.308**	0.049
E	0.051	0.017	0.051	0.051	**0.164**

表７　各ユーザの履歴と入力の類似度
（履歴データ 200，入力データ数 30，
“ ls ”を除く）

Log data	Input data				
	A	B	C	D	E
A	**0.318**	0.016	0.016	0.023	0.016
B	0.007	**0.623**	0.000	0.000	0.000
C	0.005	0.033	**0.219**	0.093	0.055
D	0.027	0.027	0.097	**0.253**	0.065
E	0.000	0.017	0.045	0.045	**0.158**

表８　各ユーザの履歴と入力の類似度
（履歴データ 200，入力データ数 10，
“ ls ”を除く）

Log data	Input data				
	A	B	C	D	E
A	**0.233**	0.000	0.015	0.023	0.015
B	0.007	**0.193**	0.000	0.000	0.000
C	0.006	0.000	**0.122**	0.077	0.077
D	0.027	0.016	0.119	**0.157**	0.065
E	0.000	0.011	0.038	0.038	**0.141**

表６から表８は，“ ls ”を類似度計算の対象外とした場合の各ユーザの入力履歴と入力系列との類似度を計算した結果である．これらから，全てのユーザが自分の入力履歴との類似度が最も高くなっている事が分かる．このような結果になるのは，ユーザ本人の入力系列にはその入力履歴の中に存在する文字列が，他人の入力系列よりも多いことが挙げられる．すなわち，“ ls ”のようにシステムを操作する上で使用する機会が多いコマンドはどのユーザの入力履歴にも存在する事が多い．これを類似度計算の対象外としたことで，S^R の値が下がった．しかしその一方で S^F は限りなく 0 に近い値をとることとなった．

４．類似度の時系列変化（実験－3）

前節では本人と他人の類似度について見てきた．ここでは，本人の類似度変化を時系列変化について考察する．実験では，各ユーザの入力履歴を用い，新しい方から一定数の入力履歴を新しく入力されたデータと見立てる．そして入力されたデータに見立てた分を取り除いた残りのデータのうち最新の 200 データを類似度計算に使う履歴データとした．ここでも入力データ数が 10，30，50 の 3 通りの場合について調べた．そしてその入力データ数ごとに計算に使うログの対象範囲をデータが古い方へスライドさせ，それを時系列変化と見立てた．

各ユーザの履歴データ数には差があるため，ログの量が多いユーザ B とユーザ E についての結果を示す．結果は 10 回分の計算結果となっており，図は縦軸が類似度，横軸が時間を表している．横軸は数字が大きいほど，過去の履歴データを用いた計算となっている．また，ここでの類似度の値は計算対象から“ ls ”を取り除いた時のものである．

図 2 と図 3 から入力データ数の変化に関わらず，その時の入力系列と入力履歴の状態によって，類似度は大きく変化することがわかる．この原因として主に考えられるのはユーザの作業目的の変化である．システムの使用目的が変われば，実際にシステムを使用しているのが正規のユーザであっても，その入力系列は今までの入力履歴にあるものとは違うものになる．類似度が前に計算した時点より下がっている場合は，そうした作業の変化が起こったものと考えられる．逆に類似度が前に計算した時点より高くなっているのは，作業が継続して行われている，もしくは以前に行った作業と同じような作業が行われたと考えられる．

５．ユーザ判別方法の検討

５．１　閾値を設定する方法

新たにシステムに入力された入力系列と入力履歴からシステムを利用したユーザが正規のユーザか，不正なユーザかを判断するために，予め閾値を設定する方法を考える．この場合，時系列の中で本人の入力であっても低い類似度を示すことが考えられる．しかし，求められた結果から，類似度の閾値を 0.1 に設定すれば，本人であるユーザを誤って他人であ

ると誤認証する問題はほぼ解決される．実験 2 の結果から，S^F が 0.1 以上の値を取るのは，入力データ数を 50 とした場合に一つ，そして，入力データ数を 10 とした場合に一つ，合わせて 2 つの場合である．そして図 4 は実験 2 の時点で求められた S^F の平均値を示している．これを見る限り S^F の値が 0.1 を超えることはない．これらの事から閾値を 0.1 に設定した場合，不正なユーザを正規のユーザだと誤認証してしまうことは比較的少ない．

実験 1 から実験 3 までの結果から，S^R は入力データ数が多ければ多いほど値が高くなる傾向にある．しかし，図 4 から S^F は入力データ数に必ずしも比例するとは限らない．したがって，入力データ数を 50 に設定した場合でも類似度が 0.1 を下回る傾向が強いユーザがいた場合は，さらに入力データ数を増やすことで本人を他人だと誤認証する問題に対処できる．

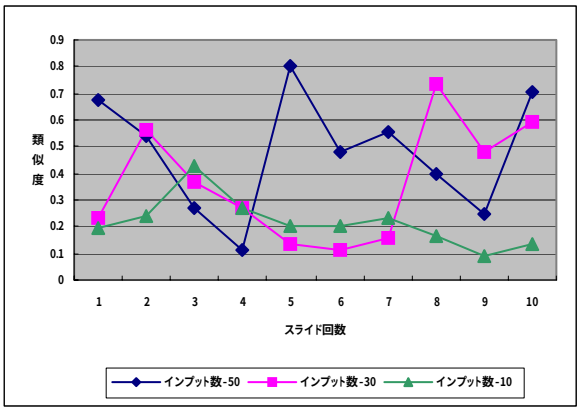

図 2　ユーザ B の時系列における類似度変化

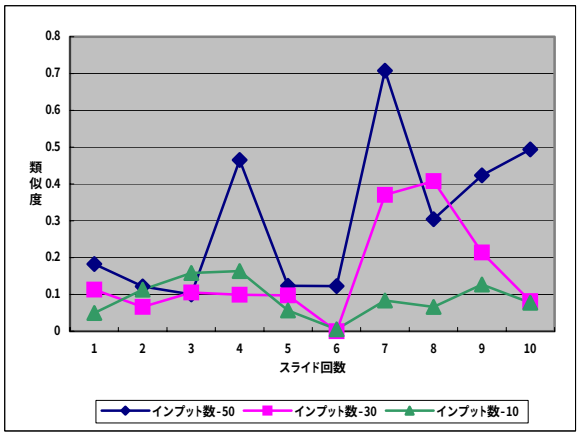

図 3　ユーザ E の時系列における類似度変化

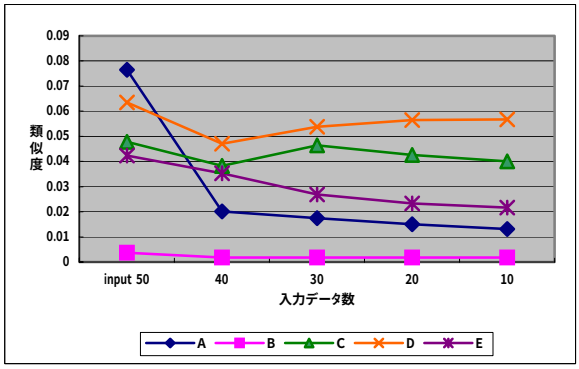

図 4　S^F の入力データ数の変化における平均値

5．2　類似度の大小で判別する方法

ユーザ判別の方法として，類似度そのものをユーザ判別の材料に使うアプローチも考えられる．あるユーザの入力系列に対し，最も類似度が高かった入力履歴のユーザが入力を行ったユーザであるとする考え方である．例えば，表 8 ではユーザ A の入力系列に対して最も高い類似度を示したのはユーザ A の入力履歴である．したがって，システムに入力操作を行ったユーザはユーザ A 本人であるとする．その他のユーザに関しても S^R がすべての S^F より高い値をとっている．よって，正規ユーザを不正なユーザと見なす場合はこの時点でゼロであり本人認証は問題なく行われる．また，ユーザ A がなりすましをするとしても，S^F の中で最も高いのはユーザ D の入力履歴に対してである．したがって，ユーザ A はユーザ D のアカウントを使ったなりすましでない限り，システムはなりすましを発見できることになる．このアプローチでの判別方法の有効性を検討するにあたり，表 6 から表 8 の中で S^F が 0.1 を超えているものに着目した．理由は，先に述べたように S^F が 0.1 を超える事はあまり無い．ゆえに，S^F が 0.1 を超えている場合はシステムがユーザを誤認証する可能性が高いからである．

S^F が 0.1 を超えているのは，表 6 で入力系列がユーザ D で入力履歴がユーザ A の場合と，表 8 で入力系列がユーザ C で入力履歴がユーザ D の場合の 2 つである．このことから，表 6 の中ではユーザ D がユーザ A に最もなりすましが出来るユーザである．また，表 8 ではユーザ C がユーザ D に最もなりすましが出来るユーザである．

表 6 の設定でユーザ A の入力履歴中に存在する文字列でその数が多いもの上位 4 つと，表 8 の設定でユーザ D の入力履歴中に存在する文字列でその数が多いもの上位 4 つを表 9 に示す．これらの文字列は，ユーザ A とユーザ D の特徴を表す文字列といえる．表 6，表 8 での結果はいずれも他人であるユーザの入力系列の中に表 9 に挙げた文字列が存在していたことから，他の S^F よりもその値が高くなったものである．具体的に文字列を挙げると，表 6 ではユーザ D の入力系列の中に“ cd .. ”が存在していた．また表 8 ではユーザ C の入力系列の中に“ exit ”と

“ history ”が存在していた．そこで，ユーザ D の入力履歴と入力系列中に“ cd .. ”がどの程度存在するかを調べた．ユーザ D の入力の中にユーザ A の特徴を表す文字列“ cd .. ”が多く存在すれば，認証システムがユーザ D をユーザ A と認証してしまう確率が高くなるからである．その結果，ユーザ D の入力履歴 200 データと入力系列 50 データの合わせて 250 データ中に ”cd .. ”は 2 個存在した。よって, ユーザ D の入力の中に“ cd .. ”が存在する割合は 0.8%である．

同様にユーザ C の入力履歴と入力系列の中に“ exit ”と“ history ”がどの程度存在するかも調べた．ユーザ C に“ exit ”は，入力履歴 200 データと入力系列 10 データの合わせて 210 データ中に 16 個存在した．よってユーザ C の入力の中に“ exit ”が存在する割合は 7.62%となる．また，“ history ”は 210 データ中 1 個存在した．したがって“ history ”が存在する割合は 0.48%となり，ユーザ D の特徴を表す文字列がユーザ C の入力中に存在する割合は 8.1%ということになる．

今このようにして求めた数値が，ユーザ D のユーザ A への，またユーザ C のユーザ D へのなりすましが本手法を用いた際に見逃される確率と考えられる．当然これは，他人であるユーザが無作為に作業を行った場合について言えることである．

表９　ユーザ A とユーザ D の特徴を表す文字列とその個数

	Log data (user A input 50)		Log data (user D input 10)	
1	ls	79	ls -a	16
2	cd ..	30	ls	15
3	su -	5	exit	12
4	cd lib	3	history	9

表１０　各ユーザの履歴と入力の類似度（各 S^R が最小のとき）

Log data	Input data A	B	C	D	E
A	**0.165**	0.015	0.045	0.053	0.053
B	0.000	**0.110**	0.000	0.000	0.000
C	0.005	0.033	**0.109**	0.071	0.033
D	0.016	0.027	0.065	**0.253**	0.065
E	0.000	0.006	0.050	0.050	**0.066**

表１０は実験３のなかで得られた各ユーザの S^R の中でその値が最も小さかった時の入力履歴を用いて，入力データ数を 30 とした時のすべての S^F を計算した結果である．これを見る限り，たとえ S^R が普段得られている値より小さくなる時があっても，S^F がその値を超える場合は無い．また，図３を見れば分かる通り，ユーザ E に関しては S^R が０となる場合があったので，表１０に示している値は，実際には 2 番目に小さいデータである．そこで，ユーザ E の S^R が 0 の時，全ての S^F について求めたところ，S^F も全て 0 となった．これにより，基本的に S^R が小さくなってもそれに伴い S^F も小さくなる事が予想できる．

以上述べたように，本人以外のユーザが無作為にシステム上で作業を行った際に，正規のユーザの特徴を示す文字列を入力する確率は非常に低い．さらに，システム利用者全員の入力履歴に現れる文字列は類似度計算の対象外とする．そのことが S^F の上昇を抑えている．したがって，類似度の大小でユーザを判別する方法を取った場合，本人を他人と誤認証する問題は無いといえる．

６．むすび

本論文では，ユーザ間の入力履歴とそこに新たに入力された入力系列との類似度によって，ユーザを判別する手法を提案した．今回のように，WWW サーバやデータベースで自分の目的の情報を検索するために過去のアクセス履歴を使う手法で求めた類似度が正規ユーザと不正ユーザを判別する問題に利用できることがわかった．

今後の課題は，実際に本手法を用いたシステムを実装することにより，その有効性を検証することである．その中でユーザ判別の為に閾値を設定した方がよいか，類似度の大小でユーザを判別した方がよいかについても検討する．

文　献

[1] 白井　治彦，西野　順二，小高　知宏，小倉　久和，“ 対話的計算機環境におけるコマンド入力連鎖を用いた認証方法 ” 信学論（A），Vol.J82-A No.10 pp.1602-1611 Oct，1999.

[2] 小高　知宏，白井　治彦，小倉　久和，“ コマンド入力系列における特徴の GA による抽出と認証への応用 ”信学論(D1)，Vol.J85-D1 No.5 pp.476-478 May，2002

[3] 早川和宏，鶴巻公治，浜田　洋，“ ユーザの利用履歴に基づく WWW サーバの類似検索 ”情処学情報メディア研報，21－2，pp. 11－17，May　1995

[4] 井原　雅行，金田　洋二，上野　圭一，金山　英明，“ ユーザの潜在的好み推定法 ” 信学論（A）Vol.J82-A No.5 pp.717-725 May　1999

[5] 小高　知宏，加藤　友彦，高田　光男，西野　順二，小倉　久和，“ 計算機利用者のシステム操作文字列に基づく認証手法の検討 ” 信学論（A），Vol.J79-A No.4 pp.1001-1003 April　1996