タイピングパターンによる認証

Authentication system using "typing-pattern" with Neural Network.

大野 祐嗣[†]，　羽賀 隆洋[‡]
Yuji OHNO,　Takahiro HAGA

Abstract　It is an authentication system that is necessary for the service which offer a computer. It confirms whether an inputted password fits what was registered in advance with the authentication system being used at present. But, when a password is omitted to the stranger, the matter that right is deceived can be done easily in this method. So, the way of certifying it by "typing-pattern" which is the movement which became a habit is proposed. It is difficult for a stranger to imitate it, and the movement made a habit can expect improvement in the security.

1.　はじめに

昨今のコンピュータ技術の目覚ましい発展は，コンピュータを生活になくてはならないものにした．またパソコンの急速な普及，さらにはインターネットの世界的な普及によって誰もがコンピュータに触れるようになり，コンピュータを使った作業やサービスはもはや特別なものではなくなった．

コンピュータを利用したサービスにはいろいろなものが考えられるが，ここでなくてはならないのが認証システム，つまり，あるサービスを受けるにあたってそのサービスを受けることを許可された本人であるかどうかを調べる手段である．

コンピュータの利用に限らず，本人であるかどうか確認されるといった状況は普段の生活でもよく遭遇する．パスポートの写真やサインを見比べて本人であることを見極めること，銀行へ行ってお金を降ろす際に印鑑を押すあるいは暗証番号を入力する，など，枚挙にいとまがない．

実生活における認証方法では個人を特定するための「鍵」は，

1. 印鑑や磁気カードのような物理的な所持品
2. 暗証番号のような記憶情報
3. サイン (筆跡) のような習慣化した動作
4. 指紋などの身体的特性

が用いられる．

[†]愛知工業大学大学院工学研究科修士課程 (豊田市)
[‡]愛知工業大学情報通信工学科 (豊田市)

ではコンピュータサービスにおける認証ではどうであろう．現在広く利用されているのは，個人の識別子たる「ＩＤ」と呼ばれる文字列と「パスワード」と呼ばれる文字列の入力を求め，その一致から本人を特定するという方法である．パスワードはキャッシュカードにおける暗証番号にあたるもので，先に述べた「鍵」のうち "記憶情報" にあたる．この方式では文字列の一致による確認しか行なっていないため万が一パスワードが漏洩した場合，他人が容易に本人になりすますことが出来る問題がある．これは大変危険である．キャッシュカードの暗証番号が他人に知れた場合の状況を想像してみれば危険であることが理解できるだろう．しかし世間一般においてコンピュータサービスにおけるパスワード漏洩による影響についての危機感は，まだまだ薄い．

そこで新たに「タイピングパターン」という "習慣化した動作" による「鍵」を与えることを提案する．習慣化された動作は他人が真似ることが困難であり，セキュリティレベルの向上が期待できる．

本稿ではタイピングパターン情報を今までのパスワードによる認証に加えることで，セキュリティの向上を図れないかどうか，ニューラルネットワークを用いた実験を実際に行ない検討する．

2.　諸理論

ここでは本研究に用いた理論を簡単に説明する．

2·1 タイピングパターン

ある程度キーボードの入力に慣れてくると，キーボード入力の際に何らかのくせが出てくるものと推測される．これを便宜上“タイピングパターン”と呼ぶことにするが，数値化するために具体的に，**ある文字列を入力するときに起こる入力時間間隔の並び**と定義する．

図1にタイピングパターンの例を示す．キーボードから文字列(例では「aitech」)を意識せずに連続して入力する際，ある文字のキーを押してから次のキーを押すまでには当然ながら時間遅延が発生する．この時間を測定し，一列に並べたものをタイピングパターンとする．

しかしながらキーボードでの入力は毎回微妙に変化し，同じものではありえない．従って，ある人物のタイピングパターンはこれ，という具合に定量的なものとして定義せず，刻々と変化する量として考える．

2·2 ニューラルネットワーク

ニューラルネットワークとは人間の脳細胞の学習の様子を計算機上でシミュレートしたもので，パターン認識などに極めて有益とされ，音声認識や手書き文字認識への応用がよく知られている．

ニューラルネットワークをパターン認識に利用するには「学習」という作業が必要である．ニューラルネットワークは概ね図2のような構造をしており，ある数値(群)を入力すると数値計算を行ない，それに対応した出力(群)が得られる．そして入力，即ちパターン認識をさせようとする問題に対して利用者が望む正しい答えであるかどうか出力と比較し，正しくなければニューラルネットワークのパラメータを調整して正しい答えが出るようにすることが学習である．

学習の完了したニューラルネットワークは入力パターンから適切な出力をするように調整されているため，これに対して未知の問題を入力して得られる出力を見極めれば入力が正しいかそうでないか，パターンに含まれるか否かが分かる．

この特性をタイピングパターンへ適用し，ニューラルネットワークを用いてタイピングパターンの認識を試みる．

2·3 タッチタイピング

高速タイピングに欠かせない技術としてタッチタイピングというものがある．

キーボードに不慣れな人はキーボードそのものを見ながらどのキーがどの位置にあるかを目視して1文字ずつ入力を行うものであるが，ある程度キーボード入力に慣れた人はキーボードを見ずに画面だけを見て入力することが可能である．

このように画面だけを見て入力することをタッチタイピング(ブラインドタッチ¶)と言う．

さて，一般的なタッチタイピングではキーボード上における指遣いが決まっている．

図3はその指遣いを示したものである．タッチタイプのできる人が全てこのような指遣いをしているとは限らないが，この指遣いが広く普及していることには間違いない．実際，タッチタイプ練習用のソフトウェアの多くはこの指遣いをもとに作られている．

本稿ではこれを参考に指の動きからタイピングパターンを測定することも行なう．具体的には，右手中指で打ってから左手小指で打つまでにかかる時間などを測定し個人の特徴が表れるデータの測定を目指す．

2·4 本人受理と他人棄却

認証システムにおける重要な要素のうちに，本人受理と他人棄却というものがある．これらは文字通り，入力が本人のものであると認めることと他人のものであるとして棄却することである．

現状で広く利用されている，コンピュータシステムにおけるパスワードやキャッシュカードにおける暗証番号のようなデータは完全一致の場合にのみ受理し，そうでなければ棄却すれば済むので判断は容易である．一方，タイピングパターンは毎回の入力で微妙に変化するアナログ値の並びであるため判断が難しい．この場合完全一致による比較では本人受理は不可能であるため，一定の許容範囲を設け融通を利かせる必要があるわけだがあまり許容範囲を広げすぎると他人を受理する誤認識に繋がるし，反対に狭めすぎると本人が受理されないという問題が発生する．

これらはいわゆるトレードオフ関係にあり，両方を満足しうることが困難なパラメータである．

ニューラルネットワークを用いたパターン認識では，このような微妙に変化するアナログデータであっても対応できるが，入力の微妙な変化に対して出力も変化するため適切なしきいを設定することが必要である．

3. 実験方法

前の章で述べた理論を用いて，実際にタイピングパターンによる認証実験を行なう．この章では実際の実験方法を順に述べる．

3·1 タイピングパターンの測定

まず，なによりも実験を行なうためにはタイピングパターンのデータが必要である．タイピングパターンデータの測定は次のように行なう．

¶差別的としてこの用語の使用を避ける傾向にある

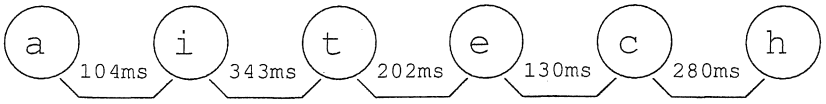

図 1: タイピングパターン

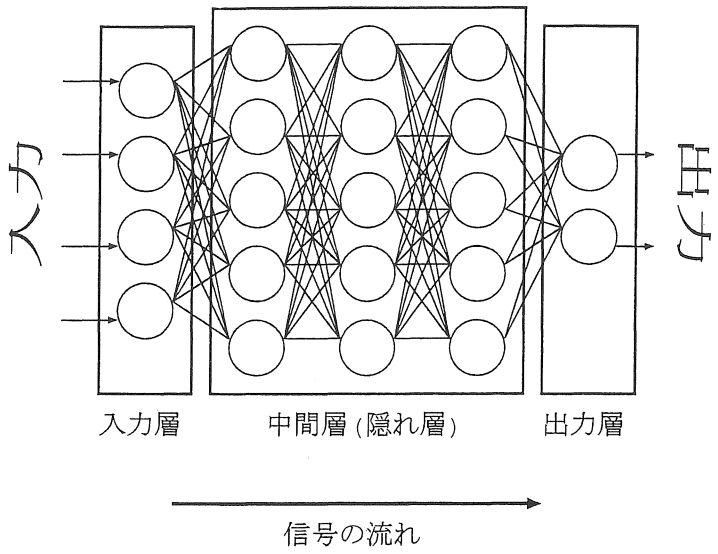

図 2: ニューラルネットワーク

3·1·1 方法

実際のタイピングパターンの測定は, キーボードを直接入力することにより行なう. C言語の関数によりパソコンの内部タイマを用いキーが押下された瞬間の時間を記憶し, 次のキーが押された瞬間までの差分を取ってミリ秒単位まで測定する.

ここで注意しなければならないのはネットワークにおけるタイムラグである. タイピングパターンを測定する場合, 時間をカウントするマシンがネットワーク越しに遠隔地にあるとすると, 被験者がキーボードを叩いてからその情報がマシンに伝わるまでにタイムラグが発生する. このタイムラグが常に一定であれば問題はないのだが, 通常は回線の混み具合などの影響を受け時々刻々と変化する. Internet の普及により, 実際にはネットワーク越しにパスワードを入力する状況も珍しいことではなくなったが, 今回はこうしたタイピングパターンに対する “ノイズ” が乗る可能性を排除するため, タイピングパターンの測定はローカルマシン上で行なうこととする.

さて, こうしてタイピングパターンのデータが取得できたわけであるが, このままのデータではニューラルネットを学習させることは出来ないので, 生の時間データではなく, 正規化した状態で保存される.

正規化は, まず入力する文字の 1 文字目から最終文字の入力までにかかる総時間で各時間間隔を割ることによって, それぞれの時間間隔が全体のどの程度の割合であるのかを求める (図 5).

この操作で得られるデータは時間間隔の割合の平均, つまり $\frac{1}{\text{文字数}-1}$ 付近に集中してしまうため, これを 0 から 1 に特徴づけるためにシグモイド (sigmoid) 関数のフィルタに通す.

シグモイド関数は図 6 のように, 中心値に対して十分に大きい数では 1 に近い値を返し十分に小さい数には 0 に近い値を返す. また中心付近の微妙な差を増幅して返すため, データの特徴づけを行なうのに有益な関数である. なお図において a は関数の立ち上がりの傾きを決定する定数であり, 小さいほど傾きは急となり強く特徴づけられる.

3·1·2 種類

タイピングパターンは “くせ” であるため本来ならばキーボードの扱いに慣れた人が, しかも各々が入力に

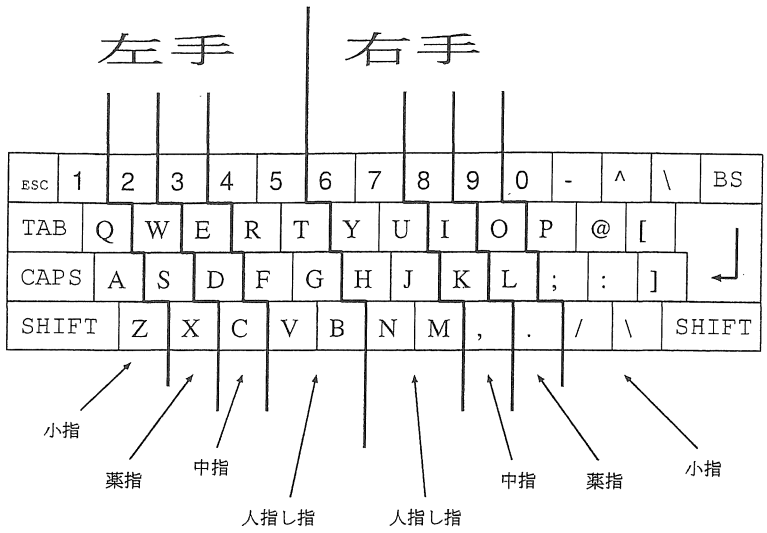

図 3: タッチタイプにおける指遣い

慣れた文字列で測定するのが望ましい．キーボードの扱いに慣れていない人はもちろんのこと，いくらキーボード入力に慣れた人だからといって，初めて入力する文字列，殊にそれがローマ字入力の日本語ではなく英単語だったりすると毎回の入力で時間間隔が大きく変わってくる可能性が高いからである．

このことを配慮して，データを取得するために入力してもらう文字列として次の 2 つの種類を用意した．

1. 特定の文字列「specialty」について
2. キーボードを入力する個人が任意に選んだ文字列について

この 2 種類について何人かに入力をお願いし，時間間隔を測定した．入力はニューラルネットワークの学習用データとして 1 人あたり 10 セット，実際の認証用として 10 セットの計 20 セットを採った．ただし，タイピングパターン測定に協力してもらった人すべてがキーボードの入力に慣れているわけではないことを記しておく．

なお，特定の文字列として「specialty」を選んだのは，タッチタイプの指遣いの図 (図 3) において，全ての指を使うからである．

3·2 ニューラルネットワークの学習

データが取得できたら認証に用いるニューラルネットワークを学習させる．

3·2·1 使用するニューラルネットワーク

今回学習に用いたニューラルネットワークは「誤差逆伝搬ニューラルネットワーク」と呼ばれるものである．

これは「教師あり学習」を行なうニューラルネットワークであり，学習に際して目標となる出力の組合せを指定すると，入力に対する出力が目標とどれだけ違うかの誤差を求め前素子にフィードバックしてニューロンの重みを調整するタイプのニューラルネットワークである．

入力には予め取得しておいたタイピングパターンのデータを一人あたり 10 セット与える．これを学習に参加する (認証実験に参加する) 人数分だけ繰り返して学習させる．

3·2·2 学習用データ

先に取得しておいた 2 種類のデータをそのまま使う方法と，それに対してさらにタッチタイプを参考にしたデータを付加する操作を行ない，これを新たな学習用データとする方法による実験を行なう．

付加するデータは 2·3 節で述べたように指の位置の移動を参考にし，既存パターン中のある指からある指までの移動時間の総和を求め，各指に対するデータを既存データの先頭に付加する．ただし，キーを叩く 8 本の指の全てに対してこの方法で時間測定すると先頭に 64 ものデータを付加することになり，ニューラルネットの学習が発散してしまう可能性や，学習がうまくいったとしても膨大な計算時間がかかってしまう可能性がある．また，同節でも述べたが全ての人が図 3 のような指遣いをしてるとは限らないため，結果として無意

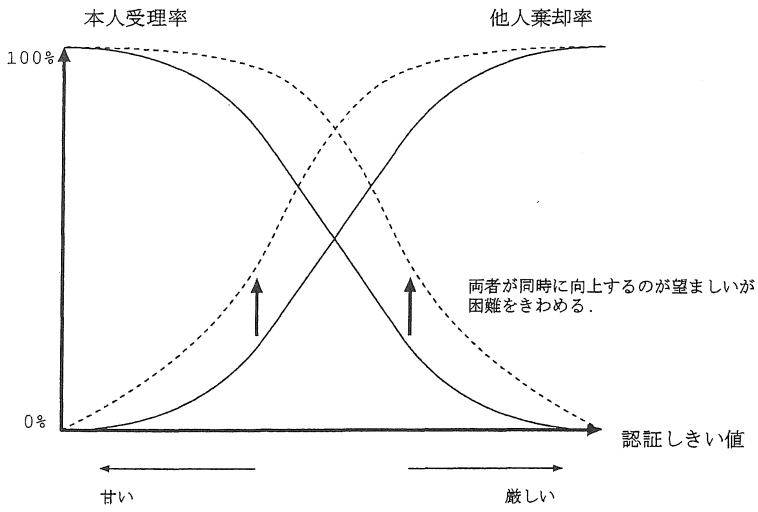

図 4: トレードオフ関係

味なデータになる恐れがある.

この問題を回避するため, 付加するデータは指遣いの図をそのまま流用するのではなく, 図 7 または図 8 のようにキーボードを 2 分割もしくは 4 分割として, 領域相互間の移動時間を測定するものとする.

3·2·3 諸条件

ニューラルネットワークは所有するニューロンの数によって学習がうまくいくかどうか (完了できるかどうか) が変化する. ニューロンが多すぎると重みが入力と出力を記憶するだけになってしまい, 逆に少なすぎると全ての学習用データに対してパラメータを調整できなくなり永遠に学習を繰り返すことになる. このことを考えると, 学習に参加する人数, つまりニューラルネットワークに与えるデータの組合せ数によってその規模を変化させるのがよいと推測できるが, 性能のよいニューラルネットワークの規模を一意に決定するのは困難である.

そこで, ニューラルネットワークの規模の決定は, 中間層の一層あたりのニューロン数を決める適当な一次式を定め, 人数に応じて変化させることとした.

なお, 中間層は 3 層で固定した.

3·2·4 学習完了

ニューラルネットワークの学習が終了したかの判断基準は, ニューラルネットワークの学習評価指標として一般的に使われる平均自乗誤差 (RMS) を用いる. これは, p 番目の学習パターンにおける j 番目の出力ユニットの目標 (教師信号) 値を t_{jp} , 実出力を x_{jp} とし, 学習パターンの個数を n_p , 出力ユニットの個数を n_o とすると,

$$\sqrt{\frac{\sum_p \sum_j (t_{jp} - x_{jp})^2}{n_p n_o}}$$

で表される. 今回はこの RMS が 0.02 を下回った場合に学習が完了したと判断して繰り返しを終了させる.

3·3 認証実験

ニューラルネットワークの学習が完了したら, それに未知の問題を入力して実際に認証を図る.

3·3·1 本人受理実験

ニューラルネットワークがタイピングパターンに対して認識を行なえるかどうか調べる手段として, まず本人受理に主眼をおいた実験を行なう.

学習の完了したニューラルネットワークに対して学習に参加した人物の, 学習に用いたものとは別のデータを与え, 出力層に現われる数値群を確認して本人のものと認識したかどうかを確認する.

ここでは, 学習目標に対する出力の誤差しきいを 0.00 (完全一致) から 0.01 刻みで 0.50 まであまくしていき, 全ての出力素子がこれを満たした場合本人として受理することとして認識率 (受理率) の変化をみた.

3·3·2 他人棄却実験

本人受理実験において比較的よい結果を示した条件に対して他人を棄却できるかどうかの実験を行なう.

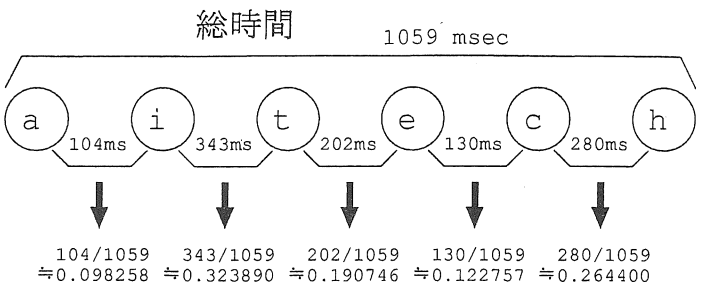

図 5: データの正規化

ある人物と同じパスワードを別の人物が入力した場合のタイピングパターンを用意し，学習の完了したニューラルネットワークに与える．そして出力される数値を，本人受理実験と同様に誤差しきいを 0.00 から 0.50 まで変化させ判断する．

4. 実験結果および検討

実際に認証実験を行なった結果を以下に示す．

実験はニューラルネットワークのパラメータと与えるデータの違いにより，非常に多くの種類について行なったが，紙面の都合上代表的な結果を示す．

4·1 本人受理実験

本人受理実験の結果のうち，任意の文字列のパスワードでタッチタイプ情報を付加した場合の結果を図 9 と図 10 に示す．図は同じ規模のニューラルネットワークに対して 2 人分～6 人分のタイピングパターンを与えて学習させたときの認識率を表したグラフである．

誤差しきい値は数字が大きくなるほど条件が甘いことになるので，グラフは右肩上がりとなる．この結果によると，任意の文字列においてキーボード 4 分割のときの情報を加えた場合では誤差しきい 0.2 より甘い場合，概ね 9 割以上の高い認識率が出ている．タイピングパターンデータの取得に協力してくれた人すべてがキーボードの入力に慣れているわけではないことを考えると，9 割という認識率は非常に高確率なものである．

これにより，タイピングパターンをニューラルネットワークに入力することによって本人の認証が可能である．と言える．

人数による違いを見比べると，学習に参加する人数を増やしていった場合，認識率が落ち込んでしまう．計算センターのように，不特定多数の人物が利用する環境における実装を考えるとこの問題は致命的である．しかし研究室や会社の部署内のような，他者の使用する可能性が低い環境における実装であれば，大人数における認識率の低下はさほど問題とならないだろう．

4·2 他人棄却実験

本人受理実験と同じ条件において他人棄却実験を試みた．

ここに挙げた図は，6 人分のデータを与えて学習させたときに本人ではない偽物のデータを 6 人分与えて棄却を試みた結果である．本人受理実験とは逆に誤差しきい値が大きくなると条件が厳しいことになるので，右下がりのグラフとなる．2 分割の場合は本人受理率が低い代わりに棄却成功率は高い．一方本人受理率の非常に高い 4 分割では棄却成功率が極端に下がってしまう．これは，同じ文字列に対して入力に極端な差がない場合，ニューラルネットワークによってまるめこまれてしまい，特徴を見出せないまま学習を行なっている可能性があることが考えられる．認証システムは裁判と違って “疑わしきは罰する” の特徴を持たせないと危険なので他人棄却に主眼を置く必要がある．図 13 は 2 つの実験結果の図を重ね合わせたものであるが，くしくもトレードオフ関係の図 (図 4) と同じような図となった．他人棄却を第一に考えて，これがうまくいくしきい値，例えば 0.1 付近を採用すると，本人認証が最悪の場合 20% 程度に落ち込み到底実用化出来なくなってしまう．他人棄却についてもう一工夫する必要があるだろう．

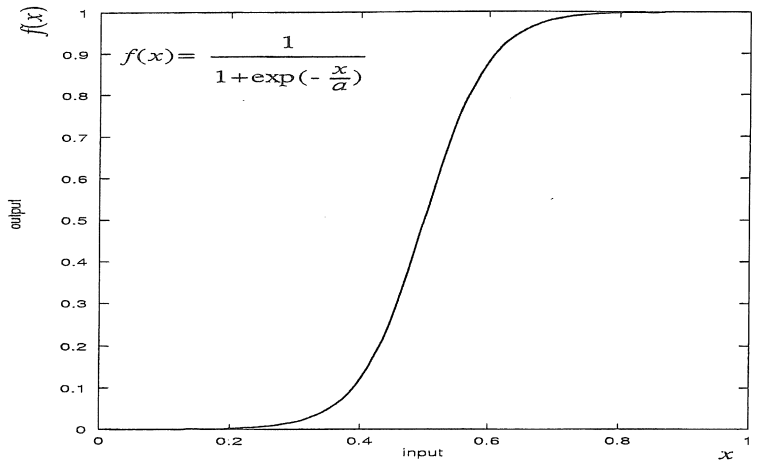

図 6: シグモイド関数

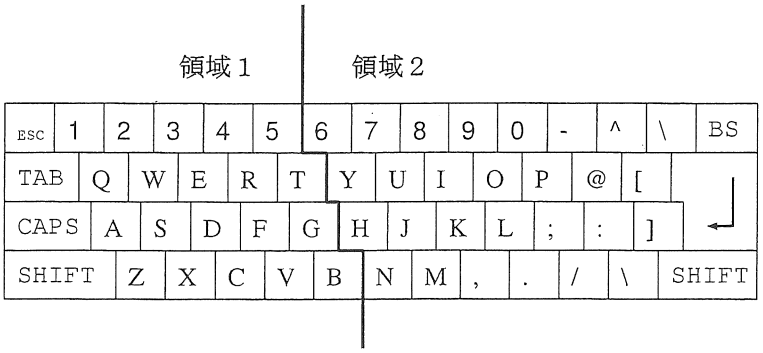

図 7: パターン測定における領域 (２分割)

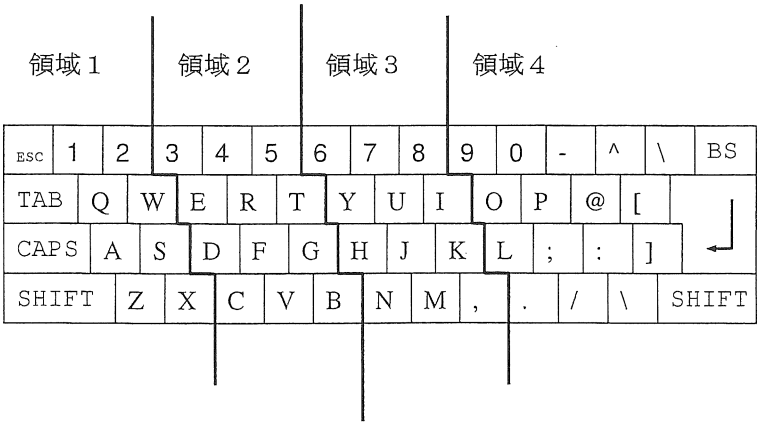

図 8: パターン測定における領域 (４分割)

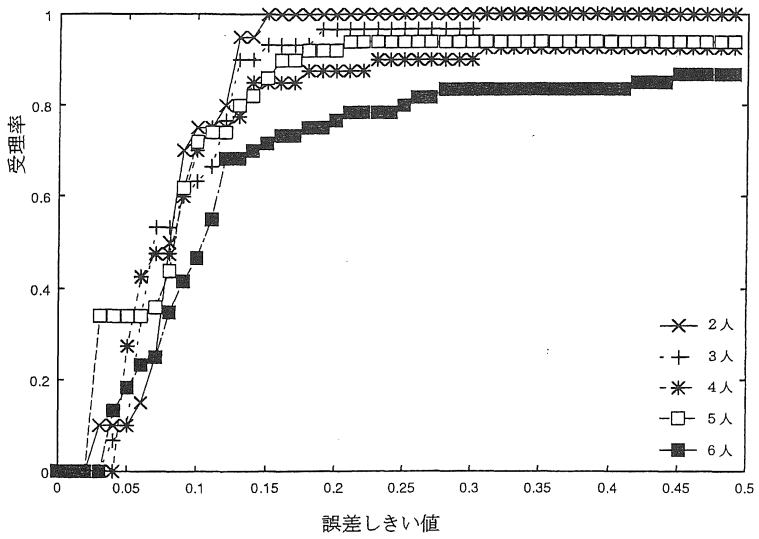

図 9: 本人受理実験 (任意文字列, 2 分割)

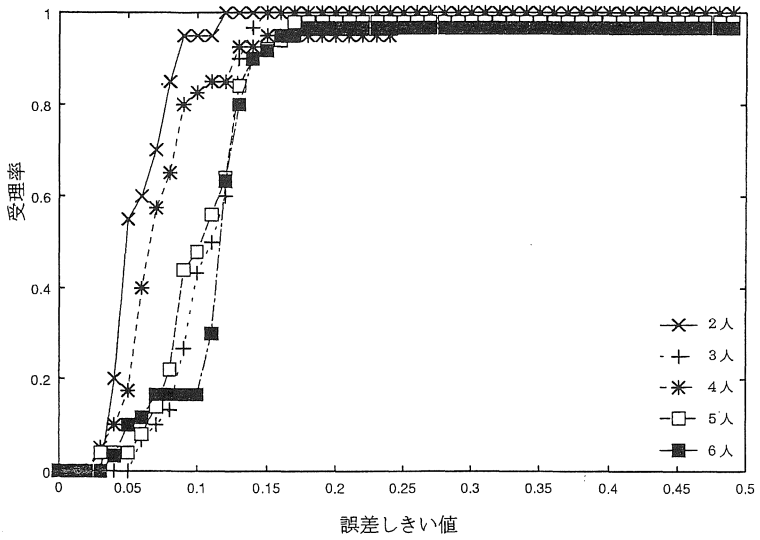

図 10: 本人受理実験 (任意文字列, 4 分割)

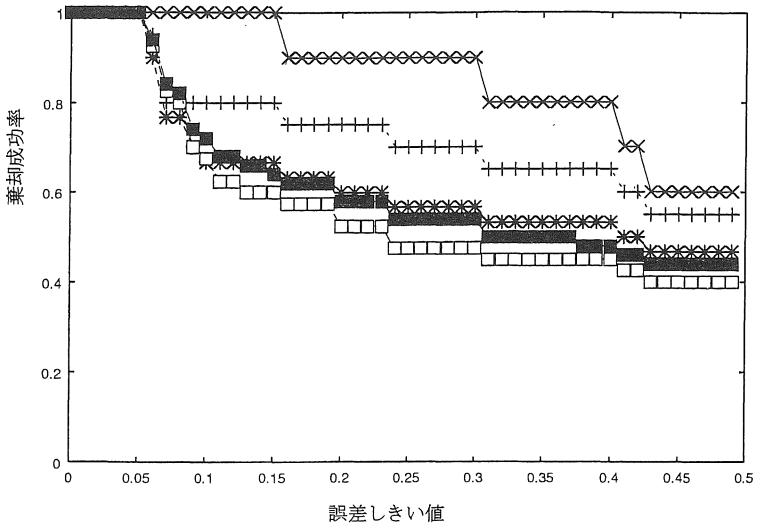

図 11: 他人棄却実験 (任意文字列, 2 分割)

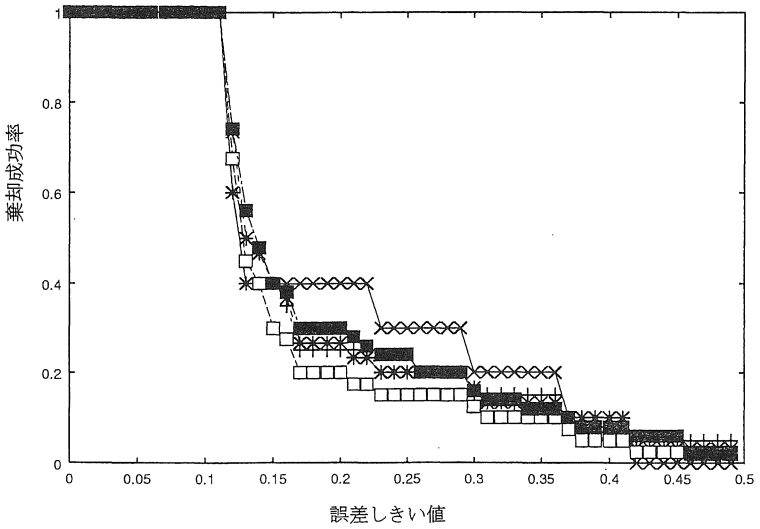

図 12: 他人棄却実験 (任意文字列, 4 分割)

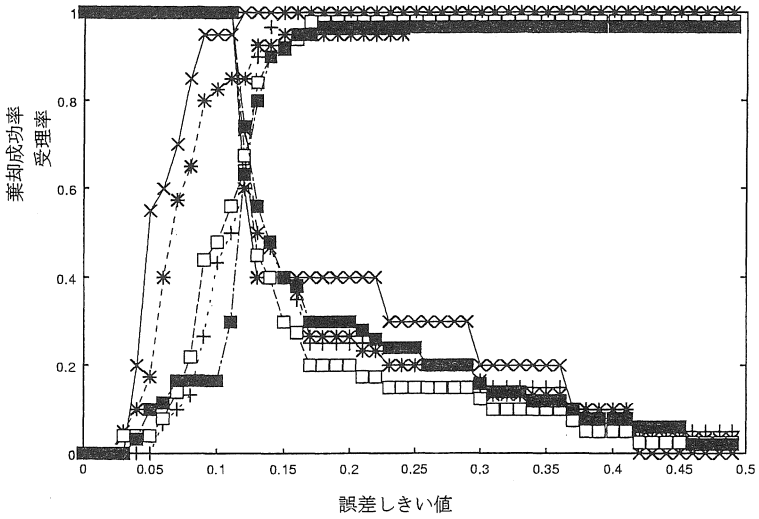

図 13: 本人受理率と他人棄却率 (図 10 と図 12 の合成)

5. まとめ

本稿ではタイピングパターンによる認証システムをニューラルネットワークを用いて実装し, 本人認証と他人棄却の２つの実験を行なうことによってセキリュティの向上が図れるかどうか検討した.

本人受理実験の結果を見れば, この方法でセキュリティ向上を図れる可能性がある, と言えるであろう. しかし, あくまでこの実験は認証システムとしての可能性が示されただけであって, このままの方式で実用になるというわけではない. 他人棄却実験の結果を併せてみると他人がうまく棄却できるしきいを設定した場合本人の受理がうまくいかない問題が発生してしまう.

ここをうまく解決しないことには, 実用に耐えうるシステムとすることは非常に困難である.

5.1 問題点

まず “タイピングパターン” 自身がキーボード入力にある程度慣れた人を想定していることである. コンピュータの利用経験の乏しい人は毎回のキーボード入力時間が極端に変化してしまうため, 実用に耐えうるしきいを設定することが事実上不可能といえる. これは同じパスワードを繰り返し入力していけば慣れてくるはずなので解決は比較的容易であろう.

重大なのはタイピングパターンの標本が少ないという問題である. 今回タイピングパターンデータとして取得できたのは, 筆者の個人的な関係者から高々十数名である. データ取得用のプログラムの動作環境が若干特殊であったために手元の環境でしか測定できなかったためである. このため大人数における耐性を示すことが困難であり, キーボードの使用に慣れた人の多くにタイピングパターンというものが表れているかどうかの確認が困難となった. コンピュータ使用人口という母集団が非常に大きいため, さらに多くの標本を用意してタイピングパターンそのものについて検証する必要があるわけである.

そして, ニューラルネットワークの特徴である「学習」という操作が認証システムにおいては欠点となる恐れがある. 登録すればすぐに利用できるパスワードによる認証システムと違って,

1. 前もってタイピングパターンを測定する
2. それを用いてニューラルネットワークを学習させる

の２段階の操作が操作が必要であり, ユーザインターフェイスとしては使いにくいシステムといえる.

学習にかかる時間は今回用いたシステムでは最長で５分程度であったが,CPU の処理速度は指数関数的に向上しており, 今後十分解決可能な問題である. しかしタイピングパターンの測定に数回, 同じ文字列の入力を求める必要があり, これがユーザ側にとっては面倒なものである. 実用化に向けては, 毎回の login 時にデータを取得し段階的にシステムに組み込むなどの工夫が必要であろう.

5.2 今後の課題

今回はニューラルネットワークを利用した認証実験を行なったわけだが, これはあくまで一つの方法でしかない.

例えば統計学的にタイピングパターンというものを解析し, 未知の入力に対して統計的検定による認証を行なう方法, タイピングパターンを関数としてとらえ, フーリエ変換やウェーブレット変換を行なうことによって解析を行なう方法など, いろいろな応用が考えられる.

また今回はタイピングパターンに対するノイズの可能性を考えネットワーク越しの遠隔操作による入力は禁止したが, こういったノイズに対する耐性ができればInternet 社会にも対応でき, 便利になるであろう.

参考文献

[1] 佐藤 宏介, 土居 元紀. “自分がパスワード”. 電子情報通信学会誌 Vol.82,No.4, 1999

[2] Judith Dayhoff, 桂井 浩 訳. “ニューラルネットワークアーキテクチャ入門”. 森北出版, 1992

[3] 合原 一幸. “ニューラルコンピュータ”. 東京電機大学出版局, 1988

[4] Dwayne Phillips. *The Backpropagation Neural Network*. C/C++ Users journal, Vol.14, No.1
和訳:太田純.“逆伝搬型ニューラルネット”. C MAGAZINE,Vol.8,No.6

[5] 馬場 則夫, 小島 史男, 小澤 誠一. “ニューラルネットの基礎と応用”. 共立出版, 1994

[6] 森田 邦明, 羽賀 隆洋. “タイピングパターンを使った認証”. 愛知工業大学研究報告 Vol.32, 1997

（受理　平成12年３月18日）