Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação

# Autenticação Biométrica via Teclado Numérico Baseada na Dinâmica da Digitação: Experimentos e Resultados

**Autor: Carlos Roberto do Nascimento Costa**
**Orientador: Prof. Dr. João Baptista Tadanobu Yabu-uti**
**Co-orientador: Prof. Dr. Lee Luan Ling**

**Dissertação de Mestrado** apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: **Telecomunicações e Telemática**.

Banca Examinadora

João Baptista Tadanobu Yabu-uti, Dr. ..... DECOM/FEEC/Unicamp
Luiz César Martini, Dr. ..... DECOM/FEEC/Unicamp
Miguel Gustavo Lizarrága, Dr. ... Samsung Inst. de Desenv. para Informática
Yuzo Iano, Dr. ..... DECOM/FEEC/Unicamp

Campinas, SP
Janeiro/2006

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

C823a

Costa, Carlos Roberto do Nascimento
Autenticação biométrica via teclado numérico baseada na dinâmica da digitação: experimentos e resultados / Carlos Roberto do Nascimento Costa. --Campinas, SP: [s.n.], 2006.

Orientadores: João Baptista Tadanobu Yabu-uti, Lee Luan Ling
Dissertação (Mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Processamento de sinais. 2. Reconhecimento de padrões. 3. Computadores – Medidas de segurança. 4. Biometria. 5. Computadores – Controle de acesso. I. Yabu-uti, João Baptista Tadanobu. II. Lee, Luan Ling. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Título em Inglês: Biometric authentication through numerical keyboard based on keystroke dynamics: experiments and results
Palavras-chave em Inglês: Digital signal processing, Pattern recognition, Security, Biometrics, Keystroke dynamics
Área de concentração: Telecomunicações e Telemática
Titulação: Mestre em Engenharia Elétrica
Banca examinadora: Luiz César Martín, Miguel Gustavo Lizarrága, Yuzo Iano
Data da defesa: 26/01/2006

# Resumo

Este trabalho apresenta uma nova abordagem para autenticação biométrica de usuários baseada em seu ritmo de digitação em teclados numéricos. A metodologia proposta é de baixo custo, não-intrusiva e pode ser aplicada tanto a um mecanismo de *login* em controle de acesso a áreas restritas como na melhoria do nível de segurança em transações bancárias. Inicialmente, o usuário indica a conta a ser acessada por meio de uma cadeia de caracteres digitada que é monitorada em tempo real pelo sistema. Simultaneamente, são capturados os tempos de pressionamento e soltura das teclas. Quatro características são extraídas do sinal: Código ASCII (*American Standard Code for Information Interchange*) da tecla, duas latências e uma duração associada com a tecla. Alguns experimentos foram feitos usando amostras reais de usuários autênticos e impostores e um classificador de padrões baseado na estimação da máxima verossimilhança. Alguns aspectos experimentais foram analisados para verificar os seus impactos nos resultados. Estes aspectos são as características extraídas do sinal, a informação alvo, o conjunto de treinamento usado na obtenção dos modelos dos usuários, a precisão do tempo de captura das entradas, o mecanismo de adaptação do modelo e, finalmente, a técnica de obtenção do limiar ótimo para cada usuário. Esta nova abordagem traz melhorias ao processo de autenticação pois permite que a senha não seja mais segredo, assim como oferece uma opção para autenticação biométrica em dispositivos móveis, como celulares.

**Palavras-chave**: Processamento digital de sinais, reconhecimento de padrões, segurança, biometria, dinâmica da digitação.

# Abstract

This work presents a new approach for biometric user authentication based on keystroke dynamics in numerical keyboards. The methodology proposed is low cost, unintrusive and could be applied in a login mechanism of access control to restricted area and/or to improve the security level in Automatic Teller Machines (*ATM*). Initially, the user indicates the account to be accessed by typing the target string that is monitored in real time by the system. Simultaneously, the times of key pressed and key released are captured. Four features are extracted from this input: The key ASCII code, two associated latencies and key durations, and some experiments using samples for genuines and impostors users were performed using a pattern classification technique based on the maximum likelihood estimation. Some experimental aspects had been analyzed to verify its impacts in the results. These aspects are the sets of features extracted from the signal, the set of training samples used to obtain the models, the time precisions where captures the inputs, the adaptation mechanism of the model and, finally, the technique to attainment of the excellent threshold for each user. This new approach brings improvements to the process of user authentication since it allows the password not to be a secret anymore, as well as it allows to include biometric authentication in mobile devices, such as cell phones.

**Keywords**: Digital signal processing, pattern recognition, security, biometrics, keystroke dynamics.

*Dedico este trabalho a rainha do meu mundo,* ***Gelly****. Você torna minha jornada melhor a cada dia. Amo você!*

*"Tem-se aprendido consideravelmente, nos últimos trinta anos, sobre a natureza dos labirintos que representam tarefas vulgares de resolução de problemas dos seres humanos - demonstrar teoremas, resolver puzzles, jogar xadrez, fazer investimentos, equilibrar cadeias de montagem, para mencionar algumas. Tudo o que aprendemos acerca destes labirintos aponta para a mesma conclusão: que a resolução humana de problemas, da mais confusa a mais intuitiva, não envolve mais do que mistura variável de tentativa-e-erro e de seletividade".*

**Herbert A. Simon (1916-2001)**

# Agradecimentos

Em primeiro lugar agradeço aos meus familiares pelo exemplo de estabilidade e harmonia onde cresci. Agradeço também a minha tia, Celina Barbosa do Nascimento, que aos 87 anos continua cercando minha vida de espirituosidade e amor. Em especial agradeço a minha mãe, Aléa Tavares do Nascimento. Seus exemplos de luta, simplicidade e honestidade vestiram-me de uma doutrina que acompanhará minha existência. Para sempre irei amá-la e idolatrá-la, e para meus filhos irei reproduzir com muito orgulho seus ensinamentos.

Em segundo lugar, é com um imenso carinho que expresso minha gratidão aos meus orientadores, Dr. João Baptista T. Yabu-uti e Dr. Lee Luan Ling. Muito obrigado por proporcionarem a oportunidade de crescimento, por contribuirem com experiências de vida e por serem compreensivos e honrados, o que os tornam para mim mais que orientadores: Amigos.

Deixo aqui meus agradecimentos a todos os funcionários da FEEC. Em especial aos professores que tive o privilégio de participar dos seus cursos, Prof. Dr. Yuzo Iano, Prof. Dr. Roberto de A. Lotufo e Prof. Dr. Ivan L. M. Ricarte.

Agradeço também aos amigos Ricardo N. Rodrigues e Glauco Yared por serem cúmplices na resolução deste problema e pelos momentos de alegria que passamos durante a jornada. E a todos os novos amigos que fiz no LRPRC e no DECOM, deixo aqui minha mensagem de igual gratidão.

Ao amigo Miguel por suas significativas e valorosas considerações durante a minha jornada. Este que foi um amigo de verdade, sempre disponível, receptivo e alegre. Como um monge, me ensinou a encarar de frente os caminhos tortuosos da vida acadêmica com muita simplicidade e sabedoria. Seus ensinamentos sobre como devemos abordar os problemas usando sempre técnicas simples, levarei comigo enquanto existir - pois a vida é simples. A você dedico meus mais sinceros agradecimentos.

Expresso também meus agradecimentos a CAPES pelo apoio financeiro e por acreditarem que o Brasil cresce junto com o crescimento técnico e científico de sua população.

Gostaria de reservar este espaço final para expressar a minha gratidão às inúmeras pessoas que foram solidárias comigo, e de alguma forma puderam contribuir para a realização deste trabalho. Em vários momentos tive meus inevitáveis pensamentos pessimistas sobre o resultado que obteria. E em todos esses momentos também tive em memória uma palavra ou um gesto de incentivo que me mantiveram determinado nesta realização. Se por algum motivo seu nome não está listado aqui, lembre-se que na minha memória seus gestos estarão guardados durante toda minha existência. Muito Obrigado!

# Sumário

# Lista de Figuras

# Lista de Tabelas

# Lista de Símbolos e Abreviaturas

| | | |
|---|---|---|
| ASCII | - | *American Standard Code for Information Interchange* |
| PIN | - | *Personal Identification Number*, número de identificação pessoal |
| FAR | - | *False Acceptance Rate*, taxa de falsa aceitação |
| FRR | - | *False Rejection Rate*, taxa de falsa rejeição |
| EER | - | *Equal Error Rate*, taxa de erro igual |
| ZeroFAR | - | *Zero False Acceptance Rate*, taxa de falsa aceitação, quando a falsa rejeição é zero |
| ZeroFRR | - | *Zero False Rejection Rate*, taxa de falsa rejeição, quando a falsa aceitação é zero |
| SVM | - | *Support Vector Machines*, máquinas de vetor suporte |
| $w$ | - | Amostra do usuário |
| $a$ | - | Conta do usuário |
| $c$ | - | Conjunto de contas armazenadas |
| $ia_w$ | - | Informação alvo da conta $w$ |
| $S_w$ | - | Conjunto de amostras usadas no treinamento da conta $w$ |
| $L_w$ | - | Conjunto de amostras adquiridas durante a autenticação da conta $w$ |
| MHZ | - | Megahertz, ou um milhão de ciclos por segundo |
| PS | - | Pressiona-Solta |
| SP | - | Solta-Pressiona |
| PP | - | Pressiona-Pressiona |
| $n$ | - | Número de caracteres em cada amostra |
| $K_i(a, w)$ | - | Conjunto de características para a amostra $w$ da conta $a$ |
| $C_a$ | - | Conjunto de códigos ASCII contidos no modelo da conta $a$ |
| $C_{a,w}$ | - | Conjunto de códigos ASCII da amostra $w$ para a conta $a$ |
| $CT$ | - | Vetor de características contando os códigos ASCII |
| $Z_w$ | - | Modelo do usuário da conta $w$ |
| $X$ | - | Vetor de características |
| $\mu$ | - | Vetor média |
| $\sigma$ | - | Desvio padrão |
| $o$ | - | Outliers |
| $f_x(x)$ | - | Função densidade de probabilidade |
| $Q$ | - | Vetor contendo os valores de densidade de probabilidade das características |
| $L$ | - | Verossimilhança |
| $\tau(w)$ | - | Limiar pré-definido para a conta $w$ |
| ROC | - | *Receive Operating Curve*, um gráfico que apresenta a falsa versus a falsa rejeição para diferentes valores de limiar |
| FEEC | - | Faculdade de Engenharia Elétrica e de Computação |
| UNICAMP | - | Universidade Estadual de Campinas |
| $N$ | - | Número de características em cada experimento |
| $k$ | - | Número de características em cada experimento |
| $S.N.M$ | - | Representação de um número de pont fixo |

# Trabalhos Publicados Pelo Autor

Durante a elaboração deste trabalho, algumas publicações foram alcançadas:

1. C.R. do N. Costa, G.F.G. Yared, R.N. Rodrigues, J.B.T. Yabu-uti, F. Violaro, L.L. Ling. **Autenticação Biométrica via Dinâmica da Digitação em Teclados Numéricos**. *Anais do XXII Simpósio Brasileiro de Telecomunicações* - SBrT'05, Campinas, São Paulo, Brasil, 04-08 de Setembro de 2005.

2. R.N. Rodrigues, G.F.G. Yared, C.R. do N. Costa, J.B.T. Yabu-uti, F. Violaro, L.L. Ling. **Biometric Access Control through Numerical Keyboard based on Keystroke Dynamics**. *Proceedings of the Second International Conference on Biometric*, ICB'2006, Hong Kong, China, 5-7 January 2006.

3. C.R. do N. Costa, L.L. Ling e J.B.T. Yabu-uti. **Autenticação Biométrica via Teclado Numérico Baseada na Dinâmica da Digitação**. *IEEE Latin America Transactions*, **(EM PROCESSO DE REVISÃO)**.

A primeira já foi apresentada e publicada em setembro de 2005, enquanto a segunda publicação foi alcançada em janeiro de 2006. A terceira publicação foi submetida e encontra-se em processo de revisão.

# Capítulo 1

# Introdução

Com o aumento da complexidade e sofisticação da sociedade, as pessoas passam cada vez mais por situações em que são obrigadas a ter que provar sua identidade. Esse fato é tão comum, que estamos acostumados a fazê-lo como se isso fosse algo totalmente normal. Assim sendo, é perfeitamente aceitável utilizar cartões, rubricas, senhas ou documentos para atestar nossa identidade. Podemos citar como exemplos, os cartões magnéticos para retirar dinheiro em bancos, assinaturas para autenticar cheques bancários, senhas para identificação em controle de acesso ou ainda a apresentação de documentos com nossa foto, como passaportes e carteiras de identidade.

O propósito de tais procedimentos é oferecer uma evidência adicional para autenticar o pedido de identidade, ou seja, auxiliar na confirmação de que nós realmente somos quem dizemos ser. Fica visto pelos exemplos apresentados anteriormente que isso pode ser feito de várias formas diferentes.

Existem basicamente quatro maneiras de autenticar uma pessoa. A primeira maneira é a de possuir fisicamente um dispositivo que em si seja a autenticação, como por exemplo, um cartão válido ou um crachá que permite o acesso a algum lugar. A segunda maneira é a de ter acesso a chaves baseadas no seu conhecimento, como por exemplo senhas e contra-senhas. A terceira opção é a validação de identidade através de um padrão ou atividade específica do indivíduo, como por exemplo sua assinatura ou fala. A quarta maneira, é a análise das possíveis características físicas que a pessoa possui, dentre as quais podemos mencionar as impressões digitais, geometria da mão, íris, dentre outras. Cada uma dessas abordagens de validação de identidade estão sujeitas a um maior ou menor sucesso, dependendo do tipo de aplicação e situação em que se deseja empregá-las.

Na maioria das vezes, a maneira de autenticar a identidade de um indivíduo recai sobre as duas primeiras categorias. É muito comum que hoje em dia seja preciso memorizar mais de dez números de identificação incluindo senha do cartão do banco ou login do computador, número de RG, de passaporte, de CPF e vários outros. Muitas vezes é necessário que o indivíduo carregue consigo vários documentos de identificação como, por exemplo, carteira de identidade cartões e chaves. Porém, ne-

nhum desses métodos são 100% confiáveis, visto que podem ser esquecidos, roubados, emprestados, perdidos, copiados ou falsificados.

Por essas razões, tem aumentado o interesse em desenvolver métodos de verificação de identidade pessoal que levem em consideração estratégias que se fundamentem na terceira e quarta categorias. Essas técnicas se baseiam em medidas biométricas, onde "medida biométrica" é definida pela *International Association for Biometrics* [1] como "*A medida de atributos/características físicas ou de comportamento de uma pessoa com o objetivo distinguí-la dentre as demais*".

As medidas biométricas podem ser divididas em dois grupos. O primeiro, a biometria fisiológica, que engloba características tais como o padrão da íris, as impressões digitais, a forma do contorno da mão e face. E o segundo, a biometria de comportamento, que se preocupa com a extração de características mais sutis, como o tipo de escrita do indivíduo, a maneira como assina seu nome, a forma como pronuncia determinadas palavras, o rítmo com que digita uma dada senha.

Embora antigamente muitos dispositivos tivessem sido apresentados como capazes de captar características biométricas com boa precisão, eles raramente satisfaziam as expectativas. Mais recentemente, a evolução da pesquisa na área da biometria em conjunto com o desenvolvimento da tecnologia, tanto na aquisição de informação como no aumento do poder de processamento dos computadores, ofereceram um novo impulso para a crescente utilização da biometria como método de autenticação.

Outro aspecto a ser considerado nasce da necessidade de controlar o acesso a áreas restritas, que torna-se cada vez mais importante nos dias de hoje. O mecanismo mais conhecido e usual para se ter maior segurança em controle de acesso dá-se através da autenticação do usuário por uma senha. Porém este tipo de mecanismo é frágil pois existem usuários descuidados que comprometem a segurança quando se utilizam de senhas frágeis e de contexto normalmente familiar, como por exemplo uma data de nascimento. Por outro lado, o custo e a simplicidade deste tipo de mecanismo clássico de segurança justificam a sua adoção, e em várias situações permanece como mecanismo principal ao lado de outras políticas de segurança como, por exemplo, cartões de identificação.

O propósito deste trabalho é melhorar o processo de autenticação por senha usando características biométricas extraídas do rítmo com que o usuário digita sua senha. A inclusão de características biométricas em sistemas de autenticação pessoal aumenta o grau de confiança dos usuários na segurança do sistema, pois características biométricas são únicas para cada pessoa e não podem ser roubadas, perdidas ou esquecidas [2].

A tecnologia biométrica tratada neste trabalho é conhecida como biometria da digitação ou dinâmica da digitação. A biometria da digitação é o processo de analisar o rítmo com que o usuário digita em um terminal através do monitoramento das entradas do teclado e seus respectivos intervalos de tempo durante as tentativas de identificação. Autenticação por dinâmica da digitação pode ser classi-

ficada em estática ou contínua. A abordagem estática analisa as entradas em um momento particular, como por exemplo o momento em que o usuário digita a sua senha. Enquanto que na abordagem contínua são analisadas todas as entradas no teclado durante a sessão do usuário. Analise da dinâmica da digitação é um processo simples que necessita apenas que o computador possua um teclado convencional acoplado a ele.

Este trabalho inova na forma como se usa a dinâmica da digitação, pois utiliza somente um teclado numérico - pouco abordado nas pesquisas dessa área - para observar a dinâmica com que o usuário digita uma senha exclusivamente numérica. O uso do teclado numérico aumenta o intervalo de tempo entre teclas e diminue ainda mais a exigência de precisão [3], além de praticamente forçar o usuário a usar somente uma das mãos. Além disso, permite que sejam propostos soluções para funcionar em telefones celulares, em sistemas de caixa automático para bancos ou mesmo no acesso a áreas restritas. A metodologia adotada neste trabalho tem um baixo custo de processamento, é não-intrusiva e verifica o usuário de maneira estática, ou seja, apenas considera a entrada digitada em um dado momento.

O trabalho aqui apresentado aborda basicamente duas áreas de conhecimento. A primeira área é a de processamento digital de sinais, visto a necessidade de implementar técnicas para capturar o sinal representando o rítmo do usuário ao digitar.

A segunda, e mais importante, é a área de reconhecimento de padrões, que compreende os métodos para classificação de elementos num conjunto de dados, com base em suas características. Entende-se por elemento, o objeto que é observado e cujas propriedades medidas constituem suas características ou padrões de medida.

Reconhecimento de padrões é o estudo de como as máquinas podem observar o meio, aprender e distinguir padrões de interesse neste meio e serem capazes de tomar decisões corretas sobre as categorias a que pertencem tais padrões [4]. No estudo da biometria, que é uma das aplicações de reconhecimento de padrões, o padrão poderia ser uma impressão digital, uma assinatura ou a dinâmica do digitador.

O problema de reconhecimento de padrões está relacionado com as tarefas de classificação ou categorização, onde as classes podem ser definidas a "*priori*"pelo projetista do sistema (classificação supervisionada) ou podem estar baseadas em um aprendizado feito sobre a similaridade dos padrões (no caso de classificação não supervisionada).

De maneira geral, um problema de reconhecimento de padrões bem definido e suficientemente delimitado apresentará pequenas variações intraclasse (elementos de uma mesma classe) e grandes variações interclasses (elementos de classes diferentes), levando-nos a uma representação compacta do padrão e uma estratégia de decisão simples.

Vale salientar que, uma das maneiras de fazer com que o sistema conheça as classes em que terá

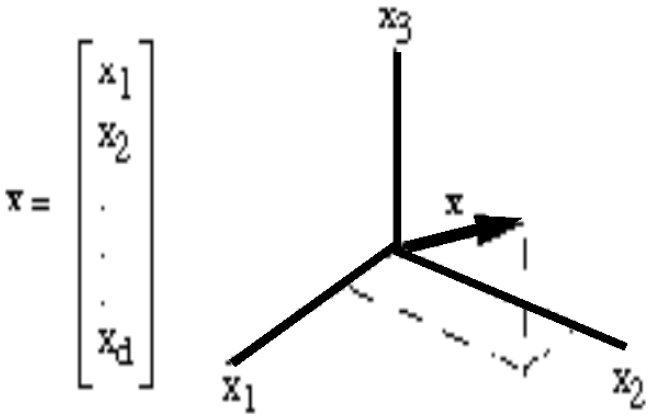


Fig. 1.1: Vetor de características $X$ e sua representação no espaço tri-dimensional.

que classificar os padrões de entrada é apresentar ao sistema um conjunto de exemplos (amostras), também chamado de conjunto de treinamento. A partir deste conjunto, o sistema poderá delimitar o espaço de características a que pertence os padrões.

Historicamente, tem-se utilizado quatro tipos de técnicas para resolver os problemas gerais de reconhecimento de padrões [5]:

- **Reconhecimento Estatístico**: As características são da forma de n-tuplas ou vetores, sendo utilizadas regras de decisão, teoria de probabilidades, funções discriminantes e outros procedimentos estocásticos. Esse é o tipo de reconhecimento mais tradicional.

- **Reconhecimento Estrutural/Sintático**: As características são da forma de sentenças de uma linguagem reconhecida por uma gramática de estrutura de frases. Reconhecimento sintático é também conhecido como reconhecimento estrutural de padrões, onde as características estruturais dos elementos, em termos de suas partes constituintes, propriedades e relacionamentos, são representadas sintaticamente.

- **Reconhecimento via Lógica Nebulosa (*Fuzzy*)**: Nesta abordagem é utilizada a lógica com múltiplos valores para modelar problemas que tratam com dados ambíguos. Os classificadores nebulosos tratam os padrões em questão de graus de pertinência a uma determinada classe, sendo uma generalização da lógica tradicional, que declara que qualquer premissa é verdadeira ou falsa, não ambas.

- **Reconhecimento via Redes Neurais**: Alguns autores consideram o reconhecimento via redes neurais como um tipo particular de reconhecimento estatístico, uma vez que as características também são da forma de n-tuplas ou vetores e existe uma equivalência entre certos modelos de

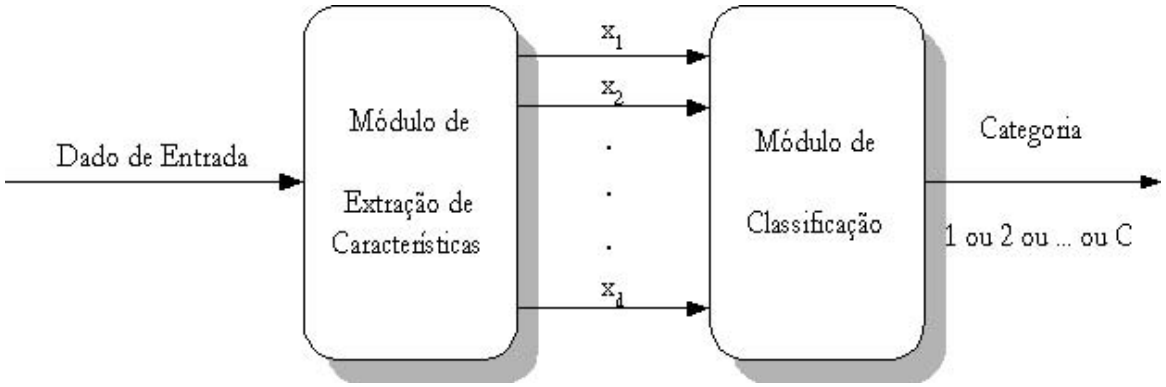


Fig. 1.2: Módulos de extração das características e classificação.

redes neurais com técnicas estocásticas fundamentais. Por possuirem propriedades peculiares, tais como a capacidade de generalização, abstração, aprendizagem a partir de exemplos, o reconhecimento por redes neurais acaba sendo tratado como uma área distinta.

É possível estabelecer uma fronteira entre as áreas de processamento digital de sinais e reconhecimento de padrões. Pode-se dizer que a primeira lida principalmente com operações sobre sinais cujo objetivo é melhorar sua qualidade de alguma forma, ou enfatizar características de importância particular. Já a segunda, trata da identificação, verificação ou interpretação de sinais através da extração e classificação de informações (em nível de abstração mais alto) a respeito do que o sinal denota. O reconhecimento de padrões por máquinas envolve técnicas para atribuição dos padrões a suas respectivas classes automaticamente, e com a mínima intervenção humana possível.

De maneira geral, pode-se dizer que as técnicas de processamento digital de sinais servem aos sistemas de reconhecimento de padrões como um passo essêncial na extração ou enfatização de características. Na figura 1.2 apresenta-se o diagrama de blocos de um sistema de reconhecimento de padrões simplificado, onde o módulo de extração de características processa a informação de entrada com o objetivo de determinar valores numéricos para o conjunto de $d$ características que compõem o vetor de características $X$ da figura 1.1. Pode-se pensar em $X$ como sendo um vetor coluna de dimensão $d$ composto de $x_1, x_2, \ldots, x_d$. De forma análoga, pode-se imaginar que $X$ representa um ponto em um espaço de características $d$-dimensional, como por exemplo um espaço tri-dimensional ($d = 3$). O módulo de classificação recebe $X$ e associa-o a uma de suas $C$ classes: $classe_1, classe_2, \ldots, classe_C$.

A implementação dos módulos de extração das características é dependente do problema. Um módulo extrator de características ideal deveria produzir o mesmo vetor de características $X$ para todos os padrões que pertencem à mesma classe, e diferentes vetores de características para padrões de classes diferentes.

Na prática, dados de entrada diferentes no módulo de extração de características produzem dife-

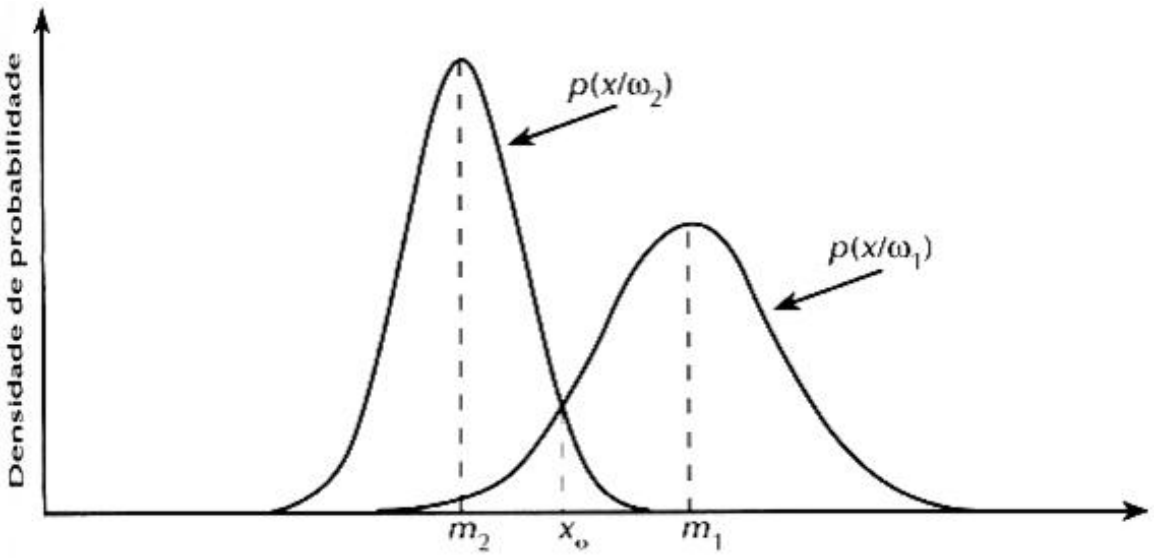


Fig. 1.3: Funções densidade de probabilidade para duas classes de padrões unidimensionais. O ponto $x_0$ é a fronteira de decisão se as duas classes tiverem a mesma probabilidade de ocorrer.

rentes vetores de características, porém, espera-se que a variabilidade intraclasse seja pequena.

Este trabalho baseou-se em técnicas de reconhecimento estatísticos de padrões, ou seja, o módulo de classificação trata o problema através de uma abordagem probabilística.

Como na maioria das áreas que envolvem a medida e interpretação de eventos físicos, considerações de probabilidade tornam-se importantes em reconhecimento de padrões, devido à aleatoriedade na qual as classes de padrões estão envolvidas. Além disso, uma vez que o conjunto de características seja suficientemente discriminante, é possível derivar uma abordagem de classificação que seja ótima no sentido que, na média, seu uso leve à menor probabilidade de erros de classificação.

Considerando-se inicialmente um problema $1D$ ($d$ = 1) envolvendo duas classes $w$ de padrões ($M$ = 2) governadas por densidades gaussianas, com médias $m_1$ e $m_2$ e desvios padrão $\sigma_1$ e $\sigma_2$, respectivamente. Percebe-se, a partir da equação 1.1, que as funções de decisão possuem a forma:

$$f_j(x) = p(x|w_j)P(w_j) = \frac{1}{\sqrt{2\pi}\sigma_j} exp\left(-\frac{(x-\mu_j)^2}{2\sigma_j^2}\right) P(w_j) \tag{1.1}$$

onde $j$=1, 2. Os padrões são, neste caso, valores escalares contidos em $X$ e denotados por $x_j$. A figura 1.3 mostra o gráfico das funções densidade de probabilidade para as duas classes. A fronteira entre as duas classes é um ponto, denotado por $x_0$, tal que $d_1(x_0)$ = $d_2(x_0)$. Se as duas classes tiverem a mesma probabilidade de ocorrer, $P(w_1)$ = $P(w_2)$ = $1/2$, e a fronteira de decisão é o valor de $x_0$ para o qual $P(x_0|w_1)$ = $P(x_0|w_2)$. Esse ponto é a interseção das duas funções densidade de probabilidade, como mostrado na figura 1.3. Qualquer padrão (ponto) à direita de $x_0$ é classificado como sendo da

classe $w_1$. Similarmente, qualquer padrão à esquerda de $x_0$ é classificado como sendo da classe $w_2$. Quando as classes não tiverem a mesma probabilidade de ocorrer, $x_0$ move à esquerda se a classe $w_1$ tiver maior probabilidade de ocorrer ou, por outro lado, para a direita se $w_2$ tiver maior probabilidade de ocorrer. Esse resultado era esperado, uma vez que o classificador está tentando minimizar as perdas da classificação errada. Por exemplo, no caso extremo, se a classe $w_2$ nunca ocorreu, o classificador não deveria nunca cometer um erro, atribuindo sempre os padrões à $w_1$ (ou seja, $x_0$ deveria se mover para infinito negativo).

Porém, na prática, o módulo de extração das características estima $d$ amostras para o vetor de características $X$, tornando impraticável ao sistema encontrar os pontos de fronteiras de decisão para cada valor das $d$-dimensões do vetor de características. Não obstante, este tipo de problema é comumente encontrado em sistemas de reconhecimento de padrões. Mas como contornar esta situação? A maneira mais simples e usual para contornar esta situação é estimando um parâmetro que maximize a probabilidade de um vetor de características $X$ de $d$-dimensões pertencer a uma dada classe.

A estimação de parâmetros é um problema clássico em estatística, e pode ser abordado de diversas maneiras. Este trabalho considera somente o procedimento para estimação conhecido como máxima verossimilhança [6]. Estimação da máxima verossimilhança é o método que vê os parâmetros como quantidades cujos valores são fixos, porém desconhecidos. A melhor estimativa é definida como aquela que maximize as probabilidades de obter realmente as amostras observadas.

Suponha-se que um conjunto de amostras $X$ =[$x_1, x_2, \ldots, x_d$] de $d$-dimensões pertencem a uma determinada classe $w$. Cada elemento de $X$ possui seu valor médio e sua variância estimados a partir de um conjunto de amostras reservadas para esse propósito - um conjunto de treinamento. Cada uma das $X_w$ amostras é independente e regida de acordo com a lei das probabilidades $p(X|w_j)$. Estimar o parâmetro de máxima verossimilhança é estimar o valor de $\theta$ que maximize a equação 1.2:

$$L(X|\theta) = \prod_{k=1}^{n} p(X|w_j) \tag{1.2}$$

Visto em função de $\theta$, $L(X|\theta)$ é conhecido como a verossimilhança de $\theta$ em relação ao conjunto de amostras. A estimação da máxima verossimilhança $\theta$ em relação à $w$ é, por definição [6], o valor que maximiza $p(X|w_j)$. Intuitivamente, corresponde ao valor de $\theta$ que em algum sentido concorda melhor com as amostras atualmente observadas.

Desta forma, um conjunto de características, extraídas pelo módulo de extração de características, pertencerá a uma classe se o módulo de reconhecimento de padrões decidir se o valor de máxima verossimilhança entre o conjunto de características e o modelo do usuário (média e variância) é válido para essa determinada classe.

Neste trabalho a forma de atestar a validade dessa decisão é feita a partir de um limiar otimizado para cada usuário (classe) do sistema. O limiar é estimado usando o modelo do usuário, e será

explicado no capítulo que trata em detalhes a metodologia apresentada.

## 1.1 Objetivos

O primeiro objetivo desta dissertação é desenvolver uma metodologia para autenticação de usuários combinando *string* alvo (senha do usuário) com a dinâmica da digitação, e que possa ser aplicada em controle de acesso a áreas restritas ou em dispositivos celulares. Foi adotado um classificador baseado na Máxima Verossimilhança entre a amostra de entrada e sua respectiva classe, onde a probabilidade de uma amostra pertencer a um determinado usuário é obtida a partir de técnicas baseadas na função densidade de probabilidade de cada característica com média e variância estimadas a partir de amostras coletada durante o cadastramento [4]. As vantagens desta metodologia são enumeradas a seguir:

1. Maior segurança: a nova abordagem agregará mais segurança ao usual mecanismo de identificação presente em controle de acesso e em dispositivos móveis, pois poderão somar as vantagens das características biométricas inerentes aos usuários com o mecanismo usual de controle e segurança por senha;

2. Baixo custo: utilizará apenas um teclado numérico convencional para aquisição das informações, já presente na maioria das tecnologias de controle de acesso a áreas restritas e celulares. Esta vantagem não é comum entre as tecnologias biométricas atualmente, que geralmente fazem uso de dispositivos caros para aquisição de dados [**?**];

3. Não-intrusivo: a aquisição de amostras necessita apenas que o usuário digite sua *string* alvo, processo com o qual já está habituado pois, a muito tempo é prática comum em sistema de segurança da informação. Para algumas tecnologias biométricas, a aquisição de amostras é intrusiva, como a da íris onde uma varredura do olho é feita utilizando uma feixe de luz, fazendo com que muitos dos usuários sintam-se desconfortáveis [2].

O segundo objetivo é discutir um conjunto de características que possam ser aplicadas na autenticação biométrica via dinâmica da digitação em teclados numéricos. Todas as características avaliadas são conhecidas pelos pesquisadores na áreas, mas até então não haviam sido avaliadas no desempenho perante os teclados numéricos.

E, finalmente, o terceiro objetivo é viabilizar a metodologia para que possa ser aplicada na prática, e que seus resultados sejam competitivos. Portanto, todos os experimentos foram feitos com amostras de usuários e impostores reais e, como forma de simular uma aplicação em sistema embarcados, foram feitos experimentos usando aritmética de ponto fixo [7], afim de avaliar se compromete a precisão do sistema.

## 1.2 Estrutura da Dissertação

Com o intuito de fornecer uma apresentação estruturada da dissertação, favorecendo uma sequência lógica de idéias, o restante do trabalho está organizado como se segue:

- No capítulo 2 discute-se a área de Biometria, enfatizando aspectos sobre reconhecimento de características comportamentais e medidas de desempenho de sistemas biométricos.

- No capítulo 3 apresenta-se a biometria da digitação ou dinâmica da digitação, bem como é apresentado o estado da arte da Dinâmica da Digitação.

- No capítulo 4 discute-se a metodologia adotada para a extração das características e o classificador adotado no reconhecimento do digitador, apresentando suscintamente as contribuições deste trabalho.

- No capítulo 5 são apresentados os experimentos conduzidos e os resultados obtidos usando um conjunto de amostras de 26 digitadores coletadas durante um período.

- No capítulo 6 são apresentadas as conclusões sobre esta dissertação, revisando as contribuições propostas, bem como são discutidas as perspectivas para trabalhos futuros.

# Capítulo 2

# Biometria

Neste capítulo são apresentados os conceitos básicos que regem a biometria e sistemas biométricos de identificação pessoal. São apresentados também alguns métodos biométricos tais como impressões digitais, padrões dos olhos, geometria da mão, verificação de voz, padrões de face, assinaturas e dinâmica da digitação, detalhando as características em que se baseiam esses métodos para realizar a tarefa de autenticação.

## 2.1 O que é biometria?

A biometria é o ramo da ciência que estuda a mensuração dos seres vivos [8]. Tecnologias biométricas são definidas como [9] "métodos automáticos de verificação ou identificação de identidade de uma pessoa viva baseados em características fisiológicas ou de comportamento". Vamos examinar algumas das palavras chaves encontradas nessa definição:

- Métodos Automáticos: Dentro do contexto de um sistema automatizado, os componentes que servem de fundamento para implementação de um sistema biométrico são três: O primeiro componente é o mecanismo de captura de um sinal digital ou analógico de características de uma pessoa; o segundo componente é aquele que trata do processamento e classificação dos sinais; finalmente o terceiro componente é a interface homem/máquina que permite ao usuário fazer a entrada de dados no sistema para que se realize a tarefa de verificação/identificação automática. O termo "automática", demanda que uma vez feita a captura do sinal, os processos que envolvem o processamento, classificação e finalmente o resultado da identificação, sejam feitos sem intervenção humana.

- Verificação versus Identificação: Um sistema automático baseado em características biométricas pode ser classificado com relação à maneira como seus dados de entrada são classificados

junto à base de dados. Nesse caso, duas categorias podem ser definidas: Sistemas um para um e sistemas um para muitos. Um sistema um para um compara a informação biométrica apresentada por um indivíduo com a informação biométrica armazenada em uma base de dados correspondente àquele indivíduo. Nesse caso, o sistema decide se existe um casamento (*matching*) entre a informação de entrada e a armazenada na base de dados. Esse tipo de sistema é chamado também de sistema de verificação. Em contrapartida, um sistema um para muitos compara a informação biométrica apresentada por um indivíduo com toda a informação biométrica armazenada na base de dados, isto é, informações de todos os indivíduos (ou determinado conjunto deles), e declara se existe um casamento com algum deles ou não. Esse tipo de sistema é também chamado de sistema de identificação.

- Pessoa viva: Inicialmente, a interpretação desse termo parece bastante óbvia, porém é importante no contexto da definição de tecnologias biométricas. Pode ocorrer, por exemplo, que diante de um sistema de verificação de locutor, um indivíduo tente se passar por outro através da reprodução do som da voz de uma pessoa que tenha sido previamente gravada. Uma das soluções para esse tipo de fraude é que os dispositivos de captura que fazem parte dos sistemas biométricos incluam meios para determinar se existe uma característica "viva". Um exemplo disso já pode ser encontrado em alguns sistemas de reconhecimento de face. Nesse caso, o sensor que faz a captura da imagem não é uma câmara de vídeo comum e sim um dispositivo que além de capturar a imagem da face como uma matriz de valores de intensidade luz, capta também a distribuição de temperatura sobre as diferentes regiões do rosto. Dessa forma, ao se apresentar uma foto comum como entrada para o sistema, mesmo que as características referentes a intensidade de luz casem com as da base de dados, aquelas referentes à distribuição de temperatura com certeza serão diferentes e portanto o resultado do pedido de autenticação de identidade será falho.

- Características Fisiológicas e de Comportamento: O ponto final sobre a definição de tecnologias biométricas é a diferença entre características fisiológicas e características de comportamento. Uma característica fisiológica é uma propriedade física relativamente estável tal como as impressões digitais, geometria da mão, padrão da íris, padrão dos vasos sangüíneos do fundo dos olhos, entre outras. Esse tipo de característica é basicamente imutável. Por outro lado, uma característica de comportamento é mais um reflexo de atitudes psicológicas do indivíduo. A assinatura é a característica de comportamento mais utilizada para autenticação. Outros comportamentos que podem ser utilizados são a maneira como se digita nos teclados e a maneira de falar.

As características de comportamento tendem a variar com o tempo e, por esse motivo, muitos sistemas biométricos permitem que sejam feitas atualizações de seus dados biométricos de referência à medida que esses vão sendo utilizados [10]. Em geral, ao executar a tarefa de atualização de dados, o sistema terá se tornado mais eficiente em autenticar o indivíduo.

As diferenças entre métodos de comportamento e fisiológicos são importantes por vários motivos. Primeiro, o grau de variação intra-pessoal numa característica física é menor do que em uma característica de comportamento. Exemplificando, com exceção de algum ferimento, as impressões digitais são as mesmas ao longo da vida de indivíduo. Uma assinatura, por outro lado, é influenciada tanto por fatores fisicamente controláveis como por fatores emocionais. Assim, sistemas baseados em comportamento tem um grande trabalho em ajustar as variações intra-pessoais. Por esse motivo, é mais fácil construir um sistema que, por exemplo, guie o usuário a colocar a palma de sua mão sempre em determinada posição, do que implementar um algoritmo que traduza o estado emocional de uma pessoa. Tanto as técnicas de comportamento como as fisiológicas provêem níveis significativamente maiores de identificação e segurança do que aquelas baseadas em senhas e cartões de forma isolada.

## 2.2 Métodos Biométricos de Identificação

Em geral, a construção de sistemas de identificação pessoal se edifica sobre três pilares:

1. Possuir um dispositivo que seja em si a autenticação (um cartão),

2. Chaves baseadas no seu conhecimento (senhas),

3. Características biométricas (uma característica biométrica).

A partir desses três pilares é possível criar diferentes esquemas de identificação. Esses esquemas podem ser mais ou menos complexos dependendo de certas exigências, como por exemplo, o grau de segurança que se deseja alcançar, o valor do que se deseja proteger, a facilidade com que o usuário pode ter acesso ao sistema de identificação, o custo do sistema, entre outros.

A figura 2.1 apresenta uma representação desses três pilares exemplificando três esquemas de acesso a diferentes locais. No primeiro esquema, o acesso a determinado lugar é permitido simplesmente se apresentando um cartão magnético. No segundo esquema, o acesso é permitido através do conhecimento de uma senha e de possuir um cartão magnético. O terceiro caso é semelhante ao anterior, com a diferença de que o acesso só é permitido se for apresentada uma característica biométrica, a qual é inerente à pessoa que se deseja autenticar. No primeiro e no segundo esquema, tanto o cartão magnético, quanto o conhecimento da senha poderiam ser passados de uma pessoa para outra.

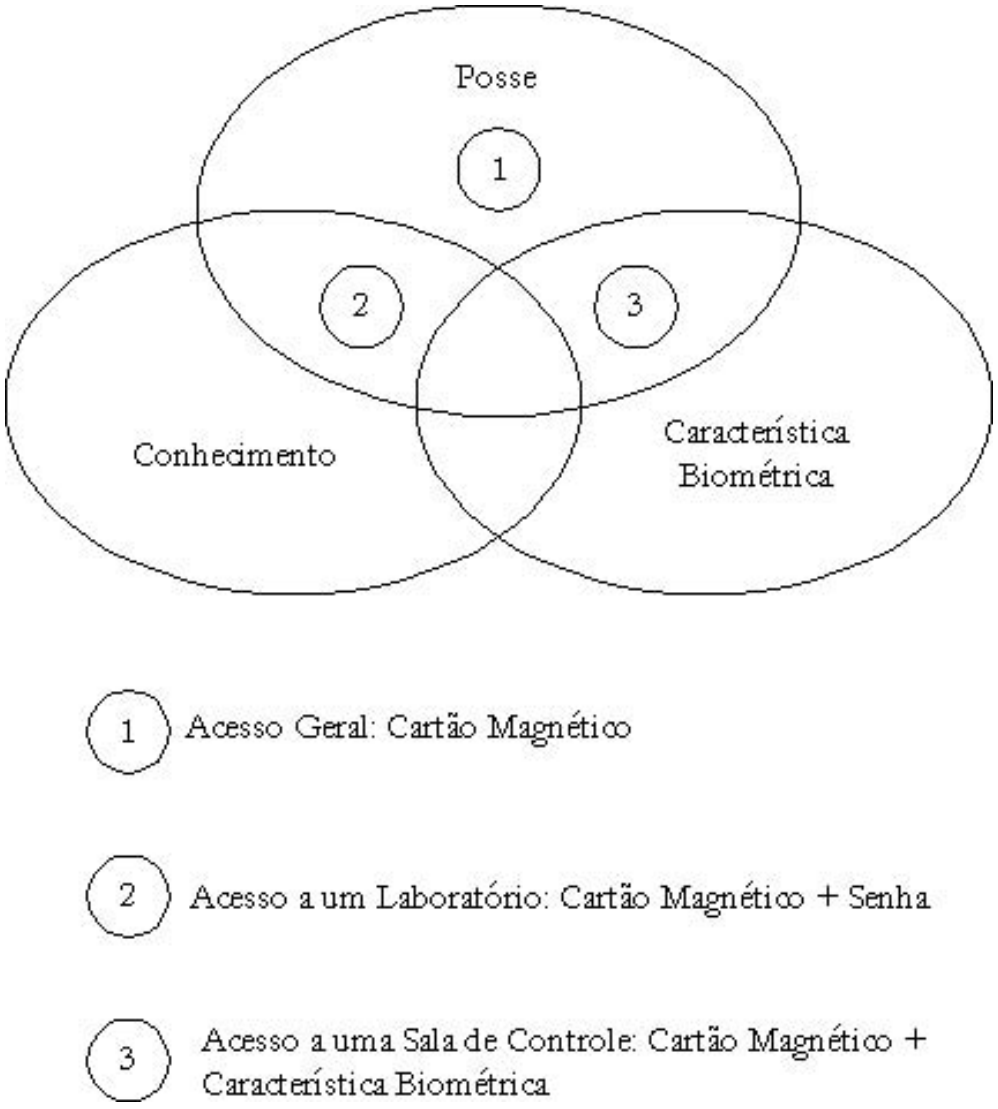


Fig. 2.1: Esquemas de acesso para identificação da identidade.

Porém, no último caso uma característica biométrica, por exemplo uma impressão digital, impõe que a pessoa que apresente o cartão magnético realmente seja quem diz ser.

A utilização de características biométricas não exclui totalmente os atuais consagrados métodos de autenticação pessoal, mas faz substancial contribuição com respeito à qualidade do serviço de identificação.

A figura 2.2 apresenta um diagrama de blocos com a tipologia de métodos de autenticação associados a sistemas baseados em características biométricas. As abordagens de posse e conhecimento podem ser combinadas com a abordagem envolvendo características biométricas visando prover uma autenticação mais segura, indicando que os métodos clássicos não são descartáveis mediante a utiliza-

ção da abordagem envolvendo características biométricas. A figura também mostra alguns exemplos de tipos de biometria para as características fisiológicas e de comportamento. Nas próximas subseções são descritos esses exemplos de forma mais suscinta.

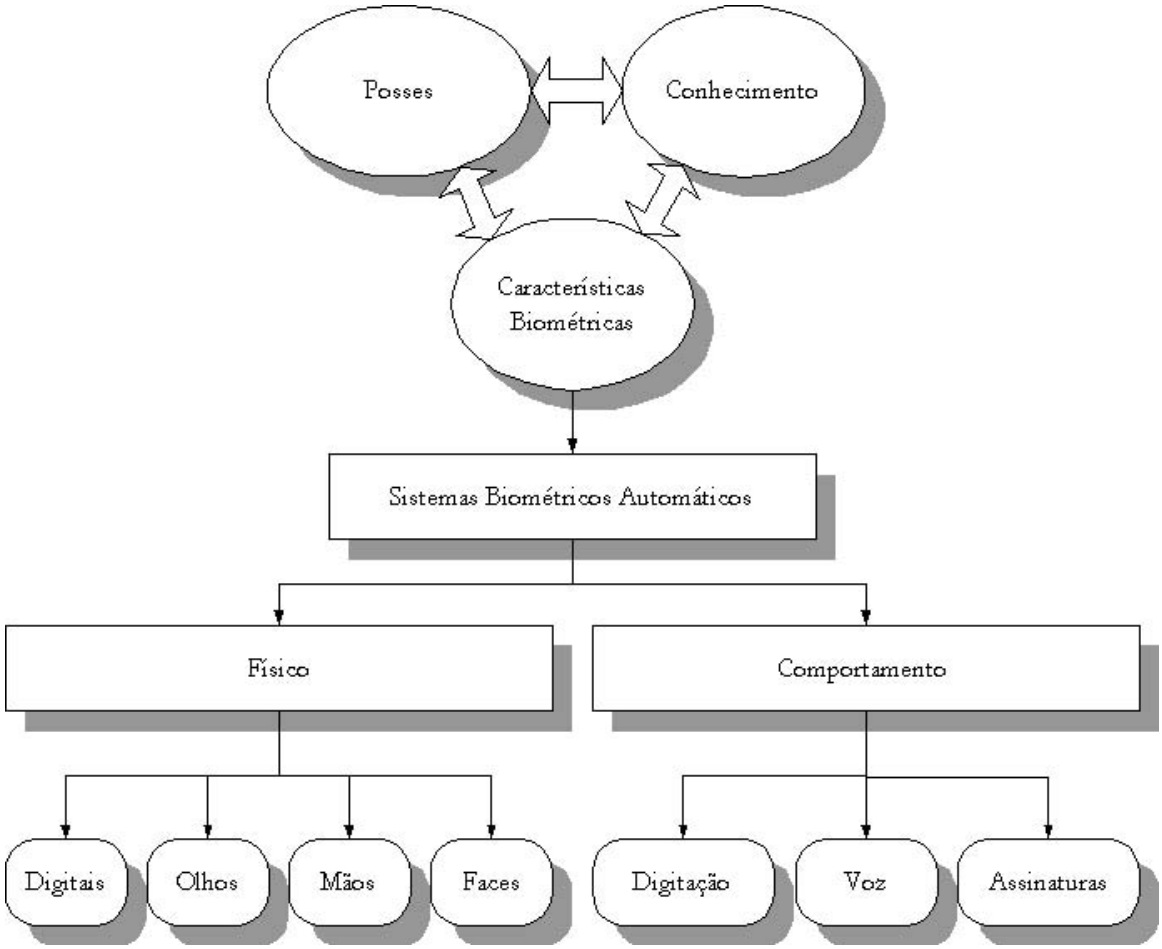


Fig. 2.2: Tipologia de métodos de identificação associados a sistemas baseados em características biométricas.

### 2.2.1 Impressões Digitais

A estabilidade e unicidade das impressões digitais são bem estabelecidas na sociedade. Segundo Wayman [9], estima-se que a chance de duas pessoas, incluindo gêmeos, tenham a mesma impressão digital é menor do que uma em um bilhão. A extração de características sobre impressões digitais se baseia em encontrar a posição de pequenos pontos chamados de minúcias que estão presentes nas digitais, tais como, pontos de finalização de linhas e pontos de junção de linhas [11]. Outros contam o número de vales e sulcos que existem entre esses pontos [11]. Dependendo do esquema

de identificação escolhido e do grau de segurança do sistema, o arquivo de referência que contêm as informações sobre a impressão digital varia de algumas centenas de bytes até milhares de bytes. Hoje em dia, a maior aplicação da tecnologia de impressões digitais é em sistemas de identificação automática utilizadas pela polícia em vários países do mundo.

### 2.2.2 Olhos

Tanto o padrão da íris quanto o padrão dos vasos sangüíneos do fundo do olho (retina) provêem uma base única para identificação [9]. A principal vantagem da captura do padrão da íris sobre a varredura da retina é que na primeira não se necessita que o olho do indivíduo que está sendo testado esteja focalizado em um determinado lugar. Ainda mais, segundo [12], a imagem da íris pode ser obtida pelo dispositivo de captura até a um metro de distância. No caso da varredura de retina, essa é realizada direcionando-se uma luz infravermelha de baixa intensidade na pupila e na parte posterior do olho. O padrão da retina é refletida de volta para a câmera a qual captura o padrão. A varredura da retina é um dos melhores métodos biométricos existentes, com taxas de erros muito baixas, base de dados de referências pequenos e processos rápidos de confirmação de identidade. O que mais dificulta a difusão desse tipo de tecnologia, continua sendo a resistência dos usuários, isto é, convencer a pessoa que vai se servir dessa técnica para a autenticação de identidade, de que a luz infravermelha que incidirá sobre seu olho não lhe irá fazer mal [9].

### 2.2.3 Mãos

A autenticação num sistema de identificação via geometria da mão se baseia em medidas das dimensões de partes das mãos, tais como o comprimento do dedo, sua largura e área. A classificação utilizando esses parâmetros, leva em conta a forte correlação que existe entre essas diferentes medidas. Os primeiros sistemas baseados nessas características datam de 1960, sendo que as medidas que utilizavam eram apenas o comprimento de quatro dedos [9].

### 2.2.4 Face

Uma das áreas que está crescendo mais rapidamente na industria da biometria em termos de novos esforços de desenvolvimento é a verificação e identificação através da faces [9]. Muitos dos trabalhos nessa área empregam tanto métodos de redes neurais como correlações estatísticas do formato geométrico da face. Esses métodos tentam imitar como os seres humanos reconhecem uma outra pessoa. A imagem das faces são adquiridas de forma direta pelos equipamentos de vídeo que hoje em dia estão disponíveis. Os atuais sistemas têm dificuldade de conseguir altos níveis de performance

quando a base de dados aumenta para alguns milhares de indivíduos [13].

### 2.2.5 Voz

A voz é utilizada em sistemas automáticos de verificação/identificação de locutor. Essa abordagem biométrica é muito atrativa visto que é considerada pouco invasiva pelos usuários. Os humanos utilizam-se de características de alto nível [9], tais como sotaque, estilo do locutor, entonação, estado emocional, dentre outros, para reconhecer uma pessoa através de sua voz. Como esse tipo de característica é difícil de ser adquirida e mensurada de forma automática pelo computador, parâmetros de baixo nível derivados de medidas acústicas do sinal de voz, como freqüência fundamental, envoltória espectral, freqüência de formantes, energia, entre outros, são empregados.

### 2.2.6 Assinaturas

Os sistemas de reconhecimento de assinaturas se dividem em sistemas dinâmicos e sistemas estáticos. Os sistemas de reconhecimento de assinaturas dinâmicos, utilizam técnicas baseadas nas pequenas diferenças do processo dinâmico da escrita da assinatura, como por exemplo, pressão, aceleração, e número de vezes que levantamos a caneta do papel. Por outro lado, os sistemas que utilizam apenas a imagem da assinatura (sistemas estáticos), utilizam características como a inclinação dos traços da escrita, o número de palavras, a razão entre a altura e o comprimento da assinatura [9].

A chave do sucesso de um sistema de verificação/identificação de assinaturas é encontrar características da assinatura que sejam mais constantes, isto é, que variem pouco durante o processo de cadastramento.

### 2.2.7 Dinâmica da Digitação

A dinâmica da digitação, também chamada de ritmo de digitação, é um método biométrico que está diretamente ligado a área de segurança de computadores. Como o nome indica, esse método analisa a maneira como os usuários digitam no teclado seu login e sua senha. Nesse caso, as característica extraídas são a seqüência de valores alfanuméricos que se está digitando, assim como o intervalo de tempo entre apertar uma tecla e outra em uma palavra [14].

Nos sistemas que se servem dessa abordagem biométrica, o mais notável é que o usuário não percebe que está sendo identificado por meio de uma característica biométrica a não ser que lhe seja dito. Por outro lado, para o sucesso desse método, quanto maior for a habilidade do usuário na digitação, será mais fácil e confiável a maneira como é reconhecido, visto que sua variação intrapessoal será pequena [15].

## 2.3 Comparação de Tecnologias Biométricas

A tabela 2.1 apresenta uma comparação entre as tecnologias biométricas apresentadas anteriormente, levando em consideração três fatores [9]:

- Desempenho: refere-se à capacidade de um sistema em autenticar corretamente um indivíduo devido a um tipo de característica biométrica.
- Aceitabilidade: indica o quanto as pessoas aceitam esse tipo de identificação biométrica na sua vida cotidiana.
- Fraudável: reflete a facilidade com que um sistema pode ser enganado por métodos fraudulentos.

Tab. 2.1: Comparação entre tecnologias biométricas

| **Característica biométrica** | **Desempenho** | **Aceitabilidade** | **Fraudável** |
|---|---|---|---|
| Impressões Digitais | Alta | Média | Baixa |
| Olhos | Alta | Baixa | Baixa |
| Mãos | Média | Média | Média |
| Face | Baixa | Alta | Alta |
| Voz | Baixa | Alta | Baixa |
| Assinaturas | Baixa | Alta | Alta |
| Digitação | Alta | Alta | Média |

## 2.4 Medidas de desempenho em sistemas biométricos

O desempenho de um sistema biométrico é um fator importante para atestar se o sistema consegue atender os objetivos citados anteriormente. No geral, apenas saber o que uma medida de desempenho significa é suficiente para entender o mecanismo de obtenção de tal medida. Para as finalidades deste trabalho, as medidas estatísticas usadas para medir o desempenho de sistemas biométricos são:

- Erro ou taxa de falsa aceitação (*false acceptance rate* - FAR) que quantifica probabilisticamente quanto o sistema erra por aceitar usuários impostores (usuários fraudulentos).
- Erro ou taxa de falsa rejeição (*false rejection rate* - FRR), uma estimativa da probabilidade do sistema rejeitar uma pessoa autorizada (usuário autêntico).

As taxas de FAR e FRR são expressos em porcentagens e variam de acordo com um limiar de decisão escolhido. Estas taxas são comumente ilustradas por curvas, e podem ser observadas na figura 2.3 e são discutidas em detalhes a seguir.

### 2.4.1 FAR

A FAR é definida como a probabilidade de um usuário mal intencionado ser aceito quando faz uma reivindicação por autenticação usando uma identidade falsa e o sistema aceitar aquele usuário como verdadeiro. Exemplificando, se o usuário $\alpha$ usando a ID do usuário $\beta$ conseguir abrir uma sessão no computador do usuário $\beta$ e o sistema autenticar $\alpha$ como $\beta$ então ocorre uma falsa aceitação.

Isto poderá acontecer porque o limiar operacional para a conta $\beta$ é ajustado demasiadamente baixo, ou pode ser que a característica biométrica $\alpha$ e $\beta$ sejam muito similares. Em ambos os casos, ocorre uma falsa aceitação.

A importância da FAR está ligada ao algorítmo de classificação do padrão biométrico. Pode-se dizer que um algorítmo forte é aquele onde quase não ocorre falsa aceitação.

### 2.4.2 FRR

A FRR é definida como a probabilidade de um usuário verdadeiro fazer uma tentativa de autenticação no sistema biométrico e ser rejeitado. Exemplificando, se o usuário $\alpha$ usando sua ID de usuário e apresentando sua característica biométrica, não conseguir abrir uma sessão no seu próprio computador, ou seja, o sistema não reconheceu $\alpha$ como sendo $\alpha$, então ocorre uma falsa rejeição.

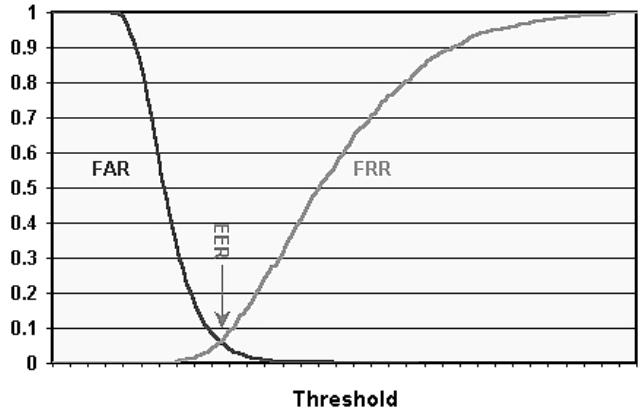


Fig. 2.3: Curvas de FAR e FRR

Isto poderá acontecer porque o limiar operacional estimado para a conta $\alpha$ é ajustado demasiadamente alto, ou pode ser que a característica biométrica apresentada pelo usuário $\alpha$ não é próxima o bastante do modelo armazenado para a conta $\alpha$. Em ambos os casos, ocorre uma falsa rejeição.

A importância da FRR está ligada a robustez do algorítmo de classificação do padrão biométrico. Quanto mais preciso for o algoritmo, menor será o número de falsas rejeições [10]. Pode-se dizer que um algorítmo forte é aquele onde sempre que um usuário legítimo tenta autenticar-se, o sistema reconhece-o com sucesso.

### 2.4.3 zeroFAR e zeroFRR

A figura 2.3 mostra um exemplo real de classificação, no qual em nenhum ponto referente a $t$ ambas as taxas são iguais a zero. No caso ideal, existirão um ou mais pontos referentes a $t$ onde ambas as taxas de erro alcançadas seriam iguais a zero. Ainda na figura, podemos observar três pontos importantes referentes à $t_a$, $t_b$ e $t_c$ chamados de ZeroFAR, ERR (*Equal Error Rate*) e ZeroFRR, respectivamente.

ZeroFRR é o valor de FAR quando FRR tem valor zero e indica a probabilidade do sistema aceitar o acesso de pessoas não-autorizadas, quando todos os acessos de pessoas autorizadas são aceitas. ZeroFAR é o valor de FRR quando FAR tem valor zero e indica a probabilidade do sistema rejeitar o acesso de pessoas autorizadas, quando todos os acessos de pessoas não-autorizadas são rejeitados.

### 2.4.4 EER

EER é definido como o ponto de cruzamento de um gráfico que contenha tanto a falsa aceitação como a falsa rejeição. Em outras palavras, é o ponto no qual os valores de FAR e FRR são iguais. Segundo Ross *et all* [**?**], a taxa de erro igual (EER) é o ponto mais importante, pois especifica a separabilidade que o sistema oferece entre os acessos permitidos e os não-permitidos. O valor de EER pode ser calculado a partir de uma curva de características operacionais (*receiver operating characteristic* - ROC), que plota FRR por FAR, determinando a precisão ou sensibilidade a erros de uma metodologia de autenticação biométrica.

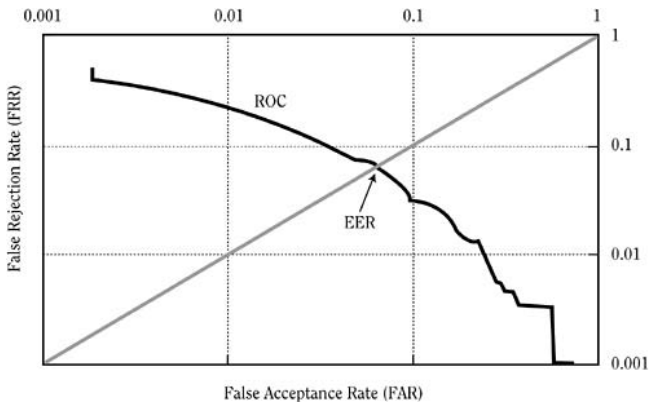


Fig. 2.4: Curva de ROC.

Para calcular a curva de ROC de um sistema biométrico, cada um dos pontos correspondentes à FAR e FRR são plotados em uma escala logarítmica (figura 2.4). O EER é encontrado estendendo uma linha a 45 graus a partir do ponto de origem $(0, 0)$. Onde esta linha cruzar a ROC, este ponto é o

EER. Isto acontece porque quando a FRR tem valor igual a 1 (FRR = 100%), a FAR assume valor 0, e onde a FRR assume o valor 0, a FAR é igual a 1 (FAR = 100%).

Escolher o uso do ponto de cruzamento entre FRR por FAR é uma questão significante. Um EER calculado usando FRR e FAR é suceptível para ser manipulado, baseado na granularidade dos valores de limiares obtidos para a FAR e FRR. A importância do EER dá-se quando queremos comparar diferentes sistemas biométricos. Isto é, cada sistema biométrico geralmente trabalha com seus próprios valores absolutos de limiares para calcular FAR e FRR, dificultando a comparação direta. Porém se conhecemos o valor relativo de EER para um sistema, podemos proceder com uma comparação estatística normalizada, muito embora, em uma aplicação real um sistema de autenticação raramente consiga operar exatamente neste ponto. Na prática, sistemas são programados para trabalharem próximos de ZeroFRR ou ZeroFAR.

# Capítulo 3

# Dinâmica da Digitação

Autenticação por senha é a forma mais popular de segurança e identificação no mundo tecnológico atual. No entanto, por se tratar de um método simples, é também considerado um método vulnerável a ataques de impostores que desejam fraudar de alguma forma os sistemas protegidos por senhas.

Porém, no ato de digitar sua senha, os usuários destes sistemas protegidos certamente [16] impõem um rítmo ao pressionar e soltar as teclas - um rítmo dinâmico de digitar senhas. Este rítmo é uma maneira única do usuário, e permite que sejam construídos sistemas tanto para verificar as senhas dos usuários, como também validar se quem está digitando a senha deseja fraudar o sistema, pois por mais que pessoas mal intencionadas conheçam a senha, quem digita deve ter o rítmo associado a esta senha.

Isto posto, pode-se dizer que dinâmica da digitação é o processo de analisar a maneira como um usuário digita em um terminal [17], monitorando suas entradas em um teclado, no intuito de autenticá-lo baseado em seu ritmo de digitação habitual.

Quando o usuário digita o conjunto de caracteres que compõem sua senha, o sistema constroi um vetor composto dos eventos "tecla-pressionada"e "tecla-solta". Se uma senha composta de $n$ caracteres é digitada, o resultado será um vetor de dimensão $(2n)$, onde cada posição representará uma unidade de tempo calculada em função dos eventos tecla pressionada e tecla solta.

Analisando-se um exemplo onde a senha consiste de $n = 4$ caracteres: "1-2-3-4". O vetor característica de dimensão 8 contendo os tempos será $[0, 0.30, 0.31, 0.60, 0.61, 0.75, 0.76, 0.95]$. A unidade de tempo adotada foi segundos.

De maneira simplificada, cada tecla da senha "1-2-3-4", está associada a ocorrência de um evento tecla-pressionada e tecla-solta, ou seja, cada tecla está associada a duas posições no vetor, tal qual exemplificado na figura 3.1.

Uma vez que os eventos foram observados durante a digitação da senha, o sistema parte para extrair um conjunto de características. Assim, podemos citar as características mais utilizadas neste

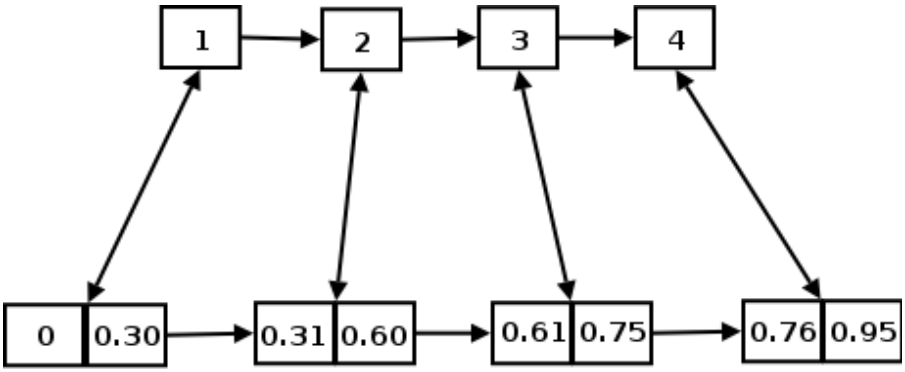


Fig. 3.1: Exemplo de como os eventos tecla-pressionada e tecla-solta ocorrem e como guardam relação com a ordem em que ocorrem as teclas.

contexto como[18]:

- Latência da digitação (*keystroke latency*): é o intervalo de tempo entre as digitações de teclas sucessivas pelo usuário.

- Duração da digitação (*keystroke duration*): é o intervalo de tempo em que uma tecla permanece pressionada pelo usuário.

Motivado pelo fato observado, ou seja, que existe um padrão repetitivo único de intervalos de tempos quando o usuário digita a senha e que torna-o distinguível de impostores, o sistema coleta uma quantidade de amostras destes vetores de características que serão usados para compor um modelo do usuário, permitindo assim validar a nova amostra no processo de autenticação. Se essa nova amostra estiver estritamente relacionada com o modelo, então o usuário é aceito, caso contrário é rejeitado.

Dinâmica da digitação é um método puramente baseado em software, de baixo custo e mais transparente para o usuário do que os sistemas de autenticação por biométria tradicionais, como impressões digitais ou reconhecimento de íris [9].

Dinâmica da digitação pode ser dividida em duas abordagens em relação ao momento da autenticação [14]:

- **Estática**: Na abordagem estática, a autenticação é realizada no início da interação do usuário com o sistema, geralmente no momento do seu *logon* ou quando o usuário pretende entrar na área restrita. Vale ressaltar que essa abordagem pode ser aplicada também de tempos em tempos, durante a sessão do usuário, sempre que uma determinada ação necessitar que seja confirmada através da senha.

- **Contínua**: Na abordagem contínua, a autenticação é realizada várias vezes no decorrer da interação do sistema com o usuário, observando todos os eventos que ocorrem no teclado. Normalmente esta abordagem analisa as ocorrências de dígrafos, ou seja, as características são extraídas de caracteres que aparecem em pares. Esta abordagem pode ser usada em conjunto com padrões de comportamento do usuário ao usar o *mouse*, como em [19]. A abordagem contínua é mais segura que a abordagem estática, pois ela pode realizar a detecção caso haja uma troca de usuários. A desvantagem desta abordagem é que ela exige mais processamento que a abordagem estática.

As características biométricas de dinâmica da digitação são em sua maioria extraídas do sinal baseado nos tempos capturados em cada uma das teclas digitadas por um indivíduo durante a digitação de uma palavra, frase ou texto.

## 3.1 Estado da Arte

A utilização de características provenientes da digitação foi sugerida por Spillane em 1975 [20]. Como declarado em [21], em meados de 1895, foi observado que os operadores de telégrafo possuíam uma maneira particular de digitar as mensagens. De fato, era possível para outros operadores identificarem quem estava transmitindo a mensagem apenas escutando o som da digitação de pontos. Desde meados dos anos 70 esta observação foi formalizada e aplicada para a área de dinâmica da digitação, acarretando em alguns trabalhos que foram então publicados utilizando-a para a autenticação de usuários, validando a hipótese de que características vindas do teclado são realmente discriminantes.

Em 1985, Umphress e Williams [21] publicaram um dos primeiros trabalhos envolvendo dinâmica da digitação para autenticação pessoal. Neste trabalho foram utilizados os intervalos de tempo entre as teclas em dois tipos de informações: um texto contendo 1400 caracteres no processo de cadastramento e um segundo texto contendo 300 caracteres no processo de autenticação. A principal desvantagem foi o grande tamanho da informação e da quantidade de dados de entrada a ser digitado, que resultou em apenas uma taxa a ser analisada: a FAR, cujo valor foi igual a 6%. Em 1988, Williams e Leggett [22] estenderam o trabalho realizado em [21], mostrando que ele poderia ser utilizado em uma verificação estática no login em conjunto com uma frase, o que compensava a desvantagem de usar textos extensos, que dificultavam a coleta de amostras. Em Leggett *et all* [20], que também é uma extensão dos trabalhos apresentados em [21] e [22], foi utilizado pela primeira vez o conceito de uma abordagem dinâmica. Nesta abordagem, a verificação da identidade do usuário é realizada no decorrer da sua interação com o sistema, e não somente no momento do login. Outros trabalhos foram publicados a partir dos anos 90 [16], [14], [23], [24] e [25] usando amostras com uma quantidade de

caracteres menor a serem digitadas do que as apresentadas em [21], [22] e [20]. Atualmente, as aplicações comerciais mais conhecidas que utilizam a dinâmica da digitação para autenticação pessoal são da Biopassword [26], porém os detalhes de suas implementações não são divulgados.

Os esquemas aplicados nesses trabalhos se encaixam na descrição que se segue. No momento da autenticação, o usuário digita a senha (*string* alvo). Enquanto o usuário está digitando-a, dados como os tempos de pressionamento e soltura das teclas são adquiridos. A partir destes dados são extraídas características que formarão uma amostra. Caso o usuário não esteja cadastrado, então uma quantidade de amostras é armazenada, formando um conjunto de treinamento, do qual será gerado o modelo. Caso o usuário esteja cadastrado, uma amostra é calculada e enviada para o classificador. O classificador, por sua vez, decide se o usuário pode acessar o sistema de acordo com a verificação ou identificação de sua identidade. Cada um dos trabalhos mencionados adota um posicionamento diferenciado com relação à quantidade de tentativas e ao mecanismo de atualização, ou seja, pode ser concedida mais de uma tentativa de autenticação para cada usuário, e/ou também pode ou não ser feita uma atualização no conjunto de treinamento com o objetivo de atualizar o modelo do usuário em questão. Nas próximas subseções, alguns aspectos importantes apresentados nos trabalhos mencionados são discutidos.

### 3.1.1 *String* ou Informação Alvo

*String* ou Informação alvo compreende o conjunto de caracteres digitados pelo usuário. Esta informação é monitorada por um sistema de autenticação que a coleta e a analisa. A informação alvo pode ser o *login* e/ou a senha. Em [16], quatro informações foram utilizadas como alvo (*login*, senha, primeiro nome, último nome). Em [27], a informação era imposta pelo sistema em níveis de dificuldade, dados pela compreensão e pelo significado da palavra. No nível de dificuldade mais baixo a informação continha uma palavra com algum sentido lexicográfico e, opcionalmente, com algum número no começo ou no final dessa palavra. No nível mais alto, a informação era composta por números e letras sem nenhum significado ou compreensão aparente. Em [28], além da informação alvo escolhida por cada usuário para fins de verificação, cada usuário deveria digitar uma frase com 30 caracteres para a realização de um processo de autenticação do tipo identificação.

A quantidade de caracteres contida na informação alvo influencia a quantidade de erros de classificação, que aumenta quando a quantidade de caracteres se torna tão pequena quanto dez caracteres [23]. Em [28] uma das informações alvo analisada continha 31 caracteres, e, em [16], a informação alvo era composta por quatro informações, que totalizavam aproximadamente 28 caracteres.

A grande maioria dos trabalhos relacionados com a área obrigam os usuários a memorizarem *string* alvo pré-estabelecidas, porém em [29] os autores comprovaram que este tipo de abordagem é menos eficiente do que permitir ao usuário que encontre uma sequência de teclas com a qual está

familiarizado.

### 3.1.2 Número de amostras

A quantidade de amostras coletadas e armazenadas no cadastramento torna-se um aspecto crucial na medida em que os erros de classificação aumentam quando a quantidade de amostras captadas é reduzida [28]. A quantidade de amostras nos trabalhos pesquisados varia desde apenas três amostras [30] até trinta amostras [28] por usuário, enquanto em [31] a quantidade de amostras varia de acordo com cada usuário. Ainda em [31], cada usuário continua fornecendo amostras até que o sistema determine quando as características extraídas destas amostras estabilizaram-se o suficiente para identificá-lo. Este processo é realizado nas duas últimas amostras adquiridas pelo sistema, que determina a parada da coleta de amostras quando ambas apresentam um certo grau de similaridade. A quantidade de amostras utilizadas até que este grau fosse atingido variou entre duas e dez.

Em [14] os autores observaram que o número mínimo de amostras para não comprometer o desempenho do sistema é de seis amostras por usuário e em [32] os autores comprovaram que quanto maior o número de amostras, melhor será o modelo do usuário.

### 3.1.3 Extração das características

As características escolhidas para representar cada classe devem ser discriminantes, ou seja, elas têm de ser altamente repetitivas no mesmo usuário e diferentes para os demais [14]. A característica mais utilizada é a latência da digitação, que representa o intervalo de tempo entre teclas sucessivas ($t_i$ e $t_{i+1}$). Esta característica pode ser extraída de duas maneiras. Na primeira maneira, o cálculo é realizado pela diferença entre o pressionamento de teclas sucessivas ($t_i$ e $t_{i+1}$). Na segunda maneira, a extração é realizada pela diferença entre o pressionamento de $t_{i+1}$ e a soltura de $t_i$. Por esta extração, a primeira resulta sempre em valores positivos, pois o pressionamento de $t_i$ ocorre sempre antes de $t_{i+1}$, enquanto a segunda pode resultar em valores negativos, pois a soltura de $t_i$ pode ocorrer depois do pressionamento de $t_{i+1}$, dependendo da maneira como cada usuário digita.

Outra característica bastante utilizada é a duração da digitação, que representa o intervalo de tempo que uma tecla permanece pressionada, ou seja, a diferença entre os tempos de pressionamento e soltura de uma mesma tecla. Esta característica foi utilizada pela primeira vez em 1997 [24].

Os trabalhos referenciados nesta dissertação utilizaram em sua maioria a latência das teclas, mas em [24] e [33] a utilização conjunta da latência da tecla e da duração da tecla trouxe melhores resultados.

A pressão da tecla é mais uma das características presente no contexto da dinâmica da digitação que representa o nível de pressão aplicada na tecla pelo usuário. Esta característica não é muito

utilizada, pois para a sua aquisição seria necessário um teclado especializado que capturasse a pressão aplicada nas teclas.

Em quase todas as referências pesquisadas envolvendo dinâmica da digitação, as características são extraídas em função dos caracteres gerados, e não das teclas utilizadas, a exceção pode ser lida em [15]. Por exemplo, na digitação de "Ana", a característica latência da tecla em função dos caracteres calculará dois intervalos: (A, n) e (n, a). Já fazendo a extração da latência da tecla em função das teclas resultará, por exemplo, em três intervalos: (shift, A), (A, n) e (n, a).

De Ru *et all* [31] analizaram uma característica diferente baseada na distância das teclas no teclado alfa-numérico e a combinação de teclas consideradas difíceis, porém obriga o usuário a decorar uma *string* à qual não está habituado, e em [34] os autores propõem o uso de dígrafos - o tempo gasto para pressionar ou soltar duas teclas consecutivas - como característica.

### 3.1.4 Precisão do Tempo

A precisão do tempo indica o grau de exatidão obtido na medição da captura de amostras de digitação. Dessa forma, quanto maior a precisão do tempo, mais os dados capturados estarão próximos da realidade. Além disso, para a determinação de uma precisão deve ser levado em consideração à ordem de grandeza relacionada aos tempos de digitação capturados dentro de uma população de usuários do sistema. A precisão estabelecida nas pesquisas referenciadas variou entre 0.1 milisegundo [33] e 1 segundo [27]. Nas conclusões de [35] foi mencionado que uma maior precisão era desejável para melhorar os resultados obtidos.

### 3.1.5 Tentativas de autenticação

Em [35] foi observado que os usuários legítimos falhavam na sua primeira tentativa de autenticação, mas, na segunda tentativa, a autenticação era realizada com sucesso. A solução neste caso foi a adoção de duas tentativas no processo de autenticação. Outra solução adotada em [28] foi a obrigatoriedade do fornecimento de duas amostras a cada autenticação. Assim, o sistema analisa as duas amostras conjuntamente, escolhendo para cada intervalo da latência de tecla, que é a característica analisada nesta referência, o menor valor entre as duas amostras. Esta técnica foi chamada de *shuffling*. Em [3] e [36] os autores perceberam que usar taxas de FRR em 30% e permitir aos usuário autênticos duas tentativas, não comprometem as taxas de FAR.

### 3.1.6 Mecanismo de adaptação

Características biométricas de comportamento podem sofrer pequenas mudanças com o passar do tempo, e, por esse motivo, muitos sistemas biométricos permitem que sejam feitas atualizações nos modelos dos usuários, ou seja, um recadastramento pode ser feito para manter o modelo do usuário sempre atualizado. Estas atualizações podem ser feitas de duas maneiras: realizando um novo cadastramento ou utilizando um mecanismo de adaptação. A maioria dos pesquisadores referenciados neste trabalho não mencionam este aspecto importante em seus trabalhos, porém em [14] os autores citam um mecanismo de adaptação onde sempre que um usuário é autenticado de maneira correta o sistema calcula outro modelo acrescentando a entrada atual como uma nova amostra e descartando a mais antiga.

Em [15] o mecanismo de atualização adota um critério que vai depender do usuário ser corretamente identificado e se a distância entre a amostra e a variância da característica armazenada for maior que um determinado limiar, então deve ser atualizado o modelo.

### 3.1.7 Classificador

O classificador é responsável pelo processo de decisão do sistema de autenticação. No caso da verificação, esta decisão é determinar se a identidade de um usuário é verdadeira ou não. Os classificadores estatísticos são os mais utilizados, como por exemplo *k-means*, Bayes, distância euclidiana, dentre outros [14]-[23], [33], [37] e [27]. Em [16], [28], [27] e [38] um classificador de distância mínima é utilizado.

Em [33] é aplicado um classificador de aprendizagem indutiva, cujo aprendizado é realizado com amostras de usuários legítimos e impostores.

Nos últimos anos, classificadores em redes neurais têm sido utilizados [30], [24] e [37]. Estes classificadores têm bons resultados em pequenas bases de dados, porém eles apresentam uma limitação em relação ao re-treinamento, condição necessária toda vez que um usuário é cadastrado. Assim, para aplicações como controle de acesso, estes tipos de classificadores não são apropriados. Em [35] os classificadores estatístico, neural e nebuloso foram combinados. Em geral, os melhores resultados são obtidos com os classificadores estatísticos.

Outro tipo de classificador é o nebuloso (*fuzzy*) que foi utilizado em [31] e [39] . Nestes trabalhos, foram aplicadas três variáveis lingüísticas: duas para entrada (intervalo de tempo e dificuldade de digitação) e uma para saída (categorização). Intervalo de tempo é a latência da tecla, enquanto a dificuldade da digitação de teclas sucessivas é um valor calculado a partir de dois critérios: o número de teclas existentes no teclado entre os caracteres digitados pelo usuário, e, se o caractere foi digitado utilizando combinações de teclas (exemplo: para digitar letras maiúsculas é necessário que mais de

uma tecla seja pressionada). A categorização representa a categoria de digitação do usuário. Esta categorização é calculada pelas entradas, utilizando uma base de regras, tal que quanto maior for o intervalo de tempo e a dificuldade da digitação, maior será a categorização do usuário. Com relação às funções de pertinência das variáveis, para o intervalo de tempo, foram utilizadas gaussianas e, para as outras variáveis, funções triangulares, as quais se adequaram melhor ao problema.

Além das técnicas discutidas anteriormente, outras técnicas vem sendo usadas mais recentemente. Em [32] e [40] foram utilizados árvores de decisão para reconhecer os usuários. Em [41], [42] e [43] os autores propôem o uso de modelos ocultos de markov para classificar usuários autênticos. Em [18] os autores discutem a aplicação de máquinas de vetores suporte (*Suport Vector Machines* SVM) aplicadas no processo de classificação da dinâmica de digitação.

## 3.2 Resumo

Um resumo das principais referências pesquisadas envolvendo verificação estática da dinâmica da digitação é de suma importância, pois um dos objetivos deste trabalho é que a metodologia a ser apresentada seja competitiva. Na tabela 3.1 pode ser visto este resumo, contendo: a informação alvo, a quantidade e o tipo de amostras, as características utilizadas, o classificador ou classificadores aplicados e as taxas de erros obtidas (FAR e FRR).

## 3.3 Aspectos na Dissertação

Com base nos estudo feitos no estado da arte em dinâmica da digitação, os aspectos mencionados anteriormente neste capítulo foram aplicados nesta dissertação. Com relação às configurações destes aspectos, elas são mostradas detalhadamente no próximo capítulo e suscintamente abaixo:

- A informação alvo deverá conter pelo menos oito caracteres, sendo todos compostos de números e colhidos na região do teclado onde encontra-se somente teclas numéricas. Esta abordagem é pouco explorada na literatura mundial, e somente em [3] e [41] este tipo de abordagem foi explorada.

- A quantidade de amostras coletadas por usuário foi determinada em dez amostras por sessão, mais do que essa quantidade aborrece os usuários. Entende-se por sessão o momento quando o usuário estava disponível a colaborar com o experimento;

- Várias características são analisadas, entre as quais estão as latências entre teclas e a duração da tecla. Uma das características surgiu da divisão da latência da tecla em duas características distintas de acordo com o cálculo que foi explicado neste capítulo;

Tab. 3.1: Resumo das principais pesquisas na área, relacionando os resultados obtidos pelos autores

| **Pesquisa** | **Informação Alvo** | **Quantidade de Amostras** | **Características** | **Classificador** | **FAR** | **FRR** |
|---|---|---|---|---|---|---|
| Joyce e Gupta (1990) [16] | login, senha, primeiro e último nomes | Oito amostras, caracteres alfanuméricos | latência da digitação | estatístico | 16.3 | 0.25 |
| de Ru e Eloff (1997) [31] | login e senha | dois a dez, caracteres alfanuméricos | latência da digitação | nebuloso | 7.4 | 2.8 |
| Bleha et al (1990) [28] | senha | Trinta, caracteres alfanuméricos | latência da digitação | estatístico | 8.1 | 2.8 |
| Ord et al (2000) [3] | senha | Seis, caracteres numéricos | latência entre teclas | neural | 9.9 | 30 |
| Haidar et alli (2000) [35] | senha | Quinze, combinando caracteres alfanuméricos | latência da tecla | estatístico, nebuloso e neural | 2.0 | 6.0 |
| Costa et al (2005) [42] | senha | Oito, caracteres numéricos | duas latências e código da tecla | modelos ocultos de markov | 3.5 | 4.9 |
| Araújo et all (2005) [38] | senha | Dez, caracteres alfanuméricos | duas latências, duração e código das teclas | estatístico | 1.45 | 1.89 |

- A precisão do tempo aplicada é de um segundo, pois é compatível com a ordem de grandeza dos dados capturados, conforme será apresentado no próximo capítulo;

- Um mecanismo de adaptação foi utilizado para manter o modelo do usuário atualizado;

- Um classificador foi aplicado e analisado, baseado na máxima verossimilhança entre o modelo

do usuário e a amostra no momento em que o usuário clama por ser autenticado.

No próximo capítulo será discutido e analisado em detalhes os aspectos que foram discutidos em função do estado da arte considerado.

# Capítulo 4

# Uma Proposta de Metodologia de Autenticação

A metodologia proposta nesta dissertação para a verificação da identidade de usuários via dinâmica da digitação é ilustrada na figura 4.1. Dois processos principais estão envolvidos na verificação pessoal: o cadastramento e a autenticação. Inicialmente, caso o usuário deseje acessar o sistema, ele deve criar uma nova conta ou informar uma conta $w$ já cadastrada. Caso uma nova conta seja criada, o processo de cadastramento é executado; caso contrário, o processo de autenticação é executado. Em ambos os processos, o usuário digita a *string* alvo escolhida por ele próprio na intenção de criar uma nova conta. A digitação é monitorada pelo sistema que captura os dados e, de posse deles, o sistema parte para o processo de extração das características que irão formar uma amostra. No processo de cadastramento, as amostras de digitação coletadas para cada usuário formam seu conjunto de treinamento que, em seguida, são usadas na geração do modelo do usuário que conterá informações importantes para representar o usuário em uma posterior sessão de autenticação. No processo de autenticação as amostras da conta são coletadas e analisadas por um classificador que verifica, utilizando o modelo correspondente do usuário, se a amostra apresentada pertence ao usuário ou não. Assim, se o classificador decidir pela veracidade da identidade do usuário, então a amostra é considerada verdadeira, proveniente de um usuário legítimo, garantindo, portanto, o acesso. Porém, se o classificador decidir que a amostra é falsa para a conta identificada na *string* alvo, então ele é considerado um impostor. Desta maneira, uma sessão de autenticação é enquadrada em uma das seguintes situações: uma tentativa com sucesso, uma tentativa com fracasso seguida de uma com sucesso, ou duas com fracasso. Finalmente, o modelo do usuário pode ser atualizado por um mecanismo de adaptação, com a finalidade de assimilar as mudanças que venham a ocorrer no ritmo de digitação do usuário ao longo do tempo.

Este trabalho inova pois, ao invés de considerar o teclado alfa-númerico e uma *string* que pode

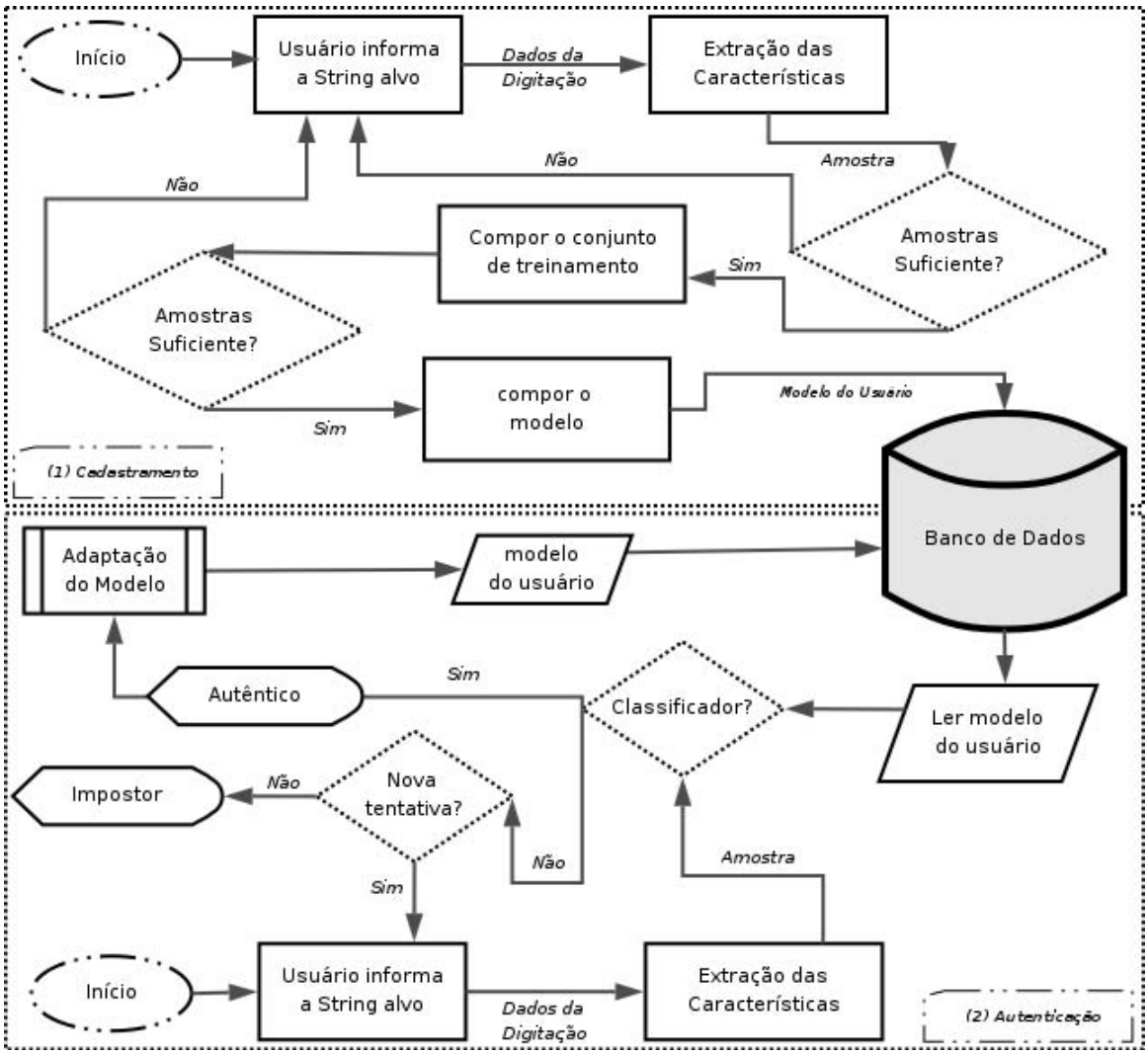


Fig. 4.1: Fluxograma de funcionamento do sistema.

conter tanto letras como números, restringe o usuário ao teclado númerico, com senhas numéricas limitadas em oito caracteres e escolhidas pelo usuário. Teclados desta natureza estão presentes em telefones celulares, caixas automáticos de banco ou em controle de acesso a áreas restritas. Além disso, também utiliza uma nova abordagem no processo de classificação que se baseia na máxima verossimilhança entre uma dada amostra e o modelo. Nas próximas sessões são apresentados aspectos importantes relacionados com a metodologia proposta.

## 4.1 Conta de Acesso

De acordo com a metodologia ilustrada na figura 4.1, durante o processo de verificação o usuário deve informar a conta de acesso válida e cadastrada no banco de dados do sistema. O conjunto $n$ de *c* contas armazenadas na base de dados é representada por *W* = $w_1, w_2, \ldots, w_n$. Cada conta $w$ possui três elementos relacionados: a informação alvo $ia_w$, o conjunto de amostras usadas no treinamento $S_w$ e o conjunto de amostras adquiridas durante a autenticação $L_w$. Os dois primeiros elementos são fornecidos durante o processo de cadastramento, enquanto o último é adquirido durante todas as autenticações. Além dos dados comentados anteriormente, o banco de dados armazena também os parâmetros calculados para o modelo do usuário, que é atualizado sempre que uma autenticação correta ocorre.

## 4.2 *String* ou Informação Alvo

A *String* alvo é o conjunto de caracteres digitados por um usuário, e é escolhida por ele no cadastramento da sua conta. Desta maneira, a informação alvo *ia* da conta $w$ é representada por $ia_w$. Quanto à escolha da *String* alvo pelo usuário, uma restrição é feita: ela deve conter oito caracteres. Ord [3] constatou que senhas menores que oito caracteres comprometem sensivelmente o desempenho do sistema e senhas maiores que oito caracteres são mais difíceis do usuário memorizar.

Além disso, a metodologia aqui proposta não trata os erros tipográficos e então os usuários não devem cometer erros no momento da digitação da informação alvo. Porém, caso os erros sejam cometidos, o usuário pode iniciar novamente a sua digitação.

## 4.3 Dados da Digitação

Enquanto o usuário está digitando a informação alvo, dados desta digitação são capturados. Para digitar uma informação alvo com $h$ caracteres é necessário que $n$ teclas sejam utilizadas, sendo $h \leq n$. Os dados da digitação são representados por $K = k_1, k_2, \ldots, k_n$, onde $k_i$ é constituido pelo código ASCII da tecla, pelo tempo de pressionamento da tecla e pelo tempo de soltura da tecla, representados respectivamente por $k_i.ct$, $k_i.tp$ e $k_i.ts$. O código ASCII é utilizado neste trabalho somente para fazer uma representação decimal de cada tecla e desta forma conduzir a comparação a um processo um-para-um, ou seja, autenticar o usuário ao invés de tentar identificá-lo. Os momentos dos tempos mencionados são capturados em função dos ciclos de *clock*, cuja abordagem é explicada na próxima sessão.

Para exemplificar, foram capturados os dados de uma digitação, tal que $ia_1 = (37883703)$ onde os

dados de digitação $K$ presentes na tabela 4.1 ocorrem na ordem de cima para baixo, até que o usuário pressione a tecla *enter*.

Tab. 4.1: Exemplo de dados de digitação para $ia_1 = (37883703)$

| caractere | tecla | $k$ | *k.ct* | *k.tp* | *k.ts* |
|---|---|---|---|---|---|
| 3 | 3 | $k_1$ | 51 | 7134052500 | 7141674418 |
| 7 | 7 | $k_2$ | 55 | 7149650522 | 7156020698 |
| 8 | 8 | $k_3$ | 56 | 7164973492 | 7170557878 |
| 8 | 8 | $k_4$ | 56 | 7171909350 | 7173264136 |
| 3 | 3 | $k_5$ | 51 | 7173650522 | 7174020698 |
| 7 | 7 | $k_6$ | 55 | 7174973492 | 7175557878 |
| 0 | 0 | $k_7$ | 48 | 7176909350 | 7178264136 |
| 3 | 3 | $k_8$ | 51 | 7187160900 | 7194419472 |
| Enter | Enter | $k_9$ | 65293 | 7215830130 | 7231191452 |

## 4.4 Captura do Tempo

A captura do tempo é necessária para saber os momentos em que as teclas envolvidas na digitação são pressionadas e soltas. A captura dos tempos pode ser feita de duas maneiras:

- Hora do Sistema: a precisão do tempo capturado depende do sistema operacional da máquina. Os sistemas operacionais da família Windows apresentam um *delay* entre 10 a 50 milissegundos [44] correspondente ao intervalo de tempo necessário para que um processo de captura do tempo requisite ao *kernel* do sistema operacional o direito de acessar o recurso necessário, ou seja, acessar o processador. Este *delay* é realidade em todos os sistemas operacionais, e seu tempo médio vai depender de fatores como quantidade de processos exigindo recurso, velocidade de processamento e memória, entre outros.

- Ciclos do *clock*: O *clock* é um trem de pulsos alternado de sinais de tensão gerados pelos circuitos do relógio (composto de um oscilador a cristal e circuitos auxiliares). Um ciclo de *clock* é delimitado pelo início da descida do sinal, equivalendo à excursão do sinal por um *"low"*, e uma próxima descida *"high"* do pulso. Utilizando ciclos do *clock*, é possível captar o tempo no nível de *hardware* e, portanto, o tempo de captura pode ser tão preciso quanto se necessita [15].

Fundamentalmente a dinâmica da digitação necessita adquirir os padrões do digitador de forma precisa no tempo [37]. Sendo assim, nesta dissertação é utilizado para a captura dos tempos os ciclos

do *clock* pelo fato de serem mais precisos que a hora do sistema, pois eliminam o *delay* ocasionado pelo sistema operacional na requisição do processo (tempo de requisição, espera e concessão para execução) ao processador, criando desta forma uma independência do tipo de sistema operacional.

Neste trabalho foi usada a função *Time Stamp Counter* para adquirir o número preciso de ciclos do processador, tal qual descrita em [37] e detalhada em [45]. Para mais detalhes sobre o processo de captura dos ciclos do *clock*, vide Apêndice A.

Depois da captura do tempo em ciclos de *clock*, é necessário converter este valor para unidades de tempo em milisegundos. A conversão é feita de acordo com a equação 4.1:

$$tempo = \left( \frac{quantidade\ de\ ciclos}{frequência\ do\ processador} \right) \times 1000 \tag{4.1}$$

onde *tempo* é o valor em milisegundos, *quantidade de ciclos* é o número de ciclos do *clock* e *frequência do processador* é a velocidade do processador dada em Hz (ciclos/segundo). Do resultado da equação que se obtém o *tempo*, todo o valor é aproveitado (parte inteira e fracionária).

Analizando-se o exemplo da tecla "3" na tabela 4.1, um intervalo de tempo medido quando a tecla é pressionada foi de *k.tp*= 7134052500, enquanto que o intervalo de tempo para que a tecla seja solta é de *k.ts* = 7141674418 - todos medidos em função de ciclos do *clock* em um processador de 1GHz, ou seja, $10^9$ Hz. Os seus valores equivalentes em milisegundos, calculados pela equação 4.1, são de 7134.052500 e 7141.674418 milissegundos, respectivamente. A tabela 4.1, após a precisão adotada, assume a configuração descrita na tabela 4.2:

Tab. 4.2: Exemplo de dados de digitação para $ia_1 = (37883703)$ após aplicar a precisão adotada

| caractere | tecla | *k* | *k.ct* | *k.tp* | *k.ts* |
|---|---|---|---|---|---|
| 3 | 3 | $k_1$ | 51 | 7134.052500 | 7141.674418 |
| 7 | 7 | $k_2$ | 55 | 7149.650522 | 7156.020698 |
| 8 | 8 | $k_3$ | 56 | 7164.973492 | 7170.557878 |
| 8 | 8 | $k_4$ | 56 | 7171.909350 | 7173.264136 |
| 3 | 3 | $k_5$ | 51 | 7173.650522 | 7174.020698 |
| 7 | 7 | $k_6$ | 55 | 7174.973492 | 7175.557878 |
| 0 | 0 | $k_7$ | 48 | 7176.909350 | 7178.264136 |
| 3 | 3 | $k_8$ | 51 | 7187.160900 | 7194.419472 |
| Enter | Enter | $k_9$ | 65293 | 7215.830130 | 7231.191452 |

Uma vez calculados os valores de tempo para cada um dos eventos associados as teclas, o passo seguinte é formatar estes valores em função da ordem em que ocorrem os eventos. O primeiro evento *k.tp* da primeira tecla passa a ser considerado como evento $0$, assim o sistema assume que este evento ocorre primeiro, e todos os eventos ocorrem sucessivamente após ele. Logo, ele é considerado como *Evento Inicial*. Em seguida, para cada próximo evento, este será calculado conforme a equação 4.2:

$$\textit{Evento Atual} = (\textit{Evento Atual} - \textit{Evento Inicial})\,/1000 \tag{4.2}$$

Depois de re-calculados os eventos na ordem que ocorrem, a tabela 4.2 assume a configuração apresentada na tabela 4.3:

Tab. 4.3: Exemplo de dados de digitação para $ia_1$ = $(37883703)$ após o cálculo dos eventos em função da ordem em que ocorrem.

| caractere | tecla | *k* | *k.ct* | *k.tp* | *k.ts* |
|---|---|---|---|---|---|
| 3 | 3 | $k_1$ | 51 | 0 | 0.007621918 |
| 7 | 7 | $k_2$ | 55 | 0.015598022 | 0.021968198 |
| 8 | 8 | $k_3$ | 56 | 0.030920992 | 0.036505378 |
| 8 | 8 | $k_4$ | 56 | 0.037856850 | 0.039211636 |
| 3 | 3 | $k_5$ | 51 | 0.039598022 | 0.039968198 |
| 7 | 7 | $k_6$ | 55 | 0.040920992 | 0.041505378 |
| 0 | 0 | $k_7$ | 48 | 0.042856850 | 0.044211636 |
| 3 | 3 | $k_8$ | 51 | 0.053108412 | 0.060366972 |
| Enter | Enter | $k_9$ | 65293 | 0.08177763 | 0.097138952 |

A partir destes novos valores, é possível extrair uma série de características que representarão a dinâmica dos usuários ao digitar. Nas próximas seções serão discutidas as características adotadas neste trabalho.

## 4.5 Extração das Características

As características são extraídas dos dados de digitação de uma *string* alvo em tempo real. Elas devem representar as classes (contas) envolvidas de maneira discriminante. Nesta dissertação são analisados 4 vetores de características, e a extração das características é feita em função das teclas envolvidas na autenticação. De outra forma, não seria possível conduzir a uma comparação 1-para-1. Em outras palavras, a primeira característica observada é a senha do usuário, que serve ao propósito de conduzir o sistema para comparar a entrada com o modelo do usuário diretamente, reduzindo assim o espaço de buscas (tempo de processamento) e, consequentemente, reduzindo as possibilidades de erro. Todas as outras características servem ao propósito de verificar a identidade. Elas são extraídas a partir dos tempos que o usuário leva para pressionar e soltar as teclas.

A figura 4.2 mostra um exemplo da extração das características em função do tempo, associadas à digitação dos caracteres $[1, 2, 3]$ = $([k_1.ct, k_1.tp, k_1.ts][k_2.ct, k_2.tp, k_2.ts][k_3.ct, k_3.tp, k_3.ts])$ e que servirá ao propósito de ilustrar as próximas discussões.

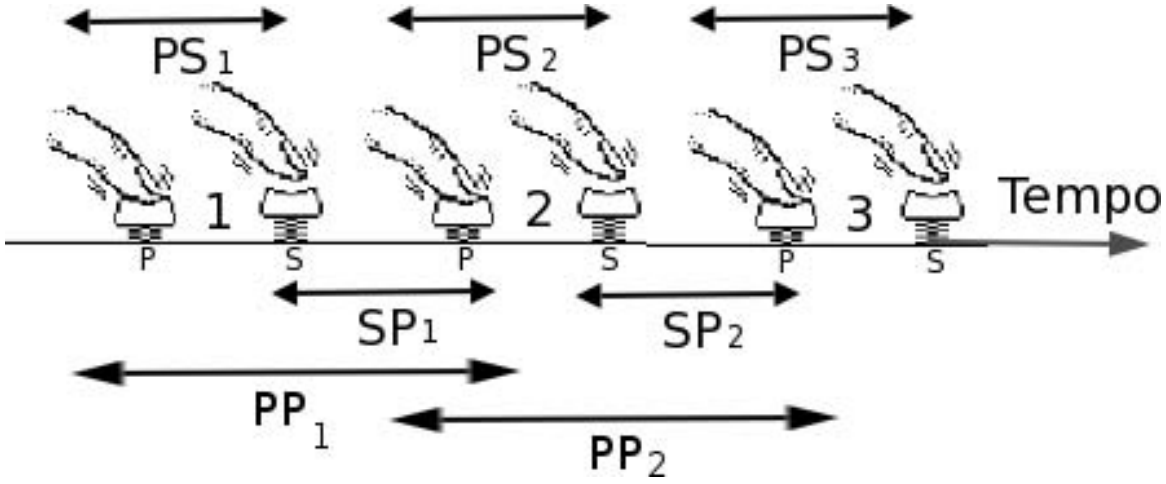


Fig. 4.2: Representação das características observadas durante a digitação dos caracteres 1, 2 e 3.

*PS* é o tempo em que a tecla permanece pressionada. Para a tecla 1, $PS_1$ é dado por $PS_1 = k_1.ts - k_1.tp$, a tecla 2, $PS_2$ é dado por $PS_2 = k_2.ts - k_2.tp$ e a tecla 3, $PS_3 = k_3.ts - k_3.tp$.

*SP* é o intervalo até a próxima tecla ser pressionada após a soltura da tecla anterior. Para a tecla 1, $SP_1$ é dado por $SP_1 = k_2.tp - k_1.ts$, enquanto que para a tecla 2, $SP_2$ é dado por $SP_2 = k_3.tp - k_2.ts$.

*PP* representa o intervalo de tempo entre os pressionamentos de teclas consecutivas. Para a tecla 1, $PP_1$ é dado por $P_1 = k_2.tp - k_1.tp$, enquanto que para a tecla 2, $PP_2$ é dado por $PP_2 = k_3.tp - k_2.tp$.

Cada amostra de *n* caracteres deverá conter 4 características de digitação, onde uma determinada característica de digitação para a amostra *w* da conta *a* pode ser definida por $K_{a,w} = (K_1(a,w), K_2(a,w), \ldots, K_n(a,w))$. Cada característica $K_i(a,w)$, onde $i \leq n$, representa uma das seguintes características observadas:

### 4.5.1 Código ASCII

O primeiro vetor de características é o codigo de teclas, que contém os códigos (decimal) ASCII das teclas utilizadas na digitação de uma *string* alvo. Ou seja, todo caracter digitado possui uma entrada na tabela ASCII. $C_a = (C_1(a), C_2(a), \ldots, C_n(a))$, onde $n$ é o número de dígitos na *string* alvo, representam os códigos das respectivas teclas para o teclado numérico e que estão contidos no modelo para a conta do usuário *a* e $C_{a,w} = (C_1(a,w), C_2(a,w), \ldots, C_n(a,w))$ representam os códigos das teclas da amostra *w* para a conta do usuário *a*.

Por exemplo, vamos examinar o vetor $C_a$ da conta fictícia ilustrada na figura 4.2. Para os caracteres "1-2-3", o vetor de características contendo os códigos ASCII será dado por $C = (49, 50, 51)$.

### 4.5.2 Pressiona-Solta

O segundo vetor de características é o intervalo de tempo em que a tecla permanece pressionada (Pressiona-Solta ou PS) [30]. As durações entre teclas são calculadas através da diferença entre o instante de pressionamento e o instante de soltura de uma tecla utilizada na digitação da *string* alvo.

Esta característica é representada pela expressão $PS_{a,w}$ = ($PS_1(a,w)$, $PS_2(a,w)$, ..., $PS_n(a,w)$), onde $PS_i(a,w)$ = $T_{i.solta}(a,w)$ - $T_{i.pressiona}(a,w)$ está relacionado com $K_i$. $T_{i.solta}(a,w)$ é o instante onde a tecla *i* é solta e $T_{i.pressiona}(a,w)$ é o instante onde a tecla *i* é pressionada.

Vamos examinar o vetor $PS_a$ da conta fictícia ilustrada na figura 4.2 para a amostra *w*. Para os caracteres "1-2-3", os tempo de pressionamento e soltura das teclas são demonstradas por ($T_{1.pressiona}$, $T_{1.solta}$), ($T_{2.pressiona}$, $T_{2.solta}$), ($T_{3.pressiona}$, $T_{3.solta}$) = $(0, 0.203)$, $(0.251, 0.342)$, $(0.396, 0.428)$, o vetor de características Pressiona-Solta será dado por $PS_{a,w}$ = ($PS_1(a,w)$, $PS_2(a,w)$, $PS_3(a,w)$) = $(0.203, 0.091, 0.032)$.

### 4.5.3 Pressiona-Pressiona

O terceiro vetor de características é o Pressiona-Pressiona (PP), que contém as latências de digitação. As latências de digitação neste vetor são calculadas pela diferença entre os pressionamentos de teclas sucessivas utilizadas na digitação da informação alvo. Esta característica é representada pela expressão $PP_{a,w}$ = ($PP_1(a,w)$, $PP_2(a,w)$, ..., $PP_{n-1}(a,w)$), onde $PP_i(a,w)$ = $T_{i+1.pressiona}(a,w)$ - $T_{i.pressiona}(a,w)$ está relacionado com ($K_i$,$K_{i+1}$).

Exemplificando o vetor de características PP da conta fictícia $C_a$ ilustrada na figura 4.2, para os caracteres "1-2-3", onde ($T_{1.pressiona}$, $T_{1.solta}$), ($T_{2.pressiona}$, $T_{2.solta}$), ($T_{3.pressiona}$, $T_{3.solta}$) = $(0, 0.203)$, $(0.251, 0.342)$, $(0.396, 0.428)$, o vetor de características será dado por $PP_{a,w}$ = ($PP_1(a,w)$, $PP_2(a,w)$) = $(0.251, 0.145)$.

### 4.5.4 Solta-Pressiona

O quarto vetor de características é o Solta-Pressiona (SP), que contém as latências de digitação calculadas a partir das diferenças entre o instante de tempo em que o usuário solta uma tecla e o instante em que o mesmo pressiona a tecla sucessiva utilizada na digitação da informação alvo.

Esta característica é representada pela expressão $SP_{a,w}$ = ($SP_1(a,w)$, $SP_2(a,w)$, ..., $SP_{n-1}(a,w)$), onde $SP_i(a,w)$ = $T_{i+1.pressiona}(a,w)$ - $T_{i.solta}(a,w)$ está relacionado com ($K_i$,$K_{i+1}$).

Para demonstrar com um exemplo do vetor $SP_a$ da conta fictícia ilustrada na figura 4.2 para a amostra *w*. Os caracteres "1-2-3"onde ($T_{1.pressiona}$, $T_{1.solta}$), ($T_{2.pressiona}$, $T_{2.solta}$), ($T_{3.pressiona}$, $T_{3.solta}$) = $(0, 0.203), (0.251, 0.342), (0.396, 0.428)$, o vetor de características Solta-Pressiona será dado por $SP_{a,w}$ = ($SP_1(a,w)$, $SP_2(a,w)$) = $(0.048, 0.054)$. Pode-se observar que para o exemplo de uma *string*

alvo da conta *a* com três teclas, o vetor de característica $SP_a$ possui somente duas posições, ou seja, conforme a formulação anterior não existe a característica $SP_a$ da última tecla.

## 4.6 Amostras

Uma amostra é composta pelas características que são extraídas dos dados de digitação. Assim, as amostras são compostas por (*C*, *PS*,*PP*,*SP*), onde cada elemento representa um dos vetores de características apresentados. Existem dois tipos de amostras: Amostras de cadastramento ou amostra provindas da autenticação.

As amostras provenientes do cadastramento formam o conjunto de treinamento *S*. Uma vez que este conjunto está completo, um modelo pode ser gerado. Nesta dissertação, um conjunto de treinamento completo é composto por uma quantidade de até 30 amostras, pois percebeu-se que esta quantidade de amostras consegue modelar a dinâmica dos usuários ao digitar em um teclado numérico onde a *string* alvo é composta exclusivamente de números, conforme constataram também alguns autores [3], [46].

Desta maneira, o conjunto de treinamento da conta *w* é representado por $S_w = (s_1, s_2, \ldots, s_{30})$, onde cada *s* é uma amostra proveniente do cadastramento, que, por sua vez, é composta pelos vetores de características apresentados. As amostras presentes no conjunto de treinamento $s_1, s_2, \ldots, s_{30}$ referentes a uma mesma característica *x* são representadas por $s_1.x, s_2.x, \ldots, s_{30}.x$. A única restrição a ser ressaltada com relação às amostras que formam o conjunto de treinamento é que todas devem ter o mesmo vetor de características *C*, pois, como foi mencionado anteriormente, as outras características são extraídas em função das teclas utilizadas. Assim, caso uma amostra não cumpra esta restrição, o usuário deve digitar novamente a informação alvo para substituí-la.

As amostras provindas da autenticação formam o conjunto de autenticação *L*. O conjunto de *L* amostras de autenticação na conta *w* é representado por $L_w = l_1, l_2, \ldots, l_{60}$, onde cada $l_i$ é uma amostra provinda da autenticação e é composta por uma *sessão* e por seus respectivos vetores de característica. Cada *sessão* é composta de pelo menos 10 tentativas de autenticação. O número de sessões por usuário variou de acordo com a disponibilidade deles em participar do experimento, mas no mínimo 9 sessões por usuário foram feitas.

## 4.7 *Modelo*

Um modelo é gerado para cada conta cadastrada, contendo informações importantes que representarão o usuário em uma posterior sessão de autenticação. Para a conta *w*, com base no conjunto de treinamento $S_w$, o modelo $Z_w$ gerado é composto por informações extraídas dos quatro vetores

de características apresentados: *C*, *PS*, *PP*, *SP*. Com relação ao vetor de características *C*, a informação alvo extraída das amostras do conjunto de treinamento é o próprio vetor. Com relação aos demais vetores, as informações extraídas são a média e o desvio padrão dos dados. Desta maneira, o modelo *Z* é composto pelos seguintes elementos: $(C, \mu_{PS}, \sigma_{PS}, \mu_{PP}, \sigma_{PP}, \mu_{SP}, \sigma_{SP})$. Para um vetor de características *X* representativo dos vetores de características *PS*, *PP* e *SP*, o cálculo de $\mu_X = \mu_{x_1}, \mu_{x_2}, \ldots, \mu_{x_i}$ e $\sigma_X = \sigma_{x_1}, \sigma_{x_2}, \ldots, \sigma_{x_i}$ é feito de acordo com as equações (4.3) e (4.4):

$$\mu_{K_i(a)} = \frac{1}{N} \sum_{j=1}^{N} K_i(a, j) \tag{4.3}$$

$$\sigma_{K_i(a)} = \sqrt{\frac{1}{N-1} \sum_{j=1}^{N} \Big[ K_i(a, j) - \mu_{K_i(a)} \Big]^2} \tag{4.4}$$

onde N é o número de amostras para gerar o modelo. Uma vez calculados os valores de média e desvio padrão, repetimos o procedimento para remover valores considerados *outliers* [16]. *Outliers* "*o*" são valores que se desviam dos valores das características contidas nas amostras do conjunto de treinamento. Assim, as médias e os desvios padrões re-calculados após a eliminação dos *Outliers* irão refletir melhor as características de um usuário. Para uma característica $x_i$ existem *u* valores que não são *outliers*. Neste contexto, os cálculos dos novos valores da média e do desvio padrão seguem as equações:

$$o_{s_j x_i} = \begin{cases} \textit{falso}\,, & (\mu_{x_i} - 3\sigma_{x_i}) \leq s_j x_i \leq (\mu_{x_i} + 3\sigma_{x_i}) \\ \textit{verdadeiro}\,, & \text{c.c.} \end{cases} \tag{4.5}$$

$$\mu_{K_i(a)} = \frac{1}{N} \sum_{j=1}^{u} (s_j x_i \Big\| o_{s_j x_i} = \textit{falso}) \tag{4.6}$$

$$\sigma_{K_i(a)} = \sqrt{\frac{1}{N-1} \sum_{j=1}^{u} \Big[ s_j x_i - \mu_{K_i(a)} \Big]^2} \Big\| o_{s_j x_i} = \textit{falso} \tag{4.7}$$

## 4.8 Classificador

O classificador é responsável pela autenticação pessoal. Para o caso abordado nesta dissertação, é utilizada a autenticação do tipo verificação, ou seja, dada uma amostra de autenticação e uma conta, o classificador valida a identidade do usuário. Desta maneira, o classificador particiona o conjunto de autenticação $L_w$ em dois subconjuntos: $L_{wf}$ e $L_{wv}$, baseado em um valor de limiar (*threshold*) $\tau$. No subconjunto $L_{wv}$, estão as amostras verdadeiras e em $L_{wf}$ estão as amostras que o classificador considerou falsas. Esta partição é exemplificada na figura 4.3.

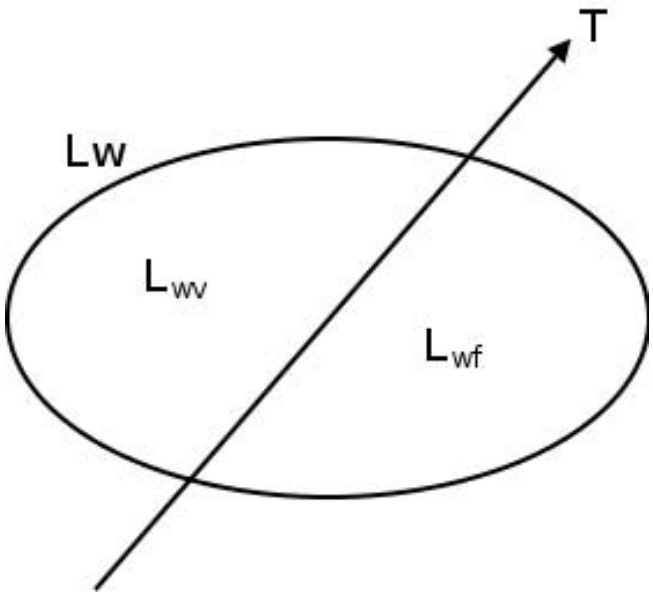


Fig. 4.3: Partição do conjunto de autenticação em dois subconjuntos: autênticos e impostores.

Inicialmente o vetor de características contendo os códigos ASCII é validado com o intuito de identificar a conta do usuário que clama por autenticação. Caso a amostra digitada possua uma conta associada ao vetor contendo os ASCII, o sistema então conduz o processo seguinte ao classificador. Na próxima sessão é descrito o classificador adotado e a implementação do mesmo relacionado a este trabalho.

### 4.8.1 Classificador usando Máxima Verossimilhança

Seja o vetor de características *X* para a conta *w* formado por variáveis aleatórias das características de digitação pertencentes a amostra de teste, onde as características estão representadas sequencialmente tal que $X = x_1, x_2, \ldots, x_i$ e seja $\mu_X = \mu_{x_1}, \mu_{x_2}, \ldots, \mu_{x_i}$ e $\sigma_X = \sigma_{x_1}, \sigma_{x_2}, \ldots, \sigma_{x_i}$ os vetores de média e desvio padrão do *template* para a conta *w*. O vetor $Q = q_1, q_2, \ldots, q_i$ representa os valores obtidos a partir da função densidade de probabilidade para cada observação, estimados usando a equação que calcula a função densidade de probabilidade para uma variável aleatória gaussiana unidimensional [4], descrita por (4.8):

$$f_x(x_i) = \frac{1}{\sqrt{2\pi\sigma_i}} exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right) \tag{4.8}$$

De posse dos valores de probabilidade das observações de teste, o valor de máxima verossimilhança entre a amostra e o modelo é dado por [6]:

$$L(X|w) = \prod_{Q=q_1}^{q_i} f_x(x_i) \tag{4.9}$$

A estimação da máxima verossimilhança $X$ em relação à $w$ é, por definição, o valor que maximiza $f_x(x_i)$. Intuitivamente, corresponde ao valor de $w$ que em algum sentido concorda melhor com as amostras atualmente observadas.

Se o valor de máxima verossimilhança pertencer ao intervalo $\tau_1(w) \leq L(X|w) \leq \tau_2(w)$, onde $\tau_1(w)$ e $\tau_2(w)$ são valores de limiares pré-definidos para a conta *w*, então o usuário é autenticado, caso contrário é considerado impostor e então rejeitado. Neste trabalho, os limiares são obtido durante o treinamento e a partir das *N* amostras que compôem o conjunto de treinamento como segue:

$$\tau_1(w) = \mu_{L(X|w)} - 3\sigma_{L(X|w)} \tag{4.10}$$

e

$$\tau_2(w) = \mu_{L(X|w)} + 3\sigma_{L(X|w)} \tag{4.11}$$

onde

$$\mu_{L(X|w)} = \sum_{w=1}^{N} \frac{L(X|w)}{N} \tag{4.12}$$

e

$$\sigma_{L(X|w)} = \sqrt{\frac{1}{N-1} \sum_{w=1}^{N} \Big[\mu_{L(X|w)} - L(X|w)\Big]^2} \tag{4.13}$$

## 4.9 Atualização do Modelo

Os modelos estimados para cada usuário devem sempre representar os usuários proprietários das contas. Porém, como a digitação tende a se modificar gradualmente com as autenticações, uma atualização destes modelos deve ser realizada. Um mecanismo de atualização dos modelos é adotado neste trabalho, o qual baseou-se nos trabalhos de [15] e [14]. Este mecanismo consiste em substituir a amostra mais antiga contida no conjunto de treinamento pela amostra mais recente provinda da autenticação. Este mecanismo é aplicado quando uma autenticação é realizada com sucesso. De posse do novo conjunto de treinamento, então partimos para executar novamente o treinamento a partir do novo conjunto.

As características extraídas da dinâmica da digitação sofrem influências temporais, e um meca-

nismo de adaptação torna-se uma obrigatoriedade para a manutenção dos modelos atualizados e do desempenho afirmativo na autenticação de usuários com o decorrer do tempo. No próximo capítulo são discutidos os experimentos conduzidos usando o classificador proposto, bem como os resultados obtidos com a base de teste.

# Capítulo 5

# Experimentos e Resultados

Neste capítulo apresentam-se os resultados obtidos com a metodologia discutida no capítulo anterior. Todos os experimentos combinam as três características apresentadas, visto que em [38] os autores perceberam que somente com a combinação delas podemos representar a dinâmica dos usuários e construir sistemas com melhores desempenhos.

Inicialmente discute-se o processo de composição de uma base para testes. Em seguida discute-se os experimentos realizados com seus respectivos resultados na base de dados coletada. Finalmente apresenta-se aspectos de implementação para plataformas que operam sobre aritmética de ponto-fixo - muito comum em tecnologias móveis - e novos testes foram executados afim de medir o desempenho do sistema após a restrição de plataforma.

## 5.1 Aquisição de Amostras

Ao contrário de outras características biométricas - como por exemplo as faces [13] que possuem bases de dados públicas para avaliação de desempenho dos algoritmos, dinâmica da digitação ainda não possui esse recurso disponível. Logo, a definição de uma base de dados foi fundamental para a avaliação experimental da metodologia proposta nesta dissertação.

Em muitos problemas de classificação de padrões, técnicas de amostragem são utilizadas para extrair um subconjunto de elementos de uma população que seja representativo, no sentido de que as propriedades obtidas a partir da observação de certas variáveis ou características deste subconjunto possam ser extrapoladas para a população como um todo.

Para se fazer tais inferências, é interessante selecionar um método de amostragem apropriado que leve em conta a possibilidade de todos os elementos da população serem parte da amostragem, ou então, de apenas alguns destes elementos fazerem parte dela. Se todos os componentes da população tiverem igual probabilidade de participar da amostragem, diz-se que o método usado é o da

amostragem causal, caso contrário, fala-se de amostragem não causal [15].

Vários critérios podem ser utilizados num método de amostragem não causal para garantir que ela não seja tendenciosa ou não representativa da população. No entanto, quando este tipo de informação não está disponível ou é dificil de ser obtida, a adoção de tais critérios torna-se proibitiva. Nesta situação, uma opção seria obter uma amostragem de conveniência, ou seja, uma amostragem que esteja naturalmente disponível e que não dependa de critérios complexos para a seleção de seus elementos. Assim, o espaço amostral poderia ser o local de trabalho, a universidade, uma cidade, entre outros.

Em virtude da dificuldade de acesso à população envolvida com o problema de verificação pessoal via dinâmica da digitação, adotou-se um método de amostragem de conveniência para a composição da base de dados de dinâmica da digitação. A Faculdade de Engenharia Elétrica e de Computação (FEEC) da Universidade Estadual de Campinas (UNICAMP) foi o local de conveniência escolhido para a coleta das amostras.

Diante das decisões tomadas, faz-se necessário fornecer respostas a algumas perguntas importantes:

1. Que tipos de objetos deverão constituir a base de dados?

2. Qual deverá ser o número de classes (contas) e a quantidade de padrões (amostras) por classe?

A resposta à pergunta 1 é direta: os objetos da base de dados são as digitações provindas de usuários juntamente com as características de interesse extraídas destas digitações. Vale ressaltar que as características são extraídas dos dados de digitação de uma *string* alvo em tempo real. Para a construção da base de testes, usou-se um teclado numérico onde dez amostras com oito caracteres cada são coletadas em cada sessão e para cada tipo de usuário. A fim de permitir a avaliação da metodologia proposta, além de digitações provindas de usuários legítimos, deverão ser incluídas digitações provindas de usuários impostores. As amostras dos usuários foram coletadas em diferentes períodos, e nunca ao mesmo tempo, intercalando um período médio de uma semana entre as sessões.

As figuras 5.1, 5.2 e 5.3 exemplificam o comportamento das características *PP*, *SP* e *PS*, respectivamente, para um dado usuário presente na base de dados. O comportamente de amostras genuínas e impostoras também aparecem nas figuras, onde é possível perceber quais são as características que melhor caracterizam o usuário.

Respondendo a pergunta 2, a definição da quantidade ideal de classes a serem consideradas em uma amostragem, mesmo em problemas de reconhecimento de padrões, não segue uma regra muito precisa. Neste contexto, costuma-se citar um conjunto de amostras pequeno ou grande. Dentro do escopo deste trabalho e do tempo disponível, escolheu-se **26** como o número total de classes constituindo a base de dados, sendo que participaram do experimento homens e mulheres de várias

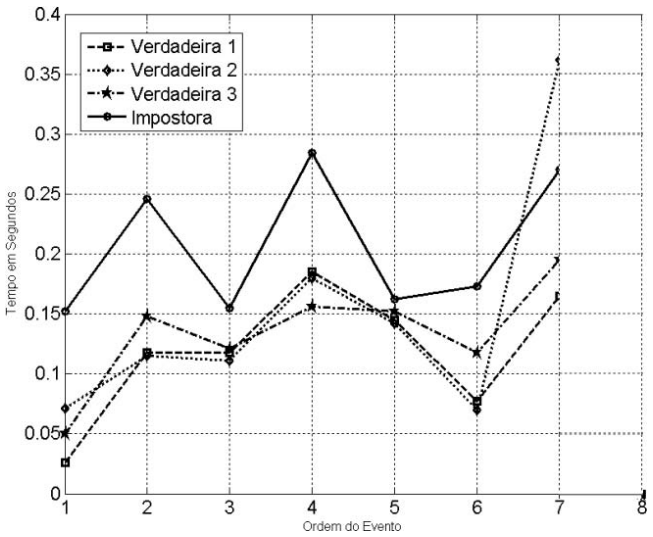


Fig. 5.1: Exemplo de amostras legítimas e impostoras da característica pressiona-pressiona.

faixas etárias e com diferentes níveis de familiaridade com o teclado numérico, usando como *string* alvo uma combinação de oito números a sua escolha. Duas situações foram observadas:

- *Usuários Autênticos*: O usuário tenta autenticar-se em sua conta. Foram coletadas dez amostras de cada usuário e em cada uma das 9 seções, totalizando 2340 amostras de usuários autênticos.
- *Usuários Impostores*: Impostores foram convidados a tentar autenticar-se nas contas dos usuários autênticos. Cada conta foi atacada 60 vezes, resultando em 1560 amostras de impostores.

A seguir são apresentados os experimentos realizados na base de dados com as amostras coletadas.

## 5.2 Experimentos

Para analisar o desempenho da metodologia apresentada em verificar corretamente a identidade dos usuários proprietários das contas, experimentos foram realizados na base de dado. Para uma sessão de autenticação, quatro situações de classificação podem ocorrer:

1. A sessão é de um usuário legítimo e o classificador considera-o legítimo;
2. A sessão é de um usuário impostor e o classificador considera-o impostor;
3. A sessão é de um usuário legítimo e o classificador considera-o impostor;
4. A sessão é de um usuário impostor e o classificador considera-o legítimo;

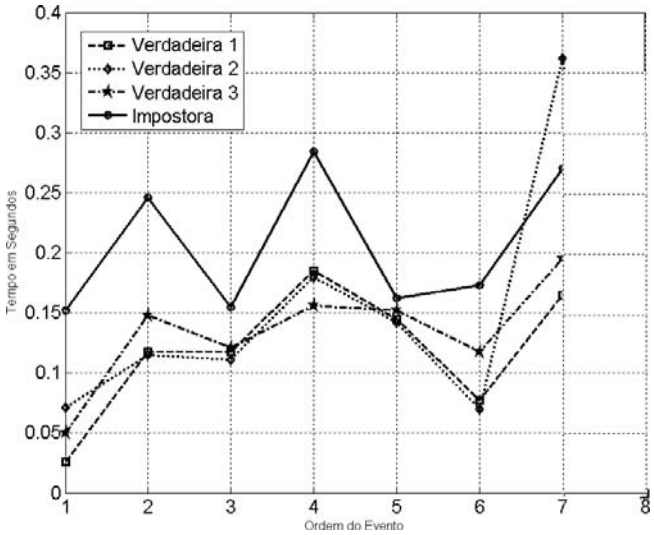


Fig. 5.2: Exemplo de amostras legítimas e impostoras da característica solta-pressiona.

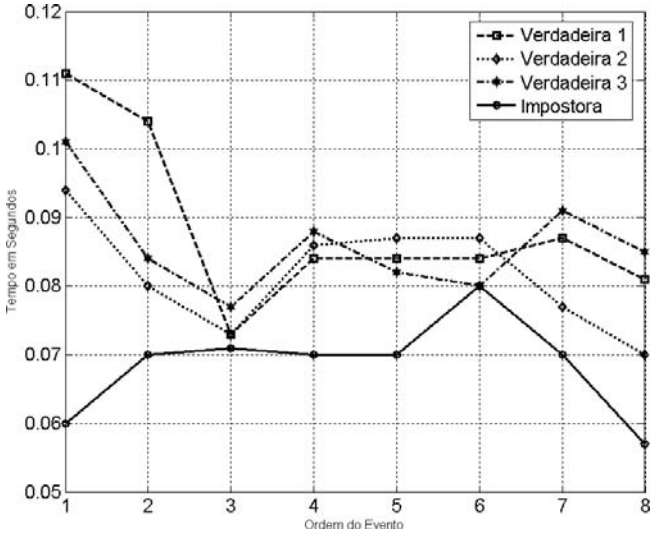


Fig. 5.3: Exemplo de amostras legítimas e impostoras da característica pressiona-solta.

Nas situações 3 e 4, como pode ser observado, o classificador cometeu erros de classificação. Nestes casos, o desempenho da metodologia é medido pelas taxas de *FAR* e *FRR*, que foram discutidas anteriormente. Estas taxas são calculadas através das equações 5.1 e 5.2:

$$FAR = \left( \frac{\textit{quantidade de sessões enquadradas na situação 3}}{\textit{quantidade total de sessões}} \right) \% \tag{5.1}$$

$$FRR = \left( \frac{\textit{quantidade de sessões enquadradas na situação 4}}{\textit{quantidade total de sessões}} \right) \% \tag{5.2}$$

### 5.2.1 Combinação de características

Foram analisadas as combinações de vetores de características quando $N = 25$ amostras na geração do modelo. A primeira e única característica que tornou-se fixa em todos os experimentos é a *string* alvo, pois é graças a ela que o processo de identificação é conduzido a uma comparação um-para-um. As outras combinações de vetores de características são:

I. Combinando a *string* alvo e *K*={PS, SP};

II. Combinando a *string* alvo e *K*={PS, PP};

III. Combinando a *string* alvo e *K*={PP, SP};

IV. Combinando a *string* alvo e *K*={PP, PS, SP}.

Tab. 5.1: *FAR* e *FRR* obtidos nos experimentos propostos.

| Experimento | FAR(%) | FRR(%) |
|---|---|---|
| (I) | 3.27 | 3.01 |
| (II) | 5,75 | 6.15 |
| (III) | 4.75 | 5.75 |
| (IV) | 2.19 | 1.22 |

A tabela 5.1 apresenta as taxas de *FAR* e *FRR* para os resultados experimentais obtidos com a combinação de características. Então, observou-se que:

- A combinação que resulta nas melhores taxas é a utilização das características *PP*, *PS* e *SP* no experimento (IV): 2.19% FRR e 1.22% FAR.

- A segunda melhor combinação deu-se com as características *PS* e *SP* no experimento (I): 3.27% FRR e 3.01% FAR.

Tab. 5.2: Comparação do *EER* de outras propostas na literatura internacional com a metodologia proposta.

| Artigo | EER(%) |
|---|---|
| Ord et al [3] | 19.95 |
| Araújo et al [38] | 1.6 |
| Costa et al [41] | 3.6 |
| Experimento (IV) | 3.52 |

A tabela 5.2 compara o valor de *EER* obtidos no experimento (IV) com outras propostas encontradas na literatura internacional. Em [3] os autores propuseram o uso de redes neurais para distinguir usuários autênticos de impostores. Cada amostra era composta de 6 caracteres e somente as características PS e SP foram usadas. A rede neural foi treinada usando o algoritmo de *back-propagation* [4].

Em [38] foi proposto o uso de um classificador utilizando a distância mínima (distância euclidiana). Originalmente a metodologia dos autores foi desenvolvida para ser usada em senhas compostas por caracteres alfanuméricos e a *string* alvo é, em média para todos os usuários, maior que 15 caracteres. O valor de EER descrito na tabela 5.2 reflete os experimentos dos autores, porém durante o desenvolvimento deste trabalho a metodologia proposta em [38] foi reproduzida da mesma forma como foi descrita no artigo dos autores, porém na base de dados coletada e mencionada anteriormente neste trabalho. Para efeito de comparação, a metodologia de Araújo *et al* [38] em um experimento diretamente aplicado na base coletada neste trabalho, apresentou EER = 6.2%.

Em [41] foi proposto para utilizar modelos de Markov para reconhecer usuários em teclado numérico. O número de amostras para criar o modelo foi de 30 e somente as características PS e SP foram usadas. Para mais detalhes sobre a técnica adotada, vide Apêndice B. Observou-se que o Experimento (IV) apresenta melhores resultados quando comparado com [41], pois deve-se ao fato de que, neste trabalho, foi adotado uma técnica para selecionar *outliers* conforme [16], permitindo construir modelos de usuários que representam melhor as características dos usuários. No futuro pretende-se utilizar esta técnica para reproduzir o experimento com modelos de Markov.

### 5.2.2 Amostras por modelo

A quantidade de padrões ou amostras por classe que vão formar os conjuntos de treinamento e de autenticação foi alvo de investigação deste trabalho. Em [46] e [3] os autores perceberam que a dinâmica da digitação em teclados numéricos necessita de mais amostras na geração dos modelos, apesar de mais simples para o usuário utilizar este tipo de teclado. Na abordagem usando teclados convencionais, os usuários utilizam duas mãos para digitar, portanto mais informação são obtidas e

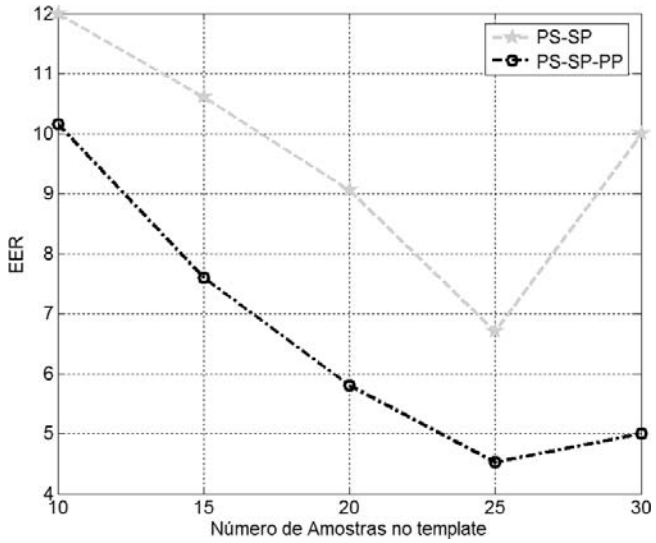


Fig. 5.4: Número de amostra por *template*.

não somente padrões de digitação de uma única mão, mais o resultado da interação entre as mãos. Sendo assim, um primeiro experimento foi conduzido para avaliar quantas amostras seriam necessárias para compor o modelo do usuário. Utilizando como parâmetro a taxa de *EER*, os seguintes experimentos com $N$ amostras e $K$ características foram feitos:

1. Utilizando $N$={10,15,20,25,30} para gerar o modelo com $K$={SP, PS};

2. Utilizando $N$={10,15,20,25,30} para gerar o modelo com $K$={PP, SP, PS};

Na figura 5.4 podemos perceber que 25 amostras são suficientes para criar os modelos dos usuários para a base coletada, tal que minimizamos as taxas de *EER*, conforme constatado também por [3].

### 5.2.3 Quantidade de caracteres

Em relação a quantidade de caracteres da informação alvo, foi feita uma restrição: ela deve conter pelo menos 8 caracteres. Para verificar o impacto da quantidade de caracteres, um experimento é realizado reduzindo a quantidade de caracteres das informações alvo da base de dados. Na figura 5.5 é mostrado o comportamento do EER com a redução da quantidade de caracteres. A taxa de EER tende a aumentar sensivelmente quando o número de caracteres diminue. A partir de 4 caracteres, a taxa de erro tende a cair. Este fato também foi constatado por [3].

Outro aspecto interessante é que um número maior de caracteres, tende a reduzir ainda mais o valor de EER. Em [38] os autores constataram isso. Porém, vale ressaltar que esta metodologia

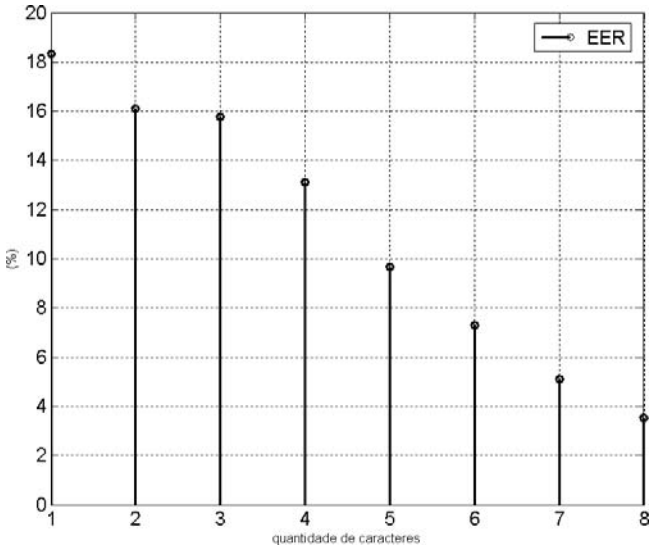


Fig. 5.5: Comportamento da EER com a quantidade de caracteres.

aplicou-se a um conjunto de caracteres composto somente por números, e qualquer valor acima de 8 caracteres torna a informação alvo mais difícil de ser lembrada.

## 5.3 Implementação Eficiente

Sistemas de autenticação biométrica são tecnologias computacionais que usam intensivamente operações de ponto flutuantes. Sendo assim, é difícil que sejam imediatamente embarcadas em dispositivos como celulares, pois devido às restrições de bateria e a falta de um co-processador matemático, toda a computação de ponto flutuante em celulares é feita através de emulação, comprometendo inclusive o consumo de bateria.

Tipos de dados de ponto fixo (também conhecidos como fracionários) são a solução mais eficiente para quem programa para dispositivos móveis, pois, enquanto pode-se executar a aritmética fracionária usando inteiros, evita-se os algoritmos complexos necessários para operações de ponto flutuante e que tendem a fazer com que o custo dos processadores de celulares tornem-se elevados, encarecendo o valor do produto final. A aritmética fracionária ou a aritmética de ponto fixo é, conseqüentemente, mais barata de executar do que a aritmética de ponto flutuante, e que, não obstante, respeita as limitações de hardware de dispositivos móveis.

Um número de ponto fixo é uma maneira de representar os bits das partes inteiras e fracionárias de um número em um valor integral. A conversão para a representação fracionária é feita para o processador da máquina. A Figura 5.6 demonstra a representação de um número de ponto fixo no

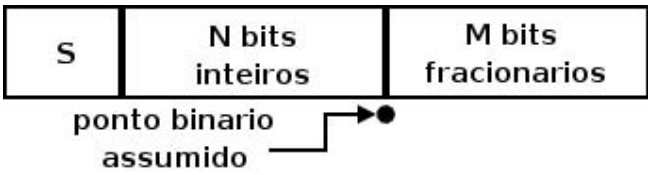


Fig. 5.6: representação de um número de ponto fixo no formato *S.N.M*.

formato *S.N.M*, onde S é o bit de sinal, N representa os bits da parte inteira e M representa o número da parte fracionária. A notação implica na posição de um ponto binário indicando onde começa a parte fracionária e onde termina a parte inteira, porém nenhum bit extra é usado para representar isso.

Bits à direita do ponto compreendem a parte fracionária do número de ponto flutuante, e estes bits atuam como um peso de potência negativa de dois. Bits à esquerda do ponto representam a parte inteira do número de ponto flutuante, e consequentemente atuam como um peso de potência positiva de 2. Para a representação do número de ponto fixo acima, a escala compreende o intervalo dinâmico $[-2^N, 2^N]$ e uma precisão de $2^{-M}$.

Como exemplo, considere a representação do número $S.2.13$. O número total de bits requeridos para armazenar é de 1+2+13 = 16 bits, sendo de fácil representação em um tipo de dados inteiro de 16 bits. O intervalo dinâmico para esta palavra pertence a $[-4, 4]$ (2 bits inteiros) e precisão de 1/8192, isto é, $2^{-13}$ onde 13 é o número de bits na parte fracionária.

A técnica adotada neste trabalho para converter um número em ponto fixo é multiplicar este número por $2^M$, onde M é o número de bits requeridos para representar a parte fracionária; $M = 13$ é usado ao longo do experimento. Exemplificando: suponha que se queira converte o número decimal $2.45_{10}$. Sua representação binária seria $10.0111001100110_2$; para obtermos este número na representação $S.2.13$, multiplicamos sua representação binária por $2^{13}$, ou seja, $10.0111001100110_2 * 2^{13} = 100111001100110_2 = 20070_{10}$.

A conversão para uma representação usando aritmética de ponto fixo tem um custo computacional, mas lembramos que este custo ainda será menor que se fizermos o sistema usar a emulação de operações de ponto flutuante do sistema operacional do dispositivo. Isso sem contar com a economia de bateria proporcionada.

Tendo em vista a restrição citada, repetimos o experimento (IV) adotando a técnica de operação de pontos flutuantes com a precisão de $S.2.13$. A figura 5.7 apresenta os resultados para o experimento (IV) com e sem a restrição. Pode-se perceber que esta restrição não compromete o desempenho do sistema, pois em ambos os casos o sistema comporta-se dentro de um intervalo de confiança onde a diferença de 3% pouco compromete o desempenho geral do sistema. No futuro um estudo mais detalhado e testes com outras precisões devem corrigir este problema.

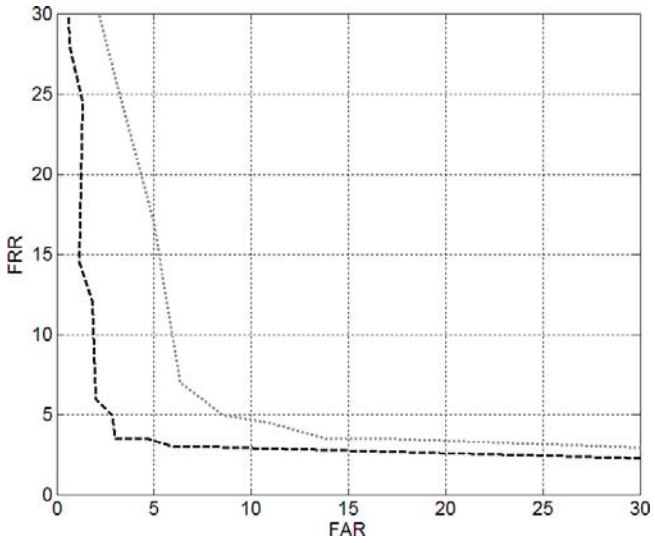


Fig. 5.7: Curvas ROCs para o experimento (IV). O gráfico em linha pontilhada possui ponto de EER em 6.67% e representa o resultado com a restrição de ponto-fixo, e o gráfico em linha tracejada a EER=3.52% para o sistema sem a restrição

## 5.4 Discussão

De acordo com os experimentos realizados, algumas observações podem ser feitas:

- O classificador proposto nesta dissertação provê bons resultados quando aplicado ao problema de dinâmica da digitação em teclados numéricos.

- A combinação da *string* alvo e das características $K$={PP, PS, SP} provê os melhores resultados, quando comparados com outras combinações, refletindo melhor a maneira como cada usuário digita, uma constatação já alcançada em trabalhos anteriores, porém nunca testadas em teclados numéricos.

- A escolha da *string* alvo é um ponto importante, pois permite que o usuário esteja familiarizado com a senha adotada e faz com que impostores sejam menos aceitos no sistema. A única restrição é que a *string* alvo seja de no mínimo 8 caracteres pois a literatura em dinâmica da digitação recomenda que quanto maior a senha, menores são as chances dos impostores burlarem a segurança.

- A quantidade de amostras presentes no conjunto de treinamento é um aspecto importante da metodologia, pois quanto maior for a quantidade de amostras, a média e o desvio padrão serão mais confiáveis com relação as características de digitação do usuário, porém a aceitabilidade

pública da metodologia diminue. No entanto, quanto menor for a *string* alvo, maior deverá ser o número de amostras para compor esse modelo, conforme constatou [3].

- A precisão na captura dos tempos adotada estava compatível com a ordem de grandeza das características extraídas. Além disso, experimentos usando uma implementação eficiente com operações de ponto flutuante torna a metodologia proposta compatível com dispositivos móveis, sem contudo comprometer o desempenho geral.

- O mecanismo de adaptação é outro ponto importante, cuja finalidade é manter sempre atualizados os modelos com as mudanças no rítmo que venham a ocorrer na digitação de cada usuário.

- A determinação do limiar de cada usuário utilizando as amostras contidas no modelos corresponde a uma proposta inovadora na literatura da dinâmica da digitação, visto que permite determinar o ponto de decisão onde o sistema decide se uma dada amostra pertence a um usuário ou a um impostor sem, contudo, um conhecimento prévio da dinâmica de impostores para uma dada conta. Em trabalhos futuros pretende-se melhorar ainda mais esta técnica, onde uma análise detalhada das características determinará um conjunto de limiares que poderão modelar melhor os usuários.

# Capítulo 6

# Conclusões e trabalhos futuros

Este trabalho apresentou uma metodologia de autenticação biométrica através das características extraídas da dinâmica da digitação em teclados numéricos. Esta metodologia foi desenvolvida com base em trabalhos já publicados na área e que foram discutidos no capítulo 1, além de vários experimentos que foram realizados ao longo deste trabalho cujo intuito era validar a aplicação em controle de acesso a áreas restritas ou como proposta para autenticação em dispositivos móveis celulares, ou ainda aumentar a segurança em transações bancárias.

Várias contribuições foram produzidas. Uma delas é relativa a revisão do estado da arte, enquanto que as demais são relativas aos experimentos que realizamos ao longo deste trabalho, onde analisamos mudanças em diferentes aspectos abordados na metodologia. Podemos citá-las abaixo:

- A comparação entre combinações de características. Neste contexto, foram comparados e analisados os resultados obtidos com a utilização de vetores de características isoladas ou a combinação entre eles.

- A utilização dos códigos das teclas como um vetor de característica é uma das inovações deste trabalho, o que permtiu adotarmos uma metodologia de autenticação partindo diretamente a uma comparação 1-para-1.

- Outra inovação é a combinação de dois tipos de latências de teclas, que foram expressadas nas características PP e SP, com a duração das teclas PS, e analisar essa combinação em teclados numéricos.

- O conjunto de características que analisamos e que apresentou os melhores resultados nos experimentos realizados é composto pelos vetores PP, SP e PS - e o vetor contendo os códigos ASCII das teclas.

- O fato de permitir aos usuários que escolham a *strings* alvo favoreceu para se obter melhores resultados, visto que assim os usuários mantem-se familiarizados com suas senhas, diminuindo as chances de ocorrerem altas variabilidades intra-classe.

- Observou-se que o número de caracteres na *string* alvo, quanto maior for, melhor será o resultado do classificador.

- A comparação entre as quantidades de amostras do conjunto de treinamento, onde observou-se que o EER aumenta com a redução da quantidade de amostras para valores menores que 25.

- A utilização de um mecanismo de combate a *outliers* tende a favorecer o procedimento de construção dos modelos, visto que assim a variância do modelo - fator de suma importância para combater impostores - tende a representar somente a dinâmica da digitação do usuário, diminuindo as condições de ataque por impostores.

- A utilização de um mecanismo de atualização dos modelos com as mudanças ocorridas na digitação. Foi discutido que este mecanismo é de suma importância para os resultados, pois as característica biométricas extraídas da dinâmica da digitação possuem uma variação ao longo do tempo. Este mecanimos já havia sido investigado em [38], mas até então não havia sido estudado em teclados numéricos.

- A determinação do limiar de decisão ótimo para cada usuário em função de suas amostras do modelo foi outra contribuição deste trabalho, e não havia sido proposto em problemas de dinâmica da digitação até então.

- As taxas obtidas (2.1% de FAR e 1.2% de FRR) na metodologia são competitivas com as obtidas em trabalhos publicados nesta área, quando observamos os dados oriundos da digitação em teclados numéricos.

## 6.1 Trabalhos Futuros

Para trabalhos futuros, pretende-se aumentar a população de usuários e capturar mais seções dos mesmos. Além disso, pretendemos adotar um classificador baseado em misturas de gaussianas [5] e um mecanismo de adaptação de modelos será proposto para este classificador, afim de baixar ainda mais a *EER* do sistema. Outro aspecto a ser utilizado é a forma de obtenção do número de gaussianas por características, na qual deve-se utilizar técnicas de seleção de estrutura de modelos [47], onde investigaremos algoritmos para encontrar o melhor número de gaussianas para cada característica modelada.

Outra intenção é adaptar a metodologia para o propósito criptográfico. Desta forma geraríamos uma chave criptográfica baseada nas características de dinâmica da digitação do usuário [14]. Isto resultaria em uma criptografia denominada *bio-key* que agregaria em segurança para aplicações que fazem uso de criptografia.

# Referências Bibliográficas


[1] International Association for Biometrics. **Glossary of Biometric Terms**. http://www.iafb.org.uk/DOC/GlossaryTerms.htm, 1999. Online, accessed 11-March-2005.

[2] A. Ross A. K. Jain and S. Prabhakar. **An Introduction to Biometric Recognition**. *IEEE Trans. on Circuit ans Systems for Video Technology*, 14(1):4–20, January 2004.

[3] T. Ord and S. M. Furnell. **User authentication for keypad-based devices using keystroke analysis**. volume 1, pages 263–272, 2000.

[4] S. Hayken. ***Redes Neurais Artificiais: Princípios e Prática***. Editora Bookman, 2001.

[5] P. E. Hart R. O. Duda and D. G. Stork. ***Pattern Classification***. Wiley-Interscience Publication, 2000.

[6] R. O. Duda and P. E. Hart. ***Pattern Classification and Scene Analysis***. Wiley-Interscience Publication, 1973.

[7] J. Kang K. Kum and W. Sung. **AUTOESCALER for C: An optmizing floating-pont to integer C program converter for fixed-point digital signal processing**. *IEEE Transactions on Circuits and Systems for Video Technology - II: Analog and Digital Signal Processing*, 47(9), 2000.

[8] Wikipedia. **Biometrics – Wikipedia, the free encyclopedia**. http://en.wikipedia.org/wiki/Biometric, 2005. Online, accessed 21-April-2005.

[9] J. Wayman. ***Biometric Systems***. Springer, 2005.

[10] INC Biometric Technology. **How Accurate is the Biometric?** http://bio-tech-inc.com/, 2003. Online, accessed 02-April-2005.

[11] L. Hong. **Automatic Personal Identification Using Fingerprints**. Technical report, Michigan State University, June 1998.

[12] J. Daugman. **How Iris Recognition Works**. *IEEE Trans. on Circuits and Systems for Video Technology*, 14(1):21–30, 2004.

[13] P. J. Rauss P. J. Phillips, H. Moon and S. Rizvi. **The FERET evaluation methodology for face recognition algorithms**. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(10), October 2000.

[14] F. Monrose and A. D. Rubin. **Keystroke Dynamics as a Biometric for Authentication**. *Future Generation Computer Systems*, 16(4):351–359, March 1999.

[15] Livia C. F. Araujo. **Uma Metodologia para Autenticação Pessoal baseada na Dinâmica da Digitação**. Tese de mestrado, Faculdade de Engenharia Elétrica e de Computação, UNICAMP, Fevereiro 2004.

[16] R. Joyce and G. Gupta. **Identity authentication based on keystroke latencies**. *Communications of the ACM*, 33(2):168–176, March 1990.

[17] X. Ke A. Peacock and M. Wilkerson. **Typing Patterns: A key to user identification**. *IEEE Security & Privacy*, pages 40–47, January 2004.

[18] A. Guven and I. Sogukpinar. **Undestanding users keystroke patterns for computer access security**. *Computers & Security*, 22(8):695–706, July 2003.

[19] M. Pusara and Carla E. Brodley. **User re-authentication via mouse movements**. pages 1–8, 2004.

[20] J. Leggett et alli. **Dynamic identity verification via keystroke characteristics**. *International Journal of Man-Machine Studies*, 35:859–870, 1990.

[21] D. Umphress and G. Williams. **Identity verification through keystroke characteristics**. *International Journal of Man-Machine Studies*, 23:263–273, 1985.

[22] J. Leggett and G. Williams. **Verify identity via keystroke characteristics**. *International Journal of Man-Machine Studies*, 28:67–76, 1987.

[23] D. Bleha and M. Obaidat. **Dimensionality reduction and feature extraction applications in identifying computer users**. *IEEE Transactions on System, Man and Cybernetics*, 21(2):452–456, April 1991.

[24] M. S. Obaidat and B. Sadoun. **Verification of computer user using keystroke dynamics**. *IEEE Transactions on Syst., Man, Cybern.*, 27(2):236–241, Mar.-Apr. 1997.

[25] F. Monrose and A. Rubin. **Authetication via keystroke dynamics**. volume 1, pages 48–56, 1997.

[26] BioPassword INC. **BioPassword Family of Products**. http://www.biopassword.com, 2005. Online, accessed 11-January-2005.

[27] J. M. badfa O. Coltell and G. Torres. **Biometric identification system based in keyboard filtering**. pages 203–209, 1999.

[28] C. Slivinsky S. Bleha and B. Hussain. **Computer-access security systems using keystroke dynamics**. *IEEE Transactions on Pattern Anal. and Machine Intelligence*, 12(12):1217–1222, December 1990.

[29] D. Gunetti and C. Picardi. **Keystroke analysis of free text**. *ACM Transactions on Information and System Security*, 8(3), 2005.

[30] D. T Lin. **Computer-access authentication with neural network based keystroke identity verification**. volume 1, pages 174–178, 1997.

[31] W. G. de Ru and J. H. P. eloff. **Enhanced password authentication through fuzzy logic**. *IEEE Expert*, 17(6):38–45, Nov.-Dec. 1997.

[32] Vir. V. Phoha Y. Sheng and S. M. Rovnyak. **A parallel decision tree-based method for user authentication based on keystroke patterns**. *IEEE Transactions on System, Man and Cybernetics - Part B: Cybernetics*, 35(4):826–833, August 2005.

[33] J. A. Michael J. A. Robison, V. M. Liang and C. L. MacKenzie. **Computer user verification using login string keystroke dynamics**. *IEEE Transactions on Syst., Man, Cybern.*, 28(2):236–241, Mar.-Apr 1998.

[34] D. Mahar *et al*. **Optimizing digraph-latency based biometric typist verification systems: inter and intra typist differences in digraph latency distributions**. *International jornal Human-computer studies*, 43:579–592, June 1995.

[35] A. Abbas S. Haidar and A. K. Zaidi. **A multi-technique approach for user identification trhough keystroke dynamics**. volume 2, pages 1336–1341, 2000.

[36] E. Yu and S. Cho. **Keystroke dynamics identity verification - its problems and practical solutions**. *Computers & Security*, 23:428–440, January 2004.

[37] A. F. Ismail L. W. Kin F. W. M. H. Wong, A. S. M. Supian and O. C. Soon. **Enhaced user authentication through typing biometrics with artificial neural network and k-nearest neighbor algorithm**. In *in Conf. Rec. 35th Asilomar Conf. Signals, Syst., comput.*, pages 911–915, 2001.

[38] L.C.F. Araujo *et al*. **User authentication through typing biometrics features**. *IEEE Transactions on Signal Processing*, 53(2), 2005.

[39] L. C. F. Araújo *et al*. **A fuzzy logic approach in typing biometrics user authentication**. pages 1038–1051, 2003.

[40] F. J. Gutierrez et al. **Biometric and Data Mining: Comparison of data mining-based keystroke dynamics methods for identity verification**. In *in Proc. MICAI 2002 - LNAI*, number 2313, pages 460–469, 2002.

[41] C.R. do N. Costa J.B.T. Yabu-uti F. Violaro L.L. Ling R.N. Rodrigues, G.F.G. Yared. **Biometric Access Control through Numerical Keyboard based on Keystroke Dynamics**. In *Proceedings of the Second International Conference o Biometric - ICB'2006*, January 2006.

[42] R.N. Rodrigues J.B.T. Yabu-uti F. Violaro L.L. Ling C.R. do N. Costa, G.F.G. Yared. **Autenticação Biométrica via Dinâmica da Digitação em Teclados Numéricos**. In *Anais do XXII Simpósio Brasileiro de Telecomunicações - SBrT 05*, Setembro 2005.

[43] W. Chen and W. Chang. **Applyin hidden markov models to keystroke pattern analysis for password verification**. In *in Proc. of the 2004 IEEE international conference on information reuse and integration - IRI 2004*, number 1, pages 467–474, November 2004.

[44] D. Hogarth. **TIMESERV for Microsoft Windows NT Resource Kit**. http://www.niceties.com/timeserv.html, 2001. Online, accessed 21-June-2005.

[45] Intel. **Using de RDTSC Instruction for Performance Monitoring**. http://developer.intel.com/drg/pentiumII/appnotes/RDTSCPM1.HTM, 1997. Online, accessed 01-April-2005.

[46] P.L. Reynolds N.L. Clarke, S. M. Furnell and P.M. Rodwell. **Advanced Subscriber Authentication Approaches for third Generation Mobile Systems**. volume 1, pages 319–323, 8-10 May 2002.

[47] M. A. T Figueiredo and A. K. Jain. **Unsupervised learning of finite mixture models**. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(3):381–396, 2002.

# Apêndice A

# Como obter o contador do *Time-Stamp*

Este apêndice pretende documentar o processo de captura do tempo no sistema proposto. Desta forma tornamos o processo menos árduo para os trabalhos futuros, visto que aplicações de dinâmica da digitação são extremamente dependentes da precisão do tempo e nenhum dos trabalhos pesquisados detalhava esse processo. Todos os trechos de códigos detalhados aqui foram testados para processadores da família Intel e semelhantes - ou seja, processadores x86.

Processadores da família Intel Pentium possuem uma instrução para ler o valor corrente do contador time-stamp a partir dos registradores (edx:eax). Esta instrução retorna uma variável de 64-bits. O contador do *Time-Stamp* é incrementado sempre que ocorre um ciclo do processador. Em um processador de 1GHz (1000MHz), o contador do *Time-Stamp* é incrementado $10^9$ vezes por segundo. Desta forma podemos medir o tempo de uma maneira mais exata do que confiando no relógio do sistema.

O seguinte arquivo de cabeçalho faz a leitura dos registradores e deve ser incluído na função implementada para ler o contador:

```
00 /* --------------------- file: rdtsc.h */
01 #ifndef __RDTSC_H_DEFINED__
02 #define __RDTSC_H_DEFINED__
03
04 /* ---------------------------------- */
05 #if defined(__i386__) || defined(__x86_64__)
06 static __inline__ unsigned long long int rdtsc(void)
07 {
08   unsigned long long int x;
09      __asm__ volatile (".byte 0x0f, 0x31" : "=A" (x));
10       return x;
11 }
```

```
12
13 #elif defined(__powerpc__)
14 static __inline__ unsigned long long int rdtsc(void)
15 {
16  unsigned long long int result=0;
17  unsigned long int upper, lower,tmp;
18  __asm__ volatile(
19                "loop:                  \n"
20                "\tmftbu   %0           \n"
21                "\tmftb    %1           \n"
22                "\tmftbu   %2           \n"
23                "\tcmpw    %2,%0        \n"
24                "\tbne     loop         \n"
25  : "=r"(upper),"=r"(lower),"=r"(tmp)
26  );
27  result = upper;
28  result = result<<32;
29  result = result|lower;
30
31  return(result);
32 }
33
34 #else
35
36 #error "No tick counter is available!"
37
38 #endif  // architecture
39
40 #endif
```

Em seguida veremos um exemplo de como usar as funções contidas no arquivo de cabeçalho anterior para capturar a informação contida no contador:

```
00 /* --------------------- file: getRdtsc.c */
01 #include <stdio.h>
02 #include "rdtsc.h"
```

```
03
04 int main(int argc, char* argv[])
05 {
07  unsigned long long a,b;
08
09  a = rdtsc();
10  b = rdtsc();
11
12  printf("%llu\n", b-a);
13  return 0;
14 }
```

No código acima percebemos que não existem instruções entre a primeira vez que foi chamada a função *rdtsc()* e a segunda vez, pois trata-se de um exemplo para ilustrar. Uma sugestão de uso é associar uma chamada a essa função sempre que um evento de tecla pressionada ou tecla solta ocorrer.

# Apêndice B

# Artigos Anexados a Dissertação

Este apêndice tem como objetivo reproduzir integralmente os artigos elaborados durante para que sirvam de referências em trabalhos futuros.

São eles:

1. C.R. do N. Costa, G.F.G. Yared, R.N. Rodrigues, J.B.T. Yabu-uti, F. Violaro, L.L. Ling. **Autenticação Biométrica via Dinâmica da Digitação em Teclados Numéricos**. *Anais do XXII Simpósio Brasileiro de Telecomunicações* - SBrT'05, Campinas, São Paulo, Brasil, 04-08 de Setembro de 2005.

2. R.N. Rodrigues, G.F.G. Yared, C.R. do N. Costa, J.B.T. Yabu-uti, F. Violaro, L.L. Ling. **Biometric Access Control through Numerical Keyboard based on Keystroke Dynamics**. *Proceedings of the Second International Conference on Biometric*, ICB'2006, Hong Kong, China, 5-7 January 2006.

O primeiro artigo foi aceito apresentado no XXII Simpósio Brasileiro de Telecomunicações, publicado nos Anais do evento, enquanto que a segunda publicação foi apresentada e publicada no Second International Conference on Biometric.

XXII SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES - SBrT'05, 04-08 DE SETEMBRO DE 2005, CAMPINAS, SP

# Autenticação Biométrica via Dinâmica da Digitação em Teclados Numéricos

Carlos Roberto do N. Costa, Glauco F. G. Yared, Ricardo Nagel Rodrigues, João B. T. Yabu-Uti, Fábio Violaro, Lee Luan Ling

***Resumo*— Este artigo apresenta nova abordagem para autenticação biométrica via dinâmica da digitação em teclados numéricos. O sinal de entrada é obtido em tempo real durante a digitação pelo usuário da *String* alvo. Cinco características são extraidas do sinal (código ASCII da tecla e quatro durações associadas) e quatro experimentos usando amostras de usuários e impostores foram analisados comparando-se dois classificadores de padrões. Obteve-se melhores resultados com *HMM* (EER=4,5%). Esta nova abordagem traz melhorias ao processo de autenticação pois permite que a senha não seja mais segredo, assim como permite incluir autenticação biométrica em dispositivos móveis, como celulares.**

***Palavras-Chave*— Processamento digital de sinais, reconhecimento de padrões, segurança, biometria, dinâmica da digitação.**

***Abstract*— This paper presents a new approach for biometric authentication using keystroke dynamics in numerical keyboards. The input signal is generated in real time when the user enters the target string. Five features are extracted from this input (key ASCII code and four associated durations) and four experiments using samples for genuines and impostors users were performed using two pattern classification techniques. The best results were achieved by the *HMM* (EER=4.5%). This new approach brings improvements to the process of user authentication since it allows the password not to be a secret anymore, as well as it allows to include biometric authentication in mobile devices, such as cell phones.**

***Keywords*— Digital signal processing, pattern recognition, security, biometrics, keystroke dynamics.**

## I. Introdução

Controlar o acesso a sistemas computacionais torna-se cada vez mais importante nos dias de hoje, e o mecanismo mais conhecido e usual para garantir segurança em sistemas de informação é através da autenticação do usuário por uma senha. Porém este tipo de mecanismo é frágil pois existem usuários descuidados que comprometem a segurança quando utilizam-se de senhas frágeis e de contexto normalmente familiar, como por exemplo uma data de nascimento. Por outro lado, o custo e a simplicidade deste tipo de mecanismo clássico de segurança justifica sua adoção, e em várias situações permanece como mecanismo principal ao lado de outras políticas de segurança. O propósito deste trabalho é melhorar o processo de autenticação por senha usando características biométricas. Características biométricas são padrões observados no ser humano que permitem criar algoritmos capazes de distinguir uma pessoa da outra, e podem ser classificadas como sendo características fisiológicas ou características de natureza comportamental [1]. A inclusão de características biométricas em sistemas de autenticação pessoal aumenta o grau de confiança dos usuários na segurança do sistema, pois características biométricas são únicas para cada pessoa e não podem ser roubadas, perdidas ou esquecidas.

A tecnologia biométrica tratada neste trabalho é conhecida como biometria da digitação ou dinâmica da digitação. A biometria da digitação é o processo de analisar o ritmo com que o usuário digita em um terminal através do monitoramento das entradas do teclado durante as tentativas de identificação. Autenticação por dinâmica da digitação pode ser classificada em estática ou contínua. A abordagem estática analisa as entradas em um momento particular, enquanto que na abordagem contínua são analisadas todas as entradas no teclado durante a sessão do usuário [2].

Este trabalho inova na forma como usa a dinâmica da digitação, pois utiliza somente um teclado numérico para observar a dinâmica com que o usuário digita uma senha exclusivamente numérica. O uso do teclado numérico diminui o intervalo de tempo entre teclas e aumenta ainda mais a exigência de precisão, além de praticamente forçar o usuário a usar somente uma das mãos. Além disso, permite que sejam propostos soluções para funcionar em telefones celulares, em sistemas de caixa automático para bancos ou mesmo no acesso a áreas restritas. A metodologia adotada neste trabalho tem um baixo custo de processamento, é não-intrusiva e verifica o usuário de maneira estática, ou seja, apenas considera a entrada digitada em um dado momento.

O resto do trabalho está organizado da seguinte maneira. Na seção 2 são apresentados resumidamente os trabalhos publicados e relacionados com a área. Na seção 3 é discutida a metodologia proposta na extração das características e são apresentados os classificadores utilizados. Na seção 4 são apresentados os experimentos conduzidos e os resultados obtidos; e finalmente na seção 5 são apresentadas as conclusões e as propostas de trabalhos futuros.

## II. Trabalhos Relacionados

Autenticação biométrica por dinâmica da digitação é uma área de pesquisa ativa desde 1990 [2]-[14]. Alguns aspectos sobre sistemas desta natureza são discutidos neste trabalho e resumidamente apresentados a seguir.

- ***String* Alvo**: É a *string* que será digitada pelo usuário e monitorada pelo sistema. Em [3] são usadas quatro

Os autores pertencem ao Departamento de Comunicações, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, São Paulo, Brasil, E-mails: {ccosta, glauco, ricardonagel, yabuuti, fabio, lee}@decom.fee.unicamp.br.

*strings* como alvo durante a autenticação (usuário, senha, primeiro nome e último nome). Porém, em alguns trabalhos, somente a senha é suficiente. Outro aspecto importante diz respeito ao tamanho da *string*. Em [4] os autores perceberam que o sistema fica sujeito a mais erros durante o processo de classificação quando *strings* de entrada menores que dez são adotadas pelos usuários.

- **Número de amostras**: Amostras são coletadas durante o processo de cadastramento dos usuários, e irão compor o conjunto de treinamento do classificador. O número de amostras varia muito, sendo que a menor quantidade foi relatada em [5], onde somente três amostras foram usadas, e o maior número de amostras foi relatado em [6], onde pediu-se ao usuário que digitasse trinta amostras. Em [2] os autores observaram que o número mínimo de amostras para não comprometer o desempenho do sistema é de seis amostras por usuário.
- **Extração das características**: Duas das características mais observadas durante a digitação são o tempo em que a tecla permanece pressionada e o intervalo de tempo entre teclas sucessivas [5]. Em [7]-[9] os autores combinam estas características, obtendo melhores resultados do que usando isoladamente cada uma delas. De Ru *et al.* [10] analiza uma característica diferente baseada na distância das teclas no teclado alfa-numérico e a combinação de teclas consideradas difíceis, porém obriga o usuário a decorar uma *string* à qual não está habituado. Como a maioria das características observadas são temporais, a precisão com que se deve observar o tempo onde uma determinada tecla é pressionada ou solta torna-se importante. Os intervalos entre teclas podem variar entre 0.1ms [7] e 1000ms (1s) [11].
- **Tentativas de autenticação**: Em [12] os autores observaram que usuários legítimos normalmente falham na primeira tentativa de autenticação, sendo normalmente autenticado na segunda tentativa. Em [6], cada usuário deve digitar sua *string* de entrada duas vezes, usando uma técnica de embaralhamento.
- **Mecanismo de adaptação**: Características biométricas podem sofrer pequenas mudanças com o passar do tempo. Portanto, faz-se necessário em sistemas biométricos a presença de mecanismos de adaptação, ou seja, um recadastramento pode ser feito para manter o modelo do usuário sempre atualizado. A maioria dos pesquisadores não menciona este aspecto importante em seus trabalhos, porém em [13] os autores citam um mecanismo de adaptação onde sempre que um usuário é autenticado de maneira correta o sistema calcula outro modelo acrescentando a entrada atual como uma nova amostra e descartando a mais antiga.
- **Classificador**: Em [2]-[4], [6], [7] e [11], os autores adotaram classificadores estatísticos em seus experimentos, como por exemplo *k-means*, Bayes, etc. Em [9] e [10], lógica nebulosa foi aplicada usando como saída um categorizador de usuários. Finalmente, em [5], [8] e [14], redes neurais artificiais foram usadas para identificar o usuário.

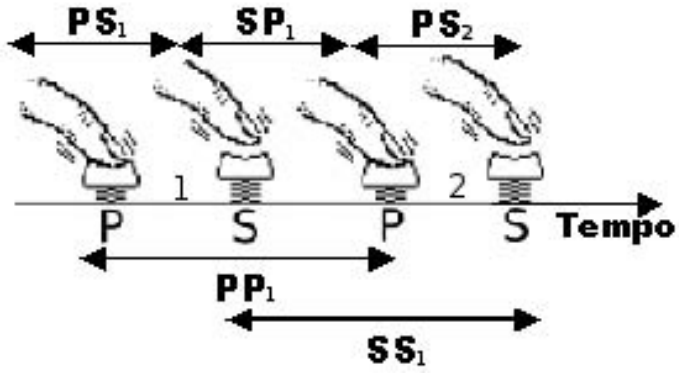


Fig. 1. Representação das características observadas durante a digitação dos caracteres 1 e 2. *PS* é o tempo em que a tecla permanece pressionada, *SP* é o intevalo até a próxima tecla ser pressionada, *PP* é o intervalo de tempo que o usuário leva para pressionar duas teclas consecutivas e *SS* é o intervalo de tempo que o usuário leva para soltar duas teclas consecutivas.

## III. METODOLOGIA

Este trabalho inova pois, ao invés de considerar o teclado alfa-númerico e uma *string* que pode conter tanto letras como números, restringe o usuário ao teclado númerico, com senhas numéricas limitadas em oito caracteres e escolhidas pelo usuário. Teclados desta natureza estão presentes em telefones celulares, caixas automáticos de banco ou em controle de acesso a áreas restritas. Além disso, também utiliza uma nova abordagem no processo de classificação que se baseia em Modelos Ocultos de Markov (*Hidden Markov Models*, *HMM*).

Fundamentalmente a dinâmica da digitação necessita adquirir os padrões do digitador de forma precisa no tempo [14]. Neste trabalho usamos a função *Time Stamp Counter* para adquirir o número preciso de ciclos do processador, tal qual descrita em [14]. A precisão adotada deve manter-se constante para toda a base de dados, onde 98% das amostras coletadas estão entre 10 e 900ms, portanto, 1ms de precisão foi adotado neste trabalho.

Nas próximas seções são discutidos os aspectos relacionados com a extração de características, construção do modelo do usuário e classificação do padrão.

### *A. Extração das características e base de dados*

Para a construção da base de teste usou-se um teclado numérico, onde dez amostras com oito caracteres cada são coletadas em cada sessão e para cada usuário, totalizando 40 amostras (4 sessões por usuário), sendo que vinte usuários foram convidados a participar do experimento. Em [15] observou-se que mais de dez amostras por sessão no cadastramento incomoda os usuários e em [2] os autores observaram que quanto menor o número de amostras, pior será o desempenho do classificador.

Cada amostra de $n$ caracteres pode conter várias características de digitação, onde uma determinada característica de digitação para a amostra $w$ da conta $a$ pode ser difinida por $K_{a,w} = (K_1(a,w), K_2(a,w),\ldots,K_n(a,w))$. Cada característica $K_i(a,w)$, onde $i \leq n$, representa uma das seguintes características observadas:

- Código ASCII: O caracter digitado possui uma entrada na tabela ASCII. $C_a$ = ($C_1(a)$, $C_2(a)$,...,$C_n(a)$) representa os códigos das respectivas teclas para o teclado numérico e que estão contidos no modelo do usuário $a$ e $C_{a,w}$ = ($C_1(a,w)$, $C_2(a,w)$,...,$C_n(a,w)$) representa os códigos das teclas da amostra $w$ para a conta do usuário $a$;
- O intervalo de tempo em que a tecla permanece pressionada (PS) [5]. Esta característica é representada pela expressão $PS_{a,w}$ = ($PS_1(a,w)$, $PS_2(a,w)$,...,$PS_n(a,w)$), onde $PS_i(a,w)$ = $T_{i.solta}(a,w)$ - $T_{i.pressiona}(a,w)$ está relacionado com $K_i$. $T_{i.solta}(a,w)$ é o instante onde a tecla $i$ é solta e $T_{i.pressiona}(a,w)$ é o instante onde a tecla $i$ é pressionada;
- O intervalo de tempo até a próxima tecla ser pressionada (SP). Esta característica é representada pela expressão $SP_{a,w}$ = ($SP_1(a,w)$, $SP_2(a,w)$,...,$SP_{n-1}(a,w)$), onde $SP_i(a,w)$ = $T_{i+1.pressiona}(a,w)$ - $T_{i.solta}(a,w)$ está relacionado com ($K_i$,$K_{i+1}$);
- O intervalo de tempo que o usuário leva para pressionar duas teclas consecutivas (PP). Esta característica é representada pela expressão $PP_{a,w}$ = ($PP_1(a,w)$, $PP_2(a,w)$,...,$PP_{n-1}(a,w)$), onde $PP_i(a,w)$ = $T_{i+1.pressiona}(a,w)$ - $T_{i.pressiona}(a,w)$ está relacionado com ($K_i$,$K_{i+1}$);
- O intervalo de tempo que o usuário leva para soltar duas teclas consecutivas (SS). Esta característica é representada pela expressão $SS_{a,w}$ = ($SS_1(a,w)$, $SS_2(a,w)$,...,$SS_{n-1}(a,w)$), onde $SS_i(a,w)$ = $T_{i+1.solta}(a,w)$ - $T_{i.solta}(a,w)$ está relacionado com ($K_i$,$K_{i+1}$);

A figura 1 mostra um exemplo de extração de durações associadas à digitação dos caracteres um e dois.

Além da base de usuários descrita anteriormente, três usuários foram convidados a participar como impostores de cada uma das vinte contas. Ao final de dez tentativas, 600 amostras de impostores foram coletadas. A figura 2 apresenta as distribuições entre autênticos e impostores usando o método descrito em [15]. Pode-se observar que a sobreposição entre as classes é visível, o que ocasiona altas taxas de erro. Esta sobreposição ocorre pois em teclados numéricos os usuários digitam de maneira similar, ou seja, usando apenas uma mão e com ritmos semelhantes.

### B. Classificador estatístico

Em [15] os autores sugerem que o modelo do usuário seja gerado a partir da média e do desvio padrão entre as amostras observadas no cadastramento. Sempre que o usuário tenta autenticar sua senha, o sistema calcula a distância da *string* alvo para o modelo e, se ela for maior que um limiar pré-definido, então o usuário é autenticado. O modelo é gerado a partir de $N$ amostras, podendo conter as características $K$={PP, SP, PS, SS} descritas anteriormente, e de acordo com as equações (1) e (2), re-escritas a seguir:

$$\mu_{K_i(a)} = \frac{1}{N}\sum_{j=1}^{N} K_i(a,j) \qquad (1)$$

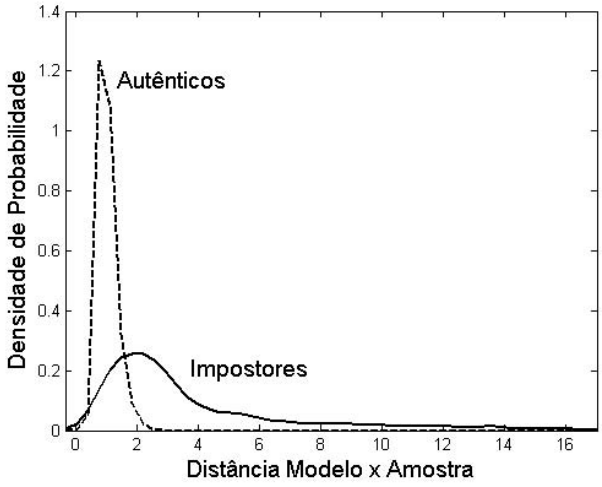


Fig. 2. Distribuição entre autênticos e impostores para a base de dados usada nos testes.

$$\sigma_{K_i(a)} = \frac{1}{N-1}\sum_{j=1}^{N} |K_i(a,j) - \mu_{K_i(a)}| \qquad (2)$$

Na autenticação, o código ASCII identifica inicialmente o usuário que deseja ser autenticado como dono da conta $a$. Conhecendo inicialmente quem clama por sua identidade, o sistema conduz a uma comparação um-para-um, onde a distância total de cada uma das características observadas para o modelo da conta $a$ é calculado, através da equação (3):

$$D_K(a,w) = \frac{1}{n}\sum_{i=1}^{n} \frac{K_i(a,w) - \mu_{K_i(a)}}{\sigma_{K_i(a)}} \qquad (3)$$

onde $n$ é o número de elementos da característica $K$, para $K$={PP, SP, PS, SS}. Se $D_K(a,w) \leq \tau_K(a)$, para todo $K$, o usuário é considerado autêntico para a conta $a$. $\tau_K(a)$ é o limiar de decisão da conta $a$ definido empiricamente.

Ainda em [15] foi proposto também um mecanismo de adaptação do modelo em que, se o usuário foi autenticado positivamente, então a *string* alvo é adicionada ao modelo e o sistema recalcula a média e o desvio padrão entre as amostras armazenadas e a nova amostra, usando uma metodologia semelhante a uma fila: descarta a amostra mais antiga e acrescenta a amostra recente. A figura 3 ilustra o processo de atualização do conjunto de treinamento ao longo do tempo.

Os autores reportaram taxas de falsa-aceitação e falsa-rejeição menores que 2% usando esta abordagem em teclados alfa-numéricos com *strings* contendo letras e/ou números.

### C. Classificador usando HMM

Os sistemas baseados em *HMM* têm sido amplamente utilizados em reconhecimento de padrões [17],[16]. A utilização de tais sistemas está, em grande parte, associada à necessidade de se modelar a variabilidade temporal dos padrões analisados. Além disso, a utilização de misturas Gaussianas permite a modelagem de distribuições complexas, cujas fronteiras são de grande importância do ponto de vista da classificação.

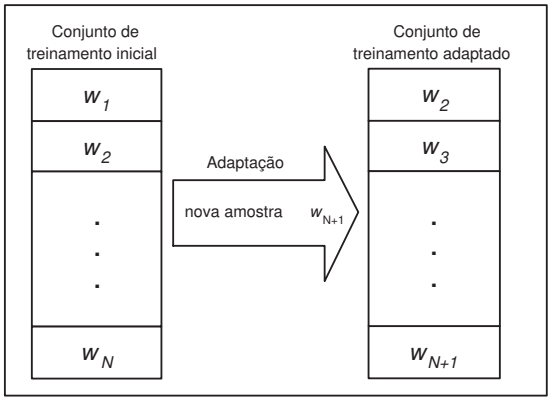


Fig. 3. Mecanismo de adaptação do modelo.

A sobreposição das distribuições de classes distintas dificulta a determinação das fronteiras e é um problema comum no processo de modelagem, podendo ser tratado pela escolha apropriada do número e localização das componentes de cada mistura. No entanto, se a quantidade de dados disponíveis para o processo de modelagem for insuficiente, torna-se difícil a determinação do número apropriado de Gaussianas por mistura para modelar cada classe.

O modelo probabilístico empregado para representar cada usuário com sua senha específica pode ser compreendido como um caso particular de um *HMM* contínuo com 15 estados e topologia *left-right*. Cada estado está associado ao tempo em que uma tecla fica pressionada ou ao intervalo para o digitador se deslocar entre uma tecla e outra. Assim:

- Estado 1, associado ao tempo em que a primeira tecla fica pressionada;
- Estado 2, associado ao tempo para o digitador se mover entre a primeira e a segunda tecla;
- Estado 3, associado ao tempo em que a segunda tecla fica pressionada;

.
.
.

- Estado 14, associado ao tempo para o digitador se mover entre a sétima e a oitava tecla;
- Estado 15, associado ao tempo em que a oitava tecla fica pressionada;

As probabilidades de transição são também dadas por $a_{i(i+1)} = 1$, ou seja, não ocorrem auto-transições ($a_{ii} = 0$). Cada novo parâmetro de entrada $K_i(a, w)$, *i* de 1 a 15, está associado ao estado *i* e é modelado através de uma mistura de 6 Gaussianas unidimensionais. A cada novo parâmetro $K_i(a, w)$, o sistema avança do estado *i* para o estado *i+1*.

A idéia consiste basicamente em modelar cada duração por um estado do modelo, uma vez que o número de durações é fixo para todos os digitadores (8 dígitos resultando em 15 durações do tipo tecla pressionada e intervalo entre teclas). A figura 4 ilustra um exemplo de digitação da senha constituída pela sequência de dígitos "2-5-0-5-7-8-9-4" e os estados do *HMM* utilizados na modelagem. Dessa forma, para uma dada senha digitada, utiliza-se o modelo *HMM* correspondente a fim

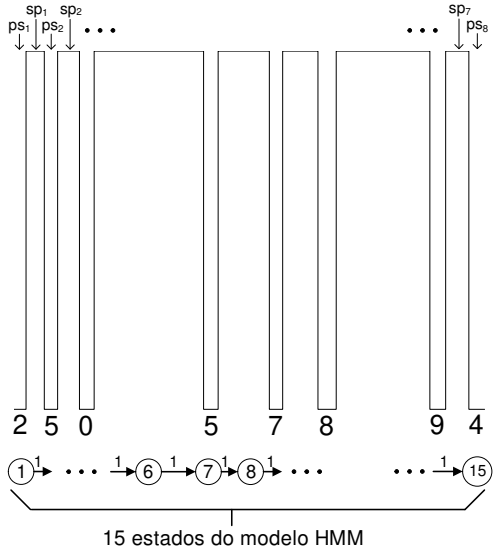


Fig. 4. Digitação da sequência "2-5-0-5-7-8-9-4". O HMM utilizado para a modelagem das durações correspondentes e as probabilidades de transição estão indicados na figura.

de se obter o valor de verossimilhança $P(O_{a,w}|\lambda_a)$ através do algoritmo de Viterbi [16], em que

$$O_{a,w} = \{PS_1(a,w), SP_1(a,w), PS_2(a,w), SP_2(a,w), \ldots, PS_{n-1}(a,w), SP_{n-1}(a,w), PS_n(a,w)\},$$

e $\lambda_a$ é o conjunto de parâmetros do modelo associado à conta *a*.

Se tal valor for superior a um limiar $\tau_K(a)$ pré-definido para a conta *a*, então o usuário é autenticado, caso contrário é considerado impostor e então rejeitado. O sistema foi treinado utilizando-se as ferramentas do HTK [18], através do algorítmo de Baum-Welch [17]. Os modelos são gerados inicialmente a partir de *N* amostras de cada digitador. Na sequência, para cada uma das amostras da base de teste, realiza-se a autenticação e, caso o usuário seja autenticado, tal amostra é adicionada à base de treinamento de maneira semelhante ao adotado para o classificador estatístico, e o sistema é treinado novamente.

Na próxima seção discute-se os experimentos conduzidos usando os classificadores propostos, bem como demonstra-se os resultados obtidos com a base de teste.

## IV. Experimentos e Resultados

Os experimentos foram conduzidos em um computador *Pentium IV* e utilizando a região do teclado composta de teclas numéricas. Vinte usuários, entre homens e mulheres de várias faixas etárias e com diferentes níveis de familiaridade com o teclado numérico participaram do experimento, usando como *string* alvo uma combinação de oito números a sua escolha. Duas situações foram observadas:

- *Usuários autênticos*: O usuário tenta autenticar-se em sua conta. Foram coletadas dez amostras do usuário em cada uma das 4 seções, totalizando 800 amostras de usuários autênticos.

- *Usuários impostores*: Impostores foram convidados a tentar autenticar-se nas contas dos usuários autênticos. Cada conta foi atacada 30 vezes, resultando em 600 amostras de impostores.

As amostras dos digitadores foram coletadas no Laboratório de Reconhecimento de Padrões e Redes de Computadores (LRPRC) da UNICAMP. As amostras dos usuários foram coletadas em diferentes períodos, e nunca ao mesmo tempo, intercalando um período médio de uma semana entre as seções.

Foram feitos três experimentos com o classificador estatístico:

1) utilizando dez amostras ($N$=10) para gerar o modelo com $K$={PP, SP, PS, SS};
2) utilizando vinte amostras ($N$=20) para gerar o modelo com $K$={PP, SP, PS, SS};
3) utilizando trinta amostras ($N$=30) para gerar o modelo com $K$={PP, SP, PS, SS}.

Além dos experimentos com o classificador estatístico, foi feito um experimento com *HMM* utilizando trinta amostras ($N$=30) para a obtenção do modelo.

### A. Resultados

O desempenho de sistemas biométricos é geralmente medido por três tipos de taxas de erros [1]:

- Taxa de falsa-aceitação (*FAR*): A probabilidade do sistema falhar na rejeição de usuários impostores.
- Taxa de falsa-rejeição (*FRR*): A probabilidade do sistema falhar quando verifica se um usuário legítimo é quem diz ser.
- Taxa de erro igual (*EER*): É o valor assumido quando *FAR* e *FRR* são iguais.

Para representar estas taxas, utiliza-se o gráfico *ROC* (*receive operating curve* [1]), que representa a *FAR* pela *FRR* para diferentes limiares, e apresenta graficamente o ponto onde os erros são iguais (*EER*).

As figuras de 5 a 7 representam as curvas *ROC* para o classificador estatístico nos experimentos (1), (2) e (3). Pode-se observar que o melhor resultado foi alcançado quando o valor de $N$ empregado na construção do modelo foi de 20 amostras, resultando uma taxa de *EER* igual a 6.2%. Vale ressaltar que em [15] os autores obtiveram taxa de *EER* de 1.6%, o que é bastante plausível pois seus experimentos foram conduzidos em um teclado alfa-numérico, os usuários utilizam senhas com *string* alvo contendo mais de 8 caracteres e com durações mais representativas do que as obtidas em teclados numéricos.

A figura 8 mostra o desempenho do sistema quando o classificador usado é o *HMM* com $N$=30. Podemos observar que os resultados obtidos com *HMM* superam os obtidos com o classificador estatístico. O método de atualização do modelo utilizado permitiu representar de uma forma mais eficiente a dinâmica do digitador, o que possívelmente é causado pela não-estacionariedade dos dados. Futuramente, deve-se utilizar técnicas de seleção de topologia [19] para o modelo *HMM* afim de se obter modelos mais consistentes. Vale ressaltar que a utilização de menos amostras para gerar o modelo *HMM* dificulta a estimação dos parâmetros dos usuários no teclado numérico, pois a quantidade de dados torna-se insuficiente, o que foi comprovado também no classificador estatístico.

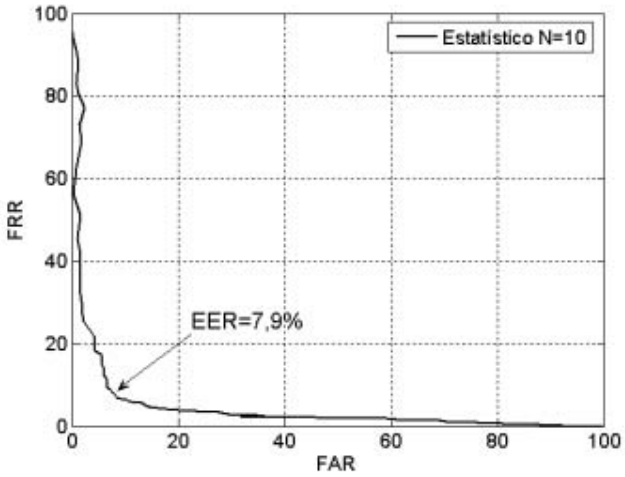


Fig. 5. *ROC* para o classificador estatístico quando $N$=10.

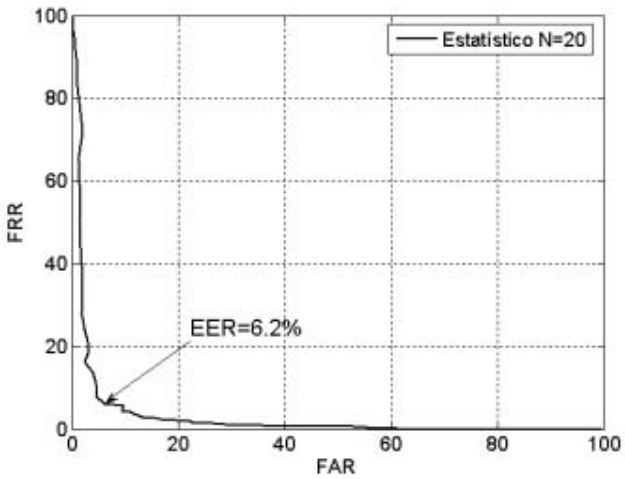


Fig. 6. *ROC* para o classificador estatístico quando $N$=20.

## V. Conclusões e trabalhos futuros

Este trabalho apresentou uma metodologia de autenticação biométrica através das características extraídas da dinâmica da digitação que permite melhorar o processo de controle de acesso a áreas restritas ou aumentar a segurança de transações bancárias. Alguns experimentos foram conduzidos e o melhor resultado foi alcançado usando um classificador baseado em modelos ocultos de Markov e combinando três características (código ASCII e as durações SP e PS), onde foi obtida *EER* de 4.5%. Esta taxa é competitiva, pois foi comparada com o classificador estatístico proposto em [15] e obteve melhores resultados para uma base de dados que restringe o usuário ao teclado numérico e usando apenas uma *string* alvo no cadastramento. Essa abordagem não havia sido feita até o momento na literatura mundial.

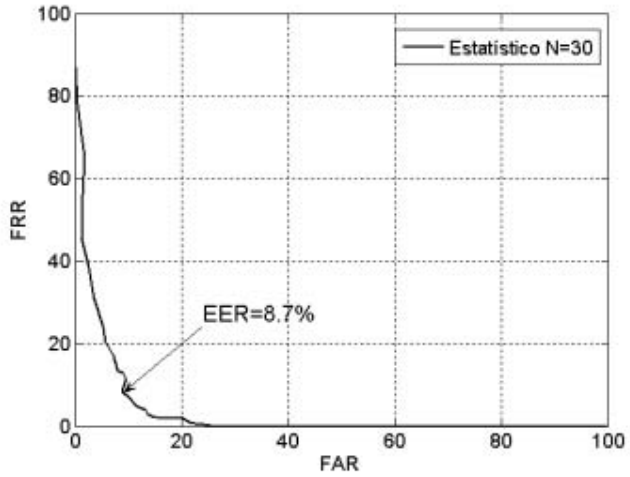


Fig. 7. *ROC* para o classificador estatístico quando *N*=30.

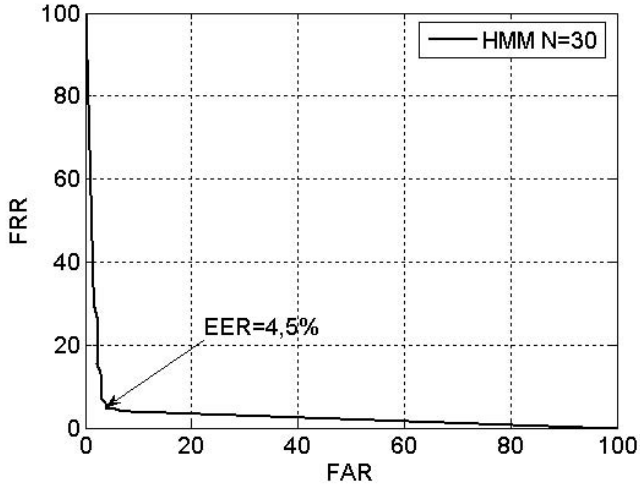


Fig. 8. *ROC* para o classificador *HMM* quando *N*=30.

Foi observado também a influência de alguns aspectos práticos, os quais foram testados e comprovados, mostrando que eles definitivamente devem ser considerados no desempenho de sistemas desse tipo. Estes aspectos são a familiaridade do usuário com a *string* alvo, o mecanismo de adaptação do modelo adotado no sistema, a precisão como os dados são adquiridos e, principalmente, o número de amostras usadas no cadastramento.

Para trabalhos futuros, pretende-se aumentar a população de usuários e capturar mais seções dos mesmos, pois observou-se a não-estacionariedade da dinâmica ao longo do tempo. Além disso, um mecanismo de adaptação do modelo *HMM* será proposto afim de baixar ainda mais a *EER* do sistema. Outro aspecto a ser melhorado é a forma de obtenção da topologia do *HMM*, na qual deve-se utilizar técnicas de seleção de estrutura de modelos [19].

## AGRADECIMENTOS

Os autores gostariam de expressar seus sinceros agradecimentos à CAPES e ao CNPQ pelo apoio financeiro e a todos os usuários que colaboraram pacientemente com a coleta de dados. Sem eles este trabalho não seria possível.

## REFERÊNCIAS

[1] A. K. Jain, A. Ross and S. Prabhakar, *An Introduction to Biometric Recognition*, IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Image and Video-Based Biometrics, Vol. 14, No. 1, pp. 4-20, January 2004.

[2] F. Monrose and A. D. Rubin, *Keystroke Dynamics as a Biometric for Authentication*, Future Generation Computer Systems, Vol. 16, no. 4, pp. 351-359, March 1999.

[3] R. Joyce and G. Gupta, *Identity authentication based on keystroke latencies*, commun. ACM, vol. 33, no. 2, pp. 168-176, 1990.

[4] d. Bleha and M. Obaidat, *Dimensionality reduction and feature extraction applications in identifying computer users*, IEEE Trans. Syst., Man, Cybern., Vol. 21, no. 2, pp. 452-456, Mar.-Apr. 1991.

[5] D. T Lin, *Computer-access authentication with neural network based keystroke identity verification*, in Proc. Int. Conf. Neural Networks, vol. 1, 1997, pp. 174-178.

[6] S. Bleha, C. Slivinsky, and B. Hussain, *Computer-access security systems using keystroke dynamics*, IEEE Trans. Pattern Anal. Machine Intell., vol. 12, no. 12, pp. 1217-1222, Dec. 1990.

[7] J. A. Robison, V. M. Liang, J. A. Michael, and C. L. MacKenzie, *Computer user verification login string keystroke dynamics*, IEEE Trans. Syst., Man, Cybern., vol. 28, no. 2, pp. 236-241, Mar.-Apr. 1998.

[8] M. S. Obaidat and B. Sadoun, *Verification of computer user using keystroke dynamics*, IEEE Trans. Syst., Man, Cybern., vol. 27, no. 2, pp. 261-269, Mar.-Apr. 1997.

[9] L. C. F. Araújo, L. H. R. Sucupira Jr., M. G. Lizárraga, L. L. Ling, and J. B. T. Yabu-uti, *A fuzzy logic approach in typing biometrics user authentication*, in Proc. 1st Indian Int. Conf. Artificial Intelligence, 2003, pp. 1038-1051.

[10] W. G. de Ru and J. H. P. eloff *Enhaced password authentication through fuzzy logic*, IEEE Expert, vol. 17, no. 6, pp. 38-45, Nov.-Dec. 1997.

[11] O. Coltell, J. M. badfa, and G. Torres, *Biometric identification system based in keyboard filtering*, in Proc. IEE 33rd Annu. Int. Carnahan Conf. Security Technology, 1999, pp. 203-209.

[12] S. Haidar, A. Abbas, and A. K. Zaidi, *A multi-technique approach for user identification trhough keystroke dynamics*, in Proc. IEEE Int. Conf. Systems, Man and Cybernetics, vol. 2, 2000, pp. 1336-1341.

[13] F. Monrose, M. K. Reiter, and S. Wetzel, *Password hardening based on keystroke dynamics*, in Proc. 6th ACM Conf. Computer Security, Singapore, Nov. 1999.

[14] F. W. M. H. Wong, A. S. M. Supian, A. F. Ismail, L. W. Kin, and O. C. Soon, *Enhanced user authentication through typing biometrics with artificial neural network and k-nearest neighbor algorithm* in Conf. Rec. 35th Asilomar Conf. Signals, Syst., comput., Vol. 2, 2001, pp. 911-915.

[15] L. C. F. Araújo, L. H. R. Sucupira Jr., M. G. Lizárraga, L. L. Ling, and J. B. T. Yabu-uti,*User authentication through typing biometrics features*, IEEE Trans. on Signal Processing, vol. 53, No. 2, Feb. 2005.

[16] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification* Wiley-Interscience Publication, 2nd Edition, Oct. 2000.

[17] L. R. Rabiner, *A tutorial on hidden Markov models and selected applications in speech recognition*. Proc. of IEEE, 77, pp. 257-286, 1989.

[18] Cambridge University Engineering Departament, *The HTK Book*, Cambridge University, 2002.

[19] M. A. T Figueiredo and A. K. Jain, *Unsupervised learning of finite mixture models*, IEEE Trans. Pattern Anal. Machine Intell., vol. 24, no. 3, pp. 381-396, Mar. 2002.

# Biometric Access Control through Numerical Keyboards based on Keystroke Dynamics

Ricardo N. Rodrigues, Glauco F. G. Yared, Carlos R. do N. Costa, João B. T. Yabu-Uti, Fábio Violaro, Lee Luan Ling

Department of Communications
School of Electrical and Computer Engineering
State University of Campinas
Albert Einstein Av., 400, PO Box 6101, Postal Code 13083-852, Campinas, SP, Brazil
ricardonagel@gmail.com, {glauco, ccosta, yabuuti, fabio, lee}@decom.fee.unicamp.br

**Abstract.** This paper presents a new approach for biometric authentication based on keystroke dynamics through numerical keyboards. The input signal is generated in real time when the user enters with target string. Five features were extracted from this input signal (ASCII key code and four keystroke latencies) and four experiments using samples for genuine and impostor users were performed using two pattern classification technics. The best results were achieved by the *HMM* (EER=3.6%). This new approach brings security improvements to the process of user authentication, as well as it allows to include biometric authentication in mobile devices, such as cell phones.

## 1 Introduction

Access control to computational systems has been becoming more important nowadays, and the most well known and usual mechanism to guarantee the security of the information systems is through user authentication by a password. However, this type of security mechanism is fragile. A negligent user compromise the security mechanism when one uses fragile passwords like the birth date, phone numbers, etc. On the other hand, the cost and the simplicity of this classic security mechanism justify its use, and in some situations it remains as the principal mechanism, supplemented with other security strategies. The intention of this work is to improve the process of password authentication using biometric features. Biometric features are patterns observed in human being that allow the design of algorithms capable of distinguishing a person from another, based on physiological and/or behavioral characteristics [1]. The use of biometric features can increases the degree reliability in user authentication since individual biometric feature, has intrinsic property of that cannot be stolen, lost or forgotten.

The biometric technology employed in this work is typing biometrics, also known as keystroke dynamics. The typing biometrics is an authentication process of analyzing the user's typing rhythm in a terminal at a keyboard during the identification. Authentication based on keystroke dynamics is a continuous or static process. The static manner analyzes the input keyboard at a particular

moment, for example when the user types his password, while the continuous approach analyzes all inputs in the keyboard during a user session [2].

The personal authentication system proposed in this paper only captures the keystroke dynamics from numerical keyboards (numerical passwords). This numerical password based approach was firstly introduced in [20]. The use of only a numerical keyboard causes more complex problems than using a full computer keyboard since only one hand is used to enter the password is the former case; as consequences, less information is available for authentication. In the latter case use of both hands for typing generates more keystroke dynamics information due to the intrinsic interaction between two hands (two strokes from distinct hands). As a result, more information available makes the authentication task easier [20].

The Numerical Typing Dynamic biometrics can be incorporated into mobile phones, Automated Teller Machine ($ATM$) systems and to control the access to restricted areas. The methodology adopted in this work has low processing cost, is non-intrusive [1] and statically authenticate users (only consider the input when the user types his password).

This paper is organized as follows. In section 2, related works in the area are presented. In section 3, the methodology of feature extraction and the pattern classification techniques are presented. In section 4, the experiments are described and discussed; and finally in section 5, the conclusions and future work are described.

## 2 Related Works

Biometric authentication by keystroke dynamics has been an active research area since 1990 [2]-[14]. In this section, we briefly provide an review of the previous.

- **Target string**: It is the string that will be provided by the user and monitored by the system. In [3] four target strings were used during the authentication (username, password, first name and last name). However, in some works the password is the only target. Another important aspect about the target string is the string size. In [4] the authors concluded that the numbers of errors in classification process increases as the length of target string decreases.
- **Amount of samples for obtaining the template**: Samples are collected during the user enrollment phase. Some or total of these form the system training set classifier. The number of samples varies largely in literature, varying from three to thirty samples per user [5], [6]. In [2] the authors have observed that the minimum number of samples that does not compromise the system performance is around six samples per user.
- **Feature extraction**: Two of more often observed features during the typing are the period of time in which the key remains pressed and the keystroke latency that corresponds to the time interval to move between successive keys [5]. In [7]-[9] the authors combine these features in order to get better

results than using each of them separately. De Ru *et al.* [10] have analyzed a different feature based on the physical distance among the keys in the alphanumeric keyboard and the combination of keys considered difficult. However, compels the user to memorize a string that he is not familiarized. As most of the features observed are time-based, the precision of key-up and key-down times when the observed key is pressed is important. The timing accuracy in previous works in this area can vary between 0.1ms [7] to 1000ms (1s) [11].

- **Trials of authentication**: In [12] the authors observed that most of the legitimate users failed in their first attempt of authentication, but normally succeeded in the following second attempt. In [6], each user had to type the target string twice, using a shuffling technique.
- **Adaptation mechanism**: Biometric features are subject to small changes over the time. Most of previous works in the literature seldomly mention this important aspect. To solve this problem, a suitable adaptation mechanism or a re-enrollment procedure can be implemented to keep the users templates updated. In [13] whenever a new positive authentication occurs, the users template are updated. The database updating consist of including the new sample, discarding the oldest one and re-training the user's reference feature model.
- **Classification**: In [2]-[4], [6], [7] and [11], the authors used statistical Classifiers in their experiments, such as k-means classifiers, Bayes decision rules, etc. In [9] and [10], a fuzzy logic classifier was implemented using a user's categorization as output. Finally, in [5], [8] and [14], artificial neural networks have been used to identify the user.

## 3 Methodology

In this work, we limit our investigation over numerical passwords collected from numerical keyboards. This keyboard type can be found in cell phones, ATM machines and most other access control systems. Each password is composed of a sequence of eight numerical characters, which is robust and easily memorized. For classifier a Hidden Markov Models ($HMM$) is implemented and tested.

A good biometric typing system acquires keystroke patterns with high time reliability [14]. In this work, the Time Stamp Counter function was summoned to acquire the processor cycle counts, as described in [14]. The clock precision is set to 1ms which is fixed to all collected data. The 98% of all latency measurements are found between 10 and 900ms.

### 3.1 Feature extraction and test database

Ten samples, each with eight numerical characters were collected from each user in each session of 4 sessions, totalizing in 40 samples per user. Twenty people have been invited to contribute with their password samples for the experiment.

The 10 sample session adopted in our experiments is due to the following observations. According to [15], collecting more than ten samples per session in the enrollment annoys the users. On the other hand, according to [2] a low number of samples degrades considerable the classifier performance.

Let $n$ denote the password sample size ($n$ characters) can contain some keystroke features. The keystroke feature vector of sample $w$ of user account $a$ can be expressed as $k_{a,w} = (k_1(a,w),\ k_2(a,w),\ \ldots,\ k_n(a,w))$. Each element $k_i(a,w)$, where $i \leq n$, represents one of the following features [5]:

- ASCII code. Each typed character generates an ASCII code sequences. $C_{a,w} = \{C_1(a,w),\ C_2(a,w),\ \ldots, C_n(a,w)\}$ represents the code sequence sample $w$ belong to user account $a$;
- The time interval when a key remains pressed (Down-Up or $DU$). This is represented by the expression $DU_{a,w} = \{DU_1(a,w),\ldots,DU_n(a,w)\}$, where $DU_i(a,w) = T_{i.up}(a,w)$ - $T_{i.down}(a,w)$. $T_{i.up}(a,w)$ is the instant where the key $i$ is released and $T_{i.down}(a,w)$ is the time instant when key $i$ is pressed;
- The time interval until the next key is pressed (Up-Down or $UD$). This is represented by the expression $UD_{a,w} = \{UD_1(a,w),\ldots,UD_{n-1}(a,w)\}$, where $UD_i(a,w) = T_{i+1.down}(a,w)$ - $T_{i.up}(a,w)$;
- The time interval between two consecutive pressed keys (Down-Down or $DD$). This is represented by $DD_{a,w} = \{DD_1(a,w),\ldots,DD_{n-1}(a,w)\}$, where $DD_i(a,w) = T_{i+1.down}(a,w)$ - $T_{i.down}(a,w)$;
- The interval between two consecutive released keys (Up-Up or $UU$). This is represented by $UU_{a,w} = \{UU_1(a,w),\ldots,UU_{n-1}(a,w)\}$, where $UU_i(a,w) = T_{i+1.up}(a,w)$ - $T_{i.up}(a,w)$.

For illustration purpose, figure 1 shows the feature extraction for a user typing a given target string.

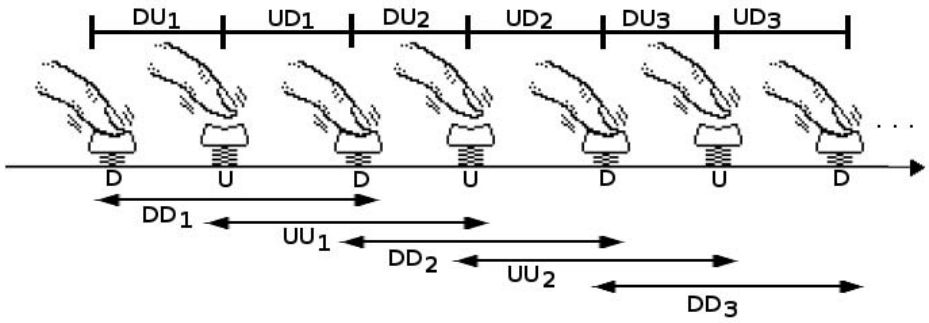


**Fig. 1.** Representation of the features observed during the typing of a given target string. DU is the time when a key remains pressed, UD is the time interval until the next key is be pressed, DD is the time interval between two consecutive pressed keys and UU is the time interval between two consecutive released keys.

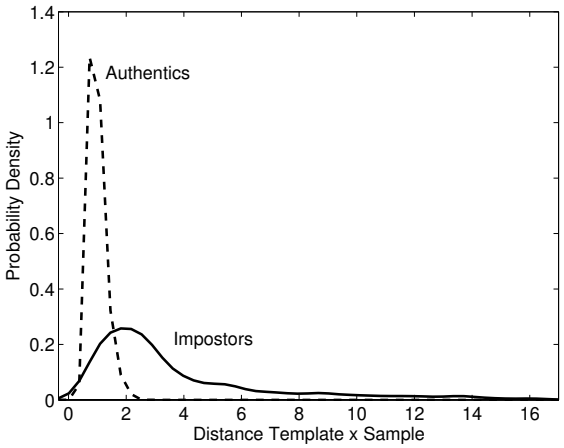


**Fig. 2.** The distribution of the genuine and impostor classes without the adaptation mechanism.

Figure 2 shows the distribution density functions of genuine and impostor classes using the method describe in [15] without the data updating mechanism. Notice hall the overlap of two functions is considerable, what contributes to high classification error rates.

### 3.2 Statistical classifier

In [15] the authors suggest that the user template be composed of the mean and standard deviation of sample feature vector acquired during the enrollment. Every time a user needs to authenticate his password, the system calculates the distance of the target string to the template. If the distance is larger than the threshold, the user is authenticated. The template implemented is derived from sample feature vector $K$={DD, UD, DU, UU} according to equations (1) and (2)

$$\mu_{K_i(a)} = \frac{1}{N}\sum_{j=1}^{N} K_i(a,j), \tag{1}$$

$$\sigma_{K_i(a)} = \frac{1}{N-1}\sum_{j=1}^{N} |K_i(a,j) - \mu_{K_i(a)}|. \tag{2}$$

In authentication, through the ASCII code provided by the system determines the intending user and the retrieves the template from the corresponding account for authentication. The 1 to 1 comparison consists of computing the

distance between the template and the input sample feature vector through the equation (3)

$$D_K(a,w) = \frac{1}{n}\sum_{i=1}^{n}\frac{K_i(a,w) - \mu_{K_i(a)}}{\sigma_{K_i(a)}}, \tag{3}$$

where $n$ is the number of latency feature in $K$, $K$={DD, UD, DU, UU}. If $D_K(a,w) \leq \tau_k(a)$, for all $K$, the user is considered authentic. $\tau_k(a)$ is a empirically defined threshold for user account $a$.

In [15] an data updating mechanism for the model is considered that is, if the user was successful authenticated, the input sample is included and the oldest one is discarded from the user's database and then the system generate a new template based on the new database.

The authors in [15] reported that both false-acceptance and false-rejection rates are below 2% using this approach in alphanumeric keyboards with target strings containing both letters and numerals.

### 3.3 Classifier using *HMM*

*HMM* based systems have been widely used in pattern recognition [17], [16]. This is due to the necessity of constructing models that analyze pattern's temporal variability. Moreover, the use of Gaussian mixtures allows ones to model complex distributions, and consequently build complex decision boundaries for classification processes.

The problem consist of determining highly overlapped class distributions. Determining the decision boundaries is not a trivial task, however a common problem in process modeling. The problem consist of determining appropriate number and positions of components for each Mixture, as soon as sufficient data information is available for distribution parameter estimation.

The probabilistic model used to represent each user's password is modeled by a continuous *HMM* with 15 states and *left-to-right* topology. Each state is associated with the time when the user presses the key or with the time interval between two consecutive pressed keys. Thus:

- State 1, associated with the time when the first key is pressed ($DU_1(a,w)$);
- State 2, associated with the time for the user to move from the first to the second key ($UD_1(a,w)$);
- State 3, associated with the time when the second key remains pressed ($DU_2(a,w)$);

.
.
.

- State 14, associated with the time for the user to move from the seventh to the eighth key ($UD_7(a,w)$);
- State 15, associated with the time when the eighth key remains pressed ($DU_8(a,w)$);

Let $A_{i(i+1)}$ denote the transition probability from state $i$ to state $i+1$. Each new input feature value $K_i(a,w)$, $i$ varying from 1 to 15 state is modeled by 6 Gaussian distribution. The input feature value $K_i(a,w)$ makes the *HMM* system advance from state $i$ to state $i+1$.

The basic idea of *HMM* model as to associate each latency measure (observed feature) to a state of the model. Therefore, a 8 digits password will resulting in 15 latencies. For each typed password, we estimate the likelihood probability $P(O_{a,w}|\lambda_a)$ of the corresponding *HMM* model through the Viterbi algorithm [16], where

$$O_{a,w} = \{DU_1(a,w), UD_1(a,w), DU_2(a,w), UD_2(a,w), \ldots,$$

$$DU_{n-1}(a,w), UD_{n-1}(a,w), DU_n(a,w)\},$$

and $\lambda_a$ is the set of feature value associated with user $a$, used for the model estimation.

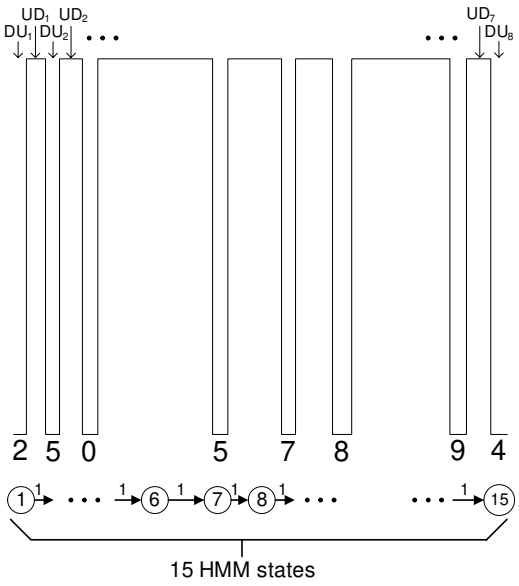


**Fig. 3.** Typing sequence "2-5-0-5-7-8-9-4". The *HMM* state used for modeling the corresponding latencies and the state transition probabilities are indicated.

If the likelihood probability estimate is superior to a threshold value, say a $\tau(a)$, the user is declared authentic; otherwise, he is considered an impostor. In this work, the threshold is obtained from the training data as follows:

$$\tau(a) = \mu_{P_a} - 3\sigma_{P_a} \tag{4}$$

where

$$\mu_{P_{(a)}} = \sum_{w=1}^{N} \frac{P(O_{a,w}|\lambda_a)}{N} \tag{5}$$

and

$$\sigma_{P_{(a)}} = \frac{1}{N-1} \sum_{w=1}^{N} |\mu_{P_{(a)}} - P(O_{a,w}|\lambda_a)| \tag{6}$$

For the system training the Baum-Welch algorithm [17] implemented in Hidden Markov Toolkit (*HTK*[18]) is used. Although the system model was initially trained by a set of fixed $N$ samples for each user $a$, the system has been trained again by most recent $N$ true samples every time a new sample is positive authenticated.

## 4 Experiments and Results

The experiments were performed on a *Pentium IV* microcomputer platform, using only numeric keyboard part. Twenty users of both sex aging from 20 to 60 years old with different levels of familiarity with the numerical keyboard have participated in the experiment. The target strings are composed of eight numbers freely chosen by the users. Two kinds of data were collected:

- *Authentic database*: Each user was undergone 4 sessions. In each session 10 sample was collected. This results in 800 samples in total.
- *Faked database*: For each true password, 30 samples are collected from ones other than the true user. This results in 600 faked samples in total.

All these samples were collected in Laboratory of Pattern Recognition and Computer Networks (LRPRC) at UNICAMP/Brazil, in different periods, were only one session was performed in each week.

For the statistical classifier design and system evaluations, three sets of training samples were used for model dimensioning:

I. Ten samples ($N$=10) to construct the model with $K$={DD,UD,DU,UU};
II. Twenty samples ($N$=20) to construct the model with $K$={DD,UD,DU,UU};
III. Thirty samples ($N$=30) to construct the model with $K$={DD,UD,DU,UU}.

In addition to the statistical classifiers, an *HMM* classifier was implemented, based on the using $K$={UD, DU} feature pattern. For model training, thirty samples ($N$=30) were used.

### 4.1 Experimental results

The performance of biometric systems is typically measured by three error rates types[1]:

- False acceptance rate (*FAR*): The overall rate at wich the system fail to reject impostor.
- False rejection rate (*FRR*): The overall rate at wich the system fail to verify legitimate users.
- Equal error rate (*EER*): It is the value when *FAR* and *FRR* are equally likely.

To represent these rates, we used an *ROC* (*Receiver Operating Characteristics* [1]) curve. Each point of the *ROC* curve represent a specific operating condition of the biometric system, wich is a function of decision thresholds.

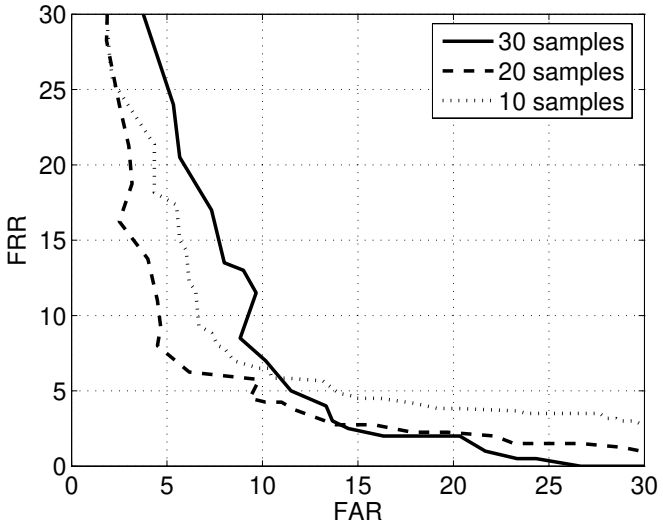


**Fig. 4.** The *ROC* performance of the statistical classifiers (trained by $N$=10 samples, $N$=20 samples and $N$=30 samples)

Figure 4 shows the *ROC* performance of the statistical classifiers design with different amounts of training samples. The lowest *EER* (6.2%) was obtained for the case of $N$=20 samples. Notice that this *EER* rate is considerably higher that supported in [15] (*EER* of 1.6%). However, this result is not surprising due to the fact that the experiments were made in an alphanumerical keyboard, in each user provided with his passwords with target string much bigger than 8 characters, what resulted in more representative latencies than in numerical keyboards.

Table 1 shows the performance of the *HMM* classifier (with $N$=30) wich outperforms all the 3 statistical classifiers in experiments. Figure 5 shows the *ROC* curve of the HMM classifier. The method of updating the model allowed

**Table 1.** Results obtained with the *HMM* classifier with $N$=30 samples and for all tests with the statistical classifier.

| EER Rate | Percentage(%) |
|---|---|
| **Experiment with *HMM*** | **3.6** |
| Experiment (I) | 7.9 |
| Experiment (II) | 6.2 |
| Experiment (III) | 8.7 |

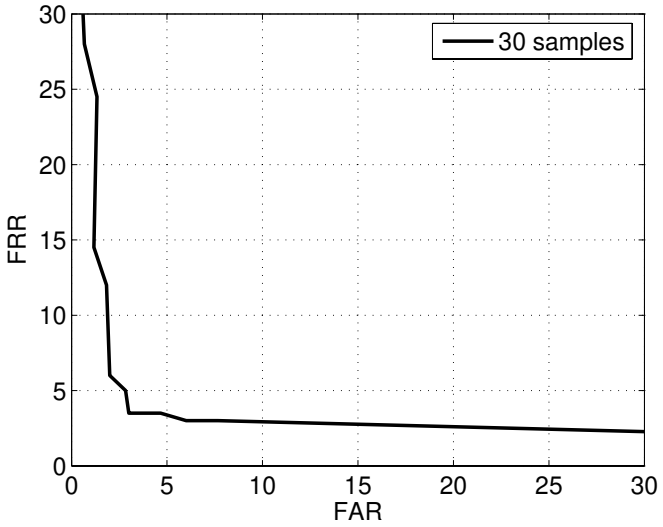


**Fig. 5.** Representation of *ROC* obtained in the experiments with the HMM classifier trained with $N$=30 samples

a more efficient form for modeling the typing dynamics of users. In the future, techniques of topology selection [19] may be used for the same *HMM* for getting more consistent models.

It is important to note that the number of parameter to train the *HMM* classifier is higher than in the statistical classifier, however his power of modeling the user data is superior.

## 5 Conclusions and future works

This work presented a novel methodology for biometric authentication based on the latency features extracted from the keystroke dynamics and *HMM* modelling approach. The biometric systems has the goal of improving the process of access control to restricted areas or to increase the security of banking transactions. The

experimental results reveal the potentially of hidden Markov models, resulting in an *EER* of 3.6%. This rate is competitive when is compared to that obtained by the statistical classifier considered in [15].

We also observed that they include the influence of some practical aspects need to be considered in the analysis of the system performance. They include the familiarity of users with target strings, the updating mechanism, the precision of data acquisition and, mainly, the number of training samples.

For the future work, we intend to make our database even more robust in forms of population size and number of data acquisition sessions. Also other topologies of the *HMM* with different model structures should be investigated [19].

## 6 Acknowledgement

The authors would like to express their sincere gratefulness to CAPES and CNPQ for the financial support and to all people that have patiently collaborated with the data collection process.

## References

1. A. K. Jain, A. Ross and S. Prabhakar, *An Introduction to Biometric Recognition*, IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Image and Video-Based Biometrics, Vol. 14, No. 1, pp. 4-20, January 2004.
2. F. Monrose and A. D. Rubin, *Keystroke Dynamics as a Biometric for Authentication*, Future Generation Computer Systems, Vol. 16, no. 4, pp. 351-359, March 1999.
3. R. Joyce and G. Gupta, *Identity authentication based on keystroke latencies*, commun. ACM, vol. 33, no. 2, pp. 168-176, 1990.
4. d. Bleha and M. Obaidat, *Dimensionality reduction and feature extraction applications in identifying computer users*, IEEE Trans. Syst., Man, Cybern., Vol. 21, no. 2, pp. 452-456, Mar.-Apr. 1991.
5. D. T Lin, *Computer-access authentication with neural network based keystroke identity verification*, in Proc. Int. Conf. Neural Networks, vol. 1, 1997, pp. 174-178.
6. S. Bleha, C. Slivinsky, and B. Hussain, *Computer-access security systems using keystroke dynamics*, IEEE Trans. Pattern Anal. Machine Intell., vol. 12, no. 12, pp. 1217-1222, Dec. 1990.
7. J. A. Robison, V. M. Liang, J. A. Michael, and C. L. MacKenzie, *Computer user verification login string keystroke dynamics*, IEEE Trans. Syst., Man, Cybern., vol. 28, no. 2, pp. 236-241, Mar.-Apr. 1998.
8. M. S. Obaidat and B. Sadoun, *Verification of computer user using keystroke dynamics*, IEEE Trans. Syst., Man, Cybern., vol. 27, no. 2, pp. 261-269, Mar.-Apr. 1997.
9. L. C. F. Araújo, L. H. R. Sucupira Jr., M. G. Lizárraga, L. L. Ling, and J. B. T. Yabu-uti, *A fuzzy logic approach in typing biometrics user authentication*, in Proc. 1st Indian Int. Conf. Artificial Intelligence, 2003, pp. 1038-1051.
10. W. G. de Ru and J. H. P. eloff *Enhaced password authentication through fuzzy logic*, IEEE Expert, vol. 17, no. 6, pp. 38-45, Nov.-Dec. 1997.

11. O. Coltell, J. M. badfa, and G. Torres, *Biometric identification system based in keyboard filtering*, in Proc. IEE 33rd Annu. Int. Carnahan Conf. Security Technology, 1999, pp. 203-209.
12. S. Haidar, A. Abbas, and A. K. Zaidi, *A multi-technique approach for user identification trhough keystroke dynamics*, in Proc. IEEE Int. Conf. Systems, Man and Cybernetics, vol. 2, 2000, pp. 1336-1341.
13. F. Monrose, M. K. Reiter, and S. Wetzel, *Password hardening based on keystroke dynamics*, in Proc. 6th ACM Conf. Computer Security, Singapore, Nov. 1999.
14. F. W. M. H. Wong, A. S. M. Supian, A. F. Ismail, L. W. Kin, and O. C. Soon, *Enhanced user authentication through typing biometrics with artificial neural network and k-nearest neighbor algorithm* in Conf. Rec. 35th Asilomar Conf. Signals, Syst., comput., Vol. 2, 2001, pp. 911-915.
15. L. C. F. Araújo, L. H. R. Sucupira Jr., M. G. Lizárraga, L. L. Ling, and J. B. T. Yabu-uti,*User authentication through typing biometrics features*, IEEE Trans. on Signal Processing, vol. 53, No. 2, Feb. 2005.
16. R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification* Wiley-Interscience Publication, 2nd Edition, Oct. 2000.
17. L. R. Rabiner, *A tutorial on hidden Markov models and selected applications in speech recognition*. Proc. of IEEE, 77, pp. 257-286, 1989.
18. Cambridge University Engineering Departament, *The HTK Book*, Cambridge University, 2002.
19. M. A. T Figueiredo and A. K. Jain, *Unsupervised learning of finite mixture models*, IEEE Trans. Pattern Anal. Machine Intell., vol. 24, no. 3, pp. 381-396, Mar. 2002.
20. T. Ord and S. M. Furnell, *User authentication for keypad-based devices using keystroke analysis*, Proc. Second International Network Conference (INC 2000), Plymouth, UK, pp. 263-272.