UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

DEPARTAMENTO DE COMUNICAÇÕES

Tese de Mestrado

Uma Metodologia para Autenticação Pessoal baseada em Dinâmica da Digitação

Lívia Cristina Freire Araújo

Orientador: Prof. Dr. João Baptista Tadanobu Yabu-uti

Co-orientador: Prof. Dr. Lee Luan Ling

Banca Examinadora:

Dr. Miguel Gustavo Lizárraga	(FEEC/UNICAMP)
Prof. Dr. Renato Baldini Filho	(FEEC/UNICAMP)
Prof. Dr. Yuzo Yano	(FEEC/UNICAMP)

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica e de Computação

Campinas – SP – Brasil

Fevereiro de 2004

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

Araújo, Lívia Cristina Freire
Ar15m Uma metodologia para autenticação pessoal baseada em dinâmica da digitação / Lívia Cristina Freire Araújo. --Campinas, SP: [s.n.], 2004.

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

1. Biometria. 2. Lógica difusa. 3. Reconhecimento de padrões. I. Yabu-uti, João Baptista Tadanobu. II. Ling, Lee Luan. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

RESUMO

Neste trabalho, apresentamos uma metodologia para autenticação de usuários baseada em seu ritmo de digitação em um teclado convencional de computador. A metodologia proposta é de baixo-custo, não-intrusiva e pode ser aplicada em um mecanismo de *login*-senha para aumentar a sua segurança.

Inicialmente, o usuário indica a conta a ser acessada e digita a informação alvo. Esta digitação é monitorada pelo sistema, que captura os tempos de pressionamento e soltura das teclas, além de seus códigos ASCII (American Standard Code for Information Interchange). A partir destes dados adquiridos, algumas características são extraídas e uma amostra é formada. Caso o usuário não esteja cadastrado, dez amostras são adquiridas para formar o conjunto de treinamento, e calcular um *template* para representá-lo. Caso o usuário já esteja cadastrado, a amostra adquirida é apresentada para o classificador que decide se ele é legítimo ou impostor. Cada usuário tem duas tentativas de ser autenticado em cada sessão. Finalmente, um mecanismo de adaptação pode ser ativado para atualizar o *template*.

A metodologia apresentada é avaliada envolvendo três tipos de usuários: legítimo, impostor simples e impostor observador. O usuário legítimo é o proprietário da conta, o impostor simples tem conhecimento sobre a informação alvo a ser digitada, e o impostor observador participa das sessões de autenticação do usuário legítimo. Alguns aspectos da metodologia são analisados para verificar os seus impactos nos resultados. Estes aspectos são: as características, o classificador, a informação alvo, o conjunto de treinamento, a precisão do tempo, as tentativas, o mecanismo de adaptação e o valor de limiar.

Este trabalho apresenta algumas inovações como: a extração das características em função das teclas utilizadas na digitação, o conjunto de características (código de teclas, duração de teclas e de duas latências de teclas) e a determinação do valor de limiar em função do desvio padrão das características. Além disto, os resultados obtidos (1.55% FRR e 1.91% FAR) são competitivos com os obtidos e publicados nesta área.

ABSTRACT

In this work, we present a methodology to authenticate users based on their typing rythm in a conventional computer keyboard. The methodology proposed is low-cost, unintrusive and could be applied in the login-password mechanism to increase its security.

Initially, the user indicates the account to be accessed and types the target string. This typing is monitored by the system and the key down and up times and the key ASCII codes (American Standard Code for Information Interchange) are captured. Using this captured data some features could be extracted and a sample is formed. If the user is a new one, ten samples are acquired to compound the training set, and to calculate a template to represent him. If the user is already enrolled, the sample captured is presented to the classifier that decides if he is legitimate or impostor. Each user has two chances to be authenticated in each session. Finally, an adaptation mechanism could be performed to update the template.

The methodology presented is evaluated involving three types of users: the legitimate, the simple impostor and the observer impostor. The legitimate user is the account's owner, the simple impostor is the user that knows the target string to type, and, the observer impostor is the user that observe a legitimate user's session of authentication. Some methodology's aspects are analyzed to verify the impact to the results. These aspects are: the features, the classifier, the target string, the training set, the timing accuracy, the attempts, the adaptation mechanism, and the threshold.

This work presents some innovations as: the features extraction in function of the keystrokes, the features set (the key code, the duration of the key and two keystroke latencies) and the threshold determination in function of the features standard deviation. Besides that, its results obtained (1.55% FRR e 1.91% FAR) are competitive to the ones obtained and published in this area.

Para o meu amado

Luiz

pelo seu amor e carinho por mim,

além da paciência e persistência,

para que tudo começasse e terminasse bem.

"No fim tudo dá certo, se não deu certo é porque ainda não chegou ao fim."

Fernando Sabino, escritor brasileiro

"A grandeza não consiste em receber honras, mas em merecê-las."

Aristóteles, filósofo grego

AGRADECIMENTOS

- A Deus, principalmente, por mais uma etapa cumprida da minha vida.
- A meus pais, Mário e Ana Cristina, por tudo o que sou hoje.
- Aos meus avós, Benedito (*in memorian*) e Antônia, por seu amor e carinho.
- Ao Prof. Yabu-uti, pela oportunidade oferecida, pela sua experiência de vida e pela sua compreensão, o que o tornou para mim mais que um orientador.
- Ao Miguel Lizárraga, pelas contribuições no desenvolvimento deste trabalho.
- Ao Prof. Marcos Negreiros, pelo seu incentivo e apoio nos momentos mais difíceis, e mais importante, pela sua amizade e carinho.
- Ao casal, Augusto e Érika, por todos os bons momentos que passamos juntos.
- A todos os membros do LRPRC, pela ajuda e paciência no processo de coleta de amostras para os experimentos desta dissertação.
- Ao CNPq pelo apoio financeiro concedido a esta pesquisa.
- E a todos que de alguma forma ajudaram na conclusão deste trabalho.

ÍNDICE

CAPÍTULO 2

CAPÍTULO 3

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE SÍMBOLOS E ABREVIATURAS

μ	Média
σ	Desvio Padrão
ASCII	American Standard Code for Information Interchange
c	Classe
ct	Código da Tecla
CT	Vetor Código de Teclas
d	Dimensão
EER	Equal Error Rate
FAR	False Acceptance Rate
FMR	False Match Rate
FNMR	False Non-Match Rate
FRR	False Rejection Rate
h	Quantidade de Caracteres
ia	Informação Alvo
IC	Intervalo de Confiança
K	Dados de Digitação
L	Conjunto de Auntenticação
LOC	Nível de Confiança
M	Vetor Medidas
n	Quantidade de Teclas
o	Outlier
PP	Vetor Pressiona-Pressiona

PS	Vetor Pressiona-Solta
RDTSC	Instrução Assembly Read Time-Stamp Counter
S	Conjunto de Treinamento
SP	Vetor Solta-Pressiona
tp	Momento do Tempo de Pressionamento da Tecla
ts	Momento do Tempo de Soltura da Tecla
T	Limiar de Decisão
v	Variância
w	Conta
x	Característica
X	Vetor de Características
Z	*Template*

CAPÍTULO 1

INTRODUÇÃO

Neste capítulo apresentaremos a motivação para o estudo da forma como cada pessoa digita em um teclado, também conhecida como dinâmica da digitação. Neste trabalho, o intuito é aplicá-la no mecanismo login-senha amplamente utilizado na autenticação de usuários. Em seguida, apresentaremos alguns conceitos relativos à biometria e reconhecimento de padrões. Finalmente, mostraremos os objetivos e a organização da dissertação.

1.1 Motivação

A tecnologia através do uso de computadores está presente em várias atividades em todo o mundo, desde pequenos comércios até bancos multinacionais. A utilização de computadores implica no armazenamento de informações que podem ser importantes e confidenciais, e tais informações podem ter os seus acessos restritos. O acesso não-autorizado às informações deste tipo pode implicar, em casos mais graves, em prejuízos significativos que podem levar, por exemplo, ao fracasso de um negócio. Uma das grandes ameaças relacionadas com acesso não-autorizado e com segurança de dados importantes e/ou confidenciais é conhecida como *impersonation*. Esta ameaça é caracterizada por uma pessoa tentar se passar por outra com o objetivo de acessar o sistema e os seus dados armazenados [1]. O processo de autenticação

fornece meios para auxiliar no combate a este tipo de ameaça. Este processo tem por finalidade identificar um indivíduo ou verificar se a identidade do mesmo é verdadeira ou falsa [1].

Existem basicamente três maneiras nas quais um indivíduo pode ser autenticado em um sistema [2]:

1. Com base em algo que só o indivíduo sabe, como por exemplo, uma senha ou um PIN (*Personal Identification Number*). Este tipo de processo de autenticação é chamado Prova por Conhecimento;

2. Com base em algo que só o indivíduo possui, como por exemplo, um cartão magnético ou um *smart card*. Este tipo de processo de autenticação é chamado de Prova por Posse;

3. Com base em algo que o indivíduo é, como uma medida fisiológica ou uma característica comportamental, que distingue, de forma confiável, um ser humano dos demais e que pode ser utilizado para autenticar a sua identidade. Este tipo de processo de autenticação é chamado de Prova por Biometria.

O mecanismo *login*-senha, que é um processo de autenticação do tipo Prova por Conhecimento, é o mais comumente utilizado, pois é de fácil manuseio e de baixo-custo, sendo necessário somente o uso de um teclado [3]. Porém, apesar destas vantagens, ele se torna frágil quando o usuário é displicente com sua senha, seja anotando-a ao invés de memorizá-la, ou escolhendo-a de forma óbvia, como o seu sobrenome, a sua data de nascimento, o time para qual torce, etc. A abordagem baseada na posse de cartões magnéticos também é muito utilizada, porém estes podem ser roubados ou falsificados. Os mecanismos baseados em biometria têm conquistado espaço atualmente em relação aos mecanismos clássicos de autenticação pessoal, por causa das vantagens de que suas características não podem ser roubadas, perdidas ou esquecidas, além do que uma característica biométrica de um indivíduo é única dentro de uma população [1]. Recentemente, o avanço nas tecnologias de aquisição de informação e o aumento do poder de processamento dos computadores também contribuíram para esta conquista de espaço.

Dentro desse contexto, nesta dissertação é proposta uma metodologia para aumentar a segurança do mecanismo de autenticação *login*-senha, agregando-lhe biometria através da dinâmica da digitação, também mencionada na literatura como *keystroke dynamics* [1].

Na próxima seção são apresentados alguns conceitos relacionados à biometria.

1.2 Biometria

A biometria é o ramo da ciência que estuda a mensuração dos seres vivos [4]. Tecnologias biométricas são definidas como "métodos automáticos de verificação ou identificação de identidade de uma pessoa viva baseados em características fisiológicas ou de comportamento" [5]. Vamos examinar algumas das palavras chaves encontradas nesta 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 mecanismo de captura de um sinal digital ou analógico de características biométricas de uma pessoa, o componente que trata do processamento dos sinais e extração das características biométricas, e, finalmente, o componente que realiza a tarefa de verificação/identificação automática a partir do sinal capturado e processado de uma pessoa. O termo "automático" demanda que uma vez realizada a aquisição dos dados, os processos que envolvem o processamento, classificação e finalmente o resultado da autenticação, sejam feitos sem intervenção humana.

Verificação versus Identificação: Um sistema biométrico pode ser classificado com relação à maneira como seus dados de entrada são comparados junto à base de dados. Neste caso, duas categorias podem ser definidas:

- Verificação: Uma comparação "um-para-um" é realizada, ou seja, a informação biométrica apresentada por um indivíduo é comparada com o *template* correspondente àquele indivíduo. *Template* é a representação das informações extraídas das amostras biométricas fornecidas pelo indivíduo no seu processo de cadastramento (*enrollment*).

- Identificação: Uma comparação "um-para-muitos" é realizada, ou seja, a informação biométrica apresentada por um indivíduo é comparada com todos os *templates* ou um conjunto deles armazenados na base de dados.

Pessoa viva: A interpretação deste termo reflete a identificação de que a informação biométrica que está sendo amostrada pertence a uma pessoa que está viva e presente, evitando assim fraudes com a utilização de amostras artificiais. Como exemplo, pode ocorrer que diante de um sistema de verificação de locutor, um indivíduo tente se passar por outro através da reprodução do som da voz de um dos usuários do sistema que tenha sido previamente gravada. Uma das soluções para este 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 faces. Neste caso, o sensor que faz a captura da imagem não é uma câmera de vídeo comum. Trata-se de dispositivo que além de capturar a imagem da face como um matriz de valores de intensidade de luz, capta também a distribuição de temperatura sobre as diferentes regiões do rosto. Dessa forma, ao apresentar uma foto comum como entrada para o sistema, mesmo que as características referentes à 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: Uma característica fisiológica é uma propriedade física relativamente estável, tal como as impressões digitais, geometria da mão, íris, faces, entre outras. 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 em processos de autenticação. Outras características de comportamento que podem ser utilizadas são a maneira como se digita nos teclados e a maneira de falar.

As características de comportamento tendem a variar com o tempo. Por este motivo, muitos sistemas biométricos permitem que sejam feitas atualizações de seus dados biométricos cadastrados à medida que estes vão sendo utilizados [6]. Em geral, ao executar a tarefa de atualização de dados, o sistema tornar-se-á mais eficiente em autenticar o indivíduo.

A figura 1.1 apresenta um diagrama de blocos com as abordagens de processos de autenticação associados a tecnologias 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 utilização da abordagem envolvendo características biométricas. A figura também mostra alguns exemplos de tipos de biometrias para as características fisiológicas e de comportamento previamente definidas.

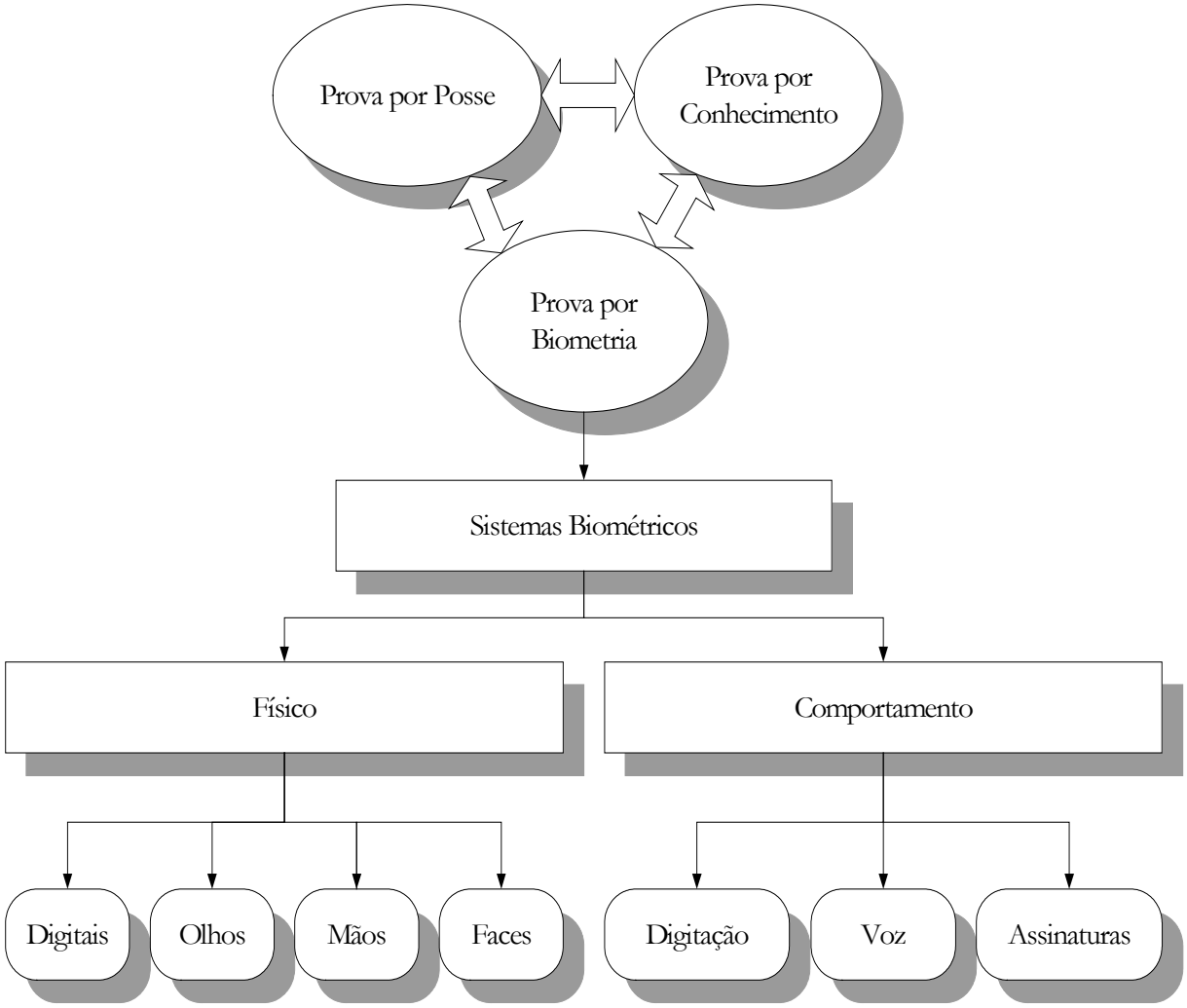


Figura 1.1: As abordagens de autenticação associadas a tecnologias biométricas.

Existem três passos básicos para realizar um processo de autenticação utilizando tecnologias biométricas [2]:

- Capturar o dado biométrico (Coleta de Dados);

- Analisar o dado biométrico capturado, e recapturá-lo caso necessário. Depois, processá-lo para criar uma amostra biométrica (Processamento de Sinais / Extração de Características);

- Comparar a amostra biométrica (Comparação) com o *template* gerado e armazenado anteriormente no cadastramento (Armazenamento). Esta comparação pode ser verificação ou identificação.

A amostra biométrica circula em um meio de comunicação (Transmissão) existente entre os componentes do sistema mencionados nos passos apresentados acima, conforme mostrado na figura 1.2.

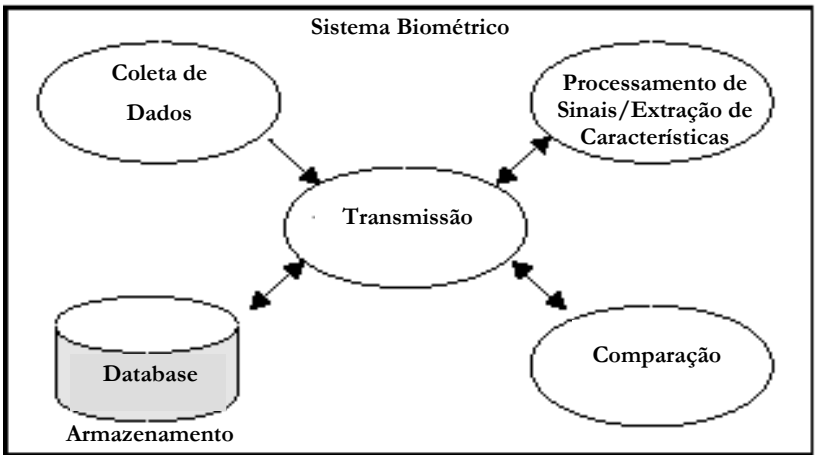


Figura 1.2: A Interatividade existente entre os Componentes de um Sistema Biométrico.

O desempenho de um sistema de autenticação é medido de acordo com dois tipos de erro: erro Tipo I e erro Tipo II. Na prática, o erro Tipo I, também conhecido como FAR (*False*

Acceptance Rate) ou FMR (*False Match Rate*), é uma estimativa da probabilidade do sistema aceitar uma pessoa não autorizada. O erro Tipo II, também conhecido como FRR (*False Rejection Rate*) ou FNMR (*False Non-Match Rate*), é uma estimativa da probabilidade do sistema rejeitar uma pessoa autorizada. O FAR e FRR são expressos em porcentagens e variam de acordo com um limiar de decisão escolhido.

As taxas FAR e FRR são comumente ilustradas por curvas, e podem ser observadas na figura 1.3.

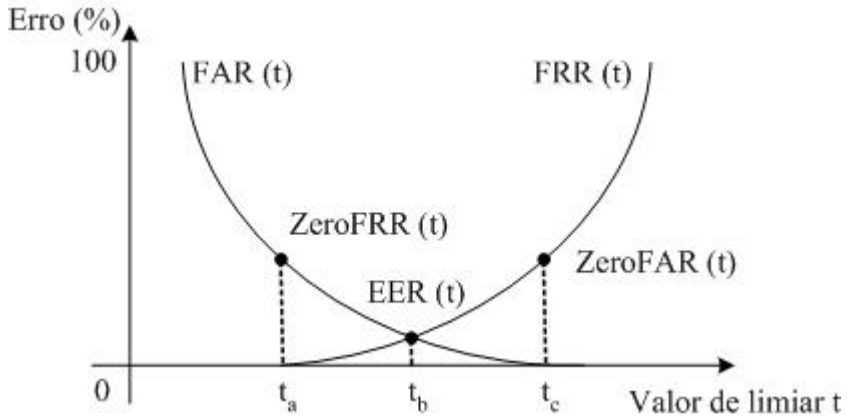


Figura 1.3: Curvas de FAR e FRR

A figura 1.3 mostra um exemplo de caso real de classificação, no qual em nenhum ponto referente a t ambas as taxas são iguais a zero. No caso ideal, existiriam um ou mais pontos referentes a t onde ambas as taxas de erro alcançadas seriam iguais a zero. Na figura 1.3, podemos observar três pontos importantes referentes à t_a, t_b e t_c, chamados de ZeroFRR, EER (*Equal Error Rate*), ZeroFAR, respectivamente.

EER é o ponto no qual os valores de FRR e FAR são iguais. Segundo [7], este ponto é o mais importante, pois especifica a separabilidade que o sistema oferece entre os acessos permitidos e os não-permitidos, embora em uma aplicação real um sistema de autenticação raramente consiga operar exatamente neste ponto. Na prática, sistemas são programados para trabalharem próximos a ZeroFRR ou ZeroFAR.

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 aceitos. 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.

Na próxima subseção é apresentada uma introdução sobre dinâmica da digitação, tecnologia biométrica base utilizada nesta dissertação.

1.2.1 Dinâmica da Digitação

Dinâmica da digitação é o processo de analisar a maneira como um usuário digita em um terminal, monitorando suas entradas em um teclado, no intuito de autenticá-lo baseado em seu ritmo de digitação habitual [8]. Esta tecnologia pode ser dividida em duas abordagens em relação ao momento da autenticação:

- Na abordagem estática, a autenticação é realizada no início da interação do usuário com o sistema, geralmente no momento do seu *logon*;

- Na abordagem contínua, a autenticação é realizada várias vezes no decorrer da interação do sistema com o usuário. 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 baseadas nos tempos capturados em cada uma das teclas digitadas por um indivíduo durante a digitação de uma palavra, frase ou texto. Assim, podemos citar as características mais utilizadas neste contexto:

- 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.

- Pressão da digitação: é a pressão aplicada pelo usuário em uma tecla.

Nesta dissertação, a última característica não será considerada, pois para a sua captura é necessário um teclado específico que capture a pressão aplicada por um usuário nas teclas.

Sabendo que a biometria é uma aplicação de reconhecimento de padrões, alguns conceitos referentes a esta área do conhecimento serão apresentados na próxima seção para melhor compreensão desta dissertação.

1.3 Reconhecimento de Padrões

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 [9]. O dicionário Aurélio [4] define padrão como "modelo oficial de pesos e medidas; protótipo, arquétipo". Na biometria, que é uma das aplicações de reconhecimento de padrões, o padrão poderia ser uma impressão digital, uma assinatura ou a dinâmica da digitação.

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 intraclasses (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 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 pertencem os padrões.

A autenticação do tipo verificação, que é a utilizada nesta dissertação, de forma ideal, pode ser apresentada como um problema de classificação com duas classes: uma classe com as amostras pertencentes a uma mesma pessoa e outra classe com as amostras não-pertencentes a esta pessoa.

As abordagens mais conhecidas em reconhecimento de padrões são [9][10]:

- Estatística: Nesta abordagem, um conjunto de treinamento é fornecido para cada classe com o objetivo de estabelecer fronteiras de decisões em um espaço de características, separando os padrões pertencentes a classes diferentes. As fronteiras de decisão são determinadas pelas distribuições de probabilidade dos padrões pertencentes a cada classe.

- Sintática: Nesta abordagem o padrão é visto como sendo composto por sub-padrões, que por sua vez são constituídos por sub-padrões mais simples, e esta subdivisão continua até que se alcance os sub-padrões elementares chamados de primitivas. O padrão é representado em termos da inter-relação entre estas primitivas. A abordagem sintática faz uma analogia entre a estrutura dos padrões e a sintaxe de uma linguagem, onde os padrões são as sentenças, as primitivas são o alfabeto, e as sentenças são geradas de acordo com uma gramática;

- Redes Neurais: Os modelos de redes neurais tentam usar alguns princípios de organização, como aprendizagem, generalização, adaptação e processamento distribuído, em uma rede de grafos direcionados e com pesos, nos quais os nodos são neurônios, e as extremidades direcionadas são conexões entre os neurônios de entrada e saída. As principais características desta abordagem são: a capacidade de aprender relações não-lineares complexas de entrada e saída, a utilização de procedimentos de treinamento seqüencial e a adaptação aos dados.

- 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, sendo uma generalização da lógica tradicional, que declara que qualquer premissa é verdadeira ou falsa, não ambas.

Estes modelos não são necessariamente independentes e existem vários trabalhos que aplicam sistemas híbridos.

Neste trabalho iremos empregar as abordagens estatística e nebulosa. Veremos o reconhecimento de padrões no sentido de classificação, isto é, associar um padrão de entrada a uma determinada classe.

1.3.1 Características

Uma maneira de classificar um objeto ou evento é fazendo medidas de suas propriedades ou características, o que está intimamente ligado ao problema específico que desejamos resolver. Por exemplo, para classificar uma letra pode ser útil saber sua área e seu perímetro. Podemos medir sua compressão através da razão entre sua área e o quadrado do seu perímetro. Podemos também medir seu grau de simetria com relação ao eixo horizontal, fazendo a comparação entre a área que fica na sua metade superior com a da metade inferior.

1.3.2 Vetor de Características

Na maioria das vezes, podemos medir um conjunto fixo de características de qualquer objeto ou evento que desejamos classificar. Sendo X o conjunto de características composto por $x_1, x_2, ..., x_d$. Neste caso, podemos pensar em X como sendo um vetor coluna de dimensão d (figura 1.4.a). De forma semelhante, podemos pensar que X representa um ponto em um espaço de características d-dimensional, como, por exemplo, um espaço tri-dimensional (figura 1.4.b).

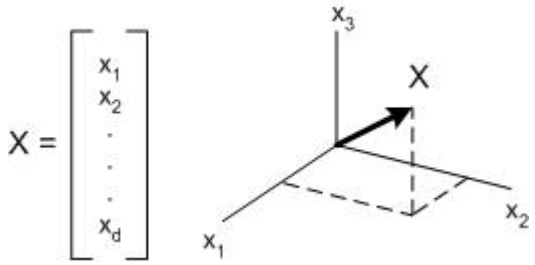


Figura 1.4: Vetor de características X e sua representação no espaço tri-dimensional.

Na figura 1.5 apresentamos 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

$x_1, x_2, ..., x_d$ que compõe o vetor de características X. A seguir, o módulo de classificação recebe X e associa-o a uma de suas c classes: classe 1, classe 2 ... classe c.

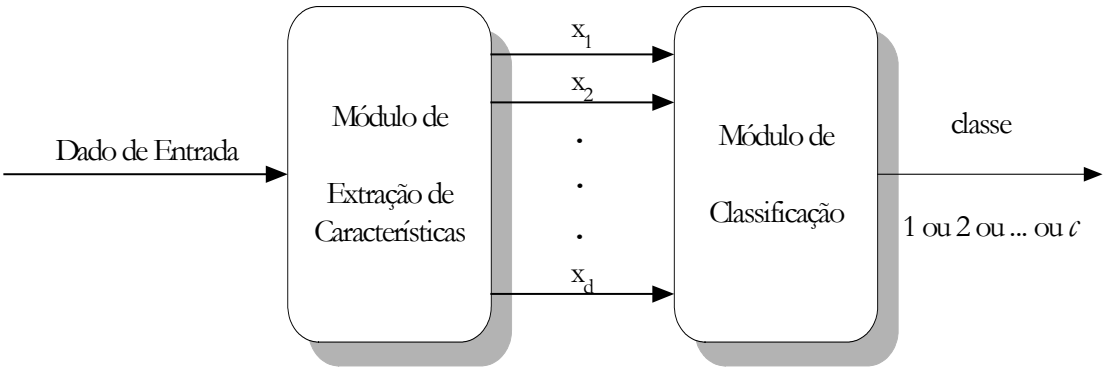


Figura 1.5: Módulos de extração de características e classificação.

A implementação do módulo de extração de 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 diferentes vetores de características, porém, espera-se que a variabilidade intraclasse seja pequena.

1.3.3 Classificadores simples

Matching é uma operação genérica em reconhecimento de padrões que é usado para determinar a similaridade entre duas entidades (pontos, curvas ou formas) do mesmo tipo. No casamento de padrões (*template matching*) um padrão montado a partir da amostra fornecida está disponível para comparação com o *template* previamente armazenado na base de dados do sistema [9]. Por exemplo, considere a letra D e a letra O adicionadas de um sinal ruidoso como mostrado na figura 1.6. As amostras sem ruído que estão à esquerda desta figura podem ser utilizadas como *templates*. Para classificar uma das amostras ruidosas, basta compará-la com os dois *templates*. Isto pode ser feito de duas maneiras equivalentes:

- Contar o número de concordâncias (pixel preto casa com pixel preto e pixel branco casa com pixel branco). Tomar a classe que tem o maior número de concordâncias. Esta abordagem é chamada de máxima correlação.

- Contar o número de discordâncias (pixel preto onde deveria ser pixel branco ou pixel branco onde deveria ser pixel preto). Tomar a classe que tem o menor número de discordâncias. Esta abordagem é chamada de mínimo erro.

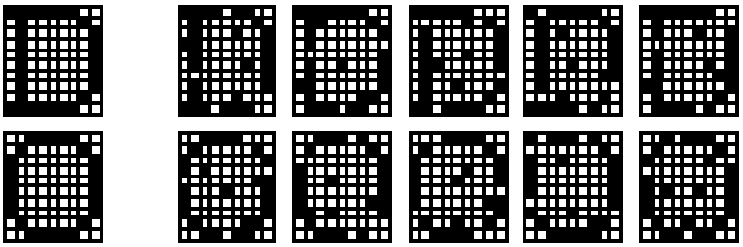


Figura 1.6: Exemplo das imagens das letras D e O "ruidosas"

O *template matching* funciona bem quando as variações na mesma classe são pequenas ou, como no caso do exemplo, quando existir "ruído aditivo". É claro que essa abordagem não funcionará bem em todos os problemas, como no caso em que existirem distorções na imagem de entrada como rotação, expansão, contração, oclusão, entre outras.

O *template matching* pode ser expresso matematicamente da seguinte forma:

Seja **X** o vetor de características de uma amostra desconhecida e sejam os vetores $\mathbf{m}_1$, $\mathbf{m}_2$, ... $\mathbf{m}_c$ os *templates* para cada uma das c classes. Então o erro do casamento entre **X** e $\mathbf{m}_k$ é dado por $\|\boldsymbol{X} - \mathbf{m}_k\|$, onde k = {1, 2, ... c}. Nesse caso $\|\mathbf{u}\|$ é a norma do vetor **u**. O classificador de erro mínimo calcula $\|\boldsymbol{X} - \mathbf{m}_k\|$ para k = 1 até c e escolhe a classe para a qual o erro é mínimo. Como $\|\boldsymbol{X} - \mathbf{m}_k\|$ é também uma medida de distância de **X** até $\mathbf{m}_k$, o classificador de erro mínimo também é chamado de classificador de distância mínima.

A figura 1.7 apresenta um diagrama de blocos de um classificador de distância mínima.

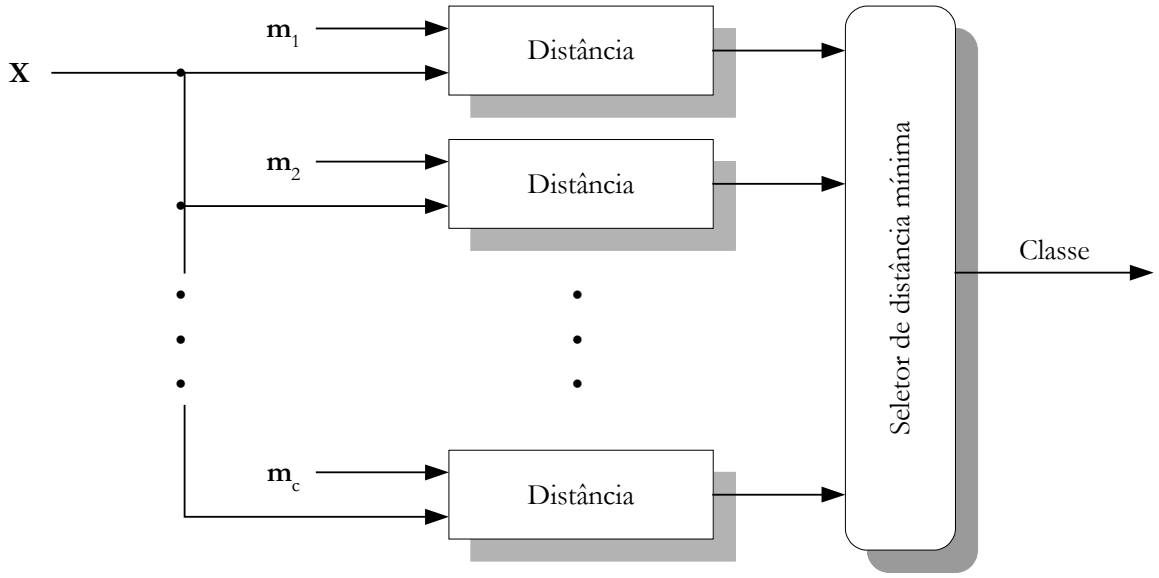


Figura 1.7: Diagrama de blocos de um classificador de distância mínima.

Existem muitas maneiras de definir a norma $\|\mathbf{u}\|$ que correspondem às diferentes métricas que podem ser utilizadas. As métricas mais utilizadas são [11]:

$$\text{Métrica Euclidiana: } \|\mathbf{u}\|_E = \sqrt{u_1^2 + u_2^2 + ... + u_d^2} \tag{1.1}$$

$$\text{Métrica Manhattan: } \|\mathbf{u}\|_M = |u_1| + |u_2| + ... + |u_d| \tag{1.2}$$

Utilizando o produto interno para expressar a distância euclidiana entre $\boldsymbol{X}$ e $\mathbf{m}_k$, podemos escrever:

$$\|\boldsymbol{X} - \mathbf{m}_k\|^2 = (\boldsymbol{X} - \mathbf{m}_k)'(\boldsymbol{X} - \mathbf{m}_k) \tag{1.3}$$

$$= \boldsymbol{X}'\boldsymbol{X} - \mathbf{m}_k'\boldsymbol{X} - \boldsymbol{X}'\mathbf{m}_k + \mathbf{m}_k'\mathbf{m}_k \tag{1.4}$$

$$= -2\left[\mathbf{m}_k'\boldsymbol{X} - 0{,}5\mathbf{m}_k'\mathbf{m}_k\right] + \boldsymbol{X}'\boldsymbol{X} \tag{1.5}$$

Note que o termo $\boldsymbol{X}'\boldsymbol{X}$ é o mesmo para todas as classes, ou seja para todo *k*. Para encontrar o *template* $\mathbf{m}_k$ que minimiza $\|\boldsymbol{X} - \mathbf{m}_k\|$ é suficiente encontrar $\mathbf{m}_k$ que maximize a

expressão entre colchetes, $\left[\mathrm{m}_k^{'}\boldsymbol{X}-0{,}5\mathrm{m}_k^{'}\mathrm{m}_k\right]$. Define-se a função de discriminante linear por [11]:

$$\mathrm{g}_k(\boldsymbol{X})=\mathrm{m}_{\boldsymbol{k}}^{'}\boldsymbol{X}-0{,}5\left\|\mathrm{m}_{\boldsymbol{k}}\right\|^2 \qquad \mathrm{k}=1,2,...,\mathrm{c} \tag{1.6}$$

Então, podemos dizer que o classificador euclidiano de distância mínima classifica um vetor de características de entrada $\boldsymbol{X}$ calculando c funções discriminantes lineares $\boldsymbol{g}_1(\boldsymbol{X})$, $\boldsymbol{g}_2(\boldsymbol{X})$,..., $\boldsymbol{g}_{\boldsymbol{c}}(\boldsymbol{X})$ e atribuindo $\boldsymbol{X}$ à classe corresponde a função discriminante máxima. Ainda, é possível analisar que as funções discriminantes lineares são medidas de correlação entre $\boldsymbol{X}$ e $\mathrm{m}_{\boldsymbol{k}}$, com a inclusão de $\left\|\mathrm{m}_{\boldsymbol{k}}\right\|^2$. Com esta inclusão, o classificador de distância euclidiana mínima é equivalente ao classificador de correlação máxima conforme a figura 1.8.

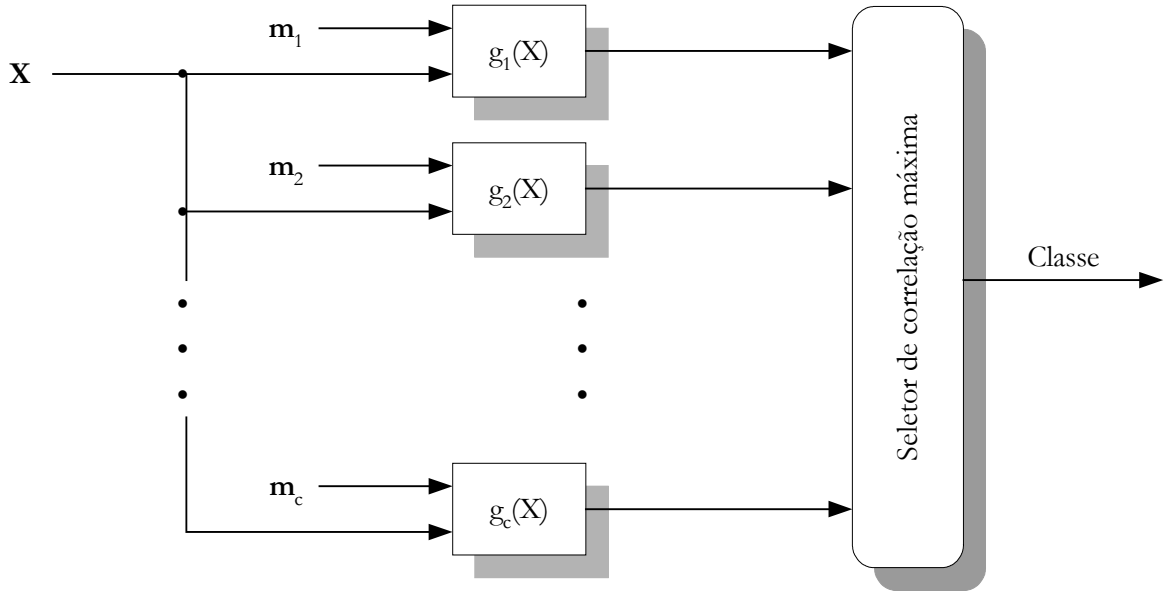


Figura 1.8: Diagrama de blocos de um classificador de correlação máxima.

Em geral, um classificador de padrões particiona o espaço de características em volumes chamados de regiões de decisão. Todos os vetores de características pertencentes a uma região de decisão são atribuídos à mesma classe. As regiões de decisão são separadas por superfícies

chamadas de fronteiras de decisão. Para um classificador de distância mínima, as fronteiras de decisão são compostas pelos pontos que se encontram eqüidistantes entre duas ou mais classes. Um classificador de distância mínima é de fácil entendimento e implementação, além de possuir um baixo custo computacional. Este classificador pode cometer erros de classificação quando ocorrem algumas das situações abaixo [11]:

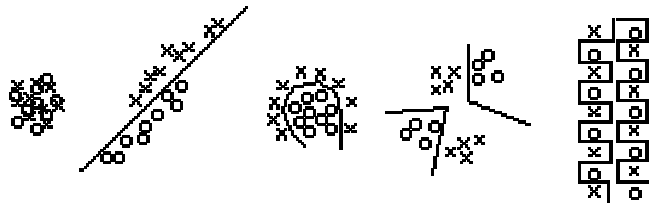
- As características podem ser inadequadas para distinguir a diferença entre as classes (figura 1.9 (a)). Se as características não possuírem informação suficiente para diferenciar as classes, nenhum classificador terá sucesso em separá-las;

- As características podem ser altamente correlacionadas (figura 1.9 (b)). Duas ou mais características são influenciadas por algum mecanismo comum e tendem a variar conjuntamente;

- O espaço de características não pode ser particionado por hiperplanos (figura 1.9 (c));

- Podem existir distintas subclasses entre os dados de treinamento, ou seja, freqüentemente acontece que as classes que foram previamente definidas não são classes "naturais" (figura 1.9 (d));

- O espaço de características pode ser muito complexo (figura 1.9 (e)).



(a) (b) (c) (d) (e)

Figura 1.9: Situações que podem ocasionar problemas de classificação para um classificador de distância mínima

1.3.4 Classificador de Distância Padrão

Algumas das limitações dos classificadores euclidianos de distância mínima, como por exemplo, os problemas devido aos diferentes valores de escala entre medidas dos vetores de características, podem ser superados utilizando-se um classificador de distância padrão.

Considere uma característica x presente no vetor de característica X. Suponha que temos e exemplos de padrões (amostras) representados por $a_1, a_2, ..., a_e$. Existem duas medidas estatísticas importantes que podemos utilizar para caracterizar este conjunto de amostras referentes a x, a média e a variância amostrais.

A média amostral μ_x é dada por:

$$\mu_x = \frac{1}{e}\sum_{j=1}^{e} a_j \tag{1.7}$$

A variância amostral υ_x é dada por:

$$\upsilon_x = \frac{1}{e-1}\sum_{j=1}^{e}\left(a_j - \mu_x\right)^2 \tag{1.8}$$

Observa-se que μ tem a mesma unidade que x, mas υ tem sua unidade ao quadrado. A raiz quadrada da variância é o desvio padrão σ, e possui a mesma unidade que x:

$$\sigma = \sqrt{\upsilon} \tag{1.9}$$

O valor numérico de uma característica x depende da unidade que é utilizada, isto é, de sua escala. Se x é multiplicado pelo fator z, então tanto a média como desvio padrão são multiplicados por z (A variância é multiplicada por z^2).

Às vezes é desejável colocar os dados numa mesma escala de modo que o desvio padrão resultante seja igual à unidade. Isto pode ser feito dividindo x pelo desvio padrão σ. De modo similar, ao medir a distância entre x e μ podemos ponderá-la pelo desvio padrão. Assim, chamamos de p a distância padrão a qual é dada pela equação (1.10):

$$p = \left| \frac{\mathrm{x} - \mu}{\sigma} \right| \tag{1.10}$$

Note que p é invariante tanto à translação quanto à escala. Isto sugere uma generalização importante para o classificador euclidiano de distância mínima.

A distância padrão entre o vetor de características X e o vetor de características de médias μ_X calculado a partir do conjunto de treinamento, é dada por:

$$p(\mu_X, X) = \sum_{j=1}^{d} \left| \frac{x_j - \mu_{x_j}}{\sigma_{x_j}} \right| \tag{1.11}$$

Esta distância tem a importante propriedade de ser invariante a escala. Isto significa que, ao utilizarmos esta medida, as unidades utilizadas nas várias características que compõem o vetor contribuem de forma equivalente no cálculo da distância. Porém, este classificador não resolve o problema apresentado na figura 1.9 (b), quando as características apresentam-se altamente correlacionadas. Neste caso, utiliza-se um classificador baseado na distância de Mahalanobis.

1.3.4.1 Classificador de Mahalanobis

A medida de distância r na equação

$$r^2 = (X - m_X)' C_X^{-1} (X - m_X) \tag{1.12}$$

é chamada de distância de Mahalanobis [11]. Na equação (1.12), X representa o vetor de características, m_x representa o vetor de características médio e C_x é a matriz de covariância para o vetor de características X. As superfícies onde r é constante são compostas por elipsóides que estão centrados ao redor do vetor de características médio m_x. No caso onde as características não são correlacionadas e as variâncias em todas as direções são as mesmas, estas superfícies tornam-se esféricas, e a distância de Mahalanobis é equivalente à distância padrão.

Fazendo uso de um classificador baseado na distância de Mahalanobis, é possível remover várias limitações ocasionadas pela utilização de um classificador baseado na distância euclidiana [11]:

1. Automaticamente ajusta a escala dos eixos das coordenadas;

2. Resolve o problema das características apresentarem-se altamente correlacionadas (figura 1.9 (b));

3. O espaço de característica pode ser particionado por hiperplanos ou por curvas utilizando o classificador de Mahalanobis, resolvendo o problema ilustrado na figura 1.9 (c).

Contudo, o classificador de Mahalanobis apresenta desvantagens, tais como [11]:

1. A matriz de covariância C_x pode ser difícil de se determinar com exatidão. C_x pode ser singular, caso a quantidade de amostras n for menor que a quantidade de características d menos 1, e, sendo assim, o termo da equação (1.12) C_X^{-1} não pode ser calculado. Mesmo que C_x não seja singular, para conseguir uma boa estimativa da matriz de covariância deve-se ter n próximo a $d(d-1)/2$.

2. Os requerimentos de memória e de tempo para o cálculo da matriz de covariância crescem quadraticamente com a quantidade de características.

A escolha do classificador de distância padrão em detrimento do classificador de Mahalanobis envolveu as duas desvantagens mencionadas, sendo principalmente o primeiro item com relação à quantidade de amostras necessárias por usuário para estimar a matriz de covariância C_x. Consideremos que a matriz de covariância C_x fosse sempre simétrica. Não se espera que tenhamos uma boa estimativa de C_x até que o total de amostras coletadas alcance o total de $d \times (d-1)/2$ elementos [11]. No nosso caso, por exemplo, um vetor de características que possua

10 características, necessitaria de um total de 45 amostras no mínimo para compor o conjunto de treinamento de um usuário, o que tornaria a metodologia inviável para uma aplicação prática.

1.3.5 Classificador Nebuloso (*Fuzzy*)

A utilização de sistemas nebulosos na forma de raciocínio aproximado representa um marco na tecnologia da informação, pois os sistemas baseados em regras nebulosas possuem grande habilidade para expressar a ambigüidade e subjetividade presentes no raciocínio humano [12]. Estes sistemas dependem da especificação de uma série de elementos, que incluem a quantidade e o tipo de regras nebulosas, os parâmetros das funções de pertinência, a semântica das regras que participam do raciocínio aproximado e os operadores do mecanismo de inferência utilizado para obter uma saída, a partir dos dados de entrada.

Nesse contexto, para se implementar um classificador nebuloso deve-se estabelecer as variáveis lingüísticas com seus referentes termos lingüísticos, as funções de pertinência, a base de regras nebulosa e o mecanismo de inferência. Cada um deles é definido a seguir:

- variáveis lingüísticas: variáveis cujos valores são palavras ou sentenças, ao invés de números;

- termos lingüísticos: valores em forma de palavra ou sentença que as variáveis lingüísticas possuem;

- função de pertinência $f(x)$: um conjunto nebuloso é definido por esta função que determina para o elemento x um grau de pertinência ao conjunto. Na teoria dos conjuntos nebulosos a transição entre pertencer e não pertencer é gradual. Em geral, o formato das funções de pertinência é restrito a uma certa classe de funções, representadas por alguns parâmetros específicos. Os formatos mais comuns são: triangular, trapezoidal e gaussiana. A escolha do formato mais adequado nem sempre é óbvia, podendo inclusive não estar ao alcance do conhecimento de um especialista para a aplicação em questão;

- base de regras nebulosa: é um conjunto de regras, também conhecidas como implicações nebulosas ou declarações condicionais nebulosas, que permitem uma maneira formal de

representação de diretivas e estratégias. As regras nebulosas são muito apropriadas quando o conhecimento do domínio resulta de associações empíricas e experiências do operador humano, ou quando se deseja uma representação lingüística do conhecimento adquirido. Em geral, as regras nebulosas assumem a forma "se <antecedente> então <conseqüente>". A expressão "se velocidade é alta então pressão é baixa" é um exemplo de uma regra nebulosa que relaciona as variáveis lingüísticas "velocidade" e "pressão", combinando os conjuntos nebulosos associados aos termos lingüísticos "alta" e "baixa".

- mecanismo de inferência: o modelo de Mamdani [13] é um dos mais utilizados modelos de inferência, que utiliza conjuntos nebulosos também no <conseqüente> das regras nebulosas. A saída final é representada por um conjunto nebuloso, resultado da agregação da saída inferida de cada regra. Para se obter uma saída final não nebulosa (*crisp value*), adota-se um dos métodos de transformação da saída nebulosa em não-nebulosa (*defuzzyfication*). Existem diferentes métodos que implementam esta transformação, dentre os mais utilizados destacam-se a média dos máximos, o centro de massa e o centro de área.

De uma forma geral, a obtenção da solução de um problema depende de um processo de busca dentro de um espaço de soluções potenciais. No caso particular, a determinação da estrutura de um sistema nebuloso, o espaço de busca que representa os possíveis modelos, é extremamente complexa. Isto porque envolve a definição de diferentes parâmetros relacionados à base de regras, a base de dados e aos mecanismos de inferência. A definição manual destes parâmetros deve ser feita por um especialista baseado em seu conhecimento [14].

1.4 Objetivos

O primeiro objetivo desta dissertação é desenvolver uma metodologia para autenticação de usuário do tipo verificação combinando o mecanismo de *login*-senha com a dinâmica da digitação em duas abordagens diferentes: estatística e nebulosa. As vantagens desta metodologia são apresentadas abaixo:

- maior segurança: será mais segura que um simples mecanismo de *login*-senha, pois serão agregadas a ela as vantagens das características biométricas;

- baixo-custo: utilizará apenas um teclado convencional para a aquisição de informações. Esta vantagem não é comum entre as tecnologias biométricas atualmente, que geralmente fazem uso de dispositivos de aquisição caros;

- não-intrusivo: a aquisição de amostras necessita apenas que o usuário digite seu *login* e sua senha. Para algumas tecnologias biométricas, a aquisição de amostras é intrusiva, como a da íris, quando uma varredura no olho é realizada utilizando um feixe de luz, fazendo com que muitos de seus usuários sintam-se desconfortáveis.

O segundo objetivo é apresentar um conjunto de características que possam ser aplicadas na autenticação via dinâmica da digitação, inovando para as pesquisas na área e contribuindo com resultados significativos.

O terceiro objetivo é que a metodologia possa ser aplicada na prática, e que seus resultados sejam competitivos com os obtidos em estudos já publicados na mesma área.

O quarto objetivo é apresentar uma ferramenta que automatize a metodologia, facilitando o processo de cadastramento e autenticação de usuários.

1.5 Estrutura da Dissertação

Com o intuito de fornecer uma apresentação estruturada do assunto, favorecendo uma seqüência lógica de idéias, esta dissertação é constituída de cinco capítulos:

1. Introdução
2. Estado da Arte em Dinâmica da Digitação
3. A Metodologia baseada em Dinâmica da Digitação
4. Resultados
5. Conclusões

No capítulo 2 apresentamos um estado da arte em dinâmica da digitação. A metodologia baseada em dinâmica da digitação proposta é descrita no capítulo 3, no qual detalhamos as características e os classificadores, e ainda, a ferramenta desenvolvida para automatizar todo o processo. No capítulo 4, os resultados dos experimentos realizados são apresentados e discutidos. Finalmente, os objetivos iniciais que foram alcançados, as contribuições e as perspectivas para novos trabalhos estão contidos no capítulo 5.

CAPÍTULO 2

ESTADO DA ARTE EM DINÂMICA DA DIGITAÇÃO

Neste capítulo abordamos os aspectos básicos em relação à dinâmica da digitação, como características e classificadores, relacionando-os com pesquisas publicadas na área. Ao final deste capítulo é apresentado um resumo das pesquisas referenciadas.

2.1 Introdução

A utilização de características provenientes da digitação foi sugerida por Spillane em 1975 [15]. Como declarado em [16], 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 identificar quem estava transmitindo a mensagem apenas escutando o som da digitação de pontos e vírgulas. 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 1895, Umphress & Williams [16] 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 & Leggett [17] estenderam o trabalho realizado em [16], mostrando que ele poderia ser utilizado em uma verificação estática no *login* em conjunto com uma frase, o que compensava a desvantagem apresentada em [16] de possuir textos extensos, que dificultavam a coleta de amostras. Em Leggett *et alli* [15], que também é uma extensão dos trabalhos apresentados em [16] e [17], 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 [3], [8], [10], [18-27] usando amostras com uma quantidade de caracteres menor a serem digitadas do que as apresentadas em [15], [16] e [17]. Atualmente, a aplicação comercial mais conhecida que utiliza a dinâmica da digitação para autenticação pessoal é o *Net Nanny's Biopassword* [28], porém os detalhes de sua implementação não são divulgados.

Os esquemas aplicados nesses trabalhos se encaixam na descrição que se segue. No momento da autenticação, o usuário digita a informação 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 *template*. Caso o usuário não esteja cadastrado, uma amostra é armazenada 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: 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 *template* do usuário em questão.

Nas próximas subseções, alguns aspectos importantes apresentados nos trabalhos mencionados são discutidos.

2.1.1 Informação Alvo

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 [3], quatro informações foram utilizadas como alvo (*login,* senha, primeiro nome, último nome). Em [25], 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 [19], 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 [20]. Em [19] uma das informações alvo analisada continha 31 caracteres, e, em [3], a informação alvo era composta por quatro informações, que totalizavam aproximadamente 28 caracteres.

2.1.2 Quantidade 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 [19]. A quantidade de amostras nos trabalhos pesquisados varia desde apenas três amostras [21] até trinta amostras [19] por usuário, enquanto em [10] a quantidade de amostras varia de acordo com cada usuário. Ainda em [10], 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. Finalmente, em [3], foi concluído que para obtenção de bons resultados, a utilização de menos do que seis amostras não é recomendada.

2.1.3 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 [24].

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 [22].

Os trabalhos referenciados nesta dissertação utilizaram em sua maioria a latência das teclas, mas em [22] e [23] 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. Na patente americana de Young & Hammon [18], o uso dessa característica é mencionado, mas os detalhes de sua aplicação não são mostrados.

Em todas as referências pesquisadas envolvendo dinâmica da digitação, as características são extraídas em função dos caracteres gerados, e não das teclas utilizadas. 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). Nesta dissertação, as características são calculadas em função das teclas. Além disso, discutiremos no capítulo 3 sobre uma característica inovadora gerada em função das teclas digitadas.

2.1.4 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 [23] e 1 segundo [25]. Nas conclusões de [26] foi mencionado que uma maior precisão era desejável para melhorar os resultados obtidos.

2.1.5 Tentativas de Autenticação

Em [26] 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 [19] 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 é chamada *shuffling*.

2.1.6 Atualização dos *Templates*

Como dito no capítulo 1 (seção 1.2), as características de comportamento tendem a variar com o tempo, e, por esse motivo, muitos sistemas biométricos permitem que sejam feitas atualizações em seus *templates,* na medida em que o sistema é acessado. Estas atualizações podem ser feitas de duas maneiras: realizando um novo cadastramento ou utilizando um mecanismo de adaptação. A maioria das referências pesquisadas não menciona a atualização de dados, mas em [24] um mecanismo de adaptação é adotado. Este mecanismo consiste na criação de um novo *template,* atualizado toda vez que uma autenticação é realizada com sucesso. A amostra mais antiga é descartada e substituída pela nova amostra.

2.1.7 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 [3], [19], [23] e [25]. Em [3], [19] e [25] um classificador de distância mínima é utilizado. Em [23] 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 [21], [22] e [27]. 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 classificados não são apropriados.

Outro tipo de classificador é o nebuloso (*fuzzy*) que foi utilizado em [10]. Neste trabalho, 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. O mecanismo de inferência aplicado foi o de Mamdani, e o de defuzzificação foi o centro de massa. 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. Em [19], foi mencionado que um classificador nebuloso foi experimentado, mas não foram obtidos resultados tão bons quanto os obtidos com o classificador estatístico.

Em [26] os classificadores estatístico, neural e nebuloso foram combinados. Em geral, os melhores resultados são obtidos com os classificadores estatísticos.

2.2 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 2.1 pode ser visto este resumo contendo: a informação alvo, a quantidade de amostras, as características utilizadas, o classificador aplicado e as taxas de erros obtidas (FAR e FRR). Além das referências já discutidas, foram adicionadas na tabela 2.1 duas publicações referentes ao trabalho que será apresentado nesta

dissertação envolvendo a metodologia para autenticação pessoal baseada em dinâmica da digitação utilizando um classificador nebuloso [30] e um classificador estatístico [45].

Tabela 2.1: Resumo das principais pesquisas relacionadas com os resultados obtidos

Pesquisa	Informação Alvo	Quantidade de Amostras	Características	Classificador	%FRR	%FAR
Joyce & Gupta (1990) [3]	*login*, senha, primeiro e último nomes	Oito	latência da digitação	estatístico	16.36	0.25
de Ru & Eloff (1997) [10]	*login* e senha	varia entre duas e dez	latência da digitação	nebuloso	7.4	2.8
Bleha et alli (1990) [19]	senha	Trinta	latência da digitação	estatístico	8.1	2.8
Robinson et alli (1998) [23]	*login*	Dez	latência e duração da digitação	estatístico	10.0	9.0
Haidar et alli (2000) [26]	senha	Quinze	latência da tecla	estatístico, nebuloso e neural	2.0	6.0
Araújo et alli (2003) [30]	senha	Oito	duas latências, duração da digitação e código da tecla	nebuloso	3,5	2,9
Araújo et alli (2004) [45]	Senha	Dez	duas latências, duração da digitação e código da tecla	estatístico	1,45	1,89

2.3 Aspectos na Dissertação

Com base nos estudo feitos no estado da arte em dinâmica da digitação, os aspectos mencionados anteriormente neste capítulo (seção 2.1) foram aplicados nesta dissertação. Com relação às configurações destes aspectos, elas são mostradas detalhadamente no capítulo 3 e sucintamente abaixo:

- A informação alvo deverá conter pelo menos onze caracteres de acordo com [20] e com os experimentos realizados e observados que serão apresentados no capítulo 4 (seção 4.2.3) desta dissertação;

- A quantidade de amostras coletadas por usuário foi determinada em dez amostras, mais do que essa quantidade aborrece os usuários;

- Várias características são analisadas, entre as quais estão a latência 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 (seção 2.1.3). Outra característica analisada é o código da tecla que surgiu em decorrência de como as características são calculadas nesta dissertação, como mencionado anteriormente em função das teclas utilizadas;

- A precisão do tempo aplicada é de um milisegundo, pois é compatível com a ordem de grandeza dos dados capturados, conforme será apresentado no capítulo 3 (seção 3.1.4);

- Duas tentativas são dadas a cada usuários de acordo com [26] e com os experimentos realizados e observados no capítulo 4 (seção 4.2.6);

- Um mecanismo de adaptação foi utilizado para manter o *template* atualizado;

- Dois tipos de classificadores são aplicados e analisados: o estatístico e o nebuloso.

No capítulo 4, faremos a análise dos resultados dos testes que foram realizados em função dos aspectos considerados.

CAPÍTULO 3

A METODOLOGIA BASEADA EM DINÂMICA DA DIGITAÇÃO

Neste capítulo apresentaremos a metodologia desenvolvida nesta dissertação para a verificação da identidade de usuários via dinâmica da digitação. Mostraremos os principais elementos que compõem a metodologia como: a informação alvo, a captura dos tempos, as características e o classificador. Por fim, apresentaremos a ferramenta implementada para a coleta das amostras de digitações e para a simulação dos experimentos.

3.1 Metodologia

A metodologia proposta nesta dissertação para a verificação da identidade de usuários via dinâmica da digitação está ilustrado no fluxograma da figura 3.1. Dois processos principais são 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 informação alvo ia_w, que é escolhida pelo usuário no cadastramento da conta. A digitação é monitorada pelo sistema, que captura os dados de digitação K. A partir destes dados, são extraídas as características que formam uma amostra. No processo de cadastramento, as amostras de digitação coletadas formam o conjunto de treinamento S_w. Uma vez que todas as amostras de

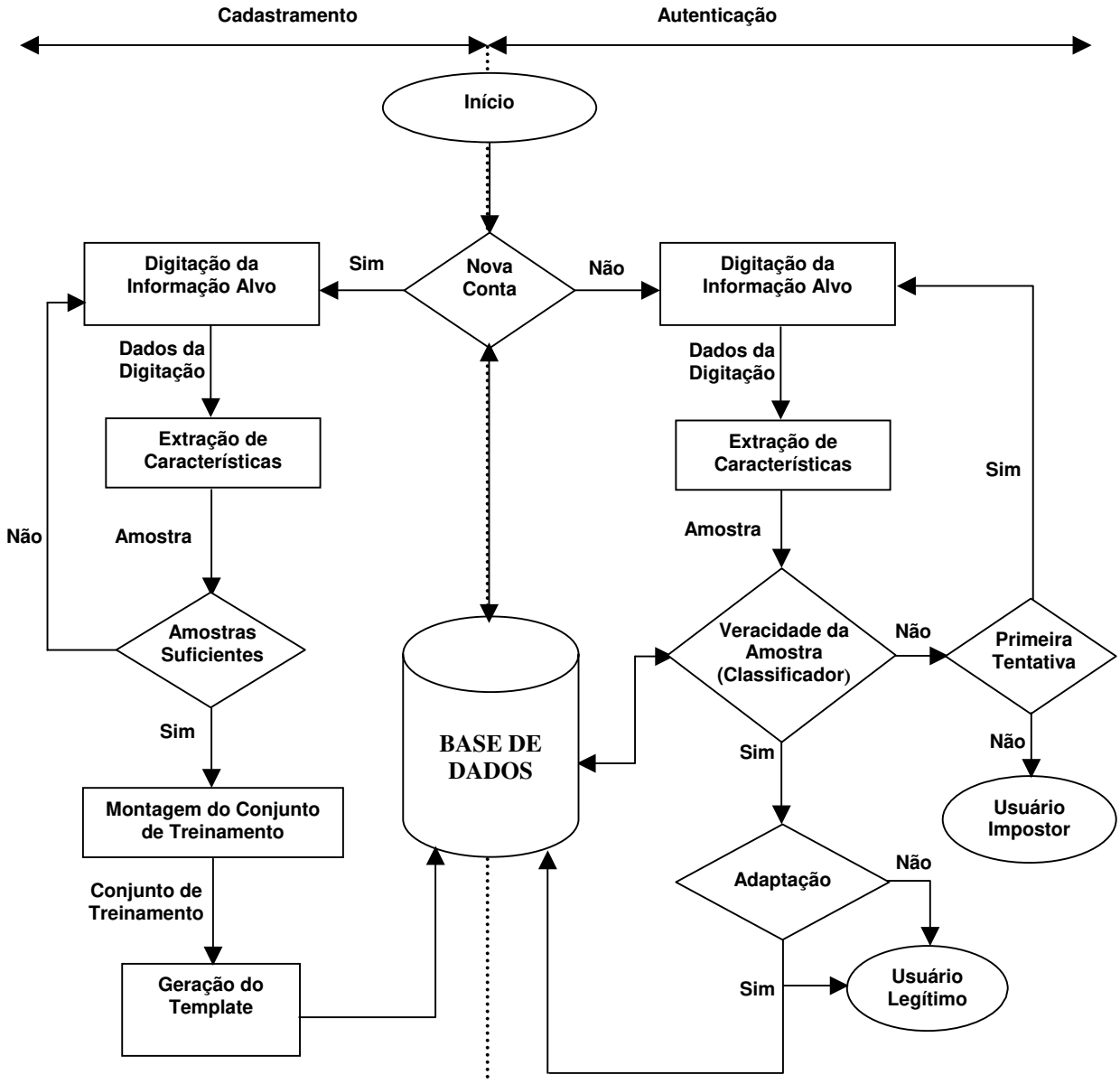


Figura 3.1: Fluxograma da metodologia

S_w tiverem sido coletadas, o *template* Z_w é gerado e contém informações importantes que representam o usuário em uma posterior sessão de autenticação. No processo de autenticação, as amostras coletadas relacionadas com a conta w formam o conjunto de autenticação L_w. Cada amostra de L_w é analisada pelo classificador que verifica, utilizando Z_w, se a amostra apresentada

pertence ao usuário proprietário da conta w. 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 ao sistema. Porém, se o classificador decidir pela falsidade da identidade do usuário, então a amostra é considerada falsa e uma nova tentativa de autenticação é concedida a ele. Nesta nova tentativa, se o classificador decidir novamente pela falsidade da identidade deste usuário, então ele é considerado um impostor. Desta maneira, uma sessão de autenticação é enquadrada em uma das seguintes três situações: uma tentativa com sucesso, uma com fracasso seguida de uma com sucesso, ou duas com fracasso. Finalmente, o *template* 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.

As particularidades e inovações desta metodologia são apresentadas e discutidas no decorrer deste capítulo.

3.1.1 Conta

De acordo com a metodologia ilustrada na figura 3.1, no processo de verificação o usuário deve informar a conta para acesso ao sistema e na qual se autenticará. O conjunto de c contas armazenadas na base de dados é representado por $W = \{w_1, w_2, ..., w_c\}$. Cada conta w possui quatro elementos relacionados: a informação alvo ia_w, o conjunto de treinamento S_w, o *template* Z_w e o conjunto de autenticação L_w. Os três primeiros elementos são fornecidos no cadastramento e o quarto na autenticação.

3.1.2 Informação Alvo

A informação 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 informação alvo pelo usuário, uma restrição é feita: ela deve conter pelo menos onze caracteres. Esta restrição está de acordo com [20], onde foi concluído que a quantidade de caracteres contida na informação alvo influencia na quantidade de erros de classificação, que aumentam quando a quantidade de caracteres é menor que dez.

Esta metodologia não trata erros tipográficos, então o usuário não deve cometer erros no momento da digitação da informação alvo. Porém, caso erros sejam cometidos, o usuário pode iniciar novamente sua digitação.

Exemplo 3.1: Para a conta 1, a informação alvo escolhida foi "UNICAMP Brasil", então este dado é representado por $ia_1 = \{"UNICAMP\ Brasil"\}$.

3.1.3 Dados de 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 $h \leq n$. A quantidade de teclas utilizadas pode ser maior que a quantidade de caracteres, pois alguns caracteres necessitam da utilização de mais de uma tecla, como, por exemplo, o caractere 'B', que necessita da combinação das teclas *shift* e B.

Os dados da digitação são representados por $K = \{k_1, k_2, ..., k_n\}$, onde cada k_i é constituído pelo código ASCII da tecla, pelo momento do tempo de pressionamento da tecla e pelo momento do 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. Os momentos do tempo mencionados são capturados em ciclos de *clock*, cuja abordagem é explicada na próxima seção.

Exemplo 3.2: Para uma digitação de $ia_1 = \{"UNICAMP\ Brasil"\}$ foram captados os dados de digitação K presentes na tabela 3.1.

Tabela 3.1: Exemplo de dados de digitação para $ia_1 = \{"UNICAMP\ Brasil"\}$

caractere	tecla	k	$k.ct$	$k.tp$	$k.ts$
	Shift	k_1	16	771340525000	772826744180
U	U	k_2	85	771496505220	771560206980
N	N	k_3	78	771649734920	771715578780
I	I	k_4	73	771669093500	771732641360
C	C	k_5	67	771871609000	771944194720
A	A	k_6	65	772059768820	772111914520
M	M	k_7	77	772358301300	772477492660
P	P	k_8	80	772424833120	772515030190
	Espaço	k_9	32	772645353040	772697246340
B	B	k_{10}	66	772792375280	772858291660
r	R	k_{11}	82	772993364840	773059432660
a	A	k_{12}	65	773073008060	773176919960
s	S	k_{13}	83	773151011540	773223853860
i	I	k_{14}	73	773366478040	773420628560
l	L	k_{15}	76	773480632580	773561116240

A ordem da ocorrência dos eventos de pressionamento e soltura das teclas, mencionada no exemplo 3.2 e conforme a tabela 3.1, pode ser melhor visualizada na figura 3.2.

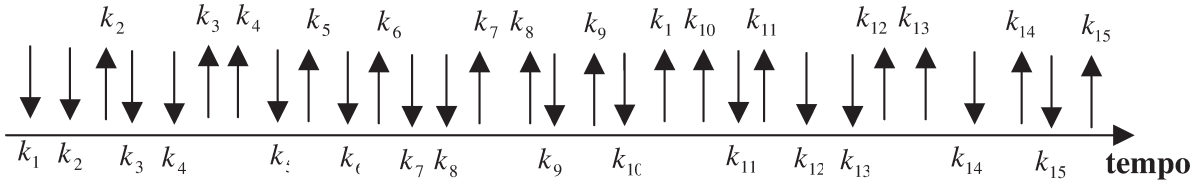


Figura 3.2: Ordem de ocorrência dos eventos de pressionamento e soltura das teclas

3.1.4 Captura do Tempo

A captura de 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 captado obtida é dependente do sistema operacional. Por exemplo, no Windows NT e no Windows 98, cada um deles apresenta um *delay* de aproximadamente 10 e 50 milissegundos, respectivamente [31][32]. Este *delay* corresponde ao intervalo de tempo necessário para que um processo requisitado para execução pelo sistema operacional seja atendido pelo processador.

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

Nesta dissertação, é utilizado, para a captura dos tempos de digitação, os ciclos de *clock*, pelo fato de serem mais precisos que a hora do sistema, pois eliminam o *delay* occasionado pelo sistema operacional na requisição de processos (tempo de requisição, espera e concessão para execução) ao processador, criando uma independência do tipo de sistema operacional.

A contagem dos ciclos de *clock* que ocorrem no processador é armazenada no contador *time-stamp* [33]. Para acessar este contador, é utilizada a instrução *assembly* RDTSC (*read time-stamp counter*). Para que a instrução RDTSC seja aplicada da forma desejada, captando os tempos reais de pressionamento e soltura de uma tecla, e impedindo que outra instrução do sistema influencie nestes valores de tempo captados, uma instrução de serialização é requerida. Uma instrução de serialização irá forçar que a execução de cada instrução precedente no código esteja completa antes de qualquer outra posterior a ela seja executada. Foi utilizada a função de serialização CPUID para forçar a execução em série e em ordem da instrução RDTSC.

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 3.1:

$$tempo = \left(\frac{ciclos}{freqüência} \right) \times 1000 \tag{3.1}$$

onde *tempo* é o valor em milisegundos, *ciclos* é o valor em ciclos de *clock* e *freqüência* é a velocidade do processador. A freqüência é dada em Hz (ciclos/segundo).

Os intervalos de tempo nesta dissertação são calculados com uma precisão de 1 milissegundo, pois em uma análise feita nos intervalos de tempos calculados, 98% destes intervalos estavam entre 10 e 900 milissegundos. Do resultado de *tempo*, calculado pela equação 3.1, a parte inteira é então extraída, refletindo a precisão de 1 milissegundo proposta. Desta maneira uma precisão menor, como 10 milissegundos, não seria desejada, pois alguns valores perderiam as suas respectivas significâncias. No caso de uma precisão maior, como 0.1 milissegundos, não traria mudanças significativas, pois quase todos os dados são maiores que 10 milissegundos.

Exemplo 3.3: Para um intervalo de tempo medido em 63701760 ciclos de *clock* em um processador de 1GHz, o seu valor equivalente em milisegundos, calculado pela equação 3.1, é de 63.701760 milissegundos. Aplicando a precisão de 1 milissegundo, obtemos o valor final de tempo de 63 milisegundos.

3.1.5 Características

As características são extraídas dos dados de digitação de uma informação alvo. Elas devem representar as classes (contas) envolvidas de maneira discriminante. Nesta dissertação são analisados cinco vetores de características para este propósito. Como foi mencionado no capítulo 2 (seção 2.1.3), a extração das características nesta dissertação é feita em função das teclas envolvidas, enquanto que nas pesquisas referenciadas [3], [8], [10], [18]-[27] é feita em função dos caracteres. Assim, a dimensão (características) dos vetores de características deste trabalho é maior ou no pior caso igual a dos vetores de características das pesquisas mencionadas anteriormente, pois para digitar uma informação alvo com h caracteres, é necessário que n teclas sejam utilizadas, sendo, neste caso, $h \leq n$. Portanto, uma quantidade maior de características é utilizada neste trabalho para proporcionar uma maior discriminação das classes. Os tamanhos dos vetores de características são dados em função de n. O vetor de características M é uma exceção, pois é extraído com base nos demais vetores utilizados neste trabalho, e não nos dados de digitação, conforme será apresentado posteriormente.

Exemplo 3.4: Utilizando os dados da digitação captados no exemplo 3.2, o tamanho de um vetor composto por latências de digitação é:

(a) Treze, se as latências são extraídas em função dos caracteres. Cada latência está relacionada a cada um dos intervalos formados: (U, N), (N, I), (I, C), (C, A), (A, M), (M, P), (P,), (, B), (B, r), (r, a), (a, s), (s, i) e (i, l);

(b) Quatorze, se as latências são extraídas em função das teclas. Cada latência está relacionada a cada um dos intervalos formados: (*shift*, U), (U, N), (N, I), (I, C), (C, A), (A, M), (M, P), (P, espaço), (espaço, B), (B, R), (R, A), (A, S), (S, I) e (I, L).

A seguir é descrito cada um dos cinco vetores de características utilizados nesta dissertação.

3.1.5.1 Vetor de características *CT*

O primeiro vetor de características é o Código de Teclas (*CT*), que contém os códigos (decimal) ASCII das teclas utilizadas na digitação de uma informação alvo. Desta maneira, para os dados de digitação $K = \{k_1, k_2, ..., k_n\}$ o vetor de características *CT* extraído é representado por $CT = \{ct_1, ct_2, ..., ct_n\}$, onde $ct_i = k_i.ct$.

Exemplo 3.5: Utilizando os dados da digitação captados no exemplo 3.2 e apresentados na tabela 3.1, o vetor de características *CT* extraído pode ser visualizado na figura 3.3.

$$CT' = [16\ \ 85\ \ 78\ \ 73\ \ 67\ \ 65\ \ 77\ \ 80\ \ 32\ \ 66\ \ 82\ \ 65\ \ 83\ \ 73\ \ 76]$$

Figura 3.3: Exemplo de vetor de características *CT*

De acordo com os dados de digitação do exemplo 3.2 e o vetor de características *CT* extraído destes dados no exemplo 3.5, podemos observar que o usuário utiliza a tecla *shift* apenas uma vez, mantendo-a pressionada para que todos as letras maiúsculas sejam geradas. A mesma informação alvo poderia ser digitada por outros usuários, utilizando um conjunto de teclas diferentes, como, por exemplo, utilizando a tecla *caps lock*, ou, pressionando a tecla *shift* mais de

uma vez. Por isso, as informações contidas neste vetor de características tornam-se discriminantes entre os usuários. Se a informação alvo possuísse somente letras minúsculas, neste caso, este vetor de características não seria discriminante: um mesmo vetor de características *CT* seria extraído sempre.

Este vetor de características é uma inovação desta dissertação, pois as pesquisas da área [3], [8], [10], [18]-[27] analisam somente os caracteres (e não as teclas que foram digitadas) provenientes da digitação das teclas correspondentes à informação alvo escolhida, como, por exemplo, o mecanismo de *login*-senha. Porém, como vimos, esta característica é importante, pois se torna mais discriminante quando a informação alvo possui letras maiúsculas.

3.1.5.2 Vetor de características *PP*

O segundo vetor de características é o pressiona-pressiona (*PP*), que contém 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. Desta maneira, para os dados de digitação K, o vetor de características *PP* extraído é representado por $PP = \{pp_1, pp_2, ..., pp_{n-1}\}$ onde $pp_i = k_{i+1}.tp - k_i.tp$ e está relacionado ao intervalo (k_i, k_{i+1}).

Exemplo 3.6: Utilizando os dados da digitação captados no exemplo 3.2, o vetor de características *PP* extraído pode ser visualizado na figura 3.4. Os valores apresentados na figura 3.4 estão em milissegundos.

$$pp_1 = (771496505220 - 771340525000)/1000000 = 155$$
$$pp_2 = (771649734920 - 771496505220)/1000000 = 153$$
:
$$pp_{14} = (773480632580 - 773366478040)/1000000 = 114$$

$$PP' = [155\ \ 153\ \ 19\ \ 202\ \ 188\ \ 298\ \ 66\ \ 220\ \ 147\ \ 200\ \ 79\ \ 78\ \ 215\ \ 114]$$

Figura 3.4: Exemplo de vetor de características PP

As latências de digitação contidas neste vetor de características são extraídas segundo a primeira maneira mencionada no capítulo 2 (seção 2.1.3), ou seja, o cálculo é realizado pela diferença entre o pressionamento de teclas sucessivas. A segunda maneira é abordada nesta dissertação como uma característica distinta através do vetor de características *SP*.

3.1.5.3 Vetor de características *SP*

O terceiro vetor de características é o solta-pressiona (*SP*), que contém as latências de digitação calculadas a partir da diferença entre o instante da soltura de uma tecla e o instante do pressionamento da tecla sucessiva utilizada na digitação da informação alvo. Desta maneira, para os dados de digitação K, o vetor de características solta-pressiona extraído é representado por $SP = \{sp_1, sp_2, ..., sp_{n-1}\}$ onde $sp_i = k_{i+1}.tp - k_i.ts$ está relacionado ao intervalo (k_i, k_{i+1}).

Exemplo 3.7: Utilizando os dados da digitação captados no exemplo 3.2, o vetor de características *SP* extraído pode ser visualizado na figura 3.5. Os valores apresentados na figura 3.5 estão em milissegundos.

$$sp_1 = (771496505220 - 772826744180)/1000000 = -1330$$
$$sp_2 = (771649734920 - 771560206980)/1000000 = 90$$
$$\vdots$$
$$sp_{14} = (773480632580 - 773420628560)/1000000 = 60$$

$$SP' = [-1330 \;\; 90 \;\; -46 \;\; 139 \;\; 116 \;\; 246 \;\; -53 \;\; 130 \;\; 95 \;\; 136 \;\; 14 \;\; -26 \;\; 143 \;\; 60]$$

Figura 3.5: Exemplo de vetor de características *SP*

As latências de digitação contidas neste vetor podem ser negativas. Na figura 3.6, são ilustradas duas situações para a ordem no tempo do momento dos pressionamentos e solturas de k_1 e k_2: (a) k_2 é pressionada depois da soltura de k_1; (b) k_2 é pressionada antes da soltura de k_1. Para uma característica sp referente a (k_1, k_2), na situação (a), o valor de sp é positivo, e, na situação (b), o valor de sp é negativo. A segunda situação ocorre bastante à medida que os

usuários se acostumam com a digitação da informação alvo, passando a pressionar mais teclas em um curto espaço de tempo. Dessa forma, pode ocorrer de um usuário pressionar uma tecla sucessora sem antes soltar sua antecessora. Como observado no exemplo 3.7 e na figura 3.5, quatro dos elementos deste vetor ($sp_1, sp_3, sp_7, sp_{12}$) ocorreram nesta situação (b).

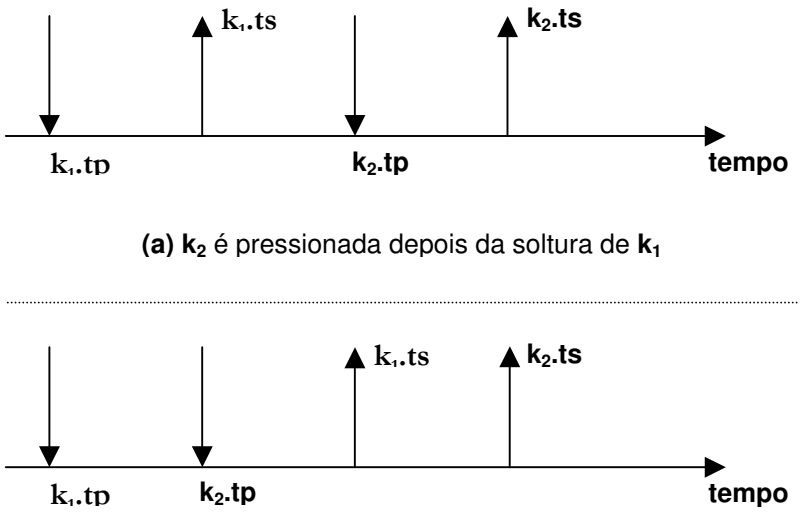


Figura 3.6: Situações em que a característica sp resulta em valores (a) positivos (b) negativos

3.1.5.4 Vetor de características *PS*

O quarto vetor de características é o pressiona-solta (*PS*), que contém as durações em que as teclas permanecem pressionadas. A duração de digitação é o intervalo de tempo que a tecla permanece pressionada. As durações de digitação são calculadas pela diferença entre o instante de pressionamento e o instante de soltura de uma tecla utilizada na digitação da informação alvo. Desta maneira, para os dados de digitação K o vetor de características *PS* extraído é representado por $PS = \{ps_1, ps_2, ..., ps_n\}$ onde $ps_i = k_i.ts - k_i.tp$ está relacionado a k_i.

Exemplo 3.8: Utilizando os dados da digitação captados no exemplo 3.2, o vetor de características PS extraído pode ser visualizado na figura 3.7. Os valores apresentados na figura 3.7 estão em milissegundos.

$$ps_1 = (772826744180 - 771340525000)/1000000 = 1487$$

$$ps_2 = (771560206980 - 771496505220)/1000000 = 64$$

:

$$ps_{15} = (773561116240 - 773480632580)/1000000 = 80$$

$$PS\ ' = [1487\ \ 64\ \ 66\ \ 64\ \ 73\ \ 52\ \ 119\ \ 90\ \ 52\ \ 66\ \ 66\ \ 104\ \ 73\ \ 54\ \ 80]$$

Figura 3.7: Exemplo de vetor de características *PS*

3.1.5.5 Vetor de características *M*

O quinto vetor de características é o medidas (*M*), que contém características extraídas dos vetores de características apresentados, com exceção do *CT*. Estas características são: a média, o desvio padrão, o valor máximo e o valor mínimo dos tempos em milissegundos encontrados nos vetores de características *PP*, *SP* e *PS*. Desta maneira, o vetor de características medidas possui doze elementos e é representado por $M = \{m_1, m_2, ..., m_{12}\}$, onde:

- m_1 é a média dos tempos do vetor de característica *PP*;
- m_2 é o desvio padrão dos tempos do vetor de característica *PP*;
- m_3 o valor máximo entre os tempos do vetor de característica *PP*;
- m_4 é o valor mínimo entre os tempos do vetor de característica *PP*;
- m_5 é a média dos tempos vetor de característica *SP*;
- m_6 é o desvio padrão dos tempos do vetor de característica *SP*;
- m_7 é o valor máximo entre os tempos do vetor de característica *SP*;
- m_8 é o valor mínimo entre os tempos do vetor de característica *SP*;
- m_9 é a média dos tempos do vetor de característica *PS*;
- m_{10} é o desvio padrão dos tempos do vetor de característica *PS*;
- m_{11} é o valor máximo entre os tempos do vetor de característica *PS*;

- m_{12} é o valor mínimo entre os tempos do vetor de característica *PS*;

onde a média é dada pela equação 3.2, o desvio padrão pela equação 3.3, o valor máximo pela equação 3.4 e o valor mínimo pela equação 3.5, para um dado vetor X de tamanho d.

$$\mu_X = \frac{1}{d}\sum_{i=1}^{d} x_i \tag{3.2}$$

$$\sigma_X = \frac{1}{d-1}\sum_{i=1}^{d} |x_i - \mu_X| \tag{3.3}$$

$$\max_X = \vee(x_1, x_2, ..., x_d) \tag{3.4}$$

$$\min_X = \wedge(x_1, x_2, ..., x_d) \tag{3.5}$$

Este vetor de características não foi analisado em nenhuma das pesquisas referenciadas da área [3], [8], [10], [18]-[27], pois elas analisam seus vetores de características baseados em latências de digitação, ou com latências e durações de digitação, e não levam em consideração as características extraídas dos tempos coletados destes vetores, como a média e o desvio padrão.

Exemplo 3.9: Utilizando os dados da digitação captados no exemplo 3.2, o vetor de características *M* extraído pode ser visualizado na figura 3.8. Os valores apresentados na figura 3.8 estão em milissegundos.

$m_1 = (155 + 153 + 19 + 202 + 188 + 298 + 66 + ... + 78 + 251 + 114)/14 = 2134/14 = 152$

$m_2 = (|155 - 152| + |153 - 152| + ... + |251 - 152| + |114 - 152|)/14 = 824/13 = 63$

$m_3 = \vee(156, 153, 19, 203, 188, 299, 67, ..., 78, 251, 114) = 299$

$m_4 = \wedge(156, 153, 19, 203, 188, 299, 67, ..., 78, 251, 114) = 19$

:

$$m_{12} = \wedge(1487, 64, 66, 64, 73, 52, 119, ..., 73, 54, 80) = 52$$

$$M' = [152\ \ 63\ \ 299\ \ 19\ \ -20\ \ 213\ \ 246\ \ -1330\ \ 167\ \ 176\ \ 1487\ \ 52]$$

Figura 3.8: Exemplo de vetor de características *M*

3.1.6 Amostra

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

As amostras provindas do cadastramento formam o conjunto de treinamento S. Uma vez que este conjunto está completo, um *template* pode ser gerado. Nesta dissertação, um conjunto de treinamento completo é composto de uma quantidade de amostras igual a dez. Pelos experimentos conduzidos neste trabalho, esta quantidade é razoável para os usuários envolvidos no processo, ou seja, o processo de aquisição não os incomodou em demasia nem os irritou. Além disto, as quantidades utilizadas pelas principais pesquisas consultadas [3], [10], [23] e [26] apresentavam valores que estavam no intervalo de 8 a 15 amostras. Desta maneira, o conjunto de treinamento da conta w é representado por $S_w = \{s_1, s_2, ..., s_{10}\}$, onde cada s é uma amostra provinda do cadastramento, que, por sua vez, é composta pelos vetores de características apresentados.

As amostras presentes no conjunto de treinamento $\{s_1, s_2, ..., s_{10}\}$ referentes a uma mesma característica x são representadas por $s_1.x, s_2.x, ..., s_{10}.x$. A única restrição que ressaltamos com relação às amostras que formam o conjunto de treinamento é que todas devem ter o mesmo vetor de características CT, pois, como já foi mencionado, 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 y amostras de autenticação na conta w é representado por $L_w = \{l_1, l_2, ..., l_y\}$, onde cada l_i é

uma amostra provinda da autenticação e, é composta por uma *sessão*, por uma *tentativa*, e pelos vetores de características. Uma *sessão* é composta por até duas *tentativas* de autenticação. Sendo, para uma amostra l, com $sessão = i$ e $tentativa = j$, então esta amostra é referente a i-ésima autenticação iniciada em uma conta w, com $i < \infty$, e, a j-ésima vez que o usuário tentou autenticar-se na sessão i, com $j \leq 2$. Os usuários em cada sessão podem tentar autenticar-se uma segunda vez, caso falhe na primeira, como será explicado na seção 3.1.9.

3.1.7 *Template*

Um *template* 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 *template* Z_w gerado é composto por informações extraídas dos cinco vetores de características apresentados: CT, PP, PS, SP, M. Com relação ao vetor de características CT, a informação 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. Desta maneira, o template Z é composto pelos seguintes elementos: $(CT, \mu_{PP}, \sigma_{PP}, \mu_{SP}, \sigma_{SP}, \mu_{PS}, \sigma_{PS}, \mu_M, \sigma_M)$. Para um vetor de característica X representativo dos vetores de características PP, PS, SP e M, o cálculo de $\mu_X = \{\mu_{x_1}, \mu_{x_2}, ..., \mu_{x_d}\}$ e $\sigma_X = \{\sigma_{x_1}, \sigma_{x_2}, ..., \sigma_{x_d}\}$ segue o procedimento abaixo em dois passos:

1. A média e o desvio padrão são calculados para cada característica x_i através das 10 amostras contidas no conjunto de treinamento de acordo com as equações 3.6 e 3.7.

$$\mu_{x_i} = \frac{1}{10}\sum_{j=1}^{10} s_j.x_i \tag{3.6}$$

$$\sigma_{x_i} = \sqrt{\frac{1}{10-1}\sum_{j=1}^{10}\left(s_j.x_i - \mu_{x_i}\right)^2} \tag{3.7}$$

2. A média e desvio padrão são novamente calculados, mas sem os valores que são considerados *outliers* [3]. *Outliers* o são valores que se desviam, de acordo com a equação 3.8, dos valores das características contidas nas amostras do conjunto de treinamento. Assim, as médias e os desvios padrões calculados neste passo refletem melhor as características de um usuário do que os calculados no passo anterior, com a eliminação dos *outliers*. Para uma característica x_i existem *u* valores que não são *outliers,* com $u \leq 10$. Neste contexto, os cálculos da média e do desvio padrão seguem as equações 3.9 e 3.10.

$$o_{s_j.x_i} = \begin{cases} falso & \left(\mu_{x_i} - 3\sigma_{x_i}\right) \leq s_j.x_i \leq \left(\mu_{x_i} + 3\sigma_{x_i}\right) \\ verdadeiro & c.c. \end{cases} \tag{3.8}$$

$$\mu_{x_i} = \frac{1}{u}\sum_{j=1}^{10}\left(s_j.x_i \middle| o_{s_j.x_i} = falso\right) \tag{3.9}$$

$$\sigma_{x_i} = \sqrt{\frac{1}{u-1}\sum_{j=1}^{10}\left(\left(s_j.x_i - \mu_{x_i}\right)^2 \middle| o_{s_j.x_i} = falso\right)} \tag{3.10}$$

Exemplo 3.10: Utilizando o procedimento descrito nesta seção, os elementos μ_{pp} e σ_{pp} pertencentes ao vetrores μ_{PP} e σ_{PP}, respectivamente, são calculados com base nas amostras do conjunto de treinamento coletado para esta característica $\{s_1.pp, s_2.pp, ..., s_{10}.pp\}$ cujos valores coletados foram $\{156, 161, 138, 173, 185, 153, 170, 153, 144, 152\}$.

Passo 1: Média = 1585/10 = 158.5

Desvio Padrão = $\sqrt{1790.5/9}$ = 14.1

Passo 2: Limite mínimo: 158.5 – 3*14.1 = 116.2

Limite máximo: 158.5 + 3*14.1 = 200.8

Não há outliers, pois não há valores fora do intervalo [116.2, 200.8]

Novo Cálculo da Média (sem *outliers*): $\mu_{pp_1} = 158.5$

Novo Cálculo do Desvio Padrão (sem *outliers*): $\sigma_{pp_1} = 14.1$

3.1.8 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, baseado em um valor de limiar (*threshold*) T, em dois subconjuntos L_w^v e L_w^f. No subconjunto L_w^v estão as amostras que o classificador considerou verdadeiras, e no subconjunto L_w^f estão as amostras que o classificador considerou falsas. Esta partição é vista na figura 3.9.

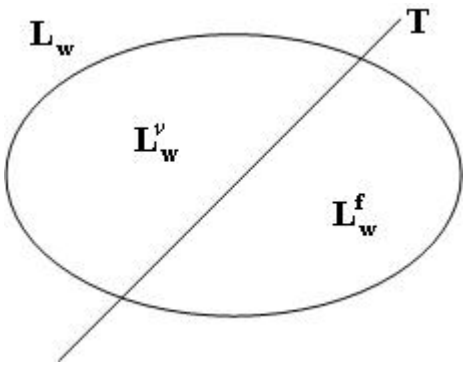


Figura 3.9: Partição do conjunto de autenticação

Dois tipos de classificadores são aplicados e analisados nesta dissertação: um classificador estatístico e um classificador nebuloso (*fuzzy*).

Em ambos os classificadores, inicialmente é feita uma decisão em relação ao vetor de características CT determinando se a autenticação continua ou não. Para uma amostra l do conjunto de autenticação L_w, se $l.ct_i = Z_w.ct_i$ para todo $i \leq n$, então a autenticação continua e o classificador analisa os outros vetores de características; caso contrário, a autenticação pára e a

amostra é considerada falsa. Os outros vetores de características são analisados por cada classificador como mostrado nas próximas subseções.

3.1.8.1 Classificador Nebuloso (*Fuzzy*)

Um classificador nebuloso foi escolhido devido à sua capacidade de lidar com dados ambíguos, analisando-os gradualmente, e não conforme a lógica tradicional (é ou não é) [10]. A intenção é que a partir das características extraídas da digitação, cada usuário seja categorizado pelo seu ritmo de digitação. Com este intuito, são utilizadas quatro variáveis lingüísticas: três de entrada (*tempo pp*, *tempo sp*, *tempo ps*) e uma de saída (*categorização*). Cada uma destas variáveis está associada aos seus termos lingüísticos que são:

- *tempo pp*: muito curto (MC), curto(C), curto médio (CM), curto longo (CL);

- *tempo sp*: realmente curto (RC), muito curto (MC), curto(C), médio curto (MEC), não-curto (NC);

- *tempo ps*: muito curto (MC), curto(C), curto médio (CM), curto longo (CL);

- *categorização*: muito baixa (MB), baixa (B), média (M), alta (A), muito alta (MA).

As características dos vetores de características PP, PS, SP são as entradas das funções de pertinências relacionadas, *pp*, *sp* e *ps*, respectivamente. No caso do vetor de característica M, os elementos $\{m_1, m_2, m_3, m_4\}$ são relacionadas a *pp*, $\{m_5, m_6, m_7, m_8\}$ a *sp*, e $\{m_9, m_{10}, m_{11}, m_{12}\}$ a *ps*.

As funções de pertinências utilizadas podem ser observadas nas figuras 3.10, 3.11, 312 e 3.13. Nestas figuras, $A(x)$ representa o grau de pertinência de x para um deterrminado conjunto nebuloso. A determinação destas funções foi feita através de experimentos feitos sobre as amostras coletadas. Com relação ao formato das funções, para as variáveis tempo foram determinadas gaussianas, pois foi observado que seus valores possuíam as características inerentes a uma distribuição normal, enquanto para a variável categorização foram determinadas funções triangulares, que além de serem tradicionais na teoria dos conjuntos nebulosos, possuem bastante

significância sem muita complexidade matemática [34]. Com relação ao domínio das funções, foi observado que, para a variável *tempo pp*, 99% das características extraídas estavam no intervalo [0,900], para a variável *tempo sp*, 98% estavam no intervalo [-200,750], e para a variável *tempo ps*, 99% estavam no intervalo [0,500].

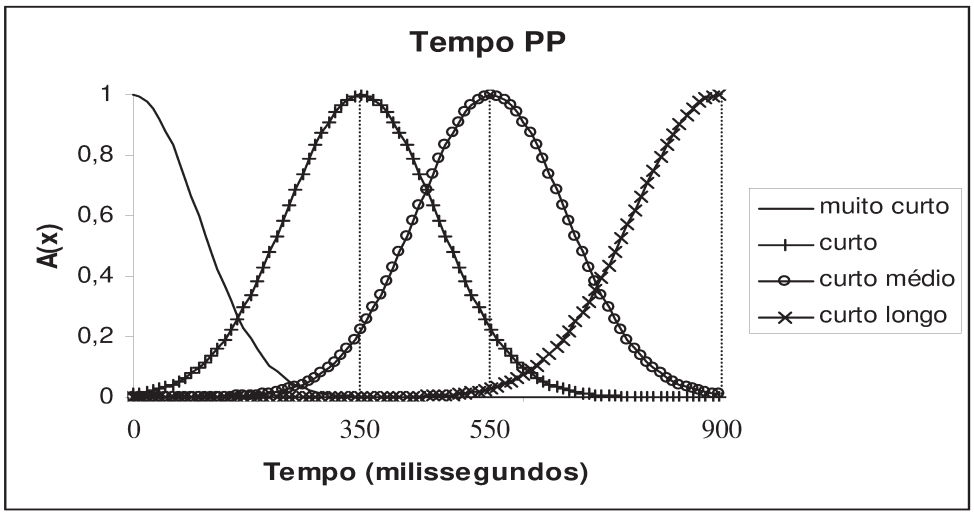


Figura 3.10: Funções de pertinência para a variável *tempo pp*

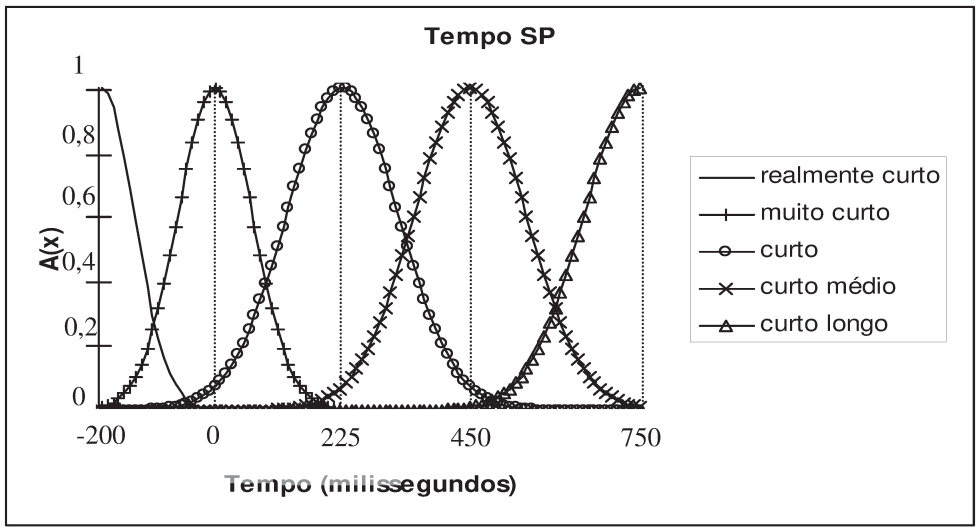


Figura 3.11: Funções de pertinência para a variável *tempo sp*

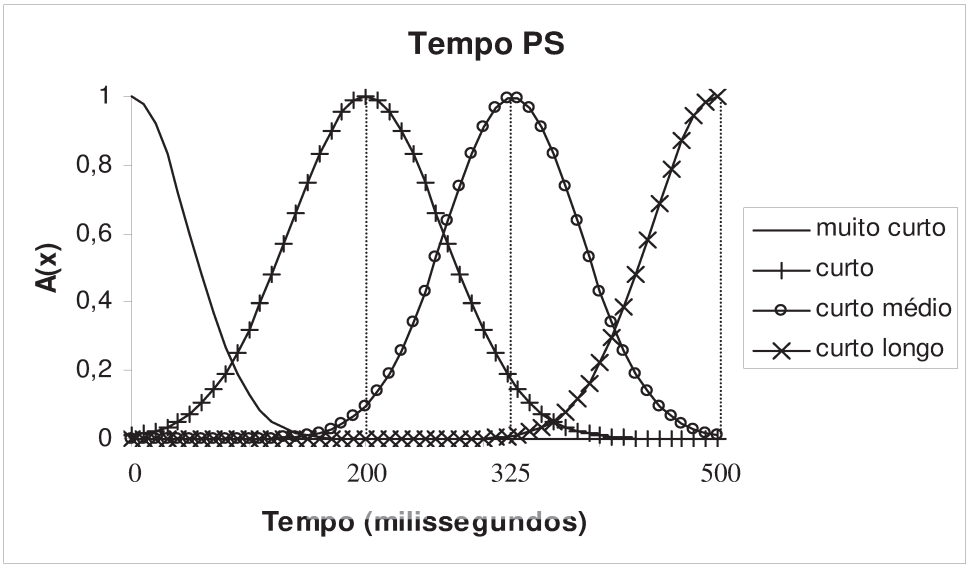


Figura 3.12: Funções de pertinência para a variável *tempo ps*

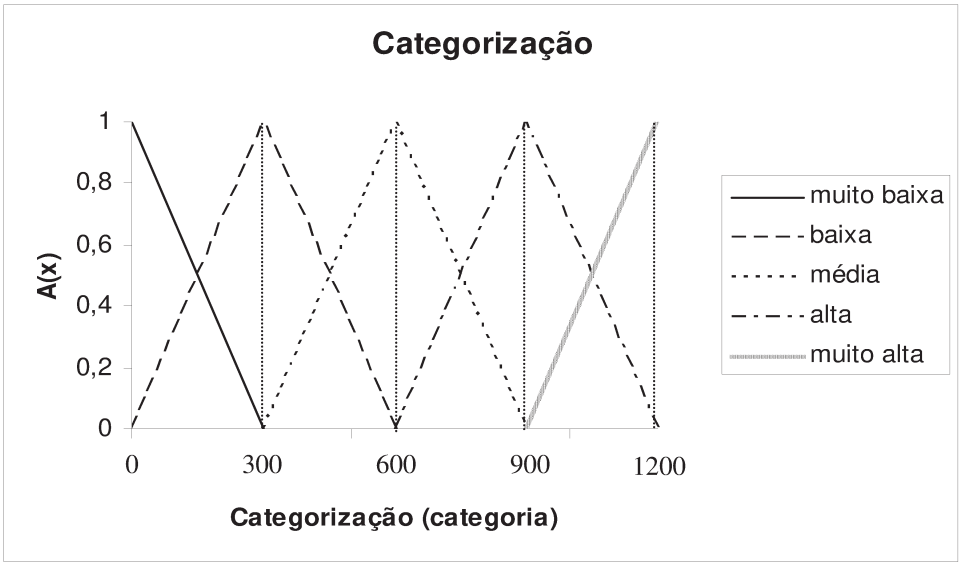


Figura 3.13: Funções de pertinência para a variável *categorização*

As regras nebulosas representam o conhecimento, indicando como as variáveis se relacionam entre si. As regras criadas foram baseadas no seguinte conhecimento intuitivo: quanto menor o tempo de digitação, maior é a categorização do usuário. Ou seja, um usuário experiente que digita mais rápido possuirá uma categorização mais alta, enquanto um usuário inexperiente possuirá uma categorização mais baixa. As regras criadas relacionando as variáveis de entrada com a variável de saída são relacionadas abaixo:

- Se *tempo pp* é muito curto, então *categorização* é alta;

- Se *tempo pp* é curto, então *categorização* é média;

- Se *tempo pp* é curto médio, então *categorização* é baixa;

- Se *tempo pp* é curto longo, então *categorização* é muito baixa;

- Se *tempo sp* é realmente curto, então *categorização* é muito alta;

- Se *tempo sp* é muito curto, então *categorização* é alta;

- Se *tempo sp* é curto, então *categorização* é média;

- Se *tempo sp* é curto médio, então *categorização* é baixa;

- Se *tempo sp* é curto longo, então *categorização* é muito baixa;

- Se *tempo ps* é muito curto, então *categorização* é alta;

- Se *tempo ps* é curto, então *categorização* é média;

- Se *tempo ps* é curto médio, então *categorização* é baixa;

- Se *tempo ps* é curto longo, então *categorização* é muito baixa.

Para cada característica presente nos vetores de características, as regras são aplicadas e algumas delas são acionadas de acordo com os graus gerados pelas variáveis de entrada nas funções de pertinência. Estes graus são chamados de graus de pertinência, e são utilizados como entradas nas funções de pertinência de categorização. Para cada regra acionada, um triângulo é truncado no grau relacionado. Caso mais de uma regra seja acionada, mais de um triângulo será

truncado, formando uma figura geométrica. O valor representativo desta figura é calculado utilizando o centro de massa. Este valor é a categorização do usuário para a característica analisada.

Exemplo 3.11: Sendo a característica pp_1 extraída de uma amostra l, $l.pp_1 = 156$, então a categorização resultante é:

- Se tempo pp é muito curto então categorização é alta. Resultando numa categorização alta com um grau de pertinência de 0,24 (figura 3.14).

- Se tempo pp é curto então categorização é média. Resultando numa categorização média com um grau de pertinência de 0,29 (figura 3.14).

Então, aplicando os graus gerados nas funções de pertinência de categorização, a figura formada pode ser vista na figura 3.15, junto com o valor calculado utilizando o centro de massa, que é a categorização.

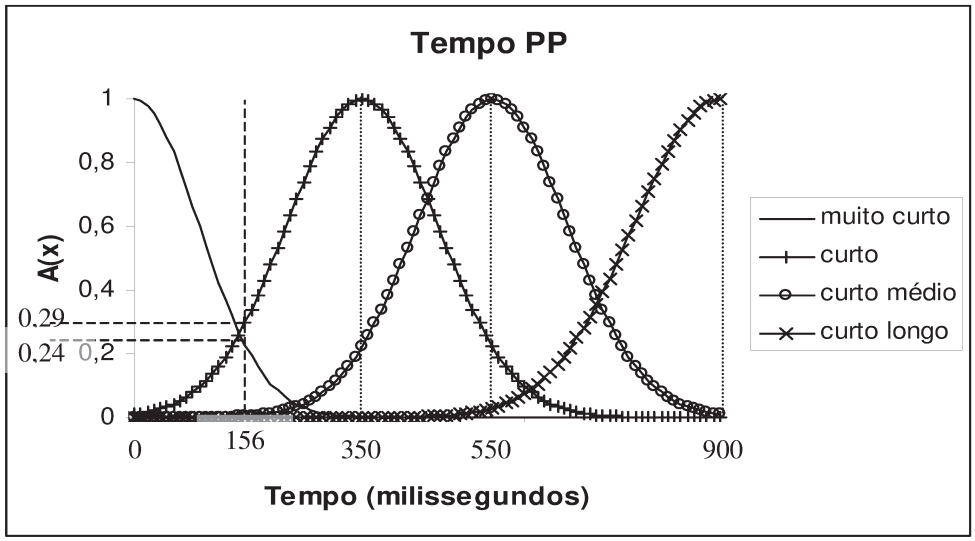


Figura 3.14: Ativação das regras com seus respectivos graus de pertinência

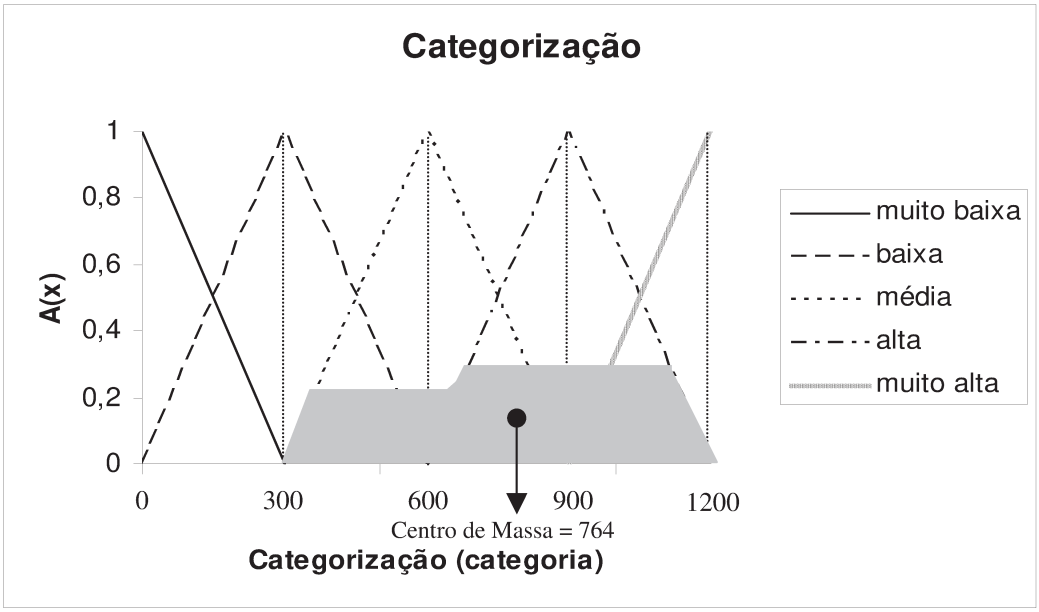


Figura 3.15: Categorização obtida pelos graus de pertinência

Uma amostra de autenticação só será considerada verdadeira se todas as categorizações geradas pelas suas características forem similares às categorizações geradas pelas médias contidas no *template*. O grau de similaridade $\varphi(x)$ de uma característica x é dado pela equação 3.11.

$$\varphi(x) = e^{-\frac{(cat(x)-cat(\mu))^2}{2*\sigma_{cat}^2}} \tag{3.11}$$

onde $cat(x)$ e $cat(\mu)$ são, respectivamente, a categorização da característica x e a categorização da média da característica x, e σ_{cat} é o desvio padrão da categorização. Este desvio é dado em função do desvio padrão da característica x, como foi observado em análise feita sobre as amostras coletadas. Nas figuras 3.16, 3.17 e 3.18, sendo referentes às características *pp*, *sp* e *ps,* respectivamente, podem ser observadas a determinação de σ_{cat}, junto com 200 pontos, representando amostras reais provindas de dez contas diferentes, sendo 100 amostras verdadeiras e 100 amostras falsas.

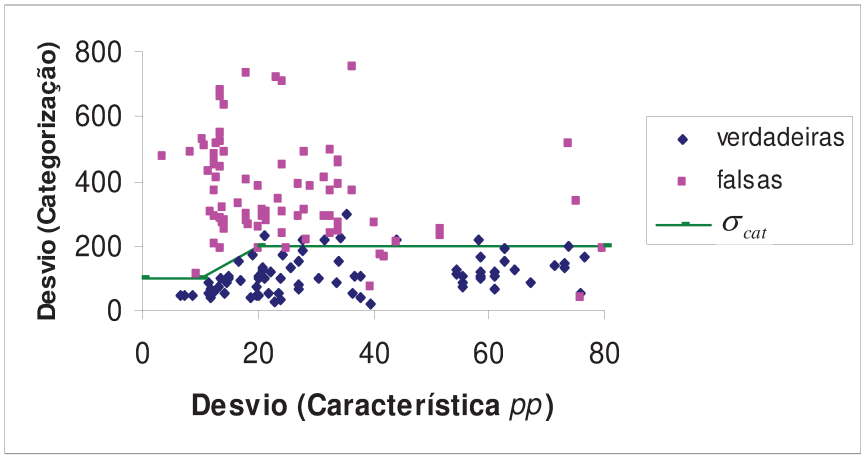


Figura 3.16: Função para determinação do desvio da categorização para característica *pp*

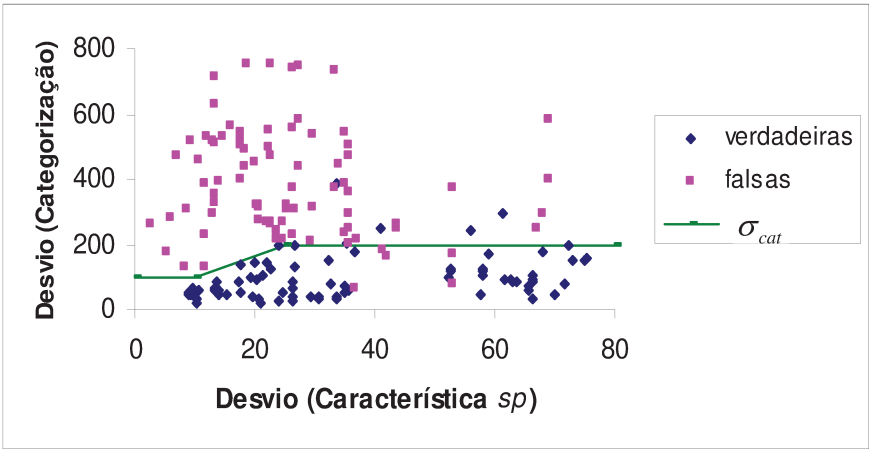


Figura 3.17: Função para determinação do desvio da categorização para característica *sp*

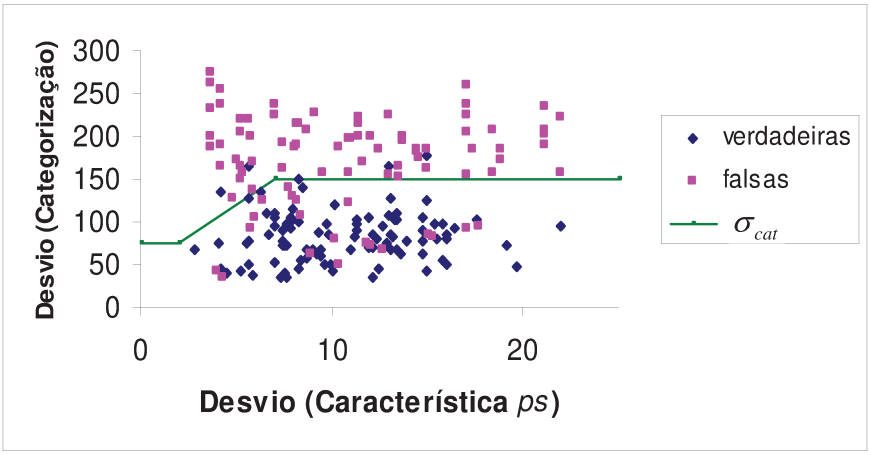


Figura 3.18: Função para determinação do desvio da categorização para característica *ps*

3.1.8.2 Classificador Estatístico

Um classificador de distância padrão foi utilizado baseado na normalização da equação 1.12. A distância entre um vetor de característica X e o template Z é dividida por d de acordo com a equação 3.12. Este procedimento é utilizado para permitir que as informações alvo escolhidas por cada usuário tenham quantidades de caracteres variáveis.

$$p_{X,Z} = \frac{1}{d}\sum_{i=1}^{d}\left|\frac{x_i - \mu_{x_i}}{\sigma_{x_i}}\right| \tag{3.12}$$

Exemplo 3.12: A distância resultante entre o template Z_w e a amostra l para a característica PP, sendo:

$\mu_{PP} = \{158.5,\ 151.6, 28.8, 211.5, 180.2, 296, 70.5, 194.9, 162.2, 192, 94.8, 67, 222.2, 125\}$,

$\sigma_{PP} = \{14.1, 6.16, 9.64, 10.3, 7.74, 17.32, 8.12, 18.05, 9.69, 7.21, 36.49, 29.01, 14.38, 7.14\}$ e

$l.PP = \{156, 144, 23, 201, 176, 251, 63, 191, 176, 218, 134, 54, 238, 120\}$, é:

$$p_{PP,Z_w} = \left(\left|\frac{158.5 - 156}{14.1}\right| + \left|\frac{151.6 - 144}{6.16}\right| + \ldots + \left|\frac{120 - 125}{7.14}\right|\right)/14 = \ 15.65/14 = 1.12$$

Uma amostra de autenticação só será considerada verdadeira, se as distâncias calculadas em seus vetores de características estiverem dentro de um valor de limiar (*threshold*) T adotado.

A determinação do valor de limiar influencia no desempenho de um sistema biométrico, pois quanto maior o seu valor mais amostras impostoras são erroneamente aceitas, aumentando a FAR, e, quanto menor o seu valor mais amostras genuínas são consideradas falsas, aumentando a FRR. O objetivo é encontrar o valor de limiar ideal onde estas duas taxas sejam as menores possíveis.

Por causa da importância deste assunto, uma análise foi feita em dados coletados para uma obtenção automática do valor de limiar que esteja o mais próximo possível do ideal. Com base nesta análise, foi observado que, quanto maior era o desvio padrão das características, menor era o valor de limiar ideal, e que o contrário também é verdadeiro. Por causa deste fato, a determinação do valor de limiar T é feita em função do desvio padrão σ das características de acordo com as equações 3.13, 3.14, 3.15 e 3.16 referentes a T_{PP}, T_{SP}, T_{PS} e T_M, respectivamente. Os dados que geraram as funções podem ser observados junto com as respectivas funções nas figuras 3.19, 3.20, 3.21 e 3.22, sendo referentes aos vetores de características *PP*, *SP*, *PS* e *M*, respectivamente. Em todas as figuras, foram introduzidos 200 pontos, representando amostras reais provindas de dez contas diferentes, sendo 100 amostras verdadeiras e 100 amostras falsas. As distâncias calculadas maiores que 10 foram truncadas para 10, por questões de visualização.

$$T_{PP} = \begin{cases} 2.5 & , \quad \sigma \le 17 \\ (74.5 - \sigma)/23 & , \quad 17 < \sigma < 40 \\ 1.5 & , \quad \sigma \ge 40 \end{cases} \tag{3.13}$$

$$T_{SP} = \begin{cases} 2.5 & , \quad \sigma \le 13 \\ (55.5 - \sigma)/17 & , \quad 13 < \sigma < 30 \\ 1.5 & , \quad \sigma \ge 30 \end{cases} \tag{3.14}$$

$$T_{PS} = \begin{cases} 2.5 & , \quad \sigma \le 5 \\ (30 - \sigma)/10 & , \quad 5 < \sigma < 15 \\ 1.5 & , \quad \sigma \ge 15 \end{cases} \tag{3.15}$$

$$T_M = \begin{cases} 2.5 & , \quad \sigma \le 5 \\ (105 - \sigma)/40 & , \quad 5 < \sigma < 45 \\ 1.5 & , \quad \sigma \ge 45 \end{cases} \tag{3.16}$$

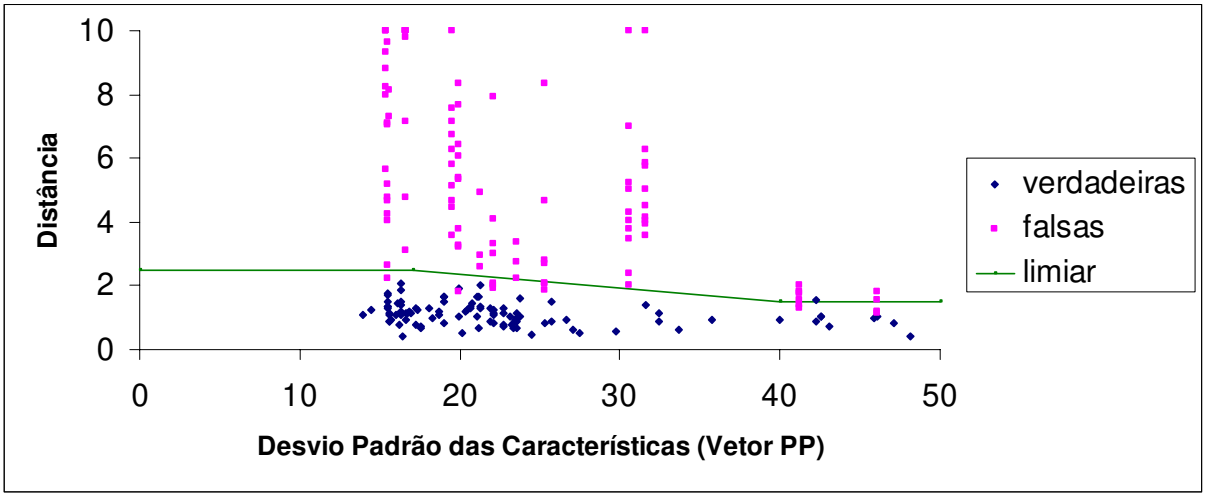


Figura 3.19: Função para determinação do valor de limiar do vetor de características *PP*

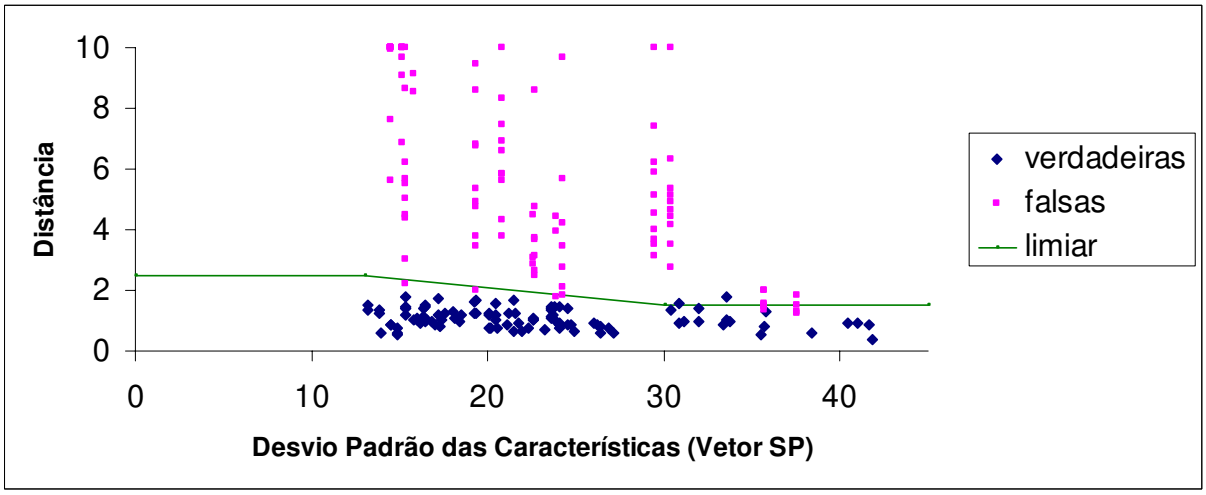


Figura 3.20: Função para determinação do valor de limiar do vetor de características *SP*

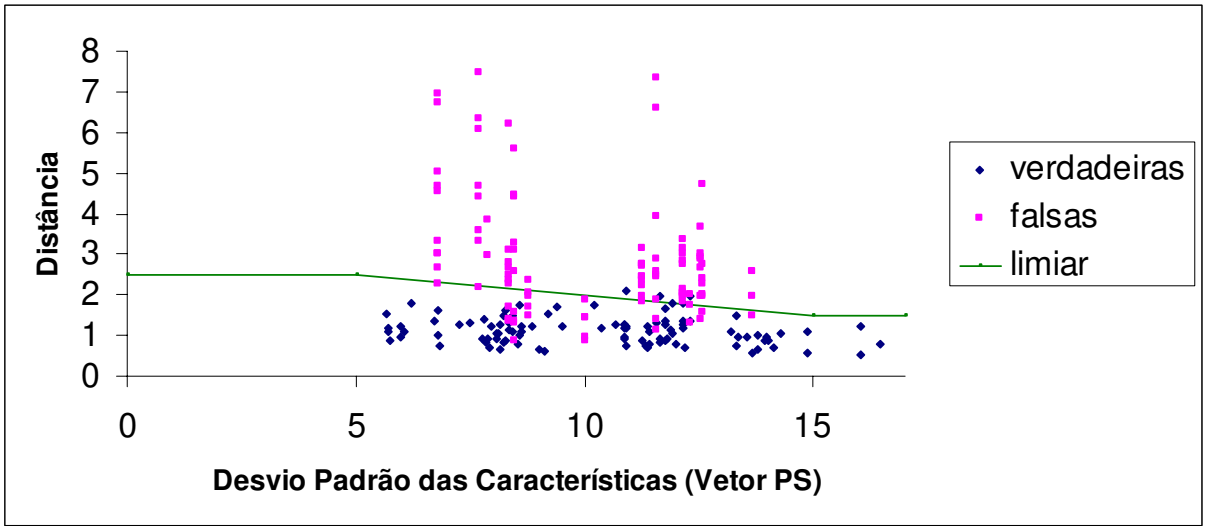


Figura 3.21: Função para determinação do valor de limiar do vetor de características *PS*

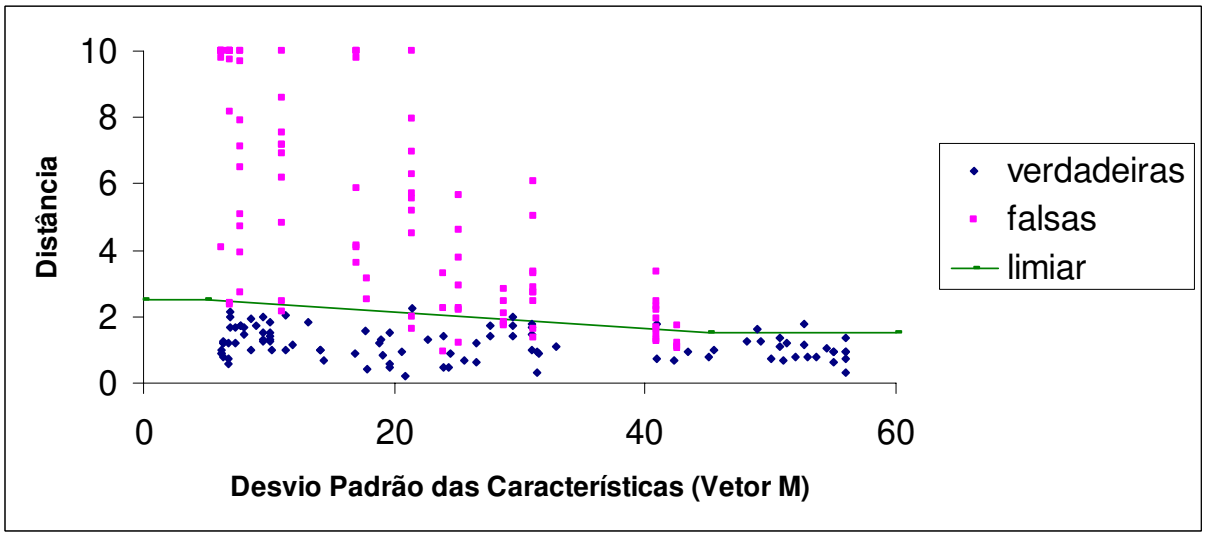


Figura 3.22: Função para determinação do valor de limiar do vetor de características M

Como observado nas figuras anteriores, o valor de limiar determinado em função do desvio padrão das características em cada um dos vetores de características proporciona uma partição melhor entre as classes de amostras verdadeiras e falsas do que se utilizássemos um único valor de limiar para todos eles. Desta maneira, são utilizados quatro valores de limiar diferentes, um para cada vetor de características, pois cada um deles possui um desvio padrão.

3.1.9 Tentativas

Como observado em [26], os usuários legítimos falhavam na primeira tentativa de autenticação, mas acertavam na segunda. Logo, nesta dissertação, foram concedidas duas tentativas de autenticação para cada usuário. Para melhor compreensão, chamamos de sessão de autenticação todo o processo de autenticação que pode ser composto por uma tentativa com sucesso, ou por uma tentativa com fracasso e uma com sucesso, ou ainda, por duas tentativas com fracasso.

3.1.10 Atualização

Os *templates* devem sempre representar os usuários proprietários das contas. Porém, como a digitação tende a se modificar gradualmente com as autenticações, uma atualização nos *templates* armazenados deve ser realizada.

Um mecanismo de adaptação é adotado nesta dissertação o qual é baseado em [24]. 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 e a maioria das características extraídas da amostra está dentro do valor de limiar determinado. De todas as pesquisas consultadas e referenciadas nesta dissertação baseadas em dinâmica da digitação, somente em [24] podemos observar a utilização de um mecanismo de adaptação. As características extraídas da dinâmica da digitação sofrem influências temporais, e um mecanismo de adaptação torna-se uma obrigatoriedade para a manutenção dos *templates* atualizados e do desempenho afirmativo na autenticação de usuários com o decorrer do tempo.

Na figura 3.23 pode ser visualizado um exemplo da evolução da média do vetor de características PP relacionado a uma conta à medida que o mecanismo de adaptação é ativado.

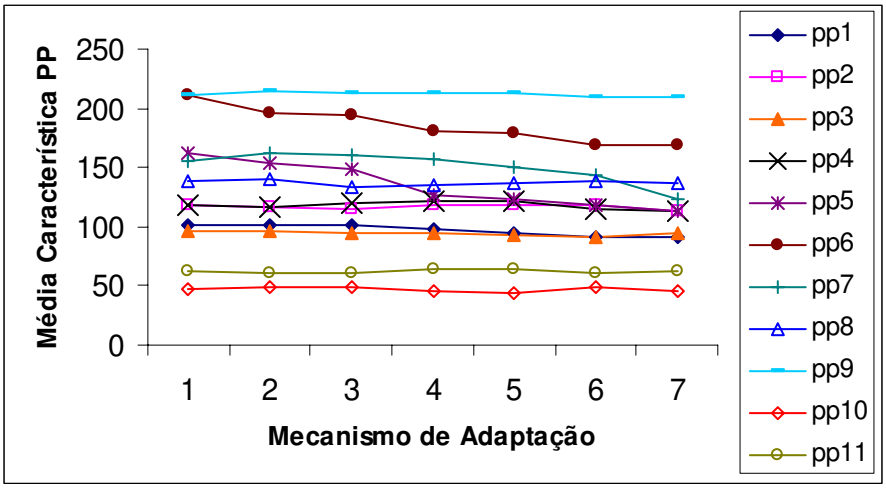


Figura 3.23: Exemplo da evolução da média com o mecanismo de adaptação no vetor de características *PP*.

Como observado na figura 3.23, algumas características (pp_5, pp_6, pp_7) se modificam bastante no decorrer da aplicação do mecanismo de adaptação, enquanto as outras permanecem mais ou menos estáveis.

3.2 Ferramenta

Uma ferramenta foi desenvolvida para automatizar a metodologia apresentada. Esta ferramenta foi implementada utilizando a linguagem de programação java, na plataforma Java 2, desenvolvida pela Sun Microsystems. Dois módulos foram implementados nesta ferramenta: um referente ao cadastramento e o outro referente à autenticação. Fizemos uso também do *Java Native Interface* (JNI) para poder acessar as instruções assembly RTDSC e CPUID implementadas em funções na linguagem de programação C. Foi utilizado também o banco de dados MySQL para o armazenamento das informações coletadas nos módulos referentes aos usuários do sistema.

3.2.1 Módulo de Cadastramento

Este módulo implementa o processo de cadastramento, conforme explicado na seção 3.1. A figura 3.24 mostra a tela deste módulo, e os números nela mostrados são explicados a seguir:

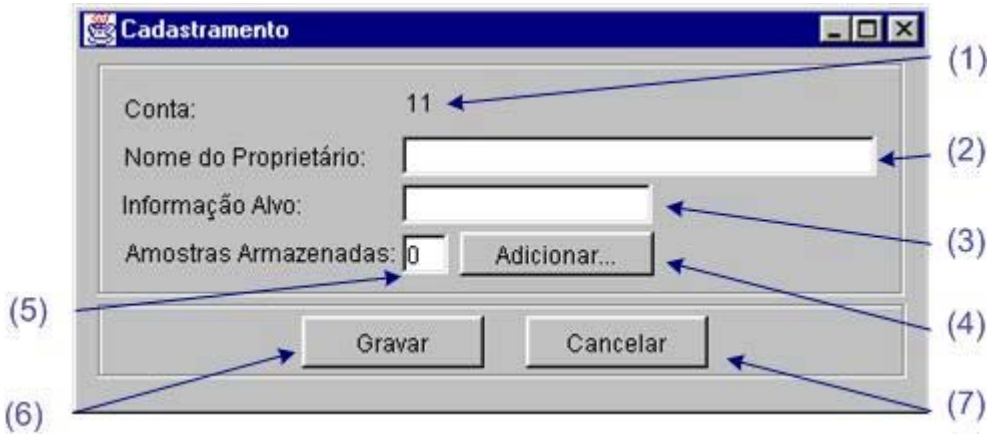


Figura 3.24: Tela do Módulo de Cadastramento

(1) A conta na qual o usuário será cadastrado. Este campo é auto-incrementado, não-editável e único para cada usuário.

(2) O nome do usuário proprietário da conta.

(3) A informação alvo escolhida pelo usuário. Este campo deve conter pelo menos onze caracteres alfa-numéricos.

(4) O botão Adicionar aciona uma tela (figura 3.25). Nesta tela, uma amostra é adicionada ao conjunto de treinamento e a quantidade de amostras coletadas é incrementada.

(5) A quantidade de amostras coletadas no momento atual.

(6) O botão Gravar gera o template com base nas amostras coletadas e grava na base de dados o nome do proprietário, a informação alvo, as amostras e o template relacionando-os com a conta. Este botão só é habilitado quando a quantidade de amostras coletadas é igual a dez.

(7) O botão Cancelar cancela todas as informações preenchidas.

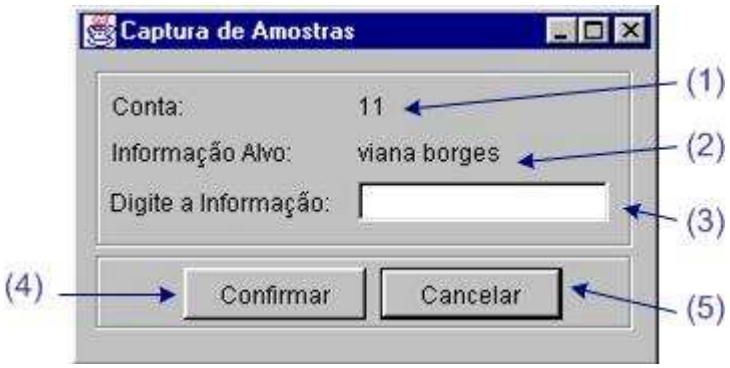


Figura 3.25: Tela de Captura de Amostra

Com relação à tela mostrada na figura 3.25, os números nela mostrados são explicados a seguir:

(1) O número da conta que está sendo cadastrada.

(2) A informação alvo que foi escolhida na tela anterior e que deve ser digitada em (3).

(3) O usuário digita a informação alvo mostrada em (2). Esta digitação é monitorada e os dados da digitação são capturados. Caso algum erro tipográfico seja cometido, o próprio programa apaga a informação digitada, fazendo com que o usuário digite a informação alvo novamente.

(4) O botão Confirmar extrai as características dos dados de digitação capturados, formando uma amostra. O vetor de características CT desta amostra é analisado, pois este vetor deve ser sempre o mesmo para todas as amostras que irão compor o conjunto de treinamento. Caso este vetor de características seja diferente do vetor extraído da primeira amostra coletada, então uma nova digitação é requerida. Este botão só é habilitado quando a informação alvo for digitada corretamente.

(5) O botão Cancelar cancela todas as informações preenchidas.

3.2.2 Módulo de Autenticação

Este módulo implementa o processo de autenticação que foi explicado na seção 3.1. A figura 3.26 mostra a tela deste módulo e os números nela mostrados são explicados a seguir:

Figura 3.26: Tela do Módulo de Autenticação

(1) O usuário indica a conta que deseja utilizar.

(2) O botão OK busca na base de dados, a informação alvo escolhida para a conta identificada em (1).

(3) A informação alvo é mostrada após o pressionamento de (2) para ser digitada pelo usuário em (4). Não há sigilo na informação alvo, pois não é o escopo deste trabalho.

(4) O usuário digita a informação alvo mostrada em (3). Esta digitação é monitorada e os dados da digitação são capturados. Caso algum erro tipográfico seja cometido, o próprio programa apaga a informação digitada, fazendo com que o usuário digite a informação alvo novamente.

(5) Caso o usuário seja impostor, esta caixa de seleção deve ser confirmada, para que após a autenticação possamos saber se ocorreu ou não um erro de classificação.

(6) O botão Autenticar extrai as características dos dados de digitação capturados formando uma amostra. Esta amostra é enviada para o classificador que decide pela sua veracidade. Caso a amostra seja considerada verdadeira então o usuário é considerado legítimo. Caso a amostra seja considerada falsa, uma nova tentativa é dada para o usuário. Se nesta nova tentativa, a nova amostra seja considerada novamente falsa, então o usuário é considerado impostor. Quando a amostra é considerada verdadeira, o mecanismo de adaptação pode ser executado. Este botão só é habilitado quando a informação alvo for digitada corretamente.

(7) O botão Cancelar cancela todas as informações preenchidas.

CAPÍTULO 4

RESULTADOS

Neste capítulo fazemos uma apresentação dos resultados obtidos na metodologia mostrada no capítulo 3. Inicialmente, explicamos o processo de aquisição de amostras de dinâmica da digitação. Depois, são apresentados os experimentos realizados com seus respectivos resultados na base de dados coletada. Finalmente, são discutidos os resultados obtidos nos experimentos realizados.

4.1 Aquisição de Amostras

A definição de uma base de dados que seja representativa no universo de dinâmica da digitação que podem ser encontradas em um problema real de reconhecimento é um requisito fundamental para a avaliação experimental da metodologia proposta nesta dissertação. Além disso, a construção desta base assume uma importância ampla na medida em que está servindo e servirá como referência para muitos outros trabalhos relacionados à área de verificação pessoal via dinâmica da digitação.

Em muitos problemas de estatística, técnicas de amostragem são utilizadas para extrair um subconjunto de elementos de uma população que seja representativo, no sentido de que 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 fazerem 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.

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 difícil 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, etc.

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, se faz necessário fornecer respostas a algumas perguntas importantes:

(1) Que tipos de objetos deverão constituir a base de dados?

(2) Qual deverá ser a quantidade de classes (contas)?

(3) Qual deverá ser 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. A fim de permitir a avaliação da metodologia proposta, além de digitações provindas de usuários legítimos, deverão ser incluídas digitações provindas de usuários impostores.

Respondendo à pergunta 2, a definição da quantidade ideal de classes a serem consideradas em uma amostragem, mesmo em estatística, não segue uma regra muito precisa. Neste contexto, costuma-se citar conjunto de amostras pequeno ou conjunto de amostras grande.

Em problemas de classificação envolvendo uma população infinita, uma amostragem contendo menos de 50 classes é considerada pequena [35]. Caso contrário, o tamanho da amostragem é dito ser significativo a um certo nível que pode ser calculado. Dentro do escopo deste trabalho e do tempo disponível, escolheu-se 50 como o número total de classes constituindo a base de dados. As classes são representadas pelas contas em que os usuários estão cadastrados.

A questão 3 trata da quantidade de padrões ou amostras por classe. Como já explicado no capítulo 3, as amostras formam dois conjuntos distintos: o conjunto de treinamento e o conjunto de autenticação. O conjunto de treinamento é composto por 500 amostras, com 10 amostras para cada classe (conta), pela identificação de um limite prático analisando a maioria das pesquisas referenciadas no capítulo 2, além da nossa própria experiência na aquisição das amostras para composição da base de dados utilizada neste trabalho, conforme o item 3.1.6. Nas figuras 4.1, 4.2, 4.3 e 4.4 podem ser visualizados exemplos de conjuntos de treinamentos para os vetores de características PP, SP, PS e M, respectivamente, capturados de um usuário. As figuras mostram os elementos componentes de cada um dos seus respectivos vetores interpretados graficamente.

O conjunto de autenticação é formado por um total de 5000 sessões de autenticações. Como explicado no capítulo 3, cada sessão pode ter uma ou duas amostras relacionadas, dependendo se a primeira amostra foi considerada verdadeira ou falsa. As amostras que compõem o conjunto de autenticação podem ser amostras verdadeiras ou amostras falsas, sendo as falsas provindas de dois tipos de usuários impostores. O primeiro tipo de usuário impostor é o simples, que somente possui o conhecimento da informação alvo que deve ser digitada. O segundo tipo de usuário impostor é o observador, que verifica a maneira como o usuário legítimo digita a informação alvo, ou seja, ele testemunha sessões de autenticação do proprietário da conta. Exemplos de amostras falsas de uma mesma conta são mostrados nas figuras 4.5, 4.6, 4.7 e 4.8, cada uma referente a um vetor de características PP, SP, PS e M, respectivamente. Para uma verificação das diferenças que existem entre uma amostra falsa e uma amostra verdadeira é mostrada também nas figuras 4.5, 4.6, 4.7 e 4.8 a média contida no *template* da conta em questão. Em um período de doze semanas, os usuários tentavam se autenticar nas contas cadastradas utilizando dois tipos diferentes de *layouts* de teclados convencionais e um teclado de *notebook*, compondo a base de dados com:

- 900 sessões provenientes de usuários legítimos, sendo entre 10 e 20 sessões para cada conta cadastrada;

- 3500 sessões provenientes de usuários impostores simples, sendo entre 30 e 100 sessões para cada conta cadastrada.

- 600 sessões provenientes de usuários impostores observadores, sendo entre 10 e 15 sessões para cada conta cadastrada.

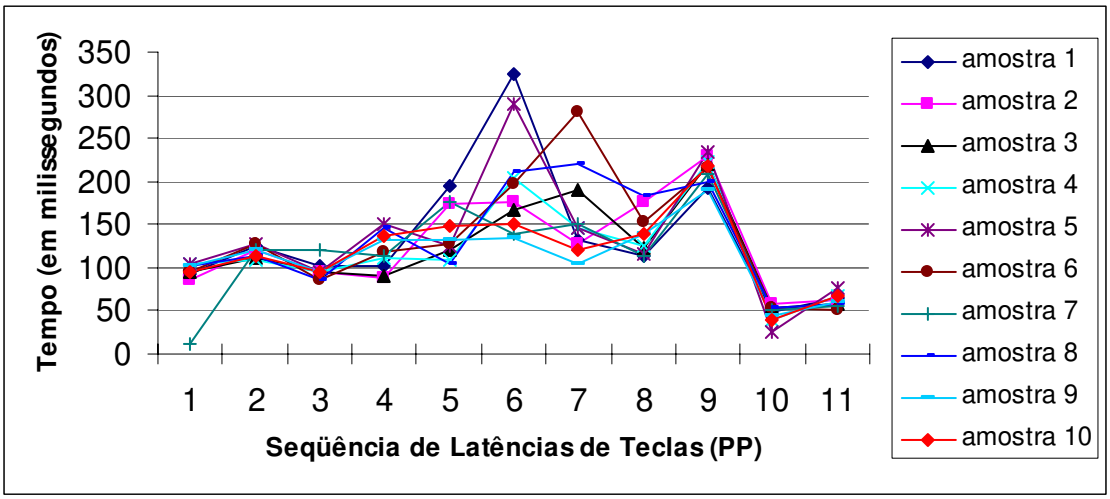


Figura 4.1: Exemplo de conjunto de treinamento (vetor de características *PP*)

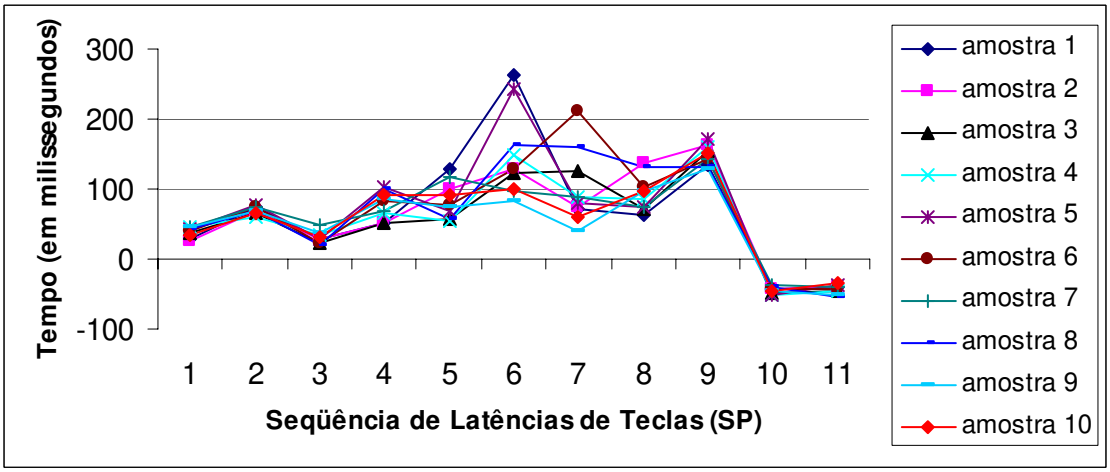


Figura 4.2: Exemplo de conjunto de treinamento (vetor de características *SP*)

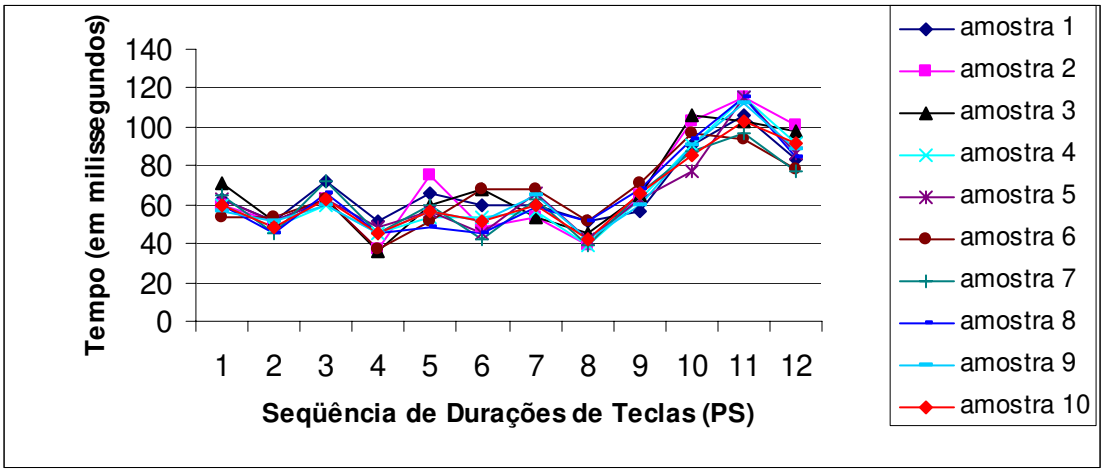


Figura 4.3: Exemplo de conjunto de treinamento (vetor de características *PS*)

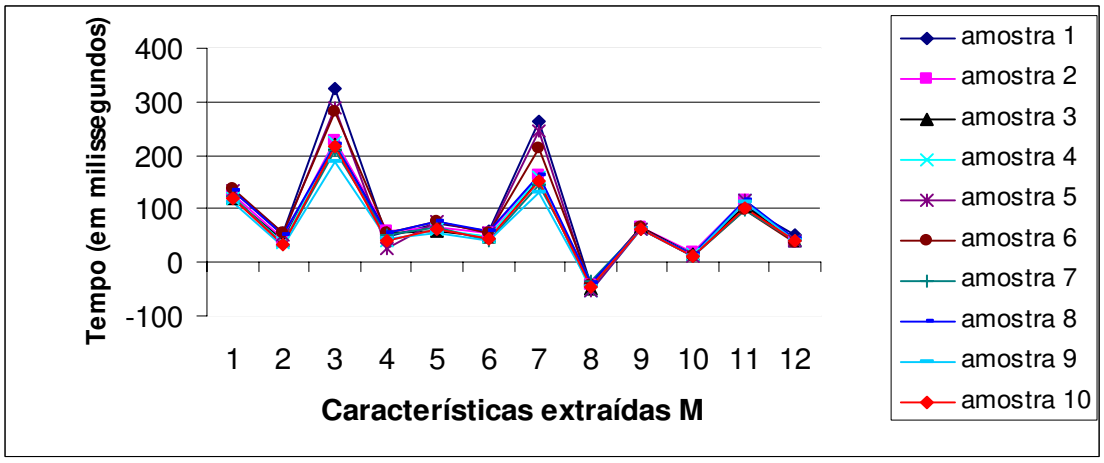


Figura 4.4: Exemplo de conjunto de treinamento (vetor de características *M*)

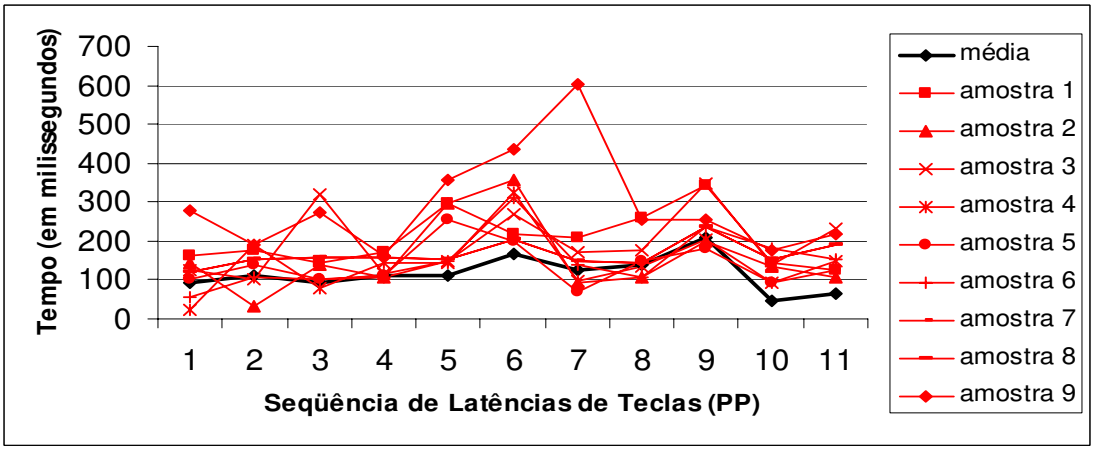


Figura 4.5: Exemplo de amostras falsas e da média contida no *template* relacionadas a uma mesma conta (vetor de características *PP*)

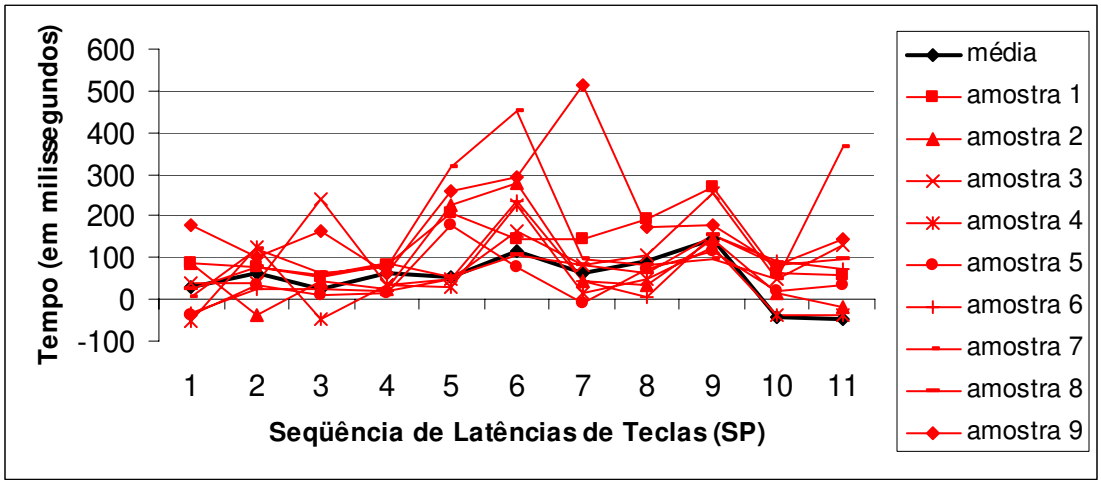


Figura 4.6: Exemplo de amostras falsas e da média contida no *template* relacionadas a uma mesma conta (vetor de características *SP*)

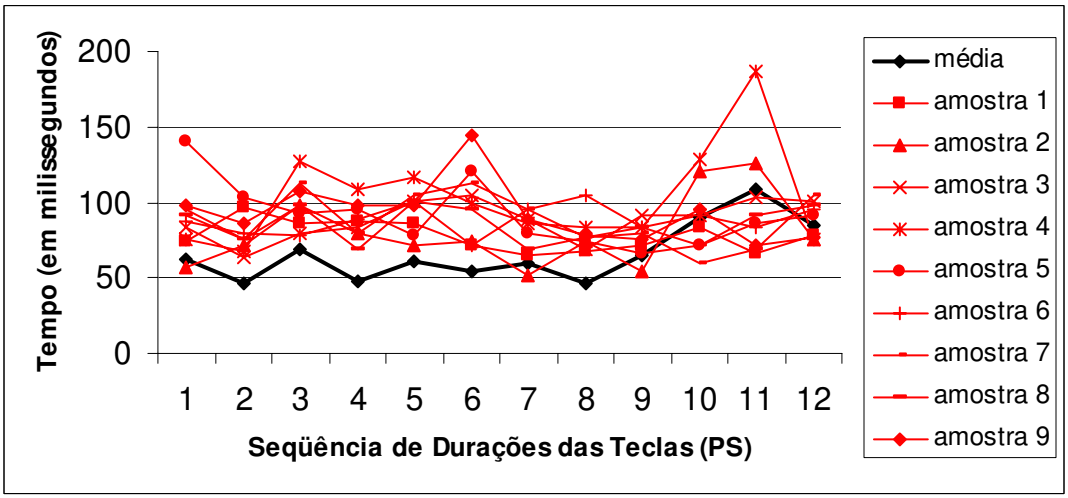


Figura 4.7: Exemplo de amostras falsas e da média contida no *template* relacionadas a uma mesma conta (vetor de características *PS*)

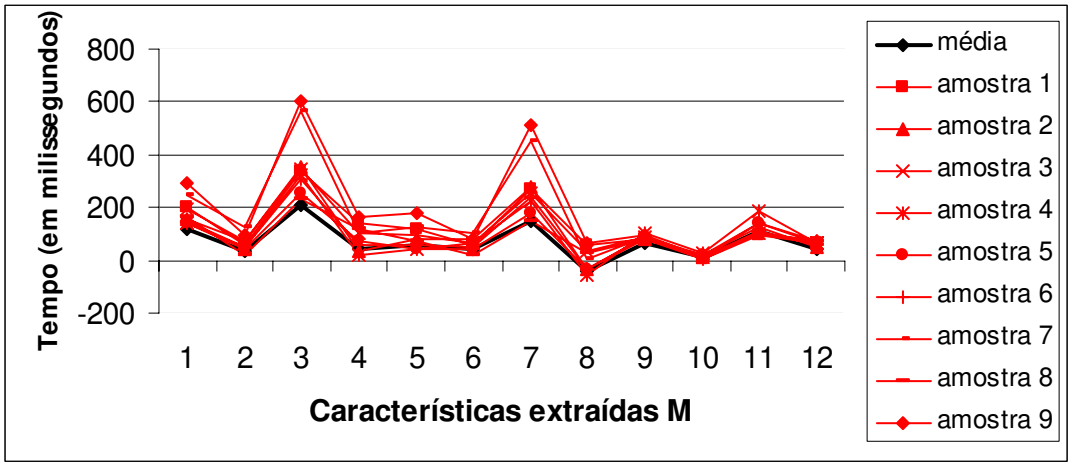


Figura 4.8: Exemplo de amostras falsas e da média contida no *template* relacionadas a uma mesma conta (vetor de características *M*)

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

4.2 Experimentos

Para analisar o desempenho da metodologia apresentada em verificar corretamente a identidade dos usuários proprietários das contas, experimentos foram realizados na base de dados. 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.

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 FRR e FAR, que foram discutidas no capítulo 1 (seção 1.2). Estas taxas são calculadas pelas equações 4.1 e 4.2.

$$FRR = \frac{\text{quantidade de sessões enquadradas na situação 3}}{\text{quantidade total de sessões}} \tag{4.1}$$

$$FAR = \frac{\text{quantidade de sessões enquadradas na situação 4}}{\text{quantidade total de sessões}} \tag{4.2}$$

Inicialmente, foram analisadas combinações de vetores de características e classificador. A primeira combinação é com relação ao vetor de características CT, que é independente do classificador aplicado. Os seus resultados são bastante discriminantes quando a digitação da informação alvo pode resultar em mais de um conjunto de teclas possível. Na base de dados coletada, quando a informação alvo estava nesta situação, aproximadamente 70% das sessões provenientes de usuários impostores foram rejeitadas utilizando apenas este vetor.

Nas tabelas 4.1, 4.2 e 4.3, são apresentados os resultados obtidos em outras combinações de vetores de características:

(I) CT e PP ;

(II) CT e SP ;

(III) CT e PS ;

(IV) CT e M ;

(V) CT , PP e SP ;

(VI) CT , PP e SP ;

(VII) CT , SP e PS ;

(VIII) CT, PP, SP e PS;

(IX) CT, PP, SP, PS e M.

Na tabela 4.1 é mostrada a FRR obtida nas sessões provenientes de usuários legítimos. Na tabela 4.2 é mostrada a FAR obtida nas sessões provenientes de usuários impostores simples. Na tabela 4.3 é mostrada a FAR obtida nas sessões provenientes de usuários impostores observadores.

Tabela 4.1: FRR (%) obtidos em sessões de usuários legítimos pelas combinações de características e classificador

	(I)	**(II)**	**(III)**	**(IV)**	**(V)**	**(VI)**	**(VII)**	**(VIII)**	**(IX)**
nebuloso	3.89	4.44	0.78	6.44	5.11	4.22	4.11	3.44	6.67
estatístico	1.55	2.22	2.11	3.33	2.11	1.44	2.0	1.55	3.33

Tabela 4.2: FAR (%) obtidos em sessões de usuários impostores simples pelas combinações de características e classificador

	(I)	**(II)**	**(III)**	**(IV)**	**(V)**	**(VI)**	**(VII)**	**(VIII)**	**(IX)**
nebuloso	32.0	24.2	36.8	37.2	12.3	14.5	9.51	2.97	8.97
estatístico	17.7	7.11	28.2	25.5	6.03	5.51	3.2	1.91	3.17

Tabela 4.3: FAR (%) obtidos em sessões de usuários impostores observadores pelas combinações de características e classificador

	(I)	**(II)**	**(III)**	**(IV)**	**(V)**	**(VI)**	**(VII)**	**(VIII)**	**(IX)**
nebuloso	36.0	30.6	36.8	51.6	15.8	17.6	10.3	7.83	9.83
estatístico	24.0	11.3	28.8	44.2	9.33	9.83	6.33	3.5	6.0

A partir das tabelas acima, podemos observar que:

- A combinação que resulta nas melhores taxas é a utilização do classificador estatístico com os vetores de características PP, SP e PS no experimento (VIII): 1.55% FRR, 1.91% FAR (impostor simples) e 3.5% FAR (impostor observador).

- A segunda melhor combinação é a do classificador estatístico com os vetores de características SP e PS, no experimento (VII): 2.0% FRR, 3.2% FAR (impostor simples) e 6.33% FAR (impostor observador).

- Em todas combinações de vetores de características, o classificador nebuloso adotado obteve taxas maiores que o classificador estatístico utilizado.

- De acordo com os resultados obtidos com as combinações (II) e (IV), podemos observar que o vetor de características SP é o mais discriminante , enquanto que o vetor de características M é o menos discriminante respectivamente, quando aplicados isoladamente.

Nas próximas subseções são mostrados experimentos realizados em relação a alguns aspectos da metodologia para verificar os seus impactos nas taxas obtidas da combinação do classificador estatístico e dos vetores de características CT , PP, SP e PS.

4.2.1 Extração de Características em Função dos Caracteres

Na metodologia mostrada, as características são extraídas em função das teclas utilizadas na digitação da informação alvo, mas nas pesquisas estudadas no capítulo 2, esta extração é feita em função dos caracteres da informação alvo. Para verificar a importância da adoção desta abordagem, um experimento é realizado mudando a extração das características de "em função das teclas" para "em função dos caracteres". Desta maneira, os vetores de características CT, PP, SP, PS e M extraídos em função dos caracteres da informação alvo ia terão tamanhos iguais a h, $h-1$, $h-1$, h e 12, respectivamente, sendo h a quantidade de caracteres de ia. Por outro lado, se fizermos em função das teclas utilizadas na digitação de ia, os vetores de características CT, PP, SP, PS e M terão tamanhos iguais a n, $n-1$, $n-1$, n e 12, respectivamente, sendo n a quantidade de teclas utilizada na digitação de ia. Na tabela 4.4 é mostrada a FRR e a FAR obtidas neste experimento (II) (em função dos caracteres) comparado com as obtidas na metodologia (I) (em função das teclas).

Tabela 4.4: FRR e FAR obtidos no experimento de extração das características em função das teclas (I) e dos caracteres (II)

	Extração de Características	
	(I)	**(II)**
FRR (%)	1.55	2.11
FAR (%)	1.91	4.11

Como observado na tabela 4.4, as taxas obtidas com as características extraídas em função dos caracteres da informação alvo são maiores que as obtidas com as características extraídas em função das teclas, a FRR aumentou de 1.55% para 2.11%, e a FAR aumentou de 1.91% para 4.11%. Desta maneira, as características extraídas pela abordagem adotada neste trabalho são mais discriminantes.

4.2.2 Imposição da Informação Alvo

Na metodologia apresentada nesta dissertação, cada usuário pode escolher livremente a informação alvo a ser digitada, mas em [25] e na identificação de usuários de [19], a informação alvo era imposta para todos os usuários cadastrados. Para verificar a importância da escolha da informação alvo, um experimento é realizado mudando a informação alvo escolhida para uma informação alvo imposta. A informação alvo imposta é "unicamp@2003", que para os usuários envolvidos é uma *string* simples. Assim, 30 usuários foram cadastrados com a informação alvo imposta, e, depois disso, foram coletadas 550 sessões de autenticação utilizando a informação alvo imposta. Na tabela 4.5 é mostrada a FRR e a FAR obtidas neste experimento (II) (com uma informação alvo imposta) comparado com as obtidas na metodologia (I) (com uma informação alvo escolhida).

Tabela 4.5: FRR e FAR obtidas no experimento de livre escolha (I) e de imposição (II) da informação alvo

	Informação Alvo	
	(I)	**(II)**
FRR (%)	1.55	18.36
FAR (%)	1.91	5.52

Como observado na tabela 4.5, a imposição da informação alvo obteve uma influência negativa nas taxas, a FRR aumentou de 1.55% para 18.36%, e a FAR aumentou de 1.91% para 5.52%. Assim, a escolha da informação deve ser feita pelo usuário, ao invés de ser imposta pelo administrador do sistema.

4.2.3 Quantidade de Caracteres

Em relação à quantidade de caracteres da informação alvo foi feita uma restrição: ela deve conter pelo menos 11 caracteres. Algumas pesquisas [21], [22], [23] e [26] utilizam informações alvos contendo menos que 10 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. A média dos caracteres contidos nas informações alvo coletadas é de 12.5 e a maior

informação alvo contém 17 caracteres. Na figura 4.9 é mostrado o comportamento da FRR e da FAR com a redução da quantidade de caracteres.

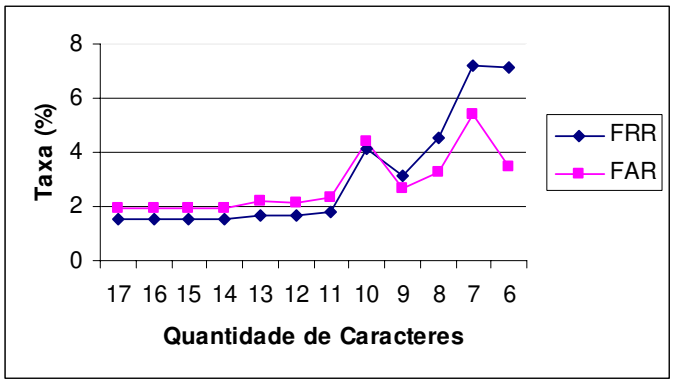


Figura 4.9: Comportamento da FRR e da FAR com a quantidade de caracteres

Como observado na figura 4.9, ambas as taxas aumentam com a redução da quantidade de caracteres, mas a partir de 10 caracteres este aumento é mais abrupto. Assim, informações alvo com menos que 11 caracteres não é recomendável para a discriminação de usuários a partir da dinâmica da digitação.

4.2.4 Quantidade de Amostras do Conjunto de Treinamento

A quantidade de amostras presentes no conjunto de treinamento é um aspecto crucial, pois à medida que esta quantidade é reduzida, os erros de classificação aumentam [19]. Nesta metodologia, a quantidade de amostras presentes no conjunto de treinamento é de 10 amostras. Para verificar o impacto da quantidade de amostras do conjunto de treinamento, um experimento é realizado reduzindo a quantidade destas amostras. Na figura 4.10 é mostrado o comportamento da FRR e da FAR com a redução da quantidade de amostras do conjunto de treinamento.

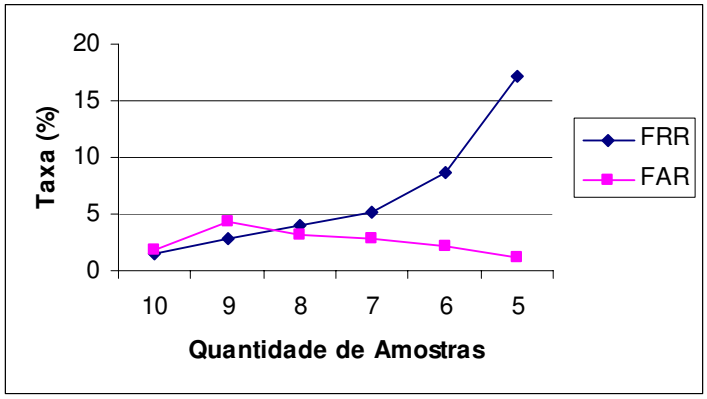


Figura 4.10: Comportamento da FRR e da FAR com a quantidade de amostras do conjunto de treinamento

Como observado na figura 4.10, a redução da quantidade de amostras do conjunto de treinamento influencia em ambas as taxas. À medida que a quantidade de amostras é reduzida, a discriminância das características também é reduzida, acarretando no aumento da FRR e numa pequena variação da FAR.

4.2.5 Precisão do Tempo

A precisão do tempo utilizado é de um milissegundo, pois está de acordo com a ordem de grandeza das características extraídas relacionadas ao tempo. Para verificar o impacto da precisão do tempo, um experimento é realizado aumentando a precisão de um milissegundo para 10 milissegundos. Na tabela 4.6 é mostrada a FRR e a FAR obtidas neste experimento (II) (com uma precisão de 10 milissegundos) comparado com as obtidas na metodologia (I) (com uma precisão de 1 milissegundo).

Tabela 4.6: FRR e FAR obtidas no experimento de precisão do tempo para (I) 1 milissegundo e (II) 10 milissegundos

	Precisão do Tempo	
	(I)	**(II)**
FRR (%)	1.55	1.67
FAR (%)	1.91	3.8

Como observado na tabela 4.6, a utilização de uma precisão de 10 milissegundos tem um impacto negativo em ambas as taxas, principalmente na FAR, que aumentou de 1.91% para 3.8%, enquanto a FRR, aumentou de 1.55% para 1.67%. Assim, a utilização da precisão de 1 milissegundo é indicada.

4.2.6 Quantidade de Tentativas

Duas tentativas são concedidas para cada usuário, pois de acordo com [26], os usuários legítimos falhavam na primeira tentativa de autenticação, mas acertavam na segunda. Para verificar o impacto da quantidade de tentativas, um experimento é realizado concedendo apenas uma tentativa para cada sessão de autenticação. Na tabela 4.7 é mostrada a FRR e a FAR obtidas neste experimento (II) (com apenas uma tentativa) comparado com as obtidas na metodologia (I) (com duas tentativas).

Tabela 4.7: FRR e FAR obtidas no experimento de quantidade de tentativas para (I) duas tentativas e (II) uma tentativa

	Quantidade de Tentativas	
	(I)	**(II)**
FRR (%)	1.55	10.2
FAR (%)	1.91	1.86

Como observado na tabela 4.7, a utilização de apenas uma tentativa em cada sessão de autenticação tem um grande impacto negativo na FRR, aumentando de 1.55% para 10.2%, e um pequeno impacto positivo na FAR, diminuindo de 1.91% para 1.86%. Assim, a concessão de duas tentativas para cada sessão de autenticação obtém taxas mais ideais comparadas com as obtidas concedendo apenas uma tentativa.

4.2.7 Mecanismo de Adaptação

Um mecanismo de adaptação é adotado para que as mudanças ocorridas na digitação sejam absorvidas, e os *templates* estejam atualizados representando sempre o usuário. Na metodologia, este mecanismo é ativado quando a maioria das características está dentro do limiar. Porém, em [24], este mecanismo é ativado sempre que um usuário é considerado legítimo, e, em [3], [8], [10], [18]-[23] e [25]-[27], este mecanismo não é utilizado. Para verificar o impacto do mecanismo de adaptação, dois experimentos são realizados: (II) desabilitando o mecanismo de adaptação e (III) o ativando sempre que um usuário é considerado legítimo. Na tabela 4.8 é mostrada a FRR e a FAR obtidas nestes dois experimentos (sem o mecanismo de adaptação e sempre o ativando) comparado com as obtidas na metodologia (I) (com o mecanismo de adaptação).

Tabela 4.8: FRR e FAR obtidas no experimento de mecanismo de adaptação para (I) com adaptação, (II) sem adaptação e (III) sempre ativada.

	Mecanismo de Adaptação		
	(I)	**(II)**	**(III)**
FRR (%)	1.55	3.89	3.44
FAR (%)	1.91	4.06	9.06

Como observado na tabela 4.8, a não-utilização do mecanismo de adaptação e a sua utilização continuada têm influências negativas em ambas as taxas. No experimento sem o mecanismo (II), a FRR aumentou de 1.55% para 3.89%, e a FAR aumentou de 1.91% para 4.06%. No experimento (III) a FRR aumentou de 1.55% para 3.44%, e a FAR aumentou bastante de 1.91% para 9.06%. Assim, o mecanismo de adaptação deve ser ativado conforme o experimento (I).

4.2.8 Limiar Único

Como já explicado no capítulo 3, a determinação do limiar é dada em função do desvio das características. Para verificar a eficácia desta abordagem, um experimento foi realizado com um limiar único para todas características. Na figura 4.11, podem ser visualizadas as curvas de FRR e

FAR variando com o limiar único para todas as características. Para realizarmos uma comparação inteligível com a figura 4.11 e os resultados produzidos na metodologia com valores de limiares para cada vetor de característica, geramos a figura 4.12. O gráfico foi calculado a partir do mapeamento dos valores de limiar calculados em cada um dos vetores de características em valores percentuais, onde 50% corresponde ao valor de limiar e 100% a 0. Nesta figura, podem ser visualizadas as curvas de FRR e FAR variando com a porcentagem mencionada.

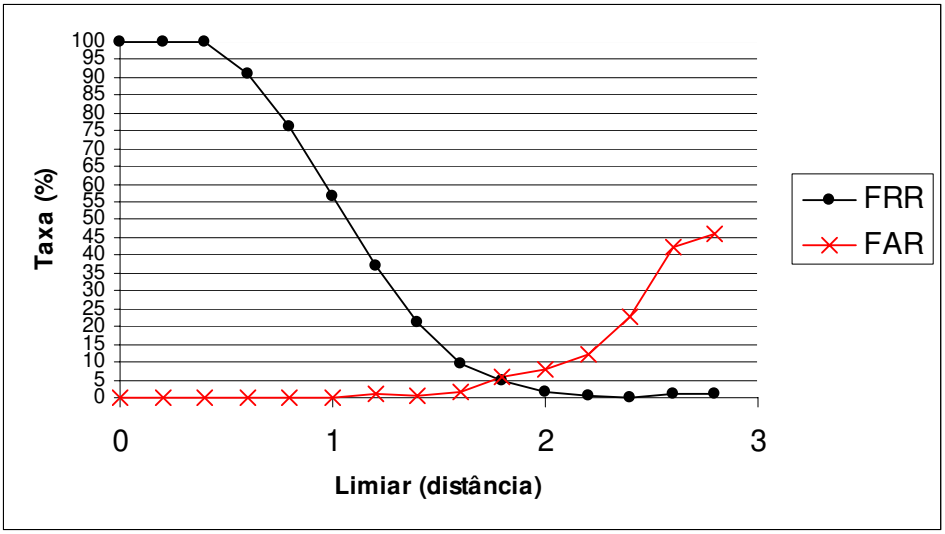


Figura 4.11: Curvas de FRR e FAR variando com um limiar

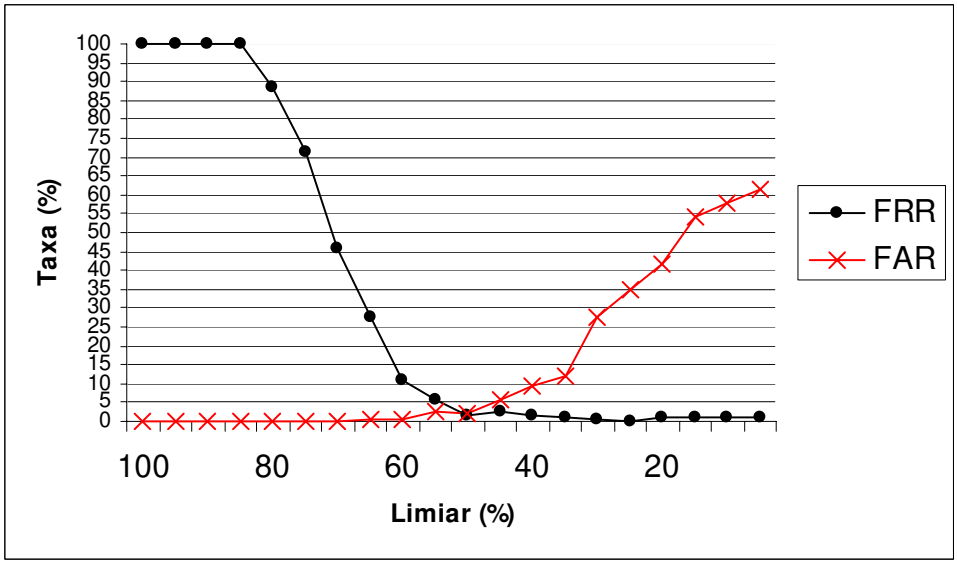


Figura 4.12: Curvas de FRR e FAR variando com uma porcentagem do limiar calculado em função do desvio

Como observado nas figuras acima, o ponto de EER (1.78%) da figura 4.12 está mais próximo de zero comparada com o da figura 4.11 (5.02%). Isto significa que a abordagem com a determinação do limiar em função do desvio é melhor que a com um único limiar, em função deste parâmetro. O EER foi adotado como fator discriminante com relação à comparação de algoritmos biométricos conforme também foi utilizado em [36].

Outra observação que pode ser feita nas figuras acima é que FAR e FRR perdem a propriedade de serem assintoticamente decrescente e assintoticamente crescente, respectivamente. Este fato ocorre por causa do mecanismo de adaptação, que atualiza os *templates* de modo que a distância calculada para a mesma amostra seja diferente à medida que os valores de limiar vão sendo modificados e os testes de autenticação vão sendo aplicados.

4.3 Discussão

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

- O classificador estatístico adotado nesta dissertação provê melhores taxas de desempenho que o nebuloso. Este resultado já era esperado, pois nos trabalhos publicados os melhores desempenhos são obtidos utilizando classificadores estatísticos;

- O conjunto de características formado pelos vetores CT, PP, SP e PS obteve os melhores resultados comparados com outras combinações, refletindo melhor a maneira como cada usuário digita;

- O vetor de características SP é o mais discriminante dentre o conjunto de características. Este vetor reflete bastante a maneira como cada usuário digita, pois capta pontos sutis, tais como, quando ele pressiona uma tecla enquanto sua antecessora ainda está pressionada, resultando em valores negativos. Ela também é discriminante pelo fato de ser difícil de se observar numa tentativa de realização de ameaça do tipo *impersonation*, mesmo como um impostor observador. Como observado no exemplo da figura 4.2, as amostras verdadeiras apresentam somente as características sp_{10} e sp_{11} negativas, enquanto que das amostras falsas apresentadas na figura 4.6 nenhuma delas contém somente as características sp_{10} e sp_{11} negativas;

- O vetor de características M não é discriminante o bastante para fazer parte do conjunto de características. Embora as amostras verdadeiras deste vetor de característica sejam bastante repetitivas, como observado na figura 4.4, o mesmo acontece para as amostras falsas provenientes de impostores como observado na figura 4.8;

- A extração de características em função das teclas utilizadas na digitação da informação alvo, resulta em vetores maiores (no pior caso igual) e mais discriminantes como observado na seção 4.2.1 (tabela 4.4);

- A escolha da informação alvo é um ponto importante, pois o grau da dificuldade de sua digitação pode ser aumentado utilizando letras maiúsculas, resultando em um vetor de CT mais discriminante. Assim, a informação pode ter algumas indicações e restrições, como possuir pelo menos 11 caracteres pelos resultados obtidos na seção 4.2.3 (figura 4.9), mas ela deve ser escolhida e não imposta ao usuário de acordo com a seção 3.2.2 (tabela 4.5).

- A quantidade de amostras presentes no conjunto de treinamento é um ponto crucial, pois quanto maior esta quantidade, a média e o desvio serão mais confiáveis com relação às características de digitação do usuário, porém a aceitabilidade pública da metodologia diminui. Na seção 4.2.4, um experimento foi realizado reduzindo a quantidade destas amostras de dez para até seis, e como observado na figura 4.10 a FRR aumenta com esta redução, enquanto que a FAR diminui;

- A precisão na captura dos tempos foi adotada em 1 milissegundo, pois estava compatível com a ordem de grandeza das características extraídas relacionadas a tempo. Na seção 4.2.5, um experimento foi realizado modificando esta precisão e as taxas obtidas foram maiores que as já obtidas (tabela 4.6);

- A concessão de duas tentativas em cada sessão de autenticação resulta em melhores resultados comparados com os obtidos com apenas uma tentativa de acordo com a seção 4.2.6. Como observado na tabela 4.7, embora a FAR diminua de 1.91% para 1.86%, a FRR aumenta bastante de 1.55% para 10.2%. Isto demonstra que os usuários legítimos falham na primeira

tentativa, mas acertam na segunda, enquanto que os usuários impostores falham na primeira e na segunda tentativas;

- O mecanismo de adaptação é outro ponto importante, com a finalidade de manter os *templates* sempre atualizados com as mudanças que venham a ocorrer na digitação de cada usuário. Na seção 4.2.7, dois experimentos foram realizados: um sem o mecanismo de adaptação e outro com a sua ativação sempre que uma autenticação era realizada com sucesso. Como observado na tabela 4.8, os dois experimentos obtiveram taxas maiores que as já obtidas.

- A determinação do limiar em função do desvio padrão das características obtém resultados melhores comparados com os obtidos com um único limiar como apresentado na seção 4.2.7. O ponto de EER (5.02%) da figura 4.11 relativa ao limiar único está acima do ponto de EER (1.78%) da figura 4.12 relativa a outra abordagem. Isto acontece, pois, como foi observado, o limiar de decisão ideal varia com o desvio padrão das características, e, portanto, cada vetor de característica deve possuir o seu próprio limiar.

A metodologia para autenticação pessoal baseada em dinâmica da digitação obteve os melhores resultados (1.91% FAR e 1.55% FRR) com a seguinte configuração:

1. Utilização de em conjunto de características com os vetores *CT*, *SP*, *PS* e *PP*;

2. Aplicação de um classificador estatístico baseado na distância padrão;

3. Extração de características foi realizada em função das teclas;

4. Escolha da informação alvo feita pelo usuário;

5. Restrição do tamanho da informação alvo para pelo menos 11 caracteres;

6. Composição do conjunto de treinamento com 10 amostras;

7. Utilização de uma precisão de tempo de 1 milissegundo;

8. Composição de uma sessão de um usuário com duas tentativas de autenticação;

9. Ativação de um mecanismo de adaptação quando a maioria das características está dentro do limiar, para atualizar o *template* de um usuário existente na base de dados;

10. Determinação dos valores de limiar em função do desvio padrão de cada vetor de característica apresentado no conjunto de treinamento de cada usuário.

Diante destes resultados, calcularemos, a seguir, o intervalo de confiança sobre as taxas de FAR e FRR obtidas.

4.3.1 Intervalo de Confiança

Ao fornecermos os valores de desempenho relacionados a um sistema biométrico, um questionamento que se torna iminente é o quanto estes valores representam a verdadeira acurácia do sistema. Os valores de FAR e FRR sofrem variações caso sejam utilizados conjuntos de testes diferentes daquele que foi utilizado para calcular estas taxas inicialmente. Estas variações estão contidas dentro de um limite medido em termos de um nível de confiança (LOC). O LOC mede o limite de erro aceitável entre o valor real de acurácia do sistema e o valor de acurácia medido pela aplicação de um conjunto de teste. A precisão deste teste é determinada pelo intervalo de confiança do teste [37].

O intervalo de confiança *IC* é calculado fazendo uso da equação 4.3 [38]:

$$IC = z/\left[p \times (1 - p)/n\right]^{1/2} \tag{4.3}$$

onde p representa a acurácia assumida para o sistema (decimal), n é a quantidade de amostras do conjunto de testes, e z é um valor derivado de uma curva normal a partir do valor de LOC. A tabela 4.9 mostra uma lista de alguns valores de z relacionados aos seus respectivos valores de LOC [38].

Tabela 4.9: Valor de z para um dado LOC.

LOC	z
99.9%	3.3
99.0%	2.577
98.5%	2.43
97.5%	2.243
95.0%	1.96
90.0%	1.645
85.0%	1.439
75.0%	1.151

Para calcularmos o intervalo de confiança dos valores de FAR e FRR obtidos pela metodologia nesta dissertação, precisamos determinar os valores que as variáveis da equação 4.3 devem assumir:

- Para o valor de p é atribuído o valor de 50%. Segundo [38], este valor equivale a dizer que não existe ou existe pouco conhecimento prévio a respeito da real acurácia de um sistema biométrico.

- Para o valor de LOC, a literatura relacionada à estatística comumente faz uso do valor de 95% para o cálculo de *IC* [38]. Assim, o valor de z na equação 4.3, de acordo com a tabela 4.9, é igual a 1.96.

- O valor de n depende da taxa (FAR ou FRR) a qual estamos calculando o intervalo de confiança. Para a FAR, o valor de n será igual a quantidade de amostras coletadas nos testes com usuários impostores, ou seja, $n = 3500$. Para a FRR, o valor de n será igual a quantidade de amostras coletadas nos testes com usuários legítimos, ou seja, $n = 900$.

Aplicando a equação 4.3 para calcular o intervalo de confiança (*IC*) para FAR, obtemos o valor de $\pm 1.65\%$, e para FRR, obtemos o valor de $\pm 3.27\%$. Baseado nestes valores, com relação à aceitação de usuários legítimos, a partir do valor de FAR calculado nos testes de 1.91%, a metodologia possui uma acurácia de 98.09% em autenticar um usuário legítimo com um intervalo de confiança de $\pm 1.65\%$. Com relação à rejeição de usuários impostores, a partir do

valor de FRR produzido nos testes de 1.55%, a metodologia possui uma acurácia de 98.45% em rejeitar um usuário impostor com um intervalo de confiança de $\pm 3.27\%$.

CAPÍTULO 5

CONCLUSÃO

Neste capítulo colocamos alguns comentários sobre o trabalho apresentado, enfatizando as contribuições oferecidas e os trabalhos futuros que podem ser desenvolvidos a partir do trabalho apresentado nessa dissertação.

5.1 Contribuições

Nesta dissertação apresentamos uma metodologia para verificação da identidade de usuários via dinâmica da digitação. Esta metodologia foi desenvolvida com base em trabalhos já publicados na área, que foram discutidos no capítulo 2, além de vários experimentos que realizamos ao longo deste trabalho.

Várias contribuições foram produzidas. Uma delas é relativa ao estado da arte apresentado no capítulo 2, enquanto as demais são relativas aos experimentos realizados no capítulo 4, onde analisamos mudanças em diferentes aspectos abordados na metodologia. Podemos citá-las abaixo:

- A comparação entre os classificadores estatístico e nebuloso, onde mostramos que o classificador estatístico adotado neste trabalho obteve os melhores resultados com relação ao classificador nebuloso;

- A comparação entre combinações de características. Neste contexto, nós comparamos e analisamos os resultados obtidos com a utilização de vetores de características isoladamente ou combinações deles.

- A utilização dos códigos das teclas como um vetor de característica é uma das inovações deste trabalho. Outra inovação é a combinação de dois tipos de latências de teclas, que foram expressadas nas características PP e SP, com a duração de teclas.

- Uma análise foi feita com relação a medidas estatísticas, como média, desvio padrão, valor máximo e valor mínimo, das latências e das durações de teclas. Porém, o vetor M relativo a estas medidas não se mostrou discriminante para os usuários das contas cadastradas na base de dados.

- O conjunto de características analisado que apresentou os melhores resultados nos experimentos que foram realizados é composto pelos vetores CT, PP, SP e PS;

- A extração das características (latências e durações de teclas) em função das teclas utilizadas é outra inovação desta dissertação, que obtém melhores resultados comparados com os obtidos com as características extraídas em função dos caracteres (abordagem utilizada por outros trabalhos);

- A comparação entre informações alvos escolhidas e impostas, onde os melhores resultados foram obtidos com àquelas escolhidas por cada usuário;

- A comparação entre as quantidades de caracteres presentes nas informações alvos, onde foi observado que os erros de classificação aumentam mais quando as informações alvos possuem menos que 11 caracteres;

- A comparação entre as quantidades de amostras do conjunto de treinamento, onde foi observado que a FRR aumenta e a FAR diminui com a redução da quantidade de amostras para um valor menor que 10.

- A comparação entre precisões diferentes na captura dos tempos, onde foi mostrado que uma precisão de 1 milissegundo está de acordo com a ordem de grandeza das características extraídas, conforme o que foi definido inicialmente;

- A comparação entre a concessão de duas tentativas e apenas uma tentativa em cada sessão de autenticação, onde os melhores resultados são obtidos com duas tentativas;

- A utilização do mecanismo de adaptação para atualizar os *templates* com as mudanças ocorridas na digitação. Foi mostrado que este mecanismo é de suma importância para os resultados, pois as características biométricas extraídas da dinâmica da digitação possuem uma variação com o decorrer do tempo. Dois experimentos foram analisados: um desabilitando o mecanismo e o outro com sua ativação contínua sempre que uma sessão era realizada com sucesso. Nestes dois experimentos os resultados foram piores que os obtidos com a sua ativação somente quando a maioria das características estava dentro do limiar de decisão;

- A determinação do limiar de decisão em função do desvio padrão das características é outra inovação, que obteve melhores resultados comparados como os obtidos com um limiar único (abordagem utilizada por outros trabalhos), que acaba por prejudicar o desempenho de classificação em alguma ou algumas das características;

- As taxas obtidas (1.55% FRR e 1.91% FAR) e as características apresentadas na metodologia são ambas competitivas com as obtidas em trabalhos publicados nesta área, quando observamos os dados apresentados na tabela 2.1.

5.2 Perspectivas para Trabalhos Futuros

Com relação a trabalhos futuros, pretendemos estender a metodologia para teclados numéricos [39]. Este tipo de teclado é muito utilizado em acesso a áreas restritas e em terminais de banco aonde os clientes realizam muitos tipos de transações bancárias [40]. A metodologia desenvolvida adicionaria em segurança para os sistemas implementados nestes tipos de aplicações práticas.

Outra intenção é adaptar a metodologia para propósitos criptográficos. Desta forma, geraríamos uma chave criptográfica baseada nas características de dinâmica da digitação do usuário [41]. Isto resultaria em uma chave criptográfica denominada *bio-key* que agregaria em segurança para aplicações que fazem uso de criptografia [42].

Uma aplicação prática da metodologia vem ocorrendo atualmente em um projeto financiado pelo CNPq no Laboratório de Reconhecimento de Padrões e Redes de Computadores (LRPRC) da Universidade Estadual de Campinas (UNICAMP), ligada a reconhecimento pessoal utilizando características biométricas em celulares e *handhelds* [43][44]. A metodologia desenvolvida neste projeto vêm sendo adaptada e incorporada em celulares utilizando a tecnologia desenvolvida na plataforma do Java MicroEdition. Desta forma, tentaremos fazer o reconhecimento pessoal via a captação das características biométricas extraídas da digitação no teclado do celular.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] D. Polemi, 1997, "Biometric Techniques: Review and Evaluation of Biometric Techniques for Identification and Authentication, Including an Appraisal of the Areas Where They are Most Applicable," Final Report. [*on-line*] Disponível na internet via *Web*. URL: ftp://ftp.cordis.lu/pub/infosec/docs/biomet.doc. Arquivo consultado em 18 de outubro de 2003.

[2] Mathyas, S.M., Stapleton, J., "A biometric standard for information management and security", *Computers & Security*, Vol.19, Nº 5, pp. 428-441, 2000.

[3] Joyce, R., Gupta G. "Identity authorization based on keystroke latencies", *Communications of the ACM*, Vol. 33., Nº 2, pp. 168 - 176, 1990.

[4] Holanda, Aurélio Buarque. Dicionário Aurélio Eletrônico, Versão 1.4, 1994.

[5] J. Ashbourn, 2000, "The Biometric White Paper". [*on-line*] Disponível na internet via *Web*. URL: http://www.biometric.freeserve.co.uk/whitepaper.htm. Arquivo consultado em 04 de fevereiro de 2003.

[6] Kapoor, T., Kapoor, M., Sharma, Gp., "Study of the form and extent of natural variation in genuine writings with age", *Journal of the Forensic Science Society*, Vol. 25, pp. 371 - 375, 1985.

[7] Bruyne, P., "Signature verification using holistic measures", *Computers and Security*, Vol 4, pp 309 - 315, 1985.

[8] Monrose, F., Rubin, A.D., "Keystroke Dynamics as a Biometric for Authentication", *Future Generation Computer Systems*, Vol. 16, Nº 4, pp. 351-359, 2000.

[9] Jain, A., Duin, R., Mao, J., "Statistical pattern recognition: a review"*, IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 22, Nº 1, pp. 4 - 37, 2000.

[10] De Ru, W. G., Eloff, J. H. P., "Enhanced Password Authentication through Fuzzy Logic", *IEEE Expert / Intelligent Systems & Their Applications*, Vol. 17, No. 6, pp. 38-45, 1997.

[11] Duda, R., Hart, P. Pattern classification and scene analysis. Wiley, New York, 1973.

[12] Pedrycz,W., Gomide, F. An Introduction to Fuzzy Sets: Analysis and Design. MIT Press, Cambridge, 1998.

[13] Mamdani, E. H., Assilian, S., "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller". *International Journal of Man-Machine Studies*, Vol. 7, Nº. 1, pp. 1 - 13, 1975.

[14] Delgado, M.R.B.S. Projeto Automático de Sistemas Nebulosos: Uma Abordagem Co-Evolutiva. Dissertação de Doutorado, UNICAMP, Campinas, 2002.

[15] Leggett. J., Williams, G., Usnik, M., "Dynamic identity verification via keystroke characteristics", *International Journal of Man-Machine Studies*, Vol. 35, pp. 859 - 870, 1990.

[16] Umphress, D., Williams, G., "Identity verification through keyboard characteristics", *International Journal of Man-Machine Studies*, Vol. 23, pp. 263 - 273, 1985.

[17] Leggett, J., Williams, G., "Verifying identity via keystroke characteristics", *International Journal of Man-Machine Studies,* Vol. 28, pp. 67 - 76, 1987.

[18] Young, J.R., Hammon, R.W., "Method and Apparatus for Verifying an Individual's Identity". Patent Number 4,805,222. U.S. Patent and Trademark Office, Washington, D.C., 1989.

[19] Bleha, S., Slivinsky, C., Hussain, B., "Computer-Access Security Systems Using Keystroke Dynamics", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 12, Nº. 12, pp. 1217-1222, 1990.

[20] Bleha, D., Obaidat, M., "Dimensionality Reduction and Feature Extraction Applications in Identifying Computer Users", *IEEE Trans. Syst., Man, Cybern.*, Vol. 21, pp. 452-456, 1991.

[21] Lin, D.T, "Computer-Access Authentication with Neural Network Based Keystroke Identity Verification", *proceedings in International Conference on Neural Networks*, Vol. 1, pp. 174-178, 1997.

[22] Obaidat, M.S., Sadoun, B, "Verification of Computer User Using Keystroke Dynamics", *IEEE Trans. Syst., Man, Cybern.*, Vol. 27, Nº. 2, pp. 261-269, 1997.

[23] Robinson, J. A., Liang, V. M., Michael, J. A., MacKenzie, C. L., "Computer User Verification Login String Keystroke Dynamics", *IEEE Trans. Syst., Man, Cybern.*, Vol. 28, No. 2, pp. 236-241, 1998.

[24] Monrose, F., Reiter, M.K., Wetzel, S, "Password Hardening Based on Keystroke Dynamics", *6th ACM Conference on Computer Security*, 1999.

[25] Coltell, O., Badfa, J.M., Torres, G., "Biometric Identification System Based in Keyboard Filtering", *Proceedings IEE 33rd Annual 1990 International Carnahan Conference on Security Technology*, pp. 203-209, 1999.

[26] Haider, S., Abbas, A., Zaidi, A. K., "A Multi-Technique Approach for User Identification through Keystroke Dynamics", *IEEE Int. Conference of Syst., Man and Cybern.*, Vol. 2, pp. 1336-1341, 2000.

[27] Wong, F.W.M.H., Supian, A.S.M., Ismail A.F., Kin, L.W., Soon, O.C, "Enhanced User Authentication through Typing Biometrics with Artificial Neural Networks and K-Nearest Neighbor Algorithm", *Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers*, Vol. 2, pp. 911-915, 2001.

[28] BioNet Systems, LLC, (2003), "BioPassword", Versão 4.6. [*on-line*] Disponível na internet via *Web*. URL: http://www.biopassword.com/home/products/bp45.asp. Arquivo consultado em 20 de outubro de 2003.

[29] Chan, K. C. C., Wong, A. C. K., "APACS: A system for the automatic analysis and classification of conceptual patterns", *Comput. Intell.*, Vol. 6, pp. 119 – 131, 1990.

[30] Araújo, L.C.F., Sucupira Jr., L.H.R., Lizarraga, M.G., Ling, L.L., Yabu-uti, J.B.T., "A Fuzzy Logic Approach in Typing Biometrics User Authentication", *Proc. of 1st Indian International Conference on Artifitial Intelligence*, pp.1038-1051, India, Dezembro, 2003.

[31] Doug Hogarth, (2001), "TIMESERV for Microsoft Windows NT Resource Kit". [*on-line*] Disponível na internet via *Web*. URL: http://www.niceties.com/timeserv.html. Arquivo consultado em 15 de março de 2003.

[32] Vladimir Roubtsov, (2003), "Reach Submillisecond Timing Precision in JAVA". [*on-line*] Disponível na internet via *Web*. URL: http://www.javaworld.com/javaworld/javaga/2003-01/01-qa-0110-timing.html. Arquivo consultado em 15 de março de 2003.

[33] Intel Corporation, (1997), "Using the RDTSC Instruction for Performance Monitoring". [*on-line*] Disponível na internet via *Web*. URL: http://developer.intel.com/drg/ pentiumII/appnotes/RDTSCPM1.HTM. Arquivo consultado em 13 de março de 2003.

[34] Belchior, A.D., Xexéo, G., da Rocha, A.R.C., "Evaluating Software Quality Requirements Fuzzy Theory", *Proceedings of ISAS 96*, Orlando, Julho, 1996.

[35] Oliveira, F. Estatística e probabilidade. Editora Atlas, 2da Edição, 1999.

[36] Maio, D., Maltoni D., Cappelli R., Wayman J. L., Jain A. K., "FVC2002: Second Fingerprint Verification Competition", *Proc. of International Conference on Pattern Recognition*, Quebec City, 2002.

[37] Bolle, R. M., Connell, J., Pankanti, S., Ratha, N. K., Senior, A. W., (2001), "Biometrics 101", IBM Research Report, version 6.05. [*on-line*] Disponível na internet via *Web*. URL: http://domino.watson.ibm.com/library/cyberdig.nsf/papers/d504f9e5ecb3ffdc85256bd40073df55/$file/rc22841.pdf. Arquivo consultado em 10 de dezembro de 2003.

[38] Biometric Technology, INC, (2003), "How Accurate is the Biometric?". [*on-line*] Disponível na internet via *Web*. URL: http://bio_tech_inc.com/how_accurate_is_the_biometric.htm. Arquivo consultado em 10 de dezembro de 2003.

[39] Targus Cases and Accessories, (2003), "Targus USB Retractable Calculator Keypad". [*on-line*] Disponível na internet via *Web*. URL: http://www.targus.com/us/acessories_key.asp. Arquivo consultado em 15 de outubro de 2003.

[40] Allen-Bradley Controls and Engineering Services, (1999), "Boletín 160 SSC Teclado Númerico Remoto FRN 1.X", Manual del Usuário. [*on-line*] Disponível na internet via *Web*. URL: http://www.ab.com/manuals/es/dr/0160-5.28ml-es.pdf. Arquivo consultado em 02 de novembro de 2003.

[41] Juels, A., Wattenberg, M., "A Fuzzy Commitment Scheme", *Proceedings of 6th ACM Conference on Computer and Communication Security*, pp. 28 – 36, 1999.

[42] Jermyn, I., Mayer, A., Monrose, F., Reiter, M. K., Rubin, A. D., "The Design and Analysis of Graphical Passwords", *Proceedings of the 8th Usenix Security Symposium*, Washington, USA, Agosto, 1999.

[43] Clarke, N. L., Furnell, S. M., Lines, B., Reynolds, P. L., "Keystroke Dynamics on a Mobile Handset: A Feasibility Study", *Information Management and Computer Security*, Vol. 11, Nº. 4, pp. 161 – 166, 2003.

[44] Jansen, W., "Authenticating Users on Handheld Devices", *Proceedings of Canadian Informaiton Technology Security Symposium*, Maio, 2003.

APÊNDICE A

ARTIGOS ELABORADOS

A.1 Introdução

Durante este trabalho, foram elaborados os seguintes artigos:

- Araújo, L.C.F., Sucupira Jr., L.H.R., Lizarraga, M.G., Ling, L.L., Yabu-uti, J.B.T., "A Fuzzy Logic Approach in Typing Biometrics User Authentication", *Proc. of 1st Indian International Conference on Artifitial Intelligence (IICAI-03)*, pp.1038-1051, India, 2003.
- Araújo, L.C.F., Sucupira Jr., L.H.R., Lizarraga, M.G., Yabu-uti, J.B.T., "User Authentication through Typing Biometrics Features", *Proc. of International Conference on Biometric Authentication (ICBA-04)*, Hong Kong, 2004.
- Araújo, L.C.F., Lizarraga, M.G., Sucupira Jr., L.H.R., Yabu-uti, J.B.T., "Autenticatión Personal por Dinámica de Tecleo Basada em Lógica Difusa", *IEEE Latin America Transactions*, submetido para revisão.
- Araújo, L.C.F., Sucupira Jr., L.H.R., Lizarraga, M.G.., Yabu-uti, J.B.T., "A Methodology for User Authentication through Keystroke Dynamics", *IEEE Transactions on Signal Processing Suplement on Secure Media*, Março, 2005, submetido para revisão.

Até o momento, os dois primeiros artigos foram aceitos e publicados. Com relação aos demais, ainda aguardamos resposta a respeito do resultado da revisão.

A.2 Artigo IICAI-03

A Fuzzy Logic Approach in Typing Biometrics User Authentication

Lívia C. F. Araújo[1], Luiz H. R. Sucupira Jr.[1], Miguel G. Lizárraga[1], Lee L. Ling[1], João B. T. Yabu-uti[1]

[1]School of Electrical and Computer Engineering, State University of Campinas

Albert Einstein Avenue, 400, PO Box 6101, Postal Code 13083-970, Campinas, SP, Brazil

{liviacri,luigijr,lizarrag,lee,yabuuti}@decom.fee.unicamp.br

Abstract. This paper uses a fuzzy logic approach in a static typing biometrics user authentication. The inputs are the down and up times, and the ASCII code of the keys captured while the user is typing a known string. In this research, we collected four features (key code, two keystroke latencies and key duration) captured in two different strings (one imposed and the other chosen by each user). Seven experiments were developed utilizing fuzzy logic classifier and combining the features. The results of the experiments are evaluated in three situations of authentication: the legitimate user, the impostor and the observer impostor. The best results were achieved utilizing all features, obtaining a false rejection rate of 3,5% and a false acceptance rate of 2,9%. This approach can be used in the usual login-password authentication for improvement of the false acceptance rate, when the password is no more a secret. Keywords - fuzzy logic, pattern recognition, biometrics, typing biometrics.

1 Introduction

The control access is a very important issue in the computer systems. The systems could grant access (authentication) using one or more of the ways below [1]:

- knowledge - on the basis of something the user knows, e.g. password;
- possession - on the basis of something the user has, e.g. smart card;
- biometrics - on the basis of something the user is, e.g. fingerprint.

The login-password authentication, based on knowledge, is the most usual mechanism. This authentication is fragile when there is a careless user and/or a weak password. Otherwise, it is low-cost and familiar. The purpose of this paper is to improve this mechanism using biometrics whose characteristics could not be stolen, lost or forgotten, besides the uniqueness among people [2].

In biometric systems, there are two main processes: enrollment and authentication. Enrollment consists of creating a template that contains the patterns found in inputs,

This work was partially supported by CNPq, CAPES and FAPESP.

and associating it to a new user. Authentication confirms (verification) or indicates (identification) the identity of the user. Verification compares the inputs with a particular template and identification compares them with more than one template [1].

The chosen biometric technology is the typing biometrics, also known as keystroke dynamics. The typing biometrics is the process of analyzing the way a user types at a terminal by monitoring the keyboard inputs in attempt to identify users based on their habitual typing rhythm patterns. There are two approaches: static and continuous. The static approach analyses the inputs just in a particular moment, and the continuous one analyses the inputs during all user's session [3].

The methodology of this paper is low-cost (using a conventional keyboard) and unintrusive (using a password or a login), and, uses a verification authentication in a static approach (using the login session).

This paper is organized as follow: In section 2, there is a resume of related studies published; in section 3, the methodology is explained and some details are described as the classifier and the features; in section 4, the results are presented and discussed; and in the section 5, there is the conclusion and the future work.

2 Related Work

Some studies were published [3]-[14] in the authentication via typing biometrics since 1990. Some aspects in the methodologies presented in these works are resumed below:

- *Target string*: It is an input string that will be the monitored by the system. In [10], four strings (login, password, first name, last name) were used as target. Two target strings were analyzed in [11]: a 31-char string and a login itself. In [9], target strings were divided in three levels based on a difficulty degree. Finally, in some studies the password itself was utilized. String length is a very important issue, considering that in [14] was stated that misclassification increases as the string length drops to as few as ten characters;
- *Amount of samples*: Samples are collected during the enrollment process for creating the template. Its amount varies a lot, since it was as few as 3 samples per users in [7] and, as many as 15 samples per user in [12];
- *Features*: A good feature has to be highly repeatable in the same user and different between users [4]. Two most used features is *duration of the key*, that is the time duration of each keystroke, and *keystroke latency*, that is the time interval between successive keystrokes [7]. In [6] and [8], the combination of these features brought better results than using them isolated. In [5] the typing difficulty based on the distance of the keys in the keyboard and the combination of keys, besides the keystroke latency, were analyzed;
- *Timing accuracy*: As the most of the typing biometrics features are time-based, the precision of the up and down times of the keys have to be analyzed. The timing accuracy in the studies varies between 0.01second [12] and 1 second [9].
- *Adaptation mechanism*: Some biometric characteristics change slowly over time [1]. The typing biometrics is one of these characteristics. An adaptation or a re-enrollment could be performed to maintain the templates updated. The majority of

the studies did not mention this issue, but, in [4], an adaptation mechanism was utilized. This mechanism consists of creating a new template updated every time when a successfully authentication is performed, including the new sample and discarding the oldest one.

- *Classifier*: In [9] and [10], a statistical classifier was used, analyzing the mean and standard deviation of the features. A pattern recognition approach was studied in [6], [3], [11] and [14] using known techniques as k-means, Bayes, etc. In [5], fuzzy logic was applied using three linguistic variables: time interval (input), typing difficulty (input) and categorization (output). Finally, in [7], neural networks were experimented, although in [3], it was explained that this classifier is not appropriated to access controls systems because of training requirements (e.g. time consuming). In [12] and [8], the results of more than one classifier were compared.

3 Methodology

Each time a user tries to access a system, he has to input a string called *target string*. The keyboard events are captured in some *timing accuracy,* and four *features* are calculated. These *features* are the key code and three time-features (down-down, down-up and up-down times). If the user is a new one, then an *amount of samples* has to be stored. A sample is just stored if the string and the key code feature both match. When the samples suffit, a template will be created containing the patterns found in the samples. If the user is not a new one, then his sample will be compared with the template stored in database. If the string matches, a c*lassifier* will be applied to determine if the sample is similar to the template, and the user will just be authenticated depending on the degree of this similarity. If the user is authenticated, an *adaptation* mechanism is called to compute a new template updated. The representative flowchart of these main steps of the methodology can be visualized in the Fig. 1.

The main issues related to the methodology are described in the next sub-sessions: target string, amount of samples, features, timing accuracy, adaptation mechanism and classifier.

3.1 Target String

Two strings were used to observe the influence of the familiarity of the string in the results: one string was imposed to all users and the other one was chosen by each user. The imposed string was "*unicamp@2003*". This string was chosen because it was simple to memorize and type for the group of users involved with the acquisition of samples. The other string chosen was familiar to each user (the user usually types it) and its length had to be of at least of 10 characters according to [14]. The majority of the users choose their names or their e-mail accounts.

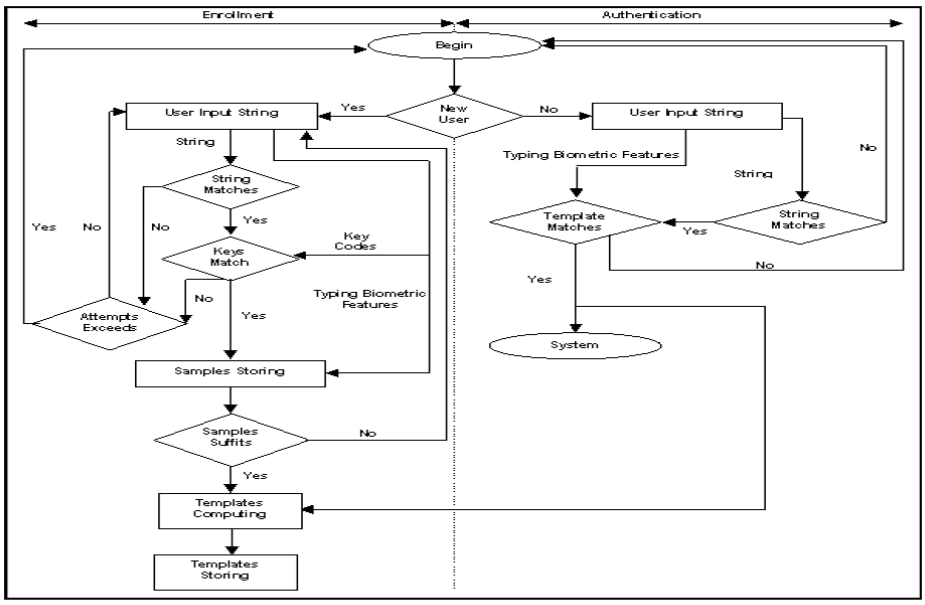


Fig. 1. Flowchart of the methodology

3.2 Amount of Samples

In enrollment, ten samples of the two target strings were collected per user. This amount will be analyzed for verifying if can be reduced without damage the results.

3.3 Features

Features are calculated using data collected while a user types a string. A target string with *m* characters will result in at least *m* keystrokes, since some characters need more than one keystroke. For example, the character "*X*" needs the "*x*" and the "*shift*" keys.

Four features were analyzed: key code, down-down time, up-down time and down-up time.

3.3.1 Key Code

It is the ASCII code of the key. When a user types a string and *n* keys are pressed, this will result in *n* elements of the key code feature.

This feature is repeatable, but only different between users when the target string contains capital letters. For example, using the two target strings "*dad*" and "*DAD*" (ASCII codes: a=65, d=68): the first one will result in just one possible set of the key code ({{68, 65, 68}}), meanwhile the second one will result in more than one possible set, pressing the shift key (ASCII code: 16) or the caps lock key (ASCII code: 20) {{20,68,65,68}, {16,68,65,68}, {16,68,16,65,16,68}, {16,68,65,16,68} …}. A user will just be authenticated, if the set of key codes is the same contained in the

template stored. So, if an impostor uses the caps lock key to type a target string and the template's owner uses the shift key, then the impostor will not be authenticated.

This feature was used combined with other features, and was applied just after the string matching.

3.3.2 Down-down Time

Keystroke latency is the time interval between successive keystrokes [7], and, digraph latency time is between only two successive keystrokes. Down-down time is a digraph latency feature. This feature is calculated performing the difference between the down times of two consecutive keys. When a user types a string and *n* keys are pressed, this will result in *n-1* elements of the down-down time feature.

3.3.3 Up-down Time

Up-down time is also a digraph latency feature. This feature is calculated performing the difference between the up time of a key and the down time of the consecutive key pressed. When a user types a string and *n* keys are pressed, this will result in *n-1* elements of the up-down time feature.

There are two situations that could be visualized in the Fig. 2. In the situation 1, the key K2 is only pressed when the key K1 was released. In the situation 2, the key K2 is pressed while the key K1 was still pressed. This feature results in negative values when the situation 2 occurs.

3.3.4 Down-up Time

Down-up time is the time duration of each keystroke [7] and is known as duration of the key. This feature is calculated performing the difference between the down and the up time of the same key. When a user types a string and *n* keys are pressed, this will result in *n* elements of the down-up time feature.

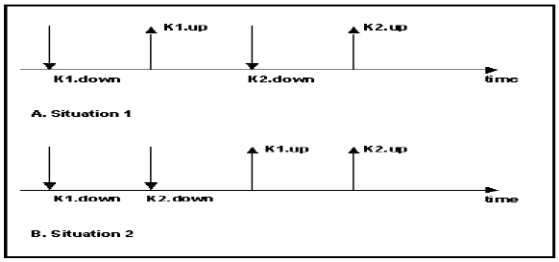


Fig. 2. Two situations: up-down time is positive in *situation 1*, and, negative in *situation 2*

3.4 Timing Accuracy

The basic foundation for the typing biometrics is to have an accurate and reliable data source of typing patterns in time [13]. The Time Stamp Counter function was used to catch the count of clock cycles as in [13]. The time stamp counter keeps an accurate count of every cycle that occurs on the processor [15]. Two instructions were utilized:

rdtsc to read the counter, and, *cpuid* to force in-order execution of the instructions, providing a more reliable timing system. A 0.001 second precision was used in this paper, and this timing accuracy will be analyzed for verifying its impact to the results.

3.5 Adaptation Mechanism

As the typing biometrics features change slowly over time, an adaptation mechanism is required to maintain templates updated. A re-enrollment could be adopted, but demands much time of users. As in [4], the adaptation mechanism consists of creating a new template updated every time when a successfully authentication is performed, including the new sample and discarding the oldest one. In the results section, this mechanism will be analyzed for verifying its efficacy in the authentication.

3.6 Classifier

This methodology does not deal with typographic errors, so the users could not make errors during their typing. Besides, the codes of the keys are analyzed and the users will just be authenticated if they match with the template stored. So, the classifier is only applied after the successful match with the target string and the key code feature.

A fuzzy logic classifier was chosen because of its capability of dealing with ambiguous data, analyzing it as a matter of degree, not as traditional logic (true or false, not both) [5]. Three steps are performed in the beginning of this approach. First, it is calculated the mean and the standard deviation of the elements of each time-feature (down-down, up-down and down-up times) through samples collected during enrollment. Second, outliers are discarded. An element is an outlier if is lower than three times its standard deviation minus its mean or if is higher than three times its standard deviation plus its mean. Third, the mean of the elements, without outliers, of each time-feature is calculated. These steps are presented in the algorithm below:

```
Algorithm

Begin

- Calculate the mean (μ) and the standard deviation (σ)
vectors of the n elements (e) of each time-feature

if ((e[i][j]<μ[i]-3σ[i])or(e[i][j]>μ[i]+3σ[i]) for all
i=1,2,..k)

Then discarte(e[i][j]);

- Calculate the mean () vector of the remaining
elements of each time-feature

End
```

where *e* is an element of a time-feature, *n* is the number of elements of each time-feature and *k* is the number of samples collected in the enrollment.

The categorization of each user based on their typing style is our intent. For this purpose, four linguistic variables are used: three inputs (down-down, up-down and down-up times) and one output (categorization). Each one of these variables is associated with linguistic terms that are:

– *Down-down time*: very short (VS), short (S), medium short (MS), long short (LS);
– *Up-down time*: really short (RS), very short (VS), short (S), medium short (MS), long short (LS);
– *Down-up time*: very short (VS), short (S), medium short (MS), long short (LS);
– *Categorization*: very low (VL), low (L), medium high (MH), high (H), very high (VH).

The means of the time-features are the inputs to its related membership function. These functions will result in degrees, between 0 and 1, of compatibility to the linguistic terms related. Different membership functions were experimented, and those ones that represented the better the situation are shown in Fig. 3.

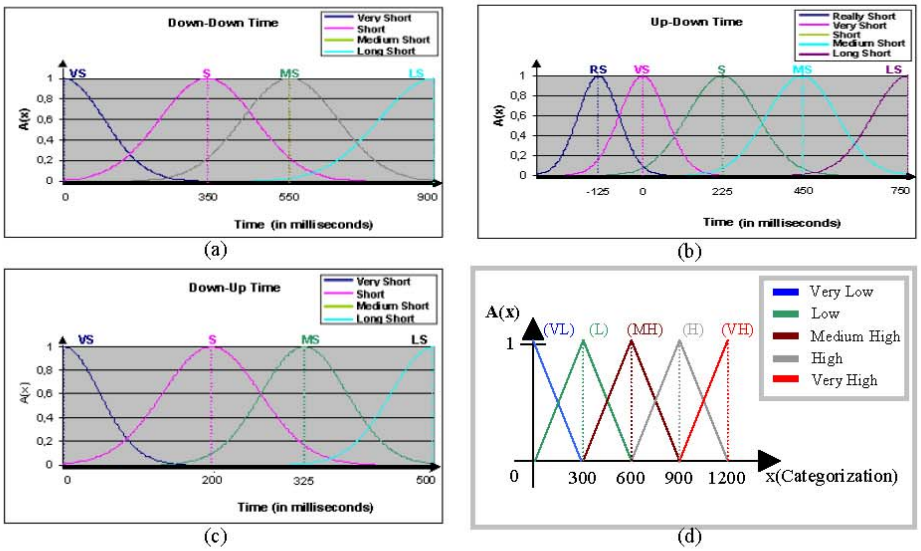


Fig. 3. Membership functions for (a) down-down time, (b) up-down time, (c) down-up time, and, (d) categorization

The degrees of compatibility will be used in rules representing knowledge. These rules were created based on: as shorter as the time-feature is, as higher as will be the user's categorization. So, an experienced user will have a higher categorization, meanwhile an inexperienced user will have a lower categorization. Five rules were created relating the inputs with the output. The rules formulated are:

Rule 1: If time is really short then the categorization is very high.

Rule 2: If time is very short then the categorization is high.
Rule 3: If time is short then the categorization is medium.
Rule 4: If time is medium short then the categorization is low.
Rule 5: If time is long short then the categorization is very low.

All rules are applied and some of them are activated depending of input degrees. The respective categorization is used in the membership function, and a triangle is truncated in the related degree. If more than one rule is activated, more than one triangle is truncated forming a single geometric form. A crisp value is calculated utilizing the center of gravity method in the form. This value is the categorization. Three categorizations are resulted, each one corresponding to a time-feature.

A user's sample is considered similar if all elements of time-features are similar to its respective categorization. A degree of similarity *S(x)* is calculated using a similarity function:

$$S(x) = e^{-k(x-m)^2} \quad . \tag{1}$$

Where *x* is the input value, *m* is the categorization and *k* is the variance. The degree of similarity *S(x)* is the threshold of this approach.

4 Experimentation

Twenty users participated of the experimentation. Initially, ten users were enrolled providing their typing biometrics samples. After, twenty users, including the enrolled ones, were informed of users' target strings. In a seven-week period, all users tried to be authenticated as the enrolled users, resulting in legitimate user or impostor user authentications. Finally, after this period, ten users were chosen randomly to observe legitimate users' sessions, resulting in observer impostor user authentications. The amount of samples collected in each one of these authentications is shown below:

- *Legitimate user authentication*: four hundred samples in two hundred sessions. In each session the user has two attempts of being authenticated.
- *Impostor user authentication*: two thousand and five hundred and sixty samples.
- *Observer impostor user authentication*: one hundred and eighty samples.

Seven experiments were analyzed, combining the features: (I) only down-down time; (II) only up-down time; (III) only down-up time; (IV) down-down and up-down times; (V) down-down and down-up times; (VI) up-down and down-up times; (VII) down-down, up-down and down-up times.

The experiments were performed using the adaptation mechanism in down-down and up-down times.

4.1 Results

Tables 1, 2 and 3 shows the results in each experiment presented. Table 1 shows the percentage of successfully legitimate user authentication. Table 2 shows the percentage of failed impostor user authentication. Table 3 shows the percentage of failed observer user authentication.

Table 1. Comparative results in ten experiments for successfully legitimate user authentication

User	Experiment Results (%)						
	I	II	III	IV	V	VI	VII
1	100	100	95	100	95	95	100
2	90	95	100	95	85	95	95
3	100	100	100	100	100	100	100
4	100	80	100	75	100	75	85
5	100	100	100	100	100	100	100
6	95	100	100	95	95	100	95
7	100	100	100	100	100	100	100
8	95	100	100	100	95	95	95
9	90	85	100	90	95	100	95
10	95	100	100	100	90	100	100
Average (%)	96,5	96	99,5	95,5	95,5	96	96,5

Table 2. Comparative results in ten experiments for failed impostor user authentication

User	Experiment Results (%)						
	I	II	III	VI	V	VI	VII
1	0	0	0,3	0	0	0	0
2	39,6	30,4	27,6	0	16	0,8	0
3	43,9	40,8	8,6	26,6	0	0	0
4	60,8	55,1	50,9	43,7	34,2	31,9	15,2
5	43,4	25,2	63,1	8,3	26,7	11,3	4,1
6	37,1	12,5	74,1	7,4	27,9	11,8	3,9
7	0	1,2	45,1	0	0	0,3	0
8	28,3	18,1	28,3	3,9	10,2	5,1	1,5
9	36,4	34,7	30,8	19,3	5,5	13,8	1,9
10	33,8	13,3	47,5	10,2	20,9	11,1	5,3
Average (%)	32,4	23,2	37,8	12,1	14,3	8,7	3,2

Table 3. Comparative results in ten experiments for failed observer impostor user authentication

User	Experiment Results (%)						
	I	II	III	IV	V	VI	VII
1	0	0	5,5	0	0	0	0
2	33,3	66,6	16,6	0	11,1	0	0
3	44,4	44,4	11,1	38,8	0	0	0
4	61,1	55,5	50	44,4	38,8	38,8	22,2
5	50	22,2	66,6	11,1	27,7	11,1	5,5
6	50	27,7	88,8	22,2	44,4	27,7	16,6
7	0	11,1	38,8	0	0	11,1	0
8	38,8	33,3	27,7	16,6	5,5	5,5	5,5
9	33,3	22,2	22,2	33,3	50	11,1	11,1
10	33,3	11,1	44,4	11,1	11,1	11,1	11,1
Average (%)	34,4	29,4	37,2	17,7	18,8	11,6	7,2

As noted in Tables 1 and 2, the best results were achieved in the experiment VII, combining all the features and the fuzzy logic classifier. In this experiment, a 96,5% successfully legitimate user authentication was obtained, resulting in a false rejection rate (FRR) of 3,5%. The false acceptance rate (FAR) obtained was 3,2% using the simple impostor's samples and increased to 7,2% when using the observer impostor's samples in the same experiment, as shown in the table 3.

In next subsections, some aspects in methodology were analyzed to verify its impacts in results.

4.1.1 Key Code Feature

The key code feature is only different between users when the target string contains capital letters. Its impact in the results was really noticed just in one target string, besides in this case 70% of impostor's attempts were detected using only this feature. In the other cases, its impact was sporadic. In the observer impostor's attempts, this feature is not significant, because the keys are previously known.

4.1.2 Familiarity of the Target String

The experiment VII was repeated, changing the target string, to observe the influence of its familiarity to the users. The chosen target string was substituted by the imposed one. A 59,5% successfully legitimate user authentication was obtained, resulting in a FRR of 41,5% was achieved. Table 4 shows the percentage of successfully legitimate user authentication in familiarity of the target string.

4.1.3 One-Trial Session

In authentication session, each user has two attempts to be authenticated. The impact of conferring just one trial in each session was performed. It was obtained an 82% successfully legitimate user authentication, resulting in a FRR of 18%. Table 4 shows the percentage of successfully legitimate user authentication in one-trial session.

4.1.4 Adaptation Mechanism

The adaptation mechanism tries to maintain the templates updated. To verify its efficacy the experiment VII was repeated without this mechanism. A 92% successfully legitimate user authentication was obtained, resulting in a FRR of 8%, while the FAR obtained was 2,3%. Table 5 shows the percentage of successfully legitimate and failed impostor user authentication in this situation.

4.1.5 Timing accuracy

A lower timing accuracy of 0,01 millisecond was analyzed to verify its impact to the results. A 92% successfully legitimate user authentication was obtained, resulting in a 8% FRR, while the FAR obtained was 3,6%. The results can be visualized in Table 5.

Table 4. Results of successfully legitimate user authentication in two situations: the imposed target string and one-trial session

User	Familiarity (%)	One-Trial Session (%)
1	70	90
2	90	75
3	05	100
4	00	50
5	80	95
6	80	95
7	85	95
8	15	65
9	85	60
10	85	95
Average (%)	59,5	82

Table 5. Results of successfully legitimate and failed impostor user authentications without adaptation and in 0.01 timing accuracy

User	Without Adaptation		0.01 Timing Accuracy	
	Legitimate (%)	Impostor (%)	Legitimate (%)	Impostor (%)
1	90	0	90	0
2	90	0	90	0
3	90	0,3	100	0
4	100	11,4	55	17,4
5	100	2,6	100	3,3
6	85	1,5	100	3,9
7	85	0,3	100	0
8	95	0,7	95	1,5
9	90	1,1	90	4,3
10	95	4,1	100	5,3
Average (%)	92	2,3	92	3,6

4.1.6 Amount of Samples

The influence of the amount of samples used in the enrollment was also analyzed. Fig. 4 shows the behavior of FRR and FAR with the amount of samples. As noted in Fig. 4, the FRR is almost unchanged until eight samples, but it increases after seven samples. Fig. 4 shows that the FAR is reduced as the amount of samples decreases, but a little increase occurs in nine samples. Then, eight samples produce a significant result compared to ten samples initially adopted in this paper.

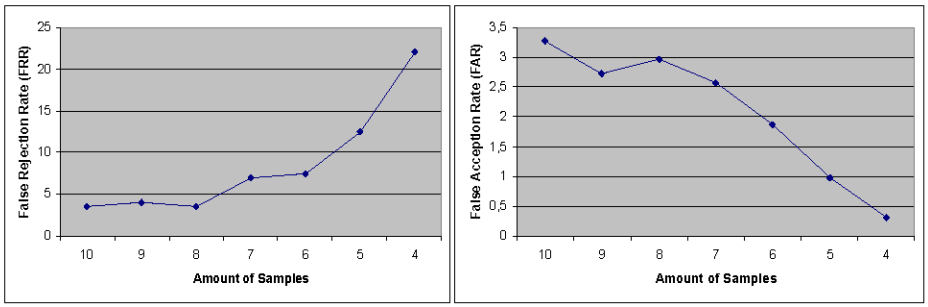


Fig. 4. The variation of the FRR and the FAR with the amount of samples

4.2 Discussion

The following observations were made according to the experiments realized:

- Even if an impostor observes a legitimate user session, this does not mean that the impostor will be authenticated.
- The choice of a target string with capital letters, which combines shift and caps lock keys, increases the difficulty of authentication of an impostor user.
- The familiarity of the target string to the user has a significant impact.
- A two-trial session reduces significantly the FRR.
- The adaptation mechanism decreases the FRR significantly and increases the FAR not much.
- A higher timing accuracy has best results in both rates (FRR and FAR).
- The amount of samples used in the enrollment could be reduced to eight samples with the same FRR (3,5%) and with a reduction of the FAR (2,9%).
- As noted in Table II, the user 4 has a discrepancy to the other users. After an analysis of the data, it was found out that this user has a high variance in the time-features; probably because of the target string chosen that was not too familiar to him. If the FRR is calculated without this user, the FRR is reduced to 1,8%.
- Table 6 shows a comparative resume of some researches and ours in typing biometrics. The 3,5% FRR obtained is superior to the one achieved (2%) in [12], but fifteen samples were used in the enrollment and its FAR (6%) is superior to our rate. The 2,9% FAR achieved in [10] is below 1%, but its FRR (13,3%) is a lot superior to our rate, and, four different strings were used as target. Finally, in [5], a fuzzy logic approach was used and a 7,4% FRR and a 2,8% FAR were obtained.

This FRR is superior and the FAR is almost equal to ours, besides it was used two different strings as target.

Table 6. Comparative resume of some researches and ours

Research	Amount of Samples	Target String	FRR	FAR
de Ru and Elof [5]	Varies depending of the user	Two strings (user login and password)	7,4%	2,8%
Joyce and Gupta [10]	Eight samples	Four strings (first name, last name, login and password)	13,3%	0,17%
Haidar et al. [12]	Fifteen samples	One string	2%	6%
Our research	Eight samples	One string	3,5%	2,9%

5 Conclusion

This paper presents a user authentication through typing biometrics features to improve usual login-password authentication, applying it in login and/or password. Some experiments were conducted, and the best performance was achieved using a fuzzy logic classifier and four features (key code, down-down time, up-down time and down-up time), obtaining a 3,5% FRR and a 2,9% FAR. These rates, as discussed in result section, are both competitive to other researches, using just one target string and eight samples in the enrollment.

This paper shows the good influence of: the familiarity of the target string, the two-trial session, the adaptation mechanism, and the timing accuracy. The FRR was reduced in 37%, 14,5%, 4,5% and 4,5%, analyzing each case, respectively.

For future work, we intend to increase our user population and to extend the methodology to numeric keyboard used in access control of restrict areas and in banking transactions. Another intention is to adapt the methodology to create a key based on this biometric technology (a typing bio-key) for cryptography purposes.

References

1. Matyas Jr, S.M., Stapleton, J.: A Biometric Standard for Information Management and Security. Computers & Security, Vol. 19 (2000) 428-441
2. Polemi, D.: EU Commission Final Report: Biometric Techniques: Review and Evaluation of Biometric Techniques for Identification and Authentication, Including an Appraisal of the areas where they are Most Applicable. Institute of Communication and Computer Systems, National Technical University of Athens (1997)
3. Monrose, F., Rubin, A.D.: Keystroke Dynamics as a Biometric for Authentication. Future Generation Computer Systems, Vol. 16, No. 4 (2000) 351-359
4. Monrose, F., Reiter, M.K., Wetzel, S.: Password Hardening Based on Keystroke Dynamics. 6th ACM Conference on Computer Security (1999)

5. de Ru, W.G., Eloff, J.H.P.: Enhanced Password Authentication through Fuzzy Logic. IEEE Expert / Intelligent Systems & Their Applications, Vol. 17, No. 6 (1997) 38-45
6. Robinson, J.A., Liang, V.M., Michael, J.A., MacKenzie, C.L.: Computer User Verification Login String Keystroke Dynamics. IEEE Trans. Syst., Man, Cybern., Vol. 28, No. 2 (1998) 236-241
7. Lin, D.T.: Computer-Access Authentication with Neural Network Based Keystroke Identity Verification. International Conference on Neural Networks, Vol. 1 (1997) 174-178
8. Obaidat, M.S., Sadoun, B.: Verification of Computer User Using Keystroke Dynamics. IEEE Trans. Syst., Man, Cybern., Vol. 27, No. 2 (1997) 261-269
9. Coltell, O., Badfa, J.M., Torres, G.: Biometric Identification System Based in Keyboard Filtering. Proceedings IEE 33rd Annual 1990 International Carnahan Conference on Security Technology, (1999) 203-209
10. Joyce, R., Gupta, G.: Identity Authentication Based on Keystroke Latencies. Communication of ACM, Vol. 33, No. 2 (1990) 168-176
11. Bleha, S., Slivinsky, C., Hussain, B.: Computer-Access Security Systems Using Keystroke Dynamics. IEEE Trans. Pattern Anal. Machine Intell., Vol. 12, No. 12 (1990) 1217-1222
12. Haidar, S., Abbas, A., Zaidi, A.K.: A Multi-Technique Approach for User Identification through Keystroke Dynamics. IEEE Int. Conference of Syst., Man and Cybern., Vol. 2 (2000) 1336-1341
13. Wong, F.W.M.H., Supian, A.S.M., Ismail A.F., Kin, L.W., Soon, O.C.: Enhanced User Authentication through Typing Biometrics with Artificial Neural Networks and K-Nearest Neighbor Algorithm. Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers, Vol. 2 (2001) 911-915
14. Bleha, D., Obaidat, M.: Dimensionality Reduction and Feature Extraction Applications in Identifying Computer Users. IEEE Trans. Syst., Man, Cybern., Vol. 21 (1991) 452-456
15. Intel Corporation: Using the RDTSC Instruction for Performance Monitoring. (1997) (available on-line at http://developer.intel.com/drg/pentiumII/appnotes/RDTSCPM1.HTM

A.3 Artigo ICBA-04

User Authentication through Typing Biometrics Features

Lívia C. F. Araújo[1], Luiz H. R. Sucupira Jr.[1], Miguel G. Lizárraga[1], João B. T. Yabu-uti[1]

[1]School of Electrical and Computer Engineering, State University of Campinas
Albert Einstein Avenue, 400, PO Box 6101, Postal Code 13083-970, Campinas, SP, Brazil
{liviacri, luigijr, lizarrag, yabuuti} @decom.fee.unicamp.br

Abstract. This paper uses a static typing biometrics in user authentication. The inputs are the key down and up times and the key ASCII codes captured while the user is typing a string. Four features (key code, two keystroke latencies and key duration) were analyzed, and, seven experiments were performed combining these features. The results of the experiments were evaluated involving three types of user: the legitimate, the impostor and the observer impostor users. The best results were achieved utilizing all features, obtaining a false rejection rate (FRR) of 1.45% and a false acceptance rate (FAR) of 1.89%. This approach can be used to improve the usual login-password authentication when the password is no more a secret. This paper innovates using the combination of four features to authenticate users. **Keywords** - typing biometrics, biometrics, pattern recognition, authentication

1 Introduction

The control access is a very important issue in the computer systems. The login-password authentication is the most usual mechanism used to grant access. This authentication is fragile when there is a careless user and/or a weak password, however, it is low-cost and familiar to the users. The purpose of this paper is to improve the login-password authentication using biometric characteristics. Biometric characteristics are defined as behavioral or physiological characteristics that distinguish one person from another [1]. Recent researches suggest that the inclusion of biometric characteristics in automated systems for personal recognition increase the trustworthiness degree. This increase occurs because of biometric characteristics are unique to each person and could not be stolen, lost or forgotten [2].

The biometric technology employed in this paper is the typing biometrics, also known as keystroke dynamics. The typing biometrics is the process of analyzing the way a user types at a terminal by monitoring the keyboard inputs in attempt to identify users. The typing biometrics authentication can be classified as static or continuous. The static approach analyses the inputs just in a particular moment, and the continuous one analyses the inputs during all user's session [3].

This work was partially supported by CNPq, CAPES and FAPESP.

The methodology of this paper is low-cost and unintrusive, and, uses a verification authentication in a static approach.

This paper is organized as follow: In section 2, there is a resume of related studies published; in section 3, the methodology is explained; in section 4, the results are presented; and finally, in the section 5, there are the conclusions and the future works.

2 Related Work

Some researches were published [3]-[13] in the authentication via typing biometrics since 1990. Some aspects presented in these works are resumed below:

- *Target string*: It is the input string that will be typed. In [9], four strings were used as target. Two target strings were analyzed in [10]: a 31-char string and a login. In [8], target strings were divided in three levels based on a difficulty degree. String length is a very important issue, considering that in [12] was stated that misclassification increases when the string contains fewer than ten characters;
- *Amount of Samples (Training Set)*: Samples are collected during the enrollment process to compound the training set. Its amount varies a lot, it was 3 samples per users in [7] and, 30 samples per user in [10]. In [3], it was concluded that fewer than six samples is not recommended to obtain good results in performance;
- *Features*: A good feature has to be highly repeatable in the same user and different between users [4]. Two most used features is *duration of the key*, that is the time interval that a key remain pressed [7], and *keystroke latency*, that is the time interval between successive keystrokes [7]. In [6] and [13] the combination of these features brought better results than using them isolated;
- *Timing accuracy*: As the most of the typing biometrics features are time-based, the precision of the key up and the key down times have to be analyzed. The timing accuracy in the researches varies between 0.1 millisecond [6] and 1 second) [8].
- *Adaptation mechanism*: Biometric characteristics change over time. An adaptation or a re-enrollment could be performed to maintain the templates updated. The majority of the researches did not mention this issue, but, in [4], an adaptation mechanism was activated every time a successful authentication was performed, creating a new template updated with the new sample and without the oldest one.
- *Classifier*: In [3], [6], [8]-[10] and [12], a statistical classifier was applied, using known techniques as k-means, Bayes, etc. In [5] and [13], fuzzy logic was applied. In [11], a statistical, a neural, and a fuzzy classifier were combined.

3 Methodology

Each time a user tries to access a system, he indicates an account *a*, and, types the target string. This string is chosen by the user and its length has to be of at least 10 characters. While the user is typing, *keystroke data* is captured in a 1 millisecond timing accuracy and, a *sample* is created containing the *features* calculated using this data. If the account is new, then 10 (ten) samples will compound the training set, and,

a *template* is created. A sample will only be stored if the *key code* feature match. If the account already exists, then a sample will be compared with the account's template. According to the *classifier decision*, the authentication will be successful or not. In an authentication session a user has two trials to be authenticated. If the user's identity is validated, then he could access the system and an *adaptation* mechanism could be called to compute a new template update.

The main issues related to the methodology are described in the next sub-sections.

3.1 Keystroke Data

A string with m characters will result in n keystrokes, where $m \leq n$, since some characters need more than one keystroke (e.g. "*X*" needs "*x*" and "*shift*" keys). $K_{a,w} = \{ k_1(a,w), \ldots, k_n(a,w)\}$ is the keystroke data captured in the sample w in the account a.

Each keystroke $k_i(a,w)$ is compound of the key down time $t_{i.down}(a,w)$ (the instant when the key is pressed), the key up time $t_{i.up}(a,w)$ (the instant when the key is released) and the key code $c_i(a,w)$ (the ASCII code).

3.2 Features

Features are calculated using keystroke data. Four features were analyzed in this paper:

- Key Code: the ASCII code that represents each key. When a string contains capital letters, there are more than one possible set of key codes, otherwise there is a single possible set of key codes. The key codes contained in the template of the account a are represented as $C_a = \{c_1(a), \ldots, c_n(a)\}$ and, the key codes contained in the sample w in the account a is represented as $C_{a,w} = \{c_1(a,w), \ldots, c_n(a,w)\}$.
- Down-Down Time: a keystroke latency defined as the time interval between successive keystrokes [7]. This feature is represented as $DD_{a,w} = \{dd_1(a,w), \ldots, dd_n(a,w)\}$ where $dd_i(a,w) = t_{i+1.down}(a,w) - t_{i.down}(a,w)$ is related to (k_i, k_{i+1}).
- Up-Down Time: a keystroke latency feature, and, is represented as $UD_{a,w} = \{ud_1(a,w), \ldots, ud_{n-1}(a,w)\}$ where $ud_i(a,w) = t_{i+1.down}(a,w) - t_{i.up}(a,w)$ is related to (k_i, k_{i+1}).
- Down-Up Time: the duration of key that is defined as the time interval that a key remain pressed [7]. This feature is represented as $DU_{a,w} = \{du_1(a,w), \ldots, du_n(a,w)\}$ where $du_i(a,w) = t_{i.up}(a,w) - t_{i.down}(a,w)$ is related to k_i.

The combination of these features is novel, previous researches used at most two features.

3.3 Template

The template contains the key code itself C_a, and, the mean $\mu(feat_i(a))$ and standard deviation $\sigma(feat_i(a))$ *of* each element i of each feature *feat* (*DD*, *DU* or *UD*).

3.4 Classifier

The sample *w* of the account *a* is analyzed by the classifier. $C_{a,w}$ is compared with C_a: if they are different, the authentication fails, otherwise for each feature *feat*, a distance between the template and the sample is calculated by (1):

$$D_{feat}(a,w) = \frac{1}{n}\sum_{i=1}^{n} d_i(a,w) \quad \cdot \tag{1}$$

where *n* is the total of elements of the feature *feat* and $d_i(a,w)$ is the distance related to each element *i* between the template and the sample, and, is given by (2):

$$d_i(a,w) = \frac{feat_i(a,w) - \mu(feat_i(a))}{\sigma(feat_i(a))} \tag{2}$$

Finally, the authentication will be successful if the condition (3) is satisfied:

$$(D_{dd}(a,w) \leq T_{dd}(a)) \text{ and } (D_{du}(a,w) \leq T_{du}(a)) \text{ and } (D_{ud}(a,w) \leq T_{ud}(a)) \tag{3}$$

where $T_{dd}(a)$, $T_{du}(a)$ and $T_{ud}(a)$ are the thresholds for the down-down, down-up, and up-down, respectively in the account *a*.

An analysis was performed in real data and, it was observed that a feature with a higher variation demands a lower threshold, meanwhile a feature with a lower variation demands a higher threshold. So, the threshold for each feature in each account is obtained based on its standard deviation.

3.5 Adaptation Mechanism

The adaptation mechanism consists of creating an updated template, including a new sample and discarding the oldest one. This mechanism is performed with a successful sample *w* if the majority elements *i* of the time-features *feat* satisfy the condition (4).

$$\left(d_{feat_i}(a,w) \leq T_{feat}(a)\right) \tag{4}$$

As the adaptation mechanism is activated, the standard deviation for each feature is modified and also the thresholds are modified.

4 Experiments

The experiments were conducted in three machines (a laptop and two PCs) with two different keyboard layouts, and, 30 (thirty) users were enrolled and participated of the experiments in three situations of authentication:

- Legitimate user: the users tried to be authenticated in their accounts.
- Impostor user: the users tried to be authenticated in other user's accounts, knowing the string typed by their owners.
- Observer impostor user: the users observed how the other user types their strings, then they tried to be authenticated in their accounts.

Seven experiments were analyzed, combining the features: (I) *DD*; (II) *UD*; (III) *DU*; (IV) *DD* and *UD*; (V) *DD* and *DU*; (VI) *UD* and *DU*; (VII) *DD, UD and DU.*

4.1 Results

The performance of biometrics systems are generally measured by two rates [1]:

- False Acceptance Rate (FAR): the probability that the system will fail to reject an impostor user.
- False Rejection Rate (FRR): the probability that the system will fail to verify the legitimate user claimed identity.

Table 1 shows the FRR for legitimate users and FAR for impostor and observer impostor users.

Table 1. Experiments comparative results for legitimate, impostor and observer impostor users

Experiment		(I)	(II)	(III)	(IV)	(V)	(VI)	(VII)
Legitimate	Sessions	553	553	553	553	553	553	553
	Errors	9	12	13	12	7	9	8
	FRR	1.63	2.17	2.35	2.17	1.27	1.63	1.45
Impostor	Sessions	2916	2916	2916	2916	2916	2916	2916
	Errors	580	179	795	151	163	91	55
	FAR	19.90	6.14	27.26	5.18	5.59	3.12	1.89
Observer Impostor	Sessions	492	492	492	492	492	492	492
	Errors	144	58	166	46	48	34	18
	FAR	29.27	11.79	33.74	9.35	9.75	6.91	3.66

As noted in table 1, the best results were achieved in the experiment (VII). Some particular aspects in the methodology were analyzed in this experiment to verify their impacts in the results:

- Key Code Feature: Its impact in the results was really noticed in target strings that contain capital letters (70% of impostor's sessions were detected). Thus, the choice of a target string with capital letters, which combines shift and Caps Lock keys, increases the difficulty of authentication of an impostor user.
- Familiarity of the Target String: The second column of the table 2 shows the results achieved using an imposed string ("unicamp@003"). The FRR increased, showing that the familiarity of the target string to the user has a significant impact.
- One-Trial Authentication: The last column of table 2 shows the results achieved conferring just one trial. The FRR increased, showing that a user must have two trials to be authenticated.

Table 2. Results achieved in a imposed string and in a one-trial authentication

Experiment (VII)	Imposed String	One-Trial
Sessions	533	553
Errors	92	64
FRR	17.26%	11.57%

– Adaptation Mechanism: Table 3 shows the results without this mechanism. Both rates ARE increased, showing that an adaptation mechanism must be performed.

Table 3. Results achieved without adaptation

Experiment (VII)	Legitimate User	Impostor User
Sessions	553	2916
Errors	23	137
Rates	4.16% FRR	4.70% FAR

– Timing Accuracy: Table 4 shows the results in a lower timing accuracy of 10 milliseconds Both rates are increased, showing that a 1 millisecond timing should be applied.

Table 4. Results achieved in a lower timing accuracy

Experiment (VII)	Legitimate User	Impostor User
Sessions	553	2916
Errors	9	110
Rates	1.63% FRR	3.77% FAR

– Amount of Samples: Figure 1 shows the behavior of FRR and FAR with the amount of samples used in enrollment. With the reduction of the amount of samples, FRR increases and FAR varies a little.

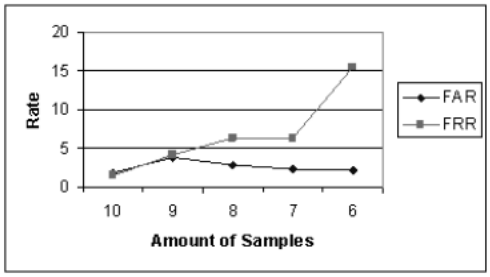


Fig. 1. The variation of the rates with the amount of samples

The rates obtained are both competitive with the published work as observed in Table 5. In this table there is a resume of some researches in keystroke dynamics including a previous one based on fuzzy logic conducted by us [13].

Table 5. A resume of some researches and ours

Research	Samples	Target String	FRR	FAR
de Ru and Elof [5]	Varies	Two	7.4%	2.8%
Joyce and Gupta [9]	Eight	Four	13.3%	0.17%
Haidar et al. [11]	Fifteen	One	2.0%	6.0%
Araújo et al. [13]	Eight	One	3.5%	2.9%
Our Research	Ten	One	1.45%	1.89%

5 Conclusion

This paper presents a methodology through typing biometrics features to improve the usual login-password authentication. Some experiments were conducted and the best performance was achieved using a statistical classifier based on distance and the combination of four features (key code, down-down, up-down and down-up times), obtaining a 1.45% FRR and a 1.89% FAR. These rates, as discussed in the results section, are both competitive with other researches, using just one target string and ten samples in enrollment. This work innovates using four features to authenticate users.

For future work, we intend to increase our user population and to extend the methodology to numeric keyboard as the ones used in banking transactions. Another intention is to adapt the methodology for cryptography purposes, generating a cryptographic key using the keystroke dynamics features (a typing biometric-key).

References

1. Matyas Jr, S.M., Stapleton, J.: A Biometric Standard for Information Management and Security. Computers & Security, Vol. 19 (2000) 428-441
2. Polemi, D.: EU Commission Final Report: Biometric Techniques: Review and Evaluation of Biometric Techniques for Identification and Authentication, Including an Appraisal of the areas where they are Most Applicable. Institute of Communication and Computer Systems, National Technical University of Athens (1997)
3. Monrose, F., Rubin, A.D.: Keystroke Dynamics as a Biometric for Authentication. Future Generation Computer Systems, Vol. 16, No. 4 (2000) 351-359
4. Monrose, F., Reiter, M.K., Wetzel, S.: Password Hardening Based on Keystroke Dynamics. 6th ACM Conference on Computer Security (1999)
5. de Ru, W.G., Eloff, J.H.P.: Enhanced Password Authentication through Fuzzy Logic. IEEE Expert / Intelligent Systems & Their Applications, Vol. 17, No. 6 (1997) 38-45
6. Robinson, J.A., Liang, V.M., Michael, J.A., MacKenzie, C.L.: Computer User Verification Login String Keystroke Dynamics. IEEE Trans. Syst., Man, Cybern., Vol. 28, No. 2 (1998) 236-241
7. Lin, D.T.: Computer-Access Authentication with Neural Network Based Keystroke Identity Verification. International Conference on Neural Networks, Vol. 1 (1997) 174-178
8. Coltell, O., Badfa, J.M., Torres, G.: Biometric Identification System Based in Keyboard Filtering. Proceedings IEE 33rd Annual 1990 International Carnahan Conference on Security Technology, (1999) 203-209
9. Joyce, R., Gupta, G.: Identity Authentication Based on Keystroke Latencies. Communication of ACM, Vol. 33, No. 2 (1990) 168-176
10. Bleha, S., Slivinsky, C., Hussain, B.: Computer-Access Security Systems Using Keystroke Dynamics. IEEE Trans. Pattern Anal. Machine Intell., Vol. 12, No. 12 (1990) 1217-1222
11. Haidar, S., Abbas, A., Zaidi, A.K.: A Multi-Technique Approach for User Identification through Keystroke Dynamics. IEEE Int. Conference of Syst., Man and Cybern., Vol. 2 (2000) 1336-1341
12. Bleha, D., Obaidat, M.: Dimensionality Reduction and Feature Extraction Applications in Identifying Computer Users. IEEE Trans. Syst., Man, Cybern., Vol. 21 (1991) 452-456
13. Araújo, L.C.F., Sucupira Jr., L.H.R., Lizárraga, M.G., Ling, L.L., Yabu-uti, J.B.T.: A fuzzy Logic Approach in Typing Biometrics User Authentication. Proc. First Indian International Conference on Artificial Intelligence, (2003) 1038-1051