http://journal.unoeste.br/index.php/ca/index
DOI: 10.5747/ce.2019.v11.n2.e272
ISSN on-line 2178-8332

Colloquium Exactarum

Submetido: 10/12/2018 Revisado: 05/03/2019 Aceito: 18/06/2019

AUTENTICAÇÃO BIOMÉTRICA PARA SISTEMAS POR MEIO DA DINÂMICA DA DIGITAÇÃO

BIOMETRIC AUTHENTICATION FOR SYSTEMS THROUGH DIGITAL DYNAMICS.

Rafael Junior Henrique Silva[1]; Mário A. Pazoti[1]; Francisco Assis da Silva[1]; Danillo Roberto Pereira[1], Leandro Luiz de Almeida[1]

[1]Universidade do Oeste Paulista (UNOESTE)
Faculdade de Informática de Presidente Prudente (FIPP)
E-mail: rafaeljhs50@gmail.com, {mario, chico, danilopereira, llalmeida}@unoeste.br

RESUMO – A autenticação biométrica por meio da dinâmica da digitação, oferece uma segurança a mais para os sistemas, se associando ao tradicional *login* e senha, pois associa autenticação com fatores biométricos. Esse trabalho apresenta metodologias para a realização da autenticação biométrica po meio da dinâmica da digitação, utilizando técnicas para a extração de características após capturar o tempo em que se pressionar e soltar as teclas. Os métodos utilizados foram: Classificador baseado na distância de Mahalanobis (classificação e autenticação) e Floresta Aleatória (classificação).
Palavras-chave: Floresta Aleatória; Mahalanobis; Dinâmica da digitação; Pressiona e Pressiona; Pressiona e Solta; Solta e Pressiona, *Threshold*.

ABSTRACT – Biometric authentication through typing dynamics offers security for more systems, coupled with the traditional login and password, as it associates authentication with biometric factors. This work presents methodologies for biometric authentication performance from the middle of the typing dynamics, techniques for extracting characteristics after capturing the time in which to press and release as keys. The methods used were: Mahalanobis distance classifier (classification and authentication) and Random Forest (classification).
Keywords: Random Forest; Mahalanobis; Dynamic typing; Press and Press; Press and Release; Release and press, Threshold.

1. INTRODUÇÃO

A autenticação pessoal é um recurso para proteger informações em um sistema, sendo mais comum utilizar algo que a pessoa conhece como senha ou cartão magnético. Segundo Costa *et al.* (2005), mesmo sendo uma forma de autenticação bem disseminada, o uso de senhas e cartões apresenta vulnerabilidades e, por esse motivo, técnicas de autenticação baseadas em características biométricas físicas ou comportamentais estão sendo cada vez mais aplicadas.

Um dos pontos positivos em relação à técnica biométrica é que as pessoas apresentam características próprias, as quais não podem ser forjadas e esquecidas. Dentre as características comportamentais, a dinâmica da digitação é uma das técnicas que podem ser utilizadas para autenticação dos usuários, seja para verificação em provas realizadas on-line, por exemplo, ou como proteção adicional associada à senha tradicional.

Segundo Alsultan e Warwick (2013), a dinâmica da digitação é uma técnica relativamente barata quando comparada às técnicas biométricas já popularizadas no mercado, como a impressão digital. Ela necessita apenas de um teclado e do software para autenticação, diferente das outras técnicas biométricas que possuem elevado custo com os dispositivos de captação e análise dos dados necessários na autenticação. A partir do ritmo da digitação, podem ser extraídas algumas características relacionadas ao intervalo e usar estas características pessoais para identificar o indivíduo.

Este trabalho apresenta uma metodologia para o reconhecimento de padrões com base nas características extraídas durante a digitação de uma tecla e, desta forma a partir dessas características conseguir por meio de um classificador, identificar um usuário autêntico ou impostor. Para a etapa de classificação, foram utilizadas as técnicas de aprendizado de máquina Floresta Aleatória (BREIMAN ,2001) e classificador de distância padrão, baseado no modelo de Mahalanobis (ARAÚJO, 2004), usando como características: o tempo em que a tecla é pressionada, o tempo em que é liberada e a qual a tecla pressionada.

O artigo está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados; na Seção 3 é apresentado o referencial teórico utilizado na metodologia; na Seção 4 é demonstrada a metodologia para o desenvolvimento do trabalho; na Seção 5 são discutidos os experimentos e os resultados; na Seção 6 são mostradas as conclusões do trabalho.

2. TRABALHOS RELACIONADOS

Nesta seção são apresentados os trabalhos relacionados ao desenvolvimento deste trabalho, que se referem à etapa de coleta dos dados e das características extraídas.

No trabalho de Araújo (2004) são utilizadas cinco vetores de características extraídas dos valores em que a tecla é pressionada e liberada, que são: o primeiro vetor correspondente ao caracteres das teclas pressionadas na informação alvo, o segundo vetor pressiona-pressiona que contém a latência entre os pressionamentos das teclas sucessivas, o terceiro vetor solta-pressiona contém a latência entre a soltura de uma tecla e o instante do pressionamento da próxima tecla, o quarto vetor pressiona-solta contém o tempo que a tecla fica pressionada e o quinto vetor contém características extraídas dos segundo, terceiro e quarto vetores de características. Os cinco vetores extraídos são utilizados para construir um *template* do usuário que posteriormente é utilizadas nos classificadores, sendo um classificador nebuloso (*Fuzzy*) e o outro um classificador estatístico. Foi observado nos experimentos do referido trabalho 1.55% FRR e 1.91% FAR utilizando o classificador estatístico, sendo resultados muito significativo para área.

O trabalho de Cruz, Duarte e Goldschmidt(2017) utiliza características extraídas das amostras dos usuários que são previamente cadastrados. Na primeira etapa tem-se: o tempo de pressionamento das teclas e o tempo da latência entre teclas; na segunda etapa é feita a construção de atributos como média e o desvio padrão das caracterizas extraídas na primeira etapa; na terceira etapa os atributos são normalizados para serem posteriormente utilizados em algoritmo de aprendizado de máquina, que resultará no melhor algoritmo para o grupo de dados. Os algoritmos utilizados no referido trabalho são: K-Vizinhos Mais Próximos (K-NN), Centroide Mais Próximo, Árvores de Decisão, algoritmo Floresta Aleatória, algoritmo de Máquina de Vetores de Suporte (SVM), Classificador Bayesiano Ingênuo e as Redes Neurais Artificiais. Esse trabalho obteve desempenho próximo de 92,44%. Entretanto mesmo abaixo dos índices de acerto de outros trabalhos na área, tem-se uma vantagem sobre os demais, pois as amostras são obtidas com textos livres, diferentes dos outros trabalhos, em que os textos são fixos.

3 REFERENCIAL TEÓRICO

Nesta seção é apresentado o referencial teórico utilizado neste trabalho

3.1. Classificador baseado na distância de Mahalanobis (CBDM).

Segundo Araújo (2004) o uso desse classificador corrigi algumas das limitações da distância Euclidiana pois, ajusta automaticamente a escala dos eixos das coordenadas e leva em consideração a correlação entre as características. Porém segundo Liden (2009) ele apresenta algumas desvantagens: as matrizes de covariância podem ser difíceis de determinar e a memória e o tempo de computação crescem de forma quadrática com o número de características.

Expressada pela forma (1):

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

Onde m_X representa o vetor de características médio, C_X é a matriz de covariância para o vetor de características representado por X. As superfícies sendo r a constante composta por elipsoides que estão centradas em m_X.

3.1.1 *Threshold*

A função de *Threshold,* utiliza um valor liminar para determinar se uma amostra é verdadeira ou não. Sendo que a determinação do valor de liminar é significativa para o desempenho do sistema biométrico (Araújo et al., 2004).

As funções utilizadas, foram feitas em função da média do desvio padrão das características PP, SP e PS:

$$T_{PP}(\sigma) = \begin{cases} 2.5, & \sigma \leq 17 \\ (74.5 - \sigma)/23, & 17 < \sigma < 40 \\ 1.5, & \sigma \geq 40 \end{cases}$$

$$T_{SP}(\sigma) = \begin{cases} 2.5, & \sigma \leq 13 \\ (55.5 - \sigma)/17, & 13 < \sigma < 30 \\ 1.5, & \sigma \geq 30 \end{cases}$$

$$T_{PS}(\sigma) = \begin{cases} 2.5, & \sigma \leq 5 \\ (30 - \sigma)/10, & 5 < \sigma < 15 \\ 1.5, & \sigma \geq 15 \end{cases}$$

3.2. Floresta Aleatória.

Esse método de classificação é baseado no algoritmo de árvore de decisão padrão que o conjunto de dados é utilizado para fazer a árvore, já no algoritmo de floresta aleatória, o conjunto de dados é dividido aleatoriamente em vários subconjuntos de tamanho menor (HAN *et al.,* 2011).

Cada uma das árvores de decisão é criada com base no subconjunto. A criação das arvores é realizado por meio de seleção de atributos aleatória dos subconjuntos, sendo que a floresta aleatória é formada por uma coleção de arvores de decisão.

Após a formação da floresta, tem-se um grande número de árvores de decisão para serem testadas. Cada arvore contribui para a classificação da amostra, por meio de um peso, sobre qual classe a amostra deve pertencer. O peso de cada árvore é relacionado com a sua precisão, sendo que: quanto mais precisa uma arvore for maior o seu peso. (HAN et al., 2011).

4 METODOLOGIA

4.1 Obtenção de Amostra

As amostras utilizadas nesse trabalho foram obtidas durante o ano de 2017, sendo de dez frases distintas. Com base no trabalho de Joyce e Gupta (1990) e Araújo (2004), a quantidade de amostras de cada usuário necessária para a autenticação são sete e segundo Bleha e Obaidat (1991) e Araújo (2004) as amostras têm que conter mais que 11 caracteres. As frases escolhidas para o trabalho são: “amizade verdadeira”, “teclado bonito”, “sou todo dia”, “vontade de viver”, “o sol amarelo”, “senha pessoal”, “fipp 30 anos”, “apagador azul”, “quadro negro” e “computador preto”.

As amostras foram obtidas pelo programa de obtenção das amostras, utilizando um teclado de computador e um computador, todo momento que uma tecla era pressionada e soltada eram salvos os tempos em milissegundos por meio da função “*Clock.systemDefaultZone().millis()*” como na tabela abaixo exemplificando uma amostra de “senha pessoal”:

Tabela 1. Exemplo de amostra.

Caracter e	***k.Tp (Pressiona)***	***k.Ts (Solta)***
s	1497974835184	1497974835281
e	1497974835376	1497974835480
n	1497974835432	1497974835512
h	1497974835592	1497974835681
a	1497974835744	1497974835856
‘ ’	1497974835801	1497974835864
p	1497974835984	1497974836081
e	1497974836105	1497974836200
s	1497974836345	1497974836417
s	1497974836505	1497974836608
o	1497974836576	1497974836664
a	1497974836697	1497974836833
l	1497974836801	1497974836904

Fonte: O autor.

4.2 Vetores de características

Dos dados brutos são obtidos 4 vetores de características;

Teclas Pressionadas (TP), o vetor correspondente as teclas pressionadas, um exemplo com base na Tabela 1 é: TP’= {s, e, n, h, a, ’ ‘, p, e, s, s, o, a, l}

Pressiona e Pressiona (PP), o vetor que contém a latência entre o pressionamento de teclas sucessivas, sendo *PP’*= {pp_1, pp_2, … pp_n} onde $pp_i = k_{i+1}.tp - k_i.tp$, exemplo com base na Tabela 1:

pp_1 = 1497974835376-1497974835184=192

pp_2 = 1497974835432-1497974835376=56

…

pp_{12} = 1497974836801-1497974836697=104

$PP' = \{192, 56, 160, 152, 57, 183, 121, 240, 160, 71, 121, 104\}$

Pressiona e Solta (PS), o vetor que contém o tempo em que a tecla permanece pressionada, sendo *PS’*= {ps_1, ps_2, … ps_n} onde $ps_i = k_i.ts - k_i.tp$, exemplo com base na tabela 1:

ps_1 = 1497974835281-1497974835184 = 97

ps_2 = 1497974835480-1497974835376 = 104

…

ps_{13} = 1497974836904-1497974836801 = 103

$PS' = \{97, 104, 80, 89, 112, 63, 97, 95, 72, 103, 88, 136, 103\}$

Solta e Pressiona (SP), o vetor que contém a latência entre a soltura de uma tecla e o instante do pressionamento da próxima tecla, sendo *SP’ = {*sp_1, sp_2, …, sp_n} onde $ps_i = k_{i+1}.tp - k_i.ts$, exemplo com base na Tabela 1:

sp_1 = 1497974835376-1497974835281= 95

sp_2=1497974835432-1497974835480=-48

...

sp_{12}
$=$ 1497974836801-1497974836833= -32

SP'
$= \{95, -48, 80, 63, -55, 120, 24, 145, 88, -32, 33, -32\}$

Após o cálculo dos 3 vetores para cada uma das 7 amostras e feita a extração da média (Eq. 2) e o desvio padrão (Eq. 3) para os vetores de PP, PS e SP das 7 amostras.

$$\mu_x = \frac{1}{7} \sum_{j=1}^{7} x_i \quad (2)$$

$$\sigma_{x_i} = \sqrt{\frac{1}{7-1} \sum_{j=1}^{7} (x_i - \mu_x)^2} \quad (3)$$

Sendo que *x* corresponde a um dos 3 vetores e μ corresponde à média do vetor e σ desvio padrão.

Os valores da média e desvio padrão são novamente calculados, mas sem os valores que são considerados *outliers,* determinados pela expressão:

$$o_{s_j.x_i} \begin{cases} falso \ (\mu_{x_i} - 3\sigma_{x_i}) \leq x_i \leq (\mu_{x_i} + 3\sigma_{x_i}) \\ verdadeiro \end{cases} \quad (4)$$

$$\mu_x = \frac{1}{u} \sum_{j=1}^{7} x_i \,|\sigma_{s_j.x_i} = falso) \quad (5)$$

$$\sigma_{x_i} = \sqrt{\frac{1}{u-1} \sum_{j=1}^{7} ((x_i - \mu_x)^2 |\sigma_{s_j.x_i} = falso)} \quad (6)$$

Sendo u é a quantidade de valores que não são *outliers.* Exemplo:

$\{s_1.pp, \ s_2.pp, \ldots, \ s_7.pp\}$ cujos valores são: {156, 161, 138, 173, 185, 153, 170}

- Passo 1:
  Média = 1136/7 = 162.28
  Desvio Padrão = $\sqrt{1142.5/6} = 13.79$
- Passo 2:

Limite mínimo: 162.28 – (3*13.79) = 120.91

Limite máximo: 162.28 + (3*13.79) = 203.65

Não há *outliers*, pois não há valores fora do intervalo [120.91, 203.65].

Após essa etapa o usuário terá um *template* que é composto por 7 vetores $TP, \mu_{pp}, \sigma_{pp}, \mu_{sp}, \sigma_{sp}, \mu_{ps}, \sigma_{ps})$

- TP = Vetor das teclas pressionadas.
- μ_{pp}= Vetor com a média dos valores de pp das amostras.
- σ_{pp}= Vetor com o desvio padrão dos valores de pp das amostras.
- μ_{sp}= Vetor com a média dos valores de sp das amostras.
- σ_{sp}= Vetor com o desvio padrão dos valores de sp das amostras.
- μ_{ps}= Vetor com a média dos valores de ps das amostras.
- σ_{ps}= Vetor com o desvio padrão dos valores de ps das amostras.

Esses vetores vão ser utilizados pelos algoritmos classificadores.

5 EXPERIMENTOS E RESULTADOS

As técnicas de aprendizado de máquina utilizadas neste trabalho foram Floresta Aleatória e o Classificador baseado na distância de Mahalanobis, descritos na seção 3.1 e 3.2.

Foi amostrado um conjunto de 1800 amostras realizadas por 18 voluntários, sendo: dez frases idênticas (detalhadas na seção 4.1) digitadas dez vezes por cada voluntário.

A linguagem escolhida para o desenvolvimento do software foi Java, pois é uma linguagem que tem uma alta portabilidade e apresentava os recurso necessário para a aplicação. A biblioteca gráfica utilizada na aplicação foi JavaFX e para a utilização do método Floresta Aleatória a biblioteca WEKA. O ambiente de desenvolvimento utilizado foi NetBeans IDE 8.2.

A Figura 1 ilustra os passos para o cadastramento e autenticação de um voluntário, como um sistema que utiliza a validação biométrica com base na dinâmica da digitação tem que se seguir.

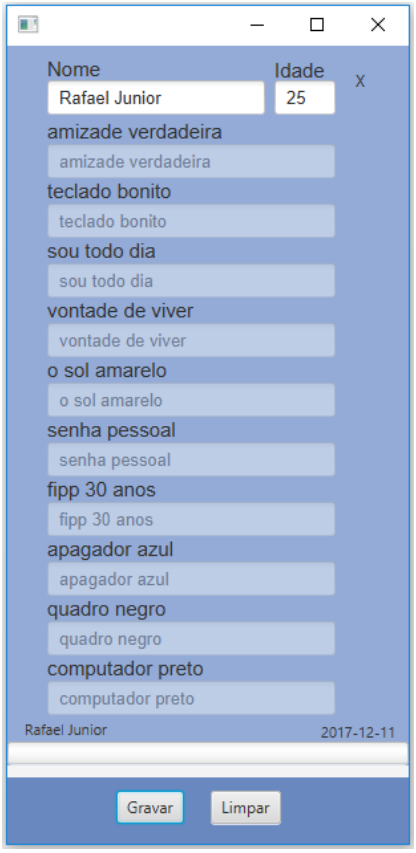
Figura 1. Fluxo de cadastro ou validação



Fonte: O autor.

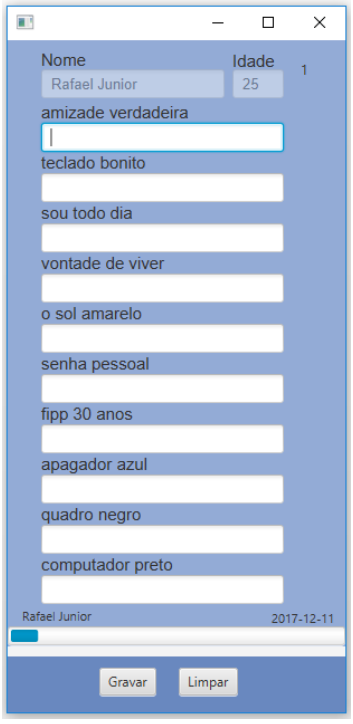
A tela de obtenção da amostra (Figura 2), apresenta as dez frases escolhidas, mais dois campos, onde o voluntário digitava seu nome e sua idade, utilizado apenas para registro. Cada campo era preenchido com a frase correspondente acima. Quando o voluntario pressionava ou soltava uma tecla um evento era executado, o evento tinha o objetivo de capturar o tempo em que a tecla era pressionada e/ou soltada, a função escolhida para a captura do tempo foi "*Clock.systemDefaultZone().millis()*" que obtém o instante em milissegundo do relógio. Após o usuário clicar em gravar as frases eram novamente habilitadas e limpas que os voluntários repetissem o processo, isso ocorria dez vezes (Figura 3).

Figura 2. Obtenção das amostras



Fonte: O autor.

Figura 3. Obtenção após uma digitação.



Fonte: O autor.

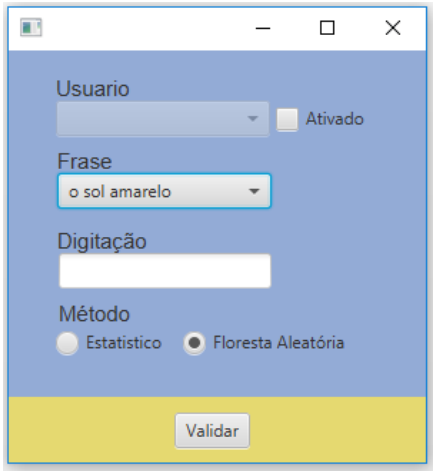
Após o processo de obtenção de amostras um arquivo texto era gerado para cada voluntário, o arquivo é formado pelo

nome do voluntário, data da obtenção das amostras, idade do voluntário e os tempos em milissegundo que as teclas foram pressionadas e soltadas.

O processo de classificação dos métodos inicia com a geração dos *template* (seção 4.1) de todos os voluntários cadastrados. Os *template* são salvo no formato ARFF (*Attribute-Relation File Format*), (HALL *et al.,* 2009) devido a necessidade de compatibilidade com a biblioteca WEKA. Após a geração dos *templates* o sistema já está pronto para as amostras serem classificada em relação aos voluntários já cadastrados.

Na Figura 4, mostra a tela onde o voluntário pode classificar sua amostra, onde ele digita uma vez a frase escolhida e o software apresenta o voluntario que pertence a amostra, sendo possível escolher o método de classificação que se deseja testar.

Figura 4. Validação de uma amostra.



Fonte: O autor.

Foram também realizado experimentos, para verificar a precisão dos métodos em relação a quantidade de voluntários e a quantidade de amostras utilizadas na geração do *template*.

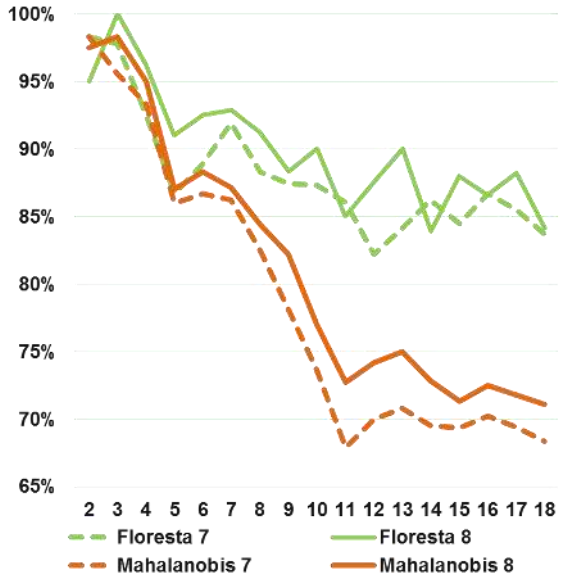
Na Figura 5 é mostrado um gráfico com a acurácia dos métodos de classificação com base no número de voluntários, sendo que nesse gráfico foram utilizadas sete amostras e oito para a geração do *template*.

A Floresta Aleatória teve uma melhor acurácia em relação ao CBDM, sendo 6,25% maior, na média quanto utiliza-se nove voluntários e 15,25% quanto utiliza dezoitos voluntários.

Na Figura 6 demonstra a acurácia dos métodos de classificação com base no número amostras utilizadas para a geração do *template*, sendo que foram realizados teste utilizando as amostras de nove e dezoitos voluntários para a classificação.

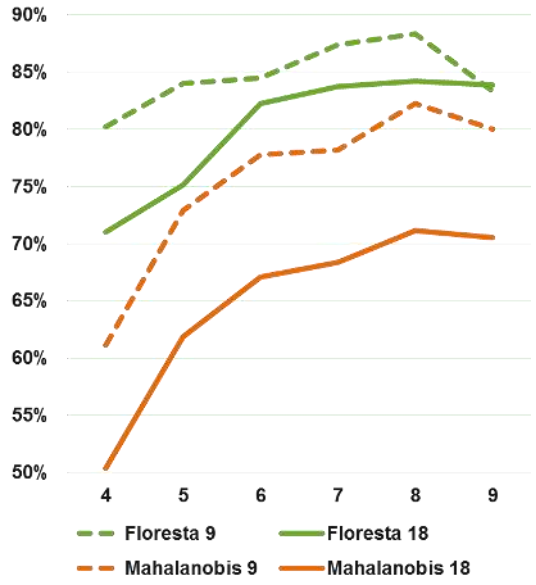
A Floresta Aleatória também obteve uma melhor acurácia, sendo 12,65% a mais que o CBDM em média.

Figura 5. Acurácia dos métodos



Fonte: Elaborado pelo autor.

Figura 6. Acurácia dos métodos com base nas amostras.



Fonte: Elaborado pelo autor.

Foram realizados teste de verificação da amostra, utilizando o método estatístico de Mahalanobis. O teste de verificação é feito utilizando como base a função de *Threshold* (seção 3.1.1).

Os resultados obtidos nos testes utilizam os valores de FAR (taxa de falsa aceitação) e FRR (taxa de falsa rejeição).

Os testes foram feitos utilizando 4, 5, 6, 7, 8 e 9 amostras para gerar o *template* do voluntário. Onde cada testes utilizavam a Equação (7):

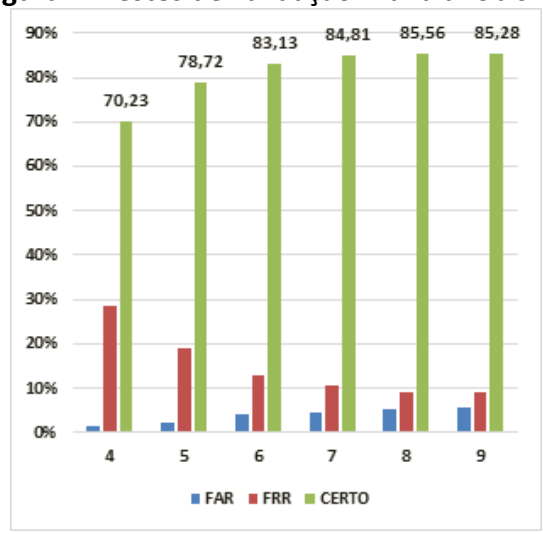
$$T = 10 - q \qquad (7)$$

Sendo T o número de amostra que serão utilizadas nos testes e q a quantidade de amostras para a geração do *template*.

Para realizar os testes de FRR e FAR o mesmo número de amostras T era utilizadas, exemplo: utilização de 6 amostras para a gerar o *template*, utilizaria 4 amostras do próprio voluntários, mais 4 amostras de voluntários aleatórios, para realizar teste de cada frase.

Como demostra na Figura 7, quando se utiliza mais de 6 amostras para a geração do *template,* tem um resultado satisfatório, sendo que se tem um resultado estável, na média de 84,69 % de acertos.

Figura 7. Testes de validação Mahalanobis

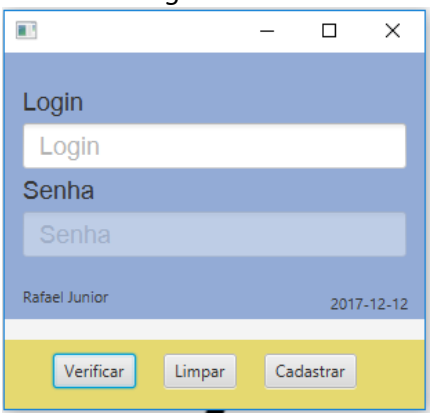


Fonte: O autor.

Com a Figura 8 é possível observar a ideia principal do projeto, que é a associação do *login* e senha a uma característica biométrica por meio da dinâmica da digitação, para o aumento da segurança. Onde o voluntario se cadastra com o *login* e sua senha, sendo necessário repetir a senha 7 vezes para a geração de um *template* que é gravado junto com o cadastro do usuário.

No processo de autenticação, o sistema é capaz de validar se o *login*, a senha e a amostra da senha digitada correspondem ao usuário, se sim o usuário é dito como verdadeiro pelo sistema.

Figura 8. Tela de *login*.



Fonte: O autor.

6 CONSIDERAÇÕES FINAIS

Os métodos utilizados contribuíram de forma positiva para o resultado final. A

remoção de valores *outliers* também foi importante para melhoria da taxa de acerto.

Os resultados foram considerados satisfatórios visto que utilizando classificadores de Floresta Aleatória, obteve-se uma média de 84,65% de acerto. Já o CBDM teve um resultado em média de 74% da acurácia.

Para a ideia inicial do projeto, que foi associar o método de classificação biométrica com o tradicional *login* e senha, observou-se que é necessário trocar os classificadores por validadores de amostra. É possível notar nas Figuras 4 e 5 que com o aumento de voluntários cadastrados a acurácia do método tende a diminuir gradativamente, reduzindo assim a precisão dos métodos. Mas como visto na Figura 6, que utiliza um validador de amostra, o método não sofre variação com o número de voluntários utilizados nos testes, já que para o processo de validação de um voluntário específico não se faz necessário realizar a comparação com as amostras dos demais voluntários, como acontece no método de classificação Floresta Aleatória.

Futuros trabalhos podem ser realizados utilizando outras técnicas para extração de características e autenticação. Outros tipos de classificadores também podem ser testados, para determinar qual melhor se comporta para um determinado conjunto de amostras.

REFERÊNCIAS

ALSULTAN, A.; WARWICK, K. Keystroke dynamics authentication: a survey of free-text methods. **International Journal of Computer Science Issues**, v. 10, n. 4, p. 1-10, 2013.

ARAÚJO, L. C. F. **Uma metodologia para autenticação pessoal baseada em dinâmica da digitação**. 2004. Dissertação (Mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Eletrica e de Computação, Campinas, SP. Disponível em: http://www.repositorio.unicamp.br/handle/REPOSIP/259073. Acesso em: 3 ago. 2018.

BREIMAN, L.. Random forests. **Machine learning**, v. 45, n. 1, p. 5-32, 2001. https://doi.org/10.1023/A:1010933404324.

COSTA, C. R. N. *et al.* Autenticação Biométrica via Dinâmica da Digitação em Teclados Numéricos. *In*: SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES–SBrT'05. 22., 2005. Campinas. **Anais** [...]. Campinas, 2005.

CRUZ, M. A. S.; DUARTE, J. C.; GOLDSCHMIDT, Ronaldo Ribeiro. Keystroke Dynamics Applied to Periodic Authentication in Virtual Learning Environments. **Brazilian Journal of Computers in Education**, v.25, n.2, p.36-60, 2017. doi:http://dx.doi.org/10.5753/rbie.2017.25.02.36.

LINDEN, R. Técnicas de agrupamento. **Revista de Sistemas de Informação da FSMA**, v. 4, p. 18-36, 2009.

JOYCE, R.; GUPTA, G. Identity authentication based on keystroke latencies. **Communications of the ACM**, v. 33, n. 2, p. 168-176, 1990. https://doi.org/10.1145/75577.75582

BLEHA, S. A.; OBAIDAT, M. S. Dimensionality reduction and feature extraction applications in identifying computer users. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 21, n. 2, p. 452-456, 1991. https://doi.org/10.1109/21.87093

HAN, J.; PEI, J.; KAMBER, M. **Data mining:** concepts and techniques. Burnaby, Canada: Elsevier, 2011.

HALL, M. *et al.* The WEKA data mining software: an update. **ACM SIGKDD explorations newsletter**, v. 11, n. 1, p. 10-18, 2009. https://doi.org/10.1145/1656274.1656278