Algoritmos de Aprendizado de Máquina Aplicados ao Reconhecimento de Usuário Baseado na Dinâmica da Digitação: Um Estudo Comparativo

Marco Aurélio da Silva Cruz[1], **Ronaldo Ribeiro Goldschmidt**[1]

[1]Seção de Engenharia da Computação (SE/8) – Instituto Militar de Engenharia (IME)
22.290-270 – Rio de Janeiro – RJ – Brasil

{marcocruz,ronaldo.rgold}@ime.eb.br

Abstract. *Comparative studies about keystroke dynamics-based user recognition with Machine Learning (ML) algorithms do not consider multiple datasets in their evaluation. Moreover, each study usually concentrates on a single and non-public dataset. Hence they cannot generalize which algorithms are, in fact, the best ones to perform this kind of recognition task. Thus, this work presents a comparative study which analyzes the performances of six popular ML algorithms applied to five different public datasets with static and predefined samples (e.g., passwords). The results of the experiments showed that Random Forest was able to outperform all the other algorithms in all datasets.*

Resumo. *Em geral, as análises comparativas feitas em trabalhos que estudam a tarefa de reconhecimento de usuários baseada na dinâmica da digitação utilizando algoritmos de Aprendizado de Máquina (AM) se restringem à avaliação considerando apenas um único conjunto de dados específico, em geral não público, o que torna difícil generalizar quais são, de fato, os algoritmos mais indicados para realizar tal tarefa. Assim, o presente trabalho tem como objetivo apresentar um estudo comparativo acerca do desempenho de alguns dos mais populares algoritmos de AM aplicados a tal tarefa utilizando cinco conjuntos de dados públicos compostos por amostras estáticas (textos fixos, como senhas, por ex.). Os experimentos realizados mostraram que o Random Forest foi capaz de superar os demais algoritmos em todos os conjuntos de dados analisados.*

1. Introdução

O uso de aplicativos computacionais em diferentes áreas e atividades do dia-a-dia somado ao aumento do volume de ataques cibernéticos em todo o mundo tem contribuído de forma expressiva para o crescimento da preocupação com questões relacionadas à segurança da informação [Sêmola, 2014]. Como consequência, diversos desses aplicativos vêm adotando algum tipo de medida de proteção [Jain et al., 2016].

Uma medida de proteção frequentemente empregada é o reconhecimento de usuário por meio de identificação e senha [Sêmola, 2014]. Apesar de sua popularidade, tal medida pode, por vezes, se mostrar insuficiente, uma vez que senhas podem ser descobertas por usuários mal intencionados [Jain et al., 2016].

Dentre as formas de reconhecimento de usuário mais confiáveis do que as baseadas em senhas estão aquelas que utilizam técnicas biométricas [Jain et al., 2016;

O'Gorman, 2003]. Tais técnicas realizam o reconhecimento de usuários a partir de atributos físicos das pessoas tais como faces ou digitais, ou a partir de características comportamentais como o padrão de digitação de cada indivíduo [O'Gorman, 2003].

A utilização do padrão de digitação como técnica biométrica para reconhecimento de usuário apresenta algumas vantagens em relação a outras alternativas. Por exemplo, possui natureza não intrusiva, uma vez que não demanda a interrupção de trabalho e nem viola a privacidade do usuário, como frequentemente ocorre com outras técnicas biométricas que revelam, por meio de imagens e sons, momentos de intimidade dos seus usuários [Banerjee and Woodard, 2012; Cruz et al., 2017]. Outra característica relevante desta técnica refere-se ao seu baixo custo, pois sua operacionalização requer apenas a presença de um dispositivo para digitação, normalmente um teclado analógico ou digital. Tais características podem ser muito importantes em diversas aplicações, como em ambientes virtuais de aprendizagem utilizados em ensino a distância, entre outras [Cruz et al., 2017].

Diferentes algoritmos podem ser empregados no reconhecimento de usuários baseado na dinâmica da digitação. Entre eles estão os algoritmos baseados em Aprendizado de Máquina (AM) [Hu et al., 2008; Teh et al., 2013]. A principal vantagem de utilizar AM é a sua capacidade de criar modelos discriminativos que aprendem a partir de conjuntos de dados [Goldschmidt et al., 2015].

Apesar de diversos trabalhos de pesquisa terem investigado o uso de algoritmos de AM na implementação do reconhecimento de usuário baseado na dinâmica da digitação, são escassas as iniciativas voltadas à comparação entre os desempenhos de diferentes algoritmos aplicados a diferentes conjuntos de dados. Em geral, as análises comparativas se restringem à avaliação considerando um conjunto de dados específico [Teh et al., 2013; Alsultan and Warwick, 2013; Ali et al., 2017], o que torna difícil generalizar quais algoritmos de AM são, de fato, mais eficazes para implementar tal reconhecimento. Além disso, por vezes, ocorrem mudanças nos métodos de experimentação adotados nos diferentes trabalhos [Killourhy and Maxion, 2009; Kobojek and Saeed, 2016; Deng and Zhong, 2013; Maheshwary et al., 2017], o que dificulta ainda mais a realização de análises comparativas.

Diante do exposto, o presente trabalho tem como objetivo apresentar um estudo comparativo acerca do desempenho de diferentes algoritmos de AM aplicados à tarefa de reconhecimento de usuário baseado na dinâmica da digitação utilizando diferentes conjuntos de dados compostos por amostras estáticas (textos fixos, como senhas, por exemplo). Neste estudo, um único método de experimentação foi adotado na avaliação de alguns dos mais populares algoritmos de AM a tal tarefa (MPL, LSTM e Deep Belief Net, K-NN, Random Forest, SVM) em cinco diferentes conjuntos de dados públicos. Os resultados obtidos nos experimentos mostraram que os desempenhos obtidos por cada algoritmo podem variar bastante quando testados em diferentes conjuntos de dados. No entanto, foi possível observar que um deles, o *Random Forest*, foi capaz de superar os demais algoritmos nos cinco conjuntos de dados analisados.

Este texto encontra-se organizado em mais seis seções. A Seção 2 apresenta os conceitos básicos sobre dinâmica da digitação. A Seção 3 discute alguns dos principais trabalhos relacionados. A descrição do método de experimentação é apresentada na Seção

4. Os experimentos realizados e seus resultados, são descritos e analisados na Seção 5. A Seção 6 resume as principais conclusões e expõe as possibilidades de trabalhos futuros.

2. Fundamentos da Dinâmica da Digitação

No reconhecimento de usuários baseado na dinâmica da digitação existem trabalhos que exploram textos estáticos e textos dinâmicos. Os estáticos, ou fixos, que são estudados no presente trabalho, são aqueles em que as sentenças digitadas pelos usuários são pré-definidas [Teh et al., 2013; Alsultan and Warwick, 2013; Pisani and Lorena, 2013] tais como senhas, por exemplo. Os dinâmicos são aqueles em que qualquer texto pode ser digitado [Teh et al., 2013; Alsultan and Warwick, 2013; Pisani and Lorena, 2013].

Na medida em que os textos são digitados, são coletados dados relacionados aos momentos de pressionamento e soltura de cada tecla. Esses dados coletados são denominados, neste trabalho, de dados brutos.

Em geral, após a aquisição dos dados brutos, é executada uma fase chamada de Engenharia de Atributos [Han et al., 2011; Teh et al., 2013; Shinde et al., 2016], que consiste em gerar novos atributos, considerados relevantes para o processo de reconhecimento do usuário, a partir dos dados coletados [Dowland and Furnell, 2004; Gunetti and Ruffo, 1999]. Diversas novas características podem ser geradas na Engenharia de Atributos, porém, as duas características mais comuns são o tempo de pressionamento e a latência entre duas teclas [Alsultan and Warwick, 2013; Banerjee and Woodard, 2012; Sim and Janakiraman, 2007; Teh et al., 2013].

Na Figura 1, existem 3 teclas fictícias apresentadas em ordem temporal de pressionamento. Cada tecla é representada por K_n, onde n indica a ordem em que ela foi digitada. Além disso, cada tecla possui dois registros de tempo, um em que a tecla foi pressionada (K_n^D) e outro em que foi solta (K_n^U).

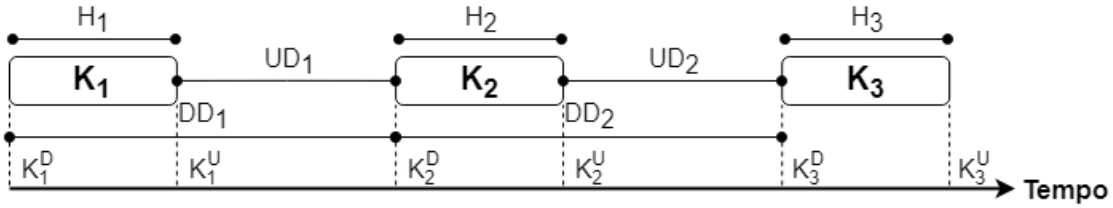

Figura 1. Tempo de Pressionamento e Latências

Para calcular tempo de pressionamento de uma tecla, basta diminuir o instante de tempo capturado no momento em que a ela foi solta pelo instante de tempo em que foi pressionada [Rodrigues et al., 2006]. Na Figura 1, por exemplo, o tempo de pressionamento de uma tecla n é representado por H_n, onde $H_n = K_n^U - K_n^D$.

Existem diferentes formas de explorar as latências entre duas teclas [Rodrigues et al., 2006]. A latência UD, ilustrada na Figura 1, é o intervalo entre o momento que uma tecla é solta e a tecla seguinte é pressionada. Logo, $UD_n = K_{n+1}^D - K_n^U$. A latência DD é obtida ao subtrair o momento em que duas teclas consecutivas foram pressionadas. Assim, $DD_n = K_{n+1}^D - K_n^D$. É claro que só é possível calcular ambas latências se a tecla K_{n+1} existir.

3. Trabalhos Relacionados

O reconhecimento de usuários por meio de padrões de digitação vem sendo estudado desde a década de 80 [Gaines et al., 1980]. Inicialmente, as pesquisas precisavam executar uma fase de coleta de dados, pois não existiam repositórios públicos que contivessem informações sobre a dinâmica da digitação das pessoas [Giot et al., 2009a]. Apenas por volta do início do século XX, alguns conjuntos de dados começaram a ser disponibilizados publicamente [Filho and Freire, 2006; Giot et al., 2009b; Killourhy and Maxion, 2009]. A Tabela 1 fornece alguns exemplos de conjuntos de dados públicos. Cabe ressaltar que todos eles se enquadram na categoria de conjuntos compostos por textos de natureza estática associados a senhas coletadas de diferentes usuários. A primeira coluna apresentada na referida tabela contém identificadores utilizados no presente artigo para referenciar inequivocamente os conjuntos de dados. A segunda coluna indica os trabalhos de pesquisa que disponibilizaram os referidos conjuntos. As terceira e quarta colunas mostram as quantidades de usuários e de amostras de digitação, respectivamente. A quinta coluna informa se o teclado utilizado na coleta de dados foi digital, característico de dispositivos móveis, ou analógico, normalmente encontrados em computadores de mesa. A última coluna informa o tamanho do(s) texto(s) utilizados em quantidade de caracteres. O "*"no DB6 é porque existiram textos que foram criados pelos voluntários, além dos dois textos impostos.

Tabela 1. Relação de conjuntos de dados públicos que contêm informações sobre a dinâmica de digitação de usuários

Id	Referência	Usuários	Amostras	T. Digital	T. T.
DB1	[Killourhy and Maxion, 2009]	51	20400	Não	10
DB2	[Giot et al., 2009a]	133	7544	Não	16
DB3	[El-Abed et al., 2014]	51	955	Sim	14
DB4	[Antal and Nemes, 2016]	54	10313	Sim	10, 15, 11
DB5	[Antal et al., 2015]	42	2142	Sim	10
DB6	[Giot et al., 2012]	118	28476	Não	6, 17, *

Apesar da disponibilização pública de conjuntos de dados ter favorecido a realização de estudos comparativos entre diferentes modelos preditivos, poucas foram as iniciativas nesta direção. Uma revisão dos trabalhos de pesquisa na área de reconhecimento de usuário baseado na dinâmica da digitação revela cinco grupos distintos.

No primeiro grupo estão os trabalhos que coletaram e realizaram experimentos em seu próprio conjunto de dados, utilizando apenas um algoritmo de AM cada. São eles: [Meng et al., 2014; Shimshon et al., 2010; Maxion and Killourhy, 2010; Pavaday and Soyjaudah, 2007; Loy et al., 2007; Sulong et al., 2009; Ali et al., 2009; Karnan and Akila, 2010; Ngugi et al., 2011; Sheng et al., 2005; Loy et al., 2005; Sen and Muralidharan, 2014; Alpar, 2017, 2018; Lee et al., 2018; Wu and Chen, 2015; Lin et al., 2018; Yu and Cho, 2003; Thierry and Chuan, 2018]. Como muitos desses trabalhos não disponibilizaram publicamente seus conjuntos de dados, uma análise comparativa dos desempenhos de diferentes algoritmos de AM não se apresentou como alternativa viável.

O segundo grupo contém os seguintes trabalhos: [Maheshwary et al., 2017; Kobojek and Saeed, 2016; Deng and Zhong, 2013; Bhatia and Hanmandlu, 2018; Giot et al.,

2009b, 2011; Pisani and Lorena, 2012; Deng and Zhong, 2015]. Apesar de cada um ter realizado seus experimentos em um conjunto de dados público, esses trabalhos avaliaram apenas um algoritmo de AM cada. Neste grupo, a comparação direta entre os resultados individuais obtidos pelo algoritmo de cada pesquisa também não se mostra viável dada a falta de padronização com relação ao método de experimentação adotado em cada caso.

Apenas [Çeker and Upadhyaya, 2017; Alshanketi et al., 2016] enquadram-se no terceiro grupo. De forma similar aos do segundo grupo, esses trabalhos também avaliaram um único algoritmo de AM cada. A diferença é tal avaliação não ficou restrita a apenas um conjunto de dados público em cada trabalho. Como ambos utilizaram conjuntos de dados e métodos de experimentação distintos, uma análise comparativa entre os resultados obtidos também não é indicada.

Os trabalhos [Ho, 2014; Thanganayagam and Thangadurai, 2015; Antal et al., 2015; Antal and Szabó, 2015; Yu and Cho, 2003; Antal and Nemes, 2016; Ali et al., 2009; Pavaday and Soyjaudah, 2007; Loy et al., 2007; Killourhy and Maxion, 2009; Loy et al., 2005; Sen and Muralidharan, 2014; Meng et al., 2014] estão no quarto grupo. Eles se caracterizam pelo fato de terem comparado mais de um algoritmo de AM em um mesmo conjunto de dados público cada. Apesar de serem iniciativas voltadas ao estudo comparativo entre os desempenhos de vários algoritmos, cada uma delas ficou limitada à experimentação em um único conjunto de dados, o que levanta questionamentos quanto ao grau de independência e de generalização dos resultados em relação aos dados.

No quinto e último grupo estão os trabalhos [Pisani et al., 2015] e [Pisani et al., 2018]. Em ambos os casos, os experimentos foram realizados nos mesmos conjuntos de dados públicos, (mais especificamente, nos conjuntos DB1, DB2, DB6, indicados na Tabela 1). Os dois estudos usaram algoritmos de AM (MLP [Rosenblatt, 1962], Floresta Randômica [Breiman, 2001], Classificador Bayesiano [Langley et al., 1992] e Árvore de Decisão [Salzberg, 1994]) para formar comitês de classificadores. Um comitê de classificadores é um modelo de classificação cujo a predição final é definida a partir da agregação de decisões individuais de cada algoritmo classificador que compõe tal modelo [Faceli et al., 2011; Goldschmidt et al., 2015]. Em Pisani et al. [2015], o método de experimentação utilizado causava desbalanceamento nos conjuntos de treino e teste. Em Pisani et al. [2018], esse problema foi corrigido, porém, apenas o desempenho geral do comitê foi demonstrado, impossibilitando a observação e comparação dos comportamentos individuais dos membros do comitê. Além disso, ambos os trabalhos utilizaram apenas conjuntos de dados compostos por amostras coletadas de teclados analógicos, o que inviabiliza a generalização dos resultados para cenários onde sejam utilizados teclados digitais.

Diferentemente dos trabalhos acima, o estudo reportado neste artigo foi realizado com o intuito de permitir uma comparação direta entre diferentes algoritmos de AM. A fim de buscar uma maior generalização de resultados, o referido estudo foi realizado em diferentes conjuntos de dados públicos, contendo amostras coletadas de teclados analógicos e digitais. O método de experimentação adotado foi o mesmo para todos os algoritmos em todos os conjuntos de dados e encontra-se descrito em detalhe na próxima seção.

4. Método de Experimentação

O método de experimentação adotado é composto por três procedimentos principais: (1) Engenharia de Atributos; (2) Seleção de Conjuntos de Treino e Teste; e (3) Confecção e Avaliação dos Modelos de AM. Entre os trabalhos relacionados, o procedimento 2 é o que mais apresenta divergências em sua implementação. Tais divergências se devem à maneira de tratar os diferentes volumes de usuários disponíveis e possíveis desbalanceamentos entre as quantidades de amostras desses usuários.

Embora inspirado em vários métodos existentes na literatura, o método de experimentação adotado neste artigo foi concebido para lidar com todos os conjuntos de dados da mesma forma, independentemente das diferenças de distribuição de dados eventualmente existentes entre eles. Esta seção tem como objetivo especificar em detalhe o referido método, explicando cada um dos três procedimentos mencionados.

Algumas notações e definições foram tomadas como base nessas descrições. Assim, sejam: C, um conjunto contendo dados sobre usuários e suas amostras de digitação; $\mathscr{U} = \{u_1, u_2, ..., u_x\}$, o conjunto de todos os usuários u_i disponíveis em C; $A'_u = \{a'_{u,1}, a'_{u,2}, ..., a'_{u,y}\}$, o conjunto de todas as amostras em estado bruto associadas a um usuário $u \in \mathscr{U}$, onde uma amostra $a'_{u,i}$ em estado bruto consiste de uma sequência de registros temporais dos momentos de pressionamento (K^D_n) e soltura (K^U_n) de n teclas usadas para escrever uma sentença de comprimento fixo (i.e. $a'_{u,i} = (K^D_1, K^U_1, ..., K^D_n, K^U_n)$); e $A' = \bigcup_{i=1}^{x} A'_{u_i}$ é o conjunto de todas as amostras disponíveis em C.

O procedimento de Engenharia de Atributos do método de experimentação adotado consiste em transformar o conjunto de amostras em estado bruto A' em um conjunto de amostras pré-processadas A da seguinte forma: para cada amostra em estado bruto $a'_{u,i} = (K^D_1, K^U_1, ..., K^D_n, K^U_n)$, são extraídas as características de tempo de pressionamento H e de latências UD e DD, de forma análoga ao descrito na Seção 2, gerando uma amostra pré-processada representada por um vetor de características com a seguinte configuração $a_{u,i} = (H_1, UD_1, DD_1, ..., H_{n-1}, UD_{n-1}, DD_{n-1}, H_n)$. A escolha das referidas características deveu-se basicamente à sua popularidade em muitos trabalhos relacionados como [Pisani and Lorena, 2013; Teh et al., 2013; Alsultan and Warwick, 2013]. Cabe ressaltar ainda que, como as latências UD_i e DD_i associadas à i-ésima tecla dependem da tecla $i + 1$, na representação resultante não existem as latências UD_n ou DD_n. Assim, para a última tecla digitada, só é possível calcular seu tempo de pressionamento.

O procedimento de seleção dos conjuntos de treino e de teste encontra-se descrito no Algoritmo 1. Ele faz iterações sobre o conjunto $\mathscr{U}$, de modo que para cada usuário u, são construídos r pares de conjuntos, sendo um de treino e outro de teste. Cada par representa a configuração de um *fold* do processo de validação cruzada com *r-folds* [Faceli et al., 2011]. Os r pares de cada u são então armazenados em λ. O algoritmo constrói esses pares após criar um conjunto de amostras balanceado S. Para formar S, é preciso obter uma quantidade de amostras falsas, AF_u, proporcional à quantidade de amostras verdadeiras, A_u. O conjunto AF_u precisa ser construído com cuidado, pois existem muito mais amostras falsas do que verdadeiras nos conjuntos de dados. Por isso, para obter AF_u foi utilizado um subconjunto dos usuários impostores. Impostor é um nome usado para designar os usuários diferentes de u, ou seja, todos os usuários v que pertencem ao conjunto $\mathscr{U} - \{u\}$. A quantidade de usuários impostores que serão utilizados é definido na linha 6 do algoritmo, no valor atribuído à β. Tal valor é calculado em função de k

(valor informado como entrada) e da quantidade de amostras verdadeiras, y.

Algoritmo 1: MÉTODO DE PARTICIONAMENTO DE DADOS

Entrada: k (quantidade de amostras falsas por usuário para formar o conjunto de treino e teste), r (quantidade de *folds*)
Saída: Conjuntos de treino e teste utilizados para construir os modelos de reconhecimento de usuários

1 **início**
2 $\lambda = \varnothing$;
3 **para** $\forall u \in \mathscr{U}$ **faça**
4 y = obterTamanhoConjunto(A_u); x = (y / 2); $\beta = \lceil$x / k$\rceil$;
5 $U^{ImpostorAleatório}$ = obterUsuariosAleatorios(β, $\mathscr{U} - \{u\}$);
6 $AF^{Treino} = \varnothing$; $AF^{Teste} = \varnothing$; i = 0;
7 **para** $\forall v \in U^{ImpostorAleatório}$ **faça**
8 **enquanto** $i < k$ **faça**
9 α = obterNumeroAleatorio($[k+1,\ y]$);
10 $AF = AF \cup \{a_{v,\alpha}\}$; $i = i + 1$;
11 **fim**
12 $AF = AF \cup \{a_{v,1},\ ...,\ a_{v,k}\}$
13 **fim**
14 $S = A_u \cup AF$
15 $\lambda = \lambda \cup criarParesTreinoTesteParaRFolders(S, r)$
16 **fim**
17 **fim**
18 **retorna** λ

Assim, AF_u, é construído no laço da linha 11, por meio da combinação das k primeiras amostras e mais k registros aleatórios de cada impostor, v. Isso faz com que os conjuntos de treino e teste sejam mais heterogêneos, uma vez que são compostos por registros falsos que os impostores não estavam acostumados a digitar (as k primeiras amostras) e também por outros registros, obtidos de forma aleatória. As amostras aleatórias são cuidadosamente selecionadas para não serem as mesmas que as k primeiras. Por isso, o valor que α recebe é o intervalo $[k+1,\ y]$ (vide linha 13), isto é, α é maior que k e ao mesmo tempo está limitado à quantidade máxima de amostras de u.

No último trecho do Algoritmo 1, a função da linha 20, retorna os conjuntos $\lambda_{u,i}$. Cada $\lambda_{u,i}$ é um par $(A_{u,i}^{Treino},\ A_{u,i}^{Teste})$. Sendo assim, ele representa a configuração da *i-ésima* iteração do processo de validação cruzada, em que existem dois conjuntos, um com amostras de treino, $A_{u,i}^{Treino}$, e um com amostras de teste, $A_{u,i}^{Teste}$. Portanto, $\lambda = \{\lambda_{1,1}, ..., \lambda_{1,r}, ..., \lambda_{n,r}\}$, sendo n a quantidade de usuários e r a quantidade de *folds*.

Cabe ressaltar que Algoritmo 1 se adapta a diferentes conjuntos de dados. No entanto, é importante notar que o valor de k deve ser menor que o número de amostras do usuário com menos amostras. Além disso, para que o algoritmo funcione, a quantidade de usuários especificada em cada β precisa ser menor ou igual a quantidade de elementos em $\mathscr{U}$.

O último procedimento consiste na confecção e avaliação dos modelos de apren-

dizado de máquina. O procedimento cria e testa um modelo de classificação para cada usuário a partir do par $\lambda_{u,i}$. Tais modelos são criados com base em duas informações, o algoritmo de AM e o conjunto de treino, A^{Treino}. Em seguida, eles são testados e seus resultados parciais são guardados. Após ter todos os resultados parciais, o desempenho individual de cada usuário é calculado, utilizando uma métrica, m. O desempenho geral é expresso pela média e pelo desvio padrão dos desempenhos individuais. Essa abordagem foi seguida por diversos trabalhos [Pisani and Lorena, 2013; Killourhy and Maxion, 2009; Kobojek and Saeed, 2016; Maheshwary et al., 2017; Deng and Zhong, 2013]. É recomendado que, m, seja a EER (Taxa de Erro Igual - Equal Error Rate), pois ela calcula o erro, equilibrando as taxas de falsa aceitação e falsa rejeição e é uma das mais utilizadas em trabalhos sobre dinâmica da digitação [Killourhy and Maxion, 2009; Pisani and Lorena, 2013; Teh et al., 2013].

5. Experimentos e Resultados

Esta seção apresenta as configurações e os resultados dos experimentos realizados. Todos os experimentos foram feitos seguindo o Método de Experimentação descrito na Seção 4. Tal método foi aplicado nos cinco primeiros conjuntos de dados da Tabela 1. O DB1 e o DB2 foram escolhidos por algumas razões: eles utilizaram dados coletados a partir de teclados analógicos; são os conjuntos de dados públicos mais utilizados pelos trabalhos relacionados encontrados; o DB1 possui um número grande de amostras; e o DB2 possui um número grande de usuários. Os conjuntos de dados DB3, DB4 e DB5 foram selecionados porque: a coleta de seus dados foi feita por meio de teclados digitais; possuem quantidades variadas de usuários e amostras; o DB4 possui amostras com textos de diferentes tamanhos.

Para ser aplicado nesses conjuntos de dados, o método de experimentação precisa que algumas variáveis dos procedimentos sejam pré-definidas. No Algoritmo 1, as variáveis usadas foram $r = 10$, para fazer o processo de r-fold, e $k = 5$, valor recomendado por alguns trabalhos [Killourhy and Maxion, 2009; Deng and Zhong, 2013; Maheshwary et al., 2017]. Este valor precisa ser um valor pequeno como cinco, pois, nas primeiras amostras registradas, os impostores ainda não estão acostumados a digitar o mesmo conteúdo [Killourhy and Maxion, 2009]. No procedimento de confecção e avaliação dos modelos de AM, λ é uma variável cujo valor vai depender da resposta do Algoritmo 1. A métrica de avaliação escolhida, m, foi a EER. Além disso, foram utilizados seis algoritmos de AM.

A Tabela 2 exibe, em sua primeira coluna, os seis algoritmos utilizados nos experimentos. A escolha desses algoritmos deveu-se basicamente à sua popularidade e ao fato de explorarem diferentes vieses de representação de conhecimento e de busca [Han et al., 2011]. Os três primeiros (MLP, Deep Belief Net, LSTM) foram implementados a partir da biblioteca Keras [1], enquanto que os demais (K-NN, Random Forest, SVM) foram implementados e configurados conforme especificação da biblioteca Scikit-learn [2]. A MLP é uma Rede Neural Perceptron com múltiplas camadas que segue a implementação feita por Maheshwary et al. [2017], possuindo três camadas intermediárias com 100, 300 e 100 neurônios. A Deep Belief Net segue as mesmas configurações descritas em Deng

[1]Para mais informações acesse: https://keras.io/

[2]Para mais informações acesse: http://scikit-learn.org/

and Zhong [2013] e possui duas camadas intermediárias com 100 neurônios que são pré treinadas com Máquinas Restritas de Boltzman [Fischer and Igel, 2014]. A LSTM é a mesma descrita por Kobojek and Saeed [2016], com duas camadas intermediárias recorrentes, sendo uma com 250 e a outra com 100 neurônios. O K-NN (k Vizinhos Mais Próximos) [Cover and Hart, 1967] foi utilizado com $k = 5$. O Random Forest [Breiman, 2001] foi configurado com 10 árvores de decisão que utilizaram Índice Gini para medir a qualidade da divisão dos nós. Por último, o SVM [Cortes and Vapnik, 1995] usou como função-núcleo o RBF e a taxa de penalidade de erro igual a 1.

Os resultados obtidos por cada um dos algoritmos em relação a cada conjunto de dados seguem apresentados na Tabela 2. Os valores numéricos descritos representam a média e o desvio padrão do EER dos modelos gerados para todos os usuários. O melhor desempenho obtido em cada conjunto de dados está destacado em negrito.

Tabela 2. Média e Desvio Padrão do EER (%) obtido por 6 Algoritmos de AM em 5 Conjuntos de Dados

	Banco de Dados				
Algoritmos	**DB1**	**DB2**	**DB3**	**DB4**	**DB5**
MLP	4,7 ± 2,6	7,1 ± 3,7	10,9 ± 9,6	12,7 ± 4,7	15,3 ± 5,1
Deep Belief Net	4,8 ± 2,5	10,2 ± 5,7	77,5 ± 4,9	60,1 ± 1,0	40,5 ± 3,7
LSTM	4,7 ± 2,6	9,0 ± 4,7	11,4 ± 10,8	23,1 ± 8,2	28,2 ± 6,6
K-NN	31,9 ± 7,4	13,6 ± 4,7	11,5 ± 9,6	21,8 ± 4,8	21,7 ± 6,1
Random Forest	**4,5 ± 3,1**	**4,7 ± 2,8**	**7,6 ± 7,1**	**11,4 ± 3,8**	**9,4 ± 4,4**
SVM	13,7 ± 6	48,1 ± 7,9	10,6 ± 13,1	35,7,0 ± 9,7	17 ± 6,1

Como é possível observar na Tabela 2, os valores obtidos por cada algoritmo variam bastante. A Deep Belief Net, por exemplo, conseguiu EER médio de 4,8% no DB1, porém, no DB3, a média foi 77,5%. Isso mostra a importância de avaliar os algoritmos em diferentes conjuntos de dados.

O Random Forest obteve os melhores resultados em todas as análises. Tais resultados confirmam o bom desempenho desse algoritmo no reconhecimento de usuários baseado na dinâmica da digitação, como apontado por trabalhos relacionados [Maxion and Killourhy, 2010; Antal and Szabó, 2015; Ho, 2014]. Uma das possíveis razões para o referido desempenho é que ele capaz de lidar com perturbações nos dados e de aprender com conjuntos de dados pequenos [Maxion and Killourhy, 2010; Breiman, 2001].

Ainda na Tabela 2, também é possível observar que os menores valores de EER foram obtidos no conjunto DB1. Uma das possíveis razões para isso é que esse conjunto de dados possui mais amostras, o que faz com que os métodos que sejam mais sensíveis ao volume de dados, como as redes neurais, possam alcançar resultados melhores. Além disso, a diversidade de exemplos de um mesmo texto aumentam a capacidade discriminativa dos modelos para tal conteúdo.

O tamanho das amostras de textos pequenos, como os comumente utilizados em senhas e emails, não parecem influenciar tanto os resultados quanto o volume de amostras. Pois embora as amostras do DB1 e DB5 tenham sido coletadas a partir de textos com 10 caracteres, os resultados obtidos no DB5 foram piores que os obtidos no DB1, para a maior parte dos algoritmos. Além disso, os conjuntos DB2, DB3 e DB4 possuem textos

com tamanho maior que 10, porém, também não apresentaram resultados significativamente melhores que os obtidos no DB1.

No DB4, o tamanho dos três textos foi padronizado. Os textos menores foram complementados com os seus dados iniciais até atingirem o tamanho do maior texto disponível no conjunto. Embora essa não seja uma abordagem comum (pois normalmente os trabalhos relacionados criaram seus modelos preditivos utilizando apenas um tipo de texto), ela foi adotada justamente para se verificar qual seria o comportamento dos algoritmos nessa situação. Por existir maior variedade de tipos de sentenças, era esperado que os piores resultados alcançados pelos algoritmos fossem no DB4. Porém, isso só aconteceu para o Random Forest. Os demais tiveram os piores resultados em outros conjuntos de dados. Uma das razões para o desempenho acima do esperado nesses casos pode ter sido o próprio método utilizado para completar as strings menores, além do volume de amostras disponíveis.

Em geral, os conjuntos cujos dados foram coletados por meio de teclados digitais apresentaram os piores resultados. Isso pode ter ocorrido por causa da afinidade dos usuários com os dispositivos que foram utilizados na coleta e pela quantidade de exemplos individuais fornecidos aos modelos, para o caso dos conjuntos DB3 e DB5, principalmente. Porém, é importante notar que os resultados poderiam ser melhorados se características específicas dos sensores dos dispositivos móveis, como o acelerômetro e giroscópio, fossem exploradas [Teh et al., 2016].

6. Considerações Finais

Diferentes técnicas biométricas são utilizadas para desempenhar a tarefa de reconhecimento de usuário. A utilização do padrão de digitação como técnica biométrica apresenta algumas vantagens em relação as outras alternativas, pois além de ser não-intrusiva, demanda baixo custo de operacionalização. Embora diversas pesquisas tenham investigado o uso de algoritmos de AM para implementação de tal tarefa usando essa técnica, são escassas as iniciativas voltadas à comparação entre os desempenhos de diferentes algoritmos aplicados a diferentes conjuntos de dados. Em geral, as análises comparativas se restringem à avaliação considerando apenas um conjunto de dados específico, o que torna difícil generalizar quais algoritmos de AM são, de fato, mais indicados para realizar a tarefa de reconhecimento.

Diante do exposto, o presente trabalho teve como objetivo apresentar um estudo comparativo acerca do desempenho de diferentes algoritmos de AM aplicados ao reconhecimento de usuário baseado na dinâmica da digitação utilizando diferentes conjuntos de dados compostos por amostras estáticas. Para tanto, um único método de experimentação foi adotado na avaliação de seis algoritmos de AM em cinco conjuntos de dados públicos.

Os resultados obtidos nos experimentos mostraram que os desempenhos obtidos por cada algoritmo podem variar bastante quando testados em diferentes conjuntos de dados. Porém, o Random Forest se mostrou o mais regular, com EER variando 4,5% e 11,4%.

Como alternativas de trabalhos futuros estão a comparação de resultados gerados ao se alterar o valor de k, no Algoritmo 1 e o ajuste no mesmo algoritmo para que as amostras aleatórias sejam obtidas de quaisquer usuários. Além disso, o estudo comparativo iniciado no presente artigo pode ser ampliado com outros conjuntos de dados, além de

outros algoritmos e variações nas configurações de seus parâmetros. Outra possibilidade é criação de modelos de reconhecimento que possam ser construídos a partir dos dados brutos das amostras, reduzindo assim a necessidade de execução da etapa de Engenharia de Atributos no processo.

7. Agradecimentos

Este trabalho foi parcialmente apoiado pela CAPES (bolsa de estudos).

Referências

Ali, H., Wahyudi, and Salami, M. J. E. (2009). Keystroke pressure based typing biometrics authentication system by combining ANN and ANFIS-based classifiers. In *2009 5th International Colloquium on Signal Processing Its Applications*, pages 198–203.

Ali, M. L., Monaco, J. V., Tappert, C. C., and Qiu, M. (2017). Keystroke Biometric Systems for User Authentication. *Journal of Signal Processing Systems*, 86(2):175–190.

Alpar, O. (2017). Frequency spectrograms for biometric keystroke authentication using neural network based classifier. *Knowledge-Based Systems*, 116:163–171.

Alpar, O. (2018). Biometric touchstroke authentication by fuzzy proximity of touch locations. *Future Generation Computer Systems*, 86:71–80.

Alshanketi, F., Traore, I., and Ahmed, A. A. (2016). Improving Performance and Usability in Mobile Keystroke Dynamic Biometric Authentication. In *2016 IEEE Security and Privacy Workshops (SPW)*, pages 66–73.

Alsultan, A. and Warwick, K. (2013). Keystroke dynamics authentication: a survey of free-text methods. *International Journal of Computer Science Issues*, 10(4):1–10.

Antal, M. and Nemes, L. (2016). The MOBIKEY Keystroke Dynamics Password Database: Benchmark Results. In *Software Engineering Perspectives and Application in Intelligent Systems*, pages 35–46. Springer.

Antal, M. and Szabó, L. Z. (2015). An Evaluation of One-Class and Two-Class Classification Algorithms for Keystroke Dynamics Authentication on Mobile Devices. In *2015 20th International Conference on Control Systems and Computer Science*, pages 343–350.

Antal, M., Szabó, L. Z., and László, I. (2015). Keystroke dynamics on android platform. *Procedia Technology*, 19:820–826.

Banerjee, S. P. and Woodard, D. L. (2012). Biometric authentication and identification using keystroke dynamics: A survey. *Journal of Pattern Recognition Research*, 7(1):116–139.

Bhatia, A. and Hanmandlu, M. (2018). Keystroke Dynamics Based Authentication Using Possibilistic Renyi Entropy Features and Composite Fuzzy Classifier. *Journal of Modern Physics*, 9(02):112.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1):5–32.

Çeker, H. and Upadhyaya, S. (2017). Sensitivity analysis in keystroke dynamics using convolutional neural networks. In *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, pages 1–6.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27.

Cruz, M. A. S., Duarte, J. C., and Goldschmidt, R. R. (2017). Dinâmica da Digitação Aplicada à Autenticação Periódica de Usuários em Ambientes Virtuais de Aprendizagem. *Revista Brasileira de Informática na Educação - RBIE*, 2:1–30.

Deng, Y. and Zhong, Y. (2013). Keystroke dynamics user authentication based on gaussian mixture model and deep belief nets. *ISRN Signal Processing*, 2013:1–30.

Deng, Y. and Zhong, Y. (2015). Keystroke dynamics advances for mobile devices using deep neural network. *Recent Advances in User Authentication Using Keystroke Dynamics Biometrics*, 2:59–70.

Dowland, P. S. and Furnell, S. M. (2004). A long-term trial of keystroke profiling using digraph, trigraph and keyword latencies. In *IFIP International Information Security Conference*, pages 275–289.

El-Abed, M., Dafer, M., and Khayat, R. E. (2014). RHU Keystroke: A mobile-based benchmark for keystroke dynamics systems. In *2014 International Carnahan Conference on Security Technology (ICCST)*, pages 1–4.

Faceli, K., Lorena, A. C., Gama, J., and Carvalho, A. (2011). *Inteligência Artificial: Uma abordagem de aprendizado de máquina*, volume 2. LTC.

Filho, J. R. M. and Freire, E. O. (2006). On the equalization of keystroke timing histograms. *Pattern Recognition Letters*, 27(13):1440–1446.

Fischer, A. and Igel, C. (2014). Training restricted Boltzmann machines: An introduction. *Pattern Recognition*, 47(1):25–39.

Gaines, R. S., Lisowski, W., Press, S. J., and Shapiro, N. (1980). Authentication by keystroke timing: Some preliminary results. Technical report, Rand Corp Santa Monica CA.

Giot, R., El-Abed, M., and Christophe, R. (2009a). GREYC Keystroke: a Benchmark for Keystroke Dynamics Biometric Systems. In *IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS 2009)*, Washington, District of Columbia, USA. IEEE Computer Society.

Giot, R., El-Abed, M., Hemery, B., and Rosenberger, C. (2011). Unconstrained keystroke dynamics authentication with shared secret. *Computers & Security*, 30(6):427–445.

Giot, R., El-Abed, M., and Rosenberger, C. (2009b). Keystroke dynamics with low constraints SVM based passphrase enrollment. In *2009 IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems*, pages 1–6.

Giot, R., El-Abed, M., and Rosenberger, C. (2012). Web-Based Benchmark for Keystroke Dynamics Biometric Systems: A Statistical Analysis. In *2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 11–15.

Goldschmidt, R., Bezerra, E., and Passos, E. (2015). *Data Mining: conceitos, técnicas, algoritmos, orientações e aplicações*. Elsevier, Rio de Janeiro, 2. ed. edition.

Gunetti, D. and Ruffo, G. (1999). Intrusion detection through behavioral data. In *IDA*, volume 99, pages 383–394.

Han, J., Pei, J., and Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Ho, G. (2014). Tapdynamics: strengthening user authentication on mobile phones with keystroke dynamics. Technical report, Technical report, Stanford University.

Hu, J., Gingrich, D., and Sentosa, A. (2008). A k-nearest neighbor approach for user authentication through biometric keystroke dynamics. In *Communications, 2008. ICC'08. IEEE International Conference on*, pages 1556–1560.

Jain, A. K., Nandakumar, K., and Ross, A. (2016). 50 years of biometric research: Accomplishments, challenges, and opportunities. *Pattern Recognition Letters*, 79:80–105.

Karnan, M. and Akila, M. (2010). Personal Authentication Based on Keystroke Dynamics Using Soft Computing Techniques. In *2010 Second International Conference on Communication Software and Networks*, pages 334–338.

Killourhy, K. S. and Maxion, R. A. (2009). Comparing anomaly-detection algorithms for keystroke dynamics. In *Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP International Conference on*, pages 125–134.

Kobojek, P. and Saeed, K. (2016). Application of Recurrent Neural Networks for User Verification based on Keystroke Dynamics. *Journal of Telecommunications and Information Technology*, 1(3):80.

Langley, P., Iba, W., Thompson, K., and Others (1992). An analysis of Bayesian classifiers. In *Aaai*, volume 90, pages 223–228.

Lee, H., Hwang, J. Y., Kim, D. I., Lee, S., Lee, S.-H., and Shin, J. S. (2018). Understanding Keystroke Dynamics for Smartphone Users Authentication and Keystroke Dynamics on Smartphones Built-In Motion Sensors. *Security and Communication Networks*, 2018.

Lin, C.-H., Liu, J.-C., and Lee, K.-Y. (2018). On Neural Networks for Biometric Authentication Based on Keystroke Dynamics. *Sensors and Materials*, 30(3):385–396.

Loy, C. C., Lai, W. K., and Lim, C. P. (2007). Keystroke Patterns Classification Using the ARTMAP-FD Neural Network. In *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007)*, volume 1, pages 61–64.

Loy, C. C., Lim, C. P., and Lai, W. K. (2005). Pressure-based typing biometrics user authentication using the fuzzy ARTMAP neural network. In *Proceedings of the Twelfth International Conference on Neural Information Processing (ICONIP 2005)*, pages 647–652. Citeseer.

Maheshwary, S., Ganguly, S., and Pudi, V. (2017). Deep secure: A fast and simple neural network based approach for user authentication and identification via keystroke dynamics. In *IWAISe: First International Workshop on Artificial Intelligence in Security*, page 59.

Maxion, R. A. and Killourhy, K. S. (2010). Keystroke biometrics with number-pad input. In *2010 IEEE/IFIP International Conference on Dependable Systems Networks (DSN)*, pages 201–210.

Meng, Y., Wong, D. S., and Kwok, L.-F. (2014). Design of Touch Dynamics Based User Authentication with an Adaptive Mechanism on Mobile Phones. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, SAC '14, pages 1680–1687, New York, NY, USA. ACM.

Ngugi, B., Kahn, B. K., and Tremaine, M. (2011). Typing Biometrics: Impact of Human Learning on Performance Quality. *J. Data and Information Quality*, 2(2):11:1—-11:21.

O'Gorman, L. (2003). Comparing passwords, tokens, and biometrics for user authentication. *Proceedings of the IEEE*, 91(12):2021–2040.

Pavaday, N. and Soyjaudah, K. M. S. (2007). Investigating performance of neural networks in authentication using keystroke dynamics. In *AFRICON 2007*, pages 1–8.

Pisani, P. H. and Lorena, A. C. (2012). Evolutionary neural networks applied to keystroke dynamics: Genetic and immune based. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8.

Pisani, P. H. and Lorena, A. C. (2013). A systematic review on keystroke dynamics. *Journal of the Brazilian Computer Society*, 19(4):573–587.

Pisani, P. H., Lorena, A. C., and d. Carvalho, A. C. P. L. F. (2015). Ensemble of Adaptive Algorithms for Keystroke Dynamics. In *2015 Brazilian Conference on Intelligent Systems (BRACIS)*, pages 310–315.

Pisani, P. H., Lorena, A. C., and de Carvalho, A. (2018). Adaptive Biometric Systems using Ensembles. *IEEE Intelligent Systems*, page 1.

Rodrigues, R. N., Yared, G. F. G., Costa, C. R. d. N., Yabu-Uti, J. B. T., Violaro, F., and Ling, L. L. (2006). Biometric access control through numerical keyboards based on keystroke dynamics. In *International Conference on Biometrics*, pages 640–646. Springer.

Rosenblatt, F. (1962). *Principles of Neurodynamics*. Spartan Book, New York.

Salzberg, S. L. (1994). C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. *Machine Learning*, 16(3):235–240.

Sêmola, M. (2014). *Gestão da segurança da informação*, volume 2. Elsevier Brasil.

Sen, S. and Muralidharan, K. (2014). Putting #x2018;pressure #x2019; on mobile authentication. In *2014 Seventh International Conference on Mobile Computing and Ubiquitous Networking (ICMU)*, pages 56–61.

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

Shimshon, T., Moskovitch, R., Rokach, L., and Elovici, Y. (2010). Clustering di-graphs for continuously verifying users according to their typing patterns. In *2010 IEEE 26-th Convention of Electrical and Electronics Engineers in Israel*, pages 445–449.

Shinde, P., Shetty, S., and Mehra, M. (2016). Survey of Keystroke Dynamics as a Biometric for Static Authentication. *International Journal of Computer Science and Information Security*, 14(4):203.

Sim, T. and Janakiraman, R. (2007). Are digraphs good for free-text keystroke dynamics? In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–6. IEEE.

Sulong, A., Wahyudi, and Siddiqi, M. U. (2009). Intelligent keystroke pressure-based typing biometrics authentication system using radial basis function network. In *2009 5th International Colloquium on Signal Processing Its Applications*, pages 151–155.

Teh, P. S., Teoh, A. B. J., and Yue, S. (2013). A survey of keystroke dynamics biometrics. *The Scientific World Journal*, 2013:1–30.

Teh, P. S., Zhang, N., Teoh, A. B. J., and Chen, K. (2016). A survey on touch dynamics authentication in mobile devices. *Computers & Security*, 59:210–235.

Thanganayagam, R. and Thangadurai, A. (2015). Fusion approach on keystroke dynamics to enhance the performance of password authentication. In *2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pages 1–6.

Thierry, E. and Chuan, C. (2018). One-class SVM for biometric authentication by keystroke dynamics for remote evaluation. *Computational Intelligence*, 34(1):145–160.

Wu, J. and Chen, Z. (2015). An Implicit Identity Authentication System Considering Changes of Gesture Based on Keystroke Behaviors. *International Journal of Distributed Sensor Networks*, 11(6).

Yu, E. and Cho, S. (2003). Novelty Detection Approach for Keystroke Dynamics Identity Verification. In Liu, J., Cheung, Y.-m., and Yin, H., editors, *Intelligent Data Engineering and Automated Learning*, pages 1016–1023, Berlin, Heidelberg. Springer Berlin Heidelberg.