Autenticação de alunos utilizando Dinâmica de Digitação e Redes Neurais Profundas em sistemas Juiz On-line

Ronem Matos Lavareda Filho[1], **Juan Gabriel Colonna**[1], **David B F de Oliveira**[1], **Edwin Juan L B Monteiro**[1], **Paulo Henrique Nellessen Gonçalves**[1]

[1]Instituto de Computação – Universidade Federal do Amazonas (ICOMP - UFAM)
Caixa Postal 15.064 – 91.501-970 – Manaus – AM – Brazil

{ronem,juancolonna,david, edwin, paulo}@icomp.ufam.edu.br

Abstract. *In online code verification systems student authentication is normally done only at the beginning of the login session. In these kind of systems sharing login and password information causes authenticity issues or identity spoofing. In this research, we present a method for authentication verification using the dynamics of typing. This technique does not require any explicit user action or additional hardware, only a keyboard. We use a Deep Neural Network architecture that was designed to automatically learn to recognize students' typing patterns in exercises and assessments that took place in these systems. To validate the method, data from the typing dynamics of 42 students in the CodeBench online code system were used. Then, several experiments were performed that demonstrated the effectiveness of the proposed method.*

Resumo. *Em sistemas juízes on-line a autenticação do aluno é feita apenas no início da sessão de login, o que pode ocasionar problemas de autenticidade ou falsificação de identidade. Esta pesquisa apresenta um método de autenticação de alunos em sistemas Juiz on-line utilizando a dinâmica de digitação após o login sem a necessidade de utilizar hardware adicional. Para isso, projetou-se uma arquitetura de Rede Neural Profunda que aprende e reconhece de forma automática os padrões de digitação dos alunos em exercícios de programação, considerando a evolução temporal de aprendizagem dos alunos. Para validar o método, foram utilizados dados de dinâmica de digitação de 42 alunos no sistema juiz on-line CodeBench. Por fim, foram realizados experimentos que demonstraram a eficácia do método proposto.*

1. Introdução

Nos últimos anos os cursos de computação atraíram um número significativo de estudantes [Hao et al. 2019]. Dentre as disciplinas ofertadas nesses cursos, as de programação estão entre as mais fundamentais, sendo essenciais para avançar no curso e na própria carreira profissional [Chaves 2014]. Diante desse contexto, uma das principais ferramentas de ensino-aprendizagem para auxiliar professores e alunos dessas disciplinas são os sistemas juízes *on-line* (JO). Esses sistemas são normalmente adotados em disciplinas introdutórias de programação, pois auxiliam na melhoria de habilidades dos alunos iniciantes e permitem que eles desenvolvam seus próprios programas como resposta a determinados exercícios solicitados pelos professores [Lavareda Filho et al. 2020].

Apesar das vantagens promovidas pelos JOs, um dos principais impasses de sua adoção é quanto à autenticidade de seus usuários. Por exemplo, o registro de autenticação

DOI: 10.5753/sbie.2022.225779

de alunos em sistemas juízes *on-line* é feito apenas no início da sessão de *login* mediante a digitação de uma senha. Apesar desse ser o método mais usado de autentificação de usuários, ele apresenta vulnerabilidades [Silva et al. 2019], pois a genuinidade do aluno logado não é verificada em nenhum outro momento após a efetuação do *login*. Desta forma, o compartilhamento de informações de *login* e senha ocasionam problemas de autenticidade ou falsificação de identidade [Ullah et al. 2019]. Ou seja, alguns alunos poderiam pedir para terceiros fazerem suas atividades sem serem descobertos. Nesta pesquisa busca-se amenizar esse tipo de problema, utilizando autenticação baseada na dinâmica de digitação dos alunos.

Segundo [Feher et al. 2012], a dinâmica de digitação tornou-se a principal fonte biométrica comportamental para resolver problemas de autenticação de usuários em sistemas computacionais. O método de autenticação através da técnica da dinâmica de digitação é fundamentado na ideia de que cada indivíduo digita no teclado de maneira única [Acien et al. 2020]. As vantagens dessa técnica incluem seu baixo custo, uma vez que não requer nenhum *hardware* adicional. Além disso, por ser um método de autenticação não intrusivo, permite que os usuários sejam autenticados de maneira silenciosa e contínua, permitindo que apenas usuários genuínos façam uso contínuo do ambiente. Este tipo de autenticação, embora empregado em vários contextos, é especialmente conveniente para sistemas *on-line*, como ferramentas de Educação a Distância (EaD) [Longi et al. 2015], sendo possível verificar se o aluno que conclui um trabalho da disciplina é o mesmo que se matriculou para fazer o curso.

Diferentes algoritmos podem ser aplicados no campo da autenticação biométrica. Entre eles estão os algoritmos de aprendizagem profunda (*Deep Learning* - DL) [Chong et al. 2019]. A principal vantagem de DL é a sua capacidade de criar modelos capazes de analisar e aprender o comportamento humano a partir de conjuntos de dados. Logo, esta pesquisa propõe o desenvolvimento e avaliação de um método de autenticação baseado na dinâmica de digitação e DL, com intuito de autenticar de forma não intrusiva alunos em sistemas JO.

2. Trabalhos Relacionados

A literatura apresenta diferentes abordagens para autenticação de usuários de sistemas utilizando a dinâmica de digitação [Feher et al. 2012], entretanto, não foram encontrados trabalhos que abordem essa técnica para a autenticação de alunos em sistemas JOs. Todavia, o estudo de [Longi et al. 2015] mostra que alunos podem ser identificados a partir da dinâmica de digitação em sessões de disciplinas de programação. Para isso, os autores utilizaram o algoritmo de classificação kNN e obtiveram cerca de 95% de precisão na identificação dos alunos. O modelo de [Longi et al. 2015] foi examinado em mais detalhes por [Peltola et al. 2017], que descobriram que é possível identificar alunos em diferentes contextos, por exemplo durante a escrita de textos ou códigos de programação. Em outro estudo, [Byun et al. 2020] propuseram uma abordagem utilizando o algoritmo de classificação *Random Forest*. Nesse estudo, foi conduzido um conjunto de experimentos em uma classe de alunos universitários, mostrando que é possível detectar casos de fraude com maior precisão do que os estudos anteriores. Como resultados, obtiveram a taxa FAR (*False Acceptance Rate*) de 15,6% e a taxa FRR (*False Rejection Rate*) de 14,1%.

Nos trabalhos citados acima, os modelos precisam ser treinados com grande quantidade de dados de cada usuário e, caso precise adicionar um novo usuário ao sistema, será necessário não apenas de novos dados, mas também atualizar o modelo de classificação. Além disso, exigem muito conhecimento específico sobre o domínio em que serão aplicados, consequentemente, fazem uso de extração manual de *features*. Para resolver esses problemas, trabalhos recentes têm adotado *Deep Learning*, mais especificamente os modelos de redes neurais siamesas que visa, sem grande número de amostras, aprender de forma automática (sem extração manual de *features*), a partir dos dados de entrada, a melhor maneira de representar os dados para a tarefa de classificação. Por exemplo, em [Giot e Rocha 2019], foi analisada a viabilidade do uso de redes neurais siamesas para autenticação de usuários utilizando dinâmica de digitação em textos de tamanho fixo. Nos experimentos, o método proposto é comparado a vários modelos da mesma linha na literatura. Seu *Equal Error Rate* (EER) supera o seu melhor trabalho de referência [Killourhy e Maxion 2009] em 28% em um contexto único (com poucas amostras) e 31% ao usar 200 amostras. Em [Acien et al. 2020], utilizaram uma rede neural recorrente siamesa e dinâmica de digitação para a autenticação de 100.000 usuários. Nos experimentos, o modelo obteve uma EER de 4,8% usando apenas 5 sequências de *features* da dinâmica da digitação e 1 sequência de teste por usuário com 50 pressionamentos de tecla por sequência.

O diferencial deste trabalho em relação aos apresentados é que, além de propor um método de autenticação para sistemas JOs utilizando a dinâmica de digitação, ele também projeta um modelo de rede convolucional siamesa que extrai automaticamente, a partir de dados brutos, novas *features* necessárias para o reconhecimento dos alunos. Além disso, nossa abordagem agrega a informação dos caracteres pressionados junto à dinâmica de digitação, com a finalidade de reconhecer melhor cada padrão biométrico. Por último, utilizamos protocolos específicos para sistemas JOs onde é levada em consideração a evolução temporal da aprendizagem dos alunos, investigando se seus padrões biométricos mudam no decorrer do tempo à medida em que eles realizam as atividades no JO.

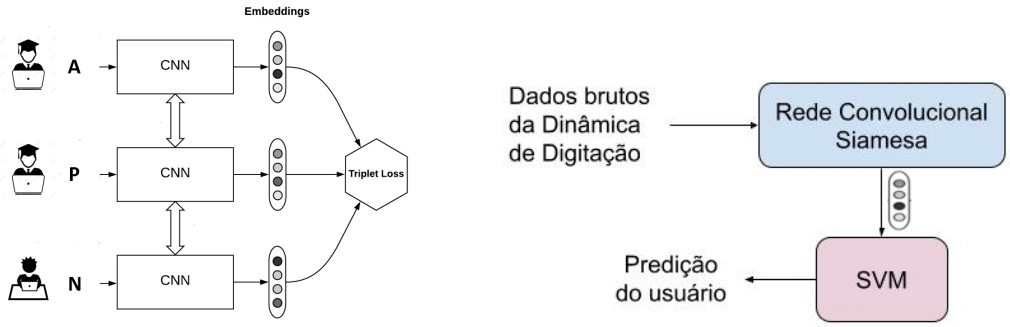
3. Solução Proposta

O método de autenticação de alunos para sistemas juiz *on-line* proposto neste trabalho utiliza a dinâmica de digitação e aprendizagem profunda. A metodologia empregada está dividida em quatro etapas: coleta dos dados, pré-processamento, treinamento e autenticação do usuário.

3.1. Coleta de dados e Pré-Processamento

Nesta etapa foi realizada a coleta dos dados, que inicia a partir do momento em que o aluno faz o *login* e começa a codificar uma solução para um dado exercício de programação através do ambiente de desenvolvimento integrado (IDE) do juiz *on-line*, e prosseguirá até o término da sessão.

Durante a coleta de dados foram considerados quase todos eventos de teclado digitados pelos alunos, exceto eventos de copiar (Ctrl+c) e colar (Ctrl+v). Além disso, foram eliminados momentos em que os alunos passaram mais de duas horas sem interagir com a IDE do Juiz *on-line*. Alunos que desistiram da disciplina nos primeiros dias de aula foram desconsiderados. Após essa limpeza restaram 158 alunos, dos quais foram selecionados 42, por fazerem parte da mesma turma.



(a) Arquitetura de rede neural proposta. (b) Pipeline da abordagem proposta.

Figura 1. Método proposto. Os vetores de *embeddings* extraídos pela rede siamesa são utilizados pelo classificador SVM.

As *features* mais comuns dos dados capturados são: 'Down' (instante de tempo capturado no momento em que uma tecla foi pressionada), 'Up' (instante de tempo capturado no momento quando a tecla foi solta/liberada), o 'TP' (tempo de pressão de uma tecla) que é obtido pela subtração de Down pelo Up (ou seja, TP = Up-Down) da mesma tecla e, por fim, as latências entre duas teclas (DD e UD). A latência 'UD' é o intervalo entre o momento que uma tecla é liberada e a tecla seguinte é pressionada, enquanto que a latência 'DD' é obtida ao subtrair o tempo em que duas teclas consecutivas foram pressionadas [Lavareda Filho et al. 2020]. Para os experimentos realizados nesta pesquisa utilizamos as *features*: TP, latência entre teclas (DD e UD) e caracteres brutos (key) digitados no JO, como *features* biométricas dos alunos.

Quanto aos caracteres (key) digitados, foram selecionados exclusivamente os 110 mais frequentes em todos os programas. A hipótese é que, quando acrescentamos os caracteres pressionados à dinâmica de digitação, estes agreguem informação e melhorem o reconhecimento dos alunos. Além disso, antes da construção do modelo de aprendizado profundo, foi preciso codificar todos os atributos da variável key, que eram categóricos, em valores numéricos, pois as redes neurais trabalham exclusivamente com dados numéricos. Uma vez coletados os dados dos alunos, estes são normalizados (entre zero e um) e segmentados para envio à rede neural responsável pelo treinamento. Vale destacar que devido a coleta dos dados para verificação ser feita de forma contínua enquanto o aluno programava, foi necessário particionar os registros (*logs*) em segmentos de tamanhos fixos (ou janelas) contendo subsequências de caracteres digitados.

3.2. Treinamento do Modelo

O modelo principal utilizado é uma rede neural siamesa. Nossa hipótese é que redes siamesas treinadas com a função de perda *Triplet Loss* geram vetores de *embeddings* que irão ajudar na identificação dos alunos. Para que a função *Triplet Loss* guie o treinamento da rede siamesa são necessárias três entradas: **Âncora (A)**: uma amostra pertencente a um usuário (usuário genuíno), **Positiva (P)**: uma amostra semelhante à amostra âncora (usuário genuíno), e **Negativo (N)**: uma amostra pertencente a um usuário impostor (Figura 1(a)).

O objetivo da *Triplet Loss* é minimizar a distância entre *embeddings* da mesma classe (**A-P**) enquanto maximiza a distância entre *embeddings* de classes diferentes (**N-**

P). No caso desta pesquisa, pressupomos como parte de nossa hipótese que há uma menor distância entre os *embeddings* **A** e **P**, se pertencentes a um mesmo aluno genuíno, quando comparada à distância entre **P** e **N**. Caso contrário, isso pode indicar aluno impostor. Entretanto, gerar todas as triplas de **A**, **P** e **N** possíveis resultaria em uma abordagem com custo exponencial em relação à quantidade de exemplos de treinamento. Nesse caso, é crucial selecionar triplas **A**, **P** e **N** que possam contribuir com melhores informações e acelerar o treinamento do modelo. Nesta pesquisa, é utilizada a estratégia de seleção de triplas conhecida como *Triplet semi-hard*, que emprega uma estratégia de escolha em tempo de treinamento, apresentando à rede exemplos não tão difíceis que colapsem o treinamento do modelo, nem tão fáceis que evitem ajustar o modelo.

Uma vez treinado o modelo de rede siamesa, podemos executar a extração de *embeddings*, conseguindo mapear as janelas de digitação com trechos de programas em vetores densos com os quais podemos treinar um classificador que indica as chances de uma amostra pertencer ao mesmo aluno (Figura 1(b)). A arquitetura rede neural convolucional siamesa é composta por três sub-redes CNN 1D idênticas, que possuem os mesmos parâmetros. A entrada *input* consiste em segmentos de amostras da dinâmica de digitação dos alunos que serão mapeados no espaço de *embeddings* aprendido pela rede.

As sub-redes CNNs 1D, possuem diversas camadas convolucionais responsáveis pela extração automática de novas *features* geradas a partir dos dados brutos da digitação (TP, DD, UD e key) e consideradas relevantes para a identificação do aluno. Assim, não há necessidade de descoberta manual de *features*, uma vez que o aprendizado da representação dos dados ocorre de forma automática. Nossa arquitetura possui *kernels* de convolução de uma dimensão com tamanho 3 que extraem novas sequências de *features* combinando as entradas das camadas anteriores. A técnica de *Batch Normalization* também foi adotada para incorporar regularização no modelo. É usada a *ReLu* como função de ativação. Finalmente, uma camada *Flatten* foi adotada para converter as sequencias finais em vetores de *embeddings*. O treinamento do modelo é guiado pela função de perda baseada na distância L_2 entre os vetores de *embeddings*.

3.3. Autenticação do aluno

Após o treinamento do modelo gerado na etapa anterior é concatenado um classificador *shallow* para verificar a autenticidade do aluno. Conforme ilustrado na Figura 1(b), a rede siamesa representa cada janela de digitação com um vetor denso. Um classificador do tipo Máquina de Vetores de Suporte (SVM) com decomposição um-contra-todos (One-vs-all) recebe os vetores e faz as previsões, ou seja, determina se um novo vetor denso não contido no conjunto de treinamento pertence a algum aluno.

Para simular uma situação real, previamente separamos um subconjunto dos dados coletados para teste. As métricas FAR e FRR são calculadas durante o teste. Desta forma, o modelo de autenticação precisa ser treinado previamente, pois o mesmo será utilizado sempre que for necessária a autenticação dos alunos nas novas atividades.

4. Experimentos e Resultados

Para conduzir os experimentos, foram consideradas duas bases de dados. A Base I é pública e altamente citada em pesquisas de dinâmica de digitação, chamada de *CMU*

Keystroke Dynamics – Benchmark Data Set[1]. Essa base de dados contém dados de 51 pessoas distintas que foram convidadas a digitar a senha ".tie5Roanl" 400 vezes cada uma, em 8 sessões de 50 vezes. Foi dado um dia de folga entre as sessões com objetivo de capturar a variação de padrões de digitação ao longo de dias. Nessa base de dados, cada sessão de coleta capturou três tipos de *features* oriundas de textos de tamanho fixo (senha): o tempo de pressionamento (TP) e as duas latências entre as teclas (DD e UD).

A Base II foi coletada nas quatro primeiras semanas de aula de sete turmas da disciplina de Introdução à Programação de Computadores da Universidade Federal do Amazonas. Essas turmas foram ofertadas no segundo semestre de 2019 para diferentes cursos de graduação. Neste período, os estudantes resolviam problemas de programação diretamente na IDE do juiz *on-line* Codebench [2]. Os conteúdos estudados nas quatro primeiras semanas de aula foram: (a) variáveis e (b) estrutura de programação sequencial. Além disso, os alunos tiveram acesso a 5 atividades (listas de exercícios) e foram submetidos a uma avaliação no final da quarta semana. Para a aquisição dos dados da dinâmica de digitação de cada aluno, ao aluno efetuar o *login* e acessar a IDE do JO, um componente de *software* iniciou a coleta das ações do teclado desempenhadas dentro da IDE, na medida em que os alunos desenvolviam suas soluções para os exercícios de programação disponibilizados pelos professores.

4.1. Protocolos de treino e validação

Para avaliar diferentes aspectos da solução proposta foram considerados quatro protocolos distintos de separação de dados, sendo denominados de Protocolos I, II, III e IV.

No **Protocolo I**, os dados (segmentos S_n) de cada uma das cinco atividades realizadas no JO dos 42 alunos são separados aleatoriamente em 70% para treino e 30% para teste. Para efetuar essa divisão foi utilizada a biblioteca *scikit-learn* da linguagem *Python*. Este protocolo é utilizado na maioria dos trabalhos relacionados. Entretanto, não é recomentado para sistemas JOs, uma vez que o mesmo não leva em consideração a ordem cronológica da realização das atividades, nem a evolução dos padrões biométricos. Com a utilização deste Protocolo, podemos investigar como a presença de amostras (segmentos) das mesmas atividades nos conjunto de treino e teste impactam na acurácia.

No **Protocolo II**, as cinco atividades $Ativ_n = \{Ativ_1, Ativ_2, Ativ_3, Ativ_4, Ativ_5\}$ realizadas no JO pelos 42 alunos são distribuídas de forma aleatória para treino e para teste. Entretanto, diferente do protocolo anterior, segmentos de dados das atividades separadas para o treinamento não são utilizados no teste. Através deste Protocolo, é possível investigar como a distribuição de amostras de atividades diferentes nos conjunto de treino e teste impactam na acurácia em relação ao "Protocolo I".

O **Protocolo III** possui configurações específicas para sistemas JOs onde considera-se a evolução temporal da aprendizagem dos alunos. Este protocolo separa primariamente todos os dados da primeira atividade ($Ativ_1 = \{s_1, s_2, \ldots, s_n\}$) de cada aluno para treinamento e, os segmentos da próxima atividade realizada ($Ativ_2 = \{s_1, s_2, \ldots, s_n\}$) para teste (autenticação). Em seguida, é calculada a acurácia do modelo. Incrementalmente, a próxima etapa separa todos os dados da primeira ($Ativ_1 = \{s_1, s_2, \ldots, s_n\}$) e segunda atividade ($Ativ_2 = \{s_1, s_2, \ldots, s_n\}$) para treino e, os dados

[1]https://www.cs.cmu.edu/ keystroke/
[2]https://codebench.icomp.ufam.edu.br/

da próxima atividade ($Ativ_3 = \{s_1, s_2, \ldots, s_n\}$) para teste, e assim sucessivamente, até que todas as atividades tenham sido utilizadas para treinamento, exceto a última atividade ($Ativ_5$), desta forma, considerando o aspecto temporal.

Por fim, no **Protocolo IV** são separados todos os dados da primeira atividade ($Ativ_1 = \{s_1, s_2, \ldots, s_n\}$) para treinamento e os dados da última atividade ($Ativ_5 = \{s_1, s_2, \ldots, s_n\}$) para teste. Em seguida, é calculada a acurácia do modelo. Incrementalmente, a próxima etapa separa todos os dados da primeira ($Ativ_1 = \{s_1, s_2, \ldots, s_n\}$) e segunda atividade ($Ativ_2 = \{s_1, s_2, \ldots, s_n\}$) para treino e, os dados da última atividade ($Ativ_5 = \{s_1, s_2, \ldots, s_n\}$) para teste, e assim sucessivamente, sempre considerando o aspecto temporal. Este protocolo é útil para simular a aplicação de uma prova que neste exemplo seria a quinta atividade ($Ativ_5$).

Destacamos que, a hipótese avaliada nestes dois últimos protocolos é que o padrão de digitação dos alunos muda conforme eles começam a se familiarizar com a linguagem e com a IDE de programação, assim, quanto mais habituados à linguagem os alunos se tornam mais rápidos ao digitar comandos que aparecem com maior frequência nos programas. Por exemplo, digitar sucessivamente o comando "*print()*" leva a uma evolução no padrão de digitação, e portanto altera a biometria comportamental.

4.2. Resultados experimentais I

Um dos principais fatores que influenciam o desempenho de um algoritmo de autenticação é o tamanho das sequências [Xiaofeng et al. 2019]. Assim, o primeiro experimento foi conduzido para avaliar os tamanhos dos segmentos enviados à rede neural. Foram avaliados os seguintes tamanhos de segmentos fixos: *S* = {10, 30 e 50}. Por exemplo, *S* = 10 significa que foi utilizado um segmento com 10 pressionamentos consecutivos das teclas. Foram utilizadas janelas deslizantes com 50% de sobreposição temporal (*overlap*) e considerando somente dados da Base II. Para esse experimento, foram selecionadas as *features* da dinâmica de digitação TP, DD e UD e o protocolo I (padrão) de separação de dados.

Tabela 1. Acurácia e a Taxa de Erro (EER %) alcançadas para diferentes tamanhos de segmentos.

Tamanho Segmento	Acurácia	EER
10	78,4	0,042
30	**81,1**	**0,021**
50	72,7	0,047

Tabela 2. Resultados obtidos quando são adicionando os caracteres pressionados ao conjunto de *features*.

Features	Acurácia	EER
TP, DD e DU	88,1	0,021
TP, DD, UD + Key	**98,0**	**0,003**

A Tabela 1 mostra a acurácia alcançada para diferentes tamanhos de segmentos. Conforme podemos observar, os melhores resultados são alcançados para segmentos fixos de tamanho 30. Também podemos observar que para sequências de *S* = 10 e 50 não há melhora significativa nos resultados. Neste experimento não foram utilizados os caracteres digitados.

4.3. Experimentos II

Neste experimento, foi verificado o impacto da incorporação dos caracteres (key) digitados às *features* na entrada da rede siamesa. Para isso, foram utilizados segmentos de

tamanho 30, janelas deslizantes com 50% de *overlap*, protocolo I de separação de dados e considerando somente dados da base II. Como citado anteriormente, a biometria tradicional considera apenas os tempos de pressionamento, tornando difícil para a rede neural distinguir se letras diferentes, mas com latências parecidas, fazem parte do perfil biométrico do aluno, uma vez que existem muitas letras e ordens que se repetem em programação. A Tabela 2 apresenta os resultados quando utilizada somente as *features* da dinâmica de digitação e quando acrescentado informação dos caracteres pressionados (key). Observamos que os resultados melhoram consideravelmente, ou seja, a informação dos caracteres agrega à dinâmica de digitação e melhora a biometria comportamental.

4.4. Experimentos III

Neste experimento comparamos o modelo proposto com trabalho de [Giot e Rocha 2019], onde os autores desenvolveram um modelo de rede neural siamesa que tem como base uma rede MLP (*Multi-layer perceptron*). Nós escolhemos uma rede CNN 1D pelo fato das entradas dos caracteres digitados possuírem uma ordem temporal (uma série de tempo), característica ignorada pelas camadas MLP. Além disso, neste trabalho foi utilizado textos livres, ou seja, qualquer texto digitado pelo aluno, não importando a quantidade de caracteres, enquanto que os autores utilizaram textos de tamanho fixo como, por exemplo, uma senha.

Primeiramente, utilizamos a Base II, considerando o Protocolo I para treinamento e teste (protocolo padrão), de forma a realizar comparações justas. Os dois modelos são treinados com quantidade de usuários iguais (40 usuários) e segmentos fixos de tamanhos iguais ($S = 30$). Na Tabela 3 podemos observar os resultados, onde a rede convolucional 1D siamesa extraiu *features* robustas dos dados brutos da dinâmica de digitação dos alunos, levando ao resultado superior no reconhecimento dos usuários. Concluímos também que a sua capacidade de processar vetores de *features* que formam séries de tempo não pode ser ignorada pelas soluções propostas ou existentes.

Tabela 3. Acurácias e Taxas de erro (%) comparadas contra o *baseline.*

	Arquitetura	Acurácia	EER
Base I	[Giot e Rocha 2019]	95,0	0,010
	CNN 1D Proposta	**99,4**	**0,002**
Base II	[Giot e Rocha 2019]	92,6	0,017
	CNN 1D Proposta	**98,0**	**0,003**

Tabela 4. Acurácias e Taxas de Erro (%) usando diferentes Protocolos de separação de dados.

Protocolos	**Acurácia**	**EER**
Protocolo I	98,1	0,003
Protocolo II	80,4	0,040

Em seguida, comparamos as duas arquiteturas, desta vez utilizando a Base I, também utilizada por [Giot e Rocha 2019]. Conforme podemos observar na Tabela 3, nossa proposta supera o estado da arte. Novamente, os melhores resultados são alcançados para segmentos fixos de $S = 30$, com acurácia de 99,4%, taxa de erro de 0,002% e considerando o Protocolo I. De acordo com a nossa revisão bibliográfica, o presente trabalho supera o estado da arte para métodos de autenticação utilizando textos fixos da dinâmica de digitação e redes siamesas.

4.5. Experimentos IV

Neste experimento verificamos como a distribuição de amostras da mesma atividade presentes nos conjuntos de treino e teste impactam o reconhecimento. Inicialmente executa-

mos e comparamos os resultados utilizando os Protocolo I e Protocolo II. Cabe ressaltar que no Protocolo I há segmentos da mesma atividade nos conjuntos de treino e teste, sendo o protocolo menos recomendado, porém utilizado ainda na literatura. Em contrapartida, no Protocolo II não há segmentos da mesma atividade nos treino e teste, e a divisão dos dados foi feita de forma manual, embora a seleção das atividades esteja na forma aleatória. Os resultados foram obtidos utilizado os mesmos parâmetros do experimento anterior.

Conforme mostrado na Tabela 4, foi possível constatar que a distribuição errada dos dados (utilização do protocolo padrão) durante a separação impactou em 18%. Esta diferença indica que há um vazamento de informações durante o treino ao se utilizar o Protocolo I.

4.6. Experimentos V

Neste experimento, verificamos a viabilidade do método proposto levando em consideração a evolução temporal da aprendizagem dos alunos, investigando se os padrões biométricos dos mesmos mudam a medida em que eles realizam as atividades. Desta vez, utilizamos o Protocolo III (descrito na seção 4.1) que possui configurações específicas para JOs. Para isso, foram utilizados os mesmos parâmetros e a Base II.

Tabela 5. Resultados utilizando o Protocolo III.

Atividades (treino)	Atividade (teste)	Acurácia	EER
1	2	76,1	0,044
1, 2	3	82,2	0,038
1, 2, 3	4	82,7	0,038
1, 2, 3, 4	5	83,1	0,037
Média		**81,0**	**0,039**

Tabela 6. Resultados utilizando o Protocolo IV.

Atividades (treino)	Atividades (teste)	Acurácia	EER
1	5	67,0	0,096
1, 2	5	75,1	0,045
1, 2, 3	5	81,7	0,039
1, 2, 3, 4	5	82,9	0,038
Média		**76,6**	**0,054**

Os resultados alcançados para diferentes quantidades de atividades utilizadas para treinamento do modelo são apresentados na Tabela 5. Como podemos observar, os melhores resultados são alcançados quando utilizados a quantidade máxima possível de atividades para treinamento. Entretanto, caso sejam utilizadas unicamente amostras da primeira e segunda atividades para treinamento o método reconheceria os alunos genuínos com 82,2% de acurácia. Assim, podemos definir o perfil biométrico uma quantidade mínima de atividades necessárias para criar o modelo e ajustá-lo.

4.7. Experimentos VI

Neste experimento utilizamos o Protocolo IV de separação de dados, onde cada atividade foi utilizada para treinamento, exceto os segmentos da atividade 5 que sempre foram utilizados para teste simulando uma avaliação. Os resultados da Tabela 6 mostram que se utilizarmos até a última atividade anterior à avaliação para treinar o modelo, melhor será o reconhecimento dos alunos.

5. Conclusões e Trabalhos Futuros

Neste artigo, apresentamos um método para autenticação de alunos em sistemas juízes *on-line* utilizando dinâmica de digitação e aprendizagem profunda. Os resultados encontrados neste trabalho indicam que as redes convolucionais siamesas 1D, tendo como

entrada caracteres brutos da dinâmica de digitação para autenticação de alunos, são capazes de alcançar uma acurácia de 98% durante o processo de reconhecimento dos usuários (alunos) em sistemas JOs. Além disso, os resultados mostram que a incorporação de informação dos caracteres pressionados agrega informação à dinâmica de digitação e melhora consideravelmente o reconhecimento dos alunos.

Foi provado que a distribuição de amostras das mesmas atividades nos conjunto de treino e teste, como utilizado no protocolo de separação de dados tradicional, impactam na acurácia, possivelmente houve um vazamento de informações entre os conjuntos de treino e teste. Por esse motivo, foi elaborada uma metodologia de avaliação que considerou a evolução temporal da aprendizagem dos alunos. Como pode ser observado nos resultados das Tabelas 5 e 6, há indícios que os padrões biométricos dos alunos mudam com o decorrer das atividades. À medida em que as atividades selecionadas para treinamento estão mais distantes no aspecto temporal das separadas para teste, mais difícil resulta o reconhecimento, fundamentando a hipótese de que o ritmo de digitação do aluno pode ter sofrido variações com o tempo, uma vez que o aluno(a) está se adaptando à linguagem de programação. Assim, este trabalho demonstrou a viabilidade de uma nova abordagem para autenticar de forma não intrusiva alunos em sistemas juiz *on-line*.

Com a utilização do método proposto, pretende-se diminuir as ocorrências de fraude por parte de alunos em exercícios e avaliações que ocorrem nos sistemas juiz *on-line* principalmente de forma remota, uma vez que será possível verificação a autenticidade durante toda a sessão. Monitorando e analisando os padrões habituais de digitação iremos saber se houve intercambio de *login* e senha para obter ajuda de terceiros ao fazer as atividades. Assim, a utilização do método proposto impacta positivamente o âmbito educacional, garantindo às instituições e professores uma segurança complementar contra fraudes, além de testificar a integridade acadêmica do aluno em todas as atividades. O método proposto também poderá ser utilizado para informar o professor ou monitor da turma, antes mesmo que o usuário impostor seja bloqueado, no qual poderá enviar um alerta ao aluno, o que poderia mudar a percepção dos alunos em relação a fraudes.

Por fim, a arquitetura de Rede Neural proposta poderia ser utilizada como um extrator de *features* genérico para classificar novos alunos, novas turmas, sem que seja necessário atualizá-la (retreino), em uma configuração *One-shot Learning*. Esta hipótese sustenta uma observação para um trabalho futuro, onde um modelo treinado em uma turma seria avaliado em outra turma. Finalmente, também pretendemos avaliar outras arquiteturas de redes neurais, aumentar a quantidade de participantes e avaliar as regras de decisão baseadas no nível de confiança da classificação.

6. Agradecimentos

Esta pesquisa, realizada no âmbito do Projeto Samsung-UFAM de Ensino e Pesquisa (SUPER), nos termos do artigo 48 do Decreto nº 6.008/2006 (SUFRAMA), foi parcialmente financiada pela Samsung Eletrônica da Amazônia Ltda. Nos termos da Lei Federal nº 8.387/1991, por meio dos convênios 001/2020 e 003/2019, firmados com a Universidade Federal do Amazonas e a FAEPI, Brasil. Além de que, o presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Referências

Acien, A., Monaco, J. V., Morales, A., Vera-Rodriguez, R., e Fierrez, J. (2020). Typenet: Scaling up keystroke biometrics. *arXiv preprint arXiv:2004.03627*.

Byun, J., Park, J., e Oh, A. (2020). Detecting contract cheaters in online programming classes with keystroke dynamics. In *Proceedings of the Seventh ACM Conference on Learning@ Scale*, pages 273–276.

Chaves, J. O. M. (2014). Uma ferramenta de apoio ao processo de ensino-aprendizagem em disciplinas de programação de computadores por meio da integração dos juízes online ao moodle.

Chong, P., Elovici, Y., e Binder, A. (2019). User authentication based on mouse dynamics using deep neural networks: A comprehensive study. *IEEE Transactions on Information Forensics and Security*.

Feher, C., Elovici, Y., Moskovitch, R., Rokach, L., e Schclar, A. (2012). User identity verification via mouse dynamics. *Information Sciences*, 201:19–36.

Giot, R. e Rocha, A. (2019). Siamese networks for static keystroke dynamics authentication. In *IEEE International Workshop on Information Forensics and Security*.

Hao, Q., Smith IV, D. H., Iriumi, N., Tsikerdekis, M., e Ko, A. J. (2019). A systematic investigation of replications in computing education research. *ACM Transactions on Computing Education (TOCE)*, 19(4):42.

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

Lavareda Filho, R. M., Colonna, J. G., e Oliveira, D. B. F. (2020). Autenticação contínua de alunos utilizando biometria comportamental em ambiente juiz on-line. In *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, pages 1193–1202. SBC.

Longi, K., Leinonen, J., Nygren, H., Salmi, J., Klami, A., e Vihavainen, A. (2015). Identification of programmers from typing patterns. In *Proceedings of the 15th Koli Calling Conference on Computing Education Research*, pages 60–67. ACM.

Peltola, P., Kangas, V., Pirttinen, N., Nygren, H., e Leinonen, J. (2017). Identification based on typing patterns between programming and free text. In *Proceedings of the 17th Koli Calling International Conference on Computing Education Research*, pages 163–167.

Silva, R. J. H., Pazoti, M. A., da Silva, F. A., Pereira, D. R., e de Almeida, L. L. (2019). Autenticaçao biométrica para sistemas por meio da dinâmica da digitaçao. In *Colloquium Exactarum. ISSN: 2178-8332*, volume 11, pages 26–33.

Ullah, A., Xiao, H., e Barker, T. (2019). A dynamic profile questions approach to mitigate impersonation in online examinations. *Journal of Grid Computing*, 17(2):209–223.

Xiaofeng, L., Shengfei, Z., e Shengwei, Y. (2019). Continuous authentication by free-text keystroke based on cnn plus rnn. *Procedia computer science*, 147:314–318.