Autenticação contínua de alunos utilizando biometria comportamental em ambiente Juiz On-line

Ronem Matos Lavareda Filho[1], **Juan Gabriel Collona**[1], **David B F de Oliveira**[1]

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

{ronem,juan,david}@icomp.ufam.edu.br

Abstract. *Student authentication in online judge environments is only verified at the beginning of the login session by password validation. However, in the case of exercises and assessments made in environments like this, there is a need for continuous and non-intrusive verification of the genuineness of students throughout the session and not just at login. In this article, we present a method to continuously authenticate students in online judge systems using behavioral biometrics, using specifically the coding dynamics. For this, a Siamese convolutional neural network architecture was designed that seeks to learn automatically, from a limited amount of raw data of the student's typing dynamics. In the experiment, our model achieves a recognition accuracy of 97.2%.*

Resumo. *Como na maioria dos sistemas computacionais, a autenticação do aluno em ambientes de juiz on-line é feita somente no início da sessão mediante a digitação de um login e senha. Entretanto, no caso de exercícios e avaliações feitas em ambientes como esse, há a necessidade da verificação contínua e não-intrusiva da identidade dos estudantes durante toda a sessão, não apenas no login. Neste artigo, apresentamos um método de autenticação contínua dos alunos em ambientes de Juiz on-line utilizando biometria comportamental, mas especificamente a dinâmica de codificação. Para isso, foi projetada uma arquitetura de rede neural convolucional siamesa que busca aprender de maneira automática a representação necessária para o reconhecimento dos alunos. Nos experimentos, nosso método atinge uma precisão de reconhecimento de 97,2%.*

1. Introdução

No cenário de ensino-aprendizagem de programação, a prática e a constante resolução de exercícios são essenciais para a aquisição de conhecimento [Galvão et al. 2016]. Entre as principais tecnologias voltadas para este contexto, destacam-se os juízes on-line. Esses ambientes são bastante utilizados em disciplinas de introdução à programação, e em geral possuem um ambiente de desenvolvimento integrado (IDE) onde os alunos podem codificar suas soluções para os exercícios disponibilizados pelos professores [Galvão et al. 2016]. Dentre as principais características desses sistemas estão o estímulo à autoaprendizagem, as correções automáticas e os *feedbacks* imediatos para os alunos.

Entretanto, apesar dos vários benefícios, uma das principais dificuldades para a adoção desses sistemas é a autenticidade dos usuários. Segundo [Ullah et al. 2019], a maior ameaça ao uso de sistemas que permitem a submissão de qualquer trabalho on-line

DOI: 10.5753/cbie.sbie.2020.1193

é a falsificação de identidade (*Impersonation*). Esse termo está relacionado ao ato de um usuário fingir ser outra pessoa para fins de fraude.

No caso dos juízes on-line, a autenticação do aluno normalmente é feita apenas no início da sessão de *login* mediante a digitação de uma senha, conhecida como autenticação estática *(Static Authentication - SA)* [Mondal and Bours 2017]. Posteriormente, durante a sessão, não é verificada nenhuma outra vez a genuinidade do usuário logado. Isso pode ocasionar problemas de autenticidade ou falsificação de identidade, como citado anteriormente. Por exemplo, alguns alunos podem autenticar-se com perfis de colegas, fingindo serem usuários genuíno, e assim, praticam fraudes. Outros convidam terceiros para fazer seus exercícios e avaliações, visando benefícios próprios.

Muitos métodos de autenticação têm adotado a biometria comportamental para resolver problemas como esses em sistemas on-line. A biometria comportamental está relacionada com padrão de comportamento de uma pessoa, por exemplo: dinâmica do mouse, dinâmica de digitação, entre outras [Giot and Rocha 2019]. Nesta perspectiva, o recurso biométrico comportamental da dinâmica de digitação tem ganhado destaque. Foi demonstrado que cada indivíduo possui uma forma distinta na maneira de digitar [Mondal and Bours 2017]. Além disso, esse recurso não requer uso de hardware adicional para executar a coleta de dados biométricos, apenas um teclado comum. Consequentemente, os atributos coletados podem ser obtidos silenciosamente, de forma transparente, sem atrapalhar o aluno legítimo e sem alertar o aluno trapaceiro que esteja sob avaliação.

Nesta pesquisa, o recurso biométrico utilizado será a dinâmica de codificação, que tem significativa relação com a dinâmica de digitação, mas que acontece dentro de um contexto de desenvolvimento de códigos de computadores. A intuição é que cada pessoa possua um perfil particular de codificação em uma dada linguagem de programação, e que esse perfil pode ser utilizado para autenticação contínua de alunos em ambientes juiz on-line. A autenticação contínua *(Continuous Authentication - CA)* refere-se ao ato de verificar a autenticidade do usuário continuamente com base na atividade que ele executa no sistema [Mondal and Bours 2017].

Diferentes algoritmos podem ser aplicados no campo da autenticação biométrica. Entre eles estão os algoritmos de aprendizagem profunda (do inglês *Deep Learning - DL*) [Chong et al. 2019]. A principal vantagem de utilizar aprendizagem profunda 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 em biometria comportamental utilizando aprendizagem profunda, com o intuito de autenticar, de forma não intrusiva e contínua, alunos em ambientes de juízes on-line.

Este artigo encontra-se organizado em mais seis seções. A Seção 2 discute alguns trabalhos relacionados. A descrição da solução proposta, experimentos realizados, bem como seus resultados, são descritos e analisados nas Seções 3, 4 e 5. As Seções 6 e 7 finalizam o trabalho, resumindo as principais contribuições obtidas e expondo possibilidades de trabalhos futuros.

2. Trabalhos Relacionados

Diversas pesquisas têm abordado o tema de autenticação contínua baseado em biometria comportamental, dentre as quais o recurso biométrico da dinâmica de digitação tem sido

bastante estudado em diversos contextos [Pisani and Lorena 2013], inclusive educacional [Young et al. 2019].

Nessa perspectiva, [Cruz et al. 2017] propuseram um mecanismo de autenticação periódica baseado na dinâmica da digitação de usuários nos ambientes virtuais de aprendizagem (AVAs). Os modelos de reconhecimento gerados apresentaram desempenho acima de 92% de acurácia, fornecendo um indicativo favorável acerca da viabilidade de utilização do mecanismo proposto. Em outro estudo, [Xiaofeng et al. 2019] propuseram um método de autenticação contínua utilizando textos sem limite de digitação. Para isso, desenvolveram um modelo de rede neural profunda. No trabalho, os dados digitados são vetorizados e, em seguida, divididos em segmentos de comprimento fixo. Uma rede neural convolucional (CNN) é responsável pela extração de novas características da digitação de cada indivíduo. Nos experimentos, o modelo foi testado e a melhor taxa de Erro Igual (EER) foi de 3,04%.

Nos trabalhos citados acima, os modelos precisam ser treinados com quantidades significativas de dados de cada usuário. Além disso, a adição de novos usuários requer que a arquitetura seja modificada e atualizada. Para resolver esses problemas, [Giot and Rocha 2019] propuseram um modelo de redes neurais siamesa que visa, sem grande número de amostras, comparar duas entradas para calcular sua semelhança. Para isso, utilizaram a dinâmica de digitação em textos estáticos como recurso biométrico dos usuários e alcançaram resultados promissores.

Não foram encontrados trabalhos que abordem a utilização da dinâmica codificação para a autenticação contínua de alunos em ambientes de juiz on-line. Entretanto, o estudo de [Longi et al. 2015] mostra que os alunos podem ser autenticados ou identificados através destas características em atividades das disciplinas de programação. Nesse estudo, os autores estudaram o quanto a quantidade dos dados afeta a precisão da identificação dos alunos. Nos experimentos, quando o tamanho do conjunto de dados de treinamento foi aumentado de uma para seis semanas, foi constatado um aumento de 78% para mais de 95% na precisão da identificação dos alunos. Nesse trabalho foram utilizados somente autenticação estática.

O diferencial deste trabalho em relação aos apresentados é que, além de propor um método de autenticação contínua para ambientes de juiz on-line utilizando a dinâmica da codificação, ele também projeta um modelo de rede convolucional siamesa que extrai automaticamente, a partir de dados brutos, novas características necessárias para o reconhecimento dos alunos. Além disso, a arquitetura proposta neste trabalho cria um extrator de característica genérico (mesmo com quantidade limitada de dados) que poderá ser usado para classificar novos alunos, sem a necessidade de retreiná-lo.

3. Solução Proposta

A solução proposta consiste em um método de autenticação contínua baseado em biometria comportamental utilizando aprendizagem profunda. Cabe ressaltar que esse método é complementar a autenticação inicial de estudantes feita por meio de *login* e senha. A Figura 3 fornece uma visão geral do método de autenticação, que é dividido em quatro etapas: coleta dos dados, pré-processamento, treinamento e autenticação do aluno/usuário. Essas etapas serão detalhadas a seguir:

Coleta: Para verificar a legitimidade do usuário logado através da dinâmica

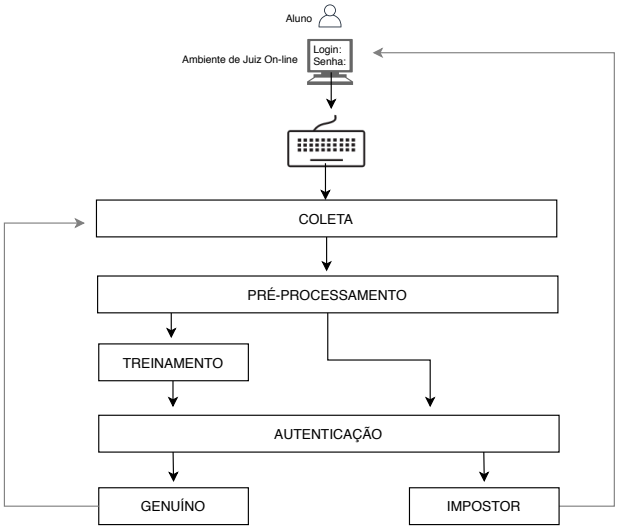

Figura 1. Esquema da arquitetura da solução proposta.

de codificação, é necessário antes coletar dados sobre o processo de codificação desse usuário. Nessa etapa será feita 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-online, e prosseguirá até o término da sessão. **Pré-Processamento:** Uma vez feita a coleta, os dados do usuário serão normalizados e segmentados para envio à rede neural responsável pelo treinamento. **Treinamento:** Será realizado um treinamento de uma rede convolucional siamesa que será responsável pela extração automática de novas características biométricas oriundas dos dados brutos temporais da dinâmica da codificação de cada aluno. Em seguida, será produzido um modelo genérico que classificará os estudantes com base em seu ritmo de codificação. **Autenticação do usuário:** Após o treinamento, o modelo gerado na etapa anterior será avaliado e, consequentemente, será verificado a autenticidade do aluno.

3.1. Rede Neural Proposta

As redes neurais siamesas foram introduzidas pela primeira vez para resolver a verificação de assinaturas [Santos et al. 2018]. O termo siamesa se deve ao fato dessa arquitetura ser composta por sub-redes idênticas que compartilham os mesmos parâmetros. Cada sub-rede recebe uma entrada, e o modelo produzirá uma pontuação de similaridade indicando as chances das duas entradas pertencerem à mesma classe [Sekhar et al. 2019].

Por exemplo, dada duas entradas, a rede siamesa cria um espaço vetorial n-dimensional aplicando a função $G_w(X) \rightarrow \mathbb{R}^n$. Cada entrada X_1 e X_2 é mapeada por G_x e cada saída é submetida a uma função de similaridade $Sim(G_w(X1), G_w(X2)) \rightarrow \mathbb{R}^1$. A função de similaridade é normalmente baseada na medida de distância Euclidiana (D), e é obtida da seguinte forma:

$$D = Sim(G_w(X1), G_w(X2)) = \sqrt{[G_w(X1) - G_w(X2)]^2}, \quad (1)$$

onde $G_w\,(X1)$ e $G_w\,(X2)$ são dois pontos em um espaço vetorial multidimensional criados pelos parâmetros compartilhados w quando mapeiam as entradas $X1$ e $X2$. Uma

vez obtida a medida de similaridade, é possível decidir se os dados de entrada pertencem ou não a uma mesma classe (ou um novo aluno, em nosso contexto) dado um limiar de distância definido. Para cada par de amostras, a distância D entre os vetores de saída das duas redes é calculada pela função de perda *(loss function)*. As redes neurais siamesas fazem uso da *Contrastive Loss* para guiar o treinamento dependendo da semelhança dos rótulos. A função *Contrastive Loss* é definida da seguinte forma:

$$\begin{aligned} L_G &= (1 - Y_A)Y_p^2, \\ L_I &= Y_A(\max(M - Y_p, 0))^2, \\ L &= L_G + L_I. \end{aligned} \tag{2}$$

onde Y_A e Y_P são, respectivamente, o valor real dos pares que alimenta a rede neural siamesa, sendo 1 se forem pares genuínos ou 0 caso contrário. M é o valor de margem da distancia que define quais pares são genuínos ou impostores.

Conforme mostrado na figura 2, nesta pesquisa é proposta uma rede neural convolucional siamesa composta por duas sub-redes CNN 1D idênticas, que compartilham os mesmos parâmetros. A entrada para a rede neural consiste em segmentos de séries temporais relacionadas às amostras de tempo da dinâmica de codificação dos usuários que serão transformadas no espaço aprendido pela rede siamesa. As sub-redes CNNs 1D possuem duas camadas convolucionais responsáveis pela extração automática de novos recursos das características (*features*) geradas a partir dos dados brutos e consideradas relevantes para a identificação do usuário. Neste caso, não há necessidade de descoberta manual de características, uma vez que o aprendizado da representação dos dados ocorre de forma automática. Além disso, as sub-redes possuem diferentes núcleos de convolução que extraem diferentes sequências de características, onde é inserido um ***kernel*** do tamanho 3 para produzir um mapa de características. A técnica de ***Batch Normalization*** também foi adotada para melhorar a velocidade, o desempenho e a estabilidade da rede. Para fazer a CNN aproximar as funções complexas e induzir não linearidade, usamos a *rectified linear activation function* (***ReLu***) como função de ativação. Por fim, a camada ***Flatten*** também foi adotada, resultando em um vetor de dados de entrada que será passado para cada CNN da rede siamesa.

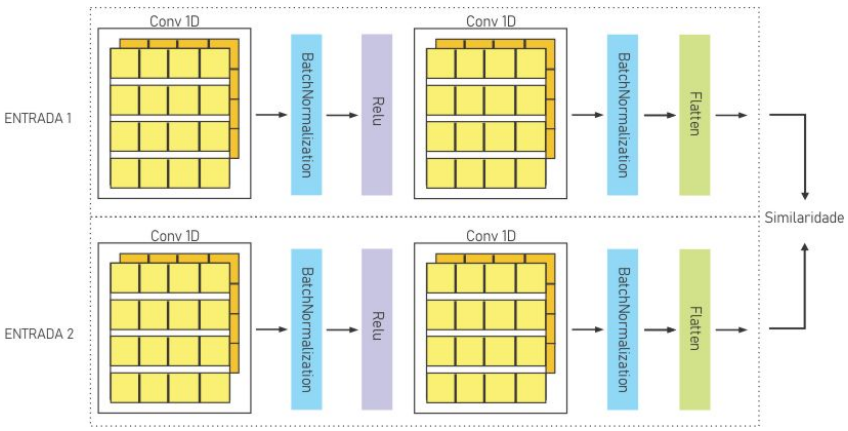

Figura 2. Arquitetura da rede neural siamesa proposta.

4. Autenticador Biométrico

Uma vez treinados, os algoritmos de reconhecimento precisam ser avaliados. As métricas adotadas para avaliação são: **Taxa de falsa-rejeicão (FRR)**, que indica quantos usuários genuínos foram classificados como impostores; e **Taxa de falsa-aceitação (FAR)**, que indica a taxa de usuários impostores que foram considerados como usuários legítimos pelo sistema biométrico. Como essas duas taxas são inversamente proporcionais, adotou-se a Taxa de erro igual (EER), que representa o valor assumido quando FAR e FRR são iguais (ponto de equilíbrio entre as taxas). Se um sistema biométrico tem EER de 0,1%, isso significa que o sua eficácia será 99,9%. Neste trabalho, as taxas de erro são calculadas para cada usuário e, em seguida, calculadas a média sobre todos os N usuários.

5. Experimentos e Resultados

5.1. Base de dados

As bases de dados desta pesquisa foram coletadas nas duas primeiras semanas de aula de 7 turmas da disciplina de Introdução à Programação da Universidade Federal do Amazonas - UFAM, com o total de 260 estudantes. 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[1]. Os módulos estudados nas duas primeiras semanas de aula foram: (a) variáveis e (b) estrutura de programação sequencial. Dentro desses módulos, os alunos tiveram acesso a uma lista de exercícios e foram submetidos a uma avaliação no final da segunda semana.

5.2. Captura de Dados

Para aquisição dos dados da dinâmica de codificação de cada estudante, após o aluno efetuar o *login* e acessar a IDE do juiz on-line, um componente de *software* escrito em *Javascript* iniciou a coleta contínua das ações do teclado desempenhadas dentro da IDE. Os dados de registro foram armazenados no servidor em arquivos *logs*, que posteriormente foram convertidos em arquivos *csv*. A ferramenta de registro também capturou mais algumas informações relacionadas às atividades dos alunos como: código da atividade, entre outras mostradas na Tabela 1.

Tabela 1. Estrutura dos dados brutos da dinâmica da codificação

Seq	TpEvent	Action	Value	Time	IdActivity	IdStudent
n	K	U D	string	ms	int	int

A Tabela 1 mostra o formato dos dados brutos capturados no juiz on-line CodeBench, sendo que: **Seq** representa a sequência da ocorrência dos eventos; **TpEvent** representa o tipo do evento, que no caso é sempre 'K' (eventos de teclado); **Action:** indica se o evento é um pressionamento de tecla (D, Down) ou liberação de tecla (U, UP); **Value** informa qual tecla foi pressionada ou liberada; **Time** registra a data e hora (com milissegundos) do evento, com um intervalo de amostragem de 16 ms; **IdActivity** informa o identificador da atividade executada pelo aluno; e por último, **IdStudent** informa um identificador numérico do aluno.

[1] http://codebench.icomp.ufam.edu.br/

5.3. Pré-processamento de Dados

Neste trabalho, foram considerados somente eventos de teclado simples, ou seja, apenas eventos que envolviam uma única tecla do teclado. Deste modo, foram excluídos eventos como copiar (Ctrl+c) e colar (Ctrl+v), ou qualquer outro evento envolvendo duas ou mais teclas simultâneas. Além disso, foram eliminados eventos com longos intervalos de tempo, como atividades iniciadas em um determinado dia e finalizadas em outro.

As características 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) e o **'Tp'** (tempo de pressão de uma tecla). Tp é obtido pela subtração de Down pelo Up (ou seja, Tp = Up-Down) da mesma tecla [Cruz et al. 2017]. Neste estudo, somente essas três características temporais de pressionamento foram utilizadas como recursos biométricos dos usuários. Novas características (neste caso, correlações entre dados da dinâmica de codificação) serão extraídas de forma automática pela rede neural proposta descrita na seção anterior utilizando a técnica *Feature Learning*, que permite ao sistema identificar de maneira automática as representações necessárias para a reconhecimento do usuário a partir dos dados brutos. Neste caso, a técnica *Feature Learning* pode trazer informações de contexto significativas ou padrões desconhecidos que podem ser úteis para o reconhecimento do aluno em dados brutos temporais da dinâmica de codificação. Em seguida, os dados foram analisados e convertidos no formato *timestamp* e, na sequência, normalizados entre os valores 0 e 1.

5.4. Segmentação dos Dados

Nesta etapa, com os dados sendo capturados de forma contínua, foi necessário particioná-los em tamanhos fixos e menores. Neste trabalho, foram analisados segmentos de tamanhos 10, 30 e 50. Esses tamanhos equivalem a quantidade caracteres digitados pelo aluno. Cabe ressaltar que os caracteres em si não são utilizados neste estudo, apenas as características temporais adquiridas no efetuar do pressionamento das teclas. Em seguida foram utilizadas janelas deslizantes. Neste caso, quando o aluno pressiona uma sequência de caracteres, a janela deslizante avança um passo e o modelo produz um novo resultado. À medida que a janela desliza, o sistema de autenticação continua verificando, alcançando assim, a finalidade da autenticação de identidade contínua.

5.5. Treinamento da Rede Neural Convolucional Siamesa

Para treinar o modelo de rede neural siamesa proposto na Seção 3.1, é necessário criar pares de amostras para as entradas. Desta forma, foram criados pares positivos, que são pares de amostras pertencentes a um aluno genuíno (classe 1), e pares negativos, que são pares de amostras de usuários distintos, desta forma, alunos trapaceiros (classe 0). Em seguida, os dados são divididos em treinamento e teste. Para isso, o conjunto de dados foi particionado em 70% para treinamento e 30% para testes. Para efetuar essa divisão foi utilizada a biblioteca *scikit-learn* da linguagem Python, que possui a função *train-test-split()* capaz de escolher de forma aleatória os recursos da dinâmica de codificação utilizados para o treinamento e para teste, sem repeti-las.

Os melhores resultados (descritos na próximaa seção) no treinamento desta rede foi considerando o algoritmo de otimização *RMSprop*, com taxa de aprendizado de 0,001, 100 épocas de treinamento, *batch size* de 100 e acurácia como métrica de treinamento. O modelo foi construído em *Keras-Tensorflow*.

5.6. Resultados

Um dos principais fatores que influenciam o desempenho de um algoritmo de autenticação contínua é a quantidade de amostras na segmentação de tamanho fixo [Xiaofeng et al. 2019]. Deste modo, o primeiro experimento foi conduzido para avaliar a quantidade de amostras necessárias dentro do problema proposto. Para isso, foram avaliados os seguintes tamanhos de seguimentos fixos: T = {10, 30 e 50}, onde T é o tamanho do segmento, ambos utilizando janelas deslizantes de 50%. A tabela 2 resume a acurácia e taxas de erro alcançadas para diferentes valores de segmentos no experimento. Conforme podemos observar na tabela, os melhores resultados são alcançados para M segmentos fixos de tamanho = 30, com acurácia de 97,2%, e taxa de erro de 0,025%.

Tabela 2. Acurácias e Taxas de Erro (%) alcançadas para diferentes tamanhos de segmentos

Tamanho Segmento	Acurácia	EER
10	92,6	0,071
30	97,2	0,025
50	78,4	0,217

Tabela 3. Acurácias e taxas de erro (%) obtidas a partir das duas arquiteturas

Arquitetura	Acurácia	EER
Giot&Rocha[2019]	90,2	0,098
Proposta	97,6	0,025

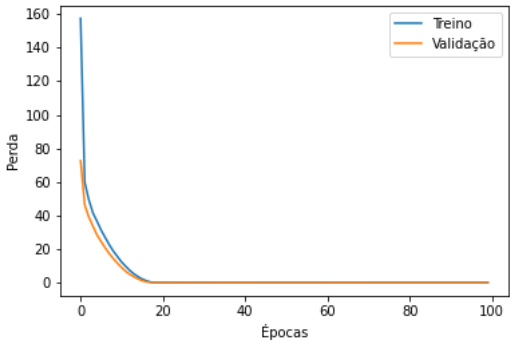

Figura 3. Gráfico das funções de perda para cada conjunto

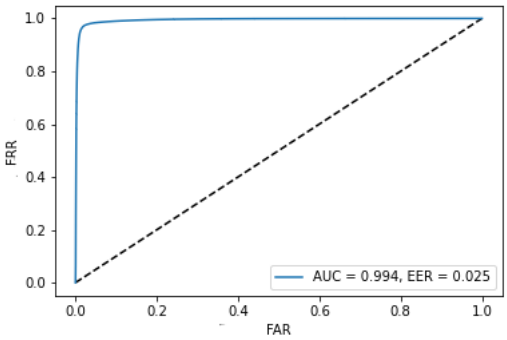

Figura 4. EER do melhor resultado do método proposto

A Figura 3 ilustra a evolução da função de perda em relação as épocas. Quando é treinado um modelo de *Deep Learning*, o gráfico que se espera é basicamente desse tipo, treinamento e validação com curvas bem similares. Por fim, a Figura 4 onde mostra a curva ROC e a métrica EER, nas quais indicam que a identificação dos usuários/alunos baseada no padrão de codificação é bastante precisa, demonstrando a confiabilidade e a eficácia do método de autenticação proposto em dados reais registrados em um ambiente juiz on-line.

5.7. Comparação com trabalhos relacionados atuais

Desta vez, comparamos a arquitetura proposta com trabalhos relacionados recentes. Entretanto, há algumas diferenças entre ambas. Por exemplo, os autores do *baseline* [Giot and Rocha 2019] desenvolveram um modelo de rede neural siamesa que tem como base uma rede MLP (*Multi-layer perceptron*). Diferentemente da arquitetura proposta, na qual desenvolvemos uma rede CNN 1D como base. Além disso, um outro diferencial

deste trabalho em relação ao *baseline* e que utilizamos a dinâmica da codificação como recurso biométrico, logo, existe diferença no contexto empregado.

Para permitir comparações justas, os dois modelos são treinados e testados com os mesmos dados (quantidade de usuários iguais para treinamento e teste) e segmentos fixos de tamanhos iguais. Na Tabela 3, são apresentadas as acurácias e taxas de erro obtidas a partir das duas arquiteturas. Podemos observar que a arquitetura apresentada neste trabalho supera o trabalho *baseline* [Giot and Rocha 2019]. Desta forma, a rede convolucional siamesa extraiu características robustas dos dados brutos da dinâmica de codificação dos alunos, levando a uma precisão superior no reconhecimento dos usuários.

6. Impactos diretos do trabalho para o âmbito educacional

Com a utilização do método proposto, pretende-se diminuir as ocorrências reais e prováveis de fraude por parte de estudantes em exercícios e avaliações que ocorrem nos sistemas juiz on-line, uma vez que será feita a verificação constante do aluno durante toda a sessão. Monitorando e analisando continuamente as entradas de dados de cada usuário com base em seus padrões habituais de codificação, evitando que alunos desonestos obtenham ajuda de terceiros para fazer suas atividades. Além disso, será possível autenticar alunos novatos, ou seja, usuários nunca vistos antes, sem a necessidade de modificação ou atualização do método. Dessa forma, a utilização do método proposto de autenticação contínua de alunos em ambientes de programação baseado em biometria comportamental impacta positivamente o âmbito educacional, uma vez que garante às instituições e professores uma segurança complementar contra fraudes, principalmente em casos em que os alunos têm a opção de fazer os exercícios e avaliações em casa. Além disso, é importante para o aluno, uma vez que testifica sua integridade acadêmica em todas as atividades, recebendo de forma justa as notas. O método também poderá ser utilizado para informar o professor ou monitor da classe, 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.

7. Conclusões e Trabalhos Futuros

Neste artigo, foi proposto e validado um método de autenticação contínua de multiusuários utilizando biometria comportamental em ambientes juiz on-line. Para isso, foi projetado um modelo de rede neural convolucional siamesa que mede a similaridade e determina se pares de entradas de dados são ou não do mesmo aluno. Para validar esse método, foi utilizado dados da dinâmica da codificação de 260 usuários (alunos) de um ambiente de juiz on-line. Em seguida, foi realizado um experimento onde o modelo proposto atingiu uma acurácia de reconhecimento de 97,3%, com segmentos fixos de tamanho 30. De acordo com a revisão bibliográfica elaborada para este paper, o presente trabalho é o primeiro método de autenticação contínua de alunos em ambientes de juiz on-line. Com a utilização deste método, como um módulo adicional de segurança em ambientes de programação, foi comprovado através de experimentos que é possível diminuir as ocorrências reais e prováveis de fraude por parte de estudantes em exercícios e avaliações de programação em ambientes juiz on-line.

Para trabalhos futuros pretendemos avaliar outras arquiteturas de redes neurais, aumentar a quantidade de participantes e por fim avaliar as técnicas de *One-shot Learning* e regras de decisão baseada no nível de confiança.

8. 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

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

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*, 25(02):36.

Galvão, L., Fernandes, D., and Gadelha, B. (2016). Juiz online como ferramenta de apoio a uma metodologia de ensino híbrido em programação. In *Brazilian Symposium on Computers in Education (SBIE)*, volume 27, page 140.

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

Longi, K., Leinonen, J., Nygren, H., Salmi, J., Klami, A., and 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.

Mondal, S. and Bours, P. (2017). A study on continuous authentication using a combination of keystroke and mouse biometrics. *Neurocomputing*, 230:1–22.

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

Santos, V. A. et al. (2018). Siamesevo-depth: odometria visual através de redes neurais convolucionais siamesas.

Sekhar, C., Mukherjee, P., Guru, D. S., and Pulabaigari, V. (2019). Osvnet: Convolutional siamese network for writer independent online signature verification. *arXiv preprint arXiv:1904.00240*.

Ullah, A., Xiao, H., and 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., and Shengwei, Y. (2019). Continuous authentication by free-text keystroke based on cnn plus rnn. *Procedia computer science*, 147:314–318.

Young, J. R., Davies, R. S., Jenkins, J. L., and Pfleger, I. (2019). Keystroke dynamics: establishing keyprints to verify users in online courses. *Computers in the Schools*, 36(1):48–68.