Behavioral Biometrics for Continuous Authentication on Mobile Devices: Anomaly Detection through Keystroke Dynamics with Machine Learning

Kelvin S. Lopes[1], Davi O. Lopes[1], Wesley G. P. Pavanello[1], César L. C. Mattos[1], Jarelio G. da S. Filho[2], José D. C. Neto[2], Rafael L. Gomes[3], Nicksson C. A. de Freitas[2], Emanuel B. Rodrigues[1]

[1]Universidade Federal do Ceará (UFC)
Av. da Universidade, 2853 – CEP 60020-181 – Fortaleza – CE – Brasil

{kelvin.sampaio, davioliveira, wesley.pavanello}@alu.ufc.br,
{cesarlincoln, emanuel}@dc.ufc.br

[2]SiDi
Av. República do Líbano, 251 – CEP 51110-160 – Recife – PE – Brasil

{j.filho, j.carneiro, nicksson.a}@sidi.org.br

[3]Universidade Estadual do Ceará (UECE)
Av. Dr Silas Munguba, 1700 – CEP 60714-903 – Fortaleza – CE – Brasil

rafa.lopes@uece.br

Abstract. *The popularity of mobile devices generates the need for security solutions to ensure user identity. One approach to deal with this scenario is behavioral biometric for continuous authentication (BBCA), which has become increasingly known with advances in hardware and data science technologies. However, this approach still lacks greater robustness in how to model the user's behavioral biometrics as well as maximize the effectiveness of this authentication. It is particularly important to note in this context that this paper presents an approach to BBCA using machine learning (ML) techniques integrated with sliding windows. Accordingly, biometric data collected from keystroke dynamics typing activities on mobile devices were used to identify genuine users, random (unskilled) imposters and skilled imposters. The ML models for impostor identification followed an anomaly detection (AD) approach, where only genuine data is available at training time. In addition, the use of sliding windows allowed the inclusion of the temporal dimension of the task. The results obtained indicate that the proposed solution has a practical feasibility in terms of its suitability to perform user identification, specifically, the KNN model stood out by achieving a superior performance in the value window 4. In this configuration, it achieved a score of 94% for the F1-score metric in the random imposter scenario, and 93% in the skilled imposter scenario.*

1. Introdução

Com o aumento significativo no número de dispositivos móveis nos últimos anos, é natural que também haja um crescimento correspondente nos ataques especializados nestes

dispositivos. Segundo o relatório da [GSMA Intelligence 2022], órgão representativo das operadoras de redes móveis globais, já existem no mundo mais de 5,3 bilhões de pessoas que utilizam dispositivos móveis, e as projeções indicam que esse número poderá alcançar aproximadamente 5,8 bilhões até 2025, o que expressa a contínua expansão desse mercado.

De acordo com [Zimperium 2023], empresa especializada em segurança móvel, em seu relatório de 2021, houve mais de 2 milhões de novas assinaturas de *malwares* detectadas. Essa estatística desperta uma grande preocupação, uma vez que os métodos convencionais podem não ser suficientes para combater efetivamente esses ataques, devido à falha no uso predominante de métodos de autenticação baseados na memória do usuário (padrões e senhas) ou em sua biometria.

Diante disso, estudos estão sendo desenvolvidos com o objetivo de acrescentar novas camadas de segurança aos dispositivos. Uma das áreas de estudo propícia para a inovação é a autenticação contínua baseada no comportamento do usuário, conhecida como Autenticação Contínua Biométrica Comportamental (*Behavioral Biometrics for Continuous Authentication* - BBCA) [Kokal et al. 2022].

A BBCA se baseia na premissa de que o comportamento humano possui padrões distintos e únicos, que permitem a criação de perfis de usuários e a detecção de alterações significativas nos padrões comportamentais, o que pode indicar uma possível invasão.

Especialmente em um contexto móvel, a BBCA tem ganhado destaque crescente, pois possibilita a criação de perfis comportamentais dos usuários a partir do monitoramento de suas interações na tela do dispositivo, como digitação, deslizes e toques. Essa coleta de dados não requer *hardware* adicional e pode ser facilmente integrada aos aplicativos já existentes, tornando a BBCA uma solução sólida e eficiente para autenticação em dispositivos móveis.

No entanto, a aplicação da BBCA, [Almohamade et al. 2021] também tem sido explorada em áreas como ciência de dados, incluindo o aprendizado de máquina (*Machine Learning* - ML) e aprendizado profundo (*Deep Learning* - DL), englobando duas técnicas principais: classificação, em que modelos preditivos são desenvolvidos para distinguir diferentes classes e detecção de anomalias (*Anomaly Detection* - AD), que permite diferenciar entre padrões normais e anormais.

Dessa forma, para reconhecer os comportamentos anômalos e potencialmente maliciosos no presente trabalho, obtivemos dados biométricos comportamentais do conjunto de dados BehavePassDB[1] criado pelo autor [Stragapede et al. 2023]. Esses dados consistem em informações coletadas a partir de sensores presentes em dispositivos móveis, nos quais os usuários realizaram atividades típicas de interação humano-computador (IHC), como digitação de texto livre, deslizamento de tela e dinâmica de toque.

Cabe ressaltar que foram criados dois cenários para as atividades IHC: o cenário do impostor aleatório e o cenário do impostor habilidoso. No primeiro, os participantes utilizaram dispositivos diferentes e não receberam nenhuma orientação sobre as tarefas. Já no segundo, simularam o comportamento do usuário genuíno ao utilizar o mesmo dispositivo. Neste experimento, focamos na tarefa de digitação, também conhecida como dinâmica de teclas (*keystroke*), que se mostrou mais eficiente para a BBCA.

[1] `https://github.com/BiDAlab/MobileB2C_BehavePassDB`

Em virtude desses aspectos apresentados, tivemos o objetivo de propor uma contribuição significativa para aprimorar a técnica BBCA por meio de AD, utilizando ML e janelas deslizantes (JD). A abordagem de AD permite treinar o modelo apenas com dados de comportamento genuíno do usuário, isso reduz a complexidade do problema e resulta na construção de um algoritmo robusto, capaz de funcionar em diversos cenários sem comprometer sua eficiência.

Adicionalmente, a aplicação da técnica de janelas deslizantes permitiu analisar diferentes intervalos temporais da atividade de IHC, o que contribuiu para a melhora nas métricas nos modelos, resultando em melhorias significativas nos resultados.

A estrutura restante deste trabalho é a seguinte: na seção 2, apresenta-se uma análise e discussão dos trabalhos relacionados, destacando suas semelhanças e diferenças em relação ao trabalho proposto. Na seção 3, é detalhada a abordagem proposta para solucionar o problema em questão. Na seção 4, são demonstrados os resultados derivados das experimentações conduzidas. Por fim, na seção 5, são apresentados as conclusões obtidas da pesquisa.

2. Trabalhos Relacionados

A seguir, serão apresentados alguns trabalhos relacionados que abordam a BBCA e suas diferentes abordagens. Analisaremos as contribuições desses estudos, destacando as vantagens e desvantagens em relação ao nosso trabalho, bem como os fatores de inovação.

[Darabseh and Pal 2020] propõem uma análise de desempenho da atividade de *keystroke* do teclado do computador com foco em AD. O estudo abrangeu a coleta de informações sobre as teclas pressionadas, tempo de pressionamento e liberação das teclas. Os autores desenvolveram um programa que emprega técnicas de ML para abordar essa análise. Embora o estudo não tenha se voltado especificamente para dispositivos móveis, ele explora em maior detalhe as características identificadas que podem ser úteis ao considerar a avaliação em sensores móveis.

[Kokal et al. 2022] exploram a BBCA por meio de sensores dinâmicos de toque e movimento. Seu estudo se concentrou na determinação dos melhores modelos de autenticação contínua usando técnicas de classificação em ML. Embora tenham considerado diversas características comportamentais entre os usuários, o cenário abordado não incluiu a AD ou a consideração de possíveis invasões, que diferem do foco da nossa proposta.

[Mekruksavanich and Jitpattanakul 2021] propõem *DeepAuthen*, uma nova abordagem para autenticação contínua que emprega técnicas de DL na detecção de padrões de atividade física de usuários de *smartphones* usando sensores móveis, e um modelo de rede neural que distingue com sucesso os comportamentos do usuário. A diferença em relação ao nosso trabalho é que eles se concentraram no reconhecimento de atividades humanas e adotaram uma abordagem de classificação, sem considerar o caso do impostor.

[Shah 2020] destaca a preocupação em projetar sistemas biométricos confiáveis baseados no comportamento, utilizando técnicas de ML com foco em AD em atividades de *keystroke*. Ele explora as características do tempo de pressionamento das teclas, bem como a utilização de janelas deslizantes. No entanto, sua abordagem se limita a analisar a diversidade de dispositivos, deixando de fora a detecção de impostores habilidosos.

[Tahoun 2021] demonstra em dispositivos móveis a utilização de modelos generativos para autenticação contínua implícita. Para realizar a identificação do usuário através de seu perfil comportamental, tomaram uma abordagem de AD bem como técnicas de DL. Na nossa proposta, por outro lado, estamos considerando utilizar apenas técnicas de ML, bem como a adoção de janelas deslizantes.

[Thapliyal et al. 2022] apresentam uma abordagem multimodal de BBCA em *smartphones*, levando em consideração os padrões de atividade do usuário, como toque na tela, digitação e movimentação do dispositivo. Essa abordagem utiliza técnicas de ML com foco em AD e aborda um dos cenários contemplados neste estudo, que envolve o uso do mesmo dispositivo. No entanto, não considera a situação de um impostor habilidoso ou um cenário com o uso de dispositivos diferentes.

[Wagata and Teoh 2022] apresentam um método de BBCA para dispositivos móveis, utilizando um pequeno número de amostras. A abordagem emprega técnicas de ML, incluindo métodos de classificação e técnicas de DL em um modelo de rede neural, para analisar os padrões de comportamento do usuário com base em AD. O estudo utiliza dados anômalos de usuários impostores em dispositivos diferentes, com ênfase em poucas amostras preliminares para aplicar DL e atividades de toque na tela, sem considerar o uso de janelas deslizantes.

Por fim, o conjunto de dados BehavePassDB, [Stragapede et al. 2023], adotou uma abordagem de classificação para identificar os usuários com base em quatro atividades IHC. Essas atividades incluíam cenários de impostor aleatório e impostor habilidoso. Apesar de termos objetivos semelhantes em relação à consideração dos impostores, nossa abordagem difere no uso de técnicas de ML e à ênfase em AD, bem como a criação de um modelo especializado em atividades de digitação e uso de janelas deslizantes.

Na Tabela 1, é apresentada a sumarização dos trabalhos mencionados nesta seção, juntamente com suas respectivas diferenciações.

Tabela 1. Sumarização dos trabalhos que utilizaram dados biométricos comportamentais

Acrônimos: DT; Dinâmica de Teclas; F, Fusão; PC, Perfil Comportamental; T, Tecla; TG, Toque Gestual; A, Acelerômetro; B, *Bluetooth*; Ba, Nível da bateria; GPS; Gr, Sensor de gravidade; Gi, Giroscópio; HAR, Reconhecimento de atividade humana; ºC, Temperatura; U, Umidade; L, Luz; AL, Acelerômetro linear; M, Magnetômetro; P, Pressão; Pr, Proximidade; D, Deslize; W, Wi-Fi.

Autores	Conjunto de dados	Modalidade	AD	Técnica Utilizada	Problema	JD	Sensores	Caso Impostor
Darabseh e Pal (2020)	Não publicado	DT	✓	ML	Multiclasse	×	T	Mesmo Dispositivo
Kokal; Pryor e Dave (2021)	HMOG	PC	×	ML	Classe	×	A, Gi, M	Não Considerado
Mekruksavanich e Jitpattanakul (2021)	WISDM-HARB	PC	×	ML	Classe	×	HAR	Não Considerado
Shah (2021)	Não publicado	DT	✓	ML	Classe	✓	T, A, Gi, M	Dispositivo Diferente
Tahoun (2021)	HMOG	PC	✓	DL	Classe	×	A, Gi, M	Mesmo Dispositivo
Thapliyal, Verma e Kumar (2022)	Não publicado	DT, TG	✓	ML, DL	Multiclasse	×	A, Gi	Mesmo Dispositivo
Wagata e Teoh (2022)	HMOG BBHAS	TG	✓	ML, DL	Classe	×	A, Gi, M, P, D	Não considerado Dispositivo diferente
Stragapede et al. (2022)	BehavePassDB	F	×	ML, DL	Classe	×	A, Gr, Gi, L, AL, M, Pr, L, GPS, W, B, ºC, Ba, U	Diferente e mesmo dispositivo (qualificado)
Trabalho atual (2023)	**BehavePassDB**	**DT**	✓	**ML**	**Classe**	✓	**A, Gr, Gi, L, AL, M, Pr, L, GPS, W, B, ºC, Ba, U**	**Diferente e mesmo dispositivo (qualificado)**

3. Abordagem Proposta

Nesta seção, abordaremos detalhes sobre o conjunto de dados empregado, as etapas de pré-processamento, o procedimento de treinamento, validação e teste, bem como os modelos

adotados e as métricas selecionadas para avaliação.

Conforme mencionamos nas seções antecedentes, empregamos o conjunto de dados BehavePassDB para a presente pesquisa. Este conjunto encontra-se particionado em dois repositórios distintos. O primeiro diretório contém os dados de treinamento e validação, com 51 usuários para treinamento e 10 para validação. O segundo repositório possui dados de avaliação de 20 usuários. Os dados foram organizados em arquivos no formato *JavaScript Object Notation (JSON)*.

Neste estudo em particular, optou-se usar o conjunto de dados de validação devido à presença das comparações entre usuários, seus rótulos correspondentes e os resultados das predições obtidas pelo autor. Essa decisão foi tomada para assegurar a consistência e confiabilidade na análise e avaliação do desempenho do modelo.

Inicialmente, os arquivos de atividade continham dados de registro de usuários pseudoanonimizados, incluindo informações de sensores de fundo (acelerômetro, acelerômetro linear, giroscópio, magnetrômetro e sensor de gravidade) e sensores de frente (tecla de texto livre, deslizamento de tela e dinâmica de toque). No entanto, a análise dos registros revelou que não era possível explorar as características detalhadas da dinâmica de teclas, como duração, latência e pressionamento.

Diante desse empacotamento, tornou-se necessário separar os dados de cada registro de usuário para viabilizar os testes. Em seguida, os dados foram organizados de acordo com as especificações do autor do conjunto de dados, no formato *Comma-separated values (CSV)*, garantindo a organização consistente dos registros e colunas no experimento.

Com os dados devidamente estruturados, avançamos para a implementação dos modelos de ML. Esta etapa revela-se fundamental para a extração de informações e padrões significativos dos dados acumulados. A Figura 1 fornece uma representação visual do fluxo de trabalho e das etapas específicas a serem seguidas na aplicação de ML, garantindo uma compreensão clara das etapas subsequentes do experimento.

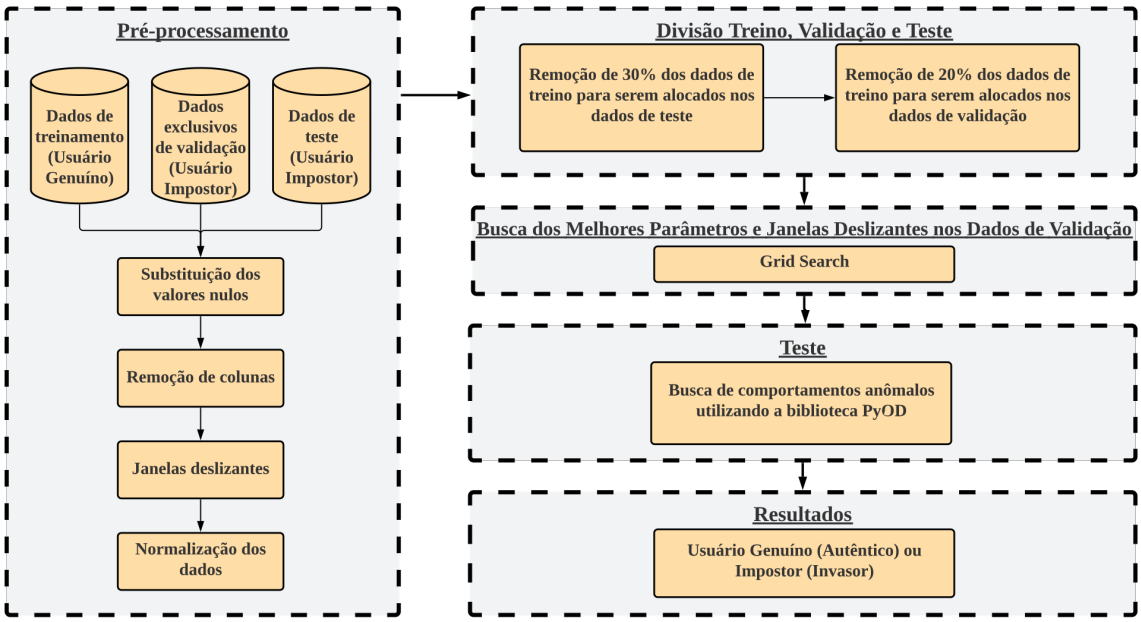

Figura 1. Fluxograma de etapas sequenciais em ML

Conforme a Figura 1, a primeira etapa consiste no pré-processamento. Nesta etapa, buscamos substituir todos os valores nulos por 0, preservando assim a continuidade dos dados.

Com os dados nulos substituídos, procedemos à exclusão das colunas *"timestamp"* e *"ascii code touch"*. Conforme apontado por [Stragapede et al. 2023], a coluna *"timestamp"* não contém valores reais, sendo sua única utilidade a sequencialidade dos dados. Em relação à coluna de *"ascii code touch"*, optou-se por removê-la devido à alta presença de valores nulos, tornando-a irrelevante para a análise.

Para capturar as características por meio das amostras temporais das atividades do usuário, optamos por adotar o conceito de janelas deslizantes, combinado com a técnica de validação cruzada. Essa estratégia não apenas permite a captura de atributos distintivos do usuário, que poderiam não ser detectados de outra forma, mas também simplifica a avaliação sistemática e a seleção otimizada dos valores mais relevantes para cada modelo [Yu et al. 2014].

Após a aplicação das janelas deslizantes, procedemos à normalização dos dados utilizando a biblioteca *scikit-learn*, por meio do método *StandardScaler*. Este método tem como finalidade padronizar as características (*features*) de um conjunto de dados, de modo que estas apresentem média zero e variância unitária [Scikit-learn developers 2023].

Para a fase de treinamento, optamos por instruir o modelo utilizando exclusivamente informações provenientes de usuários genuínos, como se trata de apenas amostras genuínas para formação, a abordagem de uma classe é uma das mais utilizadas para os algoritmos de AD [Darabseh and Pal 2020]. Essa abordagem segue um padrão comumente empregado em sistemas de AD. O propósito subjacente é dotar o modelo com a capacidade de discernir características intrínsecas a cada usuário, de modo a habilitá-lo a realizar identificações precisas em situações práticas.

Na fase de validação, destinamos cerca de 20% dos registros autênticos, previamente extraídos do conjunto de dados de treinamento. Além disso, selecionamos um usuário impostor de modo reservado, designado exclusivamente para a fase de validação. A finalidade dessa abordagem se pauta na intenção de prevenir qualquer possibilidade de o modelo incorporar atributos específicos do usuário de referência, o que poderia resultar em um aprimoramento artificial do desempenho decorrente do conhecimento prévio.

Por último, no que tange aos registros de teste, reservamos cerca de 30% dos dados genuínos, segregados do conjunto de treinamento. Com as coleções de dados assim constituídas, procedemos à busca otimizada pelos parâmetros mais pertinentes.

Com o propósito de determinar os melhores hiperparâmetros e o tamanho ideal da janela deslizante para cada usuário e modelo, escolheu-se usar a técnica de *grid search* no conjunto de validação. Isso envolve testar várias combinações de hiperparâmetros nos modelos e avaliá-las. Todas as combinações e seus resultados são registrados para possibilitar uma comparação sistemática. Isso permite encontrar os melhores hiperparâmetros e tamanhos de janela para cada modelo. Esses parâmetros otimizados serão então utilizados no conjunto de teste [Scikit-learn developers 2023].

A Tabela 2 a seguir demonstra os hiperparâmetros empregados no experimento.

Tabela 2. Hiperparâmetros utilizados no *grid search*

Modelo	Hiperparâmetros	Valores
KNN	n_neighbors	5, 10, 15, 20, 25, 30
	algorithm	auto, ball_tree, kd_tree, brute
OCSVM	nu	0.1, 0.2, 0.3, 0.4, 0.5, 0.6
	kernel	linear, rbf, sigmoid
	gamma	scale, auto
LOF	n_neighbors	5, 10, 15, 20, 25, 30
	algorithm	auto, ball_tree, kd_tree, brute
	leaf_size	30, 40, 50, 60, 70, 80
INNE	n_estimators	80, 100, 125, 135, 150, 200

Para os hiperparâmetros, decidimos adotar uma abordagem experimental alternativa ao optar por uma seleção aleatória de hiperparâmetros no decorrer do processo de *grid search*. Tal escolha foi deliberada com o propósito de explorar de forma mais abrangente o espaço de hiperparâmetros e identificar configurações potenciais que, em circunstâncias distintas, poderiam não ser detectadas.

Na fase final do experimento, prosseguimos com a etapa de teste, na qual empregamos a biblioteca *Python* conhecida como PyOD [Zhao et al. 2019], a qual se concentra nas técnicas de AD. Esta biblioteca se destaca pela sua notável versatilidade, uma vez que engloba mais de 40 algoritmos distintos de detecção de anomalias, conferindo-lhe uma ampla variedade de opções em termos de aplicabilidade. Adicionalmente, sua implementação se revela descomplicada e sua utilização é favorecida pela disponibilidade de uma documentação abrangente.

Na seleção das métricas de avaliação, optamos por adotar as que são amplamente reconhecidas pela comunidade de ML e Ciência de Dados: acurácia, precisão, *recall*, *F1-score* e área sob a curva ROC. Essas métricas são essenciais para avaliar o desempenho de modelos de classificação em tarefas como detecção de AD, classificação binária e outras aplicações similares. Além disso, levamos em consideração o desvio padrão, uma métrica estatística fundamental que complementa e enriquece a interpretação dos resultados da avaliação.

4. Resultados

Nesta seção, são apresentados os resultados da avaliação da solução proposta para a BBCA, baseada na dinâmica de teclas com o uso de AD em dispositivos móveis [2]. Os resultados consideram a média das métricas de cada modelo, obtidas diretamente da biblioteca PyOD.

A amostra temporal das janelas deslizantes foi variada de 1 a 7, contemplando desde os dados originais até a mais extensa amostragem temporal viável. Essa metodologia permitiu avaliar diversos cenários e coletar dados abrangentes sobre o desempenho dos modelos em intervalos temporais diferentes. É importante ressaltar que, devido à variabilidade na latência do processamento entre as janelas deslizantes, a apresentação dos valores não pôde ser realizada de forma precisa e exata, tornando inviável a exibição dos valores numéricos.

[2] `https://github.com/kelvin-ksl/BehavePassDB_Keystroke_AD`

Na Tabela 3, são apresentadas as médias dos resultados obtidos por cada modelo, considerando os dois primeiros valores decimais, englobando todas as janelas. Essas médias refletem o desempenho dos modelos ao identificar corretamente a presença de impostores aleatórios.

Tabela 3. Média dos resultados das métricas de cada modelo no cenário impostor aleatório, considerando todas as janelas seguido do desvio padrão médio

Impostor Aleatório					
Modelo	**AUC**	**Acurácia**	**F1-Score**	**Precisão**	**Recall**
KNN	0,99 ± 0,00	0,92 ± 0,05	0,94 ± 0,04	0,93 ± 0,03	0,96 ± 0,03
OCSVM	0,97 ± 0,04	0,90 ± 0,09	0,92 ± 0,07	0,91 ± 0,09	0,95 ± 0,05
LOF	0,97 ± 0,03	0,91 ± 0,08	0,93 ± 0,07	0,90 ± 0,09	0,97 ± 0,04
INNE	0,97 ± 0,06	0,91 ± 0,08	0,93 ± 0,06	0,90 ± 0,09	0,97 ± 0,04

Conforme se pode constatar na Tabela 3, no que diz respeito à métrica AUC, que quantifica a capacidade de distinguir entre classes, o modelo KNN demonstrou um desempenho superior com um valor de 0,99% com um desvio padrão 0,00, indicando que os valores individuais dos dados estão todos iguais à média, ou seja não há variação nos dados. Isso indica que o KNN é altamente capaz de classificar corretamente os exemplos entre classes distintas.

Em termos de acurácia, que mede a proporção de predições corretas em relação ao total de predições, o KNN novamente obteve a maior pontuação, com um valor de 0,92% e desvio padrão médio de 0,05, indicando um nível moderado de variação em relação a média. Isso sugere que o KNN é eficaz na classificação geral das amostras.

O *F1-Score*, que considera tanto a precisão quanto o *recall*, revelou resultados semelhantes para os modelos OCSVM, LOF e INNE, todos com valores de 0,92% a 0,93%, e desvio padrão médio entre 0,06 e 0,07, indicando também um nível moderado. Isso sugere que esses modelos possuem um equilíbrio razoável entre precisão e capacidade de identificar verdadeiros positivos.

No que diz respeito à precisão, que mede a proporção de verdadeiros positivos em relação a todos os exemplos positivos previstos, o KNN e o OCSVM demonstraram desempenho equiparável, com valores de 0,93% e 0,91%, e desvio padrão médio 0,03 e 0,09, respectivamente, indicando que um apresentou mais consistência e estabilidade em torno da média, enquanto o outro houve mais variabilidade.

Quanto ao *recall*, que mede a proporção de verdadeiros positivos em relação a todos os exemplos positivos reais, os modelos LOF e INNE apresentaram os valores mais altos, ambos com 0,97% e desvio padrão médio de 0,04, indicando serem estáveis e consistentes. Isso sugere que esses modelos têm uma capacidade excepcional de identificar corretamente exemplos positivos.

Em resumo, o modelo KNN se destacou na maioria das métricas, exibindo um desempenho geral superior. Os modelos OCSVM, LOF e INNE também demonstraram um desempenho satisfatório, sendo especialmente eficazes na identificação de exemplos positivos, como indicado pelos altos valores de *recall*. Portanto, a escolha do modelo a

ser utilizado dependerá das prioridades específicas do problema em questão, considerando *trade-offs* entre diferentes métricas de desempenho.

A seguir, temos a Figura 2, nela é apresentada a evolução dos modelos conforme a janela, no cenário do impostor aleatório.

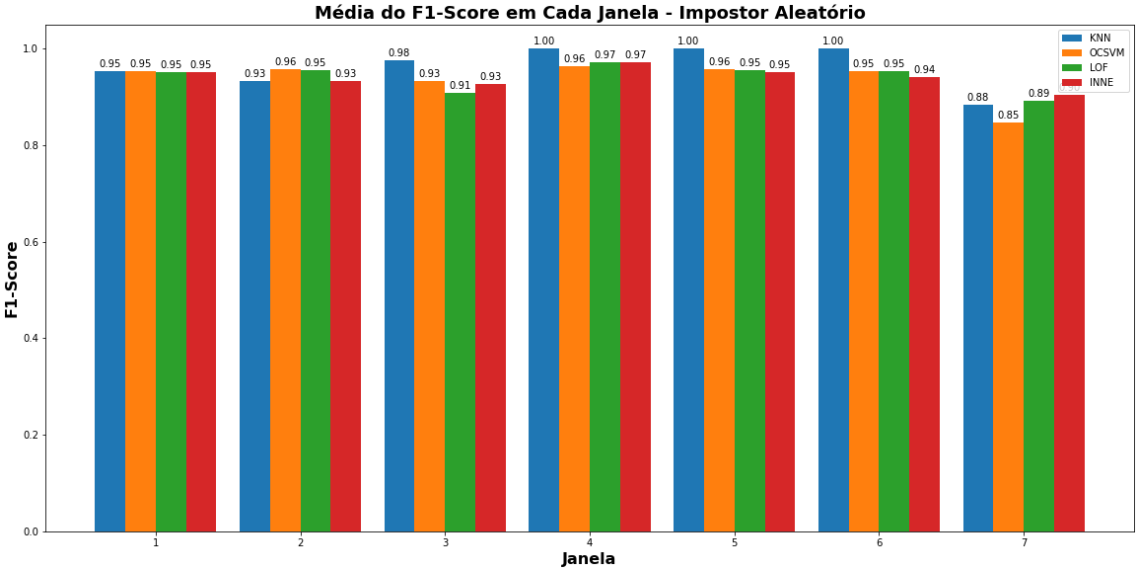

Figura 2. Médias do *F1-score* em cada janela para o cenário impostor aleatório

Com base na Figura 2, podemos analisar a evolução de cada modelo e sua respectiva janela para o cenário do impostor aleatório. Como métrica a ser considerada, utilizamos a média do *F1-score* de todos os usuários de cada janela, os melhores resultados são apresentados na janela 4. Essa análise sugere que, para a detecção eficaz de impostores aleatórios, é recomendado o uso dessa janela específica, na qual os modelos apresentaram métricas maiores.

Vale enfatizar que as demais métricas resultaram em valores semelhantes, tornando suas apresentações desnecessárias para este estudo. Diante dessa situação, a escolha foi feita para exibir a métrica *F1-score*, uma vez que ela resulta da combinação das métricas de precisão e *recall*.

Na Tabela 4, são apresentadas as médias das métricas dos modelos durante os experimentos do cenário do impostor habilidoso.

Tabela 4. Média dos resultados das métricas de cada modelo no cenário impostor habilidoso, considerando todas as janelas seguido do desvio padrão médio

Impostor Habilidoso					
Modelo	**AUC**	**Acurácia**	**F1-Score**	**Precisão**	**Recall**
KNN	0,99 ± 0,00	0,90 ± 0,03	0,93 ± 0,02	0,94 ± 0,03	0,94 ± 0,01
OCSVM	0,96 ± 0,08	0,89 ± 0,12	0,92 ± 0,11	0,90 ± 0,12	0,94 ± 0,12
LOF	0,96 ± 0,08	0,89 ± 0,11	0,92 ± 0,11	0,91 ± 0,11	0,94 ± 0,12
INNE	0,95 ± 0,09	0,89 ± 0,11	0,92 ± 0,11	0,90 ± 0,12	0,95 ± 0,12

Como demonstrado, a partir da média das métricas obtidas no cenário do impostor habilidoso, o modelo KNN se destaca em diversas delas, demonstrando resultados mais favoráveis em comparação com os modelos OCSVM, LOF e INNE.

Ao analisar os resultados da Tabela 4, no que diz respeito as métricas AUC, acurácia, *F1-Score*, precisão, *recall*, conforme mencionado no cenário anterior, o modelo KNN demonstra ter desempenho e desvio padrão médio semelhantes. Enquanto os demais modelos OCSVM, LOF e INNE, apresentam valores entre 0,89% a 0,95%, acompanhados de desvios padrões médios maiores entre 0,11 e 0,12.

Com base na Figura 3, é exibida a progressão das janelas para cada modelo no contexto do cenário do impostor habilidoso.

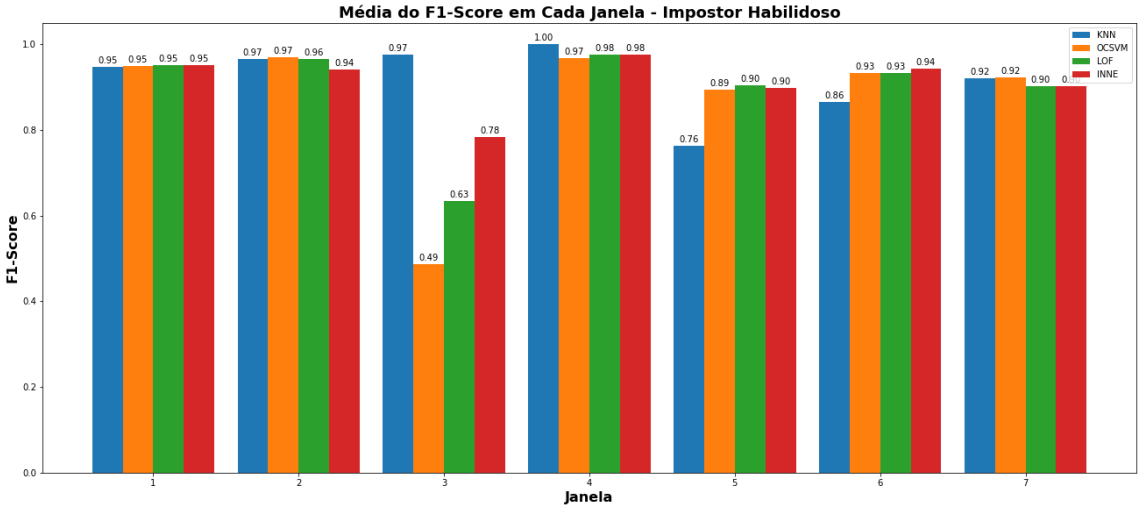

Figura 3. Médias do *F1-score* em cada janela para o cenário impostor habilidoso

Ao observarmos a Figura 3 do cenário do impostor habilidoso, é possível notar uma melhora nas métricas dos modelos já na janela de valor 2, em comparação com a janela de valor 1. No entanto, na janela 3, há uma regressão significativa nos modelos OCSVM, LOF e INNE, enquanto o modelo KNN é o único que se mantém. Na janela 4, todos os modelos atingem seu ponto mais alto.

A notável eficácia do algoritmo KNN em relação aos demais modelos nos testes pode ser atribuída a uma combinação de fatores. A presença de um conjunto de dados de treinamento relativamente pequeno, aliado à simplicidade do ajuste de poucos hiperparâmetros, simplifica o processo de sintonia e calibração do modelo. Isso se mostra particularmente vantajoso quando contrastado com modelos que demandam uma gama mais extensa de hiperparâmetros a serem ajustados, o que poderia aumentar a complexidade e o esforço necessários para atingir um desempenho satisfatório [Tahoun 2021].

Dessa forma, ao comparar os dois cenários, torna-se claro que os resultados obtidos demonstram uma notável similaridade. Em ambas as circunstâncias avaliadas, a janela com valor 4 se sobressaiu como a mais promissora em uma perspectiva global. No entanto, é necessário enfatizar que nem todos os usuários alcançaram as métricas mais otimizadas ao empregar a janela 4. Os resultados apresentados neste estudo constituem uma média global, refletindo, portanto, os valores mais apropriados para uma aplicabilidade generalizada.

5. Conclusão

A técnica BBCA com o uso de janelas deslizantes demonstra um potencial promissor como solução complementar no combate a ameaças cibernéticas em dispositivos móveis. Essa abordagem revelou-se especialmente relevante na detecção de impostores que tentam imitar o comportamento legítimo do usuário ao longo do tempo, tornando-se uma poderosa ferramenta de defesa contra ataques sofisticados.

Dentre os modelos e janelas testadas no conjunto de dados BehavePassDB, foi observado que o modelo KNN foi o que se sobressaiu, apresentando os melhores resultados nas janelas de valor 4 para ambos os cenários.

Dentre as vantagens ao utilizar a BBCA em ML com a abordagem de AD, é a possibilidade de manipular apenas os dados genuínos do usuário durante a etapa de treinamento do modelo. Isso elimina a necessidade de coletar dados de impostores, o que pode ser uma tarefa difícil ou representar um risco adicional à segurança. Essa vantagem torna a implementação da BBCA mais simples e confiável, reduzindo tanto a complexidade quanto os custos associados à obtenção dos dados de treinamento.

Uma outra vantagem significativa da abordagem de AD é a sua robustez, uma vez que, ao contrário de uma abordagem de classificação, não é necessário rotular os dados. Essa característica é especialmente benéfica, pois viabiliza a inclusão de novos dados sem comprometer a eficácia da proposta ou aumentar sua complexidade.

Apesar das possíveis vantagens, há limitações a serem consideradas ao aplicar essa abordagem. Uma delas, é a necessidade de treinar cada modelo separado para cada usuário. Isso pode exigir recursos computacionais significativos e pode ser impraticável em cenários com um grande número de usuários.

O uso da técnica de janelas deslizantes pode afetar a performance dos modelos, conforme demonstrado na seção de resultados. Aumentar o número de janelas resulta em variações nos resultados, pois o uso de múltiplas janelas possibilita capturar padrões temporais diferentes que podem ou não facilitar a detecção de comportamentos anômalos.

No entanto, embora a técnica de janelas deslizantes tenha se mostrado eficaz na detecção de impostores, é importante considerar o tamanho do conjunto de dados e o valor da janela, uma vez que podem afetar a latência na autenticação. É imprescindível encontrar um equilíbrio entre a precisão da autenticação e a velocidade de resposta, adaptando a janela deslizante ao contexto de uso e às necessidades de segurança específicas.

Cabe destacar que devido os registros de dados apresentarem usuários pseudoanonimizados, conforme mencionado nos resultados, não foi possível explorarmos melhor as características da atividade de dinâmica de teclas, tais como mensurar a duração entre teclas pressionadas e liberadas, latência entre o tempo de teclas sucessivas, e pressionamento de teclas.

Em resumo, a BBCA mostrou-se uma solução favorável para garantir a segurança em dispositivos móveis. A combinação de técnicas como a janela deslizante e a utilização exclusiva de dados genuínos no treinamento oferecem vantagens significativas. Embora existam limitações a serem consideradas, essas podem ser superadas com o avanço contínuo da pesquisa e o desenvolvimento de abordagens mais sofisticadas. A BBCA tem o potencial de se tornar uma parte fundamental do arsenal de segurança para dispositivos móveis,

protegendo a identidade dos usuários e mitigando ameaças cibernéticas cada vez mais complexas.

Como trabalhos futuros, é sugerido considerar a criação de um modelo especializado que abranja as demais atividades presentes no conjunto de dados BehavePassDB. Além disso, seria interessante desenvolver um modelo genérico capaz de detectar anomalias em todas as atividades, não se restringindo a uma única categoria. Essa abordagem permitiria uma aplicação mais ampla e abrangente do sistema, oferecendo suporte em diferentes contextos e atividades. Ao explorar essa direção, poderíamos potencialmente aprimorar a capacidade de identificar comportamentos suspeitos e garantir a segurança em dispositivos móveis de forma mais abrangente.

6. Agradecimentos

Parte dos resultados apresentados neste trabalho foram obtidos através do projeto "RESIDÊNCIA EM SEGURANÇA DA INFORMAÇÃO", executado pela UFC, em parceria com o SiDi e financiado pela Samsung Eletrônica da Amazônia Ltda., no âmbito da Lei de Informática no. 8.248/91.

Referências

Almohamade, S. S., Clark, J. A., and Law, J. (2021). Continuous user authentication for human-robot collaboration. In *16th International Conference on Availability, Reliability and Security (ARES)*.

Darabseh, A. and Pal, D. (2020). Performance analysis of keystroke dynamics using classification algorithms. In *3rd International Conference on Information and Computer Technologies (ICICT)*.

GSMA Intelligence (2022). The mobile economy 2022. Disponível em: `https://www.gsma.com/mobileeconomy/wp-content/uploads/2022/02/280222-The-Mobile-Economy-2022.pdf`. Acessado em 15 de agosto de 2023.

Kokal, S., Pryor, L., and Dave, R. (2022). Exploration of machine learning classification models used for behavioral biometrics authentication. In *8th International Conference on Computer Technology Applications (ICCTA)*, page 176–182.

Mekruksavanich, S. and Jitpattanakul, A. (2021). Deep learning approaches for continuous authentication based on activity patterns using mobile sensing. *Sensors*, 21(22):7519.

Scikit-learn developers (2023). Scikit-learn: Machine learning in python. Disponível em: `https://scikit-learn.org/stable/user_guide.html`. Acessado em 15 de agosto de 2023.

Shah, A. P. (2020). Towards engineering reliable keystroke biometrics systems. Master's thesis, University of Windsor. Disponível em: `https://scholar.uwindsor.ca/etd/8421/`.

Stragapede, G., Vera-Rodriguez, R., Tolosana, R., and Morales, A. (2023). Behavepassdb: Public database for mobile behavioral biometrics and benchmark evaluation. *Pattern Recognition*, 134.

Tahoun, E. (2021). Harnessing the power of generative models for mobile continuous and implicit authentication. Master's thesis, University of Waterloo. Disponível em: http://hdl.handle.net/10012/17133.

Thapliyal, A., Verma, O., and Kumar, A. (2022). Multimodal behavioral biometric authentication in smartphones for covid-19 pandemic. *International Journal of Electrical and Computer Engineering Systems*, 13(9).

Wagata, K. and Teoh, A. B. J. (2022). Few-shot continuous authentication for mobile-based biometrics. *Applied Sciences*, 12(20).

Yu, Y., Zhu, Y., Li, S., and Wan, D. (2014). Time series outlier detection based on sliding window prediction. *Mathematical Problems in Engineering*, 2014:1–14.

Zhao, Y., Nasrullah, Z., and Li, Z. (2019). Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20(96):1–7.

Zimperium (2023). 2022 Global Mobile Threat Report. Disponível em: https://www.zimperium.com/global-mobile-threat-report/. Acessado em 15 de agosto de 2023.