AUTENTICAÇÃO CONTÍNUA DE USUÁRIOS UTILIZANDO CONTADORES DE DESEMPENHO DO SISTEMA OPERACIONAL

César H. G. Andrade, Paulo Henrique Nellessen Gonçalves, Hendrio L S Bragança, Eduardo Souto

Instituto de Computação (ICOMP) – Universidade Federal do Amazonas (UFAM)

goersch@hotmail.com, {paulo, hendrio.luiz, esouto}@icomp.ufam.edu.br

Abstract. *Computer authentication systems based on login and password have been vulnerable to the action of unauthorized users. Currently, authentication techniques based on behavioral models predominantly use information extracted from mouse and/or keyboard to authenticate users. Operating system performance indicators can be used as an alternative. This work proposes an approach using data from performance indicators such as source data, CNN/LSTM networks for data classification, and reliability-based assessment methodology for the purpose of authenticating the user on an ongoing basis. The results obtained demonstrate the feasibility of using these attributes as the origin of the data to define a behavioral model. The best result obtained in this research is that 100% of genuine users are never inadvertently blocked and 100% of the imposters are detected after the average of three actions.*

Resumo. *Sistemas de autenticação de computadores baseados em credencias de contas (e.g. login e senha) têm sido vulneráveis à ação de usuários não autorizados. Atualmente, as técnicas de autenticação baseadas em modelos comportamentais predominantemente usam informações extraídas de mouse e/ou teclado para autenticar os usuários. Contadores de desempenho de sistema operacional podem ser utilizadas como alternativa. Este trabalho propõe uma abordagem utilizando dados de contadores de desempenho como dados de origem, redes CNN/LSTM para classificação dos dados e metodologia de avaliação baseada em nível de confiança com o propósito de autenticar o usuário de forma contínua. Os resultados obtidos demonstram a viabilidade do uso destes atributos como origem dos dados para definição de modelo comportamental. O melhor resultado obtido nesta pesquisa é que 100% dos usuários genuínos nunca são bloqueados inadvertidamente e 100% dos impostores são detectados após a média de três ações.*

1. Introdução

A maioria dos sistemas computacionais emprega uma abordagem de autenticação estática (*static authentication* - SA) baseada em credencias de contas (*logins* e senhas) como o único meio de verificação da autenticidade do usuário. Em geral, esse processo de autenticação ocorre somente na entrada do sistema pelo usuário. Um problema com esta abordagem é que o usuário pode deixar o computador sem sair da sessão ou bloquear seu acesso, possibilitando a um intruso acessar os recursos disponíveis [Ayeswarya et al. 2019].

Esse acesso não autorizado pode ocorrer de diferentes maneiras. Por exemplo, a usuária Alice faz *login* em seu computador usando suas credenciais (usuário e senha). Por alguma razão, Alice sai de perto do seu computador sem bloquear o sistema. Outro usuário, Bob, este denominado de impostor, passa a utilizar os recursos liberados pelo login de Alice de forma indevida. Alice também pode ter compartilhado a senha com Bob, ou Bob pode ter obtido a senha de Alice por meio de um ataque.

Neste cenário, os métodos de autenticação contínua (*Continuous Authentication* – CA) que recorrentemente avaliam a autenticidade do usuário podem ser utilizados para mitigar as limitações apresentadas pelos métodos estáticos de autenticação. Na literatura já existem diferentes mecanismos de autenticação que fornecem autenticação contínua para o usuário usando biometria fisiológica e comportamental [Oak et al. 2017] [Ayeswarya et al. 2019] [Ouch et al. 2017]. Os métodos que empregam biométrica fisiológica autenticam o usuário usando atributos pessoais tais como impressão digital, íris, retina e reconhecimento facial [Akash, S e Arya 2017]. Por outro lado, os métodos de autenticação baseado no comportamento avaliam as interações dos usuários com os dispositivos para extrair padrões comportamentais como, por exemplo, a partir de movimentos do mouse, dinâmica na digitação de textos ou reconhecimento de fala [Neja et al. 2018].

Uma desvantagem do uso da biometria fisiológica é a necessidade de hardware para executar a coleta de dados biométricos, acrescentando custo e outra camada de complexidade para o processo de *login* do usuário [Bailey, Okolica, e Peterson 2014]. Além disso, na autenticação contínua o usuário precisa interromper suas atividades constantemente para realizar o processo de autenticação. Por esta razão, muitos métodos de autenticação tem adotado a biometria comportamental, pois os atributos coletados podem ser obtidos silenciosamente, de forma transparente, sem atrapalhar o usuário genuíno e sem alertar o impostor que está sob avaliação [Mondal e Bours 2016].

Nos estudos de biometria comportamental, identificamos na literatura o emprego de atributos que estejam relacionadas às ações voluntárias dos usuários, tais como sequências de cliques de mouse [Mondal e Bours 2015] , dinâmica de digitação [Bours e Barghouthi 2009], ou um sistema de múltiplos atributos que empregam movimentos de mouse e dinâmica de digitação [Fridman et al. 2015].

Diferentemente das abordagens de autenticação existentes, este trabalho propõe a utilização de informações estatísticas relacionadas ao uso de hardware e software monitorados pelo sistema operacional para gerar modelos de autenticação. A ideia é usar as informações relacionadas ao uso dos recursos de um computador pelo usuário ao longo do tempo como o uso de memória, processador, rede, armazenamento e aplicações, para criar um perfil que possa ser usado para autenticar o usuário. A vantagem do uso destes atributos é que eles podem ser coletados de forma transparente, sem interferir na atividade do usuário. Além disso, os principais sistemas operacionais (e.g. Linux e Windows) já disponibilizam coletores nativos, não requerendo o desenvolvimento de softwares de coleta específicos.

Na literatura, muitos trabalhos se concentram na extração manual de características dos dados de origem (do inglês, *handcraft features* - HF) como, por exemplo, Mondal e Bours [2015] e Chen et al. [2016]. As desvantagens das abordagens manuais de extração de características são que os recursos criados ou selecionados

manualmente consomem tempo, são específicos do domínio e, por isso, exigem conhecimento especializado [Ronao e Cho 2016].

Para tratar este problema, este trabalho utiliza uma arquitetura de rede profunda híbrida, composta por camadas de convolução e por camadas de recorrência. As camadas de convolução realizam a extração automática de características (neste caso, correlações entre dados dos contadores de desempenho) e as camadas de recorrência são utilizadas para capturar características temporais dos dados processado pelas camadas convolucionais. Além disso, para diminuir a taxa de falsos positivos (i.e. usuários genuínos que foram classificados como impostores) na classificação, este trabalho emprega um modelo de confiança proposto por Bours e Barghouthi [2009]. O método proposto executa uma avaliação continuada das atividades do usuário com o objetivo de evitar o bloqueio de usuários genuínos e impedir que um impostor fique muito tempo agindo sem ser detectado.

O restante deste trabalho está organizado da seguinte forma: A Seção 2 apresenta os trabalhos relacionados. A Seção 3 descreve o método de autenticação contínua proposto. A Seção 4 apresenta os experimentos e resultados. Por fim, a Seção 5 apresenta algumas considerações finais.

2. Trabalhos Relacionados

Em estudos de biometria comportamental, as definições de quais atributos e critérios de avaliação usar na construção e avaliação dos modelos comportamentais são de suma importância. Muitos estudos utilizam atributos que estejam relacionadas às ações voluntárias dos usuários como sequências de cliques e movimentos de mouse [Y. Nakkabi, 2010][Feher et al. 2012] [Cai et al. 2014], dinâmica na digitação de textos [Bours e Barghouthi 2009], ou atributos coletados pela combinação de ações como teclado e mouse [Xiaojun et al. 2013b] [Fridman et al. 2015]. Outros estudos exploram atributos que podem não estar associados diretamente às ações diretas dos usuários, mas sofrem influências destas ações, como pela análise da sequência de chamadas de sistemas (*system calls*) [Song et al. 2013] ou do tráfego de rede [Chen et al. 2016].

Entre os estudos de CA para biometria comportamental, predominam os que empregam classificadores baseados em distâncias Euclidiana e Manhattan [Schulz 2006][Davoudi e Kabir 2009][Malatras et al. 2017] [Shen et al. 2012], vetores de suporte [Xiaojun et al. 2013b] [Chen et al. 2016] [Friedman et al. 2015] e redes neurais artificiais [Ahmed and Traore 2014] [Roth et al. 2014] [Shen et al. 2012] [Mondal e Bours 2013] [Mondal e Bours 2015].

Considerando os critérios de avaliação empregados nos estudos de CA, a maioria dos estudos avalia suas abordagens usando métricas como taxas de falso positivo, falso negativo, e de erro, além da acurácia. [Fridman et al. 2015] [Bailey, Okolica, e Peterson 2014]. Entretanto, os trabalhos que utilizam esses critérios falham por não considerar o fato que um usuário não consegue manter um padrão de comportamento ao longo do tempo. Para um sistema de CA, na verdade, não é apenas importante saber se um impostor é detectado, mas quando ele é detectado, ou seja, quanta atividade ele foi capaz de realizar antes da detecção [Mondal e Bours 2017].

Para tratar este problema alguns trabalhos propõem a utilização de um modelo de confiança [Bours e Barghouthi 2009] [Deutschmann e Lindholm 2013] [Mondal e Bours 2014] [Mondal e Bours 2017]. Nesse método de avaliação, o comportamento do

usuário é avaliado continuamente e a classificação entre usuário genuíno ou impostor é dada a partir das ações executadas pelo usuário. Um modelo de confiança é calculado utilizando métricas como o número médio de ações de um impostor (*Average Number of Imposter Actions* - ANIA) e do número médio de ações genuínas (*Average Number of Genuine Actions* - ANGA). Desta forma, estas métricas indicam o quanto um impostor pode fazer antes de ser bloqueado e quanto um usuário genuíno pode fazer antes de ser injustamente bloqueado pelo sistema.

De acordo com o nosso conhecimento, somente o trabalho de Malatras et al. [2017] apresentam uma abordagem de identificação de usuários (e não de autenticação CA) utilizando contadores de desempenho do sistema operacional como o tempo total da CPU, a porcentagem da memória livre e usada, número de conexões TCP ativas, e o número total de segmentos de dados enviados e recebidos na camada TCP.

Além disso, a utilização de aprendizagem profunda em biometria comportamental com dados de contadores de desempenho ainda não foi empregada. Neste cenário, este trabalho propõe aplicar técnicas de aprendizagem profunda em autenticação contínua baseada em dados de atributos de ações indiretas obtidos a partir da observação de contadores de desempenho do sistema operacional.

3. Autenticação Contínua Baseada em Contadores de Desempenho do Sistema Operacional

Esta seção descreve o método de autenticação contínua. A seção 3.1 descreve os contadores de desempenho do sistema operacional. A seção 3.2 detalha a fase de pré-processamento, como limpeza dos dados, segmentação e normalização. Por fim, a seção 3.3 apresenta o modelo de autenticação, que utiliza uma rede neural profunda na tarefa de classificação e o modelo de confiança, empregado como critério de avaliação.

3.1. Contadores de Desempenho do Sistema Operacional

Nesse trabalho, os registros estatísticos dos contadores de desempenho (*performance counters – PCs*) de sistema operacional (SO) são utilizados como atributos para gerar um modelo capaz de realizar a autenticação contínua de usuários. Esses contadores têm como objetivo quantificar diferentes tipos de eventos do SO, a fim de realizar análises de desempenho, otimização de algoritmos ou ajuste do sistema (*tunning*). Os eventos disponíveis e o número de contadores dependem do tipo e versão do sistema operacional. Como exemplos de alguns eventos que podem ser medidos são:

- Interface de rede: número de pacotes enviado e recebidos, número de bytes enviados e recebidos.
- Disco físico: número de leituras e escritas, volume de leituras e escritas.
- Processador: total de bytes de entrada e saída, percentual de tempo de processador.
- Processo: número de threads, percentual de tempo de processador, memória ocupada.
- Memória: memória disponível, número de páginas.

Diversos softwares podem ser utilizados com objetivo de proceder com a coleta automatizada. No sistema operacional Linux, pode-se utilizar o programa "PERF"[1], enquanto nos sistemas operacionais Windows, o programa "PERFMON"[2] pode ser utilizado para a mesma finalidade. Além disso, mesmo em ambientes virtuais como o Vmware já existem contadores de desempenho nativos nas máquinas virtuais como o programa "vRealize Hyperic"[3].

3.2. Pré-Processamento dos Dados

Ao longo do processo de coleta de dados, alguns problemas podem afetar a qualidade e a estrutura dos dados, dificultando a geração dos modelos de autenticação. Entre esses problemas podem-se destacar três deles: *i)* falta de padronização dos atributos extraídos; *ii)* diferença entre as escalas dos dados, e *iii)* dimensionalidade dos dados.

Para tratar estes problemas, a etapa de pré-processamento utiliza um conjunto de técnicas para realizar a limpeza, padronização e normalização dos dados. Na etapa de limpeza dos dados, extraímos os acentos e caracteres especiais constantes nos nomes dos atributos, que podem gerar erros durante o pré-processamento dos dados. Além disso, são excluídos todos os atributos que não variam ao longo da coleta, i.e., desvio padrão igual a zero, considerando todos os usuários coletados; e por fim, transformação de valores NULL em "0";

Como os dados são coletados em computadores diferentes, para um mesmo padrão de sistema operacional, configurações de linguagem distintas podem gerar nome de atributos diferentes para o mesmo atributo. Por essa razão, um script de padronização é utilizado para evitar que diferença na apresentação atributos entre as diferentes coletas gerem viés durante as etapas seguintes.

Além disso, os atributos também passam por um processo de normalização, visto que atributos diferentes utilizam escalas diferentes. Assim, atributos numéricos são normalizados dentro de uma escala de valores, atributos simbólicos precisam ser codificados em valores numéricos e valores desconhecidos precisam ser preenchidos usando de métodos como médias dos valores dos atributos. Os dados foram normalizados pelo desvio padrão *(standard score/z-values*).

3.3. Modelo de Autenticação

Em aprendizagem de máquina, os classificadores são capazes de examinar os dados de itens para determinar a qual dos *N* grupos (classes) cada item pertence. Frequentemente, os algoritmos de classificação produzem um vetor de probabilidades que representam as probabilidades do item de dados pertencer a cada classe. No caso de autenticação contínua, podemos simplesmente definir duas classes: usuário legítimo e impostor. Como resultado, um algoritmo de classificação pode ser usado para gerar um modelo de autenticação personalizado que pode ser interpretado como uma assinatura única capaz de autenticar um usuário.

---

[1] http://www.brendangregg.com/perf.html

[2]https://docs.microsoft.com/pt-br/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc749154(v=ws.11)

[3] https://www.vmware.com/products/vrealize-hyperic.html

A etapa de classificação é composta de essencialmente de dois componentes: Treinamento e Predição (Figura 2). A fase de treinamento tem como produto os modelos personalizados que serão utilizados na etapa de predição. Para cada usuário que se deseja autenticar e para cada algoritmo de classificação que será avaliado, deve-se gerar um modelo personalizado que pode ser interpretado como uma assinatura única capaz de autenticar o usuário treinado. Uma vez que o modelo personalizado tenha sido treinado, o ciclo completo de pré-processamento, classificação e avaliação do modelo de confiança já pode ser implementado num *pipeline* contínuo.

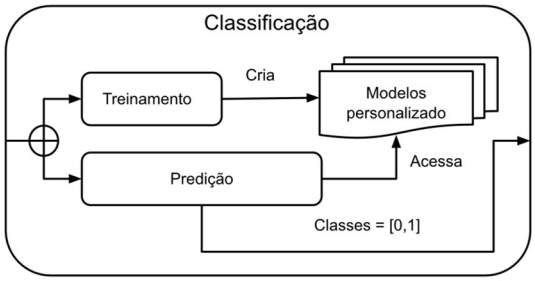

Figura 2. Processo de classificação.

3.3.1. Arquitetura de Rede DeepConvLSTM

Neste artigo, o modelo de classificação utiliza uma arquitetura de rede neural profunda denominada DeepConvLSTM, proposta por Ordóñez [2016]. A arquitetura combina camadas convolucionais, as quais atuam como extratoras de características (neste caso, correlações entre dados dos contadores de desempenho) e fornecem representações abstratas dos dados de entrada. Em seguida, camadas de recorrência são empregadas para capturar características temporais dos dados processado pelas camadas convolucionais. É importante salientar que as redes neurais recorrentes podem receber como entrada os dados brutos coletados pelos contadores de desempenho. No entanto, aplicar a estes dados brutos técnicas de extração de características, na maioria das vezes, leva a um desempenho superior do classificador [Palaz et al. 2015].

A descoberta manual de características requer conhecimento especializado, e a escala dos dados brutos produzidos pelos contadores de desempenho é um fator limitador. Por essa razão, as redes convolucionais (*Convolutional Neural network* - CNNs) podem ser utilizadas para tratar estes desafios [Yang et al. 2015]. Outro aspecto importante é que as dependências internas entre os dados dos contadores de desempenho podem trazer informações de contexto significativas ou padrões desconhecidos que podem ser úteis para identificar comportamentos. Por exemplo, uma correlação entre o uso de navegador, interface de rede e processador. Para tirar proveito destas correlações, este artigo adiciona duas camadas recorrentes LSTM (*Long Short-Term Memory*) [Hochreiter e Schmidhuber, 1997]. Segundo Ordóñez [2016], o uso de duas camadas recorrentes em profundidade é suficiente para capturar as relações temporais das características.

A entrada para a rede neural profunda consiste em segmentos da série temporal para vários canais, onde cada atributo (contador de desempenho) corresponde a um canal. Na Figura 3, o número de canais é representado por D e S é o tamanho do segmento por canal (número de amostras por canal). Assim, os dados de entrada são transformados através de três camadas convolucionais, os operadores de convolução são

exibidos como "∗", o qual é aplicado a um kernel (K^1, K^2 e K^3), representados pelos retângulos vermelhos na Figura 3. Neste caso, o kernel é representado por um vetor de pesos de tamanho 4. O kernel se desloca sobre os dados de entrada, executando uma multiplicação elementar com a parte da entrada em que está atualmente, e então somando os resultados em um único ponto de saída. Essas camadas convolucionais empregam como função de ativação unidades lineares retificadas (ReLUs) para calcular os mapas de características.

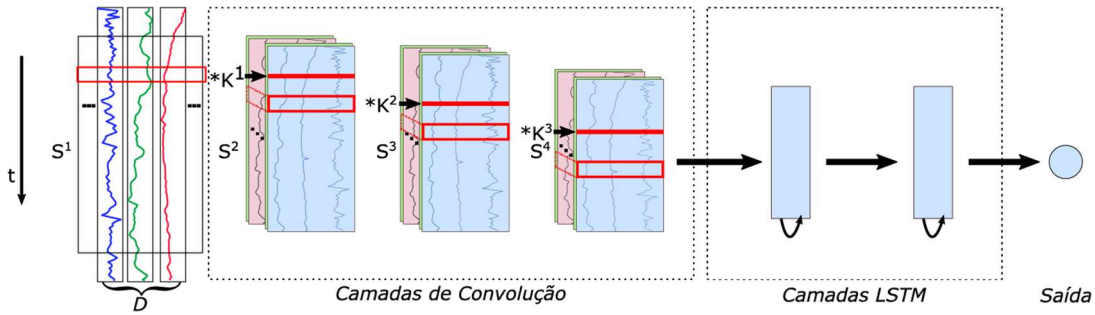

Figura 3. Rede DeepConvLSTM

Os dados resultantes das camadas de convolução são passados para as camadas densas recorrentes. As unidades de um LSTM são usadas como unidades de construção e, neste caso, cada camada recorrente é composta por 128 unidades, com saída de tamanho 20. A saída da rede é obtida através de uma camada densa de 1 unidade com a função de ativação *softmax,* que contém a probabilidade da amostra pertencer ao usuário genuíno ou impostor.

3.3.2. Modelo de Confiança

O modelo de confiança foi proposto inicialmente por Bours e Barghouthi [2009] e fundamenta-se na premissa que mesmo um usuário genuíno pode agir com um padrão diferente do habitual gerando, deste modo, falsos negativos no processo de classificação.

No modelo de confiança, o comportamento do usuário avaliado é comparado continuamente ao modelo do usuário genuíno. Caso, uma amostra do usuário seja classificada como genuína, então a confiança do sistema na autenticidade deste usuário aumenta, tal procedimento é chamado de recompensa. Caso contrário, a confiança será reduzida, i.e. o modelo é penalizado.

Penalidades sucessivas podem levar o nível de confiança a ultrapassar um limite estabelecido, o que ocasiona o bloqueio do sistema até que nova autenticação (por exemplo, autenticação por senha) seja efetuada. Por outro lado, espera-se que o usuário genuíno gere mais recompensas sucessivas ao longo de um período de avaliação se comparado a um usuário impostor.

O modelo de confiança ideal busca detectar um usuário impostor num menor tempo possível, bem como deve-se evitar bloquear indevidamente um usuário legítimo. Para alcançar estes objetivos é necessário medir o desempenho em termos de número médio de ações de um impostor (*Average Number of Imposter Actions* - ANIA) e do número médio de ações genuínas (*Average Number of Genuine Actions* - ANGA), onde ANIA deve ser o mais baixo possível, enquanto ANGA deve ser alto.

A função de cálculo de ANGA é dada por: $\mathrm{ANGA} = \frac{1}{\mathrm{n}}\sum_{1}^{\mathrm{n}}\frac{\mathrm{ag}}{\mathrm{bg}}$, onde *n* é o número usuários, *ag* é o número de ações genuínas de cada usuário e *bg* é o número de vezes que o usuário genuíno é bloqueado indevidamente (bloqueio genuíno). A função de cálculo de ANIA é dada por: $\mathrm{ANIA} = \frac{1}{\mathrm{n}}\sum_{1}^{\mathrm{n}}\frac{\mathrm{ai}}{\mathrm{bi}}$, onde *n* é o número usuários, *ai* é o número de ações impostoras de cada usuário e *bi* é o número de bloqueios impostores. O desejável é que haja um número reduzido de bloqueios genuínos (*bg*) e um elevado número de bloqueio impostores (*bi*). Quando não há bloqueios genuínos, bg é zero e ANGA tende ao infinito.

4. Resultados e Experimentos

Este artigo propõe uma abordagem onde é avaliado o emprego de biometria comportamental para fins de CA, utilizando como atributos os contadores de desempenho de sistema operacional, metodologia de classificação baseada em redes de convolução e um modelo de confiança como método de avaliação. Até o nosso conhecimento, abordagens com este escopo ainda não foram estudadas. Assim, para validar os resultados dos classificadores empregados e estabelecer uma referência comparativa, os resultados obtidos são comparados ao estudo de Mondal e Bours [2015].

Esta seção apresenta a base de dados que será utilizada para comparar os resultados, detalha as metodologias de separação de dados e os processos de verificação do nível de confiança, além de apresenta os parâmetros que serão empregados nos algoritmos. Por fim, serão discutidos os resultados encontrados.

4.1. Bases de Dados

Os experimentos realizados para avaliar o método proposto utilizam duas bases de dados: a base de dados 1, corresponde a base de dados de movimentos de mouse obtidas em Nakkabi [2010] e a base de dados 2, corresponde a uma base de dados coletada pelo próprio autor, com contadores de desempenho de sistema operacional que são usados para produzir dados de séries temporais multidimensionais, onde cada amostra corresponde a um vetor formado por contagens de eventos no momento da amostragem.

A base de dados 1 (Nakkabi dataset) é composta de dados gerados por voluntários, aos quais foi pedido que usassem o computador de maneira normal, sem quaisquer restrições às tarefas que deveriam executar. É composta da coleta de movimento de mouse de 49 usuários, onde para cada ação do mouse de um voluntário, o software de coleta de dados armazenou os seguintes atributos: *i)* tipo de ação (1: movimento do mouse; 2: silêncio; 3: *point* e *click*; ou 4: *drag and drop*; *ii)* distância percorrida em pixels; *iii)* tempo decorrido do movimento, unidade em segundo (com um intervalo de amostragem de 0,25 segundos); *iv)* direção do movimento.

A base de dados 2 de PCs está formada por um vetor de características composto por 161 atributos coletados nos computadores do Departamento de Tecnologia da Informação de uma organização pública e teve como usuários participantes um grupo de analistas de sistemas e de programadores, todos voluntários e para os quais não foram apresentadas quaisquer restrições às tarefas que deveriam executar. O ciclo de coleta de cada amostra foi de 5 segundos e teve uma duração média de aproximadamente de 24 (vinte e quatro) horas para cada um dos 26 usuários.

4.1.1. Divisão dos Dados

Os dados foram separados na proporção de 50% dos dados de cada usuário para treino e os outros 50% para teste. Como os dados foram separados a priori, pode-se garantir que as amostras que foram aplicadas as fases de teste são desconhecidas pelo classificador proposto. No caso específico da base de dados Nakkabi, para minimizar o desbalanceamento de um único usuário com um número maior de dados coletados, limitou-se em 20.000 o tamanho máximo da base de treino. Duas estratégias de verificação do nível de confiança foram implementadas e se diferenciam basicamente na composição dos conjuntos de treino e teste, conforme descrito abaixo:

1. Verificação V1: neste caso, a parte impostora dos dados de treinamento é tomada de todos os impostores (Figura 5), sendo que todos contribuem aproximadamente com a mesma quantidade de dados para o treinamento do classificador, treinamento ($T_N/(N-1)$), onde N é número de usuários e T_N.é o tamanho da base de treino do usuário N. O teste (Figura 6) é feito com todos os dos dados que não foram usados para treinamento.
2. Verificação V2: para cada usuário genuíno, é necessário separar os impostores que farão parte do conjunto de treino (50%) e os que farão parte do conjunto de teste (50%).

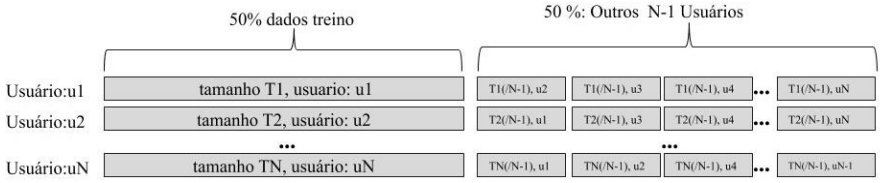

Figura 5. Separação dos dados de treino, verificação V1.

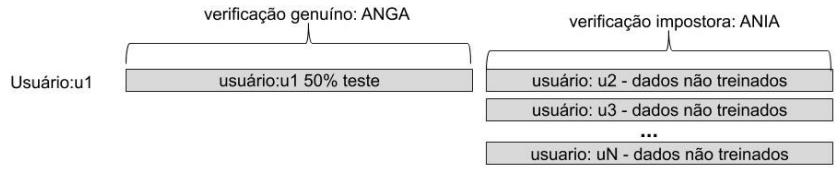

Figura 6. Verificação V1, exemplo da separação dos dados de teste para usuário u1.

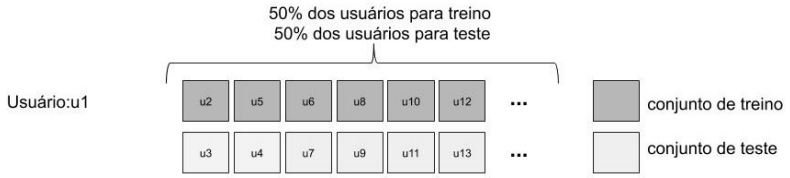

Figura 7. Verificação V2, exemplo da separação dos usuários de conjunto treino/teste.

Na verificação v2, para preparação da base de treino de cada usuário, 50% vem dos dados de treino do usuário que estamos treinando, e os outros 50% vem dos usuários que foram selecionados para o conjunto de treino (Figura 8). Neste cenário, os dados dos usuários selecionados contribuem com quantidade de dados igual para o treinamento ($T_N/(^{N-1}/_2)$), onde N é número de usuários e T_N.é o tamanho da base de treino do usuário N.

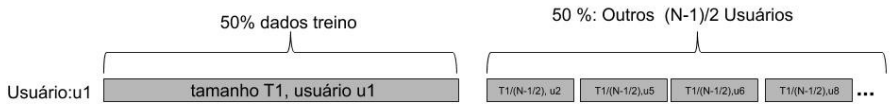

Figura 8. Exemplo de separação da base de treino do usuário u1 para verificação V2

Para teste, o usuário genuíno utiliza os 50% de dados separados para teste, e a parte impostora é tomada de 50% dos impostores do conjunto de teste, utilizando todos os dados destes impostores (figura 9).

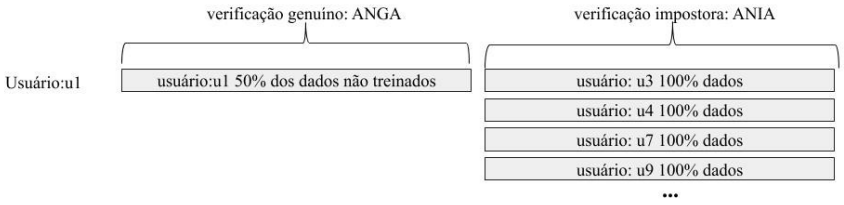

Figura 9. Exemplo de separação dos dados de teste do usuário u1 para verificação V2

4.2. Parâmetros dos Níveis de Confiança e da Rede de Convolução:

As Tabelas 1 e 2 apresentam os parâmetros utilizados na implementação do modelo de nível de confiança e para a rede de convolução, respectivamente.

Tabela 1.: Parâmetros empregados no modelo de nível de confiança.

Limiar de recompensa ou penalidade	Tc=0,5
Limiar de penalidade intermediária	Tpi = 0,4
Função de recompensa	$f_{recompensa}(x_i) = 1\ X\ x_i$, onde x_i é a estimativa gerada pelo classificador para a amostra i.
Função de penalidade	$f^1_{penalidade}\ (x_i) = 1 - x_i$, $f^2_{penalidade}\ (x_i) = 1$, onde x_i é a estimativa gerada pelo classificador para a amostra

Tabela 2.: Parametros empregados na rede profunda DeepConvLSTM.

Rede de convolução 1D	filters=64 kernel_size=4 padding='same' kernel_regularizer=l2(0,01) kernel_initializer='lecun_uniform'
TimeDistributed	kernel_regularizer=l2(0,01)
LSTM	units=128
Compilação	loss='binary_crossentropy' optimizer=Adam(lr=0.0001)
Fit	batch_size=64

4.3. Resultados

Os resultados serão apresentados num formato adaptado de uma matriz de confusão, expressando os valores de ANGA e ANIA encontrados. Serão comparados os resultados com o trabalho de Mondal e Bours [2015]. Os resultados também são analisados com base na avaliação do limite mínimo do nível de confiança estabelecido para bloqueio.

A Tabela 3 mostras os resultados obtidos por Mondal e Bours para as verificações V1 e V2. Em V1 os usuários genuínos sem bloqueios passam de 41 (85,41%) para 46 (95,83%), os impostores corretamente detectados passam de 42(87,5%) para 43 (89,58%) e com ANIA reduzindo de 94 para 68 ações. Em V2, o mesmo processo de melhoria pode ser observado, com os usuários genuínos sem bloqueios passam de 41 (85,41%) para 49 (100%), os impostores corretamente detectados passam de 43(89,58%) para 49 (100%) e com ANIA reduzindo de 165 para 109 ações.

Tabela 3.: Resultados obtidos por Mondal e Bours para base de dados 1, e verificações V1 e V2.

V1		USUÁRIOS			
LIMITE MÍNIMO	CLASSIFICAÇÃO	Genuino		Impostor	
VARIÁVEL	Genuino	46		6	
		ANGA:	∞	ANIA:	1258
	Impostor	3		43	
		ANGA:	2066	ANIA:	68
90	Genuino	41		7	
		ANGA:	∞	ANIA:	172
	Impostor	8		42	
		ANGA:	3729	ANIA:	94

V2		USUÁRIOS			
LIMITE MÍNIMO	CLASSIFICAÇÃO	Genuino		Impostor	
VARIÁVEL	Genuino	49		1	
		ANGA:	∞	ANIA:	366
	Impostor			48	
		ANGA:		ANIA:	109
90	Genuino	41		6	
		ANGA:	∞	ANIA:	781
	Impostor	8		43	
		ANGA:	32311	ANIA:	165

Tabela 4.: Resultados obtidos neste trabalho para base de dados 1, e verificações V1 e V2.

V1		USUÁRIOS			
LIMITE MÍNIMO	CLASSIFICAÇÃO	Genuino		Impostor	
VARIÁVEL	Genuino	49		2	
		ANGA:	∞	ANIA:	59
	Impostor			47	
		ANGA:		ANIA:	5
90	Genuino	49		2	
		ANGA:	∞	ANIA:	69
	Impostor			47	
		ANGA:		ANIA:	12

V2		USUÁRIOS			
LIMITE MÍNIMO	CLASSIFICAÇÃO	Genuino		Impostor	
VARIÁVEL	Genuino	49		9	
		ANGA:	∞	ANIA:	176
	Impostor			40	
		ANGA:		ANIA:	32
90	Genuino	49		9	
		ANGA:	∞	ANIA:	332
	Impostor			40	
		ANGA:		ANIA:	186

A Tabela 4 descreve os resultados obtidos usando o método proposto para as verificações V1 e V2 para a base de dados 1. Considerando o limite mínimo variável e comparando V1 e V2, enquanto em V1 foram corretamente detectados 47 impostores (97,91% dos impostores), em V2 observamos um resultado onde houve redução deste número para 40 impostores (83,333%), e com prejuízo também para os valores obtidos para ANIA, elevando de 5 para 32 ações.

Tabela 5.: Resultados obtidos neste trabalho para base de dados 2, e verificações V1 e V2.

V1		USUÁRIOS			
LIMITE MÍNIMO	CLASSIFICAÇÃO	Genuino		Impostor	
VARIÁVEL	Genuino	26			
		ANGA:	∞	ANIA:	
	Impostor			26	
		ANGA:		ANIA:	3
90	Genuino	26			
		ANGA:	∞	ANIA:	
	Impostor			26	
		ANGA:		ANIA:	11

V2		USUÁRIOS			
LIMITE MÍNIMO	CLASSIFICAÇÃO	Genuino		Impostor	
VARIÁVEL	Genuino	26		4	
		ANGA:	∞	ANIA:	345
	Impostor			22	
		ANGA:		ANIA:	16
90	Genuino	26		4	
		ANGA:	∞	ANIA:	406
	Impostor			22	
		ANGA:		ANIA:	77

A Tabela 5 apresenta os resultados obtidos para o método proposto, usando os métodos de verificação V1 e V2 para a base de dados de indicadores de desempenho. Em V1, 100% dos usuários genuínos não sofreram bloqueios indevidos e 100% dos impostores foram bloqueados, e ANIA alcançou a média de 3 ações, o melhor valor encontrado neste estudo.

4.3.1. Discussões

Análise 1: V1 X V2:

Comparando os resultados obtidos a partir da análise de V1 e V2. No trabalho de Mondal e Bours, os números da Tabela 1 indicam que V2 supera V1 quanto a capacidade de reconhecer corretamente os genuínos e os impostores, embora tenha havido um aumento no ANIA. Por outro lado, os resultados obtidos neste trabalho, em ambas as bases de dados, apontam que V1 supera V2, principalmente nos valores obtidos para ANIA e a quantidade de impostores corretamente classificados. Estudos adicionais são necessários com o objetivo de melhorar a capacidade da rede DeepConvLSTM em reconhecer impostores quando seus dados não são apresentados para treinamento do classificador, cenário visto em V2.

Análise 2: Limite mínimo variável X limite mínimo fixo (90):

Dos resultados experimentais, observamos que para o limite mínimo do nível de confiança fixado em 90, nenhum usuário genuíno foi bloqueado. A ausência de bloqueio genuínos possibilita que o limite mínimo do nível de confiança possa ser ajustado a maior. Este ajuste possibilita uma redução no número médios de ações impostoras (ANIA) e uma possível redução nos impostores não reconhecidos. Estas duas possibilidades foram observadas nestes resultados.

Análise 3: Base de dados 1 X Base de dados 2:

Os resultados obtidos quando comparamos as diferentes bases de dados não são conclusivos. Embora os números indicarem bons resultados para a base de dados 2, para a verificação V1, com 100% de genuínos corretamente classificados e 100% de impostores corretamente classificados, quando a mesma base de dados é analisada para a verificação V2, este percentual de genuínos cai para 88%. Porém esta redução pode ser decorrente do classificador DeepConvLSTM empregado, pois resultados similares são encontrados quando analisamos a base de dados 1 com o mesmo classificador.

5. Conclusões e Trabalhos Futuros

Esse trabalho demonstrou a viabilidade de uma nova abordagem para autenticação contínua de usuários utilizando dados de contadores de desempenho de sistemas operacionais. Os índices de assertividade em identificar o usuário genuíno sem seu bloqueio indevido propicia que possam ser elevados os limites mínimos do nível de confiança e, assim, reduz o número de impostores não detectados. Nos resultados apresentados, nenhum genuíno foi bloqueado indevidamente. Além disso, todos os impostores foram detectados com no mínimo de 3 ações realizadas por ele.

Tendo como foco encontrar um modelo que apresentem melhor desempenho frente aos objetivos de maximizar o tempo de uso por um usuário real sem que seja bloqueado, e minimizar o tempo para detectar um usuário impostor, outras decisões ainda precisam ser melhores investigadas, tais como:

- Selecionar o conjunto de contadores de desempenho que melhor representem o modelo comportamental do usuário, com um custo computacional menor.
- Aumentar o tempo de coleta e o número de usuários.
- Investigar outros algoritmos de nível de confiança que minimize o tempo de detecção de um usuário impostor.

Referência bibliográfica

Ayeswarya, S. e Norman, J. (2019) “A survey on different continuous authentication systems”, International Journal of Biometrics, vol. 11, p. 67-99.

Bailey, K. O., Okolica, J. S., e Peterson, G. L. (2014) “User identification and authentication using multi-modal behavioral biometric”, Comuters & Security, 43, p. 77-89.

Cai Z., Shen, C. e Guan, X. (2014) “Mitigating Behavioral Variability for Mouse Dynamics: A Dimensionality-Reduction-Based.”, IEEE Transactions on Human-Machine Systems Volume, vol. 44, p. 244 –255.

Chen, A., Brahma, P., Wu, D. O., Ebner, N., Matthews, B., Crandall, J., Xuetao, W., Faloustsos, M. e Oliveira, D. (2016) “Cross-layer personalization as a first-class citizen for situation awareness and computer infrastructure security”, Proceedings of the 2016 New Security Paradigms Workshop on – NSPW, p. 23-35.

Deutschmann, I. e Lindholm, J. (2013) “Behavioral biometrics for DARPA’s Active Authentication program”. In Lecture Notes in Informatics (LNI), Proceedings - Series of the *Gesellschaft fur Informatik* (GI).

Schulz, D. A. (2006) “Mouse curve biometrics”, Proceedings of the Biometrics Symposium, Biometric Consortium Conference, IEEE, p. 1–6.

Davoudi, H. e Kabir, E. (2009) “A new distance measure for free text keystroke authentication”, 14th International CSI Computer Conference (CSICC'09), IEEE, p. 570–575.

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

Fridman, L., Stolerman, A., Acharya, S., Brennan, P., Juola, P., Greenstadt, R., e Kam, M. (2015) “Multi-modal decision fusion for continuous authentication”, Computers & Electrical Engineering, 41, p. 142–156.

Malatras, A., Geneiatakis, D., e Vakalis, I. (2016) “On the efficiency of user identification: a system-based approach”, International Journal of Information Security, 16 (6), p. 653–671.

Mondal, S., e Bours, P. (2014) “Continuous authentication using fuzzy logic”, Proceedings of the 7th International Conference on Security of Information and Networks - SIN ’14.

Mondal, S., e Bours, P. (2015) “A computational approach to the continuous authentication biometric system”, Information Sciences, 304, p. 28–53.

Mondal, S., e Bours, P. (2016) “Combining keystroke and mouse dynamics for continuous user authentication and identification”, IEEE International Conference on Identity, Security and Behavior Analysis (ISBA).

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

Hochreiter, S., e Schmidhuber, J. (1997) “Long Short-Term Memory”, Neural Computation, 9(8), p. 1735–1780.

Neha, e Chatterjee, K. (2018) “Biometric re-authentication: an approach towards achieving transparency in user authentication”, Multimedia Tools and Applications.

Rajvardhan, O. e Mrunmayee, K. (2017) “A Novel Architecture for Continuous Authentication using Behavioural Biometrics”, International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), p. 767-771.

Ordóñez, F. J., e Roggen, D. (2016). Deep Convolutional and LSTM Recurrent Activity Recognition. Sensors, p. 1-25.

Ouch, R., Garcia-zapirain, B. e Yampolskiy, R. (2017). “Multimodal Biometric Systems : a systematic review”, Computer Science, p. 439–44.

Palaz, D., Magimai.-Doss, M. e Collobert, R. Analysis of CNN-based Speech Recognition System using Raw Speech as Input. In Proceedings of the 16th Annual Conference of International Speech Communication Association (Interspeech), Dresden, Germany, 6–10 September 2015, p. 11–15.

Bours, P. e Barghouthi, H. (2009). “Continuous Authentication using Biometric Keystroke Dynamics”. The Norwegian Information Security Conf. (NISK), p. 1–12.

Akash, S e Arya (2017). “Applications and Security of Keystroke Dynamics for User Authentication”. Arya International Journal of Computer e Mathematical Sciences IJCMS ISSN. p. 2347-8527.

Shen, C., Cai, Z. and Guan, X. (2012) ‘Continuous authentication for mouse dynamics: a pattern-growth approach’, Proc. Int. Conf. Dependable Syst. Networks.

Song, Y., Salem, B., Hershkop, S., e Stolfo, S. J. (2013) “System level user behavior biometrics using Fisher features and Gaussian mixture models”. 2013. - p. 52-59.

Sun, R., Yuan, X., He, Pan., Zhu, Q., Chen, A., Gregio, A., Oliveira, D., e Li, X.. (2017). “Learning Fast and Slow: PROPEDEUTICA for Real-time Malware Detection”.

Xiaojun, C., Zicheng, X., Yiguo, P. e Jinqiao, S.. (2013a). “A continuous re-authentication approach using ensemble learning”. *Procedia Computer Science* 17. p 870–78.

Nakkabi, Y., Traoré, I. e Ahmed, A., (2010). “Improving mouse dynamics biometric performance using variance reduction via extractors with separate features”, IEEE Trans. Syst. Man Cybern. – Part A: Syst. Hum. 40, p 1345–1353.

Nan, Z., Aaron, P. e Haining, W. (2011). “An Efficient User Verification System via Mouse Movements”.