WEAPON: Uma Arquitetura para Detecção de Anomalias de Comportamento do Usuário

Andre L. B. Molina[1], Vinícius P. Gonçalves[1], Rafael T. de Sousa Jr.[1], Felipe T. Giuntini[2], Gustavo Pessin[3], Rodolfo I. Meneguette[4], Geraldo P. Rocha Filho[5]

[1]Departamento de Engenharia Elétrica, Universidade de Brasília – UnB
Brasília – DF – Brasil.

[2]Instituto de Ciência da Computação, Universidade Federal do Amazonas – UFAM
Manaus – AM – Brasil.

[3]Instituto Tecnológico Vale – ITV
Ouro Preto – MG – Brasil.

[4]Instituto de Ciências Matemática e de Computação Universidade de São Paulo (USP)
São Carlos, SP – Brasil

[5]Departamento de Ciência da Computação, Universidade de Brasília – UnB
Brasília – DF – Brasil.

molina.albmolina@gmail.com, {vpgvinicius, desousa, geraldof}@unb.br

felipegiuntini@gmail.com, meneguette@icmc.usp.br, gustavo.pessin@itv.org

Abstract. *User behavior anomaly detection has been a successful measure contributing to cybersecurity. Much of the related literature addresses this issue without considering the individualization of users when analyzing logs generated by network and system protection devices. This paper presents WEAPON, an architecture for the detection of behavior anomalies, considering the individuality of each user, based on Wide and Deep Convolutional LSTM Autoencoders. When compared to other approaches, WEAPON proved to be more efficient, surpassing by up to 7% the second best model in the anomaly detection process.*

Resumo. *A detecção de anomalias de comportamento de usuário vem sendo aplicada com sucesso no campo da segurança cibernética. Grande parte da literatura correlata aborda essa questão sem considerar a individualização dos usuários ao analisar logs dos dispositivos de proteção de redes e sistemas. Este trabalho apresenta o WEAPON, uma arquitetura para a detecção de anomalias de comportamento, considerando a individualidade de cada usuário, com base em Wide and Deep Convolutional LSTM Autoencoders. Quando comparado com outras abordagens, o WEAPON mostrou ser mais eficiente, superando em até 7% o segundo melhor modelo no processo de detecção de anomalias.*

1. Introdução

A detecção de anomalias, também conhecida como detecção de novidade, é o processo de identificar padrões comportamentais inesperados, de modo a possibilitar a suspeita de que uma ação maliciosa foi realizada [Aggarwal 2017, Thudumu et al. 2020,

Pang et al. 2021]. Dada sua abrangência e versatilidade, a área de detecção de anomalias tem sido pesquisada em diversos domínios de aplicação [Patcha and Park 2007, Prasad et al. 2009, Agrawal and Agrawal 2015, Ahmed et al. 2016, Aggarwal 2017, Kwon et al. 2017, Thudumu et al. 2020, Pang et al. 2021], apresentando uma extensa visão acerca das técnicas de detecção de anomalias. Dentre tais domínios, a detecção de anomalias vem ganhando destaque em: (i) análise de dados multidimensionais [Prarthana and Gangadhar 2017, Thudumu et al. 2020]; (ii) detecção de intrusão ou detecção de ataques [Junior et al. 2016, Jin et al. 2017, Abu Sulayman and Ouda 2019, Pokhrel et al. 2019, Kim et al. 2019, Vilaça et al. 2019]; (iii) séries temporais para identificar padrões inesperados [Sadik and Gruenwald 2014, Gupta et al. 2014, Thomé et al. 2020, Shin et al. 2021]; e (iv) comportamento de perfil dos usuários [Geraldo Filho et al. 2013, Filho et al. 2014, Gao et al. 2017, Jin et al. 2017, Abu Sulayman and Ouda 2019, Pokhrel et al. 2019, Pang et al. 2021].

A detecção de anomalias de comportamento de usuário vem sendo objeto de atenção no campo da segurança cibernética [Zhang et al. 2021, Pang et al. 2021]. Nesse sentido, um usuário apresenta comportamento anômalo quando é possível estabelecer que ele está agindo de maneira incomum com base em um padrão de comportamento pré-estabelecido [Pokhrel et al. 2019]. Assim, mecanismos de segurança incapazes de endereçar contextos comportamentais não são adequados para detectar anomalias de comportamento de usuário. Por outro lado, o alto volume de *logs* de segurança gerados a partir de múltiplas origens requer mecanismos analíticos semelhantes às soluções de *Business Inteligence* [Prarthana and Gangadhar 2017, Thudumu et al. 2020].

Em que pese os avanços na detecção de anomalias de comportamento de usuários, em geral os trabalhos propõem modelos que buscam encontrar anomalias com base na totalidade dos dados de conjuntos de usuários, sem distinção do comportamento específico de um ou outro usuário. Assim, é comum que os modelos identifiquem anomalias de comportamento que se distinguem da grande parte dos dados, mas sem uma individualização dos padrões comportamentais. Além disso, utilizar a totalidade dos dados para gerar os modelos, indica que os cenários de estudo fogem ao ambiente real, em que é preciso traçar perfis de comportamento em horizonte temporal e espacial aceitável para tomar decisões sobre as anomalias detectadas.

Diante desse cenário, este trabalho propõe uma arquitetura denominada WEAPON, para detectar anomalias de comportamento de usuários com base em *Wide and Deep Convolutional* LSTM (*Long Short-Term Memory*) Autoencoders. O WEAPON foi modelado combinando o *Wide and Deep* com *Convolutional* LSTM Autoencoders para inserir informações do usuário e seu comportamento, de modo a garantir a detecção de mudanças significativas de padrões de maneira eficiente. Para validar o WEAPON, foi utilizado o dataset do CERT[1] que é composto por *logs* de 4000 usuários com diferentes tipos de comportamentos. Ademais, este trabalho possui como principais contribuições: (i) um modelo de deteção de anomalias de comportamento de usuários que requer menos dados de treinamento em comparação com outras soluções; (ii) a deteção de anomalias de comportamento que considerem tanto características de cada usuário como a característica geral dos dados; (iii) um modelo incremental quando novos conjuntos de dados estiverem disponíveis.

[1]https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099

O restante deste trabalho está organizado da seguinte forma. A Seção 2 apresenta trabalhos que utilizam modelos baseados em padrão de comportamento de usuários. A Seção 3 apresenta como o WEAPON foi modelado. A Seção 4 apresenta o cenário utilizado para validar o WEAPON. A Seção 5 apresenta as conclusões e os trabalhos futuros.

2. Trabalhos Relacionados

Diversas técnicas de aprendizado de máquina vêm sendo empregadas para a detecção de anomalias de comportamento de usuário. Como eventos anômalos são raros e sempre estão relacionados a um tipo especial de evento de desvio, a deteção de anomalias é considerada um problema não supervisionado [Prarthana and Gangadhar 2017, Aggarwal 2017, Thudumu et al. 2020].

Considerando tais características, [Jin et al. 2017] propõem o uso da *Robust Principal Component Analysis* (RPCA) para identificar vazamento de informações confidenciais a partir da caracterização de padrões normais de consultas a dados por parte de usuários. Já [Pokhrel et al. 2019] propõem uma solução híbrida baseada em *One Class Support Vector Machine* (OCSVM) e Naïve Bayes (NB) para criar padrões de comportamento de usuários a partir de registros do Windows. O objetivo é identificar mau uso de credenciais. A partir de um dataset composto de *logs* de auditoria do Windows, os autores estabelecem uma etapa de definição de perfil e outra de detecção. Como ambos os trabalhos são baseados em *shallow learning* (i.e., RPCA, OCSVM e NB), existe a necessidade de elaborar os dados sempre que seja necessário retreinar o modelo, diferentemente de modelos baseados em *deep learning*.

[Gao et al. 2017] apresentam o uso de clusterização para identificar padrões de usuários maliciosos usando *logs* de servidores web. A partir das transformações aplicadas sobre os *logs*, são criadas sequências descritivas dos comportamentos de cada usuário e consolidadas em matrizes contendo os *eigenvalues* das sequências. Entretanto, tal abordagem apresenta como desvantagem a necessidade de um novo treinamento para cada nova entrada de *log*, uma vez que a detecção baseia-se em algoritmo de clusterização. Outra desvantagem é a impossibilidade de treino incremental do modelo.

[Prarthana and Gangadhar 2017] abordam o aspecto multidimensional dos *logs* de segurança com base no *Online Analytical Processing* (OLAP). Para tanto, é necessário utilizar algoritmos de clusterização para análise multidimensional das *features*. Entretanto, o trabalho possui como limitação um alto custo computacional por utilizar testes estatísticos que dependem da modelagem OLAP. Os próprios autores ressaltam que o acréscimo de dimensões e execuções aumenta o tempo exponencialmente. Assim, a limitação da quantidade de dimensões analisadas dificulta avaliação de padrões comportamentais que envolvam um número alto de dimensões.

[Qu et al. 2018] endereçam o uso de autoencoder com redes neurais recorrentes (RNN) usando células do tipo *Gated Recurrent Neural Network* (GRU). O objetivo é propor um modelo que possa garatir um treinamento eficiente para identificar anomalias relacionadas a comportamento de usuários. Essa abordagem apresenta a desvantagem de não identificar unicamente os usuários que apresentam comportamento anômalo, detectando apenas dados que se distanciam significativamente da totalidade. Além disso, ainda que a abordagem tenha obtido resultados satisfatórios, a métrica utilizada é inadequada para indicar efetividade nos modelos para detecção de anomalias, pois desconsidera que

os dados são desbalanceados. Nesses casos, as métricas Recall, ROC-AUC e F1-Score podem melhor indicar a qualidade do modelo obtido.

[Kim et al. 2019] propõem a detecção de *insiders* aplicando a combinação de estimativa de densidade Gaussiana, densidade de janelas Parzen, *Principal Component Analysis* – PCA, e clusterização. Tal detecção é realizada sobre um conjunto de dados pré-processados contendo resumos diários das ações dos usuários. Uma limitação do trabalho é que a escolha das *features* foi realizada a partir da verificação de sua importância na detecção das anomalias do conjunto de teste. Além disso, o tamanho do conjunto de dados de treinamento utilizado pode não representar um cenário real. Por fim, a avaliação deixa de verificar se a grande quantidade de *features* inseridas não trouxe consigo o problema da dimensionalidade (*curse of dimensionality*) [Thudumu et al. 2020].

A partir dos trabalhos supracitados, observou-se que há espaço para melhorias e, por isso, esta pesquisa investiga as seguintes lacunas: (i) adoção de um modelo de detecção de anomalias de comportamento baseado em *deep learning* que permita treinamento incremental, na medida em que novos dados são disponibilizados; (ii) possibilidade de treinamento utilizando um pequeno conjunto de dados para gerar os padrões de comportamentos dos usuários, o que se aproxima mais de um cenário real de aplicação; e (iii) caracterização dos usuários durante o treinamento, de modo a permitir a detecção de variações comportamentais. Em razão disso, a seguir será apresentada a nossa solução.

3. WEAPON

Esta seção tem como objetivo apresentar o WEAPON, uma arquitetura para detecção de anomalias em comportamentos de usuários com base em um *Wide and Deep Convolutional LSTM Autoencoder*, a partir de características individuais e coletivas dos dados dos usuários. A seguir, será apresentada uma visão geral do funcionamento do WEAPON, a descrição do dataset utilizado, o pré-processamento para a extração das informações, e por fim uma análise dos padrões de comportamento anômalos.

3.1. Visão geral

A Figura 1 apresenta o funcionamento do WEAPON. Para tanto, o WEAPON foi modelado combinando os conceitos de *autoenconder* convolucional, LSTM *autoencoder* e *wide and deep learning*. Para iniciar o processo de deteção de anomalias, o *encoder* do WEAPON recebe como entrada os dados *deep* (Rótulo A, Figura 1), que são as *features* comportamenais dos usuários. A estrutura interna do *encoder* combina uma camada convolucional temporal (unidimensional) com camadas de redes neurais recorrentes LSTM. A camada convolucional compreende um total de 156 kernels (13 *features* x 12 filtros), retornando 12 filtros, cada um contendo a média de seus 13 kernels. As duas camadas LSTM permitem que a representação latente das *features* comportamentais seja reduzida para 8 dimensões (Rótulo B, Figura 1), mantendo uma memória de curto e longo prazo devido à construção desse tipo de camada, que utiliza portas *input gate* e *forget gate*.

Já o *decoder* recebe os dados de representação latente provenientes do *encoder* e os combinam com a entrada *wide* (Rótulo C, Figura 1), que é composta das *features* categóricas. A estrutura do *decoder* é composta de 2 camadas LSTM reversas ao *encoder*, concatenando em seguida os dados provenientes da camada LSTM com as *features* categóricas. Ainda no *decoder*, os dados concatenados são submetidos a uma camada com

ativação *Exponencial Linear Unit* (ELU), seguida de uma camada de *Batch Normalization* e outra de regularização *Dropout* com taxa de desativação de 20%, para então reconstruir as *features* da entrada *deep* com uma camada de 13 neurônios (Rótulo D, Figura 1). A camada de *Dropout* evita que o modelo apresente comportamento de *overfitting*.

A saída do *decoder* do WEAPON é comparada com as *features* comportamentais para a avaliação do erro de reconstrução segundo a métrica de erro médio absoluto (Rótulo E, Figura 1). Aplicando-se um *threshold*, é possível então realizar a detecção dos comportamentos anômalos.

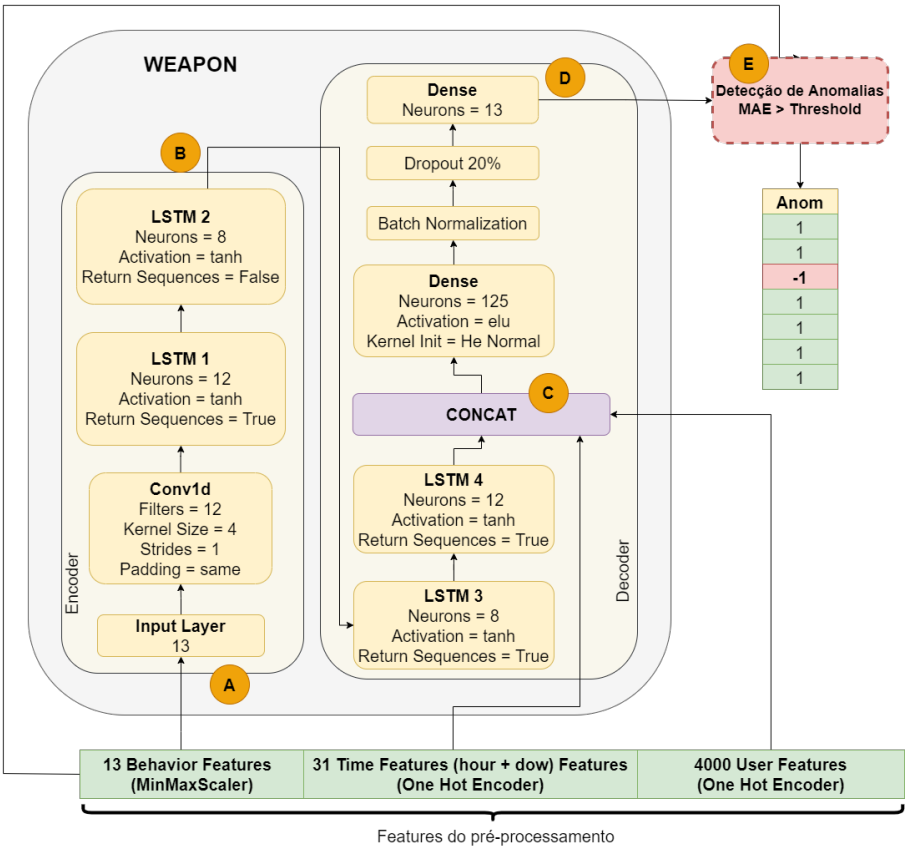


Figura 1. Arquitetura do WEAPON

3.2. CERT Dataset

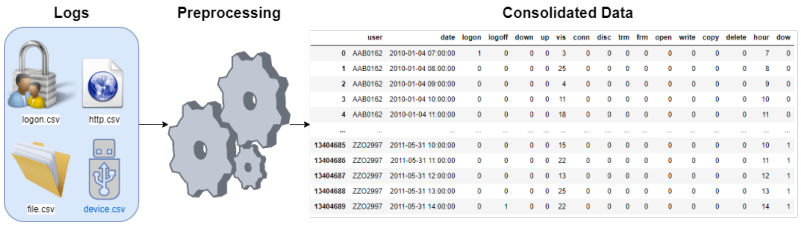
Uma anomalia de comportamento de usuário é caracterizada quando é possível estabelecer uma ou mais ações incomuns com relação a um padrão de comportamento pré-estabelecido. Para definir esse padrão de referência, assim como elaborar os modelos de detecção de anomalias, existe um desafio importante que é a obtenção de *logs* de segurança reais contendo as atividades dos usuários [Ratner et al. 2020, Kim et al. 2019], haja vista os obstáculos quanto à proteção, privacidade e dificuldade de rotular os dados.

Por essa razão, neste trabalho foi utilizado o CERT Dataset [Lindauer 2020, Glasser and Lindauer 2013], o qual é composto por um conjunto de *datasets* gerado artificialmente, tendo como cenário uma organização com 4000 usuários cujas atividades diárias são registradas em *logs*. As atividades registradas são *logon* (logon.csv), uso de dispositivo removível (device.csv e file.csv), navegação web (http.csv) e uso de correio

eletrônico (email.csv). Há ainda dados cadastrais e de papéis dos usuários (ldap.csv). Foi utilizada a versão R6.2 do dataset, que consiste em uma coleção de dados sintéticos dos usuários, contendo atividades consideradas normais (*Background*), bem como atividades de cinco atores maliciosos (i.e., *Insiders*), dos quais parte apresenta modificação comportamental anômala.

3.3. Pré-processamento dos dados e extração de *features*

A Figura 2 apresenta a visão geral do pré-processamento para a construção do *dataset* final. Os *logs* contidos nos arquivos logon.csv, device.csv, file.csv e http.csv foram utilizados para modelar o perfil dos usuários, uma vez que buscamos alterações de comportamento relacionadas ao logon, navegação *web* e uso de dispositivo removível.



	user	date	logon	logoff	down	up	vis	conn	disc	trm	frm	open	write	copy	delete	hour	dow
0	AAB0162	2010-01-04 07:00:00	1	0	0	0	3	0	0	0	0	0	0	0	0	7	0
1	AAB0162	2010-01-04 08:00:00	0	0	0	0	25	0	0	0	0	0	0	0	0	8	0
2	AAB0162	2010-01-04 09:00:00	0	0	0	0	4	0	0	0	0	0	0	0	0	9	0
3	AAB0162	2010-01-04 10:00:00	0	0	0	0	11	0	0	0	0	0	0	0	0	10	0
4	AAB0162	2010-01-04 11:00:00	0	0	0	0	18	0	0	0	0	0	0	0	0	11	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
13404685	ZZO2997	2011-05-31 10:00:00	0	0	0	0	15	0	0	0	0	0	0	0	0	10	1
13404686	ZZO2997	2011-05-31 11:00:00	0	0	0	0	22	0	0	0	0	0	0	0	0	11	1
13404687	ZZO2997	2011-05-31 12:00:00	0	0	0	0	13	0	0	0	0	0	0	0	0	12	1
13404688	ZZO2997	2011-05-31 13:00:00	0	0	0	0	25	0	0	0	0	0	0	0	0	13	1
13404689	ZZO2997	2011-05-31 14:00:00	0	1	0	0	22	0	0	0	0	0	0	0	0	14	1

Figura 2. Pré-processamento dos dados no WEAPON

A Tabela 1 apresenta os dados que foram consolidados em resumos das atividades de cada usuário a cada hora, o que possibilita mensurar a quantidade de ações do usuário nos intervalos de uma hora, e assim, traçar um perfil vinculado à hora e dia da semana. Em seguida, os dados foram combinados para gerar uma única tabela que foi utilizada para modelar o perfil de comportamento dos usuários. Após esse processo, há a etapa de normalização dos dados e codificação das *features* categóricas (*hour, down, user*). Para as *features* comportamentais (*logon, logoff, down, up, vis, conn, disc, trm, frm, open, write, copy, delete*) utilizamos o MinMaxScaler [2], que faz uma transformação de escalonamento entre zero e um, e para as *features* categóricas (*hour, dow e user*), usamos o OneHotEncoder[3], que realiza a conversão em variáveis binárias no formato de matriz esparsa.

A construção do padrão comportamental a ser considerado normal teve como premissas a necessidade de selecionar um conjunto de dados com um horizonte temporal factível em cenário real, bem como o fato de que, por princípio, não há anomalias de comportamento nesse período de referência. Por isso, foram selecionados os primeiros quatros meses de dados da etapa de pré-processamento para treinar o modelo. Já no que se refere a testar a capacidade de detecção de anomalias, os quatro meses seguintes foram selecionados por conterem eventos anômalos, permitindo a validação do WEAPON.

3.4. Padrões comportamentais anômalos

Esta seção apresenta os padrões comportamentais que influenciam no processo de detecção de anomalias identificados pelo WEAPON. Salienta-se que em um cenário real, avaliar a efetividade de um modelo de deteção de anomalias é uma tarefa desafiadora, uma vez que anomalias são raras por definição, e não se dispõe de dados rotulados [Zhang et al. 2021]. No entanto, nesta pesquisa as modificações de comportamento

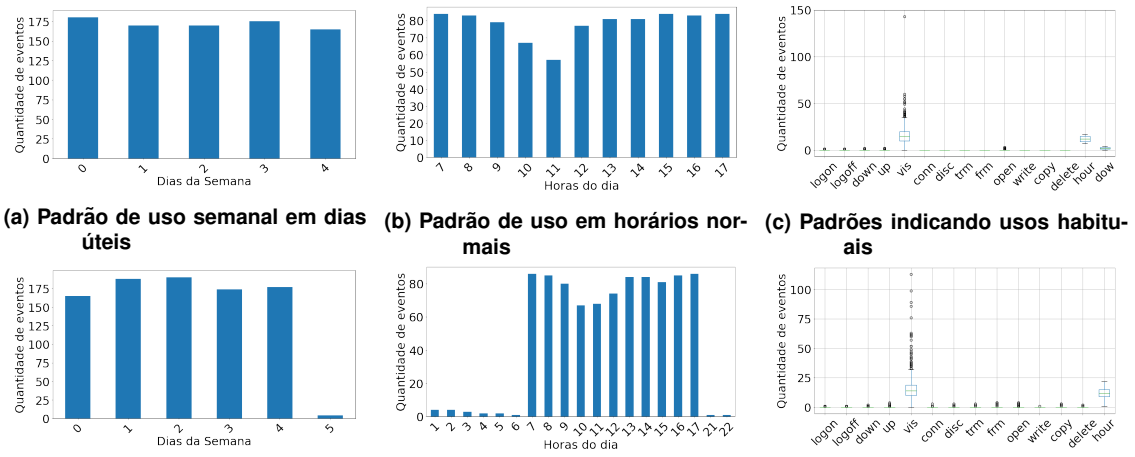
[2]https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

[3]https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html, https://www.tensorflow.org/api_docs/python/tf/one_hot

Tabela 1. *Features* extraídas do pré-processamento.

Feature	Descrição
user	Código do usuário
date	Data e hora da consolidação
logon	Quantidade de logons no computador durante a hora
logoff	Quantidade de logoffs no computador durante a hora
down	Quantidade de downloads durante a hora
up	Quantidade de uploads durante a hora
vis	Quantidade de visitas durante a hora
conn	Quantidade de conexões de dispositivo removível durante a hora
disc	Quantidade de desconexões de dispositivo removível durante a hora
trm	Quantidade de ações executadas com destino no dispositivo removível durante a hora
frm	Quantidade de ações executadas com origem no dispositivo removível durante a hora
open	Quantidade de abertura de arquivos no dispositivo removível durante a hora
write	Quantidade de escritas de arquivo no dispositivo removível durante a hora
copy	Quantidade de cópias de arquivos no dispositivo removível durante a hora
delete	Quantidade de deleções de arquivos no dispositivo removível durante a hora
hour	Hora, variando de 0 a 23
dow	Dia da Semana, variando de 0 (segunda) a 6 (domingo)

significativas são um indicativo das anomalias a serem detectadas. Por exemplo, a Figura 3 apresenta um caso de usuário com hábito de utilizar o ambiente somente nos dias úteis (Figura 3a), entre 7:00 e 17:00 (Figura 3b). Em outras palavras, não tinha hábitos de usar dispositivo removível ou trabalhar em horários após o expediente, como ratificado na Figura 3c, indicando um comportamento não anômalo. Em contraposição, o comportamento de usar o dispositivo removível com maior frequência, fazer *uploads* de arquivos sensíveis fora do expediente e em finais de semana, respectivamente Figuras 3d, 3e 3f, pode ser considerado um padrão inesperado. Portanto, a detecção deste tipo de mudança de comportamento, sem entrar no mérito sobre se a ação em si é maliciosa ou não, gera um processo anômalo que é identicado pelo WEAPON.



(a) Padrão de uso semanal em dias úteis

(b) Padrão de uso em horários normais

(c) Padrões indicando usos habituais

(d) Padrão indicando uso em fins de semana

(e) Padrão indicando uso fora do expediente

(f) Padrões indicando modificação de uso

Figura 3. Padrões comportamentais anômalos do usuário ACM2278.

4. Avaliação de Desempenho

4.1. Configuração dos Experimentos

Para avaliar o WEAPON, as seguintes métricas foram utilizadas: (i) Erro Médio Absoluto, quantifica o erro de reconstrução os dados reconstruídos; (ii) Acurácia, quantifica a taxa de acertos do modelo; (iii) Recall, quantifica a taxa de anomalias reais que foram reportadas; (iv) F1-Score, quantifica a sensibilidade do modelo para detectar anomalias; e (v) Curva ROC (*Receiver Operating Characteristic*), quantifica a relação entre a taxa de verdadeiros positivos e a taxa de falsos positivos. O ambiente de execução para gerar os experimentos é composto de um computador com processador i7-11700, 32GB de memória RAM, placa de vídeo NVIDIA RTX-3080, 1TB SSD, plataforma Anaconda para implementação dos modelos utilizando o Python versão 3.9.7 e Jupyter Notebook.

O WEAPON foi comparado com dois modelos desenvolvidos para detecção de anomalias de comportamento: (i) *Wide and Deep Stacked Autoencoder* com *Batch Normalization* (WDSAEBN); e (ii) *Stacked Autoencoder* com *Batch Normalization* (SAEBN). Para encontrar a melhor configuração dos modelos, foi utilizado a biblioteca *hyperopt*[4] que dado um conjunto de parâmetros, realiza uma combinação e busca dos melhores hiperparâmetros para os modelos. Para tratar os dados, foram utilizadas as bibliotecas Pandas[5], Scikit Learn[6], NumPy[7] e TensorFlow Data API[8]. Já para a construção dos modelos, foram utilizadas as bibliotecas TensorFlow[9] e Keras[10]. A Tabela 2 apresenta os parâmetros utilizados na configuração dos experimentos.

A detecção de anomalias ocorre a partir da definição de um *threshold* de 2% considerado como taxa de contaminação. Isto é, a taxa de anomalias que se espera encontrar em relação ao total dos dados. Essa taxa é largamente utilizada em algoritmos de detecção de anomalias [Zhao et al. 2019]. A seguir serão apresentados os resultados obtidos.

4.2. Resultados e Discussão

As métricas de desempenho avaliadas para as três arquiteturas são apresentadas na Tabela 3. Verificamos que os três modelos tem desempenho semelhante quanto à métrica de acurácia, obtendo aproximadamente 99% de acertos. Entretanto, é válido salientar que em datasets desbalanceados, a métrica de acurácia não é adequada, pois desconsidera que a classe de anomalias é muito inferior à classe de eventos normais. Assim, ao analisarmos a taxa de Recall, notamos que o WEAPON possui desempenho próximo ao do modelo SAEBN, ambos atingindo 72,22%, enquanto o modelo WDSAEBN alcançou 61,11%. O F1-Score do modelo SAEBN atingiu o maior valor dos testes, chegando a 83,87%, enquanto o WEAPON alcançou o segundo melhor desempenho, chegando a 78,79%, seguido do WDSAEBN, com 75,86%. O F1-Score favorece classificadores que têm valores de Precisão e Recall similares, o que não é desejável para detecção de anomalias, em que

[4]http://hyperopt.github.io/hyperopt/
[5]https://pandas.pydata.org/docs/index.html
[6]https://scikit-learn.org/stable/index.html
[7]https://numpy.org/
[8]https://www.tensorflow.org/api_docs/python/tf/data
[9]https://www.tensorflow.org/
[10]https://keras.io/

Tabela 2. Parâmetros utilizados na configuração dos experimentos.

	SAEBN	**WEAPON**	**WDSAEBN**
Encoder	Input: [4044] Batch Normalization Dense(250) Batch Normalization Dense(100) Batch Normalization Dense(20) Batch Normalization	Input: deep[13] Conv1D(f:12,ks:4) LSTM(12) LSTM(8)	Input: deep[13] Dense(10) Batch Normalization Dense(5)
Decoder	Dense(100) Batch Normalization Dense(250) Batch Normalization Dense*(13)	LSTM(8) LSTM(12) Concat:wide[4031],[12] Dense(125) Batch Normalization Dropout=20% Dense*(13)	Dense(10) Concat:wide[4031],[10] Dense(100) Dense*(13)
Ativação	Dense: ELU Dense*: Sem ativação	Conv1D: SELU LSTM: TANH Dense: ELU Dense*: Sem ativação	Dense: ELU Dense*: Sem ativação
Inicialização	Dense: He Normal	Dense: He Normal	Dense: He Normal
Treino	10 épocas Loss: MAE Optimizer: Nadam Metrics: MSE	10 épocas Loss: Huber Optimizer: Nadam Metrics: MAE	10 épocas Loss: MAE Optimizer: Nadam Metrics: MSE

a taxa de Recall é mais importante que a Precisão. Por isso, a análise da métrica ROC-AUC indica que o melhor modelo é o WEAPON, alcançando 99,15%, um desempenho aproximadamente 7% melhor que o SAEBN e 2,5% melhor que o WDSAEBN.

Tabela 3. Métricas de desempenho obtidas para avaliar os modelos.

Model	**Accuracy**	**Recall**	**F1-Score**	**ROC-AUC**
SAEBN	99,44%	72,22%	83,87%	92,35%
WEAPON	99,22%	**72,22%**	78,79%	**99,15%**
WDSAEBN	99,22%	61,11%	75,86%	96,76%

Na Figura 4, é apresentada a curva ROC realizando uma análise do WEAPON, comparando-o com o SAEBN e WDSAEBN. O objetivo é apresentar o comportamento do WEAPON para a detecção de anomalias com diferentes valores de *threshold*, observando qual a variação da sensibilidade e especificidade. Por meio dos resultados, observou-se que o WEAPON é o que possui melhor desempenho. Quando comparado com o WDSAEBN e SAEBN, o WEAPON possuir a curva com maior área, comprovando que para diferentes valores de *threshold*, é o modelo cuja curva mais se aproxima da extremidade superior esquerda, possuindo maior área. Em outras palavras, quanto maior forem os valores de verdadeiro positivo e menor de falso positivo. Isso ocorre porque o WEAPON foi modelado utilizando uma camada convolucional unidimensional (i.e., convolução temporal) associada às redes recorrentes LSTM, o que assegurou que o modelo aprendeu a interpretar melhor os padrões dos usuários, sendo capaz de distinguir as anomalias de comportamento e apresentando uma resposta melhor em diferentes níveis de *threshold*. Além disso, a utilização da camada *Dropout* em 20% assegura que o mo-

delo não apresente *overfitting* na reconstrução dos dados. Por fim, o reduzido número de camadas intermediárias reforça a interpretação das características latentes dos dados de comportamento.

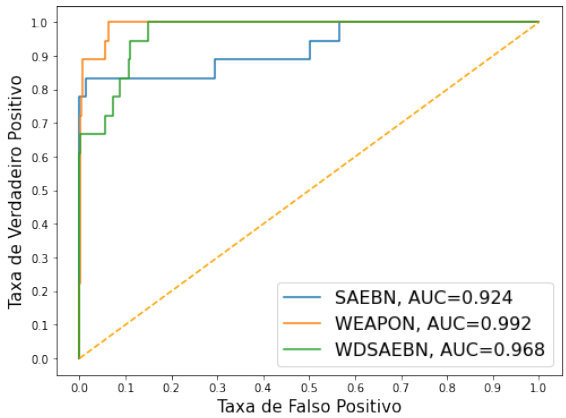
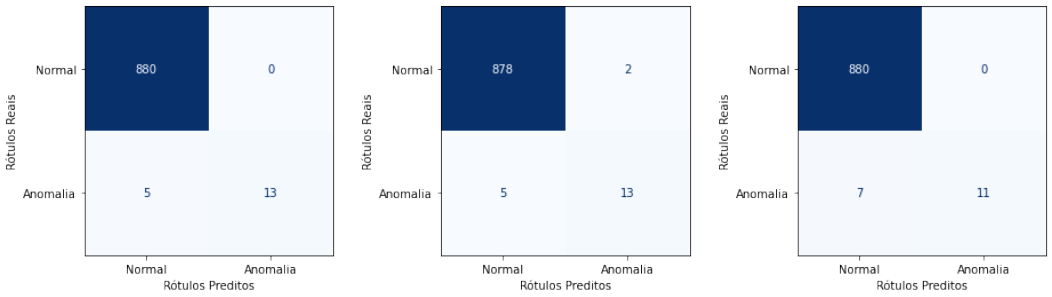


Figura 4. Curvas ROC para avaliar os modelos.

Na Figura 5 são apresentadas as matrizes de confusão obtidas para cada modelo. A partir das matrizes é possível verificar que os modelos SAEBN (Figura 5a) e WEAPON (Figura 5b) se sobressaem quanto à identificação de comportamentos anômalos, conseguindo identificar maior quantidade de anomalias do conjunto, 13 de um total de 18 (72,22%). O SAEBN se sobressaiu quanto aos falsos positivos, pois enquanto o WEAPON obteve 2 falsos positivos, o SAEBN não obteve nenhum. Já a análise de falsos negativos, tanto o WEAPON quanto o SAEBN deixaram de identificar 5 (27,78%) anomalias, enquanto o pior modelo, WDSAEBN, não identificou 7 (38,89%) anomalias. Portanto, com tais resultados, é possível constatar a eficiência do WEAPON para detecção de anomalias em comportamentos dos usuários.



(a) SAEBN **(b) WEAPON** **(c) WDSAEBN**

Figura 5. Matrizes de confusão para os modelos avaliados.

5. Conclusão

A detecção de anomalias de comportamento de usuários tem papel importante no atual cenário mundial de crescentes ataques cibernéticos. Identificar alterações comportamentais pode significar a mitigação de incidentes catastróficos para as organizações. Com os avanços decorrentes tanto da capacidade computacional como também da ciência de dados, torna-se cada vez mais possível antecipar-se aos ataques cibernéticos com soluções aptas a tratar a grande quantidade de dados complexos vinculados a esses ataques.

Diante do exposto, este trabalho propôs o WEAPON, uma arquitetura para detecção de anomalias em comportamentos de usuários com base em um *Wide and Deep Convolutional LSTM Autoencoder*. Com o WEAPON foi possível compreender as características latentes dos dados comportamentais e ainda distinguir as características comportamentais individuais. Os resultados obtidos ratificam tais afirmações e mostram que o WEAPON atingiu um Recall de 72,22% e curva ROC de 99,25%, mostrando-se mais eficiente quando comparado com os modelos da literatura. Ainda, o WEAPON permite a geração de um modelo incremental quando novos conjuntos de dados estiverem disponíveis.

Como trabalhos futuros, planeja-se desenvolver um mecanismo para atribuir pesos para as *features*, de modo a dar maior ou menor importância a cada uma, para o processo de deteção de anomalias. Ainda, temos a intenção de avaliar a utilização de agregação de usuários que compartilham perfis de comportamento similares para aprimorar o processo de detecção de anomalias.

Agradecimentos

Os autores agradecem o apoio do TED ABIN 08/2019. R.T.S.J. agradece o apoio do CNPq outorgas 465741/2014-2 e 312180/2019-5, da Advocacia Geral da União outorga 697.935/2019, do Departamento Nacional de Auditoria do SUS outorga 23106.118410/2020-85, da Procuradoria Geral da Fazenda Nacional outorga 23106.148934/2019-67, e do Conselho Administrativo de Defesa Econômica outorga 08700.000047/2019-14.

Referências

Abu Sulayman, I. I. and Ouda, A. (2019). User Modeling via Anomaly Detection Techniques for User Authentication. *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON 2019*, pages 169–176.

Aggarwal, C. C. (2017). *Outlier Analysis*. Springer Publishing Company, Incorporated, 2nd edition.

Agrawal, S. and Agrawal, J. (2015). Survey on anomaly detection using data mining techniques. *Procedia Computer Science*, 60(1):708–713.

Ahmed, M., Naser Mahmood, A., and Hu, J. (2016). A survey of network anomaly detection techniques. *J. Netw. Comput. Appl.*, 60(C):19–31.

Filho, G. P., Ueyama, J., Villas, L. A., Pinto, A. R., Goncalves, V. P., Pessin, G., Pazzi, R. W., and Braun, T. (2014). Nodepm: a remote monitoring alert system for energy consumption using probabilistic techniques. *Sensors*, 14(1):848–867.

Gao, Y., Ma, Y., and Li, D. (2017). Anomaly detection of malicious users' behaviors for web applications based on web logs. In *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, pages 1352–1355.

Geraldo Filho, P., Ueyama, J., Villas, L., Pinto, A., and Seraphini, S. (2013). Nodepm: Um sistema de monitoramento remoto do consumo de energia elétrica via redes de sensores sem fio. *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), editor, Sociedade Brasileira de Computação (SBC)*, 31:17–30.

Glasser, J. and Lindauer, B. (2013). Bridging the gap: A pragmatic approach to generating insider threat data. In *2013 IEEE Security and Privacy Workshops*, pages 98–104.

Gupta, M., Gao, J., Aggarwal, C. C., and Han, J. (2014). Outlier Detection for Temporal Data: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 26(9):2250–2267.

Jin, Y., Qiu, C., Sun, L., Peng, X., and Zhou, J. (2017). Anomaly detection in time series via robust PCA. *2017 2nd IEEE International Conference on Intelligent Transportation Engineering, ICITE 2017*, pages 352–355.

Junior, G., Carvalho, L. C., Rodrigues, J. R., and Proença, M. P. (2016). Network anomaly detection using IP flows with principal component analysis and ant colony optimization. *Journal of Network and Computer Applications*, 64(1):1–11.

Kim, J., Park, M., Kim, H., Cho, S., and Kang, P. (2019). Insider threat detection based on user behavior modeling and anomaly detection algorithms. *Applied Sciences (Switzerland)*, 9(19).

Kwon, D., Kim, H., Kim, J., Suh, S. C., Kim, I., and Kim, K. J. (2017). A survey of deep learning-based network anomaly detection. *Cluster Computing*, 22:949–961.

Lindauer, B. (2020). Insider threat test dataset. https://kilthub.cmu.edu/articles/dataset/Insider_Threat_Test_Dataset/12841247/1.

Pang, G., Shen, C., Cao, L., and Hengel, A. V. D. (2021). Deep learning for anomaly detection: A review. *ACM Comput. Surv.*, 54(2).

Patcha, A. and Park, J. M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 51(12):3448–3470.

Pokhrel, R., Pokharel, P., and Kumar Timalsina, A. (2019). Anomaly-Based – Intrusion Detection System using User Profile Generated from System Logs. *International Journal of Scientific and Research Publications (IJSRP)*, 9(2):p8631.

Prarthana, T. S. and Gangadhar, N. D. (2017). User behaviour anomaly detection in multidimensional data. In *2017 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, pages 3–10.

Prasad, N. R., Almanza-Garcia, S., and Lu, T. T. (2009). Anomaly detection. *Computers, Materials and Continua*, 14(1):1–22.

Qu, Z., Su, L., Wang, X., Zheng, S., Song, X., and Song, X. (2018). A Unsupervised Learning Method of Anomaly Detection Using GRU. *Proceedings - 2018 IEEE International Conference on Big Data and Smart Computing, BigComp 2018*, pages 685–688.

Ratner, A., Bach, S., Ehrenberg, H., Fries, J., Wu, S., and Ré, C. (2020). Snorkel: rapid training data creation with weak supervision. *The VLDB Journal*, 29.

Sadik, M. S. and Gruenwald, L. (2014). Research issues in outlier detection for data streams. *ACM SIGKDD Explorations Newsletter*, 15:33–40.

Shin, K., Hooi, B., Kim, J., and Faloutsos, C. (2021). Detecting group anomalies in tera-scale multi-aspect data via dense-subtensor mining. *Frontiers in Big Data*, 3.

Thomé, M., Prestes, A., Gomes, R., and Mota, V. (2020). Um arcabouço para detecção e alerta de anomalias de mobilidade urbana em tempo real. In *Anais do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 784–797, Porto Alegre, RS, Brasil. SBC.

Thudumu, S., Branch, P., Jin, J., and Singh, J. J. (2020). A comprehensive survey of anomaly detection techniques for high dimensional big data. *Journal of Big Data*, 7(1):1–30.

Vilaça, E. S. C., Vieira, T. P. B., de Sousa, R. T., and da Costa, J. P. C. L. (2019). Botnet traffic detection using RPCA and mahalanobis distance. In *2019 Workshop on Communication Networks and Power Systems (WCNPS)*, pages 1–6.

Zhang, C., Wang, S., Zhan, D., Yu, T., Wang, T., and Yin, M. (2021). Detecting Insider Threat from Behavioral Logs Based on Ensemble and Self-Supervised Learning. *Security and Communication Networks*, 2021.

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