Detecção de Ataques a Redes IoT Usando Técnicas de Aprendizado de Máquina e Aprendizado Profundo

Kaylani Bochie, Ernesto R. Gonzalez, Luiz F. Giserman, Miguel Elias M. Campista e Luís Henrique M. K. Costa *

[1]Grupo de Teleinformática e Automação (GTA)
PEE/COPPE-DEL/Poli
Universidade Federal do Rio de Janeiro (UFRJ)

{kaylani,ernesto,giserman,miguel,luish}@gta.ufrj.br

Resumo. *As vulnerabilidades de dispositivos IoT os tornam um alvo simples para invasão e controle por parte de atacantes. Ao mesmo tempo, a dinamicidade das redes IoT dificulta o desenvolvimento de sistemas de segurança baseados em regras. Este cenário é um convite ao emprego de técnicas de aprendizado de máquina. No entanto, a escassez de conjuntos de dados públicos torna-se um entrave para a avaliação da detecção de ataques a redes IoT. Ainda, observa-se que os desempenhos de modelos de aprendizado não são comparados quantitativamente, o que pode afetar a validade das conclusões. Este trabalho, então, avalia os desempenhos de múltiplos modelos de aprendizado de máquina tradicionais e profundos, em traces públicos, para a detecção de ataques. Modelos como redes neurais convolucionais, recorrentes e autoassociativas são usados. A comparação mostra que traces organizados por fluxo ou por pacote têm influência direta na escolha de técnicas para detecção. Além disso, redes neurais autoassociativas profundas se mostram efetivas para detectar ataques online.*

Abstract. *The vulnerabilities of IoT devices make them a simple target for intrusion and control by hackers. At the same time, the dynamics of IoT networks make it hard to develop rule-based security systems. This scenario is an invitation to the use of machine learning techniques. Nevertheless, the lack of public datasets becomes an obstacle for assessing attack detection on IoT networks. Also, it is observed that the performance of learning models are not quantitatively compared, which can affect the validity of conclusions. This paper, therefore, evaluates the performance of multiple traditional and deep machine learning models, based on public traces, for attack detection. Models such as convolutional, recurrent, and autoencoder neural networks are used. The comparison shows that traces grouped by flow or by packet have a direct impact on the choice of detection techniques. Also, deep autoencoders are shown to be effective in online attack detection.*

1. Introdução

Desde sua concepção, a Internet das Coisas (*Internet of Things* – IoT) apresenta diferentes características em relação aos paradigmas tradicionais de redes, como a presença

*O presente trabalho foi realizado com apoio do CNPq; da FAPERJ; da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES), Código de Financiamento 001; e da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), processos n^o 15/24494-8 e 15/24490-2.

de dispositivos com menor poder de processamento, capacidade de sensoriamento e largura de banda limitada. Essas particularidades são reflexos dos tipos de componentes utilizados e serviços que se deseja oferecer pelas redes IoT. Com o crescimento acelerado do número de dispositivos conectados à Internet e o aumento de conexões Máquina-a-Máquina (*Machine-to-Machine* – M2M), a preocupação com a segurança dessas redes vem aumentando proporcionalmente, graças ao atual poder de penetração das redes IoT e o seu consequente emprego em aplicações diversas, inclusive em aplicações críticas, como as de monitoramento inteligente de pacientes hospitalizados [Goasduff, 2019]. Outro fator relevante é o aumento na heterogeneidade de dispositivos e de tecnologias de rede utilizadas, como IEEE 802.11 (Wi-Fi), IEEE 802.15.4, Bluetooth, etc., o que, aliado à popularidade das redes IoT, introduzem novas brechas de segurança. Tais brechas podem tornar as redes IoT suscetíveis a ataques em proporções diferentes, ou inclusive a novos métodos de intrusão [Hassija et al., 2019].

Diversas abordagens para detecção de ataques a redes de computadores são encontradas na literatura, como o uso de técnicas tradicionais baseadas em regras [Guan e Ge, 2018] e técnicas de Aprendizado de Máquina (*Machine Learning*) [Buczak e Guven, 2016]. Esforços também são feitos a fim de gerar conjuntos de dados representativos de ataques direcionados a redes IoT. No entanto, apesar desses esforços e do relativo interesse na disponibilidade de conjuntos de dados, muitos ainda não se encontram acessíveis. Isto gera uma dificuldade na obtenção de comparações de desempenho de diferentes métodos aplicados aos mesmos conjuntos de dados, o que, por sua vez, também dificulta a criação de Sistemas de Detecção de Intrusão (*Intrusion Detection Systems* – IDSs) capazes de detectar ataques em diferentes redes IoT.

Este trabalho compara o desempenho de modelos de aprendizado tradicionais e profundos, tendo em vista as características inerentes de redes IoT. A metodologia utilizada é dividida em etapas a fim de destacar decisões importantes a serem tomadas especialmente durante o pré-processamento de dados, visto que o grande volume de dados gerado por redes IoT pode levar à especificidade de modelos tradicionais. Dentre as técnicas utilizadas na avaliação, o uso de redes neurais recorrentes para análise de sequência de pacotes e também de redes autoassociativas capazes de aprender de forma não supervisionada, constituem, neste trabalho, uma abordagem diferenciada. Os resultados são obtidos a partir de experimentos usando dois conjuntos de dados: BoT-IoT e outro chamado neste trabalho como Bezerra2018. Enquanto o BoT-IoT lista todos os pacotes rotulados recebidos em uma *botnet*, por isso chamado de *trace* orientado a pacotes, o Bezerra2018 lista todos os fluxos rotulados, por isso chamado de *trace* orientado a fluxo. Observa-se que dentre todas as técnicas avaliadas, árvores de decisão e Redes Neurais Autoassociativas (*Autoencoders*) apresentaram os melhores resultados com relação à capacidade de generalização e robustez da análise. Este resultado é promissor, visto que a operação real em redes de computadores tende a conter tráfego desbalanceado.

O restante deste trabalho está organizado da seguinte forma: a Seção 2 apresenta uma revisão dos trabalhos relacionados aplicados à geração de dados para segurança em redes, identificação de ataques a redes IoT e avaliação de desempenho de novas técnicas de aprendizado de máquina. A Seção 3 apresenta os conjuntos de dados escolhidos para avaliação dos modelos de aprendizado. Já a Seção 4 apresenta a metodologia adotada, incluindo as escolhas de projeto para a execução dos testes de desempenho. A Seção 5

apresenta os resultados experimentais. Por fim, a Seção 6 conclui este trabalho e aponta direções futuras de pesquisa.

2. Trabalhos Relacionados

Na literatura, os trabalhos que abordam a aplicação de aprendizado de máquina para detecção de ataques em redes IoT costumam realizar análises em cenários particulares, dadas as limitações na variedade dos conjuntos de dados disponíveis, não comparando a efetividade dos modelos entre diferentes conjuntos de dados. Hassija et al. descrevem múltiplas abordagens existentes para aumento da segurança de sistemas IoT em diferentes cenários, como redes elétricas inteligentes e sistemas de automação residencial [Hassija et al., 2019]. Algumas dessas abordagens incluem o uso de técnicas baseadas em aprendizado de máquina, além de soluções baseadas em *blockchain*, computação em névoa e nas bordas. Os autores revisam como as restrições encontradas em redes IoT, como a interconexão de múltiplos dispositivos computacionalmente limitados em cadeia, levam à necessidade de métodos inovadores para garantir a segurança de sistemas IoT. Hassija et al. ainda expõem tendências em IoT, como o uso de computação em névoa e nas bordas da rede para garantir que a análise dos dados seja feita localmente, mantendo a privacidade dos usuários. AL-Hawawreh et al., por outro lado, apresentam um estudo detalhado sobre técnicas de aprendizado profundo, aplicadas a um cenário particular de Internet das Coisas Industrial (*Industrial Internet of Things* – IIoT) [AL-Hawawreh et al., 2018]. Os autores avaliam o desempenho de modelos de aprendizado profundo sobre dois conjuntos de dados muito utilizados para testes de modelos preditivos, o NSL-KDD e o UNSW-NB15, que possuem como limitação o fato de terem sido coletados a partir de simulações de redes de computadores cabeadas.

As limitações dos conjuntos de dados também ficam evidentes na literatura. Sharafaldin et al. apontam problemas encontrados em conjuntos de dados utilizados até então em redes IoT e redes de computadores cabeadas, como a carência de tráfego real acompanhado dos ataques capturados e a geração de amostras artificiais através de duplicação. Além disso, os autores descrevem em detalhes o tráfego representativo de uma rede em operação real [Sharafaldin et al., 2018]. Almomani et al. utilizam redes neurais artificiais para classificar ataques a partir de um conjunto de dados gerado pelos próprios autores [Almomani et al., 2016]. O conjunto de dados, chamado de WSN-DS, é gerado a partir de uma rede de sensores sem fio que utiliza o protocolo de roteamento LEACH (*Low Energy Aware Cluster Hierarchy*). Apesar dessa contribuição, os dados coletados não são disponibilizados para futura comparação do desempenho com novos modelos preditivos. Bezerra et al. simulam uma rede IoT doméstica e coletam dados de ataques provenientes de *botnets* para a geração de um conjunto de dados [Bezerra et al., 2018]. Os dados gerados são uma combinação de atributos de dispositivo e de tráfego. Apesar da geração do conjunto de dados, os autores não utilizam os dados para avaliar o desempenho de técnicas de detecção de ataques e definem como pesquisa futura o uso do conjunto de dados para a validação de sistemas de detecção de intrusão em redes IoT. Bezerra et al. avaliam em outro trabalho o desempenho de quatro classificadores para realizar classificação binária sobre o conjunto de dados gerado [Bezerra et al., 2019]. No entanto, os autores não avaliam o desempenho de modelos de aprendizado profundo. Koroniotis et al. criam um conjunto de dados para detecção de ataques em redes IoT chamado BoT-IoT, e avaliam estatisticamente os atributos propostos para o conjunto através do uso do

coeficiente de correlação de Pearson a fim de produzir o melhor subconjunto de atributos para análise [Koroniotis et al., 2019]. Além disso, os autores também avaliam métodos de análise de redes baseados em aprendizado profundo sobre o conjunto proposto completo e sobre subconjuntos de atributos. Os conjuntos de dados gerados por Bezerra et al. e Koroniotis et al. são utilizados neste artigo e suas construções são detalhadas na Seção 3.

A extensão da busca por trabalhos que abordam problemas semelhantes permite encontrar soluções para detecção de falhas em equipamentos em ambientes IIoT [Huang et al., 2020]. Dentre elas, vale destacar o trabalho de Purohit et al. que utilizam redes neurais autoassociativas para identificar falhas em equipamentos industriais no contexto de um sistema IIoT [Purohit et al., 2019]. A natureza desbalanceada em conjuntos de dados de falhas é explorada pela habilidade das redes autoassociativas em reconstruir dados familiares e pela incapacidade de reconstruir dados inéditos. Esta solução é particularmente interessante, pois pode ser adaptada ao contexto de detecção de ataques. Purohit et al. transformam os sinais sonoros em dados visuais através de um espectrograma e, assim como os trabalhos de detecção de ataques listados, geram o seu próprio conjunto de dados. Apesar desse conjunto de dados estar disponível publicamente, ele não serve aos propósitos do atual trabalho.

Diferentemente dos artigos citados, este artigo descreve o processo de tratamento de dados, além de avaliar quantitativamente o desempenho de modelos de aprendizado tradicional e profundo aplicados a dois conjuntos de dados. Este trabalho também analisa as características do formato dos dados, tais como a granularidade das amostras, se as amostras estão ordenadas sequencialmente ou se os atributos medidos foram coletados a nível de pacotes ou de fluxos de rede, que podem levar à utilização de um modelo específico de aprendizado. A análise da literatura deixa evidente a variedade de métodos de análise e a escolha e criação de conjuntos de dados. Dada a grande quantidade de dispositivos diferentes que constituem os sistemas IoT, não há como selecionar um tipo de dispositivo para ser utilizado como "padrão" em simulações e testes. Isso torna necessária a avaliação extensiva de diferentes abordagens de detecção sobre cada cenário para que seja possível comparar o desempenho de forma quantitativa e possibilitar a escolha de técnicas mais apropriadas. O presente trabalho avalia e compara modelos tradicionais e profundos de aprendizado de máquina sobre dois conjuntos de dados, Bezerra2018 e BoT-IoT, que simulam redes IoT.

3. Conjuntos de Dados

Esta seção analisa características importantes dos conjuntos de dados estudados, podendo, assim, explorá-los de forma mais efetiva. Dois conjuntos de dados são utilizados nas avaliações dos modelos de aprendizado de máquina: um conjunto de dados criado por Bezerra et al. para a avaliação de métodos de detecção de intrusão em sistemas IoT [Bezerra et al., 2018], chamado neste artigo de Bezerra2018; e outro chamado de BoT-IoT, criado através da combinação de tráfego legítimo e tráfego gerado por *botnets* [Koroniotis et al., 2019]. Enquanto o primeiro conjunto de dados é obtido sob demanda aos autores, o segundo pode ser obtido em um repositório disponibilizado pelos autores[1]. A Tabela 1 resume as características principais dos conjuntos de dados.

[1] `https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/bot_iot.php`.

Tabela 1. Disponibilidade de amostras em cada conjunto de dados. Os fluxos de rede são caracterizados por grupos de pacotes com endereços IP e portas de origem e destino iguais.

Características \ Conjunto	Bezerra2018	BoT-IoT 5%	BoT-IoT
Amostras benignas	7.998	477	9.543
Amostras malignas	1.716.408	3.668.045	73.360.900
Tipo de amostra	Fluxos de rede	Pacotes de rede	Pacotes de rede
Tamanho do conjunto (GB)	0.62	0.97	16.00

Bezerra2018: O conjunto de dados Bezerra2018 é constituído por tráfego de rede legítimo, caracterizado por serviços como transmissão de vídeos, acesso a páginas *web* e conexões SSH (*Secure Shell*); e por tráfego malicioso, caracterizado por infecções em um microcomputador *Raspberry Pi* conectado à rede.

Diferentes *botnets* são instaladas em um microcomputador *Raspberry Pi*, que é um dispositivo com recursos limitados, simulando dispositivos IoT. As *botnets* podem ser controladas para direcionar ataques DDoS (*Distributed Denial-of-Service*). Elas são monitoradas de acordo com três perfis diferentes de rede, tornando possível a avaliação de ataques em diferentes tipos de redes IoT. Os perfis caracterizam ambientes de multimídia, com transmissão de vídeo e consumo de entretenimento; ambientes caracterizados pela transmissão exclusiva de vídeo e ambientes com transmissão de vídeo e outros serviços mais comuns, como acesso a sítios *web* e SSH. Essa variedade valida o conjunto de dados com base no princípio de que não existe um único padrão de rede IoT [Sivanathan et al., 2017].

A ferramenta *tcpdump* é utilizada para coletar o tráfego de pacotes da rede enquanto os dispositivos consomem vídeos e acessam a Internet. O conjunto de dados Bezerra2018 é agrupado em fluxos de rede e oferece, além dos dados obtidos com o *tcpdump*, dados relativos ao consumo de recursos computacionais dos diferentes dispositivos utilizados. No entanto, os últimos não são usados neste artigo, porque este trabalho visa a aplicação de modelos de aprendizado profundo sobre o tráfego da rede. Desta forma, a quantidade de amostras utilizadas desse conjunto se resume a $0,62$ GB de dados.

BoT-IoT: O conjunto de dados BoT-IoT é formado quase exclusivamente por amostras de tráfego malicioso. Como visto na Tabela 1, menos de 1% das amostras são benignas. Também vale notar que o conjunto completo atinge mais de 16 GB, então, como sugerido pelos autores, um subconjunto com 5% das amostras é utilizado nos experimentos.

O conjunto de dados é orientado a pacotes de rede e faz uso da ferramenta *tshark* para captura de pacotes. Para simular o tráfego típico de uma rede IoT, um servidor Node-Red com múltiplos sensores virtuais é instalado em máquinas virtuais Ubuntu, trocando mensagens através do protocolo MQTT (*Message Queuing Telemetry Transport*) com um servidor proprietário AWS (*Amazon Web Services*).

4. Metodologia Adotada

A aplicação de todos os modelos de aprendizado de máquina segue uma metodologia bem definida. Essa metodologia, observada na Figura 1, é descrita na Seção 4.1. Já a Seção 4.2 descreve o tratamento específico dos conjuntos de dados proposto para o funcionamento dos modelos mais complexos, como Redes Neurais Convolucionais (*Convolutional Neural Networks* – CNNs), Redes Neurais Recorrentes (*Recurrent Neural Networks* – RNNs) com LSTM (*Long Short-Term Memory*) e redes neurais autoassociativas.

4.1. Fluxo Adotado

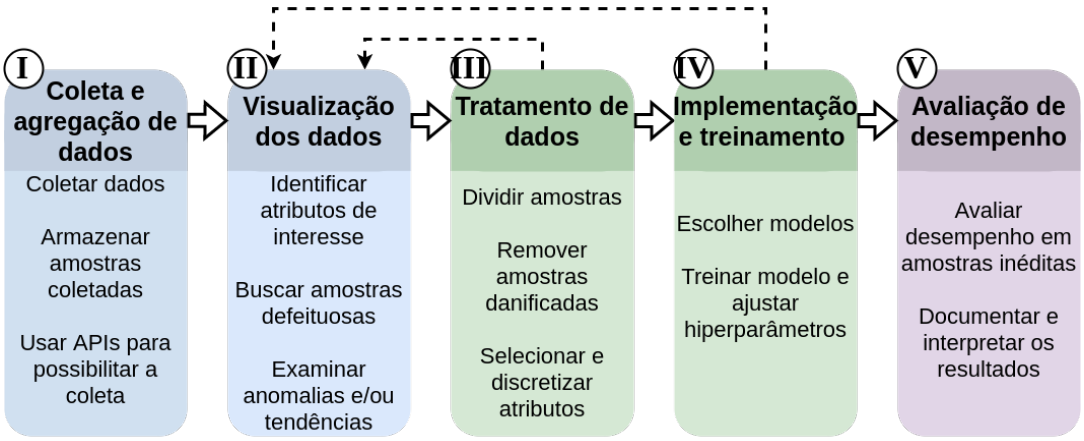


Figura 1. Fluxo de trabalho típico de aplicações de aprendizado de máquina. Adaptado de [Bochie et al., 2020].

I. **Coleta e agregação de dados**: os dados podem ser coletados e agregados a partir de diversas fontes, como sensores distribuídos, APIs (*Application Programming Interfaces*) ou sítios *web*. Neste trabalho, ao invés de coletar os próprios dados, a alternativa adotada foi utilizar os dados disponibilizados publicamente por Bezerra et al. [Bezerra et al., 2018] e Koroniotis et al. [Koroniotis et al., 2019].

II. **Visualização dos dados**: é importante realizar uma análise de alto nível sobre os dados, a fim de identificar possíveis inconsistências, como atributos repetidos, amostras corrompidas e a proporção entre cada tipo de amostra. Essa etapa é útil para identificar possíveis erros nas amostras geradas por sensores e outros dispositivos. Para os conjuntos analisados algumas características interessantes foram notadas imediatamente, em especial a grande quantidade de atributos sem amostras. Apenas em alguns casos os endereços IP das máquinas foram disponibilizados e múltiplos atributos que representavam apenas um valor. Esses pontos foram identificados para correção na etapa de pré-processamento dos dados.

III. **Tratamento e pré-processamento dos dados**: pode ser subdividido em até três etapas que são o tratamento de amostras danificadas e formatação de atributos, a seleção de atributos com maior poder preditivo e a discretização, sendo esta última opcional.
Tratamento de amostras danificadas e formatação de atributos: os atributos que não possuíam quantidade suficiente de amostras foram removidos inteiramente. Aqueles atributos com apenas um valor coletado também foram removidos a fim

de diminuir a dimensionalidade do problema e diminuir o tempo de treinamento dos modelos. Para cada cenário, o conjunto de dados original foi divido em dois conjuntos: conjunto de treino e conjunto de teste. O conjunto de treino também foi posteriormente fracionado a fim de obter um conjunto de validação. Apenas as informações presentes no conjunto de treino foram utilizadas para selecionar, normalizar e balancear as amostras em todos os conjuntos, a fim de evitar vazamento de dados. Atributos com poucos valores faltantes também tiveram seus valores preenchidos de acordo com uma estratégia definida previamente, como inserir valores médios, medianos ou mais frequentes.

Seleção de atributos com maior poder preditivo: depois da redução gerada pelo tratamento de amostras danificadas, técnicas de redução de dimensionalidade, como análise de variância e Análise de Componentes Principais (*Principal Component Analysis* – PCA) foram utilizadas para a redução de atributos. Também é importante remover atributos que não generalizam, como endereços IP e MAC de origem e destino de cada pacote, além de atributos redundantes, como representações diferentes de uma mesma grandeza.

Discretização: para modelos que utilizam a quantidade de amostras presentes em um atributo para classificação, como o Naïve Bayes, atributos com extensas faixas de valores possíveis foram mapeados para intervalos fixos. Essa técnica é uma alternativa útil a tipos de codificação como *one-hot encoding*, que não são apropriadas para atributos com alta cardinalidade, em vista do consequente aumento no número de dimensões do problema.

IV. **Escolha e implementação do modelo de aprendizado**: como descrito na etapa de pré-processamento, o conjunto de dados original foi dividido em conjunto de treino, de validação e de teste. Não utilizar um conjunto de validação para o ajuste de hiperparâmetros é um erro grave, porém muito comum, que leva o modelo ao sobreajuste (*overfitting*) e invalida os resultados. Todos os modelos são treinados com o conjunto de treino e têm seus hiperparâmetros ajustados de acordo com o desempenho no conjunto de validação. O conjunto de teste foi utilizado apenas para avaliar o desempenho final. De acordo com os resultados preliminares obtidos nesta etapa, pode ser útil reavaliar o pré-processamento aplicado aos dados. Usualmente, uma amostra é considerada como positiva quando ela é interessante de ser detectada. Nesse cenário, as amostras de ataques seriam normalmente consideradas como positivas. Porém, como será explicado na Seção 5, o desbalanceamento pronunciado dos conjuntos de dados levou à inversão da lógica e, por isso, as amostras benignas passaram a ser consideradas como positivas para fins de comparação de desempenho dos modelos.

V. **Avaliação dos resultados**: todos os modelos tiveram seu desempenho avaliado no conjunto de teste, que não foi utilizado durante o treinamento. Por se tratar de um classificador binário, a predição de uma amostra, como positiva ou negativa, se enquadra nas seguintes categorias: **Verdadeiro Positivo** (*True Positive* – TP) toda amostra positiva classificada de fato como positiva, **Verdadeiro negativo** (*True Negative* – TN) toda amostra negativa classificada como negativa, **Falso Positivo** (*False Positive* – FP) toda amostra negativa classificada como positiva e, finalmente, **Falso Negativo** (*False Negative* – FN) toda amostra positiva classificada como negativa. As seguintes métricas, específicas para problemas de classificação binária,

foram escolhidas para comparação dos resultados [Vinayakumar et al., 2019]:
Acurácia: é a fração de amostras classificadas corretamente em relação ao total de amostras avaliadas.

$$\frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precisão: mede a fração de amostras classificadas como positivas que são realmente amostras positivas.

$$\frac{TP}{TP + FP} \tag{2}$$

Sensibilidade (*Recall*): mede a fração de amostras positivas classificadas corretamente.

$$\frac{TP}{TP + FN} \tag{3}$$

F1: é a média harmônica entre precisão e sensibilidade.

$$2 * \frac{\text{Precisão} * \text{Sensibilidade}}{\text{Precisão} + \text{Sensibilidade}} \tag{4}$$

Taxa de Alarmes Falsos (*False Alarm Rate* – FAR): também conhecida como taxa de falsos positivos. É a probabilidade de uma amostra benigna ser classificada como ataque.

$$\frac{FP}{FP + TN} \tag{5}$$

4.2. Reestruturação de Dados Adotada

A diferença entre o número de amostras malignas e benignas é imediatamente notável ao se examinar os dois conjuntos de dados. O desbalanceamento entre as amostras pode ser visto na Tabela 1. Para lidar com esse tipo de problema, é usual utilizar técnicas de balanceamento de dados como amostradores aleatórios ou algoritmos não supervisionados que, respectivamente, copiam amostras ou geram amostras artificiais. Porém, essas técnicas não preservam as estruturas presentes no conjunto de dados, como características sequenciais entre pacotes e fluxos, o que pode invalidar o uso de algoritmos como redes neurais recorrentes [Haddadpajouh et al., 2018]. Além disso, redes neurais autoassociativas utilizam o desbalanceamento das amostras em uma abordagem não supervisionada para a detecção de *outliers*. Tendo isso em mente, os conjuntos de dados originais desbalanceados foram utilizados para treinar os modelos de aprendizado. Ademais, duas abordagens diferentes de separação de dados são utilizadas para os modelos de rede neural recorrente e de rede neural autoassociativa. Para serem utilizados, esses modelos precisam de uma separação característica como visto a seguir:

Rede neural recorrente: após a separação dos três conjuntos, as amostras são ordenadas em relação a seus índices para reconstituir a relação temporal coletada. Após a reorganização, um algoritmo de janela deslizante é usado para agrupar as amostras que serão apresentadas para o treinamento do modelo. Uma visão resumida da transformação pode ser vista na Figura 2. O tamanho da janela se apresenta como um novo hiperparâmetro que também é ajustado.

Rede neural autoassociativa: nessa rede, apenas as amostras de ataque são usadas durante o treino. O objetivo é utilizar a capacidade de aprendizado não supervisionado

dessas redes para detectar as amostras menos presentes no conjunto. O conjunto de teste, então, é composto de todas as amostras benignas, que são menos presentes, e a mesma quantidade de amostras de ataque. As amostras de ataque restantes são dividas nos conjuntos de teste e validação.

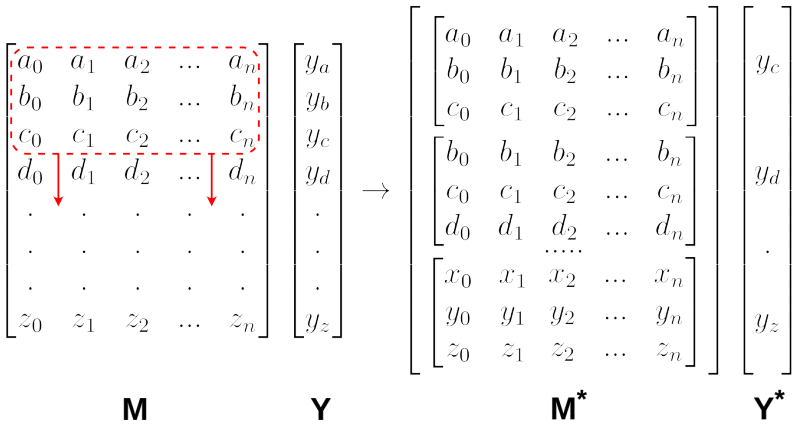


Figura 2. O conjunto de dados original é representado por uma matriz M com dimensão $z \times (n+1)$, onde z é o número total de amostras e n a quantidade de atributos de cada amostra. O conjunto é redimensionado com uma janela de largura $l = 3$ e passo $p = 1$. O resultado é uma nova matriz M^* com dimensão $(z-2) \times 3 \times (n+1)$. Como pode ser observado, as duas primeiras saídas são descartadas e a saída correspondente à nova amostra é escolhida como a saída original da última amostra usada pela rede recorrente. As novas amostras da matriz M^* são as matrizes que agrupam 3 amostras originais da matriz M.

Apesar do esforço adicional para reorganizar o formato dos dados, as redes neurais recorrentes e autoassociativas têm se provado extremamente úteis para o desenvolvimento de IDSs, especialmente para cenários com dados desbalanceados, em que a abordagem pode ser interpretada como um tipo de detecção de anomalias [Chalapathy e Chawla, 2019, Luo e Nagarajan, 2018]. Para o uso das redes neurais convolucionais, os atributos de cada amostra foram rearranjados em formato matricial e o uso de *zero padding* foi necessário para garantir o preenchimento total de cada matriz, mantendo um formato padrão de amostra de duas dimensões.

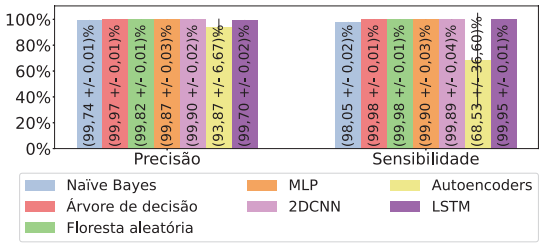
5. Resultados

Todas os códigos utilizados para implementar os modelos de aprendizado e para obter os resultados desta seção estão disponíveis em um repositório GitHub[2]. A seleção de hiperparâmetros e o espaço de busca definido para cada modelo podem ser vistos no repositório. Um computador Intel Core i7 $-$ 7700 3,60 GHz com 4 núcleos de processamento e com 32 GB RAM foi utilizado durante o treinamento.

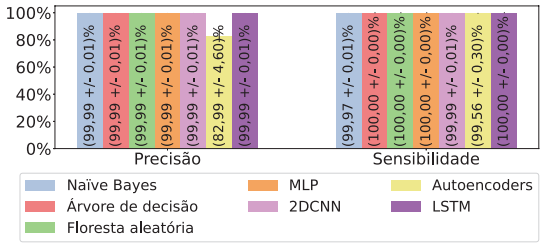
Inicialmente, o conjunto de dados é dividido com a proporção $55/15/30$ para compor os conjuntos de treino, validação e teste, respectivamente. Após o pré-processamento descrito na Seção 4, uma etapa de ajuste de hiperparâmetros é realizada e, finalmente, sete diferentes modelos de aprendizado de máquina são aplicados aos conjuntos de dados.

[2] `https://github.com/kaylani2/sbseg2020`.

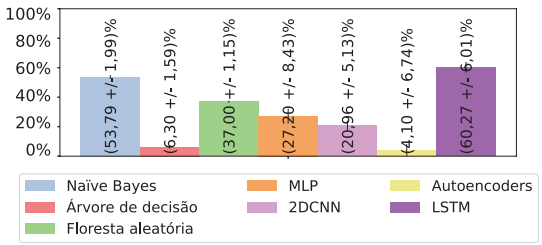
Este trabalho não trata o desbalanceamento dos conjuntos de dados, apresentado na Tabela 1, pois modelos como RNNs seriam prejudicados com a possível perda do sequenciamento temporal das amostras. A decisão de não balancear teve por objetivo tornar a comparação entre os modelos mais justa, já que nenhuma manipulação seria executada sobre os traces para análise dos modelos. A consequência, porém, foi a inversão de lógica usualmente utilizada em problemas de detecção de ataque, nos quais as amostras de ataques são consideradas positivas. Essa lógica usual aliada ao desbalanceamento nos conjuntos de dados leva a falsa noção de ótimos desempenhos para todos os modelos, visto que existem muito mais amostras de ataques nos dois conjuntos. Isso tem um efeito negativo sobre os resultados, fazendo-os predizer corretamente mais ataques do que amostras benignas. Se por um lado a taxa de verdadeiros positivos se torna superdimensionada, elevando o valor de métricas como precisão por exemplo; por outro lado, a taxa de alarmes falsos para os modelos treinados com ataques considerados como positivos revela grandes diferenças entre os desempenhos dos modelos. As Figuras 3(a) e 3(b) apresentam os resultados superdimensionados para todas as métricas propostas usando a lógica usual, enquanto as Figuras 3(c) e 3(d) mostram os resultados de alarmes falsos que demonstra desempenhos totalmente diferentes. Para contornar esse problema, este trabalho arbitra que as amostras de ataques são consideradas **negativas**.



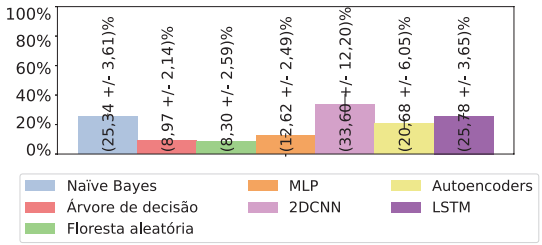
(a) Precisão e sensibilidade calculadas sobre o conjunto Bezerra2018.



(b) Precisão e sensibilidade calculadas sobre o conjunto BoT-IoT.



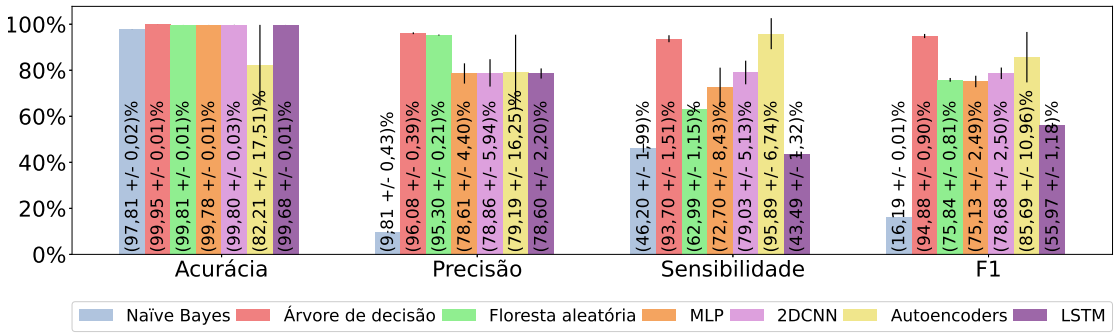
(c) Taxa de alarmes falsos calculada sobre o conjunto Bezerra2018.



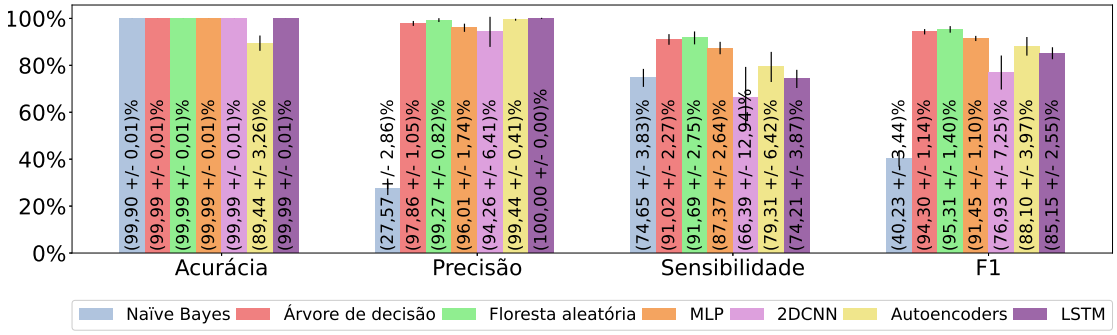
(d) Taxa de alarmes falsos calculada sobre o conjunto BoT-IoT.

Figura 3. Resultados dos modelos de aprendizado aplicados aos dois conjuntos de dados ao considerar amostras de ataques como positivas.

Levando em conta a lógica inversa, na qual as amostras de ataque são consideradas negativas, os desempenhos médios de cada modelo, a partir das métricas mais comuns em problemas de classificação, podem ser vistos na Figura 4. Nota-se, imediatamente, o baixo desempenho do modelo Naïve Bayes, com *F1* de $(16,19 \pm 0,01)\%$ e $(40,23 \pm 3,44)\%$ nos conjuntos Bezerra2018 e BoT-IoT, respectivamente. Isso se deve à dificuldade que um modelo puramente probabilístico tem em obter uma boa representação



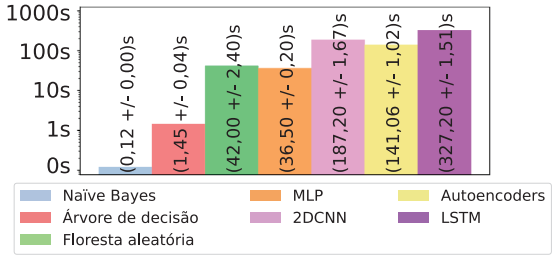
(a) Desempenho dos modelos de aprendizado no conjunto de dados Bezerra2018.



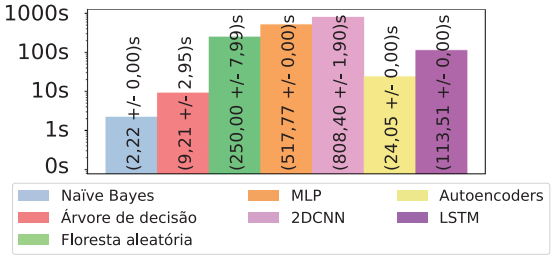
(b) Desempenho dos modelos de aprendizado no conjunto de dados BoT-IoT.

Figura 4. Resultados dos modelos de aprendizado aplicados aos dois conjuntos de dados.

de um conjunto desbalanceado, o que também é indicado pela alta acurácia obtida. O modelo fica "viciado" nas amostras mais presentes nos conjuntos de dados, mesmo adotando amostras de ataque como amostras negativas. Dentre os modelos de aprendizado não profundos, destaca-se o modelo de árvores de decisão que atingiu, em ambos os conjuntos de dados, desempenho acima de 90% em todas as métricas após a seleção de atributos. É interessante mencionar os tempos de treinamento do modelo de floresta aleatória em relação aos do modelo de árvores de decisão, que precisa de muito mais treinamento para atingir um desempenho marginalmente melhor, como ilustra a Figura 5.



(a) Tempo de treinamento dos modelos de aprendizado no conjunto de dados Bezerra2018



(b) Tempo de treinamento dos modelos de aprendizado no conjunto de dados BoT-IoT.

Figura 5. Tempo de treinamento dos modelos de aprendizado aplicados aos dois conjuntos de dados.

Já nos modelos de aprendizado profundo, como esperado, o *Multilayer Perceptron*

(MLP) atinge alta acurácia em troca de uma sensibilidade reduzida. Esses valores são explicados por sua característica de aproximador universal e pelo desbalanceamento dos conjuntos de dados. Já a rede neural recorrente com LSTM, ao utilizar células de memória e considerar informações entre amostras, atinge uma acurácia quase perfeita enquanto é capaz de diminuir o viés da rede em relação ao MLP, o que pode ser visto pela precisão elevada da RNN com LSTM no conjunto BoT-IoT. No entanto, essa melhora tem como compromisso um aumento expressivo nos tempos de treinamento.

Também é interessante observar o alto desempenho das redes neurais autoassociativas, que aparecem como uma solução robusta em relação ao desbalanceamento dos conjuntos de dados. Isso se deve ao fato de que as redes autoassociativas são treinadas de forma não supervisionada com um tipo de amostra para que o erro de reconstrução seja minimizado em amostras familiares. Esse tipo de abordagem surge como uma possível técnica de treinamento *online*, onde a rede pode ser treinada com novas amostras que são capturadas em uma interface de rede para aprender a identificar amostras não familiares.

A Figura 5 mostra a diferença existente no tempo de treinamento para os diferentes modelos e conjuntos de dados. A diferença entre as amostras serem orientadas a fluxo ou a pacotes de rede influencia a quantidade de informação apresentada por amostra, sendo maior no caso orientado a fluxo. Isso, juntamente à quantidade de amostras utilizadas de cada conjunto de dados, explica o maior tempo para o treino dos modelos sobre o Bezerra2018 se comparado ao BoT-IoT, como $(187,20 \pm 0,20)$ segundos e $(808,40 \pm 1,90)$ segundos, respectivamente, para o treino das redes neurais convolucionais.

Conclui-se que, para os dois conjuntos de dados apresentados, modelos de aprendizado tradicionais, como árvores de decisão, são capazes de atingir excelentes desempenhos, mesmo nos conjuntos desbalanceados, com pontuação F1 superior a 94% nos dois cenários. Entretanto, modelos de aprendizado profundo, como MLP e redes autoassociativas, também são capazes de atingir bons desempenhos mesmo sendo mais penalizados pelo desbalanceamento dos conjuntos de dados.

6. Conclusão e Trabalhos Futuros

Este artigo apresentou uma avaliação de desempenho de modelos de aprendizado de máquina tradicionais e profundos e os comparou quando aplicados a dois conjuntos de dados contendo ataques a redes IoT. O desempenho dos modelos tradicionais como árvore de decisão e floresta aleatória atingiram bons resultados, com as métricas acurácia, precisão e F1 próximas de 95%.

Conclui-se que, para modelos de aprendizado mais sofisticados, como redes neurais recorrentes, conjuntos de dados com características sequenciais apropriadamente definidas são necessários, visto que técnicas tradicionais de separação dos conjuntos podem prejudicar o desempenho. Devido a essa necessidade, também se faz necessário o uso de algoritmos capazes de separar os dados e preservar as estruturas temporais.

Para os modelos supervisionados mais complexos, o resultado não se mostrou necessariamente superior aos dos modelos tradicionais. Além disso, por possuírem mais parâmetros, os tempos de treinamento desses modelos foram substancialmente maiores do que os tradicionais, com $(187,20 \pm 0,20)$ segundos e $(1,45 \pm 0,04)$ segundos para os modelos 2DCNN e árvore de decisão no conjunto de dados Bezerra2018. O uso de

abordagens não supervisionadas, como redes neurais autoassociativas, se mostrou efetivo em identificar ataques em conjuntos desbalanceados. Este resultado é promissor, visto que a operação real em redes de computadores tende a conter tráfego majoritariamente benigno ou maligno.

Futuramente, pretende-se avaliar o desempenho de novas abordagens no campo de aprendizado profundo, especificamente as redes *transformers* [Vaswani et al., 2017]. Também pretende-se comparar o desempenho das técnicas estudadas, quando aplicadas ao mesmo conjunto de dados, ao ser agrupado por pacotes ou por fluxo. O uso de métodos *ensemble* também será analisado a fim de prover um melhor limite de decisão para as redes autoassociativas. A avaliação de dimensionamento dinâmico dos dados também está prevista, visto que a duração de cada fluxo de dados pode influenciar na classificação de um ataque, característica que não é aproveitada quando os modelos convolucionais e recorrentes usam janelas de tamanho fixo. Por fim, a implantação dos modelos apresentados em tempo real para avaliação do desempenho de um possível IDS na detecção e classificação de ataques *online* é uma direção possível.

Referências

AL-Hawawreh, M., Moustafa, N. e Sitnikova, E. (2018). Identification of malicious activities in industrial internet of things based on deep learning models. *Journal of Information Security and Applications*, 41:1–11.

Almomani, I., Al-Kasasbeh, B. e Al-Akhras, M. (2016). WSN-DS: A dataset for intrusion detection systems in wireless sensor networks. *Journal of Sensors*, 2016.

Bezerra, H., da Costa, V., Turrisi, V., Martins, A., Barbon, S. R., Rodrigo, M. e Zarpelão, B. B. (2018). Providing IoT host-based datasets for intrusion detection research. Em *Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*.

Bezerra, V. H., da Costa, V. G. T., Barbon Junior, S., Miani, R. S. e Zarpelão, B. B. (2019). IoTDS: A one-class classification approach to detect botnets in internet of things devices. *Sensors (Basel, Switzerland)*, 19(14).

Bochie, K., Gilbert, M. S., Gantert, L., Barbosa, M. S. M., Medeiros, D. S. V. e Campista, M. E. M. (2020). Aprendizado profundo em redes desafiadoras: Conceitos e aplicações. Em *Minicursos do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*.

Buczak, A. L. e Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys Tutorials*, 18(2):1153–1176.

Chalapathy, R. e Chawla, S. (2019). Deep learning for anomaly detection: A survey.

Goasduff, L. (2019). Gartner says 5.8 billion enterprise and automotive IoT endpoints will be in use in 2020. *Gartner*. Acessado em 22/05/2020.

Guan, Y. e Ge, X. (2018). Distributed attack detection and secure estimation of networked cyber-physical systems against false data injection attacks and jamming attacks. *IEEE Transactions on Signal and Information Processing over Networks*, 4(1):48–59.

Haddadpajouh, H., Dehghantanha, A., Khayami, R. e Choo, K.-K. R. (2018). A deep recurrent neural network based approach for internet of things malware threat hunting. *Future Generation Computer Systems*, 85:88–96.

Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P. e Sikdar, B. (2019). A survey on IoT security: Application areas, security threats, and solution architectures. *IEEE Access*, 7:82721–82743.

Huang, H., Ding, S., Zhao, L., Huang, H., Chen, L., Gao, H. e Ahmed, S. H. (2020). Real-time fault detection for IIoT facilities using GBRBM-based DNN. *IEEE Internet of Things Journal*, 7(7):5713–5722.

Koroniotis, N., Moustafa, N., Sitnikova, E. e Turnbull, B. (2019). Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset. *Future Generation Computer Systems*, 100:779–796.

Luo, T. e Nagarajan, S. G. (2018). Distributed anomaly detection using autoencoder neural networks in WSN for IoT. Em *2018 IEEE International Conference on Communications (ICC)*, p. 1–6.

Purohit, H., Tanabe, R., Ichige, T., Endo, T., Nikaido, Y., Suefusa, K. e Kawaguchi, Y. (2019). MIMII dataset: Sound dataset for malfunctioning industrial machine investigation and inspection. Em *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, p. 209–213, New York University, NY, USA.

Sharafaldin, I., Lashkari, A. H. e Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSP 2018 - Proceedings of the 4th International Conference on Information Systems Security and Privacy*, p. 108–116.

Sivanathan, A., Sherratt, D., Gharakheili, H. H., Radford, A., Wijenayake, C., Vishwanath, A. e Sivaraman, V. (2017). Characterizing and classifying iot traffic in smart cities and campuses. Em *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, p. 559–564.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. e Polosukhin, I. (2017). Attention is all you need. *ArXiv*, abs/1706.03762.

Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A. e Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. *IEEE Access*, 7:41525–41550.