Um Sistema Inteligente para Detecção de DDoS em Ambientes Inteligentes Baseado em Computação em Nuvem e em Névoa

Wanderson L. Costa[1], **Ariel L. C. Portela**[1], **Rafael L. Gomes**[1]

[1]Universidade Estadual do Ceará (UECE), Fortaleza, Ceará, Brasil.

wanderson.leonardo@ifpi.edu.br, ariel.portela@aluno.uece.br,

rafa.lopes@uece.br

Abstract. *Urban spaces are becoming Smart Environments (SE) that are composed of a huge number of heterogeneous devices: both personal devices (smartphones, notebooks, etc.) and Internet of Things (IoT) devices (sensors, actuators, and others). One of the existing problems of SEs is the detection of Distributed Denial of Service (DDoS) attacks, due to the vulnerabilities of IoT devices. Thus, it is necessary to implement solutions that can detect DDoS in SEs with scalability, adaptability and heterogeneity (application execution, hardware capacity and different protocols). Within this context, this paper presents an Intelligent System for detection of DDoS in SEs, applying Machine Learning (ML) combined with fog and cloud computing. The experiments performed, using real network traffic, suggest that the proposed system reaches 99% accuracy while reducing the volume of data exchanged and the detection time.*

Resumo. *Os espaços urbanos estão se tornando ambientes inteligentes (SEs) os quais são compostos por uma grande quantidade de dispositivos heterogêneos: pessoais (celulares, notebooks, tablets, etc) e dispositivos de Internet das Coisas (IoT) (sensores, atuadores entre outros). Um dos problemas existentes dos SEs é a detecção de ataques de Negação de Serviço Distribuído (DDoS), devido às vulnerabilidades dos dispositivos IoT. Dessa forma, é necessário implantar soluções que possam detectar DDoS em SEs com escalabilidade, adaptabilidade e heterogeneidade (execução de aplicativos, capacidade de hardware e protocolos distintos). Dentro deste contexto, este artigo apresenta um Sistema Inteligente para detecção de ataques DDoS em SEs, aplicando abordagem de Aprendizado de Máquina (ML) em conjunto com Computação em Nuvem e em Névoa. Os experimentos realizados, usando tráfego de rede real, sugerem que o sistema proposto atinge* 99% *de acurácia, enquanto reduz o volume de dados trocados e o tempo de detecção.*

1. Introdução

Em um futuro não muito distante, todos os nossos objetos do dia a dia estarão conectados à Internet e equipados com capacidades de sensoriamento, e poder de processamento suficientes para explorar todos os benefícios potenciais da chamada (do inglês, *Internet of Things* - IoT). Até mesmo os dispositivos mais simples, irão se conectar a outros objetos para compartilhar informações possibilitando o desenvolvimento de vários serviços.

A IoT é uma rede de objetos físicos, que inclui desde utensílios domésticos como lâmpadas e geladeiras, veículos, etiquetas de endereçamento, dispositivos médicos e outros dotados de tecnologia embarcada, sensores, câmeras de vigilância conectados com

a internet, esses objetos são capazes de reunir e de transmitir informações auxiliando na execução de várias tarefas.

Novos ecossistemas surgem e têm sido denominados como ambientes inteligentes (do inglês, *Smart Environments* - SEs). Esses contextos possuem serviços singulares para aprimorar a qualidade de vida dos usuários finais. Os SEs são compostos por dispositivos IoT (como sensores e atuadores) e dispositivos pessoais (como notebooks, smartphones, tablets,etc) [Li et al. 2018].

Ademais, ao observar esses equipamentos no contexto dos SEs ressaltam-se duas características indiscutíveis: grande quantidade de dispositivos e heterogeneidade. Como consequência, os SEs tendem a produzir maiores volumes de informações na rede do que redes tradicionais, devido à enorme escala de dispositivos na rede, bem como aos vários tipos de aplicativos executados nas camadas de redes mais superiores desses equipamentos. Todas essas questões trazem novos desafios relacionados à gestão e planejamento das SEs e os seus serviços neles executados [Ahmed et al. 2016].

Um desses desafios dos SEs é a detecção de ataques de negação de serviço distribuído (DDoS), que visam tornar o acesso a um ou mais alvos indisponíveis ao esgotar seus recursos por meio de múltiplas solicitações ilegítimas. Os ataques DDoS vêm de inúmeras vulnerabilidades de segurança nos dispositivos, especialmente dispositivos IoT [Andrea et al. 2015, Diro and Chilamkurti 2018], que afetam diretamente a Qualidade do Serviço (QoS) e a Qualidade da Experiência (QoE). Como resultado, nos últimos anos, diversos ataques cibernéticos realizados na Internet ocorreram por meio da infecção de dispositivos IoT [Brun et al. 2018].

Por outro lado, as limitações de hardware dos dispositivos IoT impedem a implantação de soluções de segurança que rodam neles. Uma abordagem alternativa é o uso de técnicas de aprendizado de máquina (ML), que entendem o comportamento dos dados disponíveis e melhoram progressivamente o seu entendimento [Doshi et al. 2018]. Porém, é necessário utilizar as características mais relevantes do tráfego de rede para posteriormente treinar o mecanismo de detecção de ataques DDoS utilizando ML, uma vez que a consideração de características inadequadas prejudica a acurácia das técnicas de ML.

Dentro deste contexto, este artigo apresenta um Sistema Inteligente para detecção de ataques DDoS em SEs, o qual integra computação em Névoa (*Fog Computing*) e em Nuvem (*Cloud Computing*), dividindo as tarefas realizadas entre esses dois ambientes computacionais para reduzir o tempo de resposta e melhorar a acurácia. O sistema proposto é baseado nos seguintes princípios: (I) Monitoramento da rede, coleta de dados sobre os fluxos da rede no SE; (II) Seleção de Características, identificação das principais características para detecção de DDoS em SEs; (III) Segmentação de tráfego, separação de fluxos de rede de dispositivos IoT e dispositivos pessoais; e, (IV) ML para Detecção, treinamento de modelos de ML usando os dados sobre os fluxos da rede para detectar ataques DDoS.

O sistema proposta atende simultaneamente três aspectos cruciais para detecção de ataques DDoS em SEs:

- Heterogeneidade: Aplica-se uma abordagem de segmentação de tráfego, permitindo identificar o padrão de tráfego e, consequentemente, detectar os ataques DDoS em ambientes com dispositivos de diferentes características.

- Escalabilidade: Integração entre computação em Névoa e Computação em Nuvem habilita o processamento de dados e treinamento do modelo de ML nos ambientes computacionais mais adequados. Adicionalmente, são executadas técnicas de seleção de características para melhorar o uso de dados, visto que são aplicados no modelo e enviados da Névoa pra Nuvem somente os dados mais adequados. Como consequência da integração e seleção, tem-se a redução do volume de dados trocados entre Névoa e Nuvem (menos impacto na infraestrutura de rede) e a minimização do tempo de detecção de ataques DDoS.
- Adaptabilidade: A utilização de um monitoramento constante do tráfego de rede para alimentar a base de dados usada para o treinamento periódico do modelo de ML possibilita o sistema proposta a adaptar-se as mudanças de perfil de tráfego, aumentando a capacidade de detecção de ataques DDoS.

Os experimentos realizados, usando um conjunto de dados de rede real, sugerem que o sistema proposto atinge seu objetivo de forma satisfatória, com alta acurácia de detecção (alcançando cerca de 99% de acurácia na detecção), enquanto reduz o volume de dados trocados e o tempo de detecção de ataques DDoS.

De maneira geral, este artigo possui as seguintes contribuições: (A) Projeto de um sistema que integra Computação em Nuvem e em Névoa para detectar ataques DDoS em SEs; (B) Uma avaliação das diferentes abordagens de seleção de características e modelos de ML que podem ser aplicados no contexto, analisando aspectos de acurácia, volume de dados e tempo de detecção; (C) Uma abordagem de segmentação de tráfego que permite identificar se um dispositivo em um SE é pessoal ou IoT, a qual pode ser aplicada em outros contextos; e, (D) Experimentos realizados usando dados de redes reais com ataques DDoS.

O restante deste artigo é organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados. A Seção 3 detalha o sistema proposto, enquanto que a Seção 4 descreve os experimentos realizados e uma análise dos resultados. Por fim, a Seção 5 conclui o artigo e apresenta os trabalhos futuros.

2. Trabalhos Relacionados

Nesta seção apresenta-se os trabalhos relacionados à solução para segurança cibernética e detecção de DDoS usando Inteligência Artificial (IA). A Tabela 1 apresenta os trabalhos existentes, destacando a inovação da nossa proposta quando comparada com estas pesquisas. Na Tabela 1, a coluna *Problema* apresenta o principal problema onde o trabalho relacionado atua, enquanto a coluna e *Abordagem do tema* informa o a estratégia aplicada pelos autores.

Hamamoto et al. [Hamamoto et al. 2018] propuseram um esquema baseado na combinação de algoritmo genético e lógica fuzzy. As estruturas de aprendizagem trabalham juntas para traços de rede criados pelo algoritmo genético. A lógica fuzzy define quando a rede está normal ou sob um ataque cibernético de uma assinatura gerada anteriormente. Apesar de ser baseado em tráfego real e técnicas de IA, o esquema proposto foca nas redes tradicionais, ou seja, os autores não consideram as características do SE.

Sharafaldin et al. [Sharafaldin et al. 2019] apresentam um estudo sobre as características do tráfego de redes mais importantes para detecção de diferentes tipos de ataques DDoS em redes tradicionais, ou seja, redes TCP/IP. Nos experimentos realizados foram

projetadas e implantadas duas redes com computadores tradicionais, ou seja, o comportamento extraído das amostras do conjunto de dados, se torna diferente em comparação com o de redes projetadas com dispositivos IoT. O comportamento de redes IoT se comunica com um pequeno conjunto finito de pontos de extremidade e são propensos a ter padrões de tráfego de rede repetitivos (pacotes pequenos em intervalos de tempo fixos para fins de registro, por exemplo).

Yamauchi et al. [Yamauchi et al. 2019] descrevem um modelo para detectar operações anômalas de dispositivos IoT em casas inteligentes (*Smart Homes* - SHs) com base no comportamento do usuário. O modelo aprende a sequência de atividades realizadas por hora do dia e, então, compara a sequência atual com as sequências aprendidas para a condição correspondente à condição atual. Caso apresente alguma alteração prédefinida, o método classifica a operação como uma anomalia de dispositivo IoT. Portanto, este modelo proposto pelos autores limita-se ao compreendimento de ambientes de SHs.

Doshi et al. [Doshi et al. 2018] realizaram a detecção de ataques DDoS em andamento por meio do comportamento do fluxo de IoT em casas inteligentes. A abordagem implanta *middleboxes*, agindo como *proxy* na rede para observar, armazenar, processar e controlar o tráfego de rede que vai para a Internet. Essa estratégia monitora as características do fluxo, como tempo de chegada entre pacotes, pontos de extremidade e outros. As informações coletadas servem como entrada para uma técnica de ML para criar um modelo que identifique um possível ataque. Assim, a abordagem de seleção aplicada é muito limitada quando comparada com a análise realizada para desenvolver o mecanismo proposto nesta pesquisa.

Diro e Chilamkurti [Diro and Chilamkurti 2018] e Brun et al. [Brun et al. 2018] apresentaram um sistema de detecção de ataques e projetaram uma arquitetura ML, respectivamente, baseada em aprendizado profundo para IoT, comparando esta técnica com outras abordagens de aprendizado de máquina existentes. Os autores consideram uma estrutura de nuvem, onde as informações coletadas em IoTs distintas são recebidas e processadas em um nó mestre da nuvem. A avaliação do sistema empregou como dados de entrada do NSL-KDD, considerando a centralização de todos os dados coletados na nuvem. Os autores não incluíram nenhuma estratégia para evitar a sobrecarga de transmissão da troca de dados. Em [Brun et al. 2018], o modelo de aprendizado profundo usa informações sobre a troca de mensagens entre os dispositivos IoT e o gateway IoT, e o gateway IoT e a nuvem. Todos os pacotes são enviados para o módulo de treinamento de aprendizagem profunda na nuvem. A avaliação da arquitetura não considera o volume de dados trocados para permitir o treinamento do modelo, ou seja, não evita a sobrecarga de transmissão.

Meidan et al. [Meidan et al. 2018] apresentam uma abordagem de rede neural não supervisionada usando *Deep Autoencoders* para detectar botnets em redes IoT. Os autores sugerem que apenas vinte e três características para treinar o método de aprendizagem são suficientes para atingir a precisão adequada. Os experimentos foram realizados em um ambiente de teste composto por dispositivos IoT. No entanto, este trabalho é específico para detecção de botnet e não considera a capacidade de processamento dos dispositivos e outras limitações existentes nos SEs.

Embora vários estudos tenham sido realizados com a finalidade de identificar ca-

Tabela 1. Trabalhos relacionados.

Referência	Problema	Abordagem
Hamamoto et al. [Hamamoto et al. 2018]	Anomalias de rede.	Algoritmo Genético e Lógica Fuzzy.
Sharafaldin et al. [Sharafaldin et al. 2019]	DDoS em TCP/IP.	Experimentos de *Testbed* para análise de características.
Yamauchi et al. [Yamauchi et al. 2019]	Detecção de anomalias.	Análise de comportamento do dispositivo.
Doshi et al. [Doshi et al. 2018]	Detecção de DDoS.	*Machine Learning*.
Diro et al. [Diro and Chilamkurti 2018]	Anomalias de rede.	Machine Learning em Nuvem.
Brun et al. [Brun et al. 2018]	Anomalias de rede.	Deep Learning.
Meidan et al. [Meidan et al. 2018]	Detecção de Botnet.	Deep Autoencoders.
Este trabalho	DDoS em ambientes inteligentes.	Segmentação de tráfego, Seleção de características com Machine Learning sobre Fog e Cloud.

tegorias de tráfego e anomalias no tráfego (como os ataques DDoS), nenhum desses trabalhos acima abordam um cenário realístico de um ambiente inteligente com a necessidade tratar simultaneamente aspectos de escalabilidade, heterogeneidade e adaptabilidade.

3. Proposta

Ambientes inteligentes são compostos por dispositivos IoT heterogêneos, cada um desses dispositivos segue funcionalidades específicas e, consequentemente, um comportamento de rede singular para uma determinada categoria de dispositivos.

Essas características dos ambientes inteligentes aumentam a complexidade do gerenciamento e, consequentemente, o desenvolvimento de soluções de segurança, devido às limitações dos dispositivos (potência de processamento, consumo de energia, etc). Uma das soluções de segurança mais importantes é a detecção de ataques DDoS. Com base nisso, esta pesquisa propõe um sistema de Detecção Inteligente de DDoS utilizando técnicas de ML aplicando uma arquitetura de Computação em Nuvem (*Cloud*) e em Névoa (*Fog*) combinadas. Uma visão geral do sistema proposto é apresentada na Figura 1.

O sistema proposto realiza as seguintes etapas: (I) Monitoramento da Rede; (II) Extração de características de fluxo; (III) Segmentação de tráfego; (IV) Seleção de características; (V) Formação do Conjunto de Dados de Conhecimento; (VI) Treinamento de Modelo; e, (VII) Detecção de DDoS.

Adicionalmente, o desenvolvimento do sistema proposto engloba duas etapas de pré-processamento para a construção do conhecimento básico: observação sobre DDoS, uma pesquisa para distinguir as características importantes da execução do ataque; e, análise de atributos de tráfego, um estudo para identificar as características mais impor-

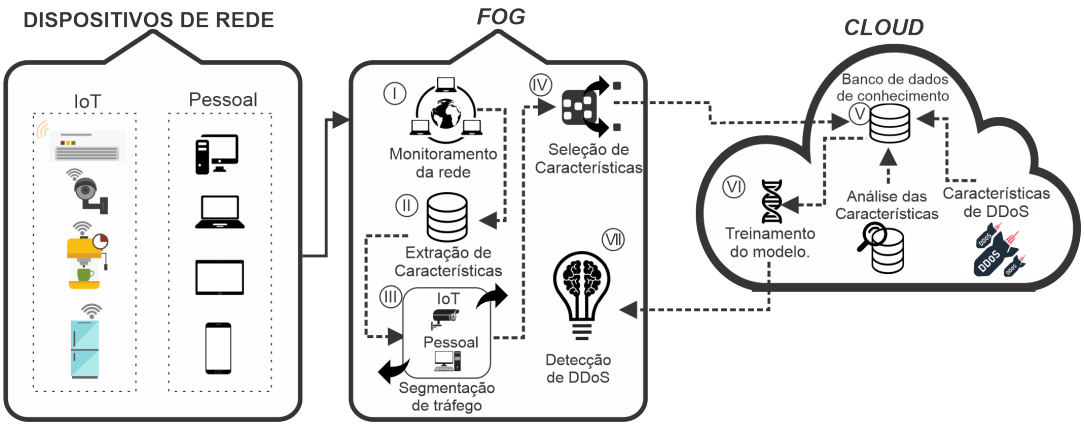

Figura 1. Visão geral do sistema proposto

tantes para melhorar a acurácia da detecção de DDoS.

Todas as etapas do fluxo de processamento de dados do sistema proposto estão ilustradas na Figura 2, onde são destacadas as tarefas realizadas e as técnicas consideradas (na seleção de características e treinamento do modelo). Essas etapas são executadas sequencialmente, trocando dados entre Névoa e Nuvem, conforme estrutura apresentada na Figura 1.

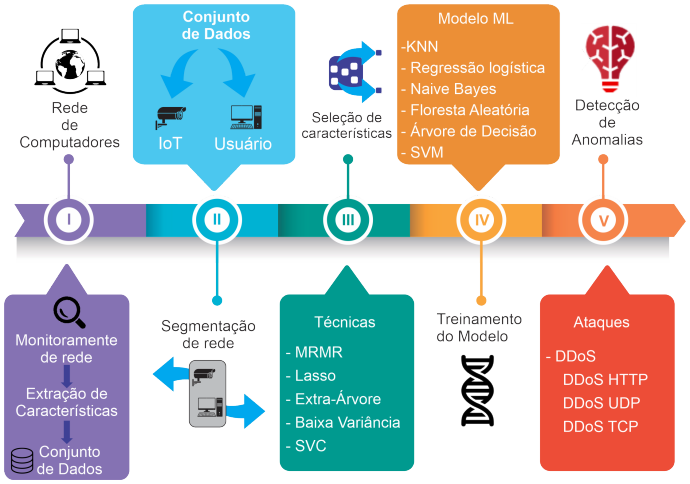

Figura 2. Processamento do fluxo de dados

A seguir, são descritas as etapas do fluxo de processamento dos dados, detalhando suas particularidades, bem como o papel de cada etapa para o funcionamento do sistema inteligente proposto como um todo.

3.1. Extração Características

A partir do monitoramento da rede é possível gerar um conjunto de dados em um formato PCAP, o qual permite a extração de até 80 (oitenta) características de fluxos de rede (usando por exemplo a ferramenta CICFlowMeter [Sharafaldin et al. 2019]). Todavia, o uso de todas essas características pode gerar ruídos que dificultam o treinamento do modelo de Rede Neural do sistema proposto. Portanto, identificar quais características maximizam a capacidade de detecção torna-se uma tarefa crucial. Além disso, quando

somente as características mais relevantes são consideradas, pode-se minimizar no tempo de treinamento do modelo e reduzir a demanda por recursos computacionais (processamento mais rápido, menor consumo de memória e menor espaço de armazenamento), devido a redução de dimensionalidade do problema.

3.2. Segmentação do Fluxo de Rede

A segmentação de rede é uma abordagem para separar o tráfego dos dispositivos IoT e dispositivos pessoais. O sistema inteligente proposto identifica os fluxos de rede de dispositivos IoT a partir de dispositivos pessoais, permitindo o treinamento da técnica de ML utilizando os dados de acordo com a categoria dos dispositivos. Com isso, é possível se adequar melhor ao comportamento de cada tipo e, consequentemente, melhorar a detecção de ataques DDoS.

A segmentação de tráfego usa ML para classificar IoT e dispositivos pessoais com base nos recursos extraídos do monitoramento dos fluxos da rede. Explora as características distintivas dos dispositivos quando se comunicam em um SE. A utilização do ML permite que a segmentação do tráfego seja adaptável aos novos dispositivos do SE, o que ocorre devido à mobilidade e expansão dos serviços nele executados.

Para o desenvolvimento da segmentação de tráfego, usamos um conjunto de dados [1] (desenvolvido por Meidan et al. [Meidan et al. 2018, Koroniotis et al. 2018]) de diferentes dispositivos IoT (como câmeras de cobertura, luzes, plugues, sensores de movimento, dispositivos, monitores de saúde, entre outros).

3.3. Seleção de Características

A redução do número de atributos traz benefícios importantes ao observar os recursos computacionais. Menor quantidade de dados processados significa tempo de treinamento reduzido, menos dados enganosos que melhoram o desempenho do modelo, processamento mais rápido, baixo consumo de memória, extração de informações mais fácil, menor espaço de armazenamento e, principalmente, redução de dimensionalidade. Assim, a seleção adequada das características possibilita otimizar o tempo de treinamento e detecção desses modelos de ML.

Com base nestas circunstâncias, essa pesquisa analisa as seguintes técnicas de seleção de características: (1) Máxima Relevância Mínima Redundância (mRMR) [Peng et al. 2005], esse método usa escores do teste de Fisher e correlação de Pearson; (2) Baixa Variância (BV) [Wood and Asada 2007], remove todas as características cuja variação não atinge determinado limite; (3) Extra-Arvore (EA) [Geurts et al. 2006], constrói um conjunto de árvores de decisão ou regressão não podadas de acordo com o procedimento clássico de cima para baixo; (4) SVC [Chang and Lin 2011], um modelo linear que estima coeficientes esparsos com base em características importantes; e, (5) Lasso [Friedman et al. 2010], um modelo linear que estima coeficientes esparsos.

Qualquer uma dessas técnicas pode ser usada para selecionar as características relevantes do conjunto de dados DDoS que melhora o desempenho dos modelos. No entanto, as diferentes estratégias aplicadas por eles (métodos de filtro, métodos de empacotamento ou métodos embutidos) levam a diferentes características selecionadas [Kaushik

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

2016] (a lista completa das características esta disponível[2]).

3.4. Treinamento do Modelo e Detecção

Após transmitir as informações processadas da Névoa para a Nuvem, o banco de dados de conhecimento é alimentado e usado como base para o treinamento de ML. O treinamento de ML envolve a entrada dos dados no conjunto de dados de conhecimento e a execução da técnica de ML. Posteriormente, o detector (modelo ML treinado) é transmitido e executado na Névoa para identificar possíveis ataques DDoS. O sistema foi projetado para permitir o uso de qualquer uma das técnicas de ML. Esta independência permite que o sistema proposto execute a técnica de ML mais adequada na etapa de treinamento. Assim, avaliamos as técnicas de ML que possuem singularidades distintas: K-vizinho mais próximo (KNN), Naive Bayes (NB), Floresta Aleatória (RA), Árvore de Decisão (AD), Regressão Logística (LR) e Máquinas de Vetores de Suporte (SVM).

O fluxo de processo definido é repetido constantemente para manter o detector atualizado de acordo com o comportamento dos dispositivos no SE. Este processo é cíclico e contínuo e permite o conhecimento de informações recorrentes sobre o ambiente inteligente. Como consequência, a solução proposta é capaz de adaptar e entender o comportamento usual dos fluxos da rede e detectar ataques DDoS.

3.5. Integração entre Computação de Névoa e Nuvem

O fluxo de dados gerado pelo monitoramento da rede precisa ser processado antes da transmissão para a nuvem. Atualmente, a nuvem é o local usual para a execução de serviços. No entanto, com a escala cada vez maior dos fluxos de rede de ambientes inteligentes e, consequentemente, o volume dos fluxos de dados gerados pelos dispositivos, pode criar uma enorme sobrecarga de transmissão para a Internet.

Nos próximos anos, a infraestrutura da Internet enfrentará o desafio de lidar com o aumento da demanda de recursos devido ao fluxo de dados das redes emergentes, atingindo a ordem de petabytes a cada dia. Nesse cenário, abordagens que destinam todas as informações para serem processadas e armazenadas na nuvem são impraticáveis em termos de tempo de comunicação, custo financeiro, degradação do desempenho computacional e consumo de energia. Uma abordagem para lidar com esses problemas é não enviar todos os dados para serem processados pela nuvem (longe da fonte de dados), implantando uma abordagem juntando Névoa e Nuvem. A inclusão da Computação em Névoa entre o ambiente inteligente e a nuvem permite o processamento, comunicação e armazenamento temporário próximo ao ambiente inteligente. Portanto, a Névoa naturalmente aumenta o desempenho, segurança e privacidade no ambiente inteligente, além de reduzir a sobrecarga de dados e a latência.

Além disso, as técnicas de ML aplicadas na detecção de DDoS demandam alto nível de recursos computacionais (processamento paralelo e capacidades de memória). Em geral, esses recursos computacionais não estão disponíveis na Névoa, exigindo vários serviços para suportar redes de acesso. Portanto, a execução de todas as funcionalidades do sistema inteligente proposto no ambiente de Névoa não é viável. Os dados brutos coletados são enviados ao ambiente de Névoa para serem processados. Após essas etapas, os dados brutos se transformam em dados processados, que serão enviados para a Nuvem.

[2]https://github.com/wandersonleo10/pesquisa/blob/master/lista%20de%20caracter%C3%ADsticas.txt

Esse processamento reduz o volume de dados, uma vez que apenas informações úteis são consideradas para serem transmitidas para a Nuvem. Após o processamento, os dados ficam disponíveis no ambiente de Nuvem e são usados como entrada das técnicas de ML para treinar o detector de DDoS. Como etapa final, o detector é implantado no ambiente de Névoa. Assim, os dados processados têm duas funções: (a) alimentar o treinamento de ML na Nuvem e (b) serem testados pelo detector de DDoS na Névoa.

Em resumo, a estrutura projetada do sistema inteligente proposto possibilita duas características importantes: (1) Pequeno *overhead* na infraestrutura de rede, devido ao baixo volume de dados transmitidos entre os ambientes de Névoa e de Nuvem; e, (2) Adequação de execução, uma vez que cada etapa dos módulos é executada no ambiente distinto, ou seja, as técnicas de ML são executadas na nuvem, enquanto o processamento de dados é executado na névoa. Esses dois recursos permitem que o sistema lide com os requisitos de escalabilidade, adaptabilidade e tempo de resposta dos ambientes inteligentes [Pisani et al. 2020].

4. Resultados

Esta seção apresenta os experimentos realizados para avaliar o Sistema Inteligente proposto para detecção de DDoS em ambientes inteligentes. Os experimentos se concentram na avaliação da abordagem projetada de integração entre Névoa e Nuvem, bem como na análise da técnica de ML mais adequada para detectar ataques DDoS.

Os experimentos foram baseados em dois conjuntos de dados, que foram mesclados para representar um SE composto por dispositivos IoT e pessoais heterogêneos. O primeiro é o conjunto de dados "BoT-IoT"[3] desenvolvido por Meidan et al. [Meidan et al. 2018, Koroniotis et al. 2018], que contém o tráfego normal (benigno) e o tráfego relacionado aos últimos ataques DDoS. O último é o "UNSW-IoT"criado por Sivanathan et al. [Sivanathan et al. 2018], que tem tráfego normal (benigno) de IoT e dispositivos pessoais. Ambos os conjuntos de dados são formatados em dados de monitoramento do mundo real (PCAPs).

Em relação ao hardware utilizado nos experimentos, a Névoa executa em uma máquina local com Linux, CPU Intel i7-8700k 4.7GHz e 8GB de memória RAM DDR4, enquanto a Nuvem é uma máquina virtual Azure F48s-V2 com 48 vCPUs de 3.4GHz e 96GB de Memória RAM. Assim, realizamos os experimentos em ambientes de nevoa e nuvem adequados para cenários realistas.

O desempenho do sistema inteligente proposto (incluindo a combinação das técnicas de seleção e ML) considerou as seguintes métricas de avaliação:

- Acurácia (em porcentagem):
  Taxa de classificações corretas de acordo com a Equação 1, ou seja, os casos Verdadeiro Positivo (VP) e Verdadeiro Negativo (VN) em relação a todos os outros casos (VP, VN, Falso Positivo - FP e Falso Negativo - FN). É importante notar que a acurácia foi medida para a Segmentação de Tráfego e detecção de DDoS.

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

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

- **Tempo de treinamento (em segundos)**: tempo necessário para treinar o detector DDoS (modelo ML) com as características de entrada selecionadas.
- **Tempo de detecção (em segundos)**: tempo gasto pelo detector de DDoS para definir se um caso é um ataque DDoS ou não.
- **Volume de Dados (em Gigabyte/Megabytes)**: o tamanho dos dados gerados (dados processados) a serem trocados entre Névoa e Nuvem.

4.1. Segmentação de tráfego

Nesta seção, avaliamos a capacidade das técnicas de ML utilizadas para identificar os dispositivos IoT na rede, apresentadas na Tabela 2. É possível notar que o AD e FA alcançam os melhores resultados, atingindo uma acurácia próxima a cem por cento.

Tabela 2. Segmentação de tráfego

ML	KNN	RL	AD	NB	FA	SVM
Acurácia	86.65	66.50	99.49	64.79	99.91	91.30

Esses melhores resultados de AD e FA ocorrem devido à sua natureza de dividir o problema em vários estágios. Desta forma, a dupla possibilidade de classificação (IoT ou dispositivo pessoal) facilita a divisão do problema, melhorando a organização das folhas e estrutura da árvore de classificação desenhada.

4.2. Acurácia

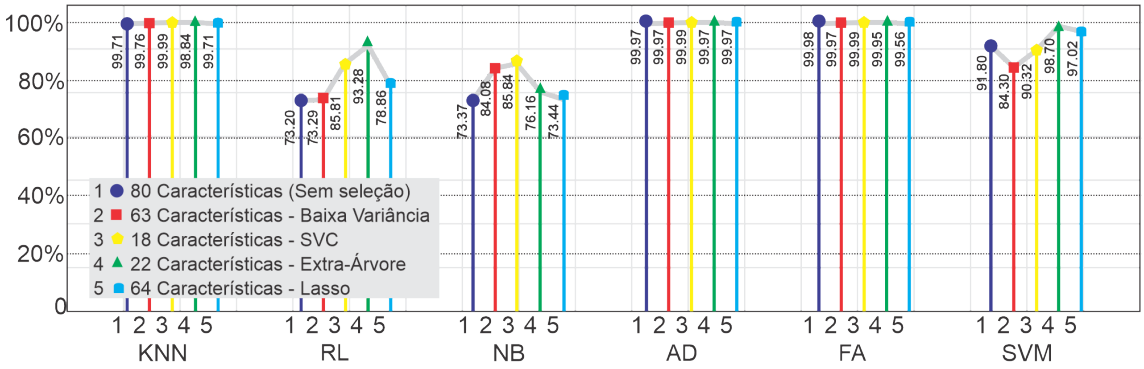

Figura 3. Acurácia para Detecção de DDoS

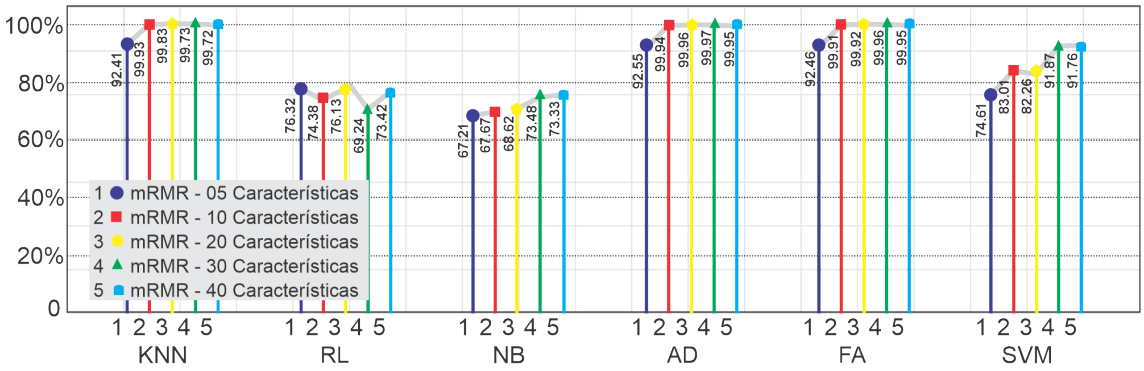

Figura 4. Acurácia mRMR

A acurácia foi dividida em duas figuras, 4 e 3, para facilitar a visualização dos resultados, onde a Tabela 4 mostra a acurácia do ML técnicas usando os casos de mRMR e Tabela 3 as combinações restantes.

A partir dos resultados apresentados nas figuras, pode-se verificar que a precisão das técnicas de ML varia de acordo com a técnica de seleção aplicada, principalmente quando essas técnicas de ML são baseadas em abordagens que focam na dimensionalidade, como os classificadores KNN, RL e SVM. Por outro lado, as técnicas de ML baseadas na divisão de subconjuntos (AD e FA) quase não têm impacto pela variação nas técnicas de seleção. Isso ocorre devido ao processo de derivação recursiva dos subconjuntos, mitigando a variação nas características selecionadas que podem resultar em possíveis ruídos para o treinamento de ML.

Em relação ao desempenho de NB e RL, ambas as técnicas de ML apresentam piores resultados que as demais abordagens, independentemente da técnica de seleção. Assim, NB e RL aparecem como soluções inadequadas para detecção de DDoS em SEs quando comparados às demais abordagens do experimento.

Tabela 3. Tempo de treinamento (em segundos)

Técnicas	KNN	RL	NB	AD	FA	SVM
80 Características	13.29	2.34	0.53	1.96	22.99	1625.04
BV	11.74	2.14	0.41	1.88	10.10	273.34
SVC	36.30	1.27	0.21	0.28	5.27	1226.96
Extra-Árvore	18.55	2.96	0.19	0.68	13.33	351.01
Lasso	11.91	1.20	0.14	0.74	6.73	304.38
mRMR 05	21.78	2.91	0.13	0.16	4.11	1055.70
mRMR 10	16.01	3.43	0.14	0.38	5.45	708.88
mRMR 20	10.67	3.42	0.30	0.81	10.12	1177.78
mRMR 30	7.91	0.93	0.29	0.49	10.68	385.93
mRMR 40	9.16	1.23	0.28	1.19	15.53	467.02

Tabela 4. Tempo de detecção (em segundos)

Técnicas	KNN	RL	NB	AD	FA	SVM
80 Características	9.02	0.02	0.53	0.02	0.53	162.57
BV	7.98	0.03	0.03	0.01	0.34	15.24
SVC	15.62	0.02	0.02	0.01	0.35	144.13
Extra-Árvore	12.95	0.01	0.06	0.02	0.44	21.15
Lasso	8.15	0.01	0.14	0.01	0.45	142.08
mRMR 05	16.29	0.02	0.01	0.01	0.48	146.50
mRMR 10	12.43	0.02	0.01	0.01	0.53	142.84
mRMR 20	7.72	0.01	0.02	0.01	0.58	198.56
mRMR 30	5.46	0.02	0.03	0.01	0.49	86.29
mRMR 40	6.26	0.05	0.04	0.01	0.52	102.73

As tabelas 3 e 4 mostram o tempo gasto para realizar o treinamento do modelo de ML (criando o detector de DDoS) e para os detectores identificarem os casos de ataques DDoS, respectivamente. Os resultados apresentados em ambas as tabelas representam a viabilidade das técnicas de ML serem implantadas em contextos distintos de SEs.

Com base nos resultados apresentados na Tabela 3, os classificadores KNN e FA e, principalmente, SVM possuem um tempo de treinamento superior às demais abordagens.

No entanto, a aplicação da técnica de seleção mRMR (com 5 e 10 características) reduz o tempo de treinamento do classificador FA, possibilitando sua implantação para SEs, chegando a um tempo mais próximo do classificador RL.

Da mesma forma que o tempo de treinamento, o tempo de detecção (apresentado na Tabela 4) dos classificadores KNN e SVM é maior do que as outras técnicas de ML. Porém, diferentemente do tempo de treinamento, o impacto das técnicas de seleção é menor. Em geral, as técnicas RL, NB e AD gastam muito pouco tempo para realizar a detecção. Perto deles está a FA, provando ser uma solução viável também.

4.3. Volume de dados

No sistema proposto, após a extração das características de fluxo, 80 atributos são criados e posteriormente esses atributos são selecionados por uma técnica específica. Assim, o fluxo de processamento de dados tende a reduzir o volume de dados a serem transmitidos da Névoa para a Nuvem. A tabela 5 mostra os resultados do volume de dados gerados em cada etapa do fluxo de processamento de dados usando o conjunto de dados descrito na Seção 3.

Tabela 5. Volume de dados brutos (PCAP) e dados processados

Técnica	Volume de dados
Dados bruto	15.16GB
Extração (80 características)	4.17GB
BV (63 características)	88MB
SVC (18 características)	21MB
EA (22 características)	22MB
Lasso (64 características)	84MB
mRMR (5 características)	5MB
mRMR (10 características)	9MB
mRMR (20 características)	22MB
mRMR (30 características)	37MB
mRMR (40 características)	50MB

Quando a seleção de características ocorre na Névoa, a quantidade de dados chega a reduzir de 15,16 GB (brutos) para aproximadamente 25MB (processados) em média, representando menos de $0,2\%$ do volume de informações. Assim, a abordagem de integração Névoa e Nuvem aumenta a escalabilidade da rede, enquanto causa um impacto muito baixo na disponibilidade dos recursos da rede.

4.4. Discussão Final

Os resultados dos experimentos destacam a importância da seleção de características para a acurácia, tempo de execução e volume de dados. Por exemplo, usando a técnica de seleção mais apropriada, o desempenho dos classificadores KNN e SVM aumenta em 8% e 7%, respectivamente. Além disso, a técnica RL usando os 80 atributos extraídos (sem seleção) tem uma acurácia inaceitável, enquanto usando a técnica Extra-Arvore, ela atinge mais de 93% de acurácia.

Considerando-se o tempo de treinamento, aumenta a sua importância em contextos em que é necessário um treino recorrente para atualizar o modelo de ML em virtude da alta dinâmica dos SEs, como cidades inteligentes. Assim, o modelo de ML será treinado

em um período muito curto de tempo para manter a detecção de ataques DDoS de forma eficaz. O mesmo raciocínio pode ser aplicado ao tempo de detecção. Nesse contexto, o AD e FA com mRMR-10, mRMR-20, Lasso ou SVC são as combinações adequadas, pois são rápidos, têm alta acurácia e geram pequeno volume de dados. Por outro lado, se a periodicidade do treinamento for maior, devido ao comportamento estático do SE (como uma indústria inteligente), outras abordagens são viáveis.

Outro ponto importante a ser destacado é a redução do volume de dados a ser transferido da Névoa para a Nuvem. Em se tratando do volume total de dados gerado pela monitoramento da rede torna-se inviável para o envio de aproximadamente 15GB da fonte dos dados para a nuvem, todavia, ao utilizar a técnica de seleção de característica adequada, por exemplo, a SVC, é possível chegar a uma redução em até 99% do volume real a ser analisado na rede, apresentada a importância do pré-processamento (seleção de características) próximo a fonte de dados (Névoa). Logo a arquitetura baseada em Névoa e Nuvem torna-se indispensável para a implementação da proposta.

5. Conclusão

A crescente inclusão de dispositivos IoT em espaços urbanos vem tornando estes mais inteligentes e heterogêneos, visto que além dos dispositivos IoT há os dispositivos pessoais dos usuários. Simultaneamente, vem aumentando o risco desses ambientes tornarem-se ferramentas de execução de ataques DDoS. A partir desta realidade, este artigo apresentou um sistema inteligente para segmentação de tráfego de rede e detecção de ataques DDoS utilizando técnicas de IA em conjunto com Computação em Névoa e em Nuvem.

Os experimentos realizados usando um conjunto de dados real apresenta que o sistema proposto atinge seu objetivo de forma satisfatória, com alta acurácia de detecção e com alto desempenho (tempo de treinamento e detecção baixo, bem como volume de dados gerados baixo). Como trabalhos futuros, pretende-se evoluir o sistema para agregar a funcionalidade de identificação de outros tipos de ataques, como por exemplo os ataque de canais laterais.

Referências

Ahmed, E., Yaqoob, I., Gani, A., Imran, M., and Guizani, M. (2016). Internet-of-things-based smart environments: state of the art, taxonomy, and open research challenges. *IEEE Wireless Communications*, 23(5):10–16.

Andrea, I., Chrysostomou, C., and Hadjichristofi, G. (2015). Internet of things: Security vulnerabilities and challenges. In *2015 IEEE Symposium on Computers and Communication (ISCC)*, pages 180–187.

Brun, O., Yin, Y., Augusto-Gonzalez, J., Ramos, M., and Gelenbe, E. (2018). Iot attack detection with deep learning. In *ISCIS Security Workshop*.

Chang, C.-C. and Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27.

Diro, A. A. and Chilamkurti, N. (2018). Distributed attack detection scheme using deep learning approach for internet of things. *Future Generation Computer Systems*, 82:761–768.

Doshi, R., Apthorpe, N., and Feamster, N. (2018). Machine learning ddos detection for consumer internet of things devices. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 29–35. IEEE.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1.

Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, 63(1):3–42.

Hamamoto, A. H., Carvalho, L. F., Sampaio, L. D. H., Abrão, T., and Proença Jr, M. L. (2018). Network anomaly detection system using genetic algorithm and fuzzy logic. *Expert Systems with Applications*, 92:390–402.

Kaushik, S. (2016). Introduction to feature selection methods with an example (or how to select the right variables?). *Analytics Vidhya*.

Koroniotis, N., Moustafa, N., Sitnikova, E., and Turnbull, B. (2018). Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *CoRR*, abs/1811.00701.

Li, H., Ota, K., and Dong, M. (2018). Learning iot in edge: deep learning for the internet of things with edge computing. *IEEE Network*, 32(1):96–101.

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., and Elovici, Y. (2018). N-baiot—network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, 17(3):12–22.

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., and Elovici, Y. (2018). N-baiot—network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, 17(3):12–22.

Peng, H., Long, F., and Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238.

Pisani, F., de Oliveira, F. M. C., Gama, E. S., Immich, R., Bittencourt, L. F., and Borin, E. (2020). Fog computing on constrained devices: Paving the way for the future iot.

Sharafaldin, I., Lashkari, A. H., Hakak, S., and Ghorbani, A. A. (2019). Developing realistic distributed denial of service (ddos) attack dataset and taxonomy. In *2019 International Carnahan Conference on Security Technology (ICCST)*, pages 1–8. IEEE.

Sivanathan, A., Gharakheili, H. H., Loi, F., Radford, A., Wijenayake, C., Vishwanath, A., and Sivaraman, V. (2018). Classifying iot devices in smart environments using network traffic characteristics. *IEEE Transactions on Mobile Computing*, 18(8):1745–1759.

Wood, L. B. and Asada, H. H. (2007). Low variance adaptive filter for cancelling motion artifact in wearable photoplethysmogram sensor signals. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 652–655. IEEE.

Yamauchi, M., Ohsita, Y., Murata, M., Ueda, K., and Kato, Y. (2019). Anomaly detection for smart home based on user behavior. In *2019 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–6. IEEE.