# Análise de Características do Tráfego de Rede para Detecção de Ataques DDoS em Ambientes IoT

Wanderson Leonardo Costa
Instituto Federal do Piauí (IFPI)
Teresina-PI, Brasil
wanderson.leonardo@ifpi.edu.br

Ariel Lima de Carvalho Portela
Universidade Estadual o Ceará (UECE)
Fortaleza-CE, Brasil
ariel.portela@aluno.uece.br

Rafael Lopes Gomes
Universidade Estadual o Ceará (UECE)
Fortaleza-CE, Brasil
rafaellgom@larces.uece.br

## ABSTRACT

The evolution of computing devices has allowed the evolution of service provision in society, applying new technologies based on the Internet of Things (IoT). Most IoT devices have security vulnerabilities, making them susceptible to Distributed Denial of Service (DDoS) Attacks. Thus, it is necessary to apply solutions that can detect this type of attack in IoT networks from the information of the network traffic. However, there is still no definition of which traffic characteristics should be used for detection, since the use of inappropriate characteristics tend to make detection difficult. Within this context, this article presents an analysis of the most important traffic characteristics for detecting DDoS in IoT networks, in order to support a detection mechanism based on Machine Learning. Experiments using a real data set suggest that the proposed mechanism has an accuracy close to 99 % when the most suitable characteristics are selected.

## KEYWORDS

Ambientes Inteligentes, Machine Learning, Tráfego, Detecção, DDoS, IoT

## 1 INTRODUÇÃO

Na sociedade moderna os usuários estão cercadas de dispositivos de comunicação que trocam informações periódicas via tecnologias sem fio. Esses usuários têm usado cada vez mais dispositivos com funcionalidades variadas a fim de evoluir a execução dos mais diversos serviços presentes no cotidiano da sociedade, tais como automação de tarefas, monitoramento de ambientes, sistemas de vigilância, dentre outros. Esta nova realidade é chamada de era da Internet das Coisas (*Internet of Things* - IoT) [1].

A principal característica de uma rede IoT é a heterogeneidade, sendo esta em diversos aspectos: (I) Comunicação, múltiplas tecnologias de transmissão podem ser utilizadas em um mesmo ambiente (tal como Zigbee, Wifi, Bluetooth, dentre outras); (II) Hardware dos Dispositivos, a capacidade computacional (nível de processamento, disponibilidade de memória, volume de armazenamento e disponibilidade de energia) dos dispositivos varia de acordo com seu objetivo (sensores, câmeras de vigilância, etc) ; e, (III) Serviços, as aplicações que atuam sobre o ambiente em questão possuem diferentes requisitos e funcionalidades (por exemplo, um serviço de monitoramento de temperatura possui um volume de tráfego muito inferior em comparação com vigilância por vídeo).

À medida em que redes IoT se popularizam, aumenta o risco desses dispositivos tornarem-se alvos de ataques cibernéticos que afetam diretamente a Qualidade de Serviço (*Quality of Service* - QoS) das aplicações executantes sobre a rede IoT. Um dos ataques mais onerosos existentes são os Ataques de Negação de Serviço Distribuídos (*Distributed Denial of Service* - DDoS) [2] que visam indisponibilizar o acesso a um ou mais alvos ao esgotar seus recursos usando múltiplas requisições ilegítimas.

A detecção de ataques DDoS em redes IoT é um desafio existente, visto que as limitações de hardware dos dispositivos impedem a implantação de soluções de segurança que executem nestes. Desta forma, é necessário implantar mecanismos de detecção que atuem externamente ao dispositivos, utilizando somente os dados relacionados ao tráfego de rede. Uma abordagem promissora para realizar esta detecção de DDoS em redes IoT é a aplicação de técnicas de Aprendizado de Máquina (*Machine Learning* - ML), as quais aprendem o comportamento dos dados disponibilizados e melhoram o entendimento sobre eles progressivamente [3]. Contudo, é necessário utilizar as características mais relevantes do tráfego de rede para, posteriormente, treinar o mecanismo de detecção de ataques DDoS utilizando ML. A análise das características do tráfego da rede IoT é crucial para o desempenho do mecanismo de detecção, visto que a seleção inadequada de características afeta diretamente a acurácia das técnicas de ML.

Dentro desse contexto, esse artigo apresenta um Mecanismo de Detecção de Ataques DDoS em Redes IoT utilizando técnicas de ML. A fim de extrair o melhor desempenho possível de cada uma das técnicas de ML, foi realizado uma análise das características do tráfego de rede mais relevantes e o impacto da seleção dessas no desempenho das técnicas de ML. Portanto, o mecanismo proposto identifica a seleção de característica mais adequada para cada técnica de ML que pode ser aplicada.

Experimentos foram realizados usando um conjunto de dados de tráfego de rede IoT real com ataques DDoS, onde avaliou-se a aplicação das técnicas *K-Nearest Neighboors* (KNN), Regressão Logística, Naive Bayes, Floresta Aleatória, Árvore de Decisão e *Support Vector Machines* (SVM). Os resultados sugerem que a seleção de características pode impactar em até 70% a acurácia das técnicas de ML, bem como o tempo de processamento para detecção em dez vezes.

Sendo assim, o artigo possui as seguintes contribuições: (A) Extrair de forma padronizada as características de tráfego do conjunto de dados, a fim de realizar a rotulagem das classes; (B) Realização de um estudo para a análise detalhada das principais características para a detecção de DDoS em redes IoT; e, (C) Desenvolvimento de um mecanismo de detecção de DDoS a partir da seleção das características mais relevantes para cada técnica de ML aplicada.

O restante deste artigo é organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados existentes na literatura. A seção 3 detalha o mecanismo proposto, enquanto que a Seção 4

detalha os experimentos realizados e discute os resultados obtidos. Por fim, a Seção 5 conclui o artigo e apresenta os trabalhos futuros.

## 2 TRABALHOS RELACIONADOS

A seguir, apresentamos os trabalhos relacionados à estratégias para detecção de ataques DDoS em redes IoT e redes tradicionais. Além disso, a Tabela 1 resumo os pontos dos trabalhos analisados, destacando o diferencial deste trabalho em relação aos trabalhos em questão.

Vinayakumar et al. [4] propõem um sistema de detecção de botnet baseado em uma estrutura de ML de dois níveis para distinguir semanticamente botnets de comportamentos legítimos na camada de aplicação dos serviços de sistema de nomes de domínio DNS. No primeiro nível são utilizadas pontuações para definir a similaridade, ao atingir uma diferença estabelecida pelos autores o nome de domínio é passado para o segundo nível que utiliza uma arquitetura de aprendizado profundo para detectar e classificar as ocorrências de DDoS. Este trabalho foca na detecção de ataques DDoS exclusivamente em servidores DNS, inviabilizando sua aplicação em outros tipos de serviço de redes IoT.

Sharafaldin et al. [2] 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 comunicam 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. [5] 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 comparando a sequência atual com as sequências aprendidas para a condição correspondente à condição atual. Caso apresente alguma alteração pré-definida, 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.

Sivanathan et al. [6] criaram um conjunto de dados de tráfego de rede em um ambiente com vinte e oito dispositivos IoT diferentes durante seis meses. A partir desse conjunto de dados os autores usaram atributos estatísticos (número de portas usados, padrões de sinalização e atividades cíclicas) para treinar um modelo de ML para identificar o tráfego oriundo de dispositivos IoT. Desta forma, o modelo ML criado foca em identificar dispositivos IoT, sem a capacidade de detecção os ataques DDoS.

Doshi et al. [3] desenvolveram um sistema para a detecção de ataques DDoS em dispositivos IoT utilizando técnicas de ML. Neste trabalho, os autores usam um número limitado de características dos dispositivos para modelar as técnicas de ML para detecção, como por exemplo número limitado de terminais e intervalos regulares de tempo entre pacotes. Assim, a abordagem de seleção aplicada é muito limitada quando comparado com a análise realizada para desenvolver o mecanismo proposto neste artigo.

Haddadpajouh et al. [7] apresentaram um modelo do aprendizado profundo de Rede Neural Recorrente (RNN) para detecção de Malwares em dispositivos IoT. Os autores usam RNN para analisar os códigos de operação de execução de aplicativos de IoT baseados em ARM (OpCodes), criando um vetor de recurso com base nos OpCodes para cada amostra. A partir disso, utiliza-se esses dados vetoriais para treinamento e ajuste de rede neural profunda para parâmetros ótimos. No entanto, o modelo treinado depende da análise dos OpCodes, limitando sua capacidade de detectar malwares de código aberto, como por exemplo o Mirai Botnet.

A partir do levantamento bibliográfico realizado, nota-se que nenhum artigo da literatura se concentrou no desenvolvimento de uma abordagem de detecção de ataques DDoS em redes IoT baseado em ML considerando uma análise robusta de características do tráfego de rede, que é o foco deste artigo. Os trabalhos encontrados na literatura focados em detecção de DDoS possuem uma abordagem incompatível com redes IoT e/ou aplicam técnicas de ML sem um embasamento adequado de seleção de características.

## 3 PROPOSTA

A atual necessidade de desenvolver soluções de segurança para as redes IoT, considerando as restrições dos dispositivos IoT, vem se tornando cada vez mais necessárias. Dentre essas necessidades, uma das mais importantes é a detecção de ataques DDoS. A partir disso, este artigo propõe um mecanismo de detecção de DDoS usando técnicas de aprendizagem de máquina (ML). Para tal, é necessário selecionar uma técnica e treinar o modelo de ML.

No contexto deste artigo, redes IoT, utilizamos os dados do tráfego de rede para extrair informações sobre esses dados e utilizalas como entrada para o treinamento do modelo ML mencionado. Contudo, a tarefa de extrair informações gera um impacto muito grande no treinamento do modelo e, consequentemente, na sua acurácia para detectar os ataques DDoS. A utilização do máximo de características que se pode extrair não necessariamente resulta em um ganho de desempenho, visto que muitas características extraídas podem atuar como ruído, comprometendo o treinamento do modelo de ML. Portanto, a seleção adequada de características é um aspecto crucial para garantir a eficiência do mecanismo proposto.

O processo de desenvolvimento do mecanismo proposto seguiu seis fases: (I) Características dos Ataques DDoS; (II) Conjunto de Dados (contem todos os dados sobre o tráfego de rede); (III) Extração de Características; (IV) Seleção de Características; (V) Treinamento do Modelo; e, (VII) Detecção de DDoS. O fluxo de execução dessas fases é ilustrado na Figura 1.

Primeiramente entende-se o funcionamento dos ataques DDoS e em conjunto com o conjunto de dados sobre o tráfego de rede, extrai-se as características possíveis desse conjunto de dados. Em posse de todas essas informações, aplica-se alguma técnica para a seleção de características. Essas características selecionadas são usada como entrada para o treinamento do modelo de ML. Após o treinamento, o classificador gerado atua na detecção de ataques DDoS.

Durante as fases de seleção de características e treinamento do modelo, diversas técnicas podem ser utilizadas. A partir disso, durante o desenvolvimento do mecanismo proposto foram analisadas

**Table 1: Trabalhos relacionados**

| Referência | Contexto | Foco |
|---|---|---|
| Vinayakumar et al. [4] | DNS | Duas camadas de ML para detecção Botnet. |
| Sharafaldin et al. [2] | Redes TCP/IP | Experimentos reais para Análise de características. |
| Yamauchi et al. [5] | SHs | Identificação de Anomalias baseado no comportamento. |
| Sivanathan et al. [6] | IoT | Identificar dispositivos IoT. |
| Doshi et al. [3] | IoT | Detecção DDoS. |
| Haddadpajouh et al. [7] | IoT | RNN para Detecção malwere. |
| Este artigo | IoT | Análise de Características para Detecção de DDoS. |

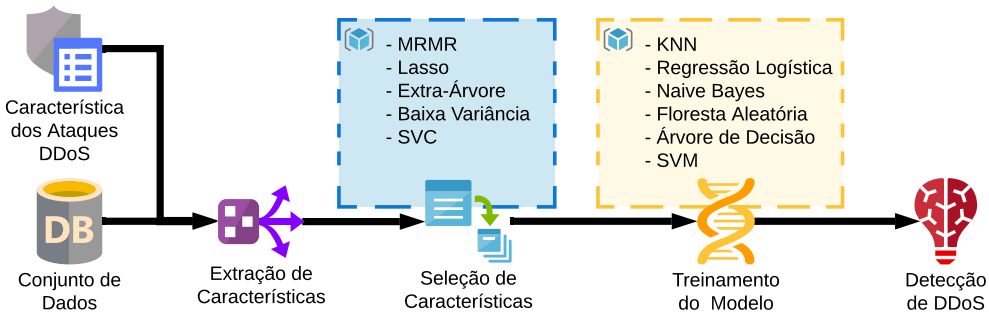



**Figure 1: Etapas de Desenvolvimento do Mecanismo Proposto**

e avaliadas diversas técnicas existentes a serem aplicadas no contexto deste trabalho.

A seguir são apresentados detalhes de cada uma das fases executadas do mecanismo de detecção de DDoS proposto, destacando suas particularidades, bem como o papel de cada fase para o funcionamento do mecanismo como um todo.

## 3.1 Característica dos Ataques DDoS

O DDoS tornou-se muito frequente com o crescimento das redes IoT. Por meio desse tipo de ataque, um agente malicioso mestre escraviza vários dispositivos e, de maneira organizada, faz esses dispositivos congestionarem um determinado alvo. Este agente malicioso mestre pode gerenciar até milhões de dispositivos IoT infectados simultaneamente [1].

Os ataques DDoS podem ser classificados de acordo com a camada envolvida [2]: (i) Aplicação, tentam invadir os serviços executantes nas redes IoT, onde os pacotes são descartados na taxa de solicitação por segundo; e, (ii) Infraestrutura, visam tornar o sistema de destino inacessível, explorando as vulnerabilidades presentes na camada de transporte ou rede da arquitetura IoT que podem ser de dois tipos baseados em protocolo e/ou em volume.

A partir dessas características dos ataques DDoS e sua compreensão, percebe-se que para detectar esses ataques faz-se necessário utilizar informações das mais diversas camadas nos pacotes de forma individual (protocolos utilizados, endereço dos dispositivos, flags de sinalização, etc), bem como as informações resultantes da análise do conjunto de pacotes trafegando pela rede (volume de dados, fluxos ativos, dentre outros).

## 3.2 Conjunto de Dados

O conjunto de dados utilizado neste trabalho foi o BoT-IoT[1], desenvolvido por Meidan et al. [8, 9]. Este conjunto de dados contém tanto tráfego normal (benígno) quanto tráfego relacionado ao mais recentes ataques DDoS, todos formatados em dados do mundo real (PCAPs), tais como: DDoS, DoS, OS Service Scan, Keylogging e Data Exfiltration.

Esta variedade de ataques DDoS, bem como sua formatação (seguindo aplicações de monitoramento reais), possibilitam a extração de todas as características necessárias para realizar o processo de seleção de características e posteriormente o treinamento do modelo de ML. Além disso, este conjunto já apresenta a rotulagem do dados, ou seja, já indica quais dados são referentes a tráfego benigno e quais dados são relacionados aos ataques DDoS.

## 3.3 Extração de características

A partir do conhecimento sobre os ataques DDoS e o conjunto de dados utilizados pode-se extrair as características do tráfego de rede de forma padronizada em relação as informações individuais dos pacotes e dos fluxos de rede. Neste trabalho foram extraídas 80 (oitenta) características de tráfego de rede do conjunto de dados usando a ferramenta CICFlowMeter [2]. O CICFlowMeter, é um extrator de características baseado em fluxo de rede e pode extrair características diretamente de arquivos de dados brutos em formato PCAP.

Dentre as características extraídas destaca-se: protocolos utilizados (camada de aplicação, rede e enlace), as flags de sinalização

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

no cabeçalho dos pacotes, o tamanho dos pacotes (total e somente payload), volume tráfego de um fluxo (entrada e saída), intervalo entre pacotes de um fluxo, tempo de inatividade de um fluxo, etc. A lista completa das 80 características extraídas foi omitida neste trabalho por limitação de espaço.

Todas as características extraídas representam algum aspecto existente nas redes IoT. Contudo, a utilização deste grande número de características pode resultar em certos ruídos que dificultam a etapa de treinamento do modelo de ML. Além disso, esses ruídos afetam de maneira singular cada uma das técnicas existentes, ou seja, uma certa característica extraída pode comprometer o desempenho de uma técnica de ML e de outra não. Sendo assim, faz-se necessário selecionar as características mais relevantes, a fim de maximizar o desempenho dos classificadores a realizar a tarefa de detecção de ataques DDoS.

## 3.4 Seleção de Características

A seleção de características para análise de tráfego de rede é um desafio para especialistas que visam construir sistemas que descobrem padrões de comportamento. Este processo se torna mais complexo ainda quando se trata de ataques de DDoS devido a variedade de tipos, bem como a complexidade do sincronismo de ação do mesmo. O objetivo da seleção de características é habilitar a construção de modelos de ML que viabilizem entender os dados e maximizar a capacidade de detecção. Portanto, a seleção de características ajuda a conhecer atributos irrelevantes e redundantes que podem ter impacto negativo no desempenho do modelo diminuindo a acurácia do modelo.

Adicionalmente, reduzir o número de características traz benefícios importantes quando observado recursos computacionais. Menos dados significa redução no tempo de treinamento, menos dados enganosos que melhoram o desempenho do modelo, processamento mais rápido, menor consumo de memória, extração de dados mais fácil, menor espaço de armazenamento e, principalmente, redução de dimensionalidade. Sendo assim, a seleção de características adequadas possibilita otimizar o tempo para treinamento e detecção desses modelos de ML.

A partir desta realidade, este artigo analisa as seguintes técnicas de seleção de características: (1) Máxima relevância Mínima Redundância (MRMR), (2) Baixa Variância (BV), (2) Extra-Árvore (EA), (4) SVC e (5)Lasso. Qualquer uma dessas técnicas pode ser usada para selecionar as características relevantes do conjunto de dados de DDoS que melhora o desempenho dos modelos. Contudo, devido as diferentes estratégias aplicadas por estas (métodos de filtro, métodos de embrulho ou métodos incorporados) levam a diferentes características selecionadas [10]. As subseções a seguir irão descrever cada uma dessas técnicas citadas:

- Máxima relevância, Mínima Redundância (MRMR) [11]: A técnica MRMR executa uma exploração sequencial, adicionando iterativamente recursos a um conjunto considerando as características mais relevantes, assim os seguintes aspectos são avaliados: o novo atributo deve conter a máxima relevância em relação ao rótulo e uma redundância mínima em relação a o subconjunto de recursos já selecionado. A relevância e redundância são calculadas usando as pontuações do teste de Fisher e correlação de Pearson.
- Lasso [12]: O Lasso é um modelo linear que estima coeficientes esparsos. Este é comumente aplicado em alguns contextos devido à sua tendência de preferir soluções com menos coeficientes diferentes de zero, reduzindo efetivamente o número de características dos quais a solução fornecida depende. Portanto, ele consiste em um modelo linear com um termo de regularização adicionado.
- Extra-Árvore (EA) [13]: O algoritmo Extra-Árvore constrói um conjunto de árvores de decisão ou regressão não podadas de acordo com o procedimento clássico de cima para baixo. Suas duas principais diferenças com outros métodos de conjuntos baseados em árvores são que ele divide os nós escolhendo pontos de corte completamente aleatoriamente e usa toda a amostra de aprendizado para cultivar as árvores. Essa classe implementa um meta-estimador que se encaixa em várias árvores de decisão aleatórias em várias subamostras do conjunto de dados e usa a média para melhorar a precisão preditiva e controlar o ajuste excessivo.
- Baixa Variância (BV): Tradicionalmente, a variância mede o espalhamento das amostras de um conjunto de dados. Dessa forma, uma baixa variância indica que os valores do conjunto estão aglomerados com proximidade uns dos outros. Logo, a alta variância, por sua vez, indica que as amostras estão mais espalhadas. A partir disso, a técnica de baixa variância remove todas as características cuja variação não atinja um limite determinado, ou seja, inicia removendo todas as características com variação zero (características que tem o mesmo valor em todas as amostras) até que atinja o valor de 80%. Logo, as características que não apresentarem uma variação de valores nas amostras de mais de 80% são excluídas.
- Vetores de Suporte Linear (SVC) [14]: Classificação de vetores de suporte linear (*Support Vector Classification* - SVC) é um modelo linear que estima coeficientes esparsos baseado em características de importância assim como o modelo Lasso. Os modelos lineares penalizados com a norma L1 têm soluções esparsas: muitos de seus coeficientes estimados são zero. Quando o objetivo é reduzir a dimensionalidade dos dados a serem usados com outro classificador, por isso, tem mais flexibilidade na escolha de funções de penalidades e perdas e deve ser dimensionado melhor para um grande número de amostras.

## 3.5 Treinamento do Modelo e Detecção de Ataques DDoS

Após a seleção das características mais adequadas, é iniciada a fase de treinamento do modelo de ML. O treinamento do modelo de ML engloba a recepção dos dados execução da técnica de ML. Cada técnica de ML aplica uma abordagem distinta para compreender os dados e detectar os ataques DDoS. Então, o mecanismo de detecção proposto é independente de um técnica de ML específica. Como resultado do treinamento, tem-se um classificador que analisa os dados de entrada e detecta os dispositivos participantes de um ataque DDoS. Portanto, nesta fase recebe-se as características selecionadas em conjunto com os rótulos do conjunto de dados (como descrito na Seção 3.2) e os casos detectados como ataques DDoS gerão gatilhos,

que podem ser alertas para o administrador de rede, implantação de regras em firewall, dentre outras ações de proteção.

Devido a independência do mecanismo proposto de técnica de ML existente, são consideradas nesse trabalho as seguintes técnicas: KNN, Naive Bayes, Floresta Aleatória, Árvore de Decisão, Regressão Logística e SVM. Estas técnicas de ML foram escolhidas devido as suas singularidades em relação as demais. A seguir cada uma delas é detalha:

- Nearest Neighbor (KNN) [15]: O KNN é um dos algoritmos não-paramétricos mais importantes no campo de reconhecimento de padrões, sendo um algoritmo de classificação de aprendizado supervisionado. As regras de classificação do KNN são geradas pelas próprias amostras de treinamento sem nenhum dado adicional. O algoritmo de classificação KNN prevê a categoria da amostra de teste de acordo com as amostras de treinamento que são os vizinhos mais próximos da amostra de teste, e a julga para aquela categoria que possui a maior probabilidade de categoria. O vizinho mais próximo refere-se ao vetor de característica multidimensional que é usado para descrever a amostra mais próxima, e o critério mais próximo pode ser a distância euclidiana do vetor de característica.
- Naive Bayes (NB) [16]: O classificador Naive Bayes é baseado em uma distribuição gaussiana, ou seja, é um classificador probabilístico de baixa complexidade que segue o Teorema de Bayes. Apesar do fato de que a independência dos atributos de uma instância é irreal, o classificador Naive Bayes é eficaz na prática, uma vez que sua decisão de classificação pode estar correta, mesmo que suas estimativas de probabilidade sejam imprecisas. Esta abordagem probabilística faz do classificador Naive Bayes um dos mais "leves", ou seja, necessita de poucos recursos computacionais e realiza a classificação dos dados de forma rápida.
- Arvore de decisão (AD) [17]: Árvore de decisão é um modelo de tomada de decisão de vários estágios, ou seja, uma abordagem que divide uma decisão complexa em várias decisões mais simples, implementando um gráfico semelhante a uma árvore para esquematizar e representar o processo de tomada de decisão. Para cada nó da árvore corresponde a uma das variáveis de entrada e é dividido em nós filhos com base nos valores da variável de entrada. Cada nó folha ou terminal representa o valor específico de uma variável de destino, por exemplo, a classe específica de uma variável categórica para o problema de classificação. Durante o processo de treino as amostras em cada nó são divididas em subconjuntos com base em um atributo, e esse processo é repetido em cada subconjunto derivado de uma maneira recursiva. A recursão é concluída quando um subconjunto em um nó tem o mesmo valor de destino ou quando a divisão não melhora a previsão.
- Floresta aleatória (FA) [18]: As florestas aleatórias são uma combinação de preditores de árvores, de modo que cada árvore depende dos valores de um vetor aleatório de experimentos independentemente e com a mesma distribuição para todas as árvores da floresta. À medida que o número de florestas cresce o erro de generalização converge para um limite. FA é uma ferramenta eficaz na previsão e quando definido o tipo certo de aleatoriedade o classificador torna-se mais preciso. Uma das vantagens da floresta aleatória é que a variação do modelo diminui à medida que o número de árvores na floresta aumenta, enquanto o viés permanece o mesmo.
- Regressão logística (RL) [19]: Regressão logística é um modelo de ML usado para prever a probabilidade de ocorrência de um evento em face de um conjunto de variáveis explanatórias. A função logística, também conhecida como função sigmoide, é usada para calcular o modelo logístico no qual cada valor do infinito negativo ao infinito positivo é fornecido como entrada e saída limitadas no intervalo de 0 e 1. Este algoritmo pode entender variáveis vetoriais e avaliar os coeficientes ou pesos para cada variável de entrada e, em seguida, prever que a classe expressou o valor do vetor de palavras.
- Suporte vector machine (SVM) [20]: SVM classifica as amostras tentando corrigir os limites do agrupamento das regiões do espaço de amostragem em que existem poucos dados. Este baseia-se na ideia de que instâncias de dados podem ser visualizadas como coordenadas em um espaço N-dimensional, com N sendo o número de características. Durante o treinamento, é procurado um hiperplano que melhor separe os dados em grupos distintos (classes) e que maximize a margem.

## 4 EXPERIMENTOS

Esta seção apresenta os experimentos realizados para avaliar o mecanismo de detecção de ataques DDoS proposto usando as técnicas de seleção descritas e os modelos de ML analisados nas seções anteriores. Para realizar os experimentos, foi utilizado o conjunto de dados BoT-IoT (apresentado na Seção 3.2). Este conjunto foi criado no Cyber Range da UNSW, incorporando tráfego maligno (ataques DDoS a partir de bots) e benigno (normal, i.e., de dispositivos não infectados). O conjunto de dados em questão já é sub-categorizado a fim de facilitar o processo de rotulação. No total, este conjunto de dados possui 69.3*GBs* de tamanho no formato PCAP, contendo mais de 72.000.000 de registros.

Portanto, os experimentos realizados avaliaram o desempenho das técnicas de seleção: Baixa Variância (BV), SVC, Extra-Árvore (EA), Lasso e MRMR, considerando os casos de 5, 10, 20, 30 e 40 características consideradas mais relevantes pelo ranking gerado. Decidiu-se avaliar esses casos de MRMR a fim de analisar melhor o comportamento desta técnica, bem como o impacto destas seleções nos modelos de ML. Cada uma das técnicas citadas foram utilizadas para selecionar as características de cada um dos modelos de ML discutidos anteriormente, ou seja, os modelos KNN, AD, FA, SVM e RL. Esta estratégia de avaliação visa analisar todas as possíveis combinações de técnicas e modelos, identificando quais seriam as combinações mais adequadas para cada contexto de redes IoT, visto que cada um possui singularidades.

A avaliação de desempenho das possíveis abordagens para mecanismo proposto (i.e., técnica de seleção e modelo de ML) consideraram os seguintes casos Verdadeiro Positivo (*True Positive* - TP), Falso Positivo (*False Positive* - FP), Verdadeiro Negativo (*True Negative* - TN) e Falso Negativo (*False Negative* - FN) para a detecção de ataques DDoS. A partir disso, as métricas de avaliação usadas foram:

- ACC - Acurácia (em porcentagem): Taxa de classificações corretas, independente da classe em questão. A acurácia é definida de acordo com a Equação 1.

$$ACC = \frac{TP + TN}{\text{TP} + FN + FP + TN} \quad (1)$$

- Tempo de Treinamento (em segundos): tempo necessário para treinar o modelo de ML com as características selecionadas de entrada, criando assim um classificador que irá detectar os ataques DDoS.
- Tempo de Detecção (em segundos): tempo necessário que o classificador criado gasta para detectar se um caso é ataque DDoS ou não.

A Figura 2 apresenta os resultados de acurácia dos classificadores avaliados em conjunto com as técnicas de seleção definidas neste artigo. Portanto, apresenta-se o desempenho dos 60 casos com todas as combinações possíveis de classificadores e técnicas de seleção de características.

A partir das informações apresentadas na Figura 2 percebe-se que a acurácia dos classificadores varia de acordo com a técnica de seleção aplicada, principalmente quando esses classificadores são baseados em abordagens que focam em dimensionalidade, como pode ser percebido nas Figuras 2(a), 2(b) e 2(f) com os resultados referentes aos classificadores KNN, RL e SVM, respectivamente.

Percebe-se nos resultados expostos que os classificadores baseados em divisão de subconjuntos, os modelos de ML AD (Figura 2(e)) e FA (Figura 2(d)) são menos impactados pela variação de técnicas de seleção, cerca de 5% de variação no desempenho do melhor (Extra-árvore) para o pior caso (MRMR com 5 características). Este fato ocorre devido ao processo recursivo de derivação dos subconjuntos consegue amenizar o impacto negativo no desempenho de características que acabem sendo ruído para o processo de treinamento do modelo.

O classificador Naive Bayes apresentou um desempenho médio abaixo dos demais classificadores, como pode ser visto na Figura 2(c), onde o melhor caso (usando a técnica de seleção MRMR com 10 características) obteve uma acurácia menor do que o pior dos demais classificadores, exceto o classificador RL. Portanto, este mostrou-se um modelo de ML pouco eficaz para realizar a detecção de ataques DDoD em redes IoT quando comparado com os demais classificadores avaliados.

Os resultados apresentados mostram a importância da etapa de seleção de características para o desempenho, no que se refere a acurácia, dos classificadores. Por exemplo, a partir do uso da técnica de seleção mais adequada, o desempenho dos classificadores KNN e SVM aumenta em 12% e 11%. Além disso, o classificador RL usando todas as 80 características extraídas possíveis (sem seleção) tem um desempenho inviável, enquanto que usando a técnica de seleção SVC, este atinge quase 90% de acurácia.

A seguir, as Tabelas 2 e 3 apresentam o tempo necessário para realizar o treinamento do modelo de ML (gerando assim o classificador que irá realizar a detecção) e para os classificadores detectarem os casos de ataques DDoS, respectivamente. Estes dados mostram a viabilidade de implantação dos classificadores em conjunto com as técnicas de seleção em diversos contextos de redes IoT.

Com relação ao tempo de treinamento, há contextos em que seja necessário realizar um treinamento recorrente a fim de adaptar o modelo a alta dinamicidade da rede (tais como casas inteligentes ou campus inteligentes), este precisa ser treinado constantemente para compreender as mudanças e assim realizar a detecção dos ataques DDoS de forma eficaz. Nestes casos, o tempo para treinamento do modelo torna-se crucial para viabilizar a detecção de DDoS para a rede IoT em questão. Por outro lado, contextos em que a periodicidade de treinamento do modelo de ML será maior, pois a tendência da rede é manter um comportamento de tráfego (tal como industrias 4.0), pode-se tolerar um tempo maior de treinamento, visto que não impactará fortemente a aplicação deste como solução.

**Table 2: Tempo para Treinamento (em segundos)**

| Técnica | KNN | RL | NB | AD | FA | SVM |
|---|---|---|---|---|---|---|
| 80 Originais | 4.3434 | 0.5696 | 0.4602 | 2.9871 | 17.6732 | 750.0124 |
| BV | 3.9986 | 1.5685 | 0.2433 | 0.4751 | 18.5551 | 136.5173 |
| SVC | 7.7277 | 1.2978 | 0.1760 | 0.4390 | 10.2803 | 429.3620 |
| EA | 1.9442 | 1.0001 | 0.1180 | 0.2464 | 14.7057 | 73.9402 |
| Lasso | 0.5768 | 0.1465 | 0.0678 | 0.2905 | 6.7494 | 149.0834 |
| MRMR 5 | 1.4612 | 0.6038 | 0.2692 | 0.4034 | 2.0133 | 71.3881 |
| MRMR 10 | 1.6668 | 0.6906 | 0.0682 | 0.1424 | 3.2605 | 122.2469 |
| MRMR 20 | 1.1667 | 1.1897 | 0.1016 | 0.3703 | 6.1411 | 315.2436 |
| MRMR 30 | 1.7109 | 0.3177 | 0.1292 | 0.7775 | 9.3966 | 249.2081 |
| MRMR 40 | 2.1239 | 0.3959 | 0.1826 | 1.3434 | 13.8375 | 331.7743 |

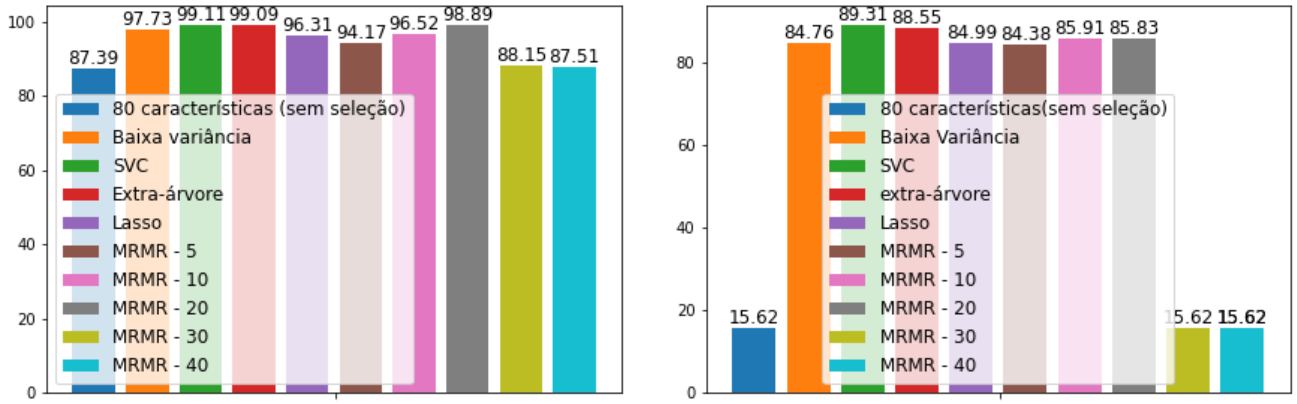
**Table 3: Tempo para Detecção (em segundos)**

| Técnica | KNN | RL | NB | AD | FA | SVM |
|---|---|---|---|---|---|---|
| 80 Originais | 2.7147 | 0.0225 | 0.1023 | 0.0270 | 0.4750 | 73.4508 |
| BV | 10.4843 | 0.0074 | 0.0675 | 0.0057 | 0.3935 | 29.5844 |
| SVC | 9.5433 | 0.0055 | 0.0540 | 0.0074 | 0.3838 | 22.6553 |
| EA | 4.6476 | 0.0047 | 0.0318 | 0.0048 | 0.3813 | 14.8496 |
| Lasso | 1.3210 | 0.0025 | 0.0108 | 0.0063 | 0.4309 | 10.8884 |
| MRMR 5 | 3.5233 | 0.0043 | 0.0079 | 0.0054 | 0.3546 | 11.1945 |
| MRMR 10 | 3.3528 | 0.0066 | 0.0134 | 0.0075 | 0.3602 | 13.2261 |
| MRMR 20 | 2.6894 | 0.0094 | 0.0291 | 0.0113 | 0.4012 | 18.4831 |
| MRMR 30 | 1.9181 | 0.0094 | 0.0368 | 0.0126 | 0.4090 | 29.9730 |
| MRMR 40 | 2.0655 | 0.0123 | 0.0544 | 0.0168 | 0.4394 | 37.5522 |

De acordo com os resultados apresentados na Tabela 2, os classificadores FA e, principalmente, SVM possuem um tempo de treinamento superior aos demais. Contudo, a aplicação da técnica de seleção MRMR (com 5 e 10 características) reduz o tempo de treinamento do classificador FA em torno de sete vezes, possibilitando sua implantação em soluções para diversos contextos de redes IoT, chegando a um patamar de tempo similar aos classificadores KNN e AD.

Assim como o tempo de treinamento, o tempo para a detecção é um aspecto crucial para redes IoT, principalmente as que são base para serviços críticos de saúde, segurança, etc. Desta forma, este tempo se torna um critério primordial que esta diretamente relacionado ao modelo de ML definido, bem como a técnica de seleção aplicada em conjunto.

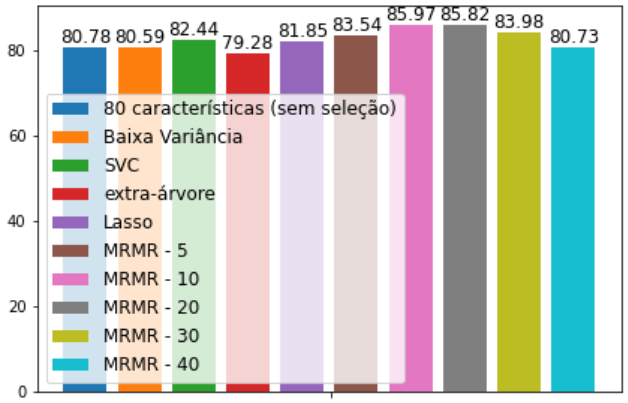
Considerando-se os dados expostos na Tabela 3, percebe-se que a técnica de seleção utilizada pode reduzir em ate doze vezes o tempo para detecção, como é o caso do classificador NB. Este fato auxilia a implantação de soluções que melhor se moldem ao contexto de rede



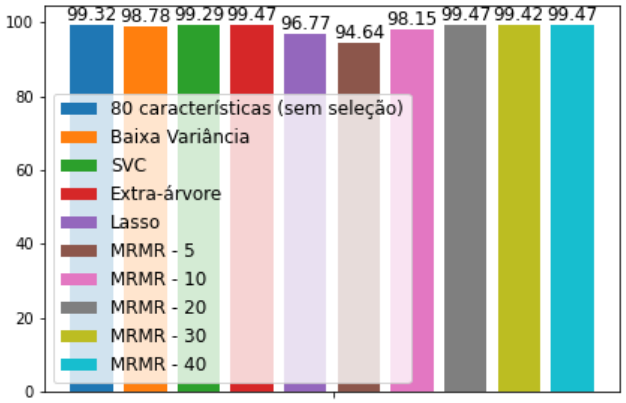

(a) KNN.



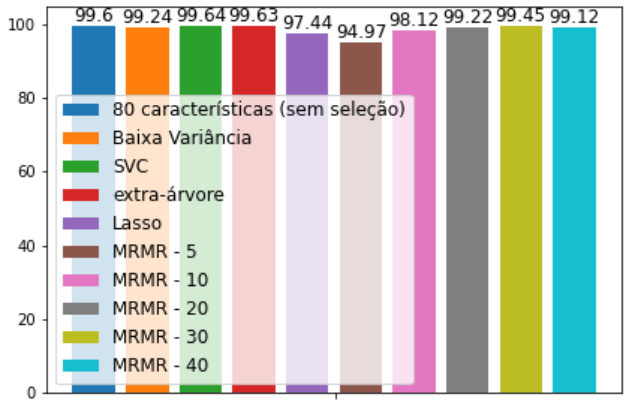

(b) Regressão Logística.



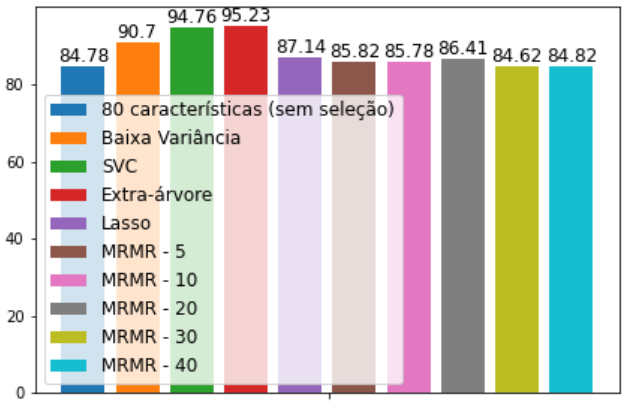

(c) Naive Bayes.




(d) Floresta Aleatória.




(e) Árvore de Decisão.




(f) SVM.

**Figure 2: Acurácia dos Classificadores com Seleção de Características.**

IoT que seja necessário detectar ataques DDoS. Os classificadores SVM, AD, RL e NB possuem um tempo para detecção são muito impactados positivamente (redução do tempo) pela aplicação das técnicas de seleção, enquanto que o classificador FA acaba tendo tempos similares independente das técnicas de seleção usadas.

Os resultados de Acurácia, Tempo de Detecção e de Treinamento sugerem em quais tipo de contexto de redes IoT cada combinação de modelo de ML com técnica de seleção se torna uma opção viável para uma solução de detecção de ataques DDoS. Em contextos que seja necessário uma alta acurácia e baixo tempo de treinamento, o classificador AD em conjunto com a seleção Extra-Árvore torna-se a melhor opção. Caso o maior interesse seja em somente realizar a detecção e treinamento no menor tempo possível, suportando uma acurácia média, uma combinação adequada seria do classificador NB com a seleção MRMR-10. Portanto, a aplicação da combinação mais adequada torna-se uma particularidade do contexto a ser aplicado.

## 5 CONCLUSÃO

O aumento constante de dispositivos IoT compartilhando informações cresce exponencialmente a cada ano, devido a isso, há uma tendência do aumento no risco desses equipamentos tornarem-se alvos de ataques de agentes mal intencionados. Recentemente, os dispositivos tem tido suas vulnerabilidades exploradas a fim de realizar ataques virtuais, objetivando comprometer a disponibilidade de serviços para usuários legítimos. Diante desta ameaça, bem como das características dos dispositivos IoT, é importante o desenvolvimento de soluções para segmentação tráfego e detectar anomalias em redes IoT.

A partir desta realidade, esta pesquisa apresentou um mecanismo para segmentação de rede e detecção de anomalia em redes IoT usando modelos de ML, o qual é baseado em uma análise das características de tráfego a fim de identificar a seleção de características que maximizem o desempenho dos classificadores analisados. Os experimentos realizados usando um conjunto de dados real apresenta que uma seleção de características adequada impacto positivamente o desempenho do mecanismo, bem como o tempo de processamento para detecção de ataques. Como trabalhos futuros, pretende-se evoluir o mecanismo para agregar a funcionalidade de identificação de outros tipos de ataques em redes IoT, como por exemplo os ataque de canais laterais.

## AGRADECIMENTOS

Os autores agradecem a Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico - FUNCAP (Processo DEP-0164-00242.01.00/19) pelo apoio financeiro.

## REFERÊNCIAS

[1] Ruchi Vishwakarma and Ankit Kumar Jain. A survey of ddos attacking techniques and defence mechanisms in the iot network. *Telecommunication Systems*, 73(1): 3–25, 2020.

[2] Iman Sharafaldin, Arash Habibi Lashkari, Saqib Hakak, and Ali A Ghorbani. Developing realistic distributed denial of service (ddos) attack dataset and taxonomy. In *2019 International Carnahan Conference on Security Technology (ICCST)*, pages 1–8. IEEE, 2019.

[3] Rohan Doshi, Noah Apthorpe, and Nick Feamster. Machine learning ddos detection for consumer internet of things devices. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 29–35. IEEE, 2018.

[4] R Vinayakumar, Mamoun Alazab, Sriram Srinivasan, Quoc-Viet Pham, Soman Kotti Padannayil, and K Simran. A visualized botnet detection system based deep learning for the internet of things networks of smart cities. *IEEE Transactions on Industry Applications*, 2020.

[5] Masaaki Yamauchi, Yuichi Ohsita, Masayuki Murata, Kensuke Ueda, and Yoshiaki Kato. Anomaly detection for smart home based on user behavior. In *2019 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–6. IEEE, 2019.

[6] Arunan Sivanathan, Hassan Habibi Gharakheili, Franco Loi, Adam Radford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman. Classifying iot devices in smart environments using network traffic characteristics. *IEEE Transactions on Mobile Computing*, 18(8):1745–1759, 2018.

[7] Hamed HaddadPajouh, Ali Dehghantanha, Raouf Khayami, and Kim-Kwang Raymond Choo. A deep recurrent neural network based approach for internet of things malware threat hunting. *Future Generation Computer Systems*, 85:88–96, 2018.

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

[9] Nickolaos Koroniotis, Nour Moustafa, Elena Sitnikova, and Benjamin Turnbull. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *CoRR*, abs/1811.00701, 2018. URL http://arxiv.org/abs/1811.00701.

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

[11] Hanchuan Peng, Fuhui Long, and Chris Ding. 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, 2005.

[12] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

[13] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

[14] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3): 1–27, 2011.

[15] Wen-Jyi Hwang and Kuo-Wei Wen. Fast knn classification algorithm based on partial distance search. *Electronics letters*, 34(21):2062–2063, 1998.

[16] Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[17] Kyoungok Kim. A hybrid classification algorithm by subspace partitioning through semi-supervised decision tree. *Pattern Recognition*, 60:157–163, 2016.

[18] Ratna Astuti Nugrahaeni and Kusprasapta Mutijarsa. Comparative analysis of machine learning knn, svm, and random forests algorithm for facial expression classification. In *2016 International Seminar on Application for Technology of Information and Communication (ISemantic)*, pages 163–168. IEEE, 2016.

[19] William J Meurer and Juliana Tolles. Logistic regression diagnostics: understanding how well a model predicts outcomes. *Jama*, 317(10):1068–1069, 2017.

[20] David Meyer and FH Technikum Wien. Support vector machines. *The Interface to libsvm in package e1071*, 28, 2015.