DIMI: Detecção Inteligente de Botnets Mirai em Redes IoT

Antonia Raiane Santos Araujo Cruz
Instituto Federal de Educação, Ciência e Tecnologia do Ceará
Canindé, CE, Brasil
raiane.santos@ifce.edu.br

Rafael Lopes Gomes
Universidade Estadual do Ceará
Fortaleza, Brasil
rafaellgom@larces.uece.br

Marcial Porto Fernandez
Universidade Estadual do Ceará
Fortaleza, Brasil
marcial@larces.uece.br

RESUMO

The emerging usage of Internet of Things (IoT) paradigm brings, together with new services, new threats to Information Security. Among these threats, we have the Mirai Botnet that performed several Distributed Denial of Service (DDoS) cyberattacks, exploring the vulnerabilites of IoT devices. Within this context, this paper presents a mechanism for detecting Mirai botnet attacks on IoT networks using ML techniques and comparing different approaches. The mechanism was evaluated using a set of traffic data from real IoT devices, achieving results with 99 % precision in detecting Mirai Botnet attacks.

KEYWORDS

Botnet, Internet das Coisas, Aprendizado de Máquina

1 INTRODUÇÃO

A evolução dos microcontroladores nos últimos anos possibilitou um grande avanço no poder computacional e, somado a isso, o paradigma Internet das Coisas (*Internet of Things* - IoT) evoluiu e colaborou com o crescimento das possibilidades de comunicação, permitindo, desde que o termo foi cunhado por Ashton em 1999 [1], a implantação de aplicações para executar tarefas diferentes em vários ambientes [2]. IoT conquistou rapidamente um espaço no cenário das tecnologias modernas e de telecomunicações, e atualmente esse paradigma abrange aplicações, como casas inteligentes [3], [4], cidades inteligentes [5], e indústria [6].

A popularização dos dispositivos IoT para aplicações de sensoriamento, processamento e comunicação trouxe também a preocupação com aspectos de segurança da informação, devido a características como número massivo de conexões e vulnerabilidades existentes nesses dispositivos. Essas características tornam as redes IoT uma poderosa ferramenta na expansão de ataques cibernéticos, o que justifica a necessidade de estudos e abordagens para a implantação de soluções que deem suporte às redes IoT e que implementem sistemas de detecção de ameaças e proteção para as aplicações executadas sobre essas redes.

Nesse cenário, uma das principais ameaças para as redes IoT são os ataques de Negação de Serviço Distribuído (*Distributed Denial-of-Service* - DDoS), que visam explorar as vulnerabilidades dos dispositivos a fim de sincronizar os ataques e congestionar um ou mais alvos, inviabilizando as aplicações existentes [7]. Segundo [8], os ataques DDoS são considerados uma das ameaças mais graves para as organizações, assim como para a manutenção da estabilidade da Internet. Nesse contexto, também se enquadram as botnets, porque atualmente os cibercriminosos optam por utilizar-se das botnets justamente pelo número massivo de dispositivos que podem ser controlados. Em 2016, a botnet Mirai comandou cerca de 100.000 dispositivos IoT para conduzir ataque DDoS contra infraestrutura de servidores *Domain Name Server* (DNS), afetando os serviços oferecidos por diversas empresas como Github, Amazon, Netflix, Twitter, CNN e Paypal [9].

Para além, um relatório da Traficon demonstra que em 2020 houve um aumento acentuado nas ocorrências de malwares em busca de vulnerabilidades em sistemas na Internet. Boa parte das observações ainda diz respeito à ameaça Mirai [10], logo que os dispositivos IoT são mais propensos a ataques. Com isso, uma abordagem promissora para realizar a detecção desses ataques é a aplicação de modelos de aprendizado de máquina (*Machine Learning* - ML), os quais utilizam os dados da rede IoT como entrada, para compreender o comportamento da rede.

Este trabalho propõe o mecanismo DIMI (Detecção Inteligente de Botnets Mirai em Redes IoT), uma abordagem para a detecção de ataques da botnet Mirai em redes IoT baseado em modelos de ML, onde para cada ataque específico realizado pelo Mirai é treinado e aplicado um modelo de ML. O mecanismo proposto utiliza como entrada o conjunto de dados de [11], que apresenta um tráfego de dispositivos IoT, os quais são tratados e as características mais inerentes são extraídas para entender o comportamento da rede IoT e da Mirai. Neste artigo foram avaliados os seguintes modelos de ML para compor o DIMI: *K-Nearest Neighbor* (KNN), *Support Vector Machines* (SVM) e *Logistic Regression* (LR).

O restante deste artigo está organizado da seguite forma: a Seção 2 contém os conceitos fundamentais sobre IoT e Botnet Mirai. A Seção 3 apresenta os trabalhos relacionados. A Seção 4 apresenta o mecanismo proposto, enquanto que a Seção 5 realiza uma avaliação do mecanismo proposto. Por fim, a Seção 6 apresenta a conclusão e os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Internet das Coisas e Aspectos de Segurança

A Internet das Coisas se apresenta como um conceito de presença difusa de uma variedade de dispositivos físicos embarcados com sensores e atuadores, que através de conexões, como rede sem fio, se comunicam usando a Internet e desenvolvendo uma rede de objetos inteligentes [12]. É um paradigma que conquistou rapidamente um espaço no cenário das tecnologias modernas e de telecomunicações, onde as “coisas” ou objetos participantes dessa comunicação são capazes de reagir conforme o ambiente em que estão inseridos.

Assim, considera-se que IoT é uma estrutura de rede dinâmica e global capaz de se autoconfigurar, baseada em protocolos de comunicação padronizados e que viabiliza interoperabilidade, onde os

objetos físicos e virtuais possuem identidade, atributos e personalidade [2]. Os objetos colaboram no processo de comunicação de forma ativa, interagindo com os demais objetos físicos interligados à Internet, colaborando com o meio.

O crescimento do número de dispositivos conectados à Internet nos últimos anos se deu de forma exponencial, porém muitos desses dispositivos não possuem requisitos de segurança adequados, tornando-se vulneráveis a uma diversidade de ameaças. Em um relatório de ameaças à segurança na Internet da Symantec [13], é possível ter uma noção de como o surgimento de novas ameaças é preocupante. Destaca-se que os atores de ataques direcionados demonstraram mais interesse em IoT como vetor de infecção, principalmente em se falando de ataques por meio de *worms* e *bots*.

Nesse contexto, os ciberataques baseados em malware estão evoluindo e explorando cada vez mais as vulnerabilidades de IoT, que torna-se um alvo fácil e atraente para atacantes que visam usá-los para criar redes de bots em larga escala [14], considerando que é muito mais complicado lidar com os problemas de segurança e privacidade que rodeiam esses dispositivos.

Para [15] o desafio de implementar mecanismos de segurança para ambientes IoT se dá devido a redes não confiáveis, protocolos utilizados e menos intervenção humana, o que cria um ambiente mais vulnerável.

2.2 Botnet Mirai

IoT surgiu como uma grande promessa e com inúmeros benefícios, entretanto a visão dos atacantes por esses dispositivos como potenciais integrantes de ataques DDoS, em larga escala, contribuiu para uma crescente onda de ataques, comprometendo diferentes dispositivos de IoT [16]. A intensidade desses ataques está diretamente ligado ao número de dispositivos infectados por *bots*, um tipo de mawalre que compromete os dispositivos tornandos-os zumbis de uma botnet.

Mirai é um dos malwares que tem ganhado mais popularidade atualmente devido o grande impacto no ataque DDoS de 2016 [17]. É baseado no Linux e converteu milhões de sistemas operacionais Linux e outros dispositivos IoT em bots. Reconhecido como o responsável pelo maior ataque DDoS registrado até o momento, que incluiu até 15 milhões de dispositivos IoT com uma velocidade de inundação de 1 Tbps, contra um provedor de hospedagem francês [15]. Projetado para infectar e controlar vários tipos de dispositivos IoT, como roteadores domésticos, DVRs e câmeras CCTV [18].

Utilizando-se de um *malware*, controla remotamente os *bots*. É constituída pelos componentes exposto na Fig. 1. O *bot*, que é o *malware* que infecta os dispositivos e realiza duas ações principais: propagar a infecção e atacar um servidor de destino assim que receber o comando correspondente da pessoa que está controlando (*botmaster*). O servidor C&C fornece ao *botmaster* uma interface de gerenciamento centralizada para verificar as condições e administrar os novos ataques DDoS. E o *BD da botnet* responsável por manter um banco de dados atualizado com detalhes sobre todos os dispositivos na botnet [7]. Deste modo, um dispositivo IoT que foi infectado pelo Mirai, se conecta ao servidor CNC para ser adicionado a botnet e se comunica regularmente com ele, esperando por seus comandos. Assim está distribuído os elementos que constituem o Mirai e o processo para lograr êxito na infecção das vítimas.

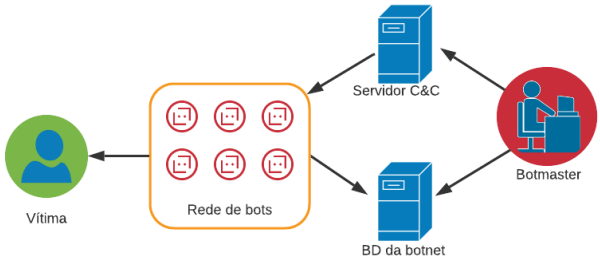


Figura 1: Operação e comunicação da botnet Mirai.
Fonte: Kolias et al. (2017)[7]

Adicionalmente, o Mirai apresenta variantes, as quais possuem singularidades (os ataques *Scan, Ack flooding, Syn flooding, UDP flooding* e *UDPplain*), dificultando assim a sua detecção. O ataque de *Scan* consiste em uma técnica de buscas em redes a fim de identificar dispositivos ativos e coletar informações sobre eles. O *Ack Flooding* ocorre no processo de *handshake* de três vias, onde o atacante envia um pacote habilitado para bit ACK com endereço de origem forjado, que obteve através do dispositivo de destino, pois ele não possui conexão com o host com endereço IP falsificado. O *Syn Flooding* visa deixar um servidor indisponível para tráfego legítimo, consumindo todos os recursos disponíveis deste servidor. Por meio do ataque *UDP Flooding*, o invasor envia pacotes UDP repetidamente para portas aleatórias no dispositivo de destino anonimamente, com a finalidade de manter o dispositivo ocupado com a requisição, logo inacessível. O que diferencia o *UDP Plain* do *UDP Flooding* é a porta de origem que se mantém fixa.

Deste modo, faz-se necessário o desenvolvimento de soluções de segurança para proteger as redes IoT da botnet Mirai, que possam compreender essas características e terem a implantação viável diante das limitações dos dispositivos IoT.

2.3 *Machine Learning*

Aprendizado de Máquina (*Machine Learning* - ML) é uma subárea da Inteligência Artificial (*Artificial Intelligence* - AI) que fornece às máquinas a capacidade de aprender sem programação explícita e aprimora o tratamento de determinada questão através da experiência [19]. O uso de técnicas de ML expande o leque de atividades automatizadas que podem ser realizadas por um computador devido a extensa variedade de algoritmos que são utilizados para análise de dados.

Através dos algoritmos, ML otimiza o desempenho na realização de determinada tarefa por meio de treinamento e aprendizado. Através dos algoritmos de aprendizado supervisionado, apresentam um modelo de classificação com base em um mapeamento aprendido, capturando os relacionamentos entre os parâmetros de entrada e a rotulagem necessária através da fase de treinamento [20]. Então, é necessário denotar o estado dos dados, sejam normal ou anômalo, de modo que o sistema aprenda a detectar intrusões através da classificação do tráfego durante a fase de treinamento.

Destacam-se como técnicas de aprendizado supervisionado, principalmente em relação a modelos de classificação, as técnicas Máquinas de Vetores de Suporte (SVM), K-vizinhos mais próximos (KNN) e Regressão Logística (RL).

O SVM é uma das técnicas de ML mais adotadas para a detecção de anomalias supervisionadas. Analisa os dados e reconhece padrões [21], o que possibilita classificar uma decisão como decisão padrão e os padrões que surgem muito longe do que foi estabelecido são determinados como anomalias, conforme Equação 1.

$$\sum_{i \in SV} y_i \alpha_i K(x_i, x) + b \tag{1}$$

Embora KNN apresente uma estratégia simples na abordagem dos problemas de classificação, o classificador implementa um conjunto de treinamento formado por vetores *n*-dimensionais e cada elemento deste conjunto representa um ponto no espaço *n*-dimensional [22], desta forma seu aprendizado baseia-se "no quão similar" é um dado do outro. A Equação 2 apresenta o modelo matemático, onde através de uma matriz de soma de todas as amostras e da probabilidade da classificação correta, *m* representa a probabilidade da amostra *i* ser classificada corretamente.

$$a(u) = argmax \sum_{i=1}^{m} [x_{i;u} = y] w(i, u) \tag{2}$$

No modelo logístico a variável resposta Y_i é binária, assumindo dois valores, como por exemplo, $Y_i = 0$ e $Y_i = 1$, o que pode ser denominado como "normal"ou "ataque", respectivamente. No caso da variável dependente Y assumir apenas dois possíveis estados (0 ou 1) e haver um conjunto de p variáveis independentes X_1 , X_2 , ... , X_p , o modelo de regressão logística pode ser escrito de acordo com a Equação 3.

$$P(Y = 1) = \frac{1}{1 + e^{-g(x)}} \tag{3}$$

Onde, g(x)= B_0 + B_1X_1+ ... + B_pX_p.

Os coeficientes B são estimados a partir do conjunto de dados analisado e, com a aplicação do método de máxima verossimilhança, é posível encontrar a máxima probabilidade da amostra ser classificada.

3 TRABALHOS RELACIONADOS

As referências da literatura discutidas nessa pesquisa, comprovam que existem vários estudos que avaliam a preocupação em discutir a segurança em IoT, inclusive relatando a necessidade de desenvolver mecanismos para a detecção de intrusões em IoT, com base nos desafios relacionados à questões de segurança da informação. Contudo, ainda há muito a se fazer para implementar mecanismos de segurança mais específicos para IoT.

Raza et al. [23] propuseram o mecanismo SVELTE para detecção de intrusões em redes IoT baseadas em endereçamento *IPv6 over Low power Wireless Personal Area Networks* (6LoWPAN) e o roteamento *IPv6 Routing Protocol for Low power and Lossy Networks* (RPL). O sistema apresenta os módulos: (a) mapeamento, que reconstrói o roteamento feito pelo protocolo RPL e o estende com parâmetros adicionais de detecção; (b) detecção, que identifica por meio de diversas técnicas, spoofing, informações alteradas e ataques de repasse seletivo; e, (c) mini-firewall, que além de proteger contra as típicas ameaças externas, fornece proteção em tempo real das ameaças internas. O SVELTE fornece proteção contra os tipos de ataques mais comuns. Apesar do SVELTE levar em consideração redes IoT, depende da aplicação de 6LoWPAN e RPL para executar as tarefas de rastreamento e detecção. Esses fatos limita a aplicabilidade do SVELTE em outras redes IoT e na detecção da botnet Mirai.

Em Doshi et al. [24] observa-se os resultados obtidos a partir da análise dos comportamentos das redes IoT, o que facilitou a seleção de características considerados no processo de detecção de ataques DDoS de alta precisão no tráfego de rede IoT, através de algoritmos de aprendizado de máquina, incluindo redes neurais. Apresentam um modelo de ameaça que faz diversas suposições sobre a rede IoT, com a finalidade de observar o tráfego entre os dispositivos IoT da rede local e a Internet. O dispositivo observa, armazena, manipula e bloqueia qualquer tráfego de rede que acesse o roteador, logo que todo e qualquer tráfego que acessa a rede LAN (*Local Area Network*) passa por este. Este mecanismo tem como objetivo de detectar e impedir o tráfego de ataques de negação de serviço originados de dispositivos dentro da LAN doméstica inteligente. No entanto, a abordagem proposta é específica para o ciberataque DDoS, pois monitora informações relacionadas ao volume de tráfego e à taxa de transmissão, que melhor caracteriza os ataques DDoS.

Meidan et al. [11] propuseram e avaliaram empiricamente um novo método de detecção de anomalias baseado em rede IoT, que extrai o comportamento instantâneo da rede usando auto-codificadores profundos para detectar tráfego de rede anômalo proveniente de dispositivos IoT comprometidos. A avaliação do método proposto foi realizado a partir da infecção de 9 dispositivos IoT comerciais com as botnets Mirai e Bashlite. Para validação empírica, extraíram características dinâmicos do conjunto de treinamento benigno e em seguida treinaram modelos de regressão para estudar os efeitos do recurso abordado. Como resultados finais, os experimentos configuram resultados eficientes para detectar com precisão e instantaneamente os ataques à medida que eles estavam sendo lançados por dispositivos IoT infectados.

Vinayakumar et al. [25] propõem um sistema de detecção de DDoS baseado em uma estrutura de ML de dois níveis para distinguir semanticamente DDoS 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.

A proposta de Said et al. [26], apresenta um mecanismo baseado em rede que pode ser aplicado na detecção de bots IoT infectados por malware do tipo Mirai. Destacam que a relevância do trabalho se deu pela capacidade do Mirai infectar uma ampla gama de dispositivos IoT através de um ataque de dicionário básico, com base em cerca de 60 entradas, principalmente pelo fato de que boa parte dos dispositivos usam credenciais padrão. Utilizaram o mecanismo YARA para a análise sintática e para implementar as regras de classificação. O mecanismo classifica uma amostra como Mirai caso ela seja classificada como qualquer variante do Mirai. Uma

segunda pesquisa [27] aborda o desenvolvimento de um algoritmo para detectar Bots de IoT infectados por Mirai em redes de grande escala. Esse algoritmo tem como ação principal a detecção de bots que procuram dispositivos vulneráveis na rede.

Tabela 1: Comparação entre os Trabalhos Relacionados.

Referências	Ataque	Mecanismo de Detecção
[23]	Sinkhole e selective-forwarding	Sistema baseado em 6LoWPAN e RPL.
[24]	DDoS	Modelos KNN, LSVM, DT, RF e NN
[27]	Mirai	YARA
[26]	Mirai	Assinaturas de tráfego Mirai e Subamostragem bidimensional
[11]	Botnet	Uso de Deep Autoencoders.
[25]	DDoS em servidores DNS	ML de dois níveis.
Este trabalho	Botnet Mirai	ML para cada ataque específico.

O levantamento bibliográfico mostrou que embora algumas pesquisas desenvolveram metodologias para a detecção de ataques Mirai, poucas discutem a implementação dos algoritmos ML voltados para a classificação de cada variante do Mirai, conforme Tabela 1 . Além disso, os trabalhos relacionados com a detecção de DDoS possuem uma abordagem incompatível com redes IoT e/ou aplicam técnicas de ML sem um embasamento adequado para detectar os ataques variantes da Botnet Mirai. Muitas das pesquisas não se preocupam em utilizar fluxo de dados reais de dispositivos IoT, e a métrica ponderada, por vezes, é a acurácia, quando na verdade acurácia não deve ser considerada se o conjunto de dados estiver desbalanceado, outra preocupação que não é relatada em boa parte dos trabalhos. Por fim, é importante avaliar o desempenho das métricas precisão e recall e considerar a classificação de cada variante do Mirai.

4 MECANISMO DIMI

Atualmente faz-se necessário desenvolver mecanismos que implementem soluções de segurança na detecção de Botnets Mirai e seus ataques específicos em redes IoT, haja vista que são redes que ainda carecem de capacidade de adaptação e identificação para ataques como (*Scan, SYN Flooding, ACK Flooding, UDP Flooding e UDP Plain flooding*). Este artigo apresenta o mecanismo DIMI baseado em técnicas de ML para detectar estes ataques usando as informações sobre o tráfego da rede IoT e, a partir da análise desses dados, modelar o comportamento dos dispositivos dessa rede, o que é entendido pelas técnicas ML, permitindo a detecção de intrusões Mirai.

O desenvolvimento do mecanismo proposto foi divido em três fases: (I) Análise do comportamento da Botnet Mirai, (II) Seleção de Características e (III) Treinamento do modelo de ML para detecção. Uma visão geral do mecanismo, bem como a interação entre as fases é apresentada na Fig. 2.

Para os experimentos e avaliação do mecanismos foi disposto o conjunto de dados disponível em [11], contendo as informações de tráfego real de dispositivos IoT como (thermostat, câmeras de segurança, babá eletrônicas, entre outros), integrando dados de tráfego de entrada e saída da rede IoT. Ressalta-se que o conjunto de dados utilizado é apenas uma entidade centralizadora das informações necessárias, desse modo o mecanismo independe do conjunto de dados aplicado. As subseções seguintes detalham as fases de desenvolvimento.

4.1 Análise do comportamento da Botnet Mirai

As soluções de segurança para detectar o Botnet Mirai em redes IoT ainda carecem de capacidade de adaptação e identificação para os ataques Mirai específicos: *Scan, Ack flooding, Syn flooding, UDP flooding* e *UDPplain*. Inicialmente, o conjunto de dados foi particionado em cada ataque específico, isolando os dados de cada um destes e, consequentemente, permitindo a análise singular dos mesmos. A Fig. 3 apresenta o número de instâncias que são ataques e tráfego normal, considerados durante esse trabalho. O tráfego normal se mantém igual para cada conjunto de treinamento com os ataques Mirai.

Nesta fase optou-se por não realizar o balanceamento dos dados, tendo em vista que o processo de balanceamento não mantém a natureza de cada ataque, e a integridade do conjunto de dados utilizado, como [11, 15, 24, 28] afirmam. Além disso, durante os testes realizados o balanceamento não interferiu na melhoria considerável dos resultados.

4.2 Seleção de Características

A partir do conjunto de dados pode-se extrair características sobre o tráfego de rede, incluindo estatísticas correlativas dos dados referentes ao endereço IP, protocolo da camada de transporte, tamanho do pacote, sinalizadores (flags) de cabeçalho TCP e outros informações coletadas pelas ferramentas de monitoramento de rede reais. Nesse trabalho a extração de características foi feita usando a mesma abordagem feita na referência [29].

Neste trabalho foram consideradas características resultantes da agregação das estatísticas baseadas em fluxos, onde um fluxo é definido como uma tupla composta: endereço IP de origem, endereço MAC de origem e porta de origem, endereço IP de destino, endereço MAC de destino, porta de destino e protocolo da camada de transporte. Assim, foram selecionadas as seguintes características sobre o tráfego da rede IoT:

- Número de pacotes transmitidos em um fluxo específico;
- Tempo de chegada (considerando média, desvio padrão e a última janela);
- Tamanho dos pacotes de saída (considerando média e variação);
- Tamanho do pacote de entrada e saída (considerando média, desvio padrão e covariância do fluxo em relação a outros fluxos).

No mecanismo proposto, a seleção de características foi projetada para considerar uma janela de tempo de 1 minuto, visto que este intervalo de tempo torna-se adequado considerando dois aspectos [11, 15]: (I) O uso de intervalos muito curtos dificulta a implantação desta solução em redes IoT compostas de um número massivo (tal como cidades inteligentes e Industrias 4.0), visto que faz-se necessário um alto poder computacional para processar todos os fluxos ativos; e, (II) O intervalo de 1 minuto se apresenta como viável para representar o comportamento da Botnet Mirai em redes IoT de acordo com a literatura.

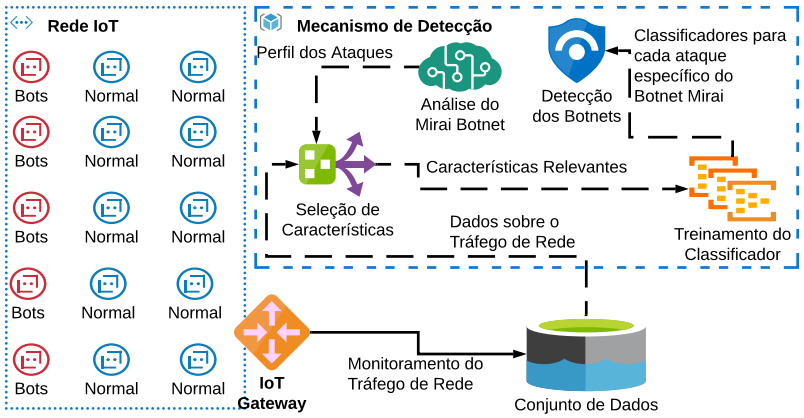


Figura 2: Mecanismo DIMI para a detecção de ataques do botnet Mirai.

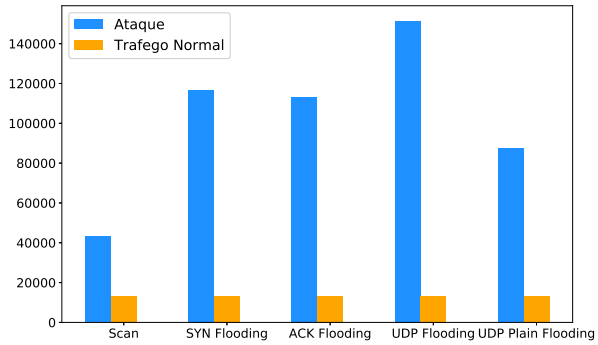


Figura 3: Quantidade de instâncias de dados.

4.3 Treinamento do Detector de Intrusões Mirai

Após a fase de seleção de características, executa-se o treinamento do classificador com base nas características extraídas do conjunto de dados de tráfego de rede e as técnicas de ML. As técnicas são treinadas considerando os atributos selecionados anteriormente, onde o classificador é treinado para cada ataque específico da Botnet Mirai. Essa abordagem permite a implantação da solução capaz de entender as especificidades de cada ataque, bem como possibilita uma análise de qual técnica de ML mais se adequa a determinado ataque.

O mecanismo proposto considerou três modelos de ML: SVM, KNN, e LR. Essas técnicas são robustas contra ruído de dados e imperfeições, o que reflete em soluções que melhor se enquadram para lidar com os requisitos de dispositivos heterogêneos e as características singulares nos contextos de IoT (como cidades inteligentes, casas inteligentes, etc.) [19]. Para além, o mecanismo proposto aplica etapas adicionais em cada abordagem utilizada, a fim de ajustar os melhores parâmetros.

A técnica SVM detecta se um fluxo faz parte de um ataque Botnet Mirai ou se é um fluxo normal. O SVM foi configurado com os parâmetros mais adequados ao mecanismo DIMI que são, kernel de função básica (RBF), que segue a função z kernel para mapear a matriz do kernel a partir de matrizes de dados, aproximando os pesos dos características às classes. O uso de RBF permite que o mecanismo proposto seja implantado em redes IoT independente do contexto, uma vez que as aproximações realizadas melhoram o tempo computacional e, consequentemente, o tempo de detecção. Também foram configurados os parâmetros *gamma='scale'* e regularização em *C=10*, o que reduziu o tempo de treinamento e produziu melhores resultados.

KNN é um classificador não paramétrico baseado no treinamento mais próximo de k exemplos no espaço de recurso, onde pesos são atribuídos aos vizinhos, de forma que os vizinhos mais próximos contribuem mais para a média do que para os mais distantes. No mecanismo proposto, o comportamento de um fluxo de ataque botnet Mirai é mais próximo dos fluxos de ataque do que os fluxos de tráfego normais. Durante os experimentos, a proposta aplicou um *k=5*, uma vez que o estudo empírico realizado sugeriu que ele alcança o melhor desempenho de precisão e tempo. O estudo empírico avaliou os valores k de 3 a n (n é o tamanho da amostra). Por fim, na Regressão Logística (LR) é utilizada uma função logística capaz de modelar uma variável dependente binária. A função logística (também chamada sigmoide) descreve as propriedades de crescimento da amostra em um determinado ambiente observado, explicando a relação entre variáveis. Assim, no mecanismo proposto, LR modela o crescimento dos características selecionados representando uma possível ocorrência do Mirai Botnet. Além disso, o mecanismo aplica o método ANOVA [30] para identificar as características mais relevantes de acordo com o teste do valor F que avalia os valores esperados de uma variável quantitativa dentro da classificação das amostras da Botnet Mirai.

Após essa fase, os classificadores encontram-se aptos a realizar a tarefa de detecção do Botnet Mirai, ou seja, são aplicados cinco detectores baseados em modelos de ML para lidar com os ataques *Scan*, *Ack Flooding*, *Syn Flooding*, *UDP Flooding* e *UDPplain*.

5 RESULTADOS E DISCUSSÕES

Esta seção apresenta os experimentos realizados para avaliar o mecanismo DIMI, o qual foi implementado na linguagem Python usando

a biblioteca Scikit-learn [1]. Para avaliar o mecanismo proposto e as técnicas ML, considerou-se a ocorrência de casos Verdadeiros Positivos (TP), Falsos Positivos (FP), Verdadeiros Negativos (TN) e Falsos Negativos (FN) para as duas classes definidas: ataques ou normal. Desta forma, com a implementação realizou-se a análise de diversos aspectos sobre o mecanismo proposto, onde as seguintes métricas de avaliação foram consideradas:

- Acurácia (ACC): Taxa de classificações corretas, independente da classe. Calcula-se usando a Equação 4.

$$ACC = \frac{\text{TP} + TN}{\text{TP} + FN + FP + TN} \tag{4}$$

- Precisão: Porcentagem de preedições corretas de casos positivos dentro de determinada classe.

$$Precisao = \frac{\text{TP}}{\text{TP} + FP} \tag{5}$$

- Recall: Eficiência do classificador em detectar as classes corretas, modelada de acordo com a Equação 6.

$$Recall = \frac{\text{TP}}{\text{TP} + FN} \tag{6}$$

- F1-Score: É a média ponderada do recall e precisão do modelo. Valor de F1-Score apresenta-se na equação 7. vspace0.1cm

$$F1 - Score = \frac{2 * \text{TP}}{(2 * \text{TP} + \text{FN} + \text{FP})} \tag{7}$$

- AUC (*Receiver Operating Characteristic*): Avalia a capacidade do classificador de evitar classificações erradas, distinguindo duas classes e sendo definida pela Equação 8.

$$AUC = \frac{1}{2}\left(\frac{\text{TP}}{\text{TP} + FN} + \frac{\text{TN}}{\text{TN} + FP}\right) \tag{8}$$

Os classificadores considerados foram treinados com dados normais e com ataques, onde 70% do conjunto de dados utilizou-se na fase de treino e os demais 30% na fase de teste. Essa mesma configuração foi utilizada para todos os ataques da Botnet Mirai. A Tabela 2 apresenta o tempo que cada técnica ML levou na fase de treino e também o número de instâncias analisadas.

A partir dos resultados, percebe-se que para cada ataque o classificador LR levou o menor tempo de execução, enquanto que SVM tem um custo temporal maior, o que pode inviabilizar o seu uso em aplicações em tempo real. SVM chega a ter um tempo de execução mais elevado, chegando próximo a 1 hora, enquanto que o menor tempo é de 0,11 segundos para a técnica LR.

A acurácia variou de aproximadamente 76% a 100%. A Tabela 3 apresenta os resultados verificados através do *score* de predições corretas. Considerando que os dados não foram balanceados, para garantir a originalidade dos ataques, a acurácia não é a métrica mais indicada para avaliação do mecanismo, deste modo, é importante avaliar os resultados das métricas de precisão, recall e f1-score. Os quais apresentam dados que comprovam que o mecanismo DIMI é capaz de encontrar amostras relevantes em um alto índice, não atribuindo classificações de ataques a tráfego normal. Assim como o recall demonstra uma alta capacidade do mecanismo apresentar as amostras positivas.

[1] https://scikit-learn.org/stable/index.html

Tabela 2: Tempo de treinamento para cada técnica.

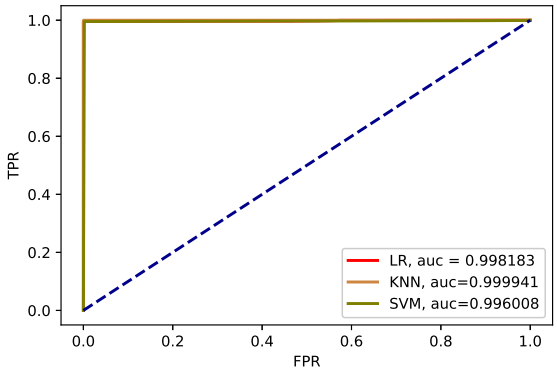
Ataque	Classificador	Tempo (s)	nº instâncias de treino
Scan	LR	0,11	
	KNN	3,92	39413
	SVM	367,91	
Ack Flooding	LR	0,76	
	KNN	5,79	88478
	SVM	1391,30	
Syn Flooding	LR	0,47	
	KNN	6,56	90944
	SVM	1063,80	
UDP Flooding	LR	2,28	
	KNN	5,98	70336
	SVM	1273,59	
UDPplain	LR	0,36	
	KNN	3,01	115215
	SVM	796,31	

Tabela 3: Desempenho de Classificação.

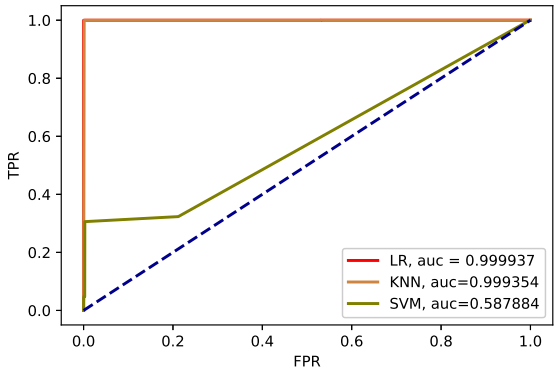
Métricas	Técnica	Scan	Ack	Syn	UDP	UDP Plain
Acurácia (%)	LR	86	99	99	99	86
	KNN	99	99	99	99	99
	SVM	76	89	89	92	86
Precisão (%)	LR	76	100	100	99	99
	KNN	99	99	99	99	99
	SVM	76	89	89	92	86
Revocação (%)	LR	100	100	99	99	99
	KNN	99	99	99	99	100
	SVM	99	100	99	99	100
F1-score(%)	LR	85	100	99	99	99
	KNN	99	99	99	99	99
	SVM	86	94	94	95	93

Os ataques baseados na abordagem de inundação são caracterizados pelo grande número de pacotes semelhantes enviados pelos dispositivos atacantes para tornar o destino indisponível. Por outro lado, o ataque *Scan* concentra-se na disseminação da infecção em redes IoT, verificando as vulnerabilidades em outros dispositivos. Consequentemente, o ideal é que as técnicas ML aplicadas alcancem um nível muito baixo de Falso Negativo (FN), uma vez que quando não identifica-se a execução de uma botnet dentro da rede, também não é possível estabelecer uma política de proteção preventiva e corretiva. À vista disso, considerar a métrica de desempenho *Recall* é crucial para uma melhor análise do mecanismos de detecção.

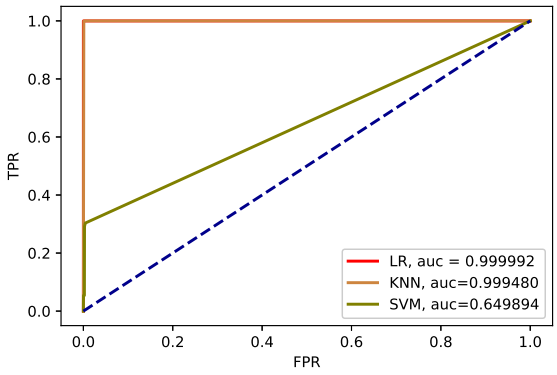
Para os ataques UDP Flooding e UDPplain, todas as técnicas de ML alcançaram altos valores de *Recall*, indicando sua adequação a



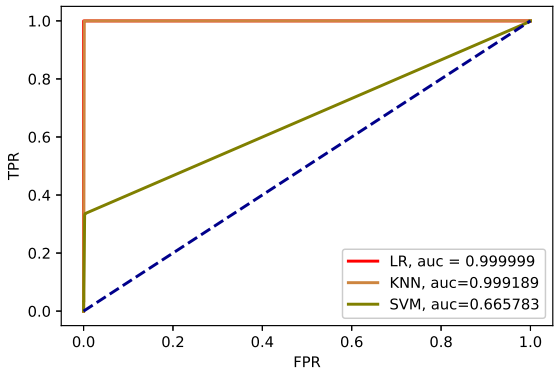
(a)Curva ROC Scan



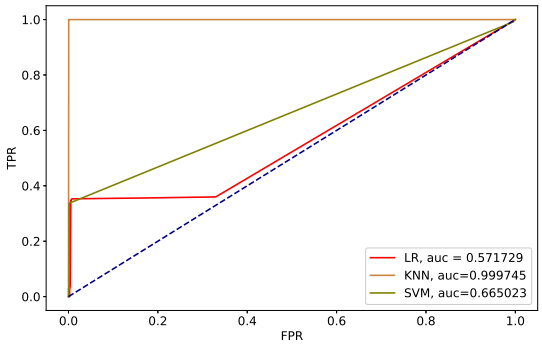
(b)Curva ROC ACK Flooding



(c)Curva ROC SYN Flooding



(d)Curva ROC UDP Flooding



(e)Curva ROC UDP Plain Flooding

Figura 4: Curva ROC.

solução do problema abordado. Entretanto, para o ataque Scan, o SVM apresentava recall de 76%, comprometendo sua aplicabilidade na detecção desse ataque. Enquanto que KNN obteve a maior capacidade de detectar os ataques específicos da Botnet Mirai, alcançando altos valores de precisão e recall. Esse bom desempenho é mais claro quando analisa-se os valores das curvas ROC AUC, ilustrados na Fig. 4, visto que a área sob curva ROC mostra o quanto o modelo

e capaz de distinguir entre as duas classes implementadas. Observa-se que o valor de AUC varia de 0 a 1, dado com base no cálculo da Taxa de Falso Positivo (FPR) e a Taxa de Verdadeiro Positivo (TPR). Portanto, os resultados da curva ROC AUC permite destacar o bom desempenho de KNN, em comparação com SVM e LR, logo esse resultado reflete também a eficiência do classificador.

6 CONSIDERAÇÕES FINAIS

A evolução dos recursos computacionais permitiu a inovação e o desenvolvimento de novas tecnologias para melhorar a acessibilidade e agilidade das atividades e tarefas do dia a dia. Dessas novas tecnologias, surgiu a implantação de redes IoT em vários contextos, como casas inteligentes, cidades inteligentes, etc. Entretanto, muitos dispositivos IoT ainda carecem de recursos de segurança adequados, estando vulneráveis a ataques cibernéticos.

Um dos ciberataques existentes que é mais perigoso para as redes IoT é a botnet Mirai, que executa os ataques específicos, com particularidades. Portanto, é necessária uma solução para detectar ataques cibernéticos em redes IoT, considerando estas particularidades. Uma abordagem promissora para aprimorar a capacidade de segurança de redes IoT é a implantação de soluções baseadas em ML.

Nesse contexto, este artigo apresentou o mecanismo DIMI para detectar esses ataques específicos do Mirai Botnet em redes IoT baseadas em técnicas de ML. Em relação às técnicas de ML, o mecanismo proposto avaliou as abordagens KNN, SVM e LR. O mecanismo DIMI foi avaliado usando um conjunto de dados de tráfego real de dispositivos IoT, indicando uma precisão de 99% na detecção de ataques Mirai em redes IoT.

Como trabalho futuro, pretende-se investigar a eficácia do mecanismo com novos ataques cibernéticos, incluindo outras ameaças à segurança das redes IoT. Pode-se também planejar ações adicionais que visem implementar ambientes de proteção para a dispositivos IoT, especificamente, além de determinar uma função de tratamento do tráfego detectado como ataque. Caso implementado também poderá refletir em abordagens que isole rapidamente os dispositivos detectados como origem dos ataques. E finalmente, propor uma política de acesso dos dispositivos externos a uma rede IoT, propondo uma avaliação e validação do dispositivo, o que já implementará previsibilidade de possíveis dispositivos atacantes.

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] Kevin Ashton. That 'internet of things' thing, jun 2009. URL https://www.rfidjournal.com/that-internet-of-things-thing.

[2] Fernando Augusto Teixeira, Gustavo Menezes Vieira, Pablo Marcondes Fonseca, Fernando Magno Quintao Pereira, Hao Chi Wong, Jose Marcos Silva Nogueira, and Leonardo Barbosa Oliveira. Defending internet of things against exploits. *IEEE Latin America Transactions*, 13(4):1112–1119, 2015.

[3] Biljana L Risteska Stojkoska and Kire V Trivodaliev. A review of internet of things for smart home: Challenges and solutions. *Journal of Cleaner Production*, 140:1454–1464, 2017.

[4] Hamza Zemrane, Youssef Baddi, and Abderrahim Hasbi. Internet of things smart home ecosystem. In *Emerging Technologies for Connected Internet of Vehicles and Intelligent Transportation System Networks*, pages 101–125. Springer, 2020.

[5] H Samih. Smart cities and internet of things. *Journal of Information Technology Case and Application Research*, 21(1):3–12, 2019.

[6] Celia Garrido-Hidalgo, Teresa Olivares, F Javier Ramirez, and Luis Roda-Sanchez. An end-to-end internet of things solution for reverse supply chain management in industry 4.0. *Computers in Industry*, 112:103127, 2019.

[7] Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou, and Jeffrey Voas. Ddos in the iot: Mirai and other botnets. *Computer*, 50(7):80–84, 2017.

[8] Natalija Vlajic and Daiwei Zhou. Iot as a land of opportunity for ddos hackers. *Computer*, 51(7):26–34, 2018.

[9] Scott Hilton. Dyn analysis summary of friday october 21 attack. *Dyn blog https://dyn. com/blog/dyn-analysis-summary-of-friday-october-21-attack*, 2016.

[10] Traficom. Malware detected by traficom. URL https://www.traficom.fi/en/statistics/malware-detected-traficom.

[11] Yair Meidan, Michael Bohadana, Yael Mathov, Yisroel Mirsky, Asaf Shabtai, Dominik Breitenbacher, and Yuval Elovici. N-baiot—network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, 17(3):12–22, 2018.

[12] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[13] Symantec. Istr internet security threat report. 24, feb 2019.

[14] Elisa Bertino and Nayeem Islam. Botnets and internet of things security. *Computer*, (2):76–79, 2017.

[15] Ruchi Vishwakarma and Ankit Kumar Jain. A survey of ddos attacking techniques and defence mechanisms in the iot network. *Telecommunication Systems*, pages 1–23, 2019.

[16] Diego M Mendez, Ioannis Papapanagiotou, and Baijian Yang. Internet of things: Survey on security and privacy. *arXiv preprint arXiv:1707.01879*, 2017.

[17] Jong-Hyouk Lee and Hyoungshick Kim. Security and privacy challenges in the internet of things [security and privacy matters]. *IEEE Consumer Electronics Magazine*, 6(3):134–136, 2017.

[18] Michele De Donno, Nicola Dragoni, Alberto Giaretta, and Angelo Spognardi. Ddos-capable iot malwares: Comparative analysis and mirai investigation. *Security and Communication Networks*, 2018, 2018.

[19] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[20] James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, Mar 2005. ISSN 0343-6993. doi: 10.1007/BF02985802. URL https://doi.org/10.1007/BF02985802.

[21] Sarah M Erfani, Sutharshan Rajasegarar, Shanika Karunasekera, and Christopher Leckie. High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. *Pattern Recognition*, 58:121–134, 2016.

[22] Onel Harrison. Machine learning basics with the k-nearest neighbors algorithm, mar 2012. URL https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761.

[23] Shahid Raza, Linus Wallgren, and Thiemo Voigt. Svelte: Real-time intrusion detection in the internet of things. *Ad hoc networks*, 11(8):2661–2674, 2013.

[24] 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.

[25] 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.

[26] Najah Ben Said, Fabrizio Biondi, Vesselin Bontchev, Olivier Decourbe, Thomas Given-Wilson, Axel Legay, and Jean Quilbeuf. Detection of mirai by syntactic and behavioral analysis. In *2018 IEEE 29th International Symposium on Software Reliability Engineering (ISSRE)*, pages 224–235. IEEE, 2018.

[27] Ayush Kumar and Teng Joon Lim. Early detection of mirai-like iot bots in large-scale networks through sub-sampled packet traffic analysis. In *Future of Information and Communication Conference*, pages 847–867. Springer, 2019.

[28] Ankur Lohachab and Bidhan Karambir. Critical analysis of ddos—an emerging security threat over iot networks. *Journal of Communications and Information Networks*, 3(3):57–78, 2018.

[29] Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. Kitsune: an ensemble of autoencoders for online network intrusion detection. *arXiv preprint arXiv:1802.09089*, 2018.

[30] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.