UNIVERSIDADE FEDERAL DO ABC

Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas

Graduação em Engenharia de Informação

ANÁLISE DE MODELOS DE MACHINE LEARNING PARA DETECÇÃO DE ATAQUES DDOS EM DISPOSITIVOS IOT

ANA BEATRIZ ROCHA DA CUNHA E SILVA

SANTO ANDRÉ,

2023

Ana Beatriz Rocha da Cunha e Silva

ANÁLISE DE MODELOS DE MACHINE LEARNING PARA DETECÇÃO DE ATAQUES DDOS EM DISPOSITIVOS IOT

Trabalho de Graduação apresentado ao curso de Engenharia de Informação da Universidade Federal do ABC, como parte dos requisitos necessários para a obtenção do grau de Bacharel em Engenharia de Informação.

Orientador: Prof. Dr. Ricardo Suyama

SANTO ANDRÉ,

2023

AGRADECIMENTOS

Gostaria de agradecer a todos que me ajudaram direta ou indiretamente no desenvolvimento deste Trabalho de Graduação, em especial:

Ao Prof. Dr. Ricardo Suyama pela orientação, suporte e disponibilidade durante todo processo.

Aos professores e colegas da UFABC, pelo apoio e por todo conhecimento compartilhado ao longo da jornada acadêmica.

À minha família, amigos e namorado, pelo apoio incondicional e incentivo constante a alcançar meus objetivos.

RESUMO

O uso de dispositivos de Internet da Coisas (IoT) se tornou popular e vem crescendo cada vez mais a cada dia, e com isso, cresce também a preocupação com a segurança e privacidade desses sistemas. Grande parte dos dispositivos IoT possuem armazenamento e processamento computacional limitado, por isso não possuem sistemas de segurança robustos, o que os torna vulneráveis a ataques. Existem diversos tipos de ataques, sendo um dos principais o ataque DDoS (Distributed Denial of Service), que utiliza botnets para atacar e causar indisponibilidade no alvo. Neste cenário, as técnicas de Machine Learning podem ser empregadas para detectar esses ataques. O objetivo deste trabalho é realizar uma análise da viabilidade do uso de algoritmos de Machine Learning para identificar ataques DDoS nesses dispositivos, para isso, foram comparados os algoritmos de Árvore de Decisão, Random Forest e LightGBM.

Palavras-chave: Internet das Coisas, Machine Learning, Ataques DDoS.

ABSTRACT

The use of Internet of Things (IoT) devices has become popular and is steadily growing every day. With this growth, concerns about the security and privacy of these systems are also increasing. Many IoT devices have limited storage and computational processing capabilities, lacking robust security systems, thus making them vulnerable to attacks. There are various types of attacks, with one of the main ones being the Distributed Denial of Service (DDoS) attack, which uses botnets to target and cause unavailability on the victim. In this scenario, Machine Learning techniques can be employed to detect these attacks. The aim of this work is to analyze the feasibility of using Machine Learning algorithms to identify DDoS attacks on these devices. For this purpose, Decision Tree, Random Forest, and LightGBM algorithms were compared.

Keywords: Internet of Things, Machine Learning, DDoS attacks.

LISTA DE FIGURAS

LISTAS DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

O termo Internet of Things (IoT) surgiu em 1999, em um artigo do cientista britânico Kevin Ashton (ASHTON, 2009), que atribuiu o termo a um conjunto de sensores conectados à internet. Há diversas definições para IoT, para (SANTOS et al.) a Internet das Coisas é uma extensão da Internet atual, que proporciona aos objetos do dia a dia que possuem capacidade computacional e de comunicação, se conectarem à Internet. Já (SCHILLER et al., 2022) definiram como um sistema inteligente, com consciência abrangente (que é capaz de tomar decisões com base em dados e informações), transmissão confiável e processamento inteligente de dados. Para (TEIXEIRA et al., 2015) é uma infraestrutura de rede dinâmica e global capaz de se autoconfigurar, tem objetos físicos e virtuais que utilizam interfaces inteligentes, são conectadas à Internet e são capazes de interagir entre si. Um sistema IoT é capaz de medir e armazenar dados do ambiente, utilizando sensores que são conectados à internet, e utilizar esses dados para diversas aplicações.

O uso e desenvolvimento de sistemas IoT cresceram rapidamente nas últimas décadas, esses sistemas se transformaram em objeto de estudo nas universidades e se tornaram um dos principais itens de interesse da indústria tecnológica. Os dispositivos IoT são utilizados em larga escala e para diversas funções e aplicações, como smartwatches, carros automatizados, sensores industriais, smart home, smart cities etc. A quantidade de dispositivos conectados à internet tende a aumentar cada vez mais e a projeção é de que em 2025 tenham 100 bilhões de dispositivos IoT conectados (ROSE et al., 2015), e quanto maior o número de dispositivos conectados, maior a quantidade de dados sendo produzidos.

Com o crescimento do uso de dispositivos IoT, cresce também a importância e preocupação com a segurança e privacidade desses sistemas. As vulnerabilidades desses dispositivos podem abrir oportunidades para ataques cibernéticos que colocam em risco diversos dados e informações de acesso, além de causarem danos e mau funcionamento no sistema. Um ataque cibernético pode afetar não só o dispositivo localmente, mas também toda rede em que ele está conectado, além de conseguir acesso a informações e dados sensíveis. Além disso, esses ataques podem custar caro para empresas, é previsto que os custos globais com crimes cibernéticos cresçam 15% ao ano nos próximos cinco anos, atingindo US$10,5 trilhões anualmente até 2025 (CYBERMAGAZINE, 2022).

As redes de dispositivos IoT podem sofrer diversos tipos de ataques, que podem ser caracterizados como ataques físicos, de rede, de software e de criptografia (HUSSAIN et al., 2020). Um dos principais é o ataque DDoS (Distributed Denial of Service, ou, ataque distribuído de negação de serviço), que é um ataque de rede no qual o atacante envia diversas solicitações para sobrecarregar o alvo e fazer com que ele interrompa o funcionamento (XIAO et al., 2018).

São utilizadas técnicas estatísticas para detecção de ataques cibernéticos, existem diversos modelos diferentes de detecção, baseados em técnicas estatísticas diferentes. De forma geral, são utilizados modelos para identificar padrões de comportamento em sistemas de segurança e detectar diferenças significativas entre os dados recebidos e o comportamento esperado (PERLIN et al., 2011).

O objetivo deste trabalho é avaliar o desempenho de diferentes modelos de Machine Learning (Aprendizado de Máquina) para identificação de ataques DDoS nas redes IoT. Machine Learning, consiste na “construção de sistemas que aprendem e melhoram o desempenho, com base nos dados que consomem” (Oracle, s.d). A escolha da utilização dessa solução em detrimento de técnicas estatísticas foi feita baseada na capacidade desses algoritmos de se adaptarem e aprenderem de acordo com os dados recebidos, o que dá a eles a capacidade de identificar padrões, detectar variações e anomalias, que poderiam passar despercebidas em técnicas tradicionais.

Uma grande dificuldade encontrada na utilização de Machine Learning como solução para detecção de ataques em dispositivos IoT é a escassez de bases de dados democratizadas. Tendo em vista essa dificuldade, Bezerra et al. criaram um dataset com dados de um dispositivo IoT com interações legítimas e malignas, que será utilizado para o treinamento, teste e verificação dos modelos avaliados neste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Estrutura IoT

O termo Internet das Coisas, quando citado por Kevin Ashton, se referia a utilização de RFID (Radio Frequency Identification ou Identificação por Radiofrequência) na cadeia de produção da Procter & Gamble (ASHTON, 2009). Com o avanço da tecnologia, ocorreu o desenvolvimento da microeletrônica, comunicações

e sensoriamento, o que tornou os objetos conectados “inteligentes”. Os sistemas IoT utilizam a internet como base para comunicação entre os dispositivos, e, portanto, são capazes de realizar a coleta e processamento de dados dos ambientes aos quais estão integrados.

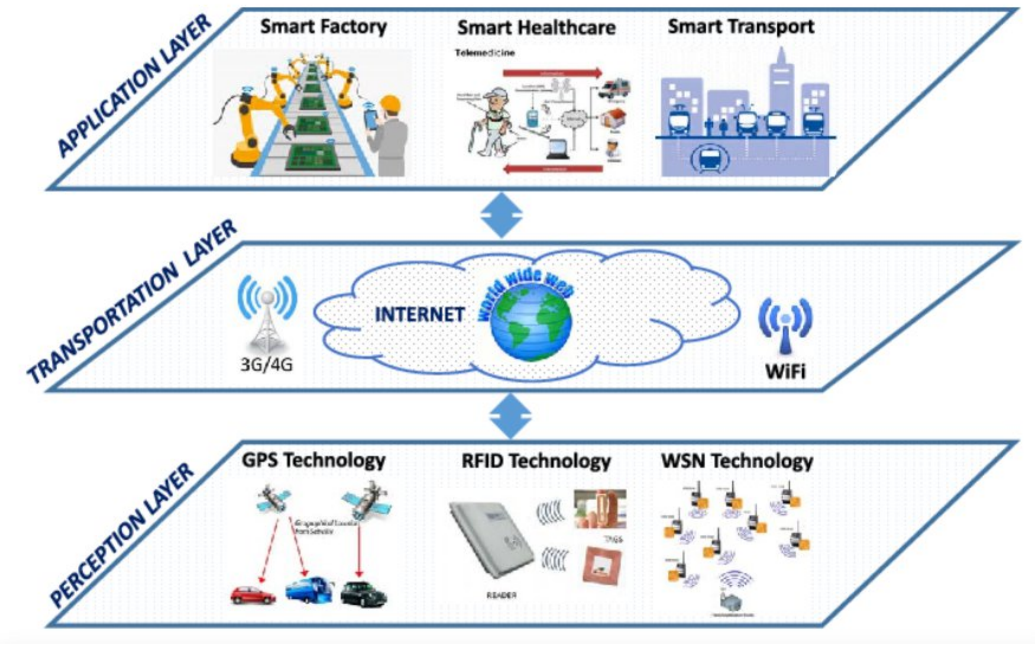
Por suas características, os dispositivos IoT podem atuar na coleta de dados, na automatização de processos, no monitoramento de algum recurso etc. As aplicações de IoT estão presentes em diversas áreas da economia (BUYYA; DASTJERDI, 2016) (LEITE et al., 2017), como as listadas abaixo:

- Indústria: controle de qualidade e monitoramento de maquinário.
- Automação Residencial: alarmes, controles de iluminação, termostato e consumo de energia.
- Hospitalar: monitoramento de sinais vitais, gerenciamento de inventario.
- Agricultura: monitoramento das condições climáticas e do solo, irrigação inteligente.
- Meio Ambiente: monitoramento da qualidade do ar e da água.

A arquitetura de redes dos sistemas IoT pode ser classificada em camadas, conforme as funções desempenhadas. O número de camadas e suas nomenclaturas variam dependendo do autor, a divisão mais comumente utilizada é a de três camadas. (Frustaci et al., 2018) utiliza as seguintes nomenclaturas de camadas para classificar um sistema IoT: Camada de Percepção, Camada de Transporte e Camada de Aplicação. As três camadas estão descritas abaixo e ilustradas na Figura 1:

- Camada de Percepção: é responsável pela detecção, coleta e processamento de dados.
- Camada de Transporte: é responsável por realizar o transporte por redes (podem ser com ou sem fio), dos dados coletados pela camada de percepção, para dispositivos conectados à internet.
- Camada de Aplicação: é responsável por utilizar os dados coletados e mostrá-los ao usuário final.

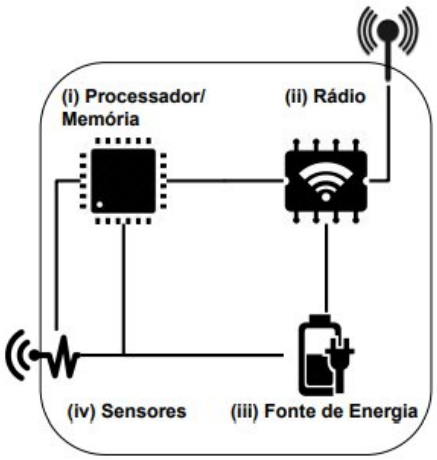
Figura 1 – Exemplo de camadas de um sistema IoT.



Fonte: Frustaci et al., 2018.

A Figura 2 apresenta a arquitetura básica dos dispositivos IoT utilizados na Camada de Percepção (SANTOS et al.), os componentes estão descritos abaixo:

Figura 2 – Estrutura básica de dispositivos IoT.



Fonte: Santos et al.

- I – Processador/Memória: composta por uma memória de armazenamento de dados e programas, um microcontrolador e conversor analógico-digital.

- II – Unidade de Comunicação: canal de comunicação (pode ser com ou sem fio).
- III – Fonte de Energia: responsável por fornecer energia (geralmente uma bateria).
- IV – Sensores: realizam o monitoramento e captura de dados do ambiente em que estão inseridos.

Uma das maiores vulnerabilidades dos dispositivos IoT é o limite de armazenamento e processamento computacional, que impede o uso de um sistema de autenticação mais robusto e utilização de criptografia, e, além desses fatores, há o custo energético e a grande variabilidade de infraestrutura e protocolos utilizados (MORAES; HAYASHI, 2021).

Há também vulnerabilidades em cada camada do sistema, como as listadas a seguir (MORAES; HAYASHI, 2021):

- Camada de Percepção:
  - É composta de hardwares e sensores que ficam vulneráveis a ataques e danos físicos.
  - Dispositivos que carecem de um sistema de autenticação robusto e de qualidade ficam vulneráveis a invasões de usuários não autorizados, que podem obter acesso aos dados armazenados, coletando-os indevidamente ou fazendo alterações.
- Camada de Rede:
  - Um usuário não autorizado pode obter acesso a algum nó da rota de transmissão e obter acesso aos dados.
  - Vulnerável a ataques de rede.
- Camada de Aplicação:
  - Um usuário não autorizado pode passar pelo sistema de autenticação e obter acesso aos dados.
  - Vulnerável a ataques de software.

Com os exemplos acima, pode-se observar que os sistemas IoT são vulneráveis a ataques físicos, de rede, de software e vazamentos de informação. Os ataques físicos têm como objetivo comprometer o hardware físico dos dispositivos IoT, os ataques de rede visam danificar a infraestrutura da rede que conecta os dis-

positivos, os ataques de software exploram as vulnerabilidades dos softwares usados pelos dispositivos IoT e os ataques de vazamento de informação tem como objetivo obter acesso e propagar os dados coletados, armazenados e processados pelos dispositivos Iot. Alguns dos principais ataques sofridos por dispositivos IoT são (XIAO et al., 2018):

- DoS: consiste em sobrecarregar o servidor do sistema com solicitações falsas, fazendo com que o sistema pare de operar.
- DDoS: é um tipo específico de ataque DoS, no qual são utilizados milhares de endereços de protocolo de internet para enviar as solicitações falsas ao servidor alvo.
- Jamming: consiste em interromper as transmissões de rádio nos dispositivos, causando o esgotamento dos recursos do sistema, como largura de banda, energia etc.
- Spoofing: consiste em se passar por um dispositivo IoT legítimo para conseguir acesso ao meio, e assim poder lançar outros ataques.

Os ataques DDoS são um dos principais ataques na área da computação, e é a maior ameaça para a Internet das Coisas (SELVARAJ, 2018). Considerando a estrutura de três camadas, esse tipo de ataque pode acontecer na camada de transporte e/ou na camada de aplicação.

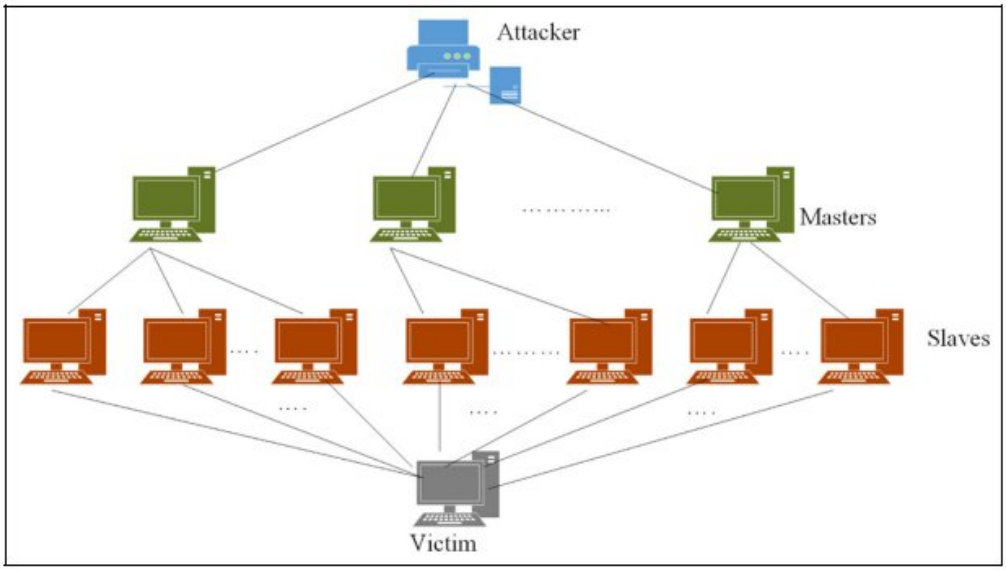
2.2 DDoS

Os ataques DDoS podem ser classificados na categoria de ataque de rede, nestes ataques, a rede é o alvo e o invasor não precisa necessariamente estar próximo dela (HUSSAIN et al., 2020). No geral, neste tipo de ataque, o invasor captura máquinas com falhas de segurança e dá o comando para que essas máquinas enviem um fluxo intenso de solicitações para a rede, que não consegue distinguir as solicitações legítimas das maliciosas. Esse ataque sobrecarrega a capacidade do servidor, e pode apenas afetar a qualidade do serviço ou pode deixá-lo completamente indisponível.

Os ataques DDoS seguem uma estrutura básica: primeiramente, ocorre a criação dos “mestres”, máquinas comprometidas que são capturadas quando apresentam alguma falha de segurança. Em seguida, os mestres recebem comando do atacante para capturar mais máquinas corrompidas, que serão controladas pelos

mestres, chamados de bots ou escravos. Por fim, o atacante envia o comando para iniciar o ataque, enviando um grande fluxo de pacotes para a vítima (MAHJABIN et al., 2017). Nesses ataques é comum que sejam utilizados endereços IPs falsificados. A Figura 3 ilustra o esquema da estrutura básica de um ataque DDoS:

Figura 3 – Estrutura básica de um ataque DDoS.



Fonte: Mahjabin et al., 2017.

O principal agravante de ataques DDoS é a sua característica distribuída, que permite um ataque muito maior e mais efetivo, além de tornar mais difícil o rastreamento do atacante original.

Existem vários tipos de ataques DDoS, que podem ser divididos em: ataques de esgotamento de largura de banda e ataques de esgotamento de recursos. Os ataques de esgotamento de largura de banda congestionam a rede alvo, de modo que os usuários não conseguem acessá-la. Já os ataques de esgotamento de recursos congestionam a memória e a CPU (Central Processing Unit ou Unidade de Processamento Central) do dispositivo, deixando a vítima impossibilitada de responder às solicitações dos usuários reais (MAHJABIN et al, 2017).

A seguir, tem-se exemplos comuns de ataques DDoS:

- HTTP Flood: o atacante manipula as solicitações HTTP, sobrecarregando o servidor e fazendo com que os usuários legítimos não consigam ser atendidos.

- UDP Flood: o atacante envia pacotes UDPs para sobrecarregar o alvo, causando esgotamento da largura de banda e levando à inacessibilidade de recursos.
- ICMP Flood: semelhante ao UDP Flood, o atacante sobrecarrega um alvo com um número de pacotes de ICMP (ping), causando esgotamento da largura de banda e consequentemente, indisponibilidade da rede.

Em 2016 ocorreu o maior ataque DDoS visto até então. O ataque teve como alvo Krebs on Security, OVH e Dyn, sendo esta última uma empresa que controla a estrutura DNS e por isso causou a queda de diversos sites, como Twitter, Netflix, The Guardian, CNN e muitos outros (THE GUARDIAN, 2016). Recentemente, as empresas Google e Amazon relataram terem sido alvo de um dos maiores ataques DDoS já registrados, chegando a 398 milhões de solicitações por segundo (FOLHA, 2023). O causador do ataque realizado em 2016 foi o botnet Mirai, que é um malware baseado em Linux que infecta centenas de milhares de dispositivos IoT e os transforma em bots, se aproveitando das vulnerabilidades de segurança desses dispositivos. Um agravante do botnet Mirai é que seu código fonte está disponível na Internet, podendo ser utilizado para outros ataques.

Em sistemas IoT, esses ataques podem sobrecarregar os dispositivos ou a infraestrutura de rede para deixar os serviços desse sistema indisponíveis. O ataque pode ocorrer na Camada de Percepção, nesse caso, o ataque é direcionado aos dispositivos e o objetivo é esgotar recursos como a CPU e memória. Pode ocorrer também na Camada de Transporte, que são ataques direcionados a infraestrutura de rede (gateways, roteadores e servidores) e o objetivo é causar esgotamento na largura de banda para interromper a comunicação entre os dispositivos. E podem ocorrer ataques na Camada de Aplicação, o alvo são aplicativos ou serviços específicos, que são atingidos com um grande volume de solicitações para deixar o serviço indisponível (AL-HADHRAMI et al., 2021).

Nesse cenário, a aplicação de Machine Learning para detecção de ataques cibernéticos em dispositivos IoT vem se tornando alvo de estudos e pode ser uma solução vantajosa para este tipo de problema.

2.3 Machine Learning

Machine Learning pode ser definido como uma área da Inteligência Artificial cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado, bem como a construção de sistemas capazes de adquirir conhecimento de forma automática (MONARD; BARANAUSKAS). É utilizado para ensinar as máquinas a lidarem com os dados de forma mais eficiente (MAHESH, 2018).

Os algoritmos de Machine Learning podem ser aplicados em (SHINDE; SHAH, 2018):

- Visão Computacional: detecção, processamento e reconhecimento de objetos.
- Previsão: classificação, análises, previsões e recomendações.
- Detecção de Anomalias.
- Análise Semântica.
- Processamento de Linguagem Natural: tradução automática, chatbots.
- Recuperação de Informações.

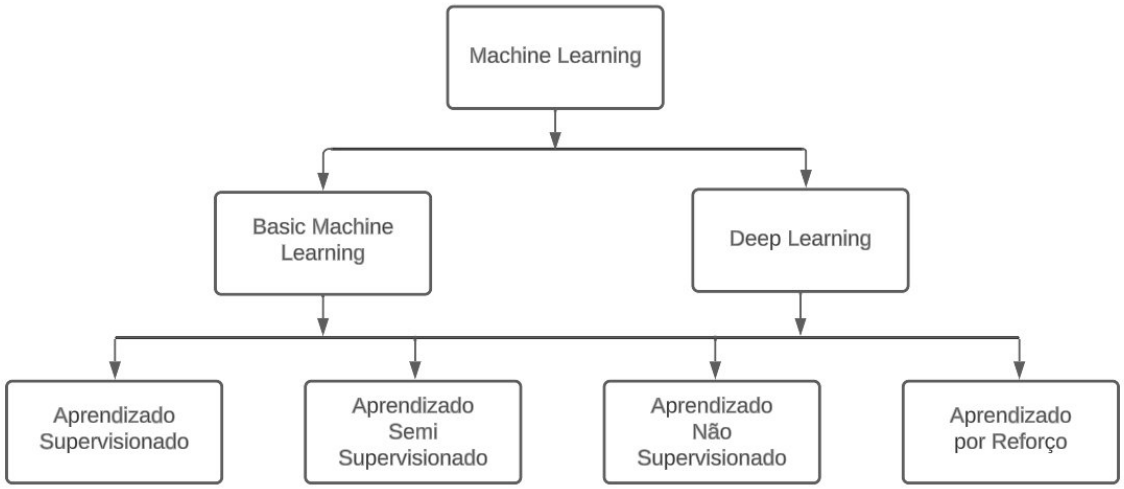
Os algoritmos são divididos em: Basic Machine Learning (Aprendizado de Máquina Básico) e Deep Learning (Aprendizado Profundo). O Deep Learning é uma técnica baseada nas Redes Neurais Artificiais, que tentam simular o comportamento do cérebro humano, quanto mais camadas uma rede neural possuir, mais precisas são as previsões. Esses algoritmos não utilizam necessariamente dados estruturados, podem processar dados como textos e imagens (IBM).

Os algoritmos Basic Machine Learning se referem a métodos tradicionais e menos complexos do que os algoritmos Deep Learning. Podem ser classificados de acordo com seu método de aprendizagem (HUSSAIN et al., 2020), as classificações estão listadas abaixo:

- Supervisionado: o treinamento é realizado com um conjunto de dados rotulados, os exemplos de entradas e saídas são correspondentes.
- Semi Supervisionado: há ausência de rótulos na maioria dos dados, mas presentes em alguns.
- Não Supervisionado: o ambiente fornece entradas que não necessariamente são rotuladas para descobrir padrões e estruturas ocultas.
- Reforço: os algoritmos aprendem por meio da interação com o ambiente e os feedbacks recebidos, são treinados para tomar sequências de decisões, recebendo recompensas ou penalidades em troca.

A Figura 4 ilustra a divisão e classificação dos algoritmos de Machine Learning e traz alguns exemplos de aplicação:

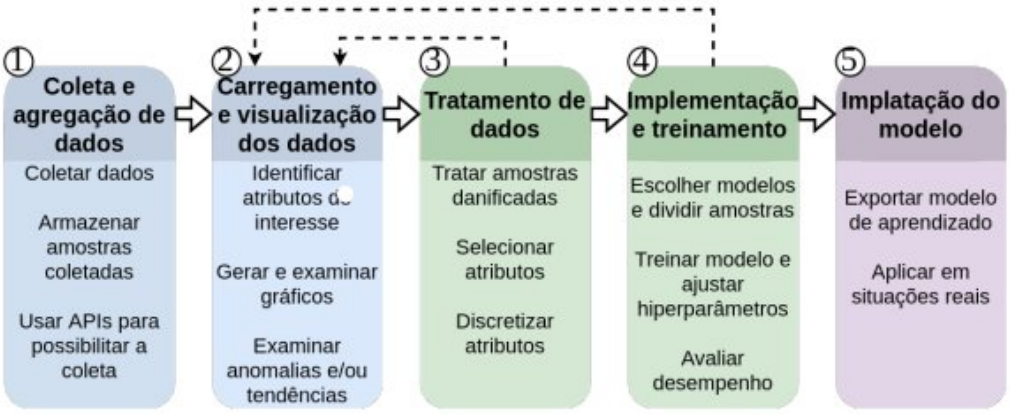
Figura 4 – Classificação de algoritmos de Machine Learning.



Fonte: Autoria Própria.

As aplicações de Machine Learning tendem a seguir um fluxo de trabalho em comum (BOCHIE et al., 2020), definido a seguir e ilustrado na Figura 5:

Figura 5 – Fluxo de trabalho de aplicações de Machine Learning.



Fonte: Bochie et al., 2020.

- Coleta e Agregação de Dados: os dados podem ser coletados através de diversas fontes, como dispositivos IoT ou dados disponibilizados publicamente.
- Carregamento e visualização dos dados: essa etapa é importante para identificar possíveis inconsistências e anomalias nos dados, como atributos repetidos e amostras corrompidas.
- Tratamento de Dados: nesta etapa são realizados os tratamentos e pré-processamentos nos dados de entrada do algoritmo.
    - Tratamento de amostras danificadas e formatação de atributos: pode ocorrer a exclusão de atributos com muitos valores ausentes, preenchimento com valores médios ou preenchimento com base em modelos. Quando necessário, a formatação dos atributos é alterada para se tornarem compatíveis com o algoritmo.
    - Selecionar atributos: seleção dos atributos com maior poder preditivo, nessa etapa os atributos que não são úteis para a previsão/classificação são excluídos. A seleção é feita aplicando métodos de seleção automáticos que utilizam técnicas estatísticas como análise de variância, importância de recursos baseada em árvores, métodos de filtro, e identifica e remove atributos que são redundantes.
- Implementação e Treinamento: nesta etapa é realizada a escolha do modelo, com base no tipo de aprendizado (Supervisionado, Semi Supervisionado, Não Supervisionado e Reforço) e nas necessidades de

aplicação (classificação, previsão, detecção de objetos etc.) e das métricas que serão utilizadas na avaliação.

  - Treinamento: primeiramente é realizada a divisão do conjunto de dados em conjuntos de treino, teste e validação. Os dados de entrada são processados pelo modelo para gerar as previsões iniciais. O algoritmo utiliza os dados de treinamento para encontrar parâmetros que otimizam a função objetivo (depende da aplicação, pode ser previsão, classificação etc.).
  - Avaliação dos resultados: é realizada utilizando as métricas escolhidas para a avaliação do modelo.

- Implementação do modelo: após o treinamento e avaliação do modelo, é possível fazer sua implantação em uma aplicação prática.

3 TRABALHOS RELACIONADOS

O uso de Machine Learning para detecção de ataques em dispositivos IoT vem se tornando uma solução interessante, é um tema que vem sendo cada vez mais abordado na literatura.

(MALIK; CHAUHAN, 2013) traz uma análise comparativa entre as técnicas e resultados de sete trabalhos que utilizaram Machine Learning para detecção de ataques em dispositivos IoT. Os trabalhos analisados abordaram diferentes tipos de ataques, utilizando diferentes algoritmos e datasets, mas utilizaram uma métrica em comum, a acurácia, que mede a fração de amostras classificadas corretamente. Nos resultados apresentados essa métrica variou entre 81% e 100%, indicando que algumas técnicas de Machine Learning têm capacidade de detectar os ataques.

(HUSSAIN et al., 2020) faz uma análise teórica de diversos algoritmos de Machine Learning que podem ser utilizados para problemas de segurança em dispositivos IoT. (CARVALHO et al., 2021) utilizaram técnicas de Deep Learning para classificar e identificar diferentes tipos de ataques em dispositivos IoT.

Na literatura é possível encontrar diversos trabalhos que realizam análises particulares utilizando Machine Learning em diferentes cenários, utilizando diferentes métodos e dados, como os citados anteriormente. Porém, são poucos trabalhos que exploraram os algoritmos de Machine Learning na detecção de ataques DDoS, especificamente, em dispositivos IoT.

(DOSHI et al., 2018) conduziram a detecção em tempo real de ataques DDoS, utilizando o comportamento do fluxo de dispositivos IoT em residências inteligentes. Foram realizados o monitoramento e a coleta das características do fluxo, como intervalo de chegada entre pacotes e pontos de chegada, e essas informações foram utilizadas como base para uma técnica de Machine Learning, a fim de desenvolver um modelo capaz de identificar possíveis ataques.

(PORTELA et al., 2021) realizaram a construção de um sistema inteligente que detecta ataques DDoS em sistemas IoT utilizando Redes Neurais (que são componentes de alguns modelos de Machine Learning) e Seleção de Características, para a seleção foram utilizadas técnicas diferentes para uma avaliação mais abrangente, as técnicas utilizadas foram: Máxima Relevância Mínima Redundância, Baixa Variância, Extra Árvore, Vetores de Suporte Linear Lasso. Foi feito o monitoramento e a coleta dos fluxos de rede do sistema IoT, seleção dos principais atributos coletados, treinamento da Rede Neural para detecção dos ataques e integração de computação em Névoa e em Nuvem.

(CRUZ, 2019) traz uma análise comparativa entre os algoritmos KNN (K-Nearest Neighbors ou k-vizinhos mais próximos, em português), SVM (Support Vector Machine ou Máquina de Vetores de Suporte, em português) e RL (Reinforcement Learning ou Aprendizado por Reforço, em português) de Machine Learning para detecção de ataques do botnet Mirai, que realiza ataques DDoS.

Este trabalho tem o propósito de fazer uma análise comparativa entre três modelos distintos de Machine Learning para detecção de ataques DDoS em dispositivos IoT. Nos estudos realizados até aqui, os modelos utilizados neste trabalho, Árvore de Decisão, Random Forest e LightGBM, não foram analisados/comparados para detecção de ataques DDoS (com mais de um botnet específico) em dispositivos IoT.

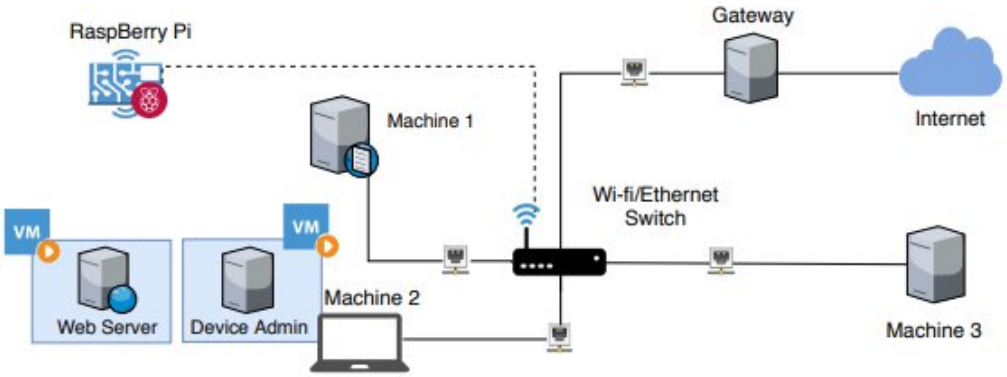
4 METODOLOGIA

A fim de verificar a eficácia dos métodos de Machine Learning na detecção de ataques DDoS em aplicações de IoT, foram realizadas simulações utilizando um conjunto de dados disponibilizado de forma gratuita e três modelos de Machine Learning para serem testados.

4.1 Dataset

O conjunto de dados utilizado nesse trabalho foi desenvolvido por Bezerra et al., com o objetivo de fornecer um conjunto de dados para pesquisas em detecção de ataques com hospedeiros. Foi obtido em um ambiente controlado, foram selecionados botnets específicos e utilizou-se um dispositivo Raspberry Pi, que foi o alvo das infecções, simulando três perfis de dispositivos IoT: central multimídia (MC), câmera de segurança (SC) e câmera de segurança com tráfego adicional (ST). Foi definida a topologia de rede, ilustrada na Figura 1:

Figura 6 – Topologia de rede experimental.



Fonte: Bezerra et al. 2018.

O gateway foi usado para fornecer acesso à Internet. A Máquina 1 hospedou um servidor utilizado pelo botnet Mirai, e também um cliente que consumiu os vídeos gerados pelos perfis SC e ST, e um servidor que gerou fluxo de vídeo para o perfil MC. A Máquina 2 hospedou o Servidor Web e o Device Admin, usados para simular o consumo de serviços web e configurar a câmera do perfil ST, respectivamente. E por fim, a Máquina 3 foi utilizada para simular atacantes que infectam dispositivos IoT com amostras de botnets (Bezerra et al., 2018).

O ambiente foi executado por uma hora apenas com atividades legítimas, e depois foram realizadas as infecções em três categorias, como mostra a Tabela 1:

Tabela 1 – Lista de infecções em cada perfil.

Tipo de Infecção	Perfil	Botnet(s)	Método de Infecção
1	MC	Hajime	SSH

	SC	Aidra	
	ST	BashLite	
2	MC	Mirai	
	SC		
	ST		
3	MC	Mirai, Doflo, Tsunami, Wro-ba	
	SC		
	ST		

Fonte: Bezerra et al. 2018.

Dos botnets utilizados para a montagem do dataset, apenas o Hajime não realiza ataques DDoS, até agora esse botnet não se comportou de maneira maliciosa. Portanto, o tipo de infecção 1 do perfil MC não foi utilizado.

Foram gerados três tipos de arquivos com dados coletados do Raspberry Pi: o primeiro com dados coletados da CPU (consumo de memória, número de tarefas etc.), o segundo com o fluxo de rede com a marcação de interação legítima ou maligna e o terceiro um arquivo PCAP com os pacotes capturados.

O conjunto de dados utilizado foi o que contém o fluxo de rede das interações, ele contém informações como: endereços de IP de origem e destino, protocolo, duração da interação, registro de data e hora, portas de origem e destino, volume de dados, a marcação de interação legítima ou maliciosa etc.

Na Tabela 2, estão listadas as características do conjunto de dados que será utilizado. É possível notar que há um desbalanceamento entre a quantidade de amostras legítimas e malignas, o que pode afetar a qualidade do treinamento dos modelos pela criação de vieses e generalização.

Tabela 2 – Características do dataset.

Características	**Total**
Amostras Legítimas	4.203
Amostras Malignas	1.357.786
Tipo de Amostra	Fluxo de Rede

Fonte: Bezerra et al. 2018.

4.2 Fluxo de Trabalho

Conforme apresentado na seção 2.3, a metodologia do trabalho seguiu o fluxo básico, sendo dividida nas seguintes etapas:

I. Obtenção do dataset: solicitação aos autores Bezerra et al.

II. Tratamento de dados: identificação e tratamento das amostras com inconsistências, remoção de atributos que não serão utilizados na análise e de atributos redundantes. Divisão do conjunto de dados em amostras que serão utilizadas para treinamento, testes e validação.

III. Escolha e pré-implementação dos modelos de aprendizado: definição do modelo de Machine Learning que será utilizado, treinamento do modelo com a amostra de dados definida para treino, análise dos resultados preliminares para avaliar necessidade de reprocessamento.

IV. Definição das métricas e avaliação dos resultados: definição das métricas utilizadas para avaliação e processamento utilizando a amostra de dados definida para validação dos modelos.

Os modelos de Machine Learning são do tipo de classificação binária: deve identificar se a interação é legítima ou maligna. Com base nisso, as métricas escolhidas para avaliar os modelos, com base em GE et al., foram:

Acurácia: é a razão entre as classificações corretas e o total de classificações realizadas.

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

Precisão: é a razão entre o número de classificações verdadeiramente positivas e o total de classificações positivas.

$$P = \frac{TP}{TP + FP} \#(2)$$

Recall (Sensibilidade): é a razão entre o número de classificações verdadeiramente positivas e o total de amostras positivas.

$$REC = \frac{TP}{TP + FN} \#(3)$$

Onde TP são as amostras verdadeiramente positivas (true positive), TN são verdadeiramente negativas (true negative), FP são falsos positivos (false positive) e FN são falsos negativos (false negative). As classificações positivas são interações malignas, enquanto as classificações negativas são as interações legitimas.

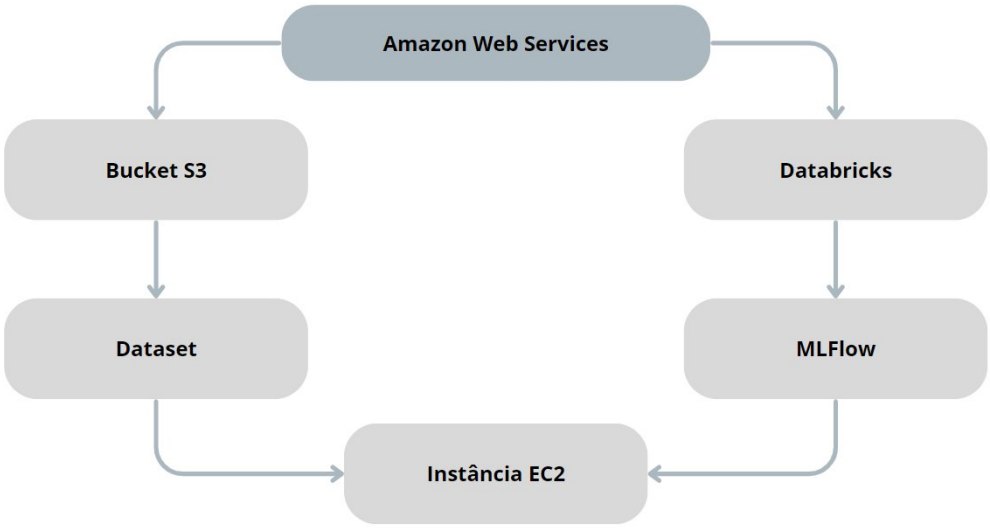
4.3 Modelos

Para auxílio na escolha e construção dos modelos, foi utilizada a solução Auto ML, que faz uma análise prévia da base de treinamento e gera os modelos que apresentam a melhor performance. O Auto ML é projetado para simplificar o processo de construção, treinamento e implantação de modelos de Machine Learning, permitindo que usuários sem conhecimentos avançados consigam treinar e utilizar modelos de alta qualidade (GOOGLE CLOUD, s.d).

Para realizar a análise foi utilizada a plataforma Databricks, dentro da AWS (Amazon Web Service). O Databricks é uma plataforma de análise de dados unificada, para engenharia de dados, Aprendizado de Máquina e Ciência de Dados. Dentro do Databricks, encontra-se a ferramenta MLFlow, que é uma plataforma de código aberto para gerenciamento do ciclo de vida de projetos de Machine Learning e é responsável por rodar o Auto ML.

Na AWS, a base de dados utilizada para treinamento e validação do modelo fica armazenada no Amazon S3 (serviço de armazenamento e recuperação de dados), “dentro” de um bucket, que é uma espécie de contêiner para armazenamento de objetos. O processamento é realizado pela EC2 (Amazon Elastic Compute Cloud), é um serviço de computação na nuvem fornecido pela AWS que permite que os usuários executem servidores virtuais, conhecidos como instâncias EC2. Portanto, para gerar um modelo de Machine Learning utilizando o MLFlow (Auto ML) na AWS, primeiro é necessário carregar o dataset que será utilizado no Amazon S3, depois acessar o MLFlow pelo Databricks e inserir os parâmetros e informações necessárias nos campos disponíveis para preenchimento, sendo um deles a seleção do dataset. O MLFlow irá executar o Auto ML, que irá realizar o pré-processamento dos dados, explorar os modelos os algoritmos de Machine Learning para encontrar o que tem melhor desempenho com o dataset utilizado, realizar o treinamento e geração dos modelos e códigos dos algoritmos (AWS). O fluxo realizado na geração dos modelos está ilustrado na Figura 7:

Figura 7 – Fluxo para geração de modelos utilizando Auto ML.



Fonte: Autoria própria.

O MLFlow constrói os algoritmos utilizando uma estrutura básica:

- Carregamento de dados: carrega o dataset que será utilizado.
- Pré-Processamento:
  - Seleciona apenas as colunas (atributos) com maior valor preditivo.
  - Realiza a formatação dos atributos, quando necessário.
  - Preenche os campos de valores numéricos que estão ausentes com o valor médio do atributo em questão.
  - Transforma os valores únicos de colunas categóricas (que possuem poucos valores exclusivos) em uma nova coluna no conjunto de dados.
  - Divide o dataset em conjunto de dados de treino, validação e teste (mantém as proporções do conjunto original de dados, nesse caso, as proporções entre interações legítimas e malignas é mantida).
- Treinamento do modelo
- Gera os resultados das métricas escolhidas.

Foram selecionados três modelos diferentes e comparados os desempenhos, com base nas métricas apresentadas na seção 4.2.

Os códigos utilizados estão disponíveis em:

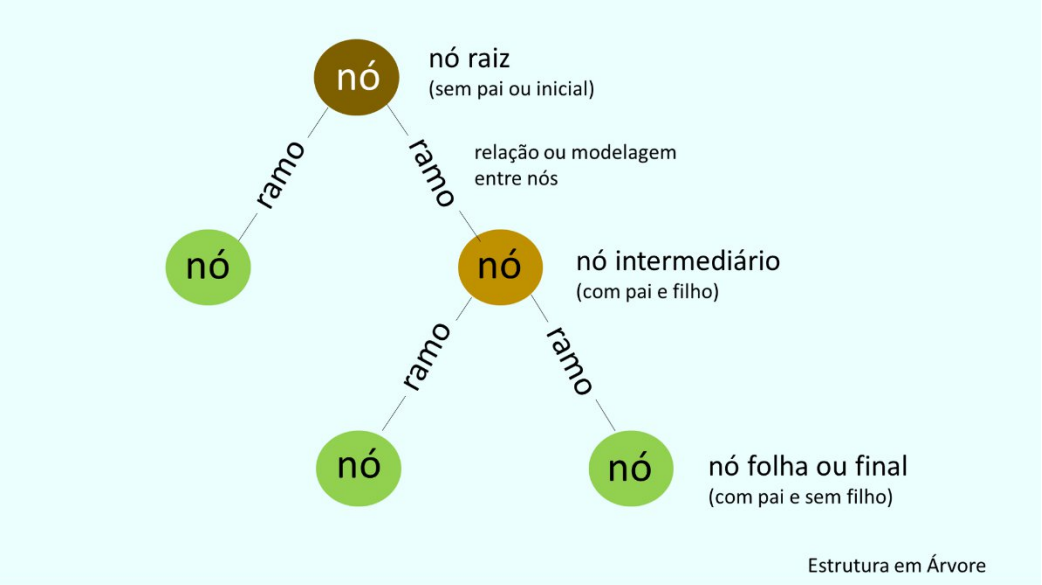
https://github.com/anabeatriizcunha/tcc_iot.

4.3.1 Árvore de Decisão

O algoritmo de Árvore de Decisão é utilizado para problemas de classificação e regressão. As Árvores de Decisão classificam os recursos com base nas suas características. Sua estrutura é composta por nós, ramos e folhas. Os nós são as características (atributos) a serem classificadas, os ramos são os valores que os nós podem assumir e as folhas são os terminais da estrutura da árvore. As folhas, ou também chamados nós folha, representam uma decisão para o conjunto de características do nó, em problemas de classificação essa decisão é chamada de classe (KOTSIANTIS et al., 2007). Os nós são divididos com base nos seus atributos, agrupando conforme as características comuns. A divisão é repetida em cada nó filho até atingir um critério de parada, pode ser quando todos os ramos do nó estão no mesmo agrupamento.

A estrutura básica do algoritmo está ilustrada na Figura 8 e definida abaixo (CARVALHO, 2005):

Figura 8 – Estrutura do algoritmo Árvore de Decisão.



Fonte: Colaborae, 2023.

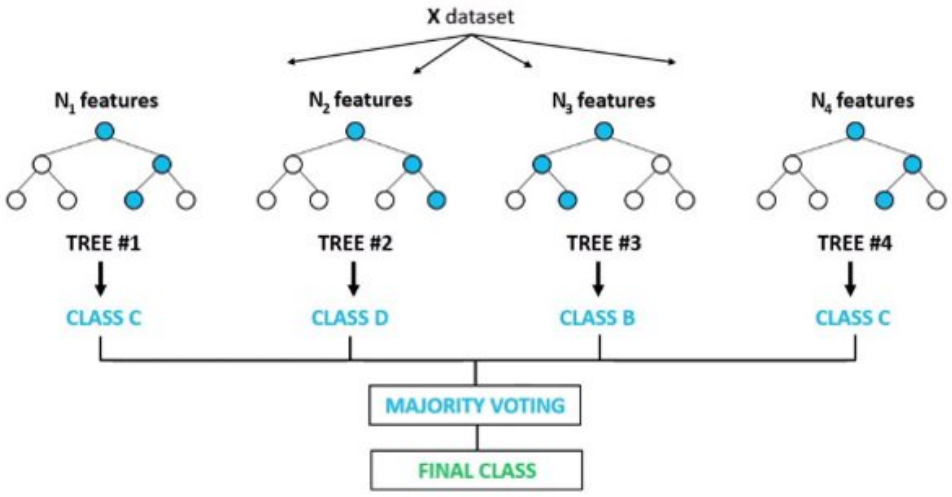
- Escolher um atributo: o algoritmo seleciona o melhor atributo de classificação para ser o nó raiz. O atributo que melhor separa os exemplos de treinamento em classes mais homogêneas é escolhido como nó raiz. As técnicas utilizadas para realizar essa seleção medem a entropia e a probabilidade de realizar uma classificação incorreta a partir de um determinado atributo.
- Estender a árvore: para cada valor de atributo, um ramo é adicionado à árvore. Dessa forma são criados os nós filhos para o nó em questão, cada nó filho representa diferentes condições associadas ao atributo.
- Passar os casos para o nó folha: cada exemplo no conjunto de dados é passado ao longo dos ramos da árvore, até atingirem uma folha, onde é atribuído à uma classe.
- Avaliação dos nós folha: ocorre a verificação de cada nó folha para ver se todos os exemplos que chegaram até ele pertencem à mesma classe. Se sim, o nó folha é rotulado com essa classe. Caso contrário, o processo é repetido, escolhendo um novo atributo para continuar a expansão da árvore.

4.3.2 Random Forest

Random Forest é um modelo que utiliza um grande número de árvores de decisão, onde cada uma delas é treinada individualmente com um subconjunto aleatório de dados. Pode ser definido como “um classificador composto por uma coleção de classificadores em forma de árvore {h(x, k), k = 1,...}, onde os {k} são vetores aleatórios independentes e identicamente distribuídos” (BREIMAN, 2001), ou seja, são k árvores de decisão, treinadas independentemente, cada uma com um subconjunto de dados aleatórios. Cada árvore emite um voto na classe que considera correta, a classe mais votada é considerada a previsão final do modelo. Como há baixa correlação entre as previsões de cada árvore, a chance de erro é menor que utilizando a árvore de decisão tradicional.

A estrutura básica do algoritmo está ilustrada na Figura 9 e definida abaixo (NETO, 2014):

Figura 9 – Estrutura do algoritmo Random Forest.



Fonte: FreeCodeCamp, 2020.

- O conjunto de dados é dividido aleatoriamente em subconjuntos, essa amostragem é chamada Bootstrap, que garante que cada subconjunto poderá ter registros incluídos mais de uma vez ou nenhuma vez.
- Para cada subconjunto amostrado, uma Árvore de Decisão é criada.
- Cada árvore é treinada de maneira independente com um subconjunto aleatório dos dados.

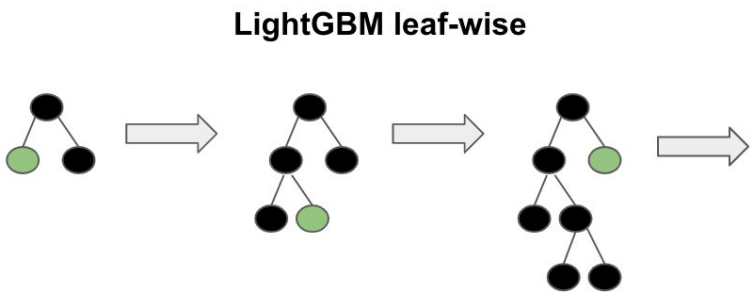
- Cada árvore contribui com um voto sobre qual classe o atributo deve pertencer. Os votos possuem pesos diferentes, quanto menor a similaridade entre as árvores, maior o peso do voto.
- Atribuição da classe mais votada ao atributo.

4.3.3 LightGBM

LightGBM ou Light Gradient Boosting Model é um modelo que também é derivado da árvore de decisão com a técnica de impulso de gradiente, que constrói as árvores de forma sequencial e cada árvore tenta corrigir o erro da anterior. Nele, as árvores são “empilhadas” em sequência, dessa forma elas são treinadas com as conclusões e os erros residuais das árvores anteriores. Os resultados das múltiplas árvores de decisão são somados para formar a saída prevista final (JU et al., 2019). Esse modelo é conhecido por sua rapidez no processamento, que se deve ao método de crescimento e construção de árvores, que escolhe a folha que melhor divide a amostra de dados para ser expandida, fazendo com que a árvore cresça verticalmente (SILVA, 2021).

A estrutura básica do algoritmo está ilustrada na Figura 10 e definida abaixo (TANG et al., 2020):

Figura 10 – Estrutura do algoritmo Random Forest.



Fonte: DataScience, 2020.

- Construção das Árvores de Decisão utilizando o método de crescimento e construção, que que divide os nós de forma mais eficiente.
    - Gradient-Based One-Side Sampling (GOSS):
        - Algoritmo que prioriza amostras com gradiente grande, que são amostras que apresentam maior erro, e por isso são importantes para dividir os nós de forma mais eficiente.
        - Seleciona as amostras com gradiente maior e uma porcentagem de amostras aleatórias com gradiente menor.
        - Supõe-se que as amostras com gradientes pequenos apresentam um erro de treinamento menor e já estão bem treinadas. Para manter a distribuição de dados, ao calcular o ganho de informações, o GOSS introduz um multiplicador constante para as instâncias de dados com pequenos gradientes. Assim, o GOSS alcança um bom equilíbrio entre reduzir o número de instâncias de dados e manter a precisão das árvores de decisão (DATARISK, 2022).
    - Exclusive Feature Building (EFB):
        - Algoritmo que combina características semelhantes em uma única característica.
- As previsões e erros das árvores vão sendo combinadas conforme as árvores são treinadas.
- O modelo final é selecionado com base na combinação das previsões de todas as árvores.

5 RESULTADOS

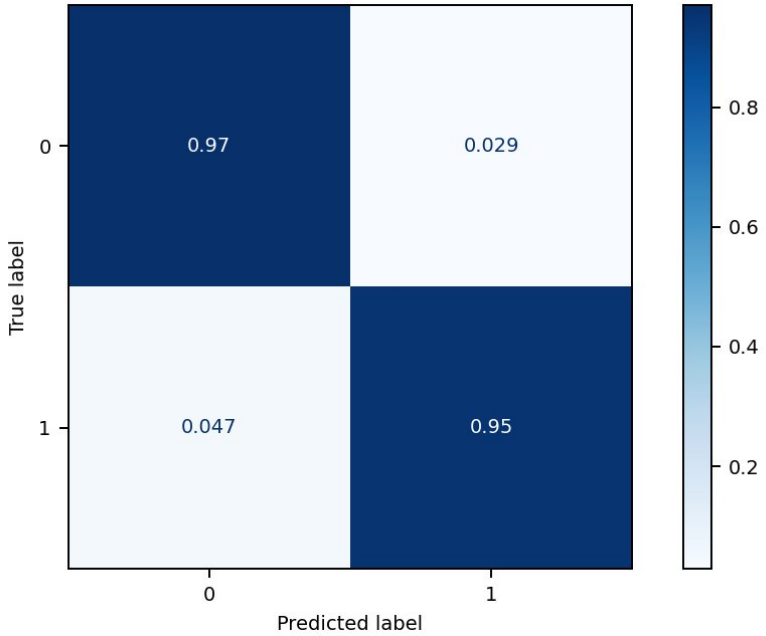
A profundidade de uma árvore de decisão é a extensão máxima da árvore e está relacionada com a complexidade do modelo utilizado. Árvores mais profundas apresentam um aprendizado mais profundo dos dados de treinamento, mas quanto mais profundas são as árvores, maior o custo computacional do modelo, além de que pode ocorrer Overfitting (o modelo se ajusta muito ao conjunto de treinamento e não tem o desempenho tão bom nas previsões de novos dados). Para o modelo Random Forest, a quantidade de árvores de decisão melhora a estabilidade do mo-

delo e reduz a variância e o Overfitting. Já para o LightGBM, quanto maior o número de árvores, maior a capacidade do algoritmo de captar padrões mais complexos.

Nos modelos Árvore de Decisão e LightGBM, a profundidade da árvore foi limitada a 9 e no modelo Random Forest foi limitada a 12. O modelo Random Forest utilizou 150 árvores de decisão e o LightGBM utilizou 13. Esses fatores não podem ser comparados para indicar qual possui melhor desempenho, mas é possível determinar que quanto mais árvores utilizadas e maior a profundidade das árvores, maior a capacidade computacional exigida.

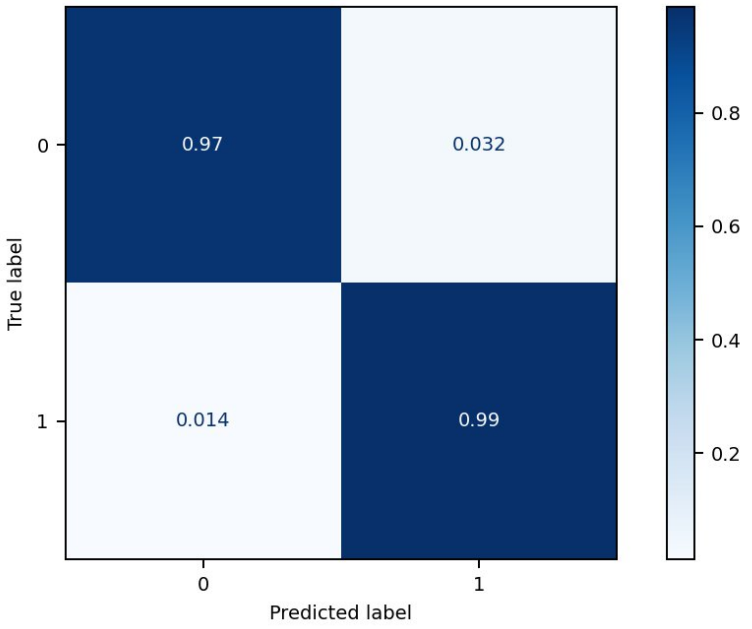
Uma Matriz de Confusão é uma ferramenta utilizada na avaliação do desempenho de modelos de Machine Learning. Nessa matriz é possível observar a frequência com que cada classe foi classificada corretamente e incorretamente. Os casos de Verdadeiros Positivos (True Label = 1 e Predicted Label = 1) e Verdadeiros Negativos (True Label = 0 e Predicted Label = 0) estão contidos na diagonal principal da matriz, sendo que os Verdadeiros Positivos indicam interação maliciosa e os Verdadeiros Negativos indicam interação legítima. Já na diagonal secundária, estão os casos de Falsos Positivos (True Label = 0 e Predicted Label = 1) e Falsos Negativos (True Label = 1 e Predicted Label = 0). O cálculo da acurácia é obtido através dessa matriz. As Figuras 11, 12 e 13 ilustram as Matrizes de Confusão dos modelos de Árvore de Decisão, Random Forest e LightGBM, respectivamente.

Figura 11 – Matriz de Confusão do modelo Árvore de Decisão.



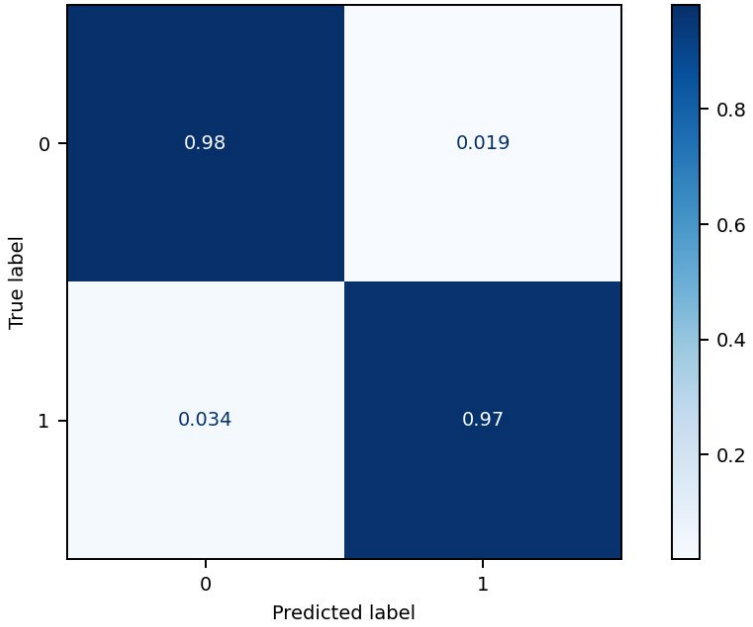
Fonte: Autoria própria.

Figura 12 – Matriz de Confusão do modelo Random Forest.



Fonte: Autoria própria.

Figura 13 – Matriz de Confusão do modelo LightGBM.



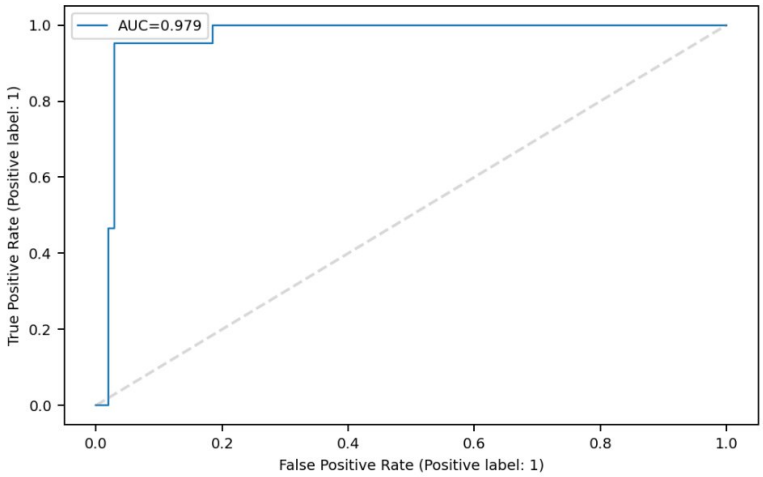
Fonte: Autoria própria.

Quanto mais próximo de 1 são os valores obtidos na diagonal principal, melhor a capacidade de classificação do modelo, enquanto os valores da diagonal secundária devem ser mais próximos de 0, indicando que a frequência com que as classes foram classificadas incorretamente foi pequena. Analisando as Figuras 11, 12 e 13 pode-se notar que os modelos tiveram desempenho semelhante, mas Random Forest e LightGBM apresentaram os maiores valores na diagonal principal.

Em relação a diagonal secundária, é importante que os casos de Falsos Negativos sejam os menores possíveis, pois esse caso indica que o algoritmo classificou uma interação maligna como legítima, ou seja, falhou na detecção de ataque. É possível observar que o modelo Random Forest apresenta menor valor de Falsos Negativos, indicando que dentre os modelos analisados é o que apresenta classificações mais exatas.

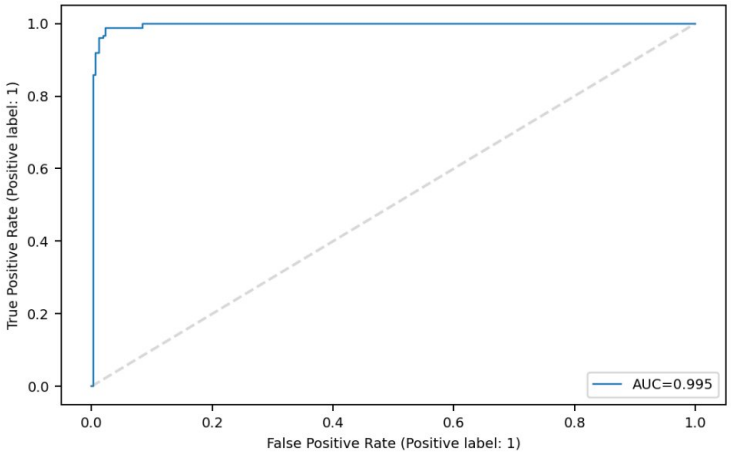
Outra maneira de avaliar o desempenho dos modelos é através da Curva ROC (Receiver Operating Characteristic ou Característica de Operação do Receptor), que mostra a relação entre a taxa de classificação de TPs e FPs. A área sob a Curva ROC indica a capacidade de classificação do modelo, quanto mais próximo de 1, maior a probabilidade de a classificação ser correta. As Figuras 14, 15 e 16 ilustram as Curvas ROC dos modelos de Árvore de Decisão, Random Forest e LightGBM, respectivamente. O valor AUC (Area Under the Curve, área sob a curva, em português) indicado nas imagens é a área sob a curva.

Figura 14 – Curva ROC do modelo de Árvore de Decisão.



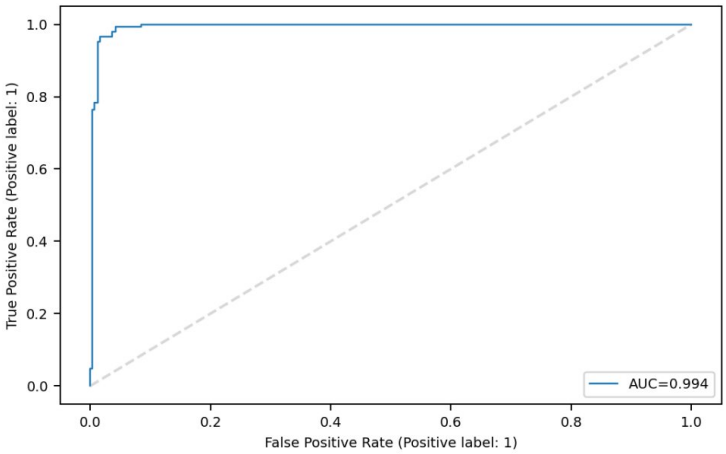
Fonte: Autoria própria.

Figura 15 – Curva ROC do modelo de Random Forest.



Fonte: Autoria própria.

Figura 16 – Curva ROC do modelo de LightGBM.

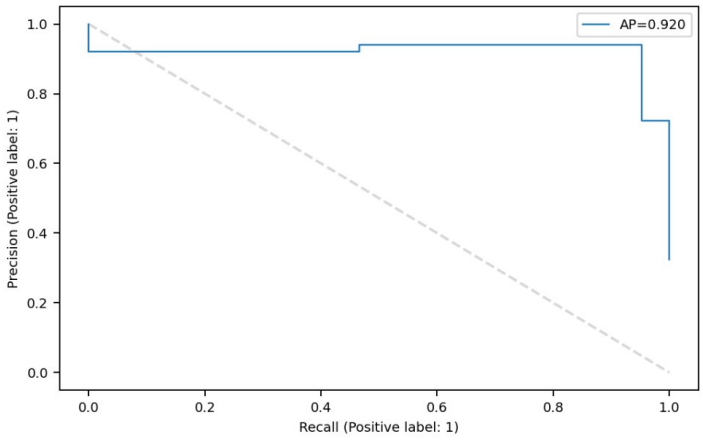


Fonte: Autoria própria.

Analisando as Figuras 13, 14 e 15 pode-se observar que os modelos Random Forest e LightGBM apresentaram melhores resultados, em especial, o modelo Random Forest apresentou maior AUC, indicando maior probabilidade de classificação correta.

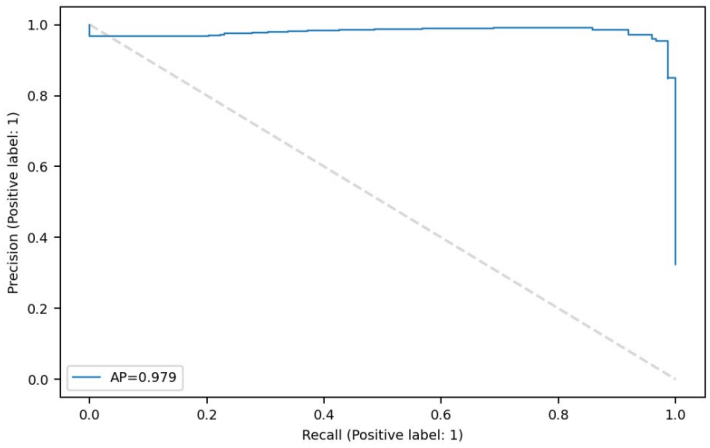
E por fim, tem-se a Curva de Precisão-Recall, também utilizada para análise de modelos, como o nome diz, essa curva mostra a relação entre a precisão e o recall e assim como nas Curvas ROC, quanto mais próximo de 1 é a área sob a curva, maior a probabilidade de a classificação ser correta. As Figuras 16, 17 e 18 ilustram as Curvas Precisão-Recall dos modelos de Árvore de Decisão, Random Forest e LightGBM, respectivamente. O valor AP indicado nas imagens é a área sob a curva.

Figura 17 – Curva Precisão-Recall do modelo de Árvore de Decisão.



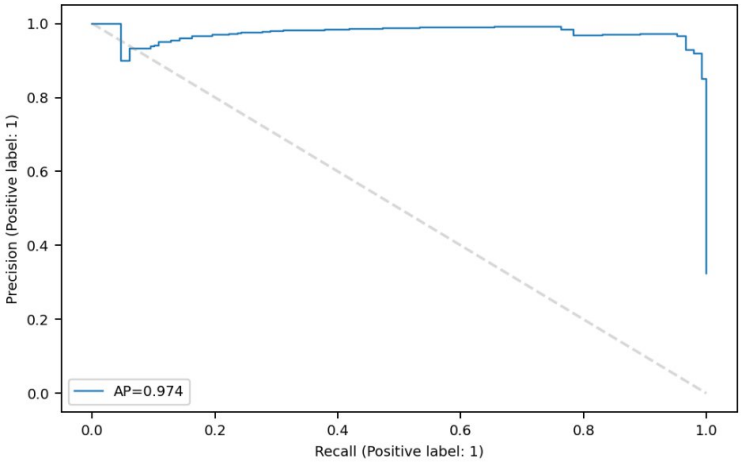
Fonte: Autoria própria.

Figura 18 – Curva Precisão-Recall do modelo de Random Forest.



Fonte: Autoria própria.

Figura 19 – Curva Precisão-Recall do modelo de LightGBM.



Fonte: Autoria própria.

Analisando as Figuras 17, 18 e 19 pode-se observar que novamente os modelos Random Forest e LightGBM apresentaram resultados muito similares, uma área sob a curva maior que o modelo Árvore de Decisão.

A Tabela 3 mostra os valores das métricas adotadas, obtidas a partir dos resultados mostrados acima:

Tabela 3 – Resultados por modelo.

Modelo	Acurácia	Precisão	Recall
Árvore de Decisão	0,96	0,94	0,95
Random Forest	0,97	0,93	0,98
LightGBM	0,97	0,95	0,96

Fonte: Autoria própria.

É possível observar na Tabela 3 que os três modelos demonstraram resultados satisfatórios na identificação e classificação dos ataques. O modelo Árvore de Decisão mostrou Acurácia e Recall menor do que os outros modelos e Precisão na média. Random Forest apresentou Acurácia maior que o modelo Árvore de Decisão e igual ao LightGBM, e valor de Recall maior. O modelo LightGBM obteve valor de Acurácia igual ao Random Forest e superior a Árvore de Decisão, maior valor de Precisão em relação aos demais e Recall na média.

É importante considerar que o desbalanceamento do conjunto de dados utilizado pode influenciar nas métricas, como 99% do conjunto de dados é de interações malignas, o algoritmo pode ficar enviesado para classificações malignas e apresentar dificuldades em reconhecer interações legítimas.

Através da análise de todos os elementos, tanto das Figuras 11 a 19, quanto da Tabela 3, pode-se concluir que todos os modelos alcançaram resultados satisfatórios nas classificações, tanto de Verdadeiros Positivos e Negativos (TP e TN) quanto de Falsos Verdadeiros e Negativos (FP e FN). Levando em conta todas as métricas, o modelo Random Forest se mostrou mais adequado para detecção de ataques DDoS em dispositivos IoT.

Em um cenário em que a aplicação de Machine Learning para a detecção de ataques DDoS ocorra em um dispositivo com pouca capacidade, seria mais vantajoso a utilização do modelo Árvore de Decisão, que possui uma estrutura mais simples e, portanto, exige menos capacidade que os outros dois modelos analisados. Como mostram os resultados, o modelo Árvore de Decisão apresentou um desempenho um pouco menor, porém não muito distante do Random Forest e LightGBM, a diferença no desempenho não é tão significativa se forem comparados a complexidade computacional dos modelos.

6. CONCLUSÃO

Os sistemas IoT são capazes de realizar a coleta e processamento de dados dos ambientes aos quais estão integrados, por isso, são utilizados em diversas aplicações, como coleta de dados, automatização de processos ou monitoramento de recursos. Com o crescimento do uso e desenvolvimento desses dispositivos e sua popularização, tanto na indústria tecnológica quanto na agricultura, meio ambiente e no dia a dia da população, é fundamental garantir a segurança e privacidade desses sistemas. Os ataques DDoS são um dos principais ataques na área da computação, e é a maior ameaça para a Internet das Coisas, podem causar sobrecarga e esgotamento de recursos no alvo, causando indisponibilidade de funcionamento. Por esses fatores é fundamental a detecção desse tipo de ataque.

Este trabalho apresentou uma avaliação do desempenho de modelos de Machine Learning para identificação de ataques DDoS em dispositivos IoT. Comparando os resultados obtidos pelas métricas escolhidas, é possível concluir que todos os modelos avaliados são aptos para realizar a identificação de ataques, todos apresentaram bons resultados, com a métrica acurácia maior ou igual a 96%. Fazendo uma análise de todas as métricas, matrizes e curvas, pode-se concluir que o modelo Random Forest apresentou um desempenho melhor e, portanto, é o mais adequado para a detecção de ataques DDoS em dispositivos IoT.

Esses resultados demonstram que é possível evitar ataques DDoS em dispositivos IoT a partir da identificação das interações. Contudo, ainda há muitos desafios na segurança e prevenção de ataques nesse tipo de dispositivo. Para trabalhos futuros, é interessante a avaliação de modelos de Machine Learning em diferentes cenários, utilizando outros tipos de dispositivos IoT, teste de outros algoritmos e para que a detecção de ataques seja mais completa, é importante a inclusão de outros tipos de ataques.

REFERÊNCIAS

AL-HADHRAMI, Yahya; HUSSAIN, Farookh. DDoS attacks in IoT networks: a comprehensive systematic literature review. World Wide Web (2021).

ANDREA, Ioannis; CHRYSOSTOMOU, Chrysostomos; HADJICHRISTOFI, George. Internet of Things: Security vulnerabilities and challenges. 2015 IEEE Symposium on

Computers and Communication (ISCC), 2015, pp. 180-187, doi: 10.1109/ISCC.2015.7405513.

ANTONAKAKIS, Manos; APRIL, Tim; BAILEY, Michael; BERNHARD, Matt; BURSZTEIN, Elie; COCHRAN, Jaime; DURUMERIC, Zakir, HALDERMAN, J; INVERNIZZI, Luca; KALLITSIS, Michalis; KUMAR, Deepak; LEVER, Chaz; MA, Zane; MASON, Joshua; MENSCHER, Damian; SEAMAN, Chad; SULLIVAN, Nick; THOMAS, Kurt; ZHOU; Yi. Understanding the Mirai Botnet. 26th USENIX Security Symposium. 16 à 18 de agosto, 2017, Vancouver, BC, Canada.

ASHTON, Kevin. "That ‘internet of things’ thing." RFiD Journal 22, no. 7 (2009): 97-114.

AWS. Databricks na AWS. Disponível em: https://aws.amazon.com/pt/solutions/partners/databricks/. Acesso em: 07/11/2023.

AWS. Soluções de Auto ML da AWS. Disponível em: https://aws.amazon.com/pt/machine-learning/automl/. Acesso em: 07/11/2023.

AWS. Perguntas frequentes sobre o S3. Disponível em: https://aws.amazon.com/pt/s3/faqs/. Acesso em: 07/11/2023.

BEZERRA, Vitor Hugo; COSTA, Victor G. Turrisi da; MARTINS, Ricardo Augusto; JUNIOR, Sylvio Barbon; MIANI, Rodrigo Sanches; ZARPELÃO, Bruno Bogaz. Providing IoT host-based datasets for intrusion detection research. In: SIMPÓSIO BRASILEIRO DE SEGURANÇA DA INFORMAÇÃO E DE SISTEMAS COMPUTACIONAIS (SBSEG), 18. , 2018, Natal. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2018 . p. 15 - 28.

BOCHIE, Kaylani; GILBERT, Mateus; GANTERT, Luana; BARBOSA, Mariana; MEDEIROS, Dianne; CAMPISTA, Miguel. Aprendizado profundo em redes desafiadoras: Conceitos e aplicações. Em Minicursos do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), 2020.

BREIMAN, Leo. Random Forests. Statistics Department, University of California, Berkeley, CA 94720, 2001.

BUYYA, R.; DASTJERDI, A. V. Internet of Things: principles and paradigms. New York: Elsevier, 2016.

CARVALHO, Deborah. Árvore de Decisão/Algoritmo genético para tratar o problema de pequenos disjuntos em classificação de dados. Tese para o programa de Pós-Graduação em Computação de Alto Desempenho/Sistemas Computacionais. Universidade Federal do Rio de Janeiro, 2005.

CARVALHO, Valdir; QUEIROZ, Ewerton; MENDONÇA, Júlio; CALLOU, Gustavo; ANDRADE, Emerson. Avaliação de Desempenho de Modelos Deep Learning para Detecção de Intrusão em Dispositvos IoT. 2021: Anais do XX Workshop em desempenho de sistemas computacionais e de comunicação.

COLABORAE. Árvore de Decisão. Disponível em: https://colaborae.com.br/blog/2023/07/19/arvore-de-decisao/. Acesso em: 29/11/2023.

CRUZ, Antonia. MECANISMO DE DETECÇÃO DE ATAQUES MIRAI BASEADO EM MACHINE LEARNING PARA SISTEMAS IOT. Programa de Pós-Graduação em Ciência da Computação, Mestrado acadêmico em Ciência da Computação. Universidade Estadual do Ceará, 2019.

CYBERMAGAZINE. How are DDoS attacks impacting businesses and services? Disponível em: https://cybermagazine.com/cyber-security/how-are-ddos-attacks-impacting-businesses-and-services. Acesso em 10/11/2023.

DATARISK. Gradiente Boosting: XGBoost vs LightGBM vs CatBoost. Disponível em: https://www.datarisk.io/gradient-boosting-parte-3-xgboost-vs-lightgbm-vs-catboost/. Acesso em: 06/12/2023.

DATASCIENCE. O que é GBM leve? Disponível em: https://datascience.eu/pt/aprendizado-de-maquina/o-que-e-gbm-leve/. Acesso em: 29/11/2023.

DOSHI, Rohan; APTHORPE, Noah; FEAMSTER, Nick. Machine Learning DDoS Detection for Consumer Internet of Things Devices. In 2018 IEEE Security and Privacy Workshops (SPW).

FOLHA DE S.PAULO. Google e Amazon dizem ter sofrido maior ataque hacker. Disponível em: https://www1.folha.uol.com.br/tec/2023/10/google-e-amazon-dizem-ter-sofrido-maior-ataque-hacker.shtml. Acesso em: 20/10/2023.

FREE CODE CAMP. Random Forest Classifier Tutorial: How to Use Tree-Based Algorithms for Machine Learning. Disponível em: https://www.freecode-camp.org/news/how-to-use-the-tree-based-algorithm-for-machine-learning/. Acesso em: 29/11/2023.

FRUSTACI, Mario; PACE, Pasquale; ALOI, Gianluca; FORTINO, Giancarlo. Evaluating Critical Security Issues of the IoT World: Present and Future Challenges, in IEEE Internet of Things Journal, vol. 5, no. 4, pp. 2483-2495, Aug. 2018, doi: 10.1109/JIOT.2017.2767291.

GE, Mengmeng; FU, Xiping; SYED, Naeem; BAIG, Zubair; TEO, Gideon; ROBLES-KELLY, Antonio. Deep Learning-based Intrusion Detection for IoT Networks. 2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC).

GOOGLE CLOUD. AutoML. Disponível em: https://cloud.google.com/automl?hl=pt-BR. Acesso em: 07/11/2023.

HUSSAIN, Fatima; HUSSAIN, Rasheed; HASSAN, Syed; HOSSAIN, Ekran. Machine Learning in IoT Security: Current Solutions and Future Challenges. IEEE Communications Surveys & Tutorials, v. 22, n. 3, terceiro trimestre de 2020.

IBM. O que é Deep Learning?. Disponível em: https://www.ibm.com/br-pt/topics/deep-learning. Acesso em: 06/12/2023.

JU, Yun; SUN, Guangyu; CHEN, Quanhe; ZHANG, Min; ZHU, Huixian; REHMAN, Mujeeb. A Model Combining Convolutional Neural Network and LightGBM Algorithm

for Ultra-Short-Term Wind Power Forecasting. School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China. 27 de fevereiro de 2019.

KOTSIANTIS, S. B.; ZAHARAKIS, I. D.; PINTELAS, P. E. Machine learning: a review of classification and combining techniques. Springer Science and Business Media B.V. 2007.

LEITE, Emiliano; MARTINS, Paulo; URSINI, Edson. A INTERNET das COISAS (IoT): Tecnologias e Aplicações. 2017 Brazilian Technology Symposium. School of Technology, University of Campinas (UNICAMP), Limeira-SP, Brazil.

MAHESH, Batta. Machine Learning Algorithms – A Review. International Journal of Science and Research (IJSR), 2018.

MAHJABIN, Tasnuva; XIAO, Yang; SUN, Guang; JIANG, Wangdong. A survey of distributed denial-of-service attack, prevention, and mitigation techniques. International Journal of Distributed Sensor Networks, v. 13, n. 12, 2017.

MAHMOUD, R., YOUSUF, T., ALOUL, F. and ZUALKERNAN, I. Internet of Things (IoT) Security: Current Status, Challenges and Prospective Measures. 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST), pp. 336 - 341, 2015.

MALIK, Shikha; CHAUHAN, Ruchi. Securing the Internet of Things using Machine Learning: A Review. 2020 IEEE International Conference on Convergence to Digital World – Quo Vadis (ICCDW 2020).

MONARD, Maria C.; BARANAUSKAS, José A. Conceitos sobre Aprendizado de Máquina. Sistemas Inteligentes para Engenharias, p. 39-56.

MORAES, Alexandre; HAYASHI, Victor. “Segurança em IoT” Alta Books; 1ª edição (10 agosto 2021).

NETO, Cesare. Potencial de técnicas de mineração de dados para o mapeamento de áreas cafeeiras. INPE, São José dos Campos, 2014.

Oracle. O que é Machine Learning?. Disponível em: https://www.oracle.com/br/artificial-intelligence/machine-learning/what-is-machine-learning/. Acesso em: 11/06/2023.

PERING, T.; FARRINGTON, K.; DAHM, T. Taming the IoT: Operationalized Testing to Secure Connected Devices. Computer, v. 51, n. 6, p. 90-94, jun. 2018. DOI: 10.1109/MC.2018.2701633.

PERLIN, Tiago; NUNES, Raul; KOZAKEVICIUS, Alice. Detecção de Anomalias em Redes de Computadores e o uso de Wavelets. Revista Brasileira de Computação Aplicada, [S. l.], v. 3, n. 1, p. 2-15, 2011. DOI: 10.5335/rbca.2013.1313. Disponível em: https://seer.upf.br/index.php/rbca/article/view/1313.

PORTELA, Ariel L. C.; COSTA, Wanderson L.; GOMES, Rafael L.. Detecção de Ataques DDoS em redes IoT usando Redes Neurais e Seleção de Características. In: Workshop de trabalhos de iniciação cientifica e de graduação – Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), 39., 2021, Uberlândia. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2021. p. 225-232. ISSN 2177-9384.

ROMÁN-CASTRO, R.; LÓPEZ, J.; GRITZALIS, S. Evolution and Trends in IoT Security. Computer, v. 51, n. 7, p. 16-25, jul. 2018. DOI: 10.1109/MC.2018.3011051.

ROSE, Karen; ELDRIDGE, Scott; CHAPIN, Lyman. The internet of things: An overview. The internet society (ISOC), 2015.

SANTOS, Bruno; SILVA, Lucas; CELES, Clayson; NETO, João; PERES, Bruna; VIERA, Marcos; VIERA, Luiz; GOUSSEVSKAIA, Olga; LOUREIRO, Antonio. Internet das Coisas: da Teoria à Prática. In XXXIV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, chapter 1, pages 1–50.

SCHILLER, E.; AIDOO, A.; FUHRER, J.; STAHL, J.; ZIÖRJEN, M.; STILLER, B. Landscape of IoT security. Computer Science Review, v. 44, 100467, 2022. ISSN 1574-0137.

SELVARAJ, Vigneshwaran. Distributed Denial of Service Attack Detection, Prevention and Mitigation Service on Cloud Environment. Journal of Computer Engineering and Information Technology, v. [volume], n. [número], p. [páginas], ago. 2018.

SHINDE, Pramila P.; SHAH, Seema. A Review of Machine Learning and Deep Learning Applications. 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-6, doi: 10.1109/ICCUBEA.2018.8697857.

SILVA, Catarina. Previsão de valor Brix: Aplicação de algoritmos de Machine Learning. Mestrado em métodos quantitativos para a decisão econômica e empresarial, Universidade de Lisboa, 2021.

SUO, Hui; WAN, Jiafu; ZOU, Caifeng; LIU, Jianqi. Security in the Internet of Things: A Review. 2012 International Conference on Computer Science and Electronics Engineering, 2012, pp. 648-651. DOI: 10.1109/ICCSEE.2012.373.

TANG, Mingzhu; ZHAO, Qi; DING, Steven; WU, Huawei; LI, Linlin; LONG, Wen; HUANG, Bin. An Improved LightGBM Algorithm for Online Fault Detection of Wind Turbine Gearboxes. Energies, v. 13, n. 4, p. 807, 2020.

TEIXEIRA, Fernando; VIEIRA, Gustavo; FONSECA, Pablo; PEREIRA, Fernando; WONG, Hao; NOGUEIRA, José; OLIVEIRA, Leonardo. Defending internet of things against exploits. IEEE Latin America Transactions, IEEE, v. 13, n. 4, p. 1112–1119, 2015.

THE GUARDIAN. DDoS attack that disrupted internet was largest of its kind in history, experts say. The Guardian, 26 out. 2016, disponível em: https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet. Acesso em: 03/06/2023.

XIAO, Liang; WAN, Xiaoyue; LU, Xiaozhen; ZHANG, Yanyong; WU, Di. IoT Security Techniques Based on Machine Learning: How Do IoT Devices Use AI to Enhance Security? IEEE Signal Processing Magazine, v. 35, n. 5, p. 41-49, set. 2018. DOI: 10.1109/MSP.2018.2825478.